

Universidad ORT Uruguay

Facultad de Ingeniería

Migración gráfica de sistema SCADA

(Supervisory Control And Data Acquisition)

Entregado como requisito para la obtención del título de
Ingeniero en Sistemas

Andrés Martínez – 168474

Renato Moraes – 150952

Tutor: José Szyman

2017

Declaración de autoría

Nosotros, Andrés Martínez y Renato Moraes, declaramos que el trabajo que se presenta en esa obra es de nuestra propia mano. Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos el proyecto de grado de fin de curso de la carrera Ingeniería en Sistemas;
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad;
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra;
- En la obra, hemos acusado recibo de las ayudas recibidas;
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros;
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.



Andrés Martínez
23/08/2017



Renato Moraes
23/08/2017

Agradecimientos

Antes que nada agradecer a nuestro tutor José Szyman tanto por su disposición y constante desvelo a lo largo del proyecto, como por su honestidad intelectual en todos los aportes realizados al mismo.

A la universidad ORT, en particular al Laboratorio de Investigación en Sistemas de Información (LISI) por el asesoramiento brindado desde el momento que el proyecto era una idea.

Agradecer a la empresa UTE, principalmente a la unidad de Software de Telecontrol-DIS que en nombre de Lidia Dávila, nos permitieron participar de tan importante proyecto, y también por la entera disposición y colaboración de cada uno de los integrantes de la unidad en cada momento que lo necesitamos: Diego García, Eduardo Vidal, Gabriel Paladino, Martín Lamaison y Sebastián Ferreira.

Y un especial agradecimiento a Paloma y a Antonio, que siempre regalaron una sonrisa a pesar de las innumerables ausencias, y a Silvana y a Ángela por la el apoyo incondicional.

Abstract

Las grandes empresas dedican recursos en sistemas que monitorean y controlan sus principales activos. Dichos sistemas, denominados SCADA, van quedando obsoletos conforme al desarrollo tecnológico existente, y esto obliga a las empresas a cambiar o actualizar sus sistemas sin perder prestaciones ni calidad en sus procesos productivos.

El presente proyecto hizo hincapié en la migración entre sistemas SCADA utilizados por la empresa de energía eléctrica UTE, que monitorea y opera la red eléctrica nacional mediante dichos sistemas.

Cuando se habla de migración se tiende a pensar en mantener las funcionalidades de un sistema de origen en el sistema de destino, sin embargo consistió en un proceso ordenado, bajo un estándar de trabajo ITIL V3, que nos permitió de forma automatizada la migración gráfica para luego integrar con los desarrollos de UTE, sin olvidar las constantes pruebas en el proceso.

El resultado obtenido, por un lado, nos dejó experiencia en la gestión de proyectos, aplicación de estándares y desarrollo de software; y por otro, al cliente se le dejó la representación gráfica de la red de Canelones y Florida, y un proceso documentado que contiene piezas de software listos para ser replicados en la migración de una de las redes eléctricas más importantes del cliente, como lo es la red de Montevideo.

Palabras clave

SCADA, UTE, Mirage, ITIL, Mímico, Unifilar, Celdas de Corte, Elementos de Corte, Elementos Comandables, Estaciones, Subestaciones, Puntos Intermedios, Reenganches, PAT o Puesta a Tierra, Interruptores, Conectividad, Red Eléctrica, Consola, Migración.

Índice

Declaración de autoría.....	2
Agradecimientos	3
Abstract.....	4
Palabras clave	5
Índice	6
Índice de Figuras	12
Índice de Tablas	16
1 Introducción	18
1.1 Sistema SCADA.....	18
1.2 UTE como empresa y la utilización de sistemas SCADA	20
1.3 Origen del proyecto	21
2 Descripción del proyecto.....	23
2.1 Objetivos generales.....	23
2.1.1 Académicos.....	23
2.1.2 Proyecto.....	23
2.1.3 Producto	23
2.2 Descripción del problema	24
2.2.1 Situación actual	24
2.2.2 Descripción de la solución	25
2.2.3 Principales beneficios	27
2.3 Riesgos del proyecto	28
2.4 Estudio de factibilidad.....	29
2.4.1 Introducción.....	29
2.4.2 Factibilidad técnica	29
2.4.3 Factibilidad económica	29
2.4.4 Factibilidad operativa	30
2.4.5 Factibilidad legal	30
2.4.6 Conclusión	30
3 Ingeniería de Requerimientos	31
3.1 Introducción	31
3.2 Técnicas y resultados de especificación	31
3.3 Entrevistas.....	31

3.4	Observación	33
3.5	Estudio de documentación	33
3.6	Justificación de la estrategia de relevamiento.....	33
3.7	Requerimientos definidos.....	34
3.8	Requerimientos funcionales	35
3.8.1	Estaciones, subestaciones y puesto de conexión.	35
3.8.2	Puntos intermedios.....	39
3.8.3	Unifilares en general.....	41
3.8.4	Mímicos Comunicaciones	47
3.8.5	Módulo de pruebas automatizadas.....	48
3.9	Requerimientos no funcionales	53
3.10	Casos de uso y casos de prueba.....	54
3.10.1	Estaciones, subestaciones y puesto de conexión.	54
3.10.2	Puntos Intermedios.	59
3.10.3	Unifilares en general.....	61
3.10.4	Módulo de pruebas automatizadas.....	73
4	Arquitectura	78
4.1	Atributos de calidad	78
4.1.1	Testeabilidad	79
4.1.2	Modificabilidad	80
4.1.3	Usabilidad	81
4.2	Descripción de la Arquitectura	82
4.2.1	Presentación de componentes	82
4.2.2	Detalle de componentes Software Migración.	83
4.2.3	Diagramas de Secuencia	85
4.2.4	Diagrama de entidades Oracle	88
4.2.5	Diagrama de entidades SQL Server	90
4.2.6	Despliegue del sistema	91
4.3	Conclusiones y lecciones aprendidas.....	92
5	Gestión de la calidad	93
5.1	Objetivos	93
5.2	Planificación	94
5.3	Aseguramiento de la calidad (SQA)	94

5.3.1	Actividades con foco en el proceso	95
5.3.1.1	Reuniones mensuales.....	95
5.3.1.2	Instancias ORT de revisión de todos los procesos.....	95
5.3.2	Actividades con foco en el producto	95
5.3.2.1	Estándares de producto	95
5.3.3	Estándares de proceso.....	97
5.3.4	Revisiones	97
5.3.4.1	Revisiones con expertos y de equipo	97
5.3.4.2	Revisiones de código	98
5.3.5	Validaciones	98
5.3.6	Pruebas	98
5.3.6.1	Pruebas unitarias.....	99
5.3.6.2	Pruebas de integración.....	99
5.3.6.3	Pruebas cruzadas.....	100
5.3.6.4	Pruebas Automáticas.....	100
5.3.6.5	Pruebas en campo	101
5.3.6.6	Pruebas de enlace:	102
5.4	Métricas	102
5.4.1	Métricas del producto:	102
5.4.2	Métricas del proyecto:.....	103
5.5	Análisis de las métricas	103
5.5.1	Análisis de métricas del producto	103
5.5.2	Análisis de Métricas del producto	105
5.6	Conclusiones y lecciones aprendidas.....	107
6	Gestión de la configuración.....	108
6.1.1	Definiciones	108
6.1.2	Identificación de los elementos de configuración.....	109
6.1.3	Control de versiones	109
6.1.4	Organización de documentos	110
6.1.5	Organización del código fuente	110
6.2	Herramientas a utilizar.....	111
6.2.1	Repositorio para gestión de documentos.....	111
6.2.2	Repositorio para gestión de código	112

6.3	Seguimientos.....	113
7	Gestión del Proyecto	114
7.1	Metodologías de gestión del proyecto.	114
7.1.1	Proceso	114
7.1.2	Ciclo de Vida	115
7.1.3	Metodología de proceso de construcción de software.....	116
7.1.3.1	Fase 0.....	116
7.1.3.2	Fase 1.....	116
7.1.3.3	Fase 2.....	118
7.1.4	Roles	119
7.1.4.1	Asignaciones de Roles	120
7.2	Gestión de la comunicación	121
7.2.1	Interesados del proyecto	121
7.2.2	Herramientas de comunicación.....	121
7.2.2.1	Comunicación interna	122
7.2.2.2	Comunicación externa.....	123
7.2.3	Planificación de la comunicación.....	123
7.2.4	Conclusiones y lecciones aprendidas	124
7.3	Gestión del tiempo.....	125
7.3.1	Estimación de recursos y duración	125
1.1.1.1.	Desarrollo del cronograma	127
7.3.2	Métricas relacionadas a la gestión del tiempo.....	127
7.4	Gestión de riesgos.....	134
7.4.1	Análisis del impacto y riesgo al negocio	134
7.4.1.1	Lista de riesgos	136
a)	Riesgos asociados al desarrollo:.....	136
b)	Riesgos asociados a la tecnología:	136
c)	Riesgos asociados al equipo:.....	136
d)	Riesgos asociados al cliente:	137
e)	Riesgos asociados al proyecto:.....	137
7.4.2	Análisis del impacto y riesgo al negocio	137
7.4.3	Monitorización del riesgo	140
7.4.3.1	Requerimientos (R3).....	142

7.4.3.2	Centro de cómputos (R14)	142
7.4.3.3	Migración front end (R15)	143
7.4.3.4	Migración Base de datos (R16)	143
7.4.3.5	Documentación (R20)	144
7.4.4	Conclusiones y lecciones aprendidas	145
8	Gestión de Lanzamiento	146
8.1	Finalidad y Objetivos:	146
8.2	Políticas de gestión de lanzamiento e implementación	146
8.2.1	Componentes.	147
8.2.2	Unidades de lanzamiento	148
8.2.3	Opciones de implementación y consideraciones.	148
8.2.4	Modelos de lanzamiento e implementación.	149
8.3	Actividades, métodos y técnicas del proceso	150
8.3.1	Planificación de lanzamiento y despliegue	151
8.3.2	Creación y prueba de la versión	154
8.3.2.1	Puesta en producción y crear documentación	154
8.3.2.2	Pruebas de servicio	155
8.3.3	Despliegue	156
8.3.4	Revisar y Cerrar	158
8.3.4.1	Revisar y cerrar una transición de servicio	158
8.4	Conclusiones	159
9	Gestión del cambio	160
10	Conclusiones	167
10.1	Conclusiones del proyecto	167
10.2	Conclusiones de equipo de trabajo	168
10.3	Conclusiones de producto	168
10.4	Conclusiones de proceso	169
10.5	Lecciones aprendidas	169
11	Referencias bibliográficas	170
12	Glosario	172
13	Anexos	178
13.1	Anexo 1 – <i>Framework</i> ITIL v3	178
13.2	Anexo 2 – Funcionalidades destacables SCADA X	180

13.3	Anexo 3 - Variables SCADA Mirage	181
13.4	Anexo 4 – Color por nivel de tensión en SCADA Mirage.....	184
13.5	Anexo 5 – SIO (Sistema integrado de Operación)	184
13.6	Anexo 6 – Plan de Calidad.....	187
13.7	Anexo 7 – Revisiones formales.	189
13.8	Anexo 8 – Revisiones informales con técnicos UTE.	191
13.9	Anexo 9 – Revisiones informales entre equipo.....	192
13.10	Anexo 10 – Aprobación de sistema de <i>Testing</i>	193
13.11	Anexo 11 – Pruebas unitarias.....	194
13.12	Anexo 12 – Pruebas de Integración en unifilar.....	195
13.13	Anexo 13 – Pruebas Automáticas.....	196
13.14	Anexo 14 – Pruebas de Campo.....	199
13.15	Anexo 15 – Pruebas de Enlace	201
13.16	Anexo 16 – Ejemplo de tabla de errores detectados en corridas	202
13.17	Anexo 17 – Reuniones Semanales.....	204
13.18	Anexo 18 – Archivo Propietario.....	206
13.19	Anexo 19 – Diapositiva Heurística para estimación.	207
13.20	Anexo 20 – Cronograma del proyecto.....	207
13.21	Anexo 21 – Planilla de horas semanales.	211
13.22	Anexo 22 – Tabla evaluación Riesgos.	213
13.23	Anexo 23 – Políticas de Lanzamiento.	216
13.24	Anexo 24 – Procedimiento de <i>Rollback</i>	237
13.25	Anexo 25 – Manual de Usuario SegmentToMimx	240
13.26	Anexo 26 – Manual <i>Testing</i> Grafico Mirage.....	249
13.27	Anexo 27 – Editor Gráfico Mirage	254
13.28	Anexo 28 – Pruebas de Integración de los tres sistemas.	261

Índice de Figuras

Figura 1.1: Diagrama de un Sistema SCADA.	19
Figura 2.1: Sistema SCADA de telecontrol.	25
Figura 2.2: Diagrama de la solución del proyecto.	26
Figura 2.3: Unifilar o Mímico de estación en SCADA Mirage	28
Figura 3.1: Método de procesar los lineamientos definidos.	32
Figura 3.2: Esquema de composición de objetos entre SCADA.....	36
Figura 3.3: Algunos ejemplos de configuraciones de celdas de corte.....	36
Figura 3.4: Esquema de elementos dinámicos de una celda	37
Figura 3.5: Reenganches y sus respectivas barras	38
Figura 3.6: Conector reenganche a barra.	38
Figura 3.7: Mímico donde se representa un seccionalizador de línea.	39
Figura 3.8: Mímico donde se representa un reconectador de línea	40
Figura 3.9: Algunos ejemplos de elementos dinámicos	40
Figura 3.10: Esquema de elementos comandables.	41
Figura 3.11: Pantallas del SIO.	42
Figura 3.12: Instalación alimentada desde varios orígenes.....	43
Figura 3.13: Ejemplo celdario.	43
Figura 3.14: Unidades (kV).....	44
Figura 3.15: Cuadro de instalación.	44
Figura 3.16: Botón que abre la vista de Comandos hab/deshab relé.	45
Figura 3.17: Vista del Comando Hab/deshab relé en SCADA Mirage.....	45
Figura 3.18: Comando Hab/deshab relé.....	45
Figura 3.19: DPF (Detector de paso de falta).....	46
Figura 3.20: Nombre variable que representa en un mímico.	47
Figura 3.21: Botón de comunicaciones en un mímico Mirage	47
Figura 3.22: Panel de comunicaciones	47
Figura 3.23: Celda con su respectiva referencia en el celdario	48
Figura 3.24- Ejemplo de elementos pertenecientes a sus respectivas celdas.	49
Figura 3.25: Mímico de instalación y de comunicaciones.	50
Figura 3.26: Mímico de distrito	51
Figura 3.27: Conectividad	52
Figura 4.1: Atributos de calidad priorizados.....	78
Figura 4.2: Patrón composite aplicado	79
Figura 4.3: Patrón Fábrica Aplicado.	80
Figura 4.4: Elección de objetos candidatos.	81
Figura 4.5: Diagrama de componentes y conectores	82
Figura 4.6: Diagrama de detalles de C&C	84
Figura 4.7: Diagrama de Secuencia.....	85
Figura 4.8: Diagrama de secuencia con interacción usuario	86
Figura 4.9: Diagrama de entidades Oracle	88
Figura 4.10: Diagrama de entidades SQL Server	90

Figura 4.11: Diagrama de Despliegue	91
Figura 5.1: Errores detectados por ejecuciones.	105
Figura 5.2: Horas dedicados vs horas planificadas	106
Figura 6.1: Estructura de carpetas con documentación	110
Figura 6.2: Estructura de carpetas con los planes de gestión	110
Figura 6.3: Repositorio Google Drive.....	111
Figura 6.4: Repositorio BitBucket.	112
Figura 6.5: Repositorio Mirage.	112
Figura 7.1: Diagrama genérico del desarrollo evolutivo incremental	116
Figura 7.2: Diagrama de Fases y Tareas	118
Figura 7.3: Resumen cronograma.....	127
Figura 7.4: Desviación respecto a la estimación	129
Figura 7.5: Distribución de las horas trabajadas según el proceso.	129
Figura 7.6: Relación entre el tiempo dedicado a gestión y programación.	130
Figura 7.7: Tiempo dedicado a programación.....	130
Figura 7.8: Tiempo dedicado a testing.	131
Figura 7.9: Tiempo dedicado a Conversión.	131
Figura 7.10: Desfasaje entre estimación y el tiempo real.	132
Figura 7.11: Esfuerzo mes total	132
Figura 7.12: Esfuerzos Individuales.	133
Figura 7.13: Evolución Riesgos.....	141
Figura 7.14: Evolución Principales Riesgos	142
Figura 8.1: Esquema desarrollo proyecto Migración.....	147
Figura 8.2: Unidades de lanzamiento.	148
Figura 8.3: Esquema de Implementación BIG BANG.	149
Figura 8.4: Fases de la gestión de lanzamiento y despliegue.	151
Figura 9.1: Proceso gestión del cambio	161
Figura 9.2: Subproceso Valorar y Evaluar el cambio.....	162
Figura 13.1: Volúmenes ITIL V3	178
Figura 13.2: Objetos de corte individuales en SCADA X	180
Figura 13.3: Único Objeto de corte en Mirage	180
Figura 13.4: Identificación de variables	182
Figura 13.5: Objeto seleccionado CC_SECINTSECSPTDW	183
Figura 13.6: Vistas objeto CC_SECINTSECSPTDW	183
Figura 13.7: Elemento compuesto.....	184
Figura 13.8: Color por nivel de tensión	184
Figura 13.9: Vista de operación y manejo de incidencias en sistema SIO.....	185
Figura 13.10: Ejemplo de un unifilar de estación en el sistema SIO.....	186
Figura 13.11: Ventana de trabajo del sistema SIO.....	186
Figura 13.12: Revisión informal	191
Figura 13.13: Coordinación Reunión Informal.....	192
Figura 13.14: Ventana Selección Objetos	194
Figura 13.15: Sistema SCADA X.....	195
Figura 13.16: Sistema SCADA Mirage	195

Figura 13.17: Software pruebas automáticas.....	196
Figura 13.18: Asignación cambiada en Boton Protocolo 101.....	196
Figura 13.19: Mapa de Distrito	197
Figura 13.20: Sin continuidad seleccionada	197
Figura 13.21: Con continuidad seleccionada.....	198
Figura 13.22: Amarillo energizado - Verde sin energizar.....	198
Figura 13.23: Primer comando probado en campo.....	199
Figura 13.24: CMD Canelones	200
Figura 13.25: Pruebas de Enlace.....	201
Figura 13.26: Medidas SCADA X	201
Figura 13.27: Medidas en SCADA Mirage	201
Figura 13.28: Archivo Propietario	206
Figura 13.29: Heurísticas para estimación.....	207
Figura 13.30: Cronograma Fase 0	207
Figura 13.31: Cronograma Fase 1 – Iteración 1	208
Figura 13.32: Cronograma Fase 1 – Iteración 2	208
Figura 13.33: Cronograma Fase 1 – Iteración 3	209
Figura 13.34: Cronograma Fase 1 – Iteración 4	209
Figura 13.35: Cronograma Fase 1 – Iteración 5	210
Figura 13.36: Cronograma Fase 1 – Iteración 6	210
Figura 13.37: Cronograma Fase 2	211
Figura 13.38: Situación actual, SCADA X funcionando.	237
Figura 13.39: Situación futura, funcionamiento a pleno del CMD Canelones	238
Figura 13.40: Sistema funcionando de forma parcial en ambos SCADA.	239
Figura 13.41: Archivos necesarios para la ejecución.....	241
Figura 13.42: Ámbito de trabajo recomendado.	244
Figura 13.43: Ventana principal de SegmentToMimx.....	245
Figura 13.44: Borrado de selección de objetos de celdas	245
Figura 13.45: Ventana de selección de objetos Mirage.....	246
Figura 13.46: Ventana para nombrar Mímico.....	247
Figura 13.47: Ventana de remplazo de mímico.	247
Figura 13.48: Archivos a ser utilizados en el testing.....	250
Figura 13.49: Ventana principal de Testing Gráfico Mirage.....	252
Figura 13.50: Ventana de pruebas con los errores encontrados.....	253
Figura 13.51: Formulario para agregar excepción.	253
Figura 13.52: Menú Vista en el Editor Mirage.	254
Figura 13.53: Selección de la vista CANFLO.visx	254
Figura 13.54: Ventana Principal del Editor	255
Figura 13.55: Ventana de gestión de archivos en el repositorio Mirage.....	255
Figura 13.56: Menú Vista en el contexto de la vista CANFLO.....	256
Figura 13.57: Lista de Mímicos con opción de agregar y eliminar.....	256
Figura 13.58 Formulario de agregado de Mímico.	257
Figura 13.59: Mímico en modo edición.....	257
Figura 13.60: Barra de herramientas.....	258

Figura 13.61: Barra edición de textos.....	258
Figura 13.62: Barra de agrupamiento de elementos gráficos.	258
Figura 13.63: Menú Biblioteca en el editor.....	258
Figura 13.64: Biblioteca de objetos Mirage.....	258
Figura 13.65: Mapeo de variables Mirage a objetos.	259
Figura 13.66: Mímico de la estación EST5017 pintado por nivel de tensión.....	259
Figura 13.67: Mímico de la estación EST5017 pintado por conectividad.....	260
Figura 13.68: Unifilar Mirage conectado EST5044 La Balsa.	261

Índice de Tablas

Tabla 3-1: Listado de Requerimientos	35
Tabla 3-2: Caso de Prueba CP1	56
Tabla 3-3: Caso de Prueba CP2	57
Tabla 3-4: Caso de Prueba CP2.1	57
Tabla 3-5: Caso de Prueba CP3	58
Tabla 3-6: Caso de Prueba CP4	59
Tabla 3-7: Caso de Prueba CP5	59
Tabla 3-8: Caso de Prueba CP6	60
Tabla 3-9: Caso de Prueba CP7	61
Tabla 3-10: Caso de Prueba CP8	62
Tabla 3-11: Caso de Prueba CP9	63
Tabla 3-12: Caso de Prueba CP10	64
Tabla 3-13: Caso de Prueba CP10.1	65
Tabla 3-14: Caso de Prueba CP11	65
Tabla 3-15: Caso de Prueba CP12	66
Tabla 3-16: Caso de Prueba CP13	67
Tabla 3-17: Caso de Prueba CP14	68
Tabla 3-18: Caso de Prueba CP15	69
Tabla 3-19: Caso de Prueba CP16	70
Tabla 3-20: Caso de Prueba CP17	71
Tabla 3-21: Caso de Prueba CP18	72
Tabla 3-22: Caso de Prueba CP19	73
Tabla 3-23: Caso de Prueba CP20	74
Tabla 3-24: Caso de Prueba CP21	75
Tabla 3-25: Caso de Prueba CP22	75
Tabla 3-26: Caso de Prueba CP23	76
Tabla 3-27: Caso de Prueba CP24	77
Tabla 3-28: Caso de Prueba CP25	77
Tabla 4-1: Catálogo Elementos C&C	82
Tabla 4-2: Catálogo de elementos de detalles de C&C	84
Tabla 4-3: Catálogo Elementos Entidades	89
Tabla 4-4: Catalogo de elementos de entidades de SQL Server	90
Tabla 4-5: Catálogo de elementos de Despliegue	91
Tabla 5-1: Duración en horas para conversión automático.	99
Tabla 5-2: Duración en horas para conversión manual.	99
Tabla 5-3: Errores detectados en pruebas de enlace	104
Tabla 5-4: Duración tareas estimadas vs duración real.	105
Tabla 6-1: Elementos de configuración.....	109
Tabla 6-2: Seguimientos modificaciones	113
Tabla 7-1: Responsabilidades por rol.....	120
Tabla 7-2: Roles asignados.....	120

Tabla 7-3: Herramientas de comunicación de uso interno.....	122
Tabla 7-4: Herramientas de comunicación de uso externo.	123
Tabla 7-5: Planificación de la comunicación	124
Tabla 7-6: Estimación software.....	125
Tabla 7-7: Horas disponibles por integrante.....	126
Tabla 7-8: Estimación duración tareas.....	128
Tabla 7-9: Desviaciones en duración tareas	128
Tabla 7-10: Probabilidad de Ocurrencia	135
Tabla 7-11: Impacto de los riesgos	135
Tabla 7-12: Riesgos asociados al desarrollo.....	136
Tabla 7-13: Riesgos asociados a la tecnología	136
Tabla 7-14: Riesgos asociados al equipo.....	136
Tabla 7-15: Riesgos asociados al cliente	137
Tabla 7-16: Riesgos asociados al proyecto	137
Tabla 7-17: Plan de respuesta.....	138
Tabla 7-18: Plan de contingencia y disparadores	139
Tabla 7-19: Matriz de Probabilidad e Impacto	140
Tabla 7-20: Niveles de Prioridades	140
Tabla 7-21: Riesgos altos.....	141
Tabla 8-1: Niveles de configuración para construir y probar.	153
Tabla 8-2: Preguntas que se deben responder al planificar la implementación.	154
Tabla 10-1: Cantidad de líneas XML.....	168
Tabla 13-1: Plan de calidad	188
Tabla 13-2: Errores Detectados	203
Tabla 13-3: Interesados Proyecto	204
Tabla 13-4: Reuniones Semanales	205
Tabla 13-5: Horas Semanales.....	213
Tabla 13-6: Evaluación Riesgos	213
Tabla 13-7: Nomenclaturas Riesgos.....	215
Tabla 13-8: Revisiones	216
Tabla 13-9: Propietario DOC Lanzamiento.....	216
Tabla 13-10: Abreviaturas	217
Tabla 13-11: Convención Lanzamiento.....	223
Tabla 13-12: Administración de configuración	225
Tabla 13-13: Roles y Responsabilidades.	230
Tabla 13-14: Matriz RACI.	232
Tabla 13-15: KPIs.....	234
Tabla 13-16: Verificación Entregables.....	236

1 Introducción

El presente documento describe el proyecto “Migración gráfica de sistema SCADA”, realizado como requisito para la obtención del título de Ingeniero en Sistemas de la Universidad ORT Uruguay.

1.1 Sistema SCADA

Un sistema SCADA (*Supervisory Control And Data Acquisition*) es un software que permite monitorear, controlar y supervisar procesos industriales a distancia desde una pantalla de un computador [1].

La misión del SCADA es facilitar la retroalimentación en tiempo real con los dispositivos de campo como son los sensores y actuadores, controlando el proceso automáticamente.

Consta de tres partes:

- ✚ Los sensores, que proveen diferente tipo de información.
- ✚ La inteligencia, que toma decisiones a partir de los datos proporcionados por los sensores, de forma automática o por un usuario.
- ✚ Por último, los actuadores, que se encargan de ejecutar las decisiones.

En la industria, un sistema SCADA se utiliza para proveer toda la información que se genera en el proceso productivo (supervisión, control de calidad, control de producción, almacenamiento de datos, etc.). No solo sirve para recolectar datos de los diferentes sensores, sino que también permite su gestión e intervención, por ejemplo posibilita a distancia cortar el riego en una cosecha de arroz, debido a que ha empezado a llover. De existir un sensor que indique que el suelo no requiere más humedad, el sistema puede ser programado para dejar de regar.

Un sistema SCADA también podría utilizarse para uso doméstico, por ejemplo para encender el aire acondicionado (acción realizada por un actuador, que enciende el aire mediante una señal remota) si la temperatura ambiente es superior a 20 grados (dato proporcionado por un sensor de temperatura). Pero estos pequeños procesos no son implementados mediante un sistema SCADA debido al costo de las licencias de operación.

En el caso de UTE los sensores son los diferentes medidores de parámetros eléctricos de la red (tensión, corriente, temperatura, etc.) y diferentes tipos de alarmas (falta de corriente, puerta abierta, etc.), mientras que los actuadores básicamente son los elementos de corte automáticos a distancia.

“Corte” refiere a una interrupción en un circuito eléctrico, por ejemplo cortar la energía en una estación dada, para realizar mantenimiento.

Los elementos de corte pueden ser:

- ✚ Únicamente de maniobra, permitiendo el corte o la reposición en el tramo de la red en el cual los interruptores actúan.
- ✚ O pueden ser de maniobra y protección, que son aquellos que se activan para proteger equipos, personas o tramos de red. Por ejemplo, un rayo en la red eléctrica de UTE podría causar daño en diferentes electrodomésticos en los hogares, pero los elementos de protección actúan rápido e impiden que la sobre tensión llegue al usuario final, que puede ser nuestro hogar o una línea costosa de producción en una industria.

A continuación podremos observar el diagrama de un Sistema SCADA.

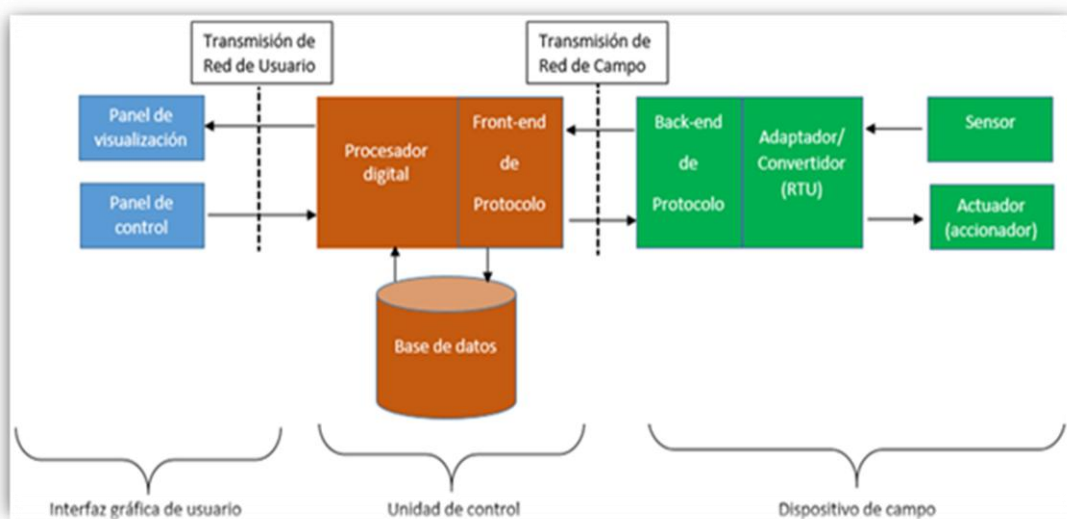


Figura 1.1: Diagrama de un Sistema SCADA.

Desde la perspectiva del software, cada Sistema SCADA está compuesto de los siguientes módulos.

- **Interfaz gráfica:** Pantalla por donde se visualizan los diferentes parámetros y permite la iteración usuario SCADA.
- **Front end de protocolo:** Parte del software que interpreta los datos recibidos desde la interfaz gráfica o desde el *back end*.
- **Back end de protocolo:** Parte del software que interpreta los diferentes datos de los sensores para enviarlos al *front end*, también interpreta los datos recibidos para enviarlos a los actuadores.
- **Base de datos:** Donde se guardan todos los parámetros recolectados por los sensores.

1.2 UTE como empresa y la utilización de sistemas SCADA

La Administración Nacional de Usinas y Trasmisiones Eléctricas (UTE), es una empresa propiedad del Estado uruguayo que se dedica a las actividades de generación, transmisión, distribución y comercialización de energía eléctrica (utilizando la red eléctrica), prestación de servicios anexos y consultoría [2].

En lo que respecta al proyecto, el rol desempeñado por UTE es el de cliente, ya que es para quien se hizo el desarrollo del mismo y quien hizo y hará su usufructo.

La red eléctrica tiene como propósito suministrar electricidad desde los proveedores hasta los consumidores. La misma se compone por las siguientes subredes eléctricas:

Red de generación: Constituidas por las plantas generadoras que mayormente están localizadas cerca de una fuente de agua, y alejadas de áreas pobladas. Por lo general son muy grandes, para así aprovecharse de la economía de escala. A la energía eléctrica generada se le incrementa su tensión la cual se va a conectar con la red de transmisión.

Red de transmisión: La red de transmisión transporta la energía a grandes distancias, pasando por nodos llamados estaciones de transmisión, las cuales son instalaciones eléctricas diseñadas para manejar y transformar niveles de tensión de entre 60 y 150 kV (kilo volts), llegando así hasta el consumidor final que en la mayoría de los casos con estaciones de la red de distribución.

Red de distribución: Se compone de instalaciones capaces de bajar la tensión, a la cual llamamos estaciones de distribución, para luego hacer llegar la energía eléctrica hasta la subestación, que se trata de una instalación que reduce aún más el nivel de tensión donde la energía llegue finalmente al servicio requerido, como por ejemplos, a los hogares, industrias o comercios [3].

En UTE, la utilización principal de los sistemas de tipo SCADA es para controlar a distancia instalaciones eléctricas, tanto la recepción de estados de elementos, alarmas y medidas de una instalación, como el envío de comandos.

Nuestro proyecto se enmarca dentro del área o red de distribución de UTE, que es la encargada de la red de baja tensión, que va desde los 230V a los 60 kV. Internamente al área de distribución se la denomina CMD (Centro de Maniobra de Distribución).

El área o red de distribución está dividida en las siguientes 5 gerencias:

- CMD Montevideo: Comprende el departamento de Montevideo y el centro de maniobra se localiza en la propia ciudad de Montevideo.
- CMD Centro: Comprende los departamentos de Canelones y Florida, ubicándose en la ciudad de Canelones el centro de maniobra.

- CMD Paysandú: Con sede en la ciudad de Paysandú, el CMD Abarca la zona norte del país, comprendiendo los departamentos al norte del Río Negro: Artigas, Salto, Rivera, Tacuarembó, Paysandú y Río Negro.
- CMD San José: Ubicado en la ciudad de San José de Mayo, el CMD maniobra la zona oeste del país que comprende los departamentos de Colonia, Soriano, Flores, Durazno y San José.
- CMD Maldonado: Se ubica en la ciudad de Maldonado y abarca la zona este del país, es decir, a los departamentos de Cerro Largo, Treinta y Tres, Lavalleja, Rocha y Maldonado.

Actualmente se están utilizando dos sistemas SCADA en los diferentes CMD para controlar a distancia las instalaciones de todo el país. Con uno de los sistemas SCADA, al cual llamaremos Mirage [4], se controlan remotamente tres CMD: Paysandú, San José y Maldonado.

Los CMD de Montevideo y Centro son monitoreados por otro sistema SCADA, que por razones de competencia directa con el SCADA Mirage y para que el proyecto no incurra en un acto desleal de explotación de la reputación ajena [5], lo llamaremos SCADA X.

1.3 Origen del proyecto

La idea de los integrantes del equipo era la de desarrollar un proyecto con una utilidad real, y ambos integrantes trabajando en empresas del rubro eléctrico, comenzaron a indagarse distintas necesidades del negocio. Mientras tanto en UTE se presentó la necesidad de migrar todo el telecontrol¹ del sistema X al sistema Mirage.

Dicha migración significaba para UTE, entre otras cosas, pasar toda la estructura de instalaciones ya probadas y operativas de un sistema que se viene manteniendo por casi 20 años a otro sistema sin perder operatividad, máxime cuando la misma tiene un alcance sobre más de la mitad de los clientes del país.

La migración de SCADA se compone de tres partes: por un lado, la migración del *front end*, que contiene la información del protocolo de comunicación de cada instalación; por otro lado, la migración de la base de datos alfanumérica que contiene las señales que llegan de campo; y finalmente la migración de la interfaz gráfica de todas las instalaciones eléctricas.

Fue entonces que se le planteó a UTE Distribución el proyecto de la migración del sistema SCADA X al sistema SCADA Mirage en los CMD Montevideo y Centro (Durante el proyecto nos vamos a referir como Canelones, es como UTE lo denomina internamente). El

¹ Telecontrol: Según la real academia española, por telecontrol se refiere a mando de un aparato o sistema, ejercido a distancia.

proceso consistirá en construir una pieza de software que permita migrar la parte gráfica e integrar con los sistemas migrados de *front end* y base de datos, testear el producto y supervisar la puesta en producción. En lo que respecta a la migración del *front end* y de la base alfanumérica corrió por cuenta de UTE, obviamente que existió coordinación entre los dos proyectos.

2 Descripción del proyecto

En este capítulo se realiza una descripción global de nuestro proyecto. En el mismo se incluyen los principales objetivos perseguidos, un detalle del problema identificado, así como nuestro entendimiento sobre la mejor solución del mismo.

2.1 Objetivos generales

Se definen los objetivos necesarios para cumplir con los entregables del proyecto final de carrera, teniendo en cuenta el interés de los miembros del proyecto. Por medio de estos, es posible dar seguimiento y evaluar el cumplimiento de los mismos.

2.1.1 Académicos

Aplicar los conocimientos adquiridos a lo largo de la carrera con el fin de gestionar el proceso de migración de un sistema. Para dicha migración nos apoyaremos en la construcción de una pieza de software, pruebas robustas y gestiones basadas en recomendaciones establecidas por ITIL. Ver Anexo 1 – *Framework* ITIL v3

Enfrentarse a un problema real con mucha responsabilidad, sin margen a errores, por tratarse de un proyecto a implementarse en UTE, en los sistemas de control de maniobras de la red eléctrica, que comprende la mayor cantidad de clientes del país (Montevideo, Canelones y Florida). Las lecciones aprendidas nos acercarán a las responsabilidades que afrontaremos como profesionales.

2.1.2 Proyecto

Cumplir con los plazos establecidos y acordes al cronograma de trabajo definido.

Lograr que el equipo se auto gestione adecuadamente, promoviendo el trabajo en equipo, la comunicación y el compromiso.

Elaborar un proyecto que por su calidad que posibilite una devolución y calificación satisfactorias por parte del comité evaluador.

2.1.3 Producto

Lograr migrar el sistema actual SCADA X al nuevo sistema SCADA Mirage, en los CMD Montevideo y Centro, y para eso deberá seguir las políticas de lanzamiento de ITIL con el fin de lograr un producto de calidad, fuertemente testeado por software y testeado en campo.

Para lograrlo se deberá construir una pieza de software que permita la migración de un sistema SCADA a otro, y mediante un sistema de pruebas robustas se deberá cumplir con

los atributos de calidad requeridos, para algo tan importante como se trata del control de distribución de la energía para más de la mitad de la población de Uruguay.

2.2 Descripción del problema

En el presente punto se expone la situación actual con la problemática que dio origen al proyecto, la descripción de la solución y los principales beneficios a obtenerse.

2.2.1 Situación actual

Nuestro cliente, como se ha descrito anteriormente se dedica a las actividades de generación, transmisión, distribución y comercialización de energía eléctrica, utilizando la red eléctrica para suministrarla a sus clientes.

Para cada una de estas áreas existe una gerencia independiente en UTE, además de la gerencia comercial, por lo tanto, UTE está compuesto de las siguientes 4 gerencias [6]:

- Generación: encargada de generar la energía.
- Transmisión: encargada de la red eléctrica que está por encima de los 60 kV (kilo voltios).
- Distribución: encargada de la red de baja tensión, que va de los 230V a los 60 kV. Denominado internamente como CMD (Centro de Maniobra de Distribución).
- Comercial: es la encargada de comercializar la energía.

Nuestro proyecto es desarrollado para el CMD. La propuesta consiste en realizar la migración de un sistema SCADA a otro, sin perder funcionalidades.

Los sistemas implicados son, como sistema origen, un SCADA que corre sobre la plataforma UNIX; y como sistema destino el SCADA Mirage de la empresa uruguaya Controles S.A. que corre sobre la plataforma Windows.

Cada CMD debe controlar los siguientes tipos de instalaciones:

-  Estaciones
-  Subestaciones
-  Puestos de conexión
-  Puntos intermedios

En los CMD de Paysandú, Maldonado y San José, el sistema utilizado es SCADA Mirage, tanto en los sistemas locales como en el sistema central remoto. En los CMD de Montevideo y Centro, cada estación o subestación tienen un SCADA Mirage, pero los dos controles centrales están bajo el sistema SCADA X.

Esta es otra de las causas por la cual es necesario contar con un programa de migración entre SCADA, ya que el sistema local utilizado es distinto al sistema SCADA usado en los CMD.

Para lograr el control existe un SCADA local en cada instalación, Servidores, y un SCADA remoto que controla todos los puntos en cada CMD, (ver figura 2.1 – Sistema SCADA de telecontrol).

Como se muestra en la siguiente figura (imagen obtenida de manual SISTEMA DE TELECONTROL MIRAGE elaborado por UTE), el sistema consta de las siguientes partes:

- ✚ Gabinete Principal del Centro de Control.
- ✚ Puestos de trabajo (MMI's) del Centro de Control.
- ✚ Unidades Remotas de Telecontrol (URT o RTU – *Remote Terminal Unit*).
- ✚ Equipos de comunicaciones.

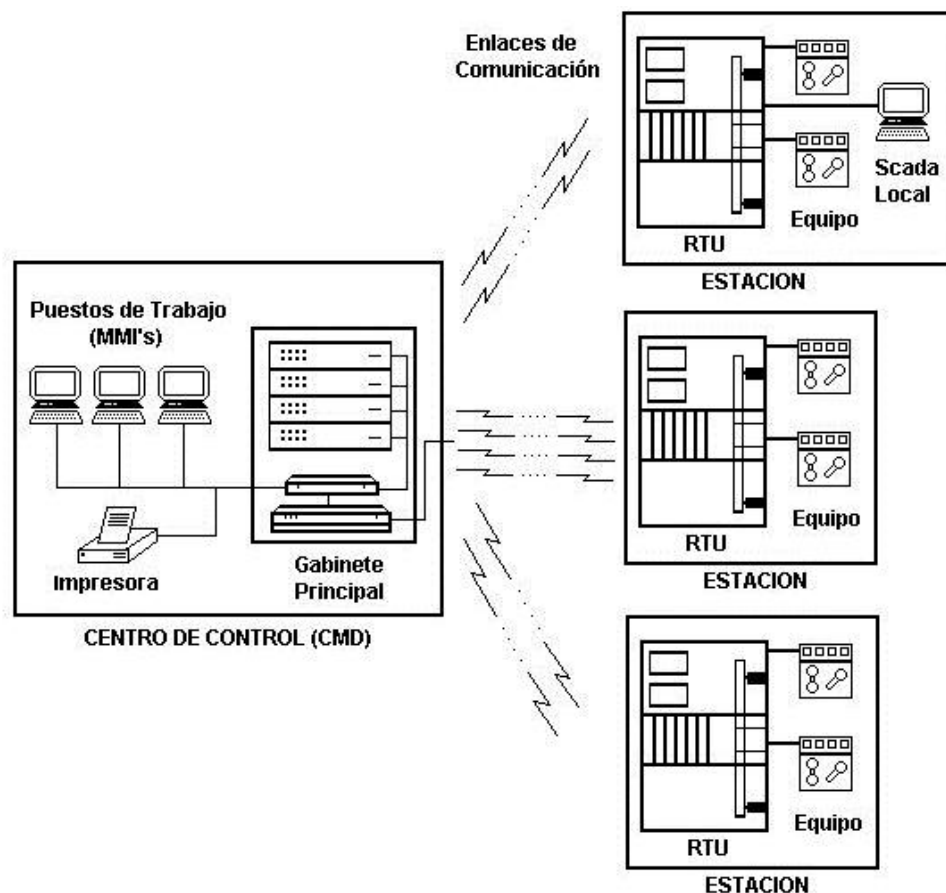


Figura 2.1: Sistema SCADA de telecontrol.

2.2.2 Descripción de la solución

El producto de nuestro proyecto consiste en realizar un proceso de migración completo, no se trata únicamente de una pieza de software.

Consecuentemente la solución para lograr un proceso exitoso de migración estará compuesta de las siguientes etapas:

- ✚ Construcción de una pieza de software que permita migrar de manera automática la parte gráfica del Sistema SCADA.
- ✚ Supervisar la migración del *front end* y la base de datos, que la realizará el cliente.
- ✚ Integrar el resultado del software de migración con el *front end* y la base de datos.
- ✚ Construir pieza de software que permita probar de manera automática la solución integrada (gráfica, *front end*, base de datos).
- ✚ Coordinar pruebas en campo.
- ✚ Asistir en la migración (puesta en producción).

La pieza de software deberá interpretar un archivo propietario y una base de datos Oracle; a partir de estos dos elementos el software deberá generar un archivo XML, que representa toda la parte gráfica.

Para que pueda funcionar el nuevo sistema, además se deberá migrar el *front end* y la base de datos que, si bien corren por parte del cliente, habrá de controlarse que las mismas se desarrollen en tiempo y forma, para luego poder integrar los tres sistemas.

Conjuntamente se deberá construir un sistema de pruebas automáticas, el sistema leerá el archivo XML y buscará errores de diferentes tipos.

Para coordinar las pruebas en campo se deberá hacerlo con 4 grupos a la vez; grupo de software, grupo de hardware, grupo de protecciones y CMD. No son muchas las instalaciones en las que se pueden realizar dichas pruebas, esperamos poder al menos probar en una de cada tipo y cubrir el 10% de las mismas.

Durante todo el proceso se deberá seguir las políticas de lanzamiento establecidas por el marco de trabajo ITIL. Ver en figura 2.2 diagrama de la solución, proyecto migración SCADA.

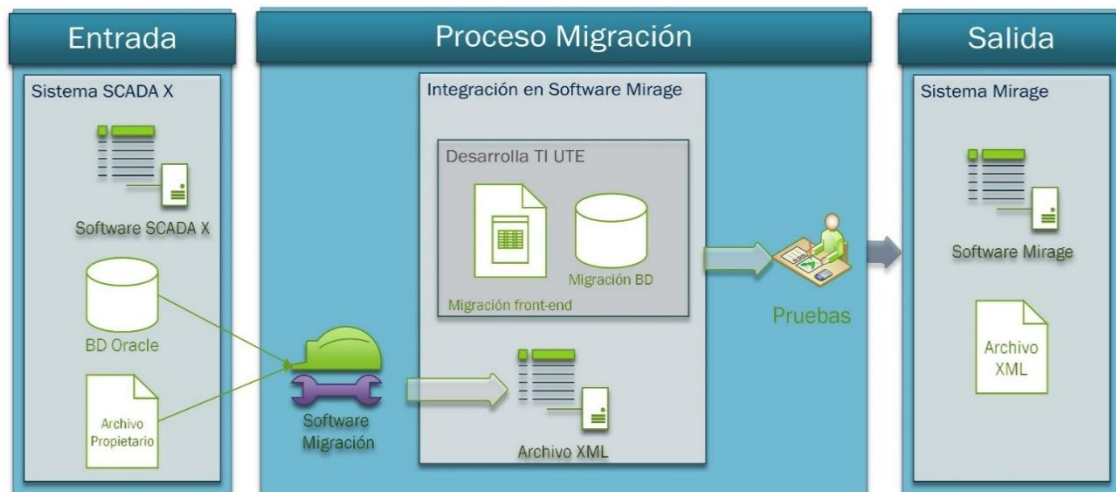


Figura 2.2: Diagrama de la solución del proyecto.

2.2.3 Principales beneficios

A continuación se enumeran los beneficios de la **migración**:

- UTE se ahorrará una suma importante de gastos por conceptos de licencias y mantenimientos.
- Se contará con el mismo SCADA en todos los CMD y, localmente, en todas las estaciones y subestaciones de Distribución facilitando el mantenimiento.
- Los SCADA dedicados a la industria eléctrica, utilizan el término "unifilar" o "mímico" en el área de visualización para referirse a los esquemas que representan cualquier instalación eléctrica. Estos unifilares en el sistema Mirage son persistidos en archivos con una estructura XML, característica que facilita la exportación a archivos SVG y, a su vez, estos permiten visualizaciones dinámicas vía web de dichos unifilares. Por lo tanto, tener la representación web de las instalaciones es otra utilidad importante que se obtendría a partir de la migración de los unifilares.
- Permite desarrollo de automatismos² para redes inteligentes. En SCADA X hay que solicitar al proveedor que lo desarrolle. En cambio, la empresa que desarrollo Mirage "Controles SA" le suministro un módulo, por el cual UTE mismo puede efectuar el desarrollo de automatismos.
- UTE desarrolló un simulador para el SCADA Mirage, dicho simulador es utilizado para ensayar maniobras, herramienta muy útil para entrenamiento de operadores en CMD. No existe herramienta similar en SCADA X.
- Finalmente, tener una migración automatizada de un sistema en el cual se viene trabajando desde hace casi 20 años, permitiría obtener un CMD totalmente operativo en un SCADA Mirage con mínimo trabajo.

² Automatismos: En el contexto de las redes eléctricas se refiere algoritmos programados, que evitan la intervención humana sobre la red.

A lo largo de la documentación utilizaremos el término unifilar y mímico de forma indistinta. A modo ilustrativo en la figura 2.3 vemos un unifilar del sistema Mirage.

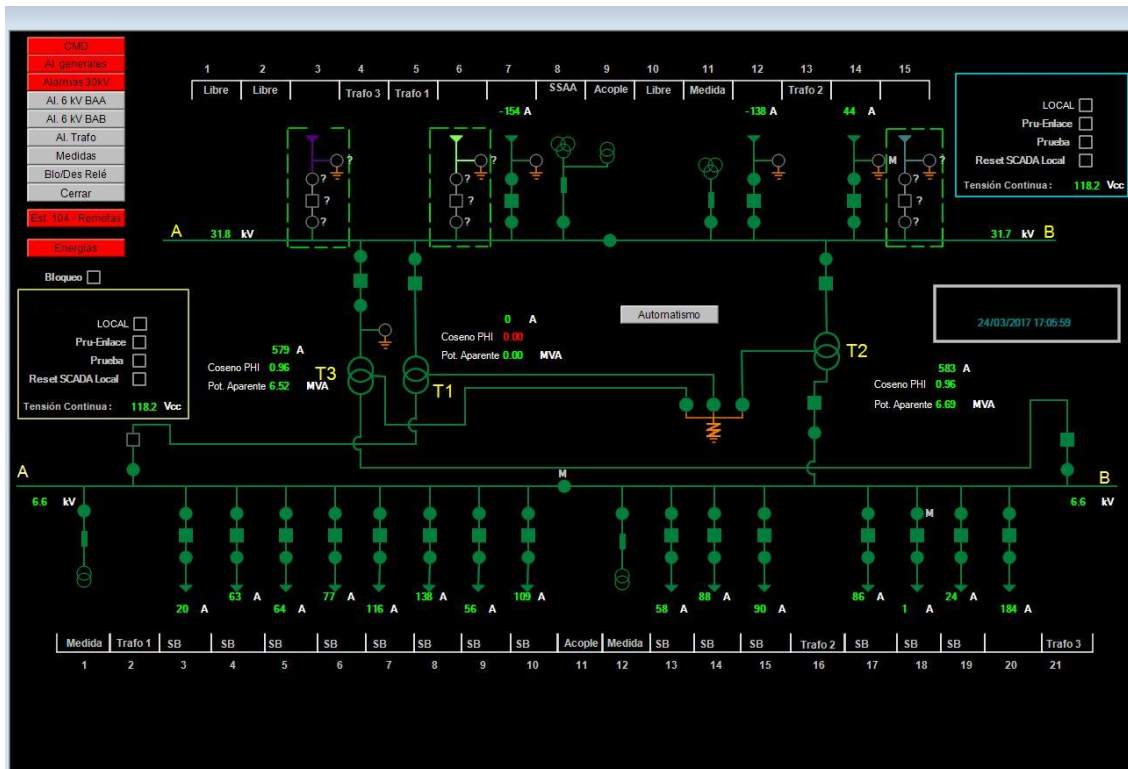


Figura 2.3: Unifilar o Mímico de estación en SCADA Mirage

2.3 Riesgos del proyecto

Antes de efectuar la puesta en producción se deberán minimizar las posibilidades de errores, para eso se diseñó un robusto sistema de pruebas.

Por ejemplo, qué podría pasar si algo NO funciona correctamente cuando el sistema nuevo entre en producción:

- ✚ Podríamos dejar el Aeropuerto Internacional de Carrasco sin energía eléctrica.
- ✚ Podríamos, en el medio de una compleja cirugía, dejar sin luz a un hospital.
- ✚ Podríamos cortar el proceso productivo de una industria, causando miles de dólares en pérdidas.
- ✚ El mal funcionamiento no solo afecta a los clientes de UTE, sino que también puede afectar a la propia red eléctrica en maniobras indebidas sobre ciertos elementos.

Se desarrollará un producto para un sistema muy crítico, como lo es el comando remoto de toda la distribución de energía en Canelones, Florida y Montevideo, más de la mitad de los usuarios de UTE.

La aplicación deberá hacer lo que tiene que hacer, no existen posibilidades de errores; si tenemos que cortar la energía de una estación durante una tormenta, deberá ser esa estación, no podemos terminar apagando el CTI o dejando el aeropuerto sin energía. Si queremos restaurar la energía en un sector que está en mantenimiento, no debemos restaurar otro sector que aún sigue en mantenimiento.

El proceso se desarrollará siempre con el foco en un producto de misión crítica, planificando rigurosos sistemas de pruebas y aplicando prácticas recomendadas para la gestión por el marco de trabajo ITIL v3.

Antes de comenzar con el desarrollo, aclaramos que en los casos de hospitales y aeropuertos existen generadores de respaldo y cuando están realizando mantenimiento, el control para restaurar la energía es manual. De todas formas nuestro cliente no quedaría para nada conforme con un corte provocado erróneamente por un software.

2.4 Estudio de factibilidad

2.4.1 Introducción

En el siguiente estudio se detallan los tipos de recursos con que se cuentan para llevar a cabo la realización del proyecto.

Cabe destacar que algunos detalles económicos no se mencionan por razones de confidencialidad.

2.4.2 Factibilidad técnica

Por un lado se cuenta con el sistema Mirage, que permite modificar cualquiera de sus configuraciones por fuera de los programas Mirage; y por otra parte tenemos el sistema SCADA X, que permite descargar o extraer toda la información cargada en el mismo. Con lo cual es posible traducir los elementos de la configuración del sistema X al sistema Mirage.

2.4.3 Factibilidad económica

El costo de adquisición de los equipos (plataforma Windows), la confección de un nuevo data center y las horas de desarrollo corren por cuenta de UTE. Dichos costos son sensiblemente inferiores al costo de licencias y mantenimiento del SCADA X.

Justamente el ahorro de una importante suma de dinero es uno de los objetivos de este proyecto al prescindir del sistema en cuestión.

2.4.4 Factibilidad operativa

Desde el punto de vista operativo, los sistemas SCADA son similares al momento de operar la red, por lo que los operadores no necesitarán mayor entrenamiento al momento de realizar una maniobra. De todas maneras, UTE tiene planificado realizar cursos a los operadores para comunicarles las nuevas funcionalidades con las que contarán que le dan valor agregado al sistema y facilitan la operación diaria de la red, como lo son, por ejemplo, las gráficas de medidas históricas.

Además, no existe ninguna resistencia al cambio por parte de los operadores, que son los usuarios netos del sistema SCADA.

Conjuntamente se cuenta con desarrolladores con experiencia en la plataforma .Net, que es la solicitada para desarrollar dicha migración, y que son, a su vez, administradores de los sistemas SCADA, proporcionando así un mejor entendimiento de la problemática del proyecto.

2.4.5 Factibilidad legal

UTE posee las licencias necesarias de todo el software implicado en la migración.

Por otra parte, la empresa sería propietaria de todo el desarrollo del software de migración, tanto a nivel de base de datos como gráfica.

Por cuestiones de explotación de la reputación ajena [5] no se menciona la marca del SCADA de origen y se lo denomina "X". Esto se debe básicamente a una eventual publicación de este trabajo.

Además, por razones de confidencialidad, se omite dar detalles económicos o de otra naturaleza que sean de interés de UTE y que no dan aporte académico al trabajo.

2.4.6 Conclusión

Se ve factible la realización del proyecto, ya que los recursos operativos, técnicos, económicos y legales invertidos por UTE justifican ampliamente los intereses perseguidos. Sumando además experiencia y *know how* para la empresa en temas de migración.

3 Ingeniería de Requerimientos

3.1 Introducción

El proceso de Ingeniería de Requerimientos se compone de los procesos de relevamiento, especificación y validación, que se detallarán a continuación.

En ocasiones el hecho de pensar en una migración es sinónimo de "mover" funcionalidades de un lado para otro o de que las operaciones que se ejecutan en una plataforma se codifiquen en otra plataforma manteniendo el mismo resultado.

No obstante, en este caso existen dos sistemas con filosofías distintas, si bien los dos SCADA operan la red, lo que puede ser una funcionalidad en uno puede que no sea necesaria en otro, o lo que son dos funcionalidades en uno sea una sola en otro. En consecuencia, se tomaron un conjunto de funcionalidades del SCADA X (véase Anexo 2 – Funcionalidades SCADA X) y en base a esas funcionalidades y a algunos elementos de control de la migración se fueron construyendo los requerimientos funcionales del proyecto.

3.2 Técnicas y resultados de especificación

En esta sección se describen las técnicas de relevamientos y las consideraciones a la hora de definir tanto los requerimientos funcionales como los no funcionales.

A continuación se presentan en orden cronológico cómo fue el proceso realizado para obtener los requerimientos:

-  Entrevista realizada con Lidia Dávila y técnicos de UTE, con el objetivo de definir lineamientos de la migración.
-  Reunión con técnicos de UTE para definir los requerimientos desprendidos de los lineamientos generales definidos previamente.
-  Reunión con Lidia Dávila para aprobar los requerimientos establecidos.
-  Una vez definidos los requerimientos, se necesitaron reuniones varias con técnicos de UTE, por lo general con un formato de tormenta de ideas para resolver complejidades técnicas que fueron surgiendo en las implementaciones.

3.3 Entrevistas

Entrevista 03/10/16: En esta primera entrevista, realizada con la Lidia Dávila y técnicos de UTE, se definieron los siguientes lineamientos:

-  Realizar la migración por elementos, considerando que algunos elementos se agrupan para formar uno solo en Mirage. Ir probando elemento a elemento durante la migración.

- ✚ En elementos que no se pueda determinar de manera automática a que elemento corresponde en Mirage, se deberá mostrar listado acotado de posibilidades para que se el usuario escoja el correcto.
- ✚ Mantener todas las funcionalidades actuales, incluyendo conexiones entre elementos de la red, alarmas y protocolos.
- ✚ Mantener integración de instalaciones con sistema SIO de UTE. (Anexo 5 - SIO: Sistema Integrado de Operación).
- ✚ Realizar un módulo de software que lea los archivos convertidos y encuentre diferentes incompatibilidades.
- ✚ Los diagramas serán migrados más de una vez cada uno, con cada corrección que se realice en el software, por lo que se deberá guardar las elecciones del usuario para que en futuras corridas, se realice de forma totalmente automática la conversión.

Entrevistas 10/10/16 y 17/10/16: En estas dos entrevistas se definieron los requerimientos obtenidos a partir de los lineamientos generales.

A continuación vemos un diagrama de cómo se procesó cada uno de los lineamientos.

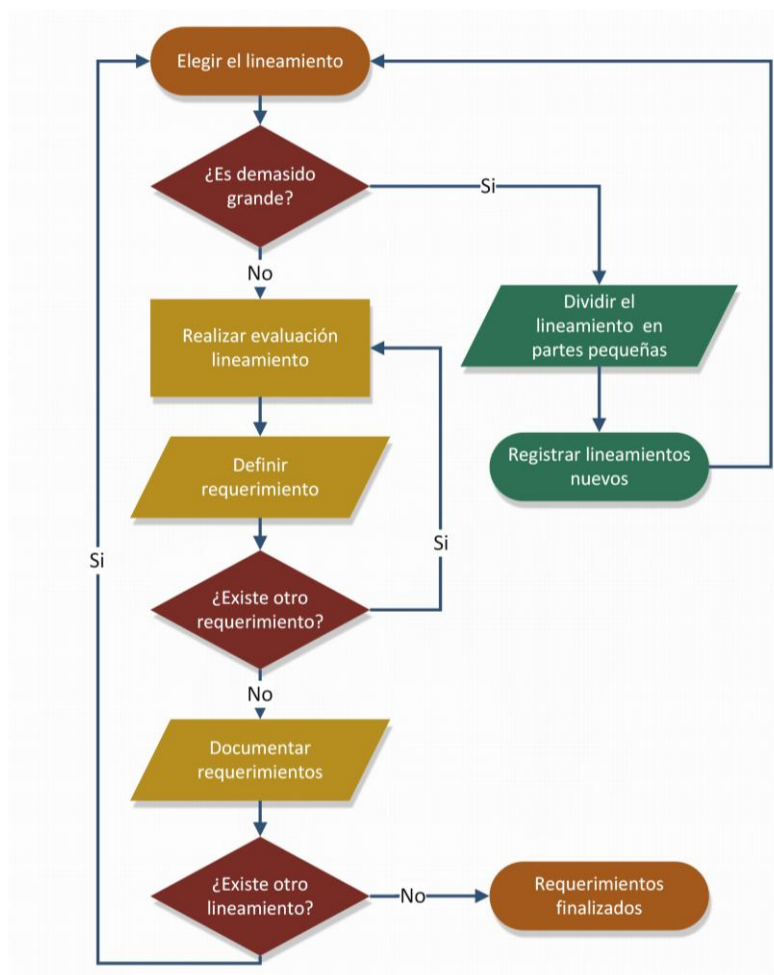


Figura 3.1: Método de procesar los lineamientos definidos.

3.4 Observación

La pieza de software automatiza la migración entre los sistemas SCADA

, evitando el diseño manual de todos los diagramas y el posible error humano que eso conlleva.

Con el objetivo de deducir las necesidades del cliente, así como también de entender mejor el proceso y obtener información de primera mano, se observó cómo se realiza la edición manual de un diagrama en el editor de mímicos Mirage.

Dicha observación no solo fue útil para inferir necesidades del cliente, sino que además se pudo observar los hábitos de trabajo del usuario, cómo se llevaban a cabo los procesos y qué orden natural seguían durante la edición de un mímico.

3.5 Estudio de documentación

Se estudiaron las siguientes documentaciones de ambos sistemas y del cliente:

Sistema SCADA X:

-  Manuales SCADA X.
-  Análisis de archivo propietario y base de datos Oracle, el modelado de los diferentes elementos, así como también sus estados son desplegados en ambos sistemas.

Sistema SCADA Mirage:

-  Manuales Mirage.
-  Análisis de archivo XML, donde se guardan las configuraciones de los diferentes elementos que representan el mímico. Se aplicó ingeniería inversa, se diseñaron diferentes elementos en el editor de Mirage y se estudió la representación de cada uno.

Cliente:

-  Manual realizado por UTE, explicando el funcionamiento del Sistema de Telecontrol Mirage.

3.6 Justificación de la estrategia de relevamiento

Para elegir las estrategias de relevamientos necesarias, previamente se tuvieron las siguientes consideraciones:

-  Aprovechamos todo el conocimiento que teníamos en el dominio.
-  Se estudiaron los manuales de ambos SCADA.

- ✚ Observamos el sistema en acción, analizando todas las funcionalidades que actualmente está cumpliendo el sistema.
- ✚ Analizamos las estructuras de los diferentes archivos, por ejemplo del archivo propietario y la base de datos que usa SCADA X, así como también el archivo XML que utiliza el SCADA Mirage.

Luego de estas actividades se hicieron bosquejos de cómo sería la migración y se definió el orden para realizar las diferentes entrevistas y cómo se efectuarían las mismas.

Decidimos, en una primera instancia, validar nuestro análisis y obtener una serie de lineamientos con la Lidia Dávila. Para luego definir en detalles cada uno de los requerimientos con técnicos de UTE.

Resumiendo, nuestra estrategia se basó en aprovechar en el conocimiento previo del sistema y en la observación y estudio, optimizando el tiempo insumido las diferentes entrevistas.

3.7 Requerimientos definidos

En esta sección se presenta el listado de los requerimientos definidos:

Requerimientos Funcionales
Estaciones, subestaciones y puesto de conexión.
RF1. Agrupar elementos de corte.
RF2. Identificar configuraciones de celda de corte.
RF3. Persistir selección de usuario.
RF4. Elementos dinámicos distintos a elementos de corte.
RF5. Conectividad en reenganches
Puntos intermedios
RF6. Elementos dinámicos
RF7. Color de elementos comandables
Unifilares en general.
RF8. Mantener la conectividad de las instalaciones.
RF9. Mantener integración de instalaciones con sistema SIO de UTE.
RF10. Inyecciones de conectividad
RF11. Formato textos estáticos y textos en celdarios (número y destino de celdas)
RF12. Textos estáticos de unidades de medidas
RF13. Migración nombre interno instalación
RF14. Comandos Habilitar/Deshabilitar relé
RF15. Selección por área
RF16. Selección por instalación
RF17. Variables en bandera DPF
RF18. Nombre de variables en proceso de conversión
Mímicos Comunicaciones
RF19. No migrar panel comunicaciones

Módulo de pruebas automatizadas
RF20. Instalaciones cruzadas
RF21. Celdas bien referenciadas
RF22. Celdas solapadas gráficamente
RF23. Botón de comunicaciones 101
RF24. Mímicos invocados
RF25. Conectividad de instalaciones
RF26. Mímicos con instalaciones migradas automáticamente
Requerimientos no funcionales
RNF1. Sistema operativo Windows 7
RNF2. Acceso a bases de datos
RNF3. Lenguaje de programación
RNF4. IDE de programación
RNF5. Pruebas automatizadas
RNF6. <i>Testing</i> unitarios
RNF7. Auditoría del sistema (log)
RNF8. Utilización del <i>Framework</i> ITIL V3.

Tabla 3-1: Listado de Requerimientos

3.8 Requerimientos funcionales

A continuación se enumeran y agrupan los requerimientos funcionales de acuerdo a los distintos tipos de representaciones gráficas de estaciones, subestaciones y puestos de conexión que podemos encontrar en el telecontrol de instalaciones eléctricas y que requieren ser migrados. A las representaciones gráficas que se verán en las pantallas de los ordenadores se las conoce como Mímicos.

3.8.1 Estaciones, subestaciones y puesto de conexión.

Las estaciones y subestaciones eléctricas son aquellas instalaciones compuestas por los adecuados elementos de mando, corte, medida, regulación, transformación y protección, y cuya misión es la de reducir valores de tensión. Mientras que los puestos de conexión y medida son equipamientos eléctricos necesarios para atender y registrar la energía y potencia.

Mientras que los puestos de conexión y medida son equipamientos eléctricos necesarios para atender y registrar la energía y potencia.

RF1. Agrupar elementos de corte.

Descripción: Agrupar los elementos de corte y PAT (en caso de tener) de una celda en un solo objeto compuesto de Mirage que representa la misma configuración de celda.

Complejidad: Alto

Referencias cruzadas: RF2, RF8, RF10, RF18

Especificación: CU1, CU7, CU9, CU17

Prioridad: 1

Imágenes descriptivas:

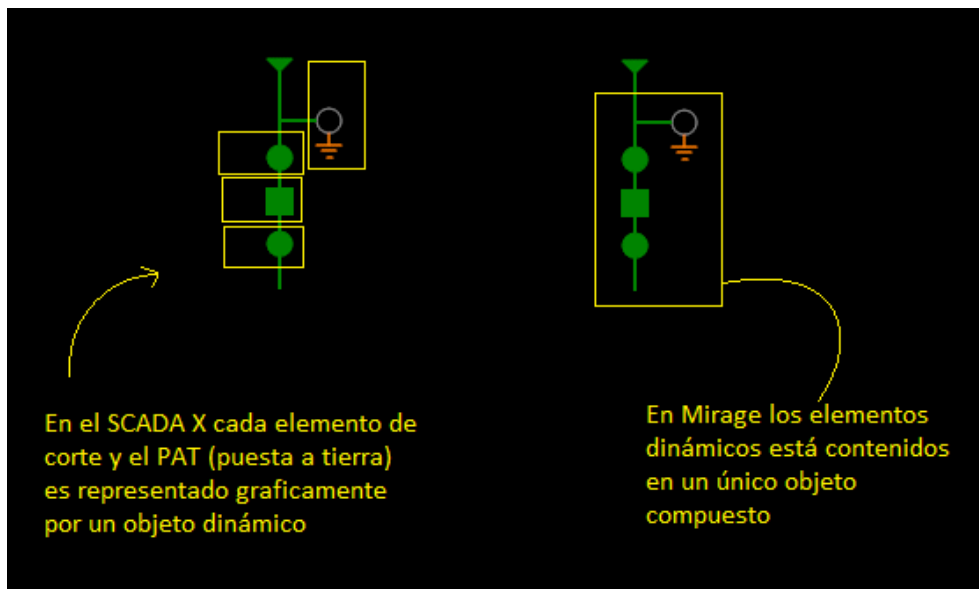


Figura 3.2: Esquema de composición de objetos entre SCADA

RF2. Identificar configuraciones de celda de corte.

Descripción: El sistema debe identificar una determinada configuración de celda y disponer al usuario un listado de posibles objetos compuestos de Mirage.

Complejidad: Alto

Referencias cruzadas: RF1, RF3, RF8, RF10, RF18

Especificación: CU1, CU2, CU7, CU9, CU17

Prioridad: 1

Imágenes descriptivas:

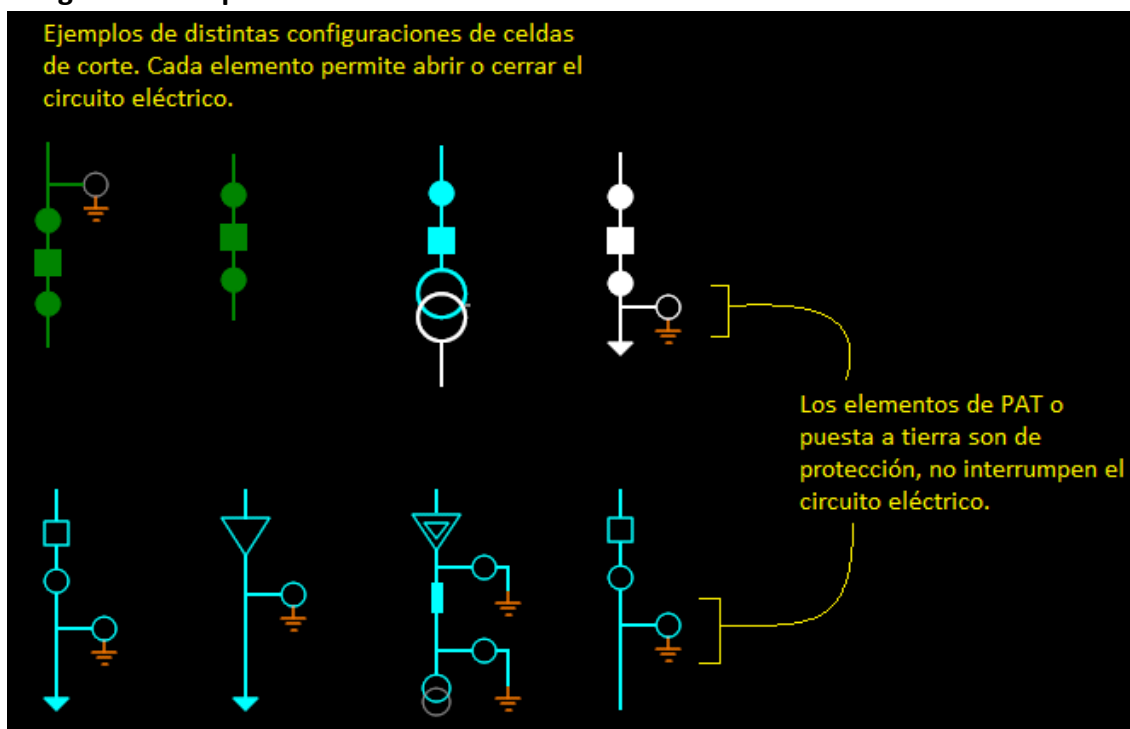


Figura 3.3: Algunos ejemplos de configuraciones de celdas de corte

RF3. Persistir selección de usuario.

Descripción: El sistema debe guardar las selecciones de objetos realizadas por el usuario para cada mímico convertido, para evitar re seleccionar objetos en ejecuciones posteriores.

Complejidad: Medio

Referencias cruzadas: RF2

Especificación: CU2

Prioridad: 2

RF4. Elementos dinámicos distintos a elementos de corte.

Descripción: Elementos dinámicos que no pertenecen a elementos de corte de una celda (excepto PAT), deben traducirse a un objeto correspondiente de Mirage sin consultar al usuario.

Complejidad: Medio

Referencias cruzadas: RF6, RF12, RF13

Especificación: CU3, CU5

Prioridad: 2

Imágenes descriptivas:

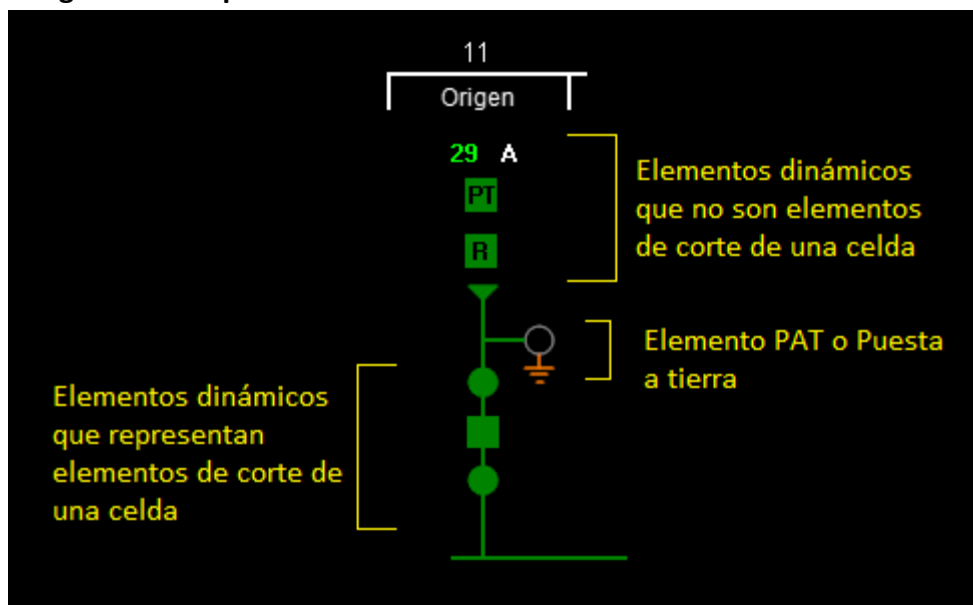


Figura 3.4: Esquema de elementos dinámicos de una celda

RF5. Conectividad en reenganches

Descripción: Los elementos reenganches deben conectarse a la barra más próxima mediante un conector transparente.

Complejidad: Baja

Especificación: CU4

Prioridad: 3

Imágenes descriptivas:

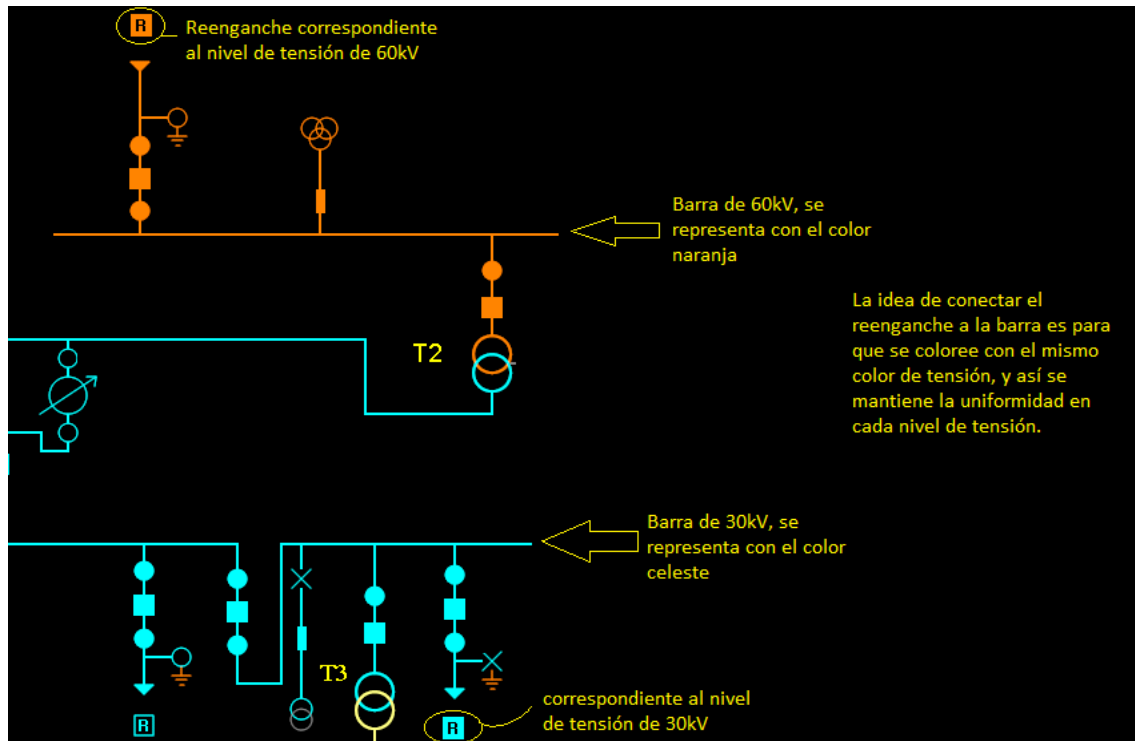


Figura 3.5: Reenganches y sus respectivas barras

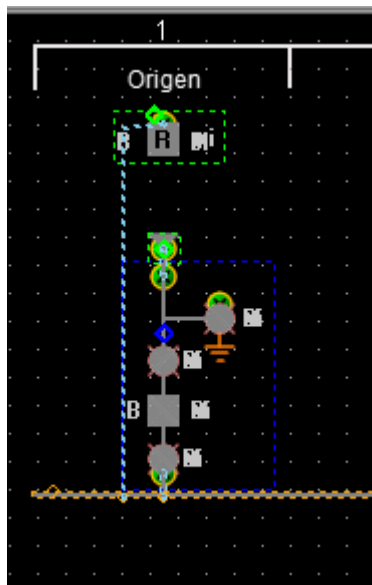


Figura 3.6: Conector reenganche a barra.

En la Figura 3.6 se observa la vista en modo edición del conector transparente del reenganche a la barra.

3.8.2 Puntos intermedios

Los puntos intermedios sirven para monitorear las líneas rurales y detectar faltas de energía, son reconectores que, en caso de que haya una falta de energía (por ejemplo, provocado por una rama caída en la línea), se abren y vuelven a cerrarse. En caso de que no quede cerrado, línea sin tensión, se envía una cuadrilla a la zona.

En las figura 3.7 y figura 3.8 vemos ejemplos genéricos de cómo son representados en SCADA los seccionalizadores y reconectores, respectivamente.

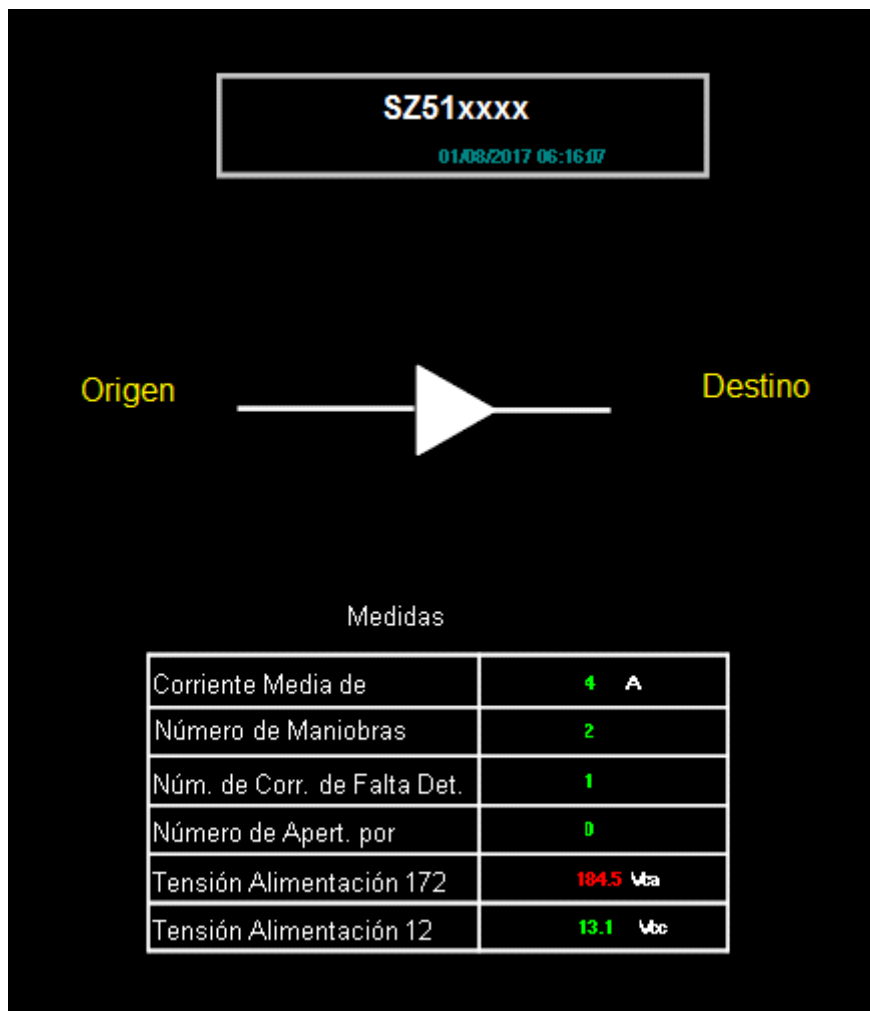


Figura 3.7: Mímico donde se representa un seccionalizdor de línea.

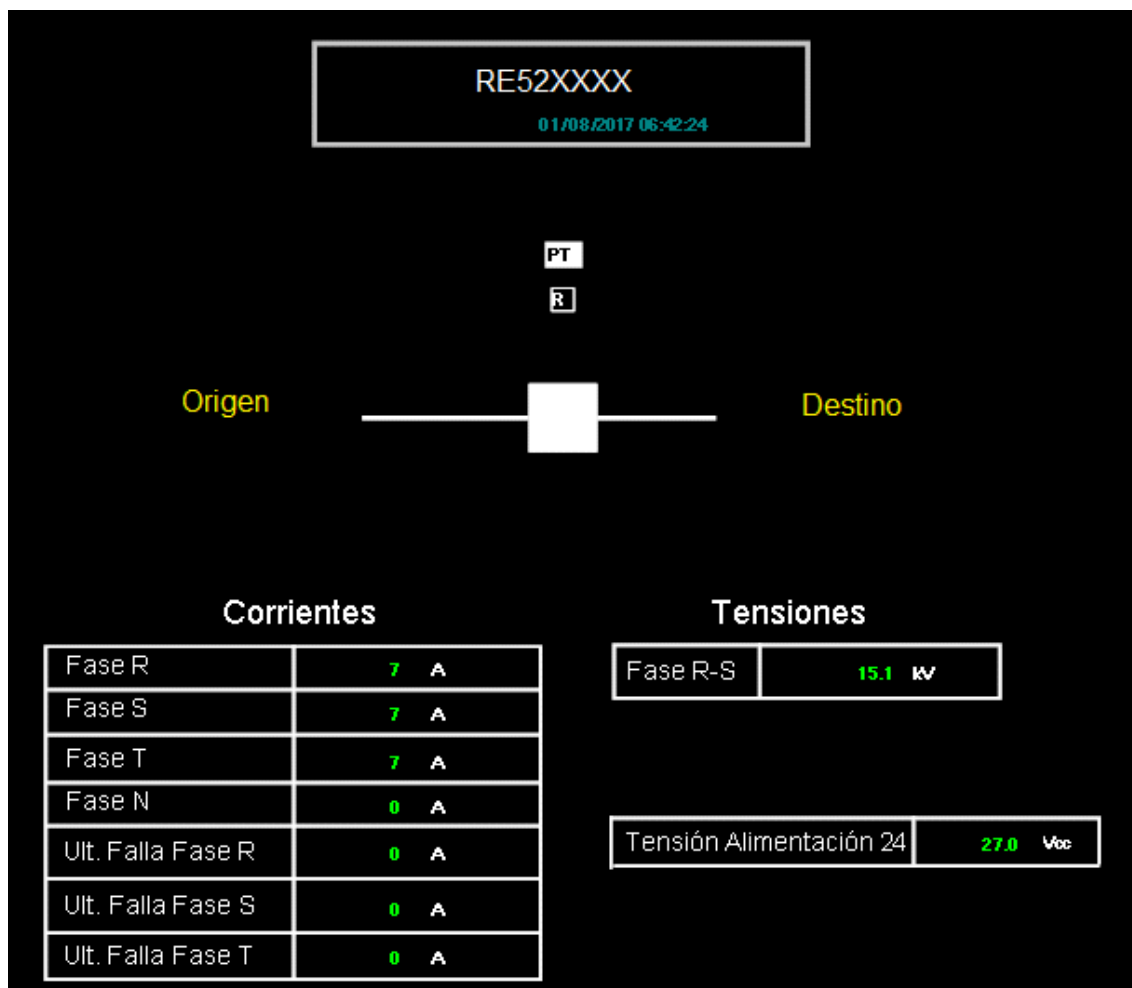


Figura 3.8: Mímico donde se representa un reconector de línea

RF6. Elementos dinámicos

Descripción: Los elementos dinámicos deben traducirse de forma directa a un objeto Mirage, el usuario no debe seleccionar nada. Un objeto Mirage es una entidad reutilizable que maneja el SCADA a nivel gráfico tanto para representar los elementos de la red eléctrica.

Complejidad: Medio

Referencias cruzadas: RF4, RF12, RF13

Especificación: CU3, CU5

Prioridad: 1

Imágenes descriptivas:

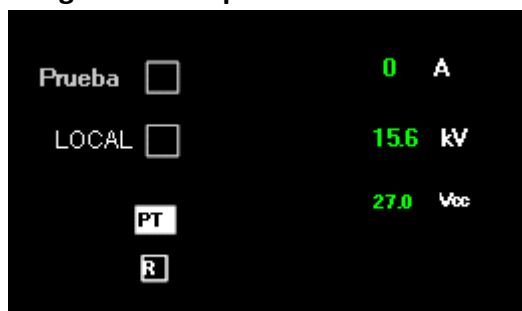


Figura 3.9: Algunos ejemplos de elementos dinámicos

RF7. Color de elementos comandables

Descripción: Elementos comandables que pertenecen al mismo nivel de tensión que el elemento de corte principal deben mantener el mismo color.

Complejidad: Medio

Especificación: CU6

Prioridad: 1

Imágenes descriptivas:

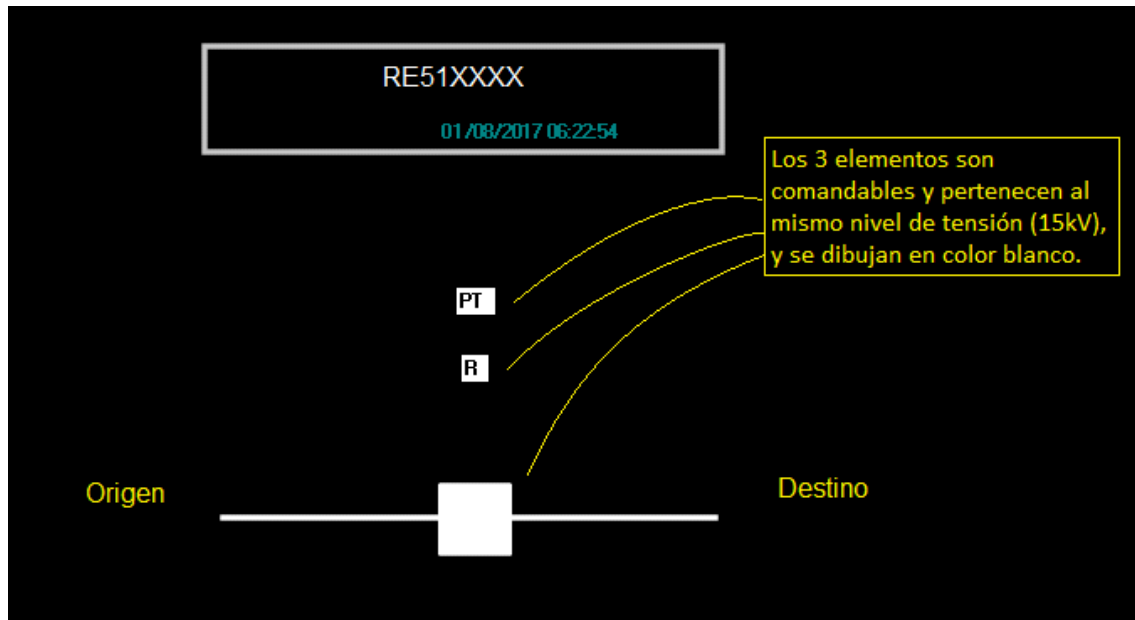


Figura 3.10: Esquema de elementos comandables.

3.8.3 Unifilares en general

Un diagrama unifilar es una representación gráfica de una instalación eléctrica o de parte de ella. El esquema unifilar se distingue de otros tipos de esquemas eléctricos en que el conjunto de conductores de un circuito se representa mediante una única línea, independientemente de la cantidad de dichos conductores. Típicamente el esquema unifilar tiene una estructura de árbol, aunque no es la única. [7]

RF8. Mantener la conectividad de las instalaciones.

Descripción: Mantener la misma configuración de conexiones entre los elementos que estén conectados en la instalación.

Complejidad: Alto

Referencias cruzadas: RF1, RF2, RF10, RF18

Especificación: CU1, CU7, CU9, CU17

Prioridad: 2

RF9. Mantener integración de instalaciones con sistema SIO de UTE.

Descripción: Todas las instalaciones migradas deberán mantener referenciado su id de integración con SIO en sus correspondientes mímicos.

Complejidad: Bajo

Especificación: CU8

Prioridad: 2

Imágenes descriptivas:

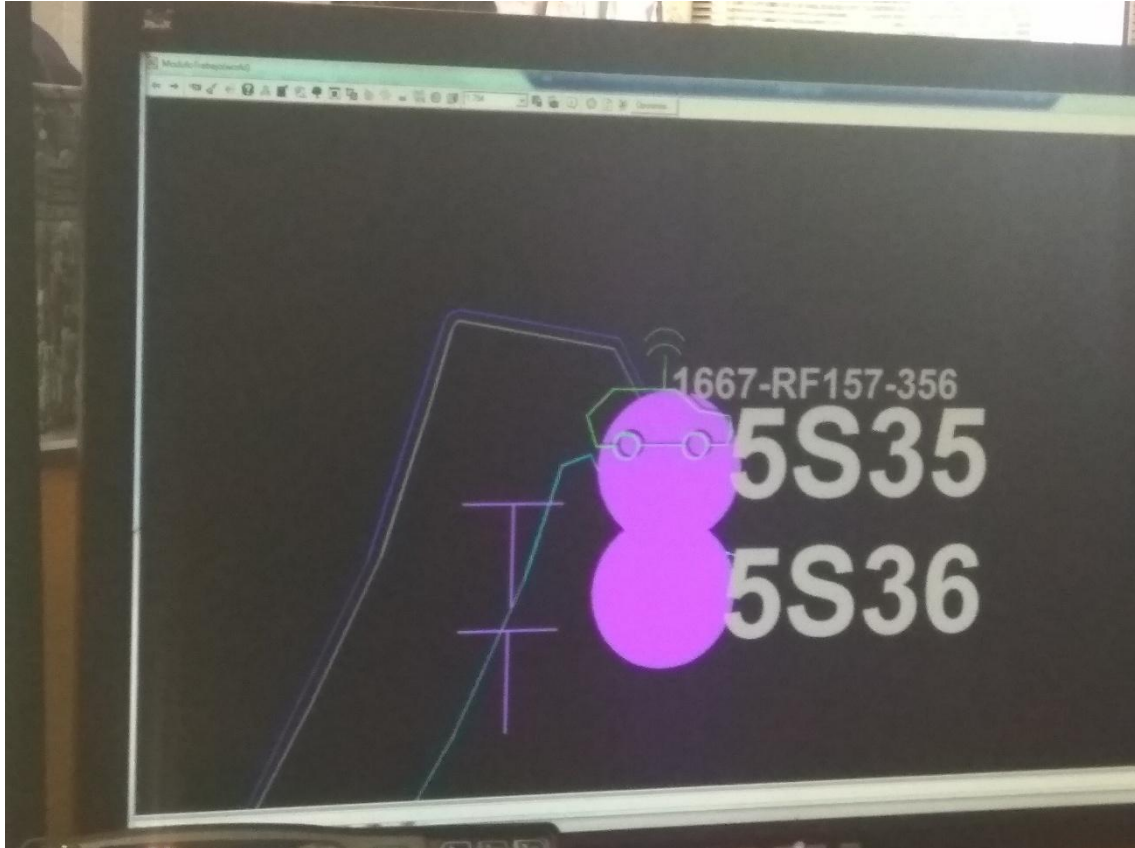


Figura 3.11: Pantallas del SIO.

La figura 3.11 muestra una de las pantallas donde se visualizan dos instalaciones, la topología eléctrica, y la presencia de un móvil de brigada. Desde SCADA se pueden localizar instalaciones en las pantallas de SIO, siempre y cuando la misma tenga su mímico integrado con SIO.

RF10. Inyecciones de conectividad

Descripción: Las inyecciones que pintan conectividad deberán tener distintos colores entre sí en el unifilar de Mirage.

Complejidad: Medio

Referencias cruzadas: RF1, RF2, RF8, RF18

Especificación: CU1, CU7, CU9, CU17

Prioridad: 2

Imágenes descriptivas:

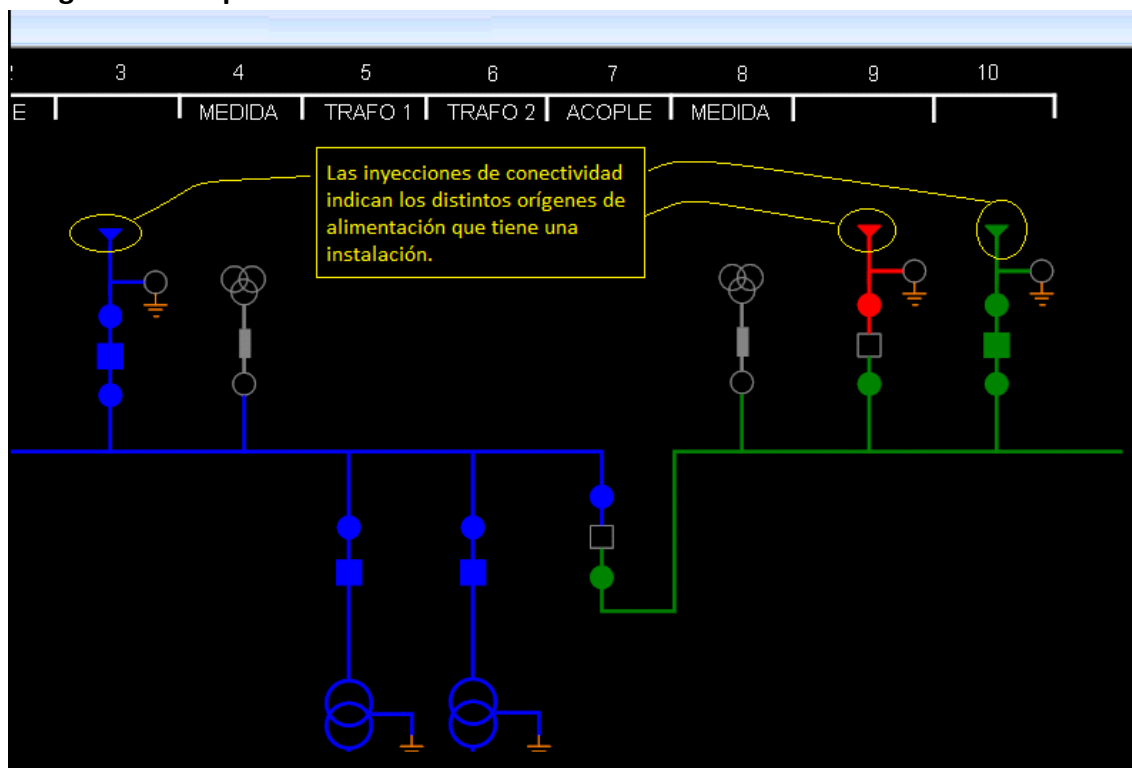


Figura 3.12: Instalación alimentada desde varios orígenes.

RF11. Formato textos estáticos y textos en celdarios (número y destino de celdas)

Descripción: Los textos estáticos deben mantener sus proporciones relativas al unifilar, mientras que los textos en los celdarios deben tener tipo de letra “Arial” de tamaño 10 y color blanco.

Complejidad: Alto

Especificación: CU10

Prioridad: 2

Imágenes descriptivas:

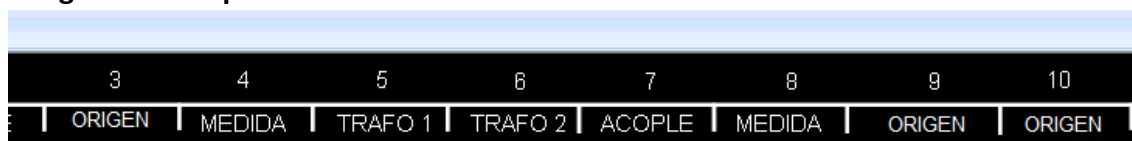


Figura 3.13: Ejemplo celdario.

En la figura 3.13 se puede ver el celdario identificando la celda 3 hasta la celda 10 con el tipo de texto requerido.

RF12. Textos estáticos de unidades de medidas

Descripción: No migrar textos estáticos correspondientes unidades de medidas, ya que los objetos Mirage ya tienen dichos textos embebidos.

Complejidad: Bajo

Especificación: CU3, CU11

Prioridad: 2

Imágenes descriptivas:



Figura 3.14: Unidades (kV)

En la figura 3.14 se muestra un ejemplo de unidades kV (kilovoltios), un tipo de unidad que el sistema debe obviar.

RF13. Migración nombre interno instalación.

Descripción: Traducir el nombre interno de la instalación al objeto "Cuadro de instalación" perteneciente a Mirage.

Complejidad: Medio

Referencias cruzadas: RF4, RF6

Especificación: CU12

Prioridad: 2

Imágenes descriptivas:



Figura 3.15: Cuadro de instalación.

En la Figura 3 15 tenemos la vista del objeto "Cuadro de instalación" configurado para la estación 5XXX Barrio.

RF14. Comandos habilitar/deshabilitar relé

Descripción: Los comandos habilitar/deshabilitar relé no se deberían dibujar en la conversión del unifilar, se debe agregar solamente el botón que hacer referencia a los scripts automáticos de Mirage.

Complejidad: Medio

Especificación: CU13

Prioridad: 1

Imágenes descriptivas:



Figura 3.16: Botón que abre la vista de Comandos hab/deshab relé.



Figura 3.17: Vista del Comando Hab/deshab relé en SCADA Mirage

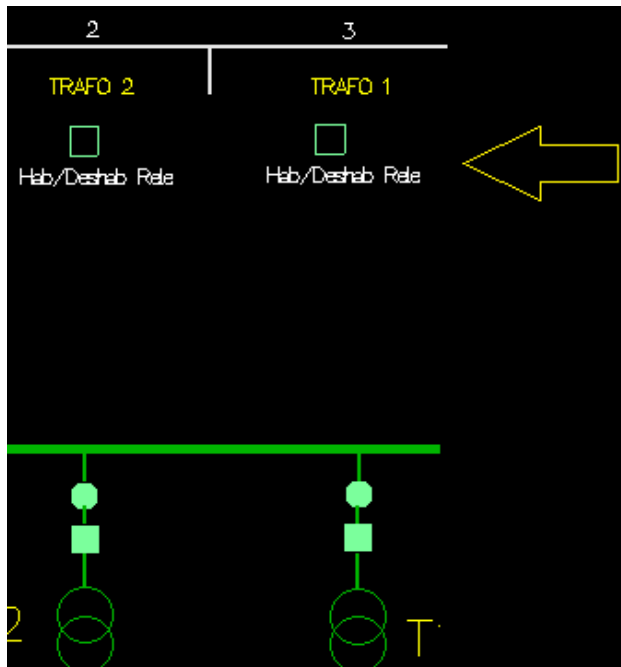


Figura 3.18: Comando Hab/deshab relé

En figura 3.18 se puede observar que SCADA X, el comando Hab/deshab relé se representa sobre el mismo mímico.

RF15. Selección por área

Descripción: El usuario deberá poder seleccionar fácilmente las estaciones a migrar de área 1 (prueba), área 2 (Canelones y Florida) y área 3 (Montevideo).

Complejidad: Baja

Referencias cruzadas: RF16

Especificación: CU14, CU15

Prioridad: 3

RF16. Selección por instalación

Descripción: El usuario debe poder seleccionar una o varias instalaciones a migrar.

Complejidad: Baja

Referencias cruzadas: RF15

Especificación: CU14, CU15

Prioridad: 2

RF17. Variables en bandera DPF

Descripción: El objeto que representa la alarma DPF (detector de paso de falta), cuya representación gráfica es una bandera, deberá contener la misma cantidad de variables de origen, en caso de que sea un "or" de más de una variable, o deberá contener la misma variable calculada correspondiente de la base Mirage (VDM: Variable digital de memoria, denominación que Mirage le da a las variables digitales calculadas).

Complejidad: Baja

Especificación: CU16

Prioridad: 2

Imágenes descriptivas:



Figura 3.19: DPF (Detector de paso de falta)

RF18. Nombre de variables en proceso de conversión

Descripción: El sistema deberá mostrar al usuario el nombre de la variable en el proceso de conversión.

Complejidad: Baja

Referencias cruzadas: RF1, RF2, RF8, RF10

Especificación: CU1, CU7, CU9, CU17

Prioridad: 1

Imágenes descriptivas:

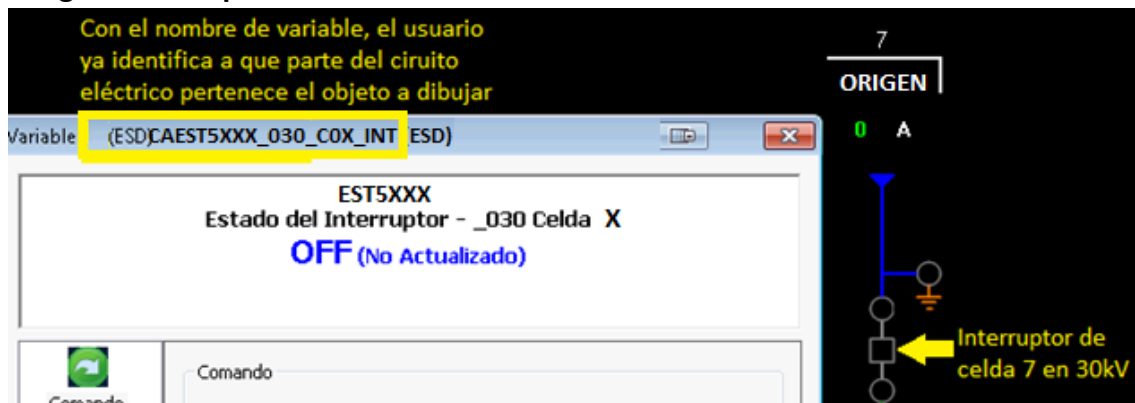


Figura 3.20: Nombre variable que representa en un mímico.

3.8.4 Mímicos Comunicaciones

Es en las pantallas del SCADA donde se muestran el estado de las comunicaciones de las instalaciones. Muestran si el enlace de comunicación está levantado (si se está comunicado), la cantidad de paquetes enviados, recibidos, con error y, además, se puede hacer un comando de "demanda general" (forzar a recibir todos los puntos).

RF19. No migrar panel comunicaciones

Descripción: No migrar el panel de comunicaciones, solo se debe agregar un botón de script automático de Mirage parametrizado con la instalación, de tal forma que abra el mímico de comunicaciones correspondiente a la instalación.

Complejidad: Media

Especificación: CU18

Prioridad: 1

Imágenes descriptivas:

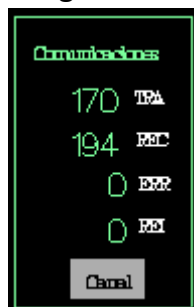


Figura 3.22: Panel de comunicaciones

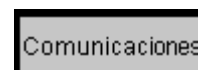


Figura 3.21: Botón de comunicaciones en un mímico Mirage

3.8.5 Módulo de pruebas automatizadas

A continuación se enumeran los requerimientos que debe cumplir el módulo que ejecuta las pruebas automatizadas. Dichas pruebas verifican aspectos gráficos importantes que inciden directamente en la operación de la red. La idea es detectar errores sustanciales a nivel gráfico sobre los mímicos, en vista de que dichos errores podrían ser originados por el software de migración u originados por el usuario que realizó modificaciones manuales sobre los mímicos.

Por ejemplo, si el usuario modificó un elemento comandable sobre un mímico y dicho elemento hace referencia a otra instalación, se corre el riesgo de operar una instalación erróneamente.

RF20. Instalaciones cruzadas

Descripción: Se debe verificar que las variables en un mímico pertenezcan a una misma instalación y, en caso que haya más de una instalación por mímico, informar o tener en cuenta si es un caso particular que ya tiene contemplado el usuario.

Complejidad: Alta

Especificación: CU19

Prioridad: 1

RF21. Celdas bien referenciadas

Descripción: Se debe verificar que las variables pertenecientes a una misma celda correspondan a la numeración en el celdario.

Complejidad: Media

Referencias cruzadas: RF22

Especificación: CU20

Prioridad: 1

Imágenes descriptivas:

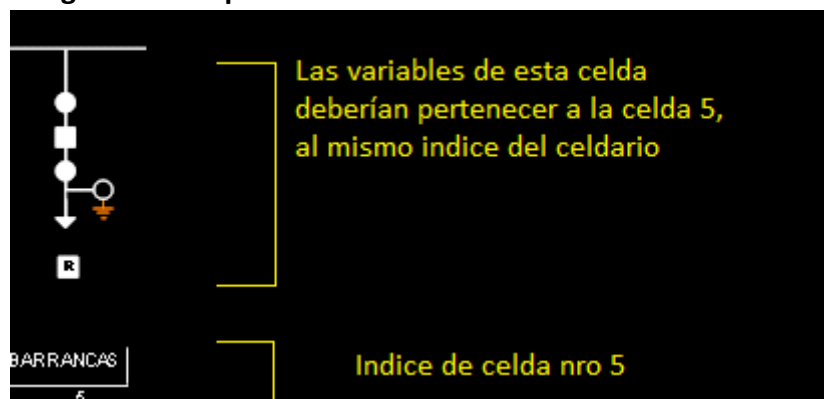


Figura 3.23: Celda con su respectiva referencia en el celdario

RF22. Celdas solapadas gráficamente

Descripción: Se debe verificar que las celdas de corte de corte no se solapen dentro de un mímico, es decir, que no haya elementos de una celda posicionados gráficamente en ubicaciones correspondientes a otra celda de corte.

Complejidad: Media

Referencias cruzadas: RF21

Especificación: CU24, CU20

Prioridad: 1

Imágenes descriptivas:

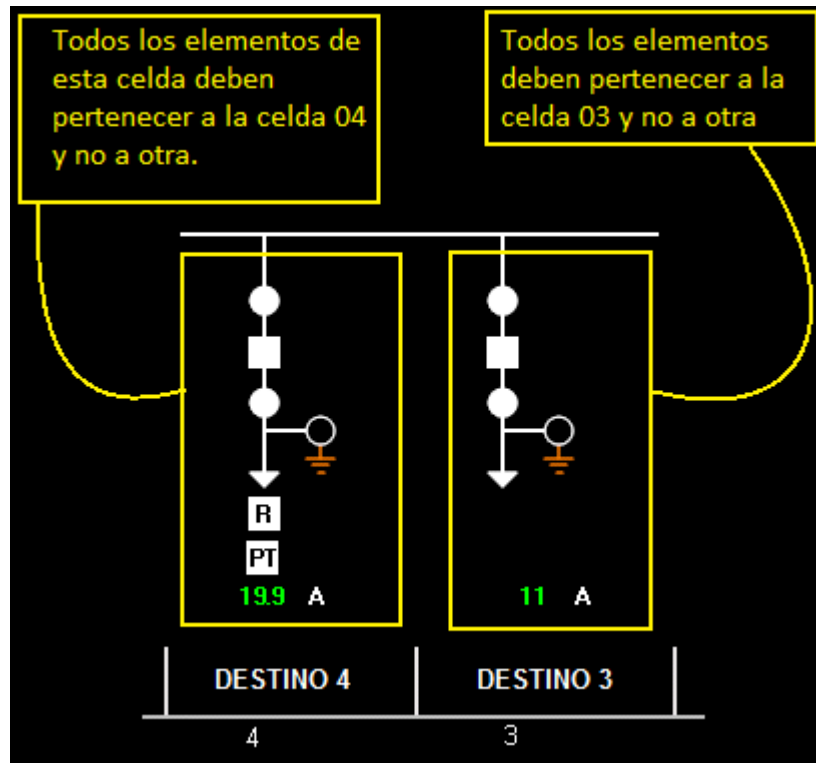


Figura 3.24- Ejemplo de elementos pertenecientes a sus respectivas celdas.

RF23. Botón de comunicaciones 101

Descripción: Se debe verificar que las instalaciones cuyo protocolo de comunicación sea 101, tengan bien asociado su sistema parcial al botón de comunicaciones en su mímico.

Complejidad: Alta

Especificación: CU21

Prioridad: 1

Imágenes descriptivas:

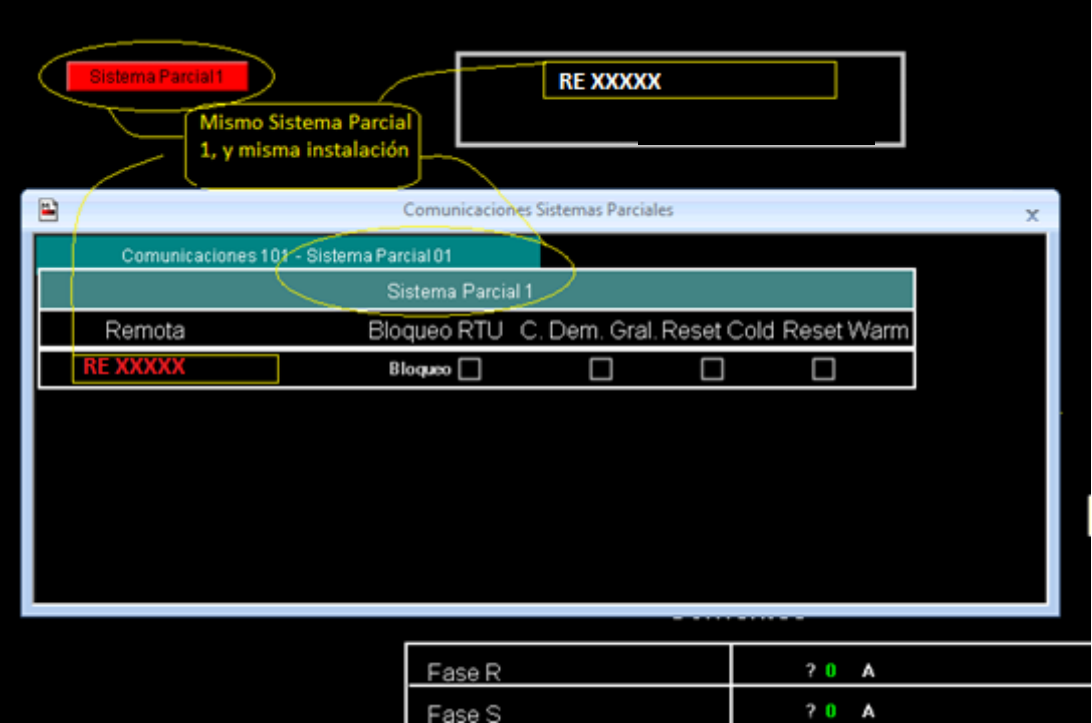


Figura 3.25: Mímico de instalación y de comunicaciones.

La información en el mímico de instalación y en el mímico de comunicaciones debe ser consistente como se puede ver en la Figura 3.25.

RF24. Mímicos invocados

Descripción: Se debe verificar que los mímicos convertidos por medio del software de migración sean invocados desde algún botón.

Complejidad: Baja

Especificación: CU22

Prioridad: 1

Imágenes descriptivas:

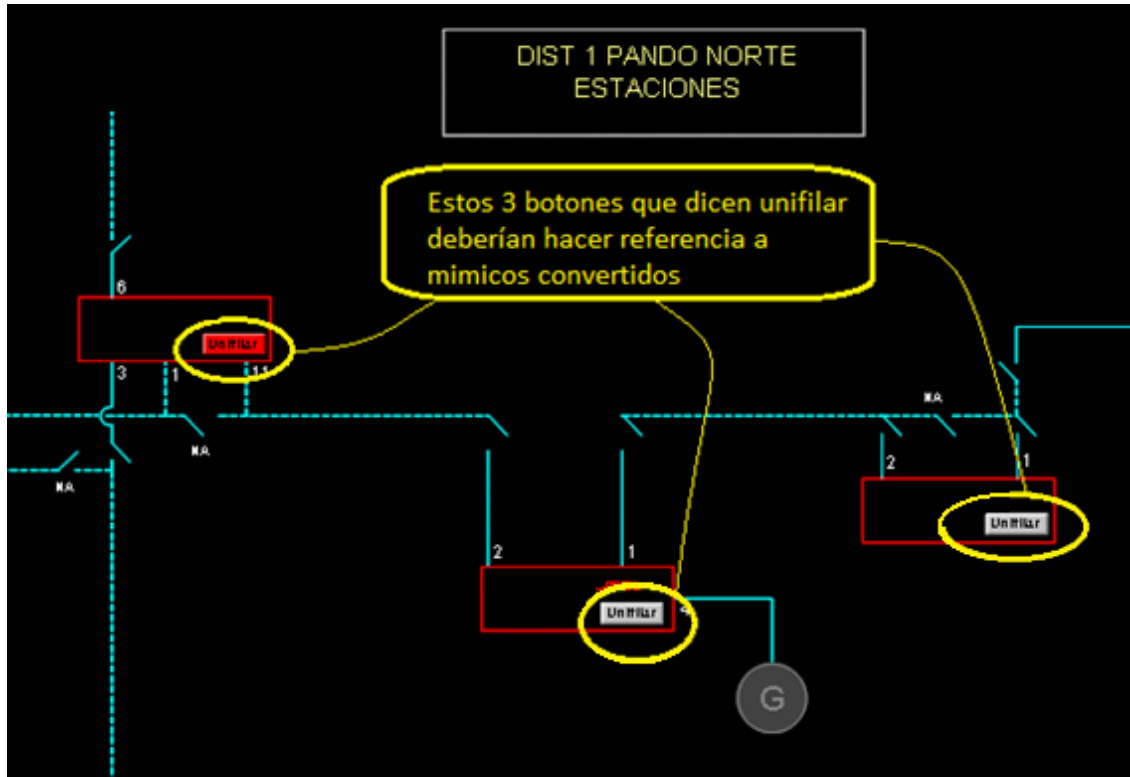


Figura 3.26: Mímico de distrito

Figura 3.26 se poder ver el mímico de distrito confeccionado manualmente, el mismo permite invocar unifilares de distintas instalaciones.

RF25. Conectividad de instalaciones

Descripción: Se debe verificar que las instalaciones migradas mantengan la misma conectividad entre sus elementos de corte que en el SCADA X.

Complejidad: Alta

Especificación: CU23

Prioridad: 3

Imágenes descriptivas:

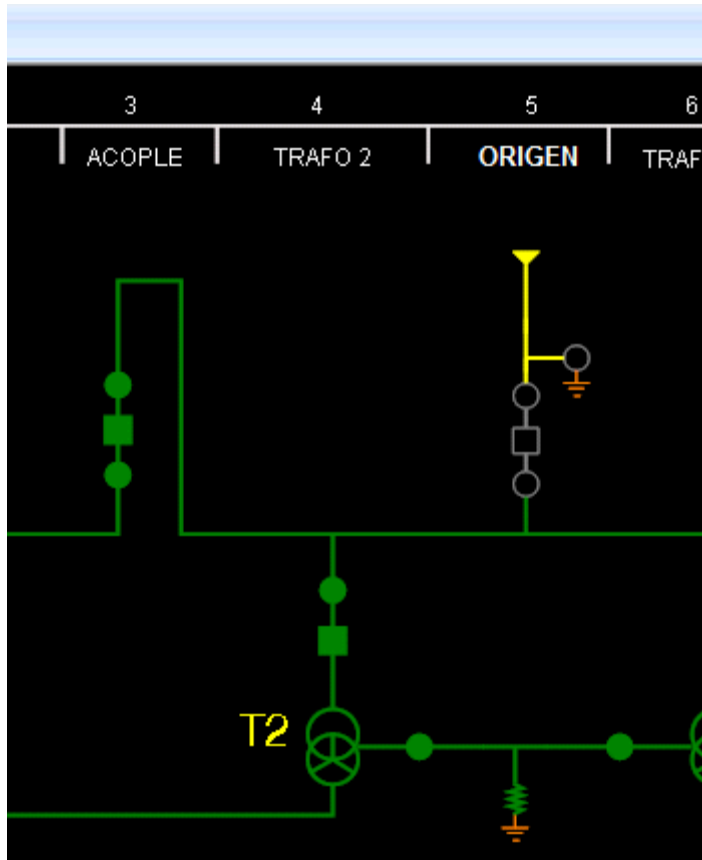


Figura 3.27: Conectividad

Se puede observar en la figura 3.27 que La conectividad permite visualizar por medio de colores la presencia o ausencia de energía eléctrica en un circuito.

RF26. Mímicos con instalaciones migradas automáticamente.

Descripción: Se debe verificar que los mímicos contengan instalaciones migradas automáticamente por el sistema de conversión gráfica. (Verifica integración con SIO)

Complejidad: Baja

Especificación: CU25

Prioridad: 1

3.9 Requerimientos no funcionales

RNF1. Sistema operativo Windows 7

Descripción: El sistema deberá correr bajo la plataforma Windows 7 (SP1).

Referencias cruzadas: Todos los RF.

Prioridad: 1

RNF2. Acceso a bases de datos

Descripción: El sistema deberá poder acceder tanto a bases Oracle 8 como a SQL Server 2008.

Referencias cruzadas: RF1, RF2, RF3, RF4, RF5, RF6, RF7, RF8, RF9, RF10, RF13, RF14, RF15, RF16, RF17, RF18, RF20, RF23, RF25, RF26.

Prioridad: 1

RNF3. Lenguaje de programación

Descripción: El sistema deberá codificarse en el lenguaje C#, lenguaje con el cual trabaja UTE.

Referencias cruzadas: Todos los RF.

Prioridad: 1

RNF4. IDE de programación

Descripción: El IDE a utilizarse debe ser Visual Studio 2012, IDE con el cual trabaja UTE y tiene licencias.

Referencias cruzadas: Todos los RF.

Prioridad: 1

RNF5. Pruebas automatizadas

Descripción: Las pruebas sobre el sistema deben ser automatizadas e implementadas en conjunto con las funcionalidades.

Referencias cruzadas: RF20 al RF26

Prioridad: 1

RNF6. *Testing* unitarios

Descripción: Se deberán poder realizar *testing* unitarios al sistema.

Referencias cruzadas: Todos los RF.

Prioridad: 1

RNF7. Auditoría del sistema (log)

Descripción: Toda operación realizada en el sistema, así como los errores ocurridos durante la utilización del mismo, debe ser almacenada en un log para su posterior auditoría.

Referencias cruzadas: Todos los RF.

Prioridad: 1

RNF8. Utilización del *Framework* ITIL V3.

Descripción: Se utilizará ITIL V3 para la gestión de lanzamiento y la gestión de cambios, desarrollados en sus correspondientes capítulos.

Referencias cruzadas: Todos los RF.

Prioridad: 1

3.10 Casos de uso y casos de prueba

3.10.1 Estaciones, subestaciones y puesto de conexión.

CU1. Agrupar elementos de corte e identificación de configuración de celda

Pre-condiciones:-

Actores: Sistema, Usuario.

Post-condiciones:

Referencia cruzada: RF1, RF2

Curso normal:

1. El sistema identifica elementos de corte de una instalación.
2. El sistema agrupa los elementos de corte de una misma celda.
3. El sistema identifica posibles objetos compuestos de Mirage que representan dicha celda.
4. El sistema dispone al usuario un listado de esos posibles objetos con sus correspondientes representaciones gráficas.
5. El usuario selecciona el objeto que se ajusta a la configuración origen en el SCADA X.
6. El sistema imprime en el mímico final el objeto compuesto Mirage con las variables correspondientes.

Cursos alternativos:

3.1. Configuración de la celda no identificable con un objeto Mirage.

3. El sistema no identifica la configuración de la celda con ningún objeto compuesto de Mirage.
4. El usuario recarga una lista con todos los objetos de Mirage con sus respectivas representaciones gráficas.
5. El usuario selecciona el objeto que se ajusta a la configuración origen en el SCADA X.

6. El sistema imprime en el mímico final el objeto compuesto Mirage con las variables correspondientes.

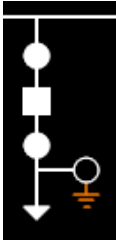
4.1. Los objetos en el listado no corresponden a la configuración de la celda.

4. El sistema dispone al usuario un listado de esos posibles objetos con sus correspondientes representaciones gráficas.
5. El usuario recarga una lista con todos los objetos de Mirage con sus respectivas representaciones gráficas.
6. El usuario selecciona el objeto que se ajusta a la configuración origen en el SCADA X.
7. El sistema imprime en el mímico final el objeto compuesto Mirage con las variables correspondientes.

4.2. No existe un objeto compuesto para a la configuración de la celda.

4. El sistema dispone al usuario un listado de esos posibles objetos con sus correspondientes representaciones gráficas.
5. El usuario recarga una lista con todos los objetos de Mirage con sus respectivas representaciones gráficas.
6. El usuario no selecciona ningún objeto.
7. El sistema no imprime en el mímico final ningún objeto compuesto Mirage referenciando a la celda.

Casos de pruebas:

Caso de prueba	CP1	
Propósito	Comprobar que el sistema agrupa los elementos de corte de una celda en un objeto de biblioteca Mirage.	
Prerrequisitos	La ejecución del programa debe corresponder a la conversión del mímico perteneciente a la estación EST5089.	
Datos de entrada	El sistema identifica la celda 6 en el nivel de tensión de 15kV del mímico de la estación EST5089.	
Imagen ilustrativa	 La imagen corresponde a la celdas 6 en 15kV de la estación EST5089 en el sistema SCADA X.	 La imagen corresponde a la representación de la celda 6 en 15kv de la estación EST5089 que debe tener el SCADA Mirage. Notar que elemento de PAT está a la derecha de la celda, esto es una convención de Mirage.

Pasos	• Seleccionar el objeto “CC_SECINTSECSPTDW” del listado de objetos mostrados por el sistema para la celda correspondiente. • Confirmar la selección de objeto realizada.
Resultado esperado	El sistema genera un mímico donde la representación de la celda 6 en 15kV está dada por el objeto Mirage “CC_SECINTSECSPTDW”.

Tabla 3-2: Caso de Prueba CP1

CU2. Persistir selección de usuario

Pre-condiciones:-

Actores: Sistema, Usuario.

Post-condiciones: El objeto Mirage queda asociado a la celda en el sistema.

Referencia cruzada: RF3

Curso normal:

1. El sistema dispone de un listado de objetos posibles para la celda.
2. El usuario selecciona el objeto que se ajusta a la configuración origen en el SCADA X.
3. El sistema guarda esa selección y ya queda ese mapeo-celda para posteriores ejecuciones.

Cursos alternativos:

2.1. El usuario omite la selección.

2. El usuario no selecciona ningún objeto.
3. El sistema no persiste la selección

Casos de pruebas:

Caso de prueba	CP2
Propósito	Comprobar que el sistema persiste la selección de objetos del usuario.
Prerrequisitos	La ejecución del programa debe corresponder a la conversión del mímico perteneciente a la estación EST5089.
Datos de entrada	El sistema identifica la celda 6 en el nivel de tensión de 15kV del mímico de la estación EST5089.
Imagen ilustrativa	 La imagen corresponde a la celdas 6 en 15kV de la estación EST5089 en el sistema SCADA X.

Pasos	• Seleccionar el objeto “CC_SECINTSECSPTDW” del listado de objetos mostrados por el sistema para la celda correspondiente. • Confirmar la selección de objeto realizada.
Resultado esperado	El sistema guarda la asociación entre la celda y el objeto “CC_SECINTSECSPTDW” seleccionado quedando disponible para posteriores ejecuciones.

Tabla 3-3: Caso de Prueba CP2

Caso de prueba	CP2.1
Propósito	Comprobar que el sistema no persiste las omisiones del usuario en la selección de objetos Mirage.
Prerrequisitos	La ejecución del programa debe corresponder a la conversión del mímico perteneciente a la estación EST5089.
Datos de entrada	El sistema identifica la celda 7 en el nivel de tensión de 15kV del mímico de la estación EST5089.
Imagen ilustrativa	 La imagen corresponde a cualquiera de las celdas 7 en 15kV de la estación EST5089 en el sistema SCADA X.
Pasos	• No seleccionar ningún objeto de lista mostrada por el sistema y continuar con la migración del mímico.
Resultado esperado	El sistema continúa con la migración sin guardar asociación de la celda con algún objeto. En futuras ejecuciones el sistema desplegará nuevamente un listado de objetos para dicha celda.

Tabla 3-4: Caso de Prueba CP2.1

CU3. Elementos dinámicos distintos a elementos de corte.

Pre-condiciones: El sistema posee un mapeo de elemento-objeto Mirage.

Actores: Sistema.

Post-condiciones:-

Referencia cruzada: RF4, RF6

Curso normal:

1. El sistema identifica un elemento dinámico, que no es elemento de corte.
2. El sistema imprime dicho elemento dinámico en el mímico generado.

Casos de pruebas:

Caso de prueba	CP3
Propósito	Comprobar que el sistema agrega automáticamente elementos que no corresponden a elementos de corte.
Prerrequisitos	La ejecución del programa debe corresponder a la conversión del mímico perteneciente al punto intermedio RE530014.
Datos de entrada	El sistema identifica la tensión de línea del RE530014.
Imagen ilustrativa	 La imagen corresponde a la tensión de línea del punto intermedio RE530014 en el sistema SCADA X.
Pasos	Ejecutar el sistema de conversión.
Resultado Esperado	El sistema debe agregar automáticamente en el mímico generado un objeto que represente a la tensión de línea del punto intermedio RE530014 con la migración sin guardar asociación de la celda con algún objeto.

Tabla 3-5: Caso de Prueba CP3

CU4. Conectividad en reenganches

Pre-condiciones: El mímico a convertir debe ser de una instalación que posea elemento de barra y reenganche, no aplica a puntos intermedios.

Actores: Sistema.

Post-condiciones:-

Referencia cruzada: RF5

Curso normal:

1. El sistema identifica los reenganches en el mímico del SCADA X.
2. El sistema identifica barras con sus respectivos niveles de tensión en el mímico SCADA X.
3. El sistema conecta con conectores transparentes todos los reenganches a la barra más próxima.

Casos de pruebas:

Caso de prueba	CP4
Propósito	Comprobar que el sistema agrega un conector simple y transparente entre un objeto de reenganche Mirage y la barra que le corresponde.

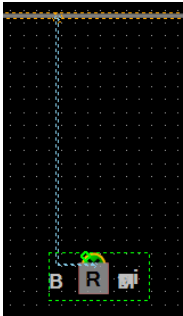
Prerrequisitos	La ejecución del programa debe corresponder a la conversión del mímico perteneciente a la estación EST5089.
Datos de entrada	El sistema identifica el reenganche de celda 06 en 15kV de la EST5089.
Imagen ilustrativa	 La imagen corresponde a la vista del reenganche en el editor Mirage, se visualiza el conector transparente entre la barra y el reenganche.
Pasos	Ejecutar el sistema de conversión.
Resultado Esperado	El sistema debe agregar automáticamente en el mímico generado un objeto reenganche conectado a la barra mediante un conector transparente.

Tabla 3-6: Caso de Prueba CP4

3.10.2 Puntos Intermedios.

CU5. Elementos dinámicos

Pre-condiciones: El mímico a convertir pertenece a un punto intermedio y el sistema posee un mapeo de elemento-objeto Mirage.

Actores: Sistema.

Post-condiciones: El sistema genera un mímico Mirage que representa un punto intermedio.

Referencia cruzada: RF6, RF4

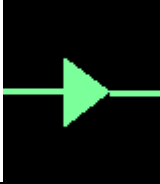
Curso normal:

1. El sistema identifica en SCADA X un elemento dinámico.
2. El sistema obtiene de su catálogo el objeto Mirage que le corresponde a dicho elemento dinámico.
3. El sistema agrega el objeto Mirage al mímico generado.

Casos de pruebas:

Caso de prueba	CP5
Propósito	Comprobar que el sistema agrega elementos dinámicos de puntos intermedios a su correspondiente mímico Mirage.
Prerrequisitos	La ejecución del programa debe corresponder a la conversión del mímico perteneciente al punto intermedio SZ530008.
Datos de entrada	El sistema identifica el seccionalizador en 6kV perteneciente al SZ530008.

Tabla 3-7: Caso de Prueba CP5

Imagen ilustrativa	 La imagen corresponde al seccionalizador del SZ530008 en el SCADA X.
Pasos	• Ejecutar el sistema de conversión.
Resultado esperado	El sistema debe agregar automáticamente en el mímico generado el objeto Mirage “D_006_SECV_DOBLE” que representa al seccionalizador en 6kV.

CU6. Color de elementos comandables.

Pre-condiciones: El sistema posee un mapeo de elemento-objeto por nivel de tensión Mirage.

Actores: Sistema.

Post-condiciones: El sistema genera un mímico Mirage que representa un punto intermedio.

Referencia cruzada: RF7

Curso normal:

1. El sistema identifica en SCADA X un elemento comandable y su nivel de tensión.
2. El sistema obtiene de su catálogo el objeto Mirage que le corresponde a dicho elemento comandable y a su mismo nivel de tensión.
3. El sistema agrega el objeto Mirage al mímico generado.

Casos de pruebas:

Caso de prueba	CP6
Propósito	Comprobar que el sistema agrega elementos comandables Mirage, con su respectivo color de tensión, a mímicos de puntos intermedios.
Prerrequisitos	La ejecución del programa debe corresponder a la conversión del mímico perteneciente al punto intermedio RE520002.
Datos de entrada	El sistema identifica al reconectador en 15kV perteneciente al RE520002.
Imagen ilustrativa	 La imagen corresponde al reconectador RE520002, cuyo elemento comandable en Mirage se representa con un cuadrado blanco. (En Mirage el color blanco está asociado 15kV para elementos de corte).
Pasos	• Ejecutar el sistema de conversión.
Resultado esperado	El sistema debe agregar automáticamente en el mímico generado el objeto Mirage “D_015_INTH_DOBLE_GR” que representa al reconectador en 15kV.

Tabla 3-8: Caso de Prueba CP6

3.10.3 Unifilares en general

CU7. Mantener conectividad en instalaciones.

Pre-condiciones: La instalación a migrar debe tener conectividad entre sus elementos.

Actores: Sistema, Usuario.

Post-condiciones: El sistema genera un mímico Mirage con conectividad.

Referencia cruzada: RF8, RF1, RF2

Curso normal:

1. El sistema identifica elementos conectados.
2. El sistema dispone al usuario un listado de objetos compuestos Mirage para su selección.
3. El usuario selecciona el objeto correspondiente a la configuración de elementos en SCADA X.
4. El sistema agrega el objeto seleccionado al mímico generado, respetando las conexiones en SCADA X.
5. El sistema agrega conectividad entre los elementos seleccionados por el usuario y otros elementos no seleccionables, como son las barras.
6. El sistema genera un mímico con conectividad, respetando la topología del mímico en el SCADA X.

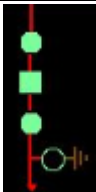
Cursos alternativos:

3.1. El usuario selecciona erróneamente un objeto.

3. El usuario selecciona un objeto que no cumple la configuración de elementos en SCADA X.
4. El sistema agrega conectividad entre los elementos seleccionados por el usuario y otros elementos no seleccionables, como son las barras.
5. El sistema genera un mímico que no respeta la conectividad topológica del SCADA X.

Tabla 3-9: Caso de Prueba CP7

Casos de pruebas:

Caso de prueba	CP7
Propósito	Comprobar que el sistema mantiene la conectividad de una instalación del sistema SCADA X en SCADA Mirage.
Prerrequisitos	La ejecución del programa debe corresponder a la conversión del mímico perteneciente a la estación EST5079.
Datos de entrada	El sistema identifica todos los elementos conectados de la estación EST5079.
Imagen ilustrativa	 Las figuras verdes representan los elementos de corte mientras que las líneas rojas son los conectores entre los mismos.

Pasos	• Ejecutar el sistema de conversión. • Seleccionar los objetos que representan celdas de corte, cuando sea necesario.
Resultado esperado	El sistema debe generar un mímico Mirage donde los elementos de corte de la EST5079 estén conectados de forma análoga a como lo están en el SCADA X.

CU8. Integrar mímicos con sistema SIO.

Pre-condiciones: El mímico a migrar debe estar integrado con SIO

Actores: Sistema.

Post-condiciones: El sistema guarda un mapeo entre el mímico generado y el identificador correspondiente en SIO.

Referencia cruzada: RF9

Curso normal:

1. El sistema genera un mímico correspondiente a una o más instalaciones.
2. El sistema obtiene el identificador SIO de dichas instalaciones.
3. El sistema persiste un mapeo entre el mímico generado y el identificador SIO de las instalaciones contenidas en el mímico.

Casos de pruebas:

Caso de prueba	CP8
Propósito	Comprobar que el sistema persiste el mapeo entre nombre del mímico migrado y el identificador SIO de cada instalación.
Prerrequisitos	La ejecución del programa debe corresponder a la conversión de al menos una instalación integrada con SIO.
Datos de entrada	El mímico a migrar corresponde a la llave compuesta por los reconvertadores RE530001 y RE530002.
Pasos	• Ejecutar el sistema de conversión. • Cuando el sistema solicite un nombre para guardar el mímico, guardarlo con el nombre de LLAPOLPRANDI.
Resultado esperado	El sistema debe guardar, en algún lugar el mapeo entre el mímico LLAPOLPRANDI y los identificadores 72595742 y 72598440 pertenecientes a los reconvertadores RE530001 y RE530002 respectivamente.

Tabla 3-10: Caso de Prueba CP8

CU9. Inyecciones de conectividad

Pre-condiciones: El mímico a convertir debe tener conectividad.

Actores: Sistema.

Post-condiciones: El sistema genera un mímico con conectividad y con sus distintas inyecciones o fuentes de conectividad.

Referencia cruzada: RF10, RF1, RF2, RF7

Curso normal:

1. El sistema identifica inyecciones de conectividad en el mímico de SCADA X.
2. El sistema define un color distinto a cada inyección.
3. El sistema conecta las inyecciones a los elementos correspondientes en el mímico Mirage.
4. El sistema genera un mímico con conectividad, cuyas inyecciones están claramente diferenciadas.

Casos de pruebas:

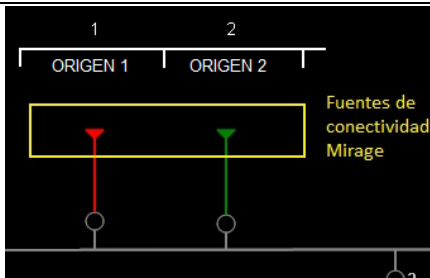
Caso de prueba	CP9
Propósito	Comprobar que el sistema agrega fuentes de conectividad de distinto color al mímico Mirage.
Prerrequisitos	La ejecución del programa debe corresponder a la conversión del mímico perteneciente a la estación EST5052.
Datos de entrada	El sistema identifica las inyecciones de conectividad en el SCADA X para las celdas 1 y 2 en 30kV de la estación EST5052.
Imagen ilustrativa	 Representación de la fuente de conectividad en Mirage.
Pasos	• Ejecutar el sistema de conversión.
Resultado esperado	El sistema debe generar un mímico Mirage correspondiente a la EST5079 que contenga fuentes de conectividad de distinto color en celdas 1 y 2 en 30kV.

Tabla 3-11: Caso de Prueba CP9

CU10. Formato de textos estáticos y celdarios.

Pre-condiciones:-

Actores: Sistema.

Post-condiciones: El sistema genera un mímico Mirage con textos uniformizados.

Referencia cruzada: RF11

Curso normal:

1. El sistema identifica por separado textos estáticos y textos de celdarios en el mímico del SCADA X.

2. El sistema define a los textos de celdarios con tamaño 10 y fuente "Arial" color blanco.
3. El sistema define al resto de textos estáticos con un tamaño de fuente acorde a las proporciones del mímico, y le deja el mismo color de origen y le define un tipo de fuente por defecto.
4. El sistema genera un mímico Mirage conteniendo textos ajustados y celdarios uniformizados.

Cursos alternativos:

1.1. El mímico de SCADA X corresponde a un punto intermedio y no posee celdario.

1. El sistema identifica por textos estáticos en el mímico del SCADA X.
2. El sistema define a los textos estáticos con un tamaño de fuente acorde a las proporciones del mímico, y le deja el mismo color de origen y le define un tipo de fuente por defecto.
3. El sistema genera un mímico Mirage conteniendo textos estáticos ajustados.

Casos de pruebas:

Caso de prueba	CP10	
Propósito	Comprobar que el sistema genera un mímico cuyos textos de celdarios están definidos con fuente "Arial", tamaño 10, de color blanco. Además, el resto de los textos estáticos están definidos con las proporciones aproximadas al mímico de origen.	
Prerrequisitos	La ejecución del programa debe corresponder a la conversión del mímico perteneciente a la estación EST5079.	
Datos de entrada	El sistema identifica textos del celdarios y textos estáticos en el unifilar de origen.	
Imagen ilustrativa	 Ejemplo de textos en celdarios de la EST5079.	 Ejemplos de texto estático en el unifilar que no pertenece al celdario. En este caso corresponde a la letra "A", que identifica la barra A en 15kV.
Pasos	• Ejecutar el sistema de conversión.	
Resultado esperado	El sistema debe generar un mímico Mirage para la EST5079 con los textos de celdarios en el formato especificados y el resto de los textos aproximadamente con las mismas proporciones de origen.	

Tabla 3-12: Caso de Prueba CP10

Caso de prueba	CP10.1
Propósito	Comprobar que el sistema genera un mímico Mirage de punto intermedio cuyos textos estáticos están definidos en proporciones aproximadas al mímico de origen.
Prerrequisitos	La ejecución del programa debe corresponder a la conversión del mímico perteneciente al punto intermedio SZ530008.
Datos de entrada	El sistema identifica textos estáticos en el unifilar de origen.
Imagen ilustrativa	 Ejemplo de textos estáticos, que corresponde a la cabecera de la línea del punto intermedio SZ530008.
Pasos	Ejecutar el sistema de conversión.
Resultado esperado	El sistema debe generar un mímico Mirage para el SZ530008 con los textos estáticos aproximadamente con las mismas proporciones de origen.

Tabla 3-13: Caso de Prueba CP10.1

CU11. Descartar textos estáticos de unidades de medidas eléctricas.

Pre-condiciones: El mímico origen contiene variables de medida.

Actores: Sistema.

Post-condiciones: El sistema genera un mímico Mirage cuyas medidas están representadas con un objeto Mirage.

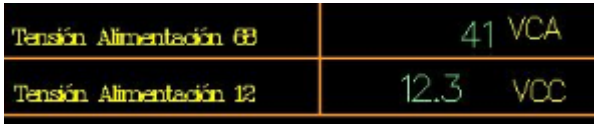
Referencia cruzada: RF12

Curso normal:

1. El sistema identifica variables de medida en el mímico del SCADA X.
2. El sistema identifica texto estático correspondiente a unidades.
3. El sistema no tiene en cuenta dichos textos y no los incluye en el mímico a generar.
4. El sistema genera un mímico Mirage sin textos de unidades, en su lugar contiene un objeto Mirage que contiene la unidad de la medida correspondiente.

Tabla 3-14: Caso de Prueba CP11

Casos de pruebas:

Caso de prueba	CP11
Propósito	Comprobar que el sistema genera un mímico Mirage que no tiene ningún texto que represente una unidad de medida, y en su lugar inserta un objeto Mirage que ya contiene la medida correspondiente.
Prerrequisitos	La ejecución del programa debe corresponder a la conversión del mímico perteneciente al punto intermedio SZ530008.
Datos de entrada	El sistema identifica textos que representan unidades de medidas en el unifilar de origen.
Imagen ilustrativa	

	Los textos estáticos “VCA” y “VCC” representan medidas de tensión alterna y continua, respectivamente, en el mímico del punto intermedio SZ530008.
Pasos	Ejecutar el sistema de conversión.
Resultado esperado	El sistema debe generar un mímico Mirage para el SZ530008 sin textos estáticos que representen unidades de medidas eléctricas.

CU12. Migración nombre interno instalación.

Pre-condiciones: Aplica a cualquier mímico de instalación.

Actores: Sistema.

Post-condiciones: El sistema inserta el objeto Mirage que muestra el nombre de la instalación.

Referencia cruzada: RF13

Curso normal:

1. El sistema identifica el nombre de la instalación a partir de las variables del mímico.
2. El sistema imprime en el mímico el objeto “Cuadro instalación” con los parámetros referenciados a la instalación.
3. El sistema genera un mímico Mirage que contiene el objeto “Cuadro instalación”.

Casos de pruebas:

Caso de prueba	CP12
Propósito	Comprobar que el sistema genera un mímico Mirage que contiene el objeto Mirage “000_ESTACION” parametrizado correctamente.
Prerrequisitos	La ejecución del programa debe corresponder a la conversión del mímico perteneciente a la estación EST5079.
Datos de entrada	El sistema identifica el nombre de la instalación.
Imagen ilustrativa	 La imagen corresponde a la vista en el editor Mirage del objeto 000_ESTACION que debe estar inserto en el mímico con los parámetros de la EST5079.
Pasos	Ejecutar el sistema de conversión.
Resultado esperado	El sistema debe generar un mímico Mirage para la EST5079 que contenga el cuadro de estación de la misma, es decir, el objeto Mirage “000_ESTACION”.

Tabla 3-15: Caso de Prueba CP12

CU13. Comando habilitar/deshabilitar relé.

Pre-condiciones: El mímico origen debe contener al menos un elemento de comando habilitar/deshabilitar relé.

Actores: Sistema.

Post-condiciones: El sistema genera un mímico que contiene un botón Mirage que genera automáticamente una ventana con los comandos habilitar/deshabilitar relé de la instalación.

Referencia cruzada: RF14

Curso normal:

1. El sistema identifica los elementos de comando habilitar/deshabilitar relé en el mímico de SCADA X.
2. El sistema imprime en el mímico Mirage un botón de script automático parametrizado adecuadamente para que genere la ventana de los comandos habilitar/deshabilitar relé de la instalación.
3. El sistema genera un mímico Mirage que contiene un botón de script automático para visualizar los comandos de habilitar/deshabilitar relé.

Casos de pruebas:

Caso de prueba	CP13	
Propósito	Comprobar que el sistema genera un mímico Mirage que contiene el objeto Mirage “AAA_SCRIPT_RELE” parametrizado correctamente.	
Prerrequisitos	La ejecución del programa debe corresponder a la conversión del mímico perteneciente a la estación EST5079.	
Datos de entrada	El sistema identifica elementos de Hab/Deshab relé en la estación EST5079.	
Imagen ilustrativa	 Ejemplo del comando hab/dehab rele en SCADA X.	 Botón de script automático Mirage que abre ventana con los comandos de hab/dehab relé de la estación.
Pasos	• Ejecutar el sistema de conversión.	
Resultado esperado	El sistema debe generar un mímico Mirage para la EST5079 que contenga el objeto Mirage “AAA_SCRIPT_RELE” que representa un botón que abre una ventana con los comandos Hab/Deshab relé de la instalación EST5079.	

Tabla 3-16: Caso de Prueba CP13

CU14. Selección por área.

Pre-condiciones: Las áreas 1, 2 y 3 deben estar definidas en el SCADA X.

Actores: Sistema, Usuario.

Post-condiciones: El sistema genera un mímico Mirage para cada instalación de cada área seleccionada.

Referencia cruzada: RF15, RF16

Curso normal:

1. El usuario le indica al sistema el área de las instalaciones a migrar.
2. El usuario inicia el sistema.
3. El sistema identifica las instalaciones correspondiente al área definida.
4. El sistema inicia la migración de las instalaciones.
5. El sistema genera para cada instalación su correspondiente mímico Mirage.

Casos de pruebas:

Caso de prueba	CP14
Propósito	Comprobar que el sistema genera los mímicos correspondientes al área definida por el usuario.
Prerrequisitos	La ejecución del programa debe corresponder a la conversión de mímicos de instalaciones de área 2.
Datos de entrada	El sistema identifica todas las instalaciones de área 2 del SCADA X.
Pasos	• Definir área 2 al sistema para la conversión. • Ejecutar el sistema de conversión. • Se pueden omitir todas las interacciones con el sistema
Resultado esperado	El sistema debe informar al usuario, cada vez que comienza una nueva migración de mímico, la instalación a migrar y todas deben pertenecer a área 2.

Tabla 3-17: Caso de Prueba CP14

CU15. Selección de instalación a migrar.

Pre-condiciones: La o las instalaciones seleccionadas deben existir en el SCADA X.

Actores: Sistema, Usuario.

Post-condiciones: El sistema genera un mímico Mirage para cada instalación seleccionada.

Referencia cruzada: RF16, RF15

Curso normal:

1. El usuario le indica al sistema un conjunto de instalaciones a migrar.
2. El usuario inicia el sistema.
3. El sistema identifica las instalaciones seleccionadas por el usuario.
4. El sistema inicia la migración de las instalaciones.
5. El sistema genera para cada instalación su correspondiente mímico Mirage.

Casos de pruebas:

Caso de prueba	CP15
Propósito	Comprobar que el sistema genera los mímicos correspondientes a las instalaciones definidas por el usuario.
Prerrequisitos	La ejecución del programa debe corresponder a la conversión de mímicos de instalaciones EST5052 y EST5079.
Datos de entrada	El sistema identifica los unifilares a migrar de las instalaciones EST5052 y EST5079.
Pasos	• Definir la estación EST5052 y EST5079 al sistema para la conversión. • Ejecutar el sistema de conversión. • Se seleccionan los objetos correspondientes a las celdas de corte de cada estación cada vez que el sistema lo requiera.
Resultado esperado	El sistema debe generar un mímico para la estación EST5052 y otro para la EST5079.

Tabla 3-18: Caso de Prueba CP15

CU16. Variables en bandera DPF.

Pre-condiciones: El mímico a convertir debe contener al menos un DPF (detector de paso de falta).

Actores: Sistema.

Post-condiciones: El sistema genera un mímico Mirage que contiene banderas DPF.

Referencia cruzada: RF17

Curso normal:

1. El sistema identifica variables correspondientes a DPF en el mímico a convertir.
2. El sistema discrimina cuáles variables de esas corresponden a variables calculadas (VDM) y cuáles no.
3. El sistema asigna las variables calculadas al objeto Mirage que representa una bandera DPF y solo admite variables calculadas.
4. El sistema asigna variables simples al objeto Mirage que representa una bandera DPF y solo admite una sola variable.
5. El sistema genera un mímico Mirage con banderas DPF.

Casos de pruebas:

Caso de prueba	CP16
Propósito	Comprobar que el sistema genera un mímico Mirage que contiene el objeto Mirage que representa la bandera DPF (detector de paso de falta).
Prerrequisitos	La ejecución del programa debe corresponder a la conversión del mímico perteneciente a la estación EST5S21.
Datos de entrada	El sistema identifica la variable correspondiente a la alarma de paso de falta en la EST5S21.
Imagen ilustrativa	 Representación en los sistemas SCADA de la bandera DPF.
Pasos	• Ejecutar el sistema de conversión. • Interactuar con el sistema cuando sea necesario.
Resultado esperado	El sistema debe generar un mímico Mirage para la EST5S21 que contenga el objeto “000_DPF_FLAG” correspondiente a la alarma DPF para las celdas 1, 2 y 5.

Tabla 3-19: Caso de Prueba CP16

CU17. Mostrar nombre de variable en la conversión.

Pre-condiciones: El mímico a convertir no pertenece a un punto intermedio.

Actores: Sistema, Usuario.

Post-condiciones: El usuario identifica la celda de origen en SCADA X para que sea migrada por el sistema en un mímico Mirage.

Referencia cruzada: RF18, RF1, RF2, RF10

Curso normal:

1. El sistema identifica un elemento dinámico a convertir.
2. El sistema obtiene la variable de instalación correspondiente a dicho elemento dinámico.
3. El sistema le provee al usuario el nombre de variable del elemento y una lista de objetos Mirage que pueden representar dicho elemento.
4. El usuario selecciona un objeto para representar dicho elemento, teniendo en cuenta su representación gráfica y la ubicación de dicha variable en el mímico original del SCADA X.
5. El sistema imprime el objeto Mirage en el mímico generado, cuya animación estará dada por la variable en cuestión.

Casos de pruebas:

Caso de prueba	CP17
Propósito	Comprobar que el sistema muestra el nombre de la variable en el proceso de conversión.
Prerrequisitos	La ejecución del programa debe corresponder a la conversión del mímico de la estación EST5079.
Datos de entrada	El sistema identifica la variable de “Reset de SCADA Local” de la EST5079.
Pasos	• Ejecutar el sistema de conversión. • El sistema muestra la variable “CAEST5079_000_GEN_RSL”, “Reset de SCADA Local”. • Seleccionar el objeto “000_RES_SCADA_LOC”.
Resultado esperado	El sistema debe mostrar la variable al usuario y generar un mímico Mirage de la EST5079, que contenga el objeto “000_RES_SCADA_LOC” referenciado a la variable “CAEST5079_000_GEN_RSL”.

Tabla 3-20: Caso de Prueba CP17

CU18. Reemplazar panel de comunicaciones

Pre-condiciones:-

Actores: Sistema.

Post-condiciones: El sistema genera un mímico Mirage con botón de comunicaciones.

Referencia cruzada: RF19

Curso normal:

1. El sistema identifica elementos del panel de comunicaciones en el mímico original en el SCADA X.
2. El sistema descarta dichos elementos.
3. El sistema obtiene parámetros de la comunicación de la instalación a migrar.
4. El sistema define esos parámetros en un botón Mirage de script automático para las comunicaciones.
5. El sistema imprime dicho botón al mímico Mirage.
6. El sistema genera un mímico Mirage que contiene un botón de comunicaciones.

Casos de pruebas:

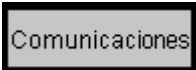
Caso de prueba	CP18	
Propósito	Comprobar que el sistema genera un mímico Mirage que contiene un botón que abre una ventana con información de comunicaciones.	
Prerrequisitos	La ejecución del programa debe corresponder a la conversión del mímico de la estación EST5079.	
Datos de entrada	El sistema identifican las variables de comunicación correspondientes a la estación EST5079.	
Imagen ilustrativa		Panel de Comunicaciones en SCADA X.  Botón de comunicaciones en Mirage
Pasos	• Ejecutar el sistema de conversión.	
Resultado esperado	El sistema debe generar un mímico Mirage para la EST5079 que contenga el objeto "AAA_SCRIPT_COM" que corresponde al botón de comunicaciones Mirage.	

Tabla 3-21: Caso de Prueba CP18

3.10.4 Módulo de pruebas automatizadas.

CU19. Instalaciones cruzadas.

Pre-condiciones: Aplica sobre mímicos del SCADA Mirage.

Actores: Sistema, Usuario.

Post-condiciones: El sistema debe reportar las instalaciones que tienen variables en mímicos de otras instalaciones.

Referencia cruzada: RF20

Curso normal:

1. El usuario le define al sistema el directorio que contiene los mímicos Mirage.
2. El usuario inicia el sistema.
3. El sistema chequea las variables de cada mímico Mirage y a partir de ellas obtiene la instalación o instalaciones pertenecientes a dicho mímico.
4. El sistema evalúa si la instalación pertenece o no al mímico.
5. El sistema provee al usuario un listado de instalaciones que están en mímicos de otras instalaciones.

Casos de pruebas:

Caso de prueba	CP19
Propósito	Comprobar que el sistema identifica elementos de instalaciones en mímico de otras instalaciones donde no corresponde estar.
Prerrequisitos	Deben existir mímicos Mirage.
Datos de entrada	Archivos correspondientes a los mímicos Mirage del CMD y el mapeo de mímico-instalación generado por el sistema de migración gráfica.
Pasos	• Definir directorio donde se encuentran los mímicos Mirage. • Ejecutar el sistema de prueba.
Resultado Esperado	El sistema despliega al usuario un listado de instalaciones en mímicos donde no deben estar.

Tabla 3-22: Caso de Prueba CP19

CU20. Referencias de celda en celdario.

Pre-condiciones: No aplica a mímicos de puntos intermedios.

Actores: Sistema, Usuario.

Post-condiciones: El sistema genera un reporte con celda mal referenciadas.

Referencia cruzada: RF21, RF22

Curso normal:

1. El usuario le define al sistema el directorio que contiene los mímicos Mirage.
2. El usuario inicia el sistema.
3. El sistema chequea las variables de cada mímico Mirage que corresponden a una misma celda y las agrupa.
4. El sistema obtiene a partir de la agrupación el número de dicha celda y las coordenadas máximas y mínimas en el eje x relativas al mímico.

5. El sistema obtiene los textos que representan números de los celdarios.
6. El sistema chequea que el texto más próximo a la celda corresponda al número de dicha celda y además que el texto tenga las coordenadas en el eje x contenidas entre las coordenadas en el eje x de la celda.
7. El sistema reporta al usuario las celdas que no tienen correspondencia en el celdario.

Casos de pruebas:

Caso de prueba	CP20
Propósito	Comprobar que el sistema identifica celdas de corte que no tienen su número de celdario bien referenciados.
Prerrequisitos	Deben existir mímicos Mirage.
Datos de entrada	Archivos correspondientes a los mímicos Mirage del CMD.
Pasos	• Definir directorio donde se encuentran los mímicos Mirage. Ejecutar el sistema de prueba.
Resultado Esperado	El sistema despliega al usuario un listado de celdas de corte mal referenciadas en el celdario correspondiente.

Tabla 3-23: Caso de Prueba CP20

CU21. Botón de comunicación 101.

Pre-condiciones: Solo aplica a mímico con instalaciones con enlace 101.

Actores: Sistema, Usuario.

Post-condiciones: El sistema genera un reporte de mímicos cuyo botón de comunicaciones 101 está mal parametrizado.

Referencia cruzada: RF23

Curso normal:

1. El usuario le define al sistema el directorio que contiene los mímicos Mirage.
2. El usuario le define al sistema el archivo de configuración del protocolo 101, que mapea instalaciones con los sistemas parciales 101.
3. El usuario inicia el sistema.
4. En el sistema se genera un índice de instalaciones que tienen enlace 101, y sus correspondientes sistemas parciales.
5. El sistema chequea el botón de comunicaciones en los mímicos de dichas instalaciones tengan como parámetro el mismo sistema parcial.
6. El sistema despliega al usuario un listado de mímicos cuyo botón de comunicaciones no tiene parametrizado correctamente el sistema parcial, junto con el cual sería el correcto sistema parcial para su corrección

Casos de pruebas:

Caso de prueba	CP21
Propósito	Comprobar que el sistema identifica botones de comunicación mal parametrizados en mímicos de instalaciones cuyo enlace es el protocolo 101.
Prerrequisitos	Deben existir mímicos Mirage.
Datos de entrada	Archivos correspondientes a los mímicos Mirage del CMD y archivo de configuración del protocolo 101.
Pasos	- Definir directorio donde se encuentran los mímicos Mirage. - Definir directorio donde se encuentra el archivo de configuración 101. Ejecutar el sistema de prueba.
Resultado Esperado	El sistema despliega al usuario un listado de mímicos cuyo botón de comunicaciones 101 está mal configurado, proponiendo una configuración correcta para el mismo.

Tabla 3-24: Caso de Prueba CP21

CU22. Invocación de mímicos.**Pre-condiciones:-****Actores:** Sistema, Usuario.**Post-condiciones:** El sistema genera un reporte con los mímicos que no son llamados de ningún botón en los mímicos Mirage.**Referencia cruzada:** RF24**Curso normal:**

1. El usuario le define al sistema el directorio que contiene los mímicos Mirage.
2. El usuario inicia el sistema.
3. El sistema recorre los mímicos y chequea todos los botones que hacen referencias a mímicos.
4. El sistema despliega al usuario un listado de mímicos Mirage que no están referenciados en ningún botón.

Casos de pruebas:

Caso de prueba	CP22
Propósito	Comprobar que el sistema identifica mímicos Mirage que no son invocados desde ningún lugar.
Prerrequisitos	Deben existir mímicos Mirage.
Datos de entrada	Archivos correspondientes a los mímicos Mirage del CMD.
Pasos	Definir directorio donde se encuentran los mímicos Mirage. Ejecutar el sistema de prueba.
Resultado Esperado	El sistema despliega al usuario un listado de mímicos que no son invocados desde ningún lugar.

Tabla 3-25: Caso de Prueba CP22

CU23. Conectividad de instalaciones.**Pre-condiciones:** No aplica a mímicos de puntos intermedios.**Actores:** Sistema, Usuario.**Post-condiciones:** El sistema genera un reporte indicando que elementos y en que mímicos no cumple la conectividad de origen en el SCADA X.**Referencia cruzada:** RF25**Curso normal:**

1. El usuario le define al sistema el directorio que contiene los mímicos Mirage.
2. El usuario inicia el sistema.
3. El sistema recorre los mímicos y para cada uno compara la conectividad que tiene actualmente en el mímico Mirage con la que tenía en el mímico SCADA X.
4. El sistema despliega al usuario un listado de elementos que no cumplen con su conectividad original en SCADA X.

Casos de pruebas:

Caso de prueba	CP23
Propósito	Comprobar que el sistema identifica elementos en los mímicos Mirage que no cumplen con su conectividad original en SCADA X.
Prerrequisitos	Deben existir mímicos Mirage con conectividad entre sus elementos.
Datos de entrada	Archivos correspondientes a los mímicos Mirage del CMD, que contengan conectividad entre sus elementos.
Pasos	• Definir directorio donde se encuentran los mímicos Mirage. Ejecutar el sistema de prueba.
Resultado Esperado	El sistema despliega al usuario un listado mímico-elementos, donde esos elementos no cumplen la conectividad de origen.

Tabla 3-26: Caso de Prueba CP23

CU24. Celdas de corte solapadas.**Pre-condiciones:** No aplica a mímicos de puntos intermedios.**Actores:** Sistema, Usuario.**Post-condiciones:** El sistema genera un reporte indicando que elementos y en que mímicos no cumple la conectividad de origen en el SCADA X.**Referencia cruzada:** RF22, RF21**Curso normal:**

1. El usuario le define al sistema el directorio que contiene los mímicos Mirage.
2. El usuario inicia el sistema.
3. El sistema chequea las variables de cada mímico Mirage que corresponden a una misma celda y las agrupa.
4. El sistema obtiene a partir de las agrupaciones, distintas celda y sus coordenadas máximas y mínimas en el eje x relativas al mímico.
5. El sistema chequea las celdas dos a dos para que no hayan coordenadas de una contenidas en las coordenadas de la otra.

6. El sistema reporta al usuario las celdas de un mímico que tengan solapamientos en coordenadas de otra celda.

Casos de pruebas:

Caso de prueba	CP24
Propósito	Comprobar que el sistema identifica celdas de corte que se solapan con otras.
Prerrequisitos	Deben existir mímicos Mirage.
Datos de entrada	Archivos correspondientes a los mímicos Mirage del CMD.
Pasos	• Definir directorio donde se encuentran los mímicos Mirage. • Ejecutar el sistema de prueba.
Resultado esperado	El sistema despliega al usuario un listado de celdas de corte con sus correspondientes coordenadas y con qué otras celdas tiene solapamiento.

Tabla 3-27: Caso de Prueba CP24

CU25. Mímicos con instalaciones migradas automáticamente.

Pre-condiciones:-

Actores: Sistema, Usuario.

Post-condiciones: El sistema genera un reporte con los mímicos que contienen instalaciones no migradas por el sistema de conversión gráfica.

Referencia cruzada: RF26

Curso normal:

1. El usuario le define al sistema el directorio que contiene los mímicos Mirage.
2. El usuario inicia el sistema.
3. El sistema recorre los mímicos y chequea que las instalaciones contenidas estén en el mapeo mímico-instalación del sistema de conversión gráfica.
4. El sistema despliega al usuario un listado de mímicos con instalaciones que no fueron contempladas en el proceso de migración gráfica.

Casos de pruebas:

Caso de prueba	CP25
Propósito	Comprobar que el sistema identifica mímicos Mirage que tienen instalaciones que no fueron contempladas en la conversión gráfica.
Prerrequisitos	Deben existir mímicos Mirage.
Datos de entrada	Archivos correspondientes a los mímicos Mirage del CMD.
Pasos	• Definir directorio donde se encuentran los mímicos Mirage. • Ejecutar el sistema de prueba.
Resultado esperado	El sistema despliega al usuario un listado de mímicos con instalaciones que no fueron migradas por el sistema de conversión gráfica.

Tabla 3-28: Caso de Prueba CP25

4 Arquitectura

En esta sección nos enfocaremos en la arquitectura de la pieza clave en el proceso de migración, como lo es el software de conversión. Dicha sección incluirá:

- ✚ Decisiones de diseño que se adoptaron para cumplir con los atributos de calidad más relevantes del sistema.
- ✚ Perspectiva general de la arquitectura, sus objetivos y las principales decisiones de diseño tomadas.
- ✚ Descripción detallada a nivel de arquitectura y diseño.
- ✚ Análisis de la satisfacción de los atributos de calidad planteados al inicio.
- ✚ Conclusiones y lecciones aprendidas luego de la definición de la arquitectura y la implementación de la solución.

4.1 Atributos de calidad

Basados en los atributos de calidad se definió la arquitectura. Son el motivo por el cual se tomaron la mayoría de las decisiones a nivel arquitectónico o de diseño.

Se realiza un análisis de los atributos de calidad para asegurar el cumplimiento de los requerimientos no funcionales.

En la Figura 4-1 se muestran los atributos de calidad de acuerdo a la prioridad otorgada, para luego explicar su importancia en el contexto particular de la solución a construir.

Testeabilidad
Usabilidad **Modificabilidad**

Figura 4.1: Atributos de calidad priorizados

4.1.1 Testeabilidad

Entendemos que la clave del éxito del proyecto radica en el diseño de las pruebas lo más robustas y completas que puedan ser.

Además de las pruebas unitarias en software de conversión, se harán muchas pruebas que se enfocan en el producto, que surgen de la integración entre:

- Los outputs del software de conversión
- Sistemas *front end*.
- Base de datos.

Estos dos últimos migrados por el cliente.

Por lo tanto, favorecer este atributo de calidad desde la definición de la arquitectura aparece como fundamental; si el producto obtenido del software de conversión tiene muchos defectos ocasionará que durante las pruebas de la solución los mismos se multipliquen.

Consecuentemente, para que el sistema cumpla con la propiedad de ser testeable se optó por usar el patrón de diseño Composite, donde se comenzará por objetos más simples realizando pruebas unitarias en los mismos, para luego pasar a los objetos más complejos y realizar pruebas de integración.

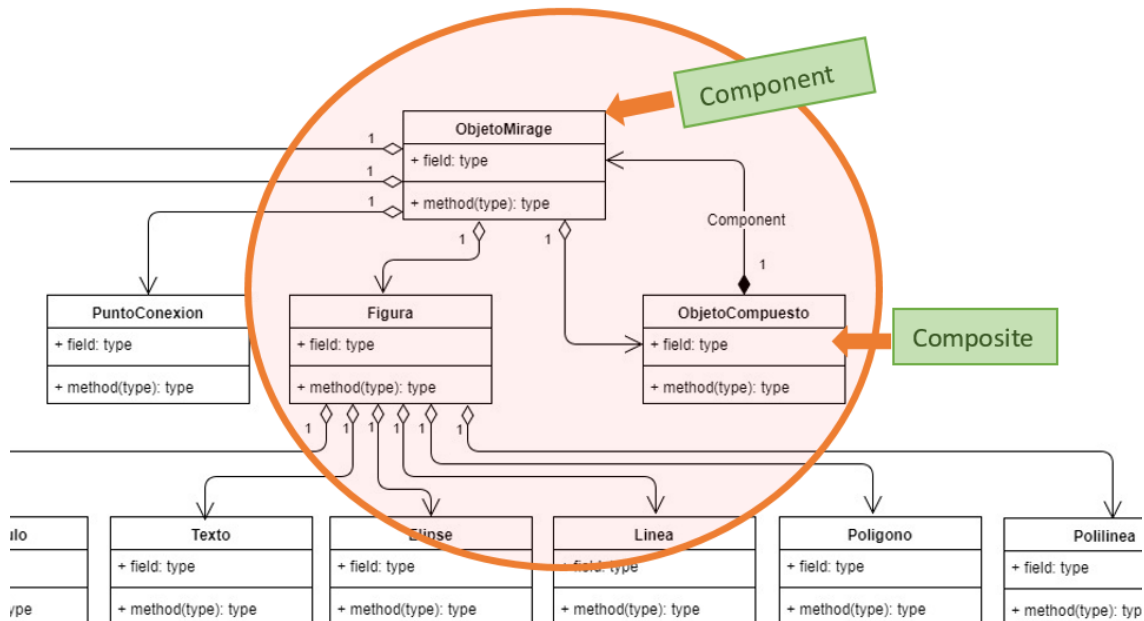


Figura 4.2: Patrón composite aplicado

4.1.2 Modificabilidad

Una vez que se conviertan todos los diagramas y se integre la solución comenzarán las pruebas de integración. Los defectos encontrados se deberán reportar y disparará cambios en el software.

Se definió un sistema de pruebas para la solución, en los cuales se reportan los bugs y se deben corregir en la aplicación, por lo tanto, la misma debe ser modificable. Los bugs reportados van a generar que muchos objetos puedan ser cambiados o agregados, por lo que se optó por usar el método Fabrica para la creación de objetos.

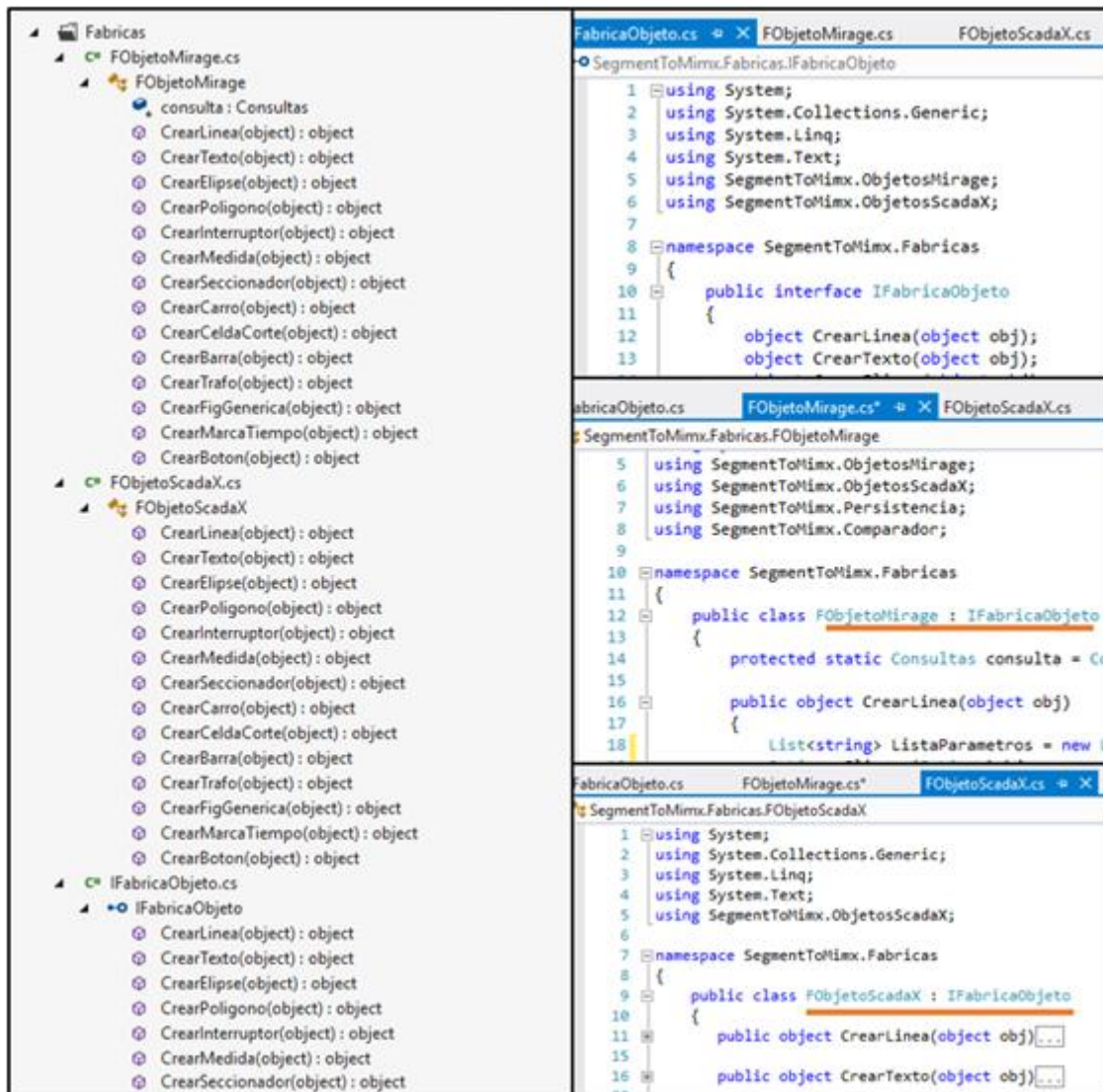


Figura 4.3: Patrón Fábrica Aplicado.

4.1.3 Usabilidad

El sistema se usará para dos migraciones, que serán realizadas por diferentes usuarios y para migrar un gran número de diagramas; si bien la cantidad de usuarios será reducida, no se deberá perder el foco usabilidad. De los tres es el atributo menos importante.

Para favorecer la usabilidad se hizo énfasis en bajar la cantidad de interacción con el cliente, únicamente se debe ingresar el diagrama a convertir y en caso de ambivalencias de elementos o que no reconozca el objeto origen se deberá escoger qué elemento de todos los de Mirage corresponde.

En beneficio de la usabilidad a la hora de elegir la opción correcta, se tomaron las siguientes medidas:

- El sistema acota las opciones lo más que puede, dejando un número pequeño de elecciones posibles. Además, le señala la opción que cree la más acertada, según los parámetros analizados.
- Para cada elemento de Mirage se muestra por pantalla su correspondiente diseño gráfico, facilitando la elección del mismo.
- Se persiste en una base de datos que fue elegida para dicho caso, cuando se vuelve a repetir el mismo contexto el programa lo hace de manera automática.

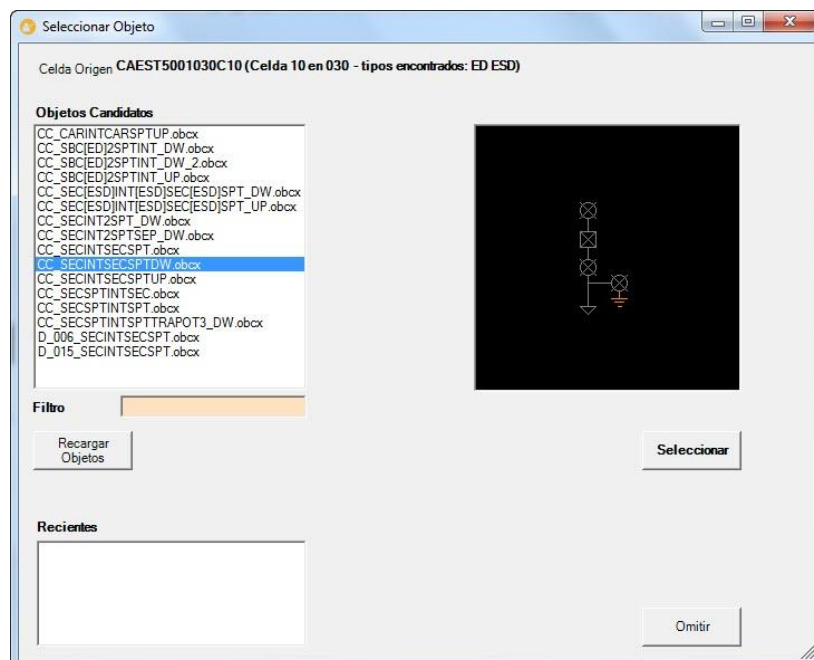


Figura 4.4: Elección de objetos candidatos.

4.2 Descripción de la Arquitectura

4.2.1 Presentación de componentes

A continuación, se presentan los distintos componentes relevantes que integran la solución del proyecto, se diseñó el sistema actual, el software de migración, la situación futura y cómo interactúan entre sí. Luego se incluye un catálogo que brinda información de los mismos.

Diagrama de componentes y conectores.

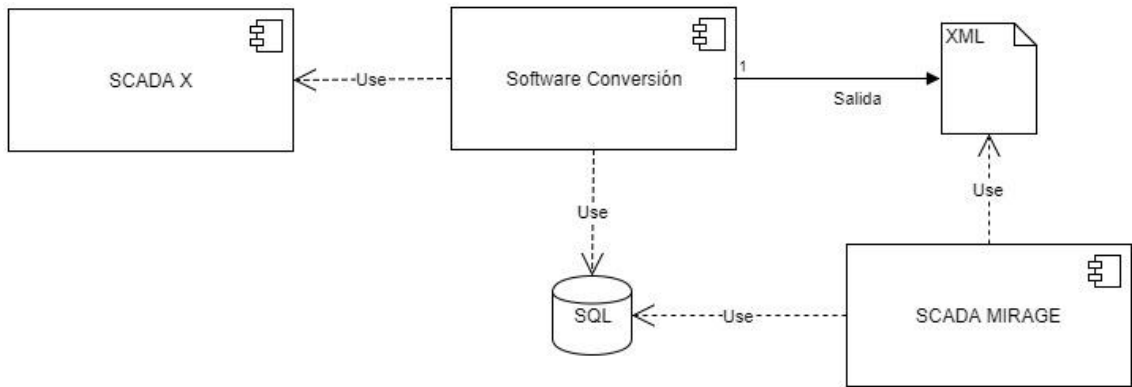


Figura 4.5: Diagrama de componentes y conectores

Catálogo de elementos

Elemento	Responsabilidad
SCADA X	Actual sistema SCADA, recibe señales de campo (estaciones) y actúa en elementos de corte. Internamente usa una base de datos Oracle y un archivo propietario. Corre bajo plataforma Linux.
Software Conversión	Pieza de software que convierte de un SCADA a otro, recibe de entrada los datos del SCADA X, (archivo propietario y base de datos ORACLE), y base de datos SQL y como salida genera archivos XML para cada instalación, que luego serán interpretados por el SCADA Mirage.
SQL	Base de datos con la correspondencia entre las variables de ambos sistemas SCADA.
XML	Archivo que usa el SCADA Mirage, están todos los datos de configuración de cada diagrama. El archivo es producto de la conversión automática.
SCADA Mirage	Futuro sistema SCADA, reemplazará al actual, internamente usa un archivo XML, corre bajo plataforma Windows. Los principales beneficios fueron descritos en capítulo 2.2.3.

Tabla 4-1: Catálogo Elementos C&C

4.2.2 Detalle de componentes Software Migración.

En esta sección se describen aspectos del diseño de la herramienta de conversión y se presentan los principales paquetes, que muestran como estos componentes están divididos en distintas agrupaciones lógicas con las dependencias entre las mismas.

Hay un conjunto de principios de diseño que estuvieron presentes en todos los componentes para mejorar la calidad del código y favorecer los atributos de calidad, testeabilidad, modificabilidad y usabilidad. Estos son [8]:

- ✚ Principio de responsabilidad única: Toda información gestionada por una clase es coherente y relacionada a su responsabilidad.
- ✚ Principio de bajo acoplamiento: Para favorecer la testeabilidad y la modificabilidad cada elemento fue representado por una clase, de tal forma que se pudieran testear independientes y si se produce una modificación no tenga repercusión en el resto de las clases o, al menos, sea mínima.
- ✚ Principio de alta cohesión: Al dividirse cada elemento del diagrama en una clase, se logró que las clases estén estrechamente relacionadas y que cumplan únicamente su responsabilidad, que es modelar un elemento.
- ✚ Principio de clausura común: Las clases candidatas a cambiar por un mismo motivo se agruparon en un mismo paquete de forma que se minimice el impacto del cambio.
- ✚ Principio de dependencias acíclicas: si se observa en el diagrama de paquetes (figura 4-6) no se presentan referencias acíclicas.
- ✚ Principio de inversión de dependencias: Los módulos que implementan funcionalidad de alto nivel no deben depender de los módulos que implementan la funcionalidad de bajo nivel, ambos deben depender de interfaces bien definidas.
- ✚ Auditoría del sistema: Es importante que se capturen excepciones para determinar que sucedió o para su posterior análisis.

A continuación se presenta la estructura interna de la aplicación:

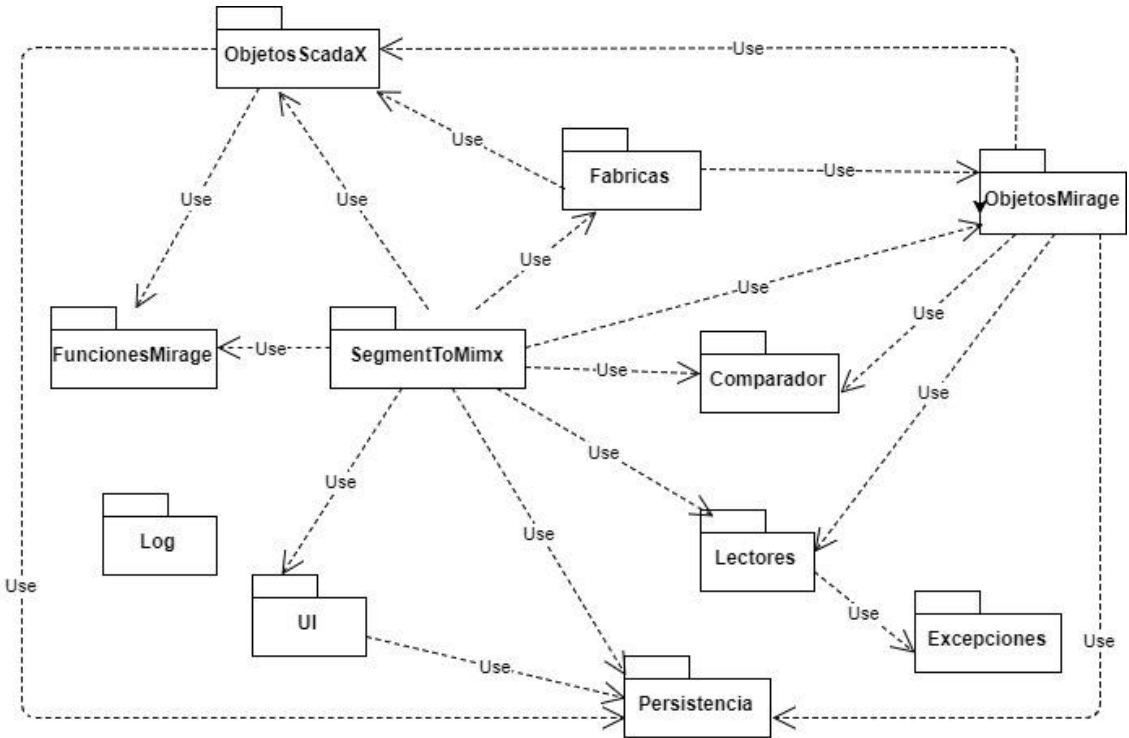


Figura 4.6: Diagrama de detalles de C&C

Catálogo de elementos:

Elemento	Responsabilidades
ObjetosScadaX	Contienen las clases que representan los objetos del SCADA X.
ObjetosMirage	Contienen las clases que representan los objetos del SCADA Mirage.
Fabrica	Contiene los constructores e interfaz, implementa el patrón fachada.
FuncionesMirage	Funciones auxiliares para correcciones estéticas sobre el mímico Mirage.
SegmentToMimx	Paquete principal, donde está el main.
ManejadorXML	Levantar en memoria los objetos Mirage y tenerlos disponibles durante la ejecución del programa.
Lectores	Están las clases que interpretan los archivos de datos, en el caso del SCADA X y del SCADA Mirage.
Persistencia	Contienen las clases para realizar las conexiones con las base de datos, ORACLE perteneciente a SCADA X y la base de datos SQL, donde se guarda la relación entre las variables de ambos SCADA.
UI	Contiene las ventanas para interactuar con el usuario durante la conversión.
Log	Registro de operaciones y errores del sistema.

Tabla 4-2: Catálogo de elementos de detalles de C&C

4.2.3 Diagramas de Secuencia

En esta sección se presentarán los distintos diagramas de secuencia que se consideran importantes para el entendimiento de la arquitectura.

Diagrama de secuencia sin interacción usuario.

A continuación se muestra un diagrama de secuencia correspondiente a la generación de un unifilar de un punto intermedio (sin interacción del usuario)

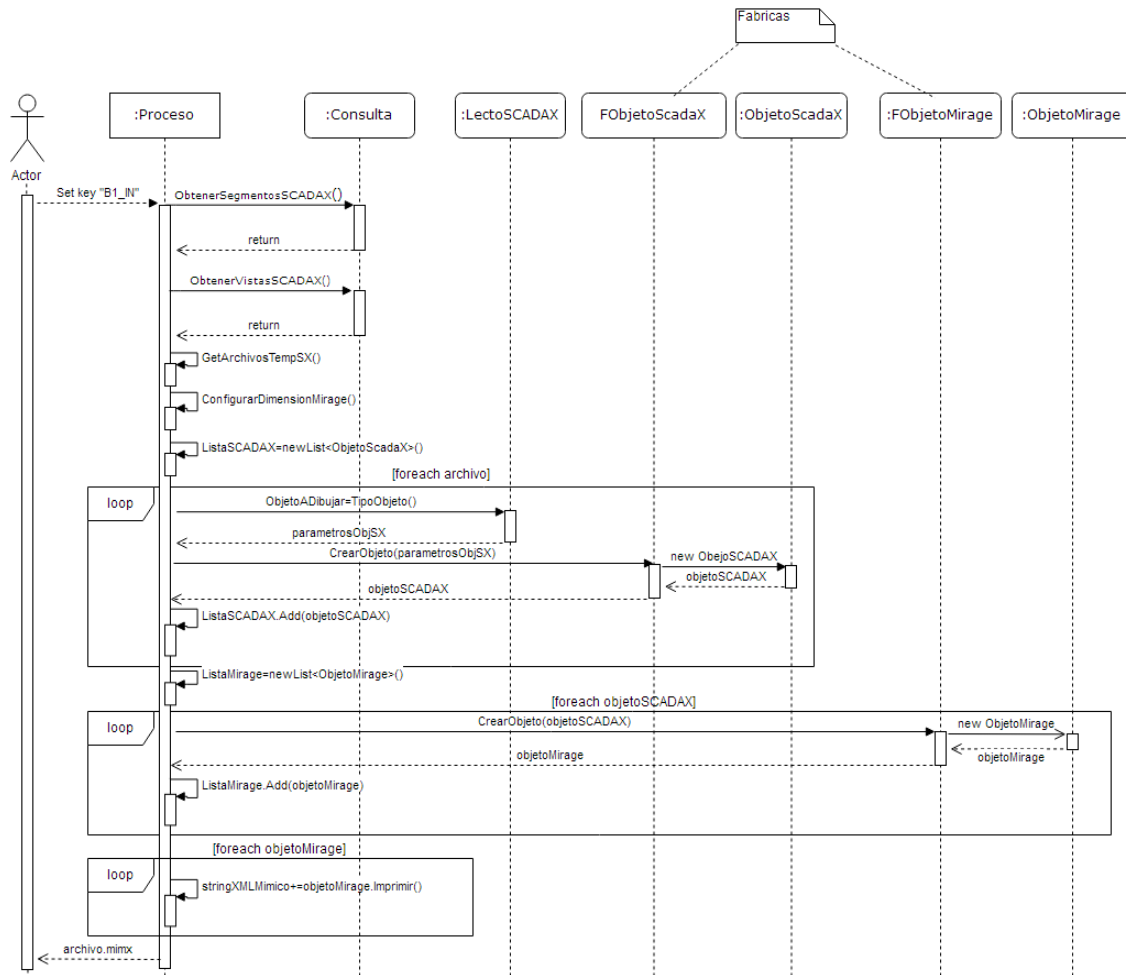


Figura 4.7: Diagrama de Secuencia

El proceso comienza cuando el usuario define a qué instalación del SCADA X le va a generar su mímico.

Luego de seleccionada la instalación, lo primero que hace el proceso de migración es consultar a ORACLE parte de la información gráfica de la instalación a migrar, con el objetivo de redimensionar todos los elementos gráficos y almacenar todas las proporciones para su uso futuro en el diseño del mímico Mirage.

Además, obtiene la ruta de los archivos propietarios en donde está la información de los elementos gráficos de SCADA X como los son las coordenadas y tamaños relativos. Dichos archivos son leídos por el LectorSCADAX, obteniendo el tipo de elemento y sus parámetros, para luego enviárselo a FObjetoScadaX para que creará el objeto respectivo.

Todos esos objetos son guardados en una lista de objetos de SCADA X para posteriormente convertir cada uno en su correspondiente objeto Mirage. Esos objetos también son guardados en una lista para su posterior manejo e impresión en un archivo “.mimx” interpretado por Mirage.

Diagrama de secuencia con interacción usuario.

A continuación se muestra un diagrama de secuencia correspondiente a la generación de un unifilar con interacción del usuario.

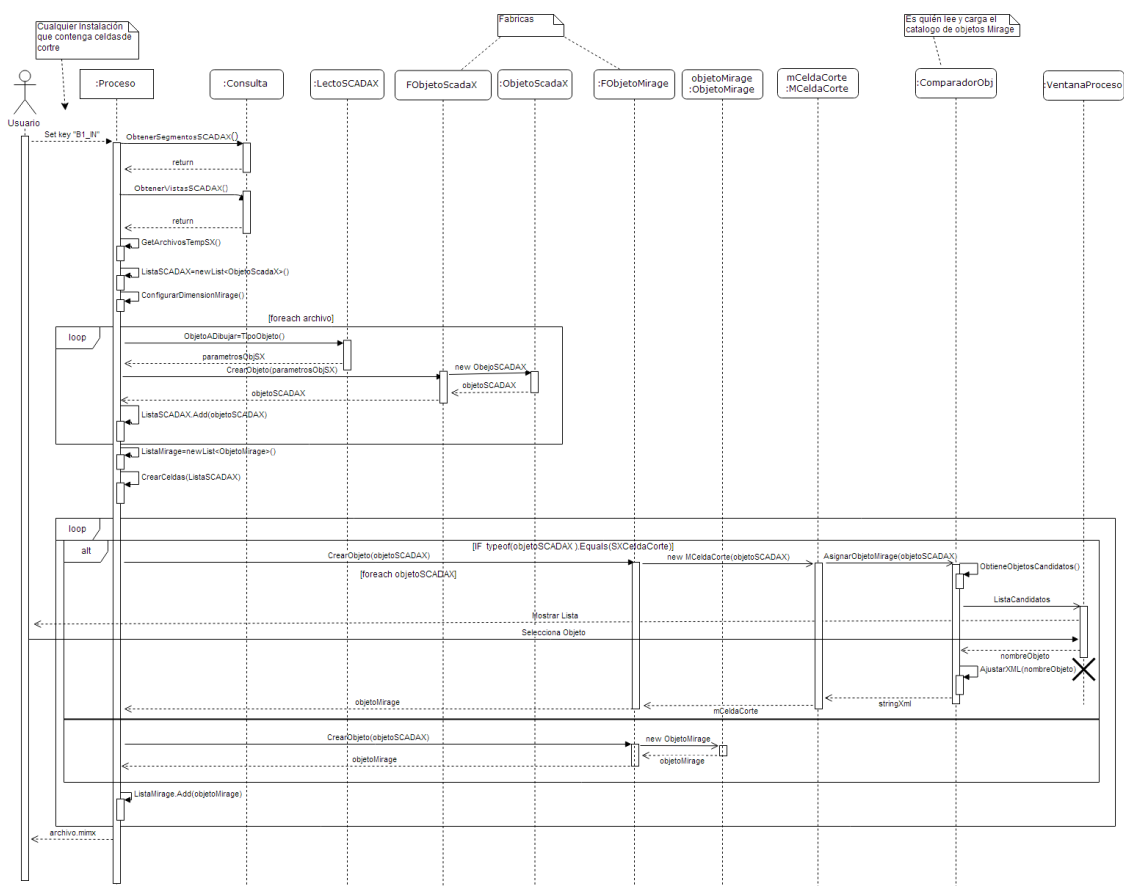


Figura 4.8: Diagrama de secuencia con interacción usuario

En el diagrama se puede apreciar que hasta el primer “loop” comparte la misma secuencia que el diagrama anterior, pero a continuación dentro de “proceso” se crean celdas de corte a partir de la lista de objetos de SCADA X.

Estas celdas son agregadas a la lista de objetos de SCADA X, que posteriormente se recorre para crear su correspondiente objeto Mirage.

Para el caso de objetos que no son celdas, la fábrica de objetos Mirage (FObjetoMirage) se encarga de crear el objeto correspondiente. Pero en el caso en que el objeto a crear

es una celda de corte, la FObjMirage le pasa como parámetro a MceldaCorte un objetoSCADAX que posteriormente será utilizado por el ComparadorObj para, según su forma y orden de sus elementos, inferir qué objetos de biblioteca Mirage son los candidatos a representar dicha celda.

Esa lista es enviada al formulario VentanaProceso para que sea mostrada al Usuario y este seleccione el objeto que funcionalmente se ajuste a la celda.

Una vez seleccionado el objeto, se le envía el nombre del mismo al ComparadorObj para que construya un XML que represente la celda de corte Mirage, con las variables sustituidas y la ubicación relativa en el futuro mímico.

4.2.4 Diagrama de entidades Oracle

A continuación se presenta las entidades de la base de datos de Oracle, que usamos para obtener los datos de los diferentes elementos para la migración:

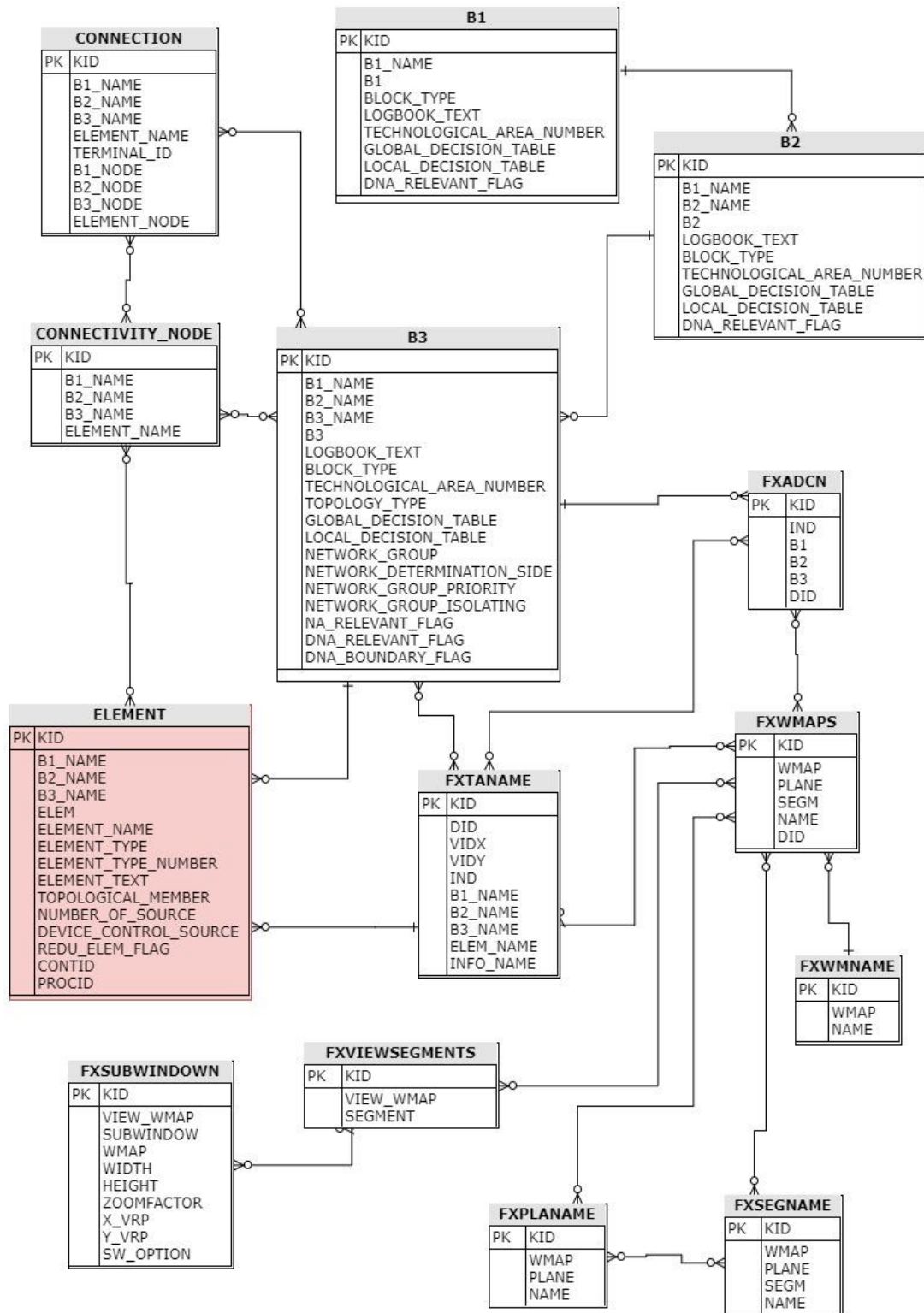


Figura 4.9: Diagrama de entidades Oracle

Catálogo de elementos:

Elementos	Responsabilidades
B1	Representa las instalaciones eléctricas.
B2	Contiene los niveles de tensión correspondiente a las instalaciones eléctricas.
B3	Contiene las celdas de los niveles de tensión correspondientes a las instalaciones eléctricas.
CONNECTION	Contiene los elementos conectados de las instalaciones y a qué nodo pertenecen.
CONNECTIVITY_NODE	Contiene los distintos nodos de conectividad.
ELEMENT	Contiene todos los elementos que se monitorean de una instalación en SCADA X. Esta tabla es consultada por UTE en el desarrollo de la conversión de la estructura de las estaciones, pero no es utilizada para la conversión gráfica directamente sobre Oracle.
FXADCN	Contiene información de en qué mapa están dibujados los B3.
FXPLNAME	Contiene los planos contenidos en los mapas.
FXSGNAME	Contiene el nombre de los segmentos que pertenecen a distintos planos y mapas.
FXSUBWINDOW	Contiene información de las vistas de un mapa, tales como alto, ancho, zoom, etc.
FXTANAME	Contiene la información de en qué segmento se dibuja un determinado elemento de una instalación.
FXVIEWSEGMENTS	Contiene información de a qué segmento pertenece una vista.
FXWMAPS	Contiene los identificadores de los segmentos que contienen los mapas.
FXWMNAME	Contiene el nombre de los mapas.

Tabla 4-3: Catálogo Elementos Entidades

4.2.5 Diagrama de entidades SQL Server

A continuación se presenta las entidades de la base de datos de SQL Server, que usamos para obtener el mapeo de variables SCADA X a SCADA Mirage que realizó UTE en su migración correspondiente, para luego utilizar dichas variables en el mímico.

Además en estas tablas se persisten las excepciones de las pruebas automáticas y las elecciones del usuario durante las conversiones de cada mímico.

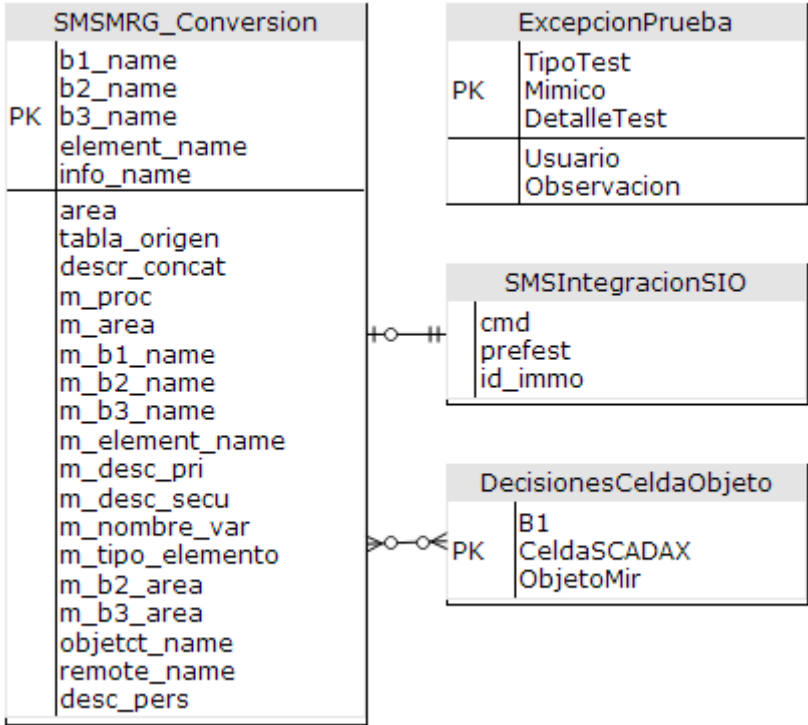


Figura 4.10: Diagrama de entidades SQL Server

Catálogo de elementos:

Elementos	Responsabilidades
SMSMRG_Conversion	Contiene el mapeo de variables SCADA X a Mirage, este mapeo fue realizado por UTE. El software de migración grafica la consulta contantemente para generar los mímicos. Entre otra cosas, se consulta para obtener la variable Mirage que debe ir en cada mímico
SMSIntegracionSIO	Contiene el mapeo entre las instalaciones y el id de integración SIO. Esta tabla fue confeccionada por UTE. El sistema de migración grafica lo consulta para hacer la integración correspondiente a cada mímico que genera
DecisionesCeldaObjeto	Contiene el mapeo entre las celdas de las instalaciones y el objeto Mirage que la representa. Este mapeo se va completando en la medida que el usuario va seleccionando los objetos para cada celda en el software de migración gráfica.
ExcepcionPrueba	Contiene las excepciones identificada por los usuarios del sistema de <i>testing</i> automático.

Tabla 4-4: Catalogo de elementos de entidades de SQL Server

4.2.6 Despliegue del sistema

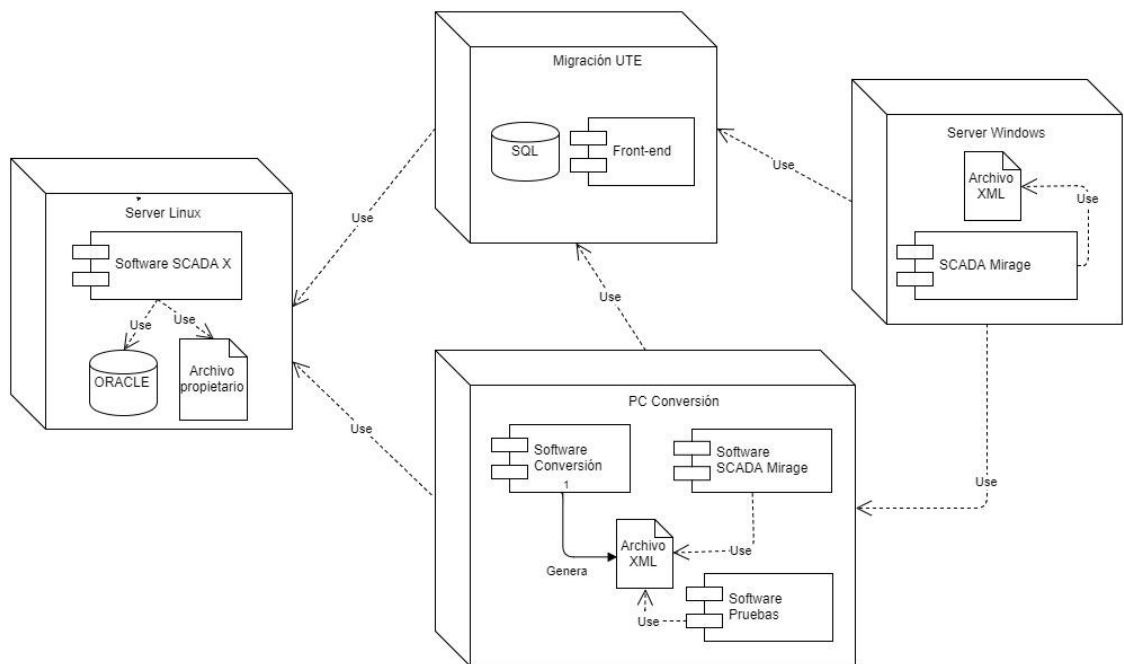


Figura 4.11: Diagrama de Despliegue

Catálogo de elementos:

Elemento	Responsabilidades
SCADA X – Server Linux	Actual servidor con Plataforma LINUX donde corre el SCADA X. Sus principales elementos es el software de visualización gráfica, la base de datos ORACLE y el archivo propietario.
Migración UTE	Migración efectuado por UTE de la base de datos y del <i>front end</i> .
SCADA Mirage – Server Windows	Contiene los constructores e interfaz, implementa el patrón fachada.
PC-Conversión.	Nodo donde se ejecuta la aplicación de conversión y se realizan pruebas.

Tabla 4-5: Catálogo de elementos de Despliegue

4.3 Conclusiones y lecciones aprendidas

La arquitectura de un sistema es trascendental e influye directamente en el resultado de un producto de software. Considerando la naturaleza crítica de nuestro proyecto en cuanto a pruebas, se le dedicó esfuerzo en diseñar de tal manera que permitiera la testeabilidad y que además a la hora de corregir los errores el software fuese de fácil modificación.

Pensamos que se logró una arquitectura muy buena para la pieza de software, considerando que satisface las necesidades planteadas. Uno de los puntos fuertes es que se puede detectar los problemas muy rápidamente para poder corregirlos.

Sin dudas, desarrollar y testear todos los elementos por separado en fases tempranas fue un acierto importante, esto permitió detectar fallas tempranamente, así como también validar decisiones de diseño y a nivel arquitectónico.

Por otra parte, este proceso dejó lecciones aprendidas, como planificar el *testing* de un sistema tan importante. Por tratarse de un sistema de riesgo no se puede dejar nada sin testear antes del despliegue.

5 Gestión de la calidad

El aseguramiento de la calidad es el conjunto de actividades planificadas y sistemáticas aplicadas en un sistema de gestión de la calidad para que los requisitos de calidad de un producto o servicio sean satisfechos [9]. Nos establece un marco de procedimientos y de estándares con el objetivo de lograr un *software* de alta calidad.

Al inicio del proyecto se definen las principales actividades y estándares para la codificación y la documentación.

A continuación se describen los objetivos iniciales, la planificación, la ejecución y la evaluación de todas las actividades relacionadas con el aseguramiento de la calidad, tanto para el producto final como para el proceso, finalizando con el análisis de las métricas obtenidas y las conclusiones a las que permiten arribar.

5.1 Objetivos

Los objetivos establecidos asociados al área de calidad, tanto a nivel del producto como a nivel del proceso, sirvieron como guía al equipo durante el proyecto. Considerando los aspectos relativos a nuestro emprendimiento se consideraron los siguientes objetivos iniciales, los mismos se presentan por orden de importancia:

-  Obtener de la migración de diagramas unifilares funcionales, los más próximos a la versión final, implica que únicamente se tengan que efectuar mínimas correcciones gráficas.
-  Migrar el 100% de los unifilares del CMD Centro y dejar preparado la versión final del software para migrar el CMD de Montevideo. Debido a la gran cantidad de unifilares en Montevideo, el objetivo es aplicar las lecciones aprendidas en la migración de Canelones, para que los detalles estéticos sean mínimos a la hora de la migración y no se necesite ningún otro tipo de corrección.
-  Probar en campo el 15% de los unifilares.
-  Probar el 100% con pruebas automáticas.
-  Cumplir con la totalidad de los requerimientos relevados para el sistema.
-  Obtener un proceso ordenado, controlado, eficaz y eficiente, siguiendo las pautas de la metodología en espiral, con el ciclo de vida incremental.
-  Cumplir con las métricas definidas para el proyecto.
-  Cumplir con estándares de buenas prácticas de programación.

- ✚ Cumplir con los estándares de documentación definidos por la ORT.
- ✚ Adquirir conocimiento y experiencia en las tecnologías seleccionadas para el desarrollo. [10]

5.2 Planificación

Como ya hemos mencionado, el producto a ser obtenido es el resultado de migrar de un sistema SCADA a otro. La pieza de software que permite la automatización de la migración gráfica es solamente una parte de todo el proceso. Por lo tanto, las pruebas se deberían planificar sobre el producto que surge de la integración entre la parte gráfica, del desarrollo de la migración del *front end* y del desarrollo de la migración de la base de datos.

El software recibe como entrada unifilares en sistema SCADA origen y la salida son diagramas diseñados en otro sistema SCADA.

No se debería descuidar la construcción del software, pero optamos por enfocarnos en las pruebas sobre los unifilares, que es lo que definitivamente entraría en producción. De manera que la planificación de la calidad tuvo foco en el proceso completo, siendo el software apenas una parte del mismo.

Esta planificación puede ser vista en el Anexo 6 – Plan de Calidad

5.3 Aseguramiento de la calidad (SQA)

Se pretende asegurar que el producto final cumplirá con los requerimientos relevados y podrá entrar en producción manteniendo con el objetivo mínimo de cumplir con todas las funcionalidades del sistema SCADA actual. Para poder lograr estos objetivos se definió un vasto sistema de *testing*.

Para la planificación de la calidad de la documentación se tomarán en cuenta las políticas, estándares y criterios de aceptación definidos por Universidad ORT, así como también el enunciado del alcance y los objetivos del mismo. Las tareas de aseguramiento de la calidad se realizarán en conjunto con la planificación de la gestión de lanzamiento, gestión de cambio, gestión de las comunicaciones y gestión de riesgos.

En los siguientes apartados se describe en detalle cada actividad realizada para asegurar la calidad del proyecto.

5.3.1 Actividades con foco en el proceso

Se presentarán las actividades de gestión vinculadas directamente al proceso de SQA.

5.3.1.1 Reuniones mensuales

Las actividades relacionadas con la calidad del proceso se basaron fundamentalmente en reuniones mensuales de planificación y seguimiento, en las cuales se revisaron el cumplimiento del cronograma y la gestión de riesgos y alguna otra gestión como puede ser la gestión de cambios y la gestión de lanzamiento según lo ameritaba. Se procuró la mejora continua del proceso eliminando problemas y cumpliendo con las métricas establecidas.

Se ha evaluado que las reuniones mensuales para tratar los temas de SQA han permitido lograr un mejor compromiso del equipo e identificar potenciales riesgos asociados al proceso.

5.3.1.2 Instancias ORT de revisión de todos los procesos

En la instancia obligatoria de revisión propuestas por ORT se pudieron revisar todos los procesos en los cuales se encontraba trabajando el equipo, incluidos los procesos de gestión. Se insistió en la documentación del proceso y en el compromiso a asumir para cumplir con los estándares pretendidos para un trabajo final de una carrera de Ingeniería, de una Universidad tan prestigiosa como la ORT.

5.3.2 Actividades con foco en el producto

Son actividades que se realizaron en diferentes fases del proceso de ingeniería, primero se describen las actividades para detectar desvíos, y luego las actividades realizadas para asegurar las correcciones de los mismos.

5.3.2.1 Estándares de producto

La definición de los estándares nos permite obtener una línea base para determinar si se logró el nivel de calidad deseado.

a) Estándares de documentación

La Universidad ORT ha definido un estándar para la presentación de los proyectos finales de carrera de Ingeniería. Por lo tanto, el trabajo se basó en la siguiente documentación:

- “documento 302-FI” – Normas específicas para la presentación de trabajos finales de carrera (TFDC) de la Facultad de Ingeniería (excepto la Licenciatura en Biotecnología).
- “documento 303-FI” – Hoja de verificación pautas de presentación de trabajos finales de carreras (excepto la Licenciatura en Biotecnología).
- “documento 306” – Orientación para títulos, resúmenes o abstracts e informes de corrección de trabajos finales de carrera.

Los documentos se encuentran disponibles en el sitio de Aulas de la Universidad ORT Uruguay [10].

b) Estándares de codificación

Con respecto a la codificación se siguieron convenciones de código de C# (Guía de Programación de C#) para así favorecer buenas prácticas de programación que brinden una buena legibilidad del código fuente y mejor mantenibilidad del mismo [11].

Se adoptaron normas que posibilitaron generar código de manera uniforme, por medio de la utilización de variables nemotécnicas. Se usaron comentarios únicamente donde fue necesario y en español, ya que nuestro producto de software es una pieza más del proceso, y la vida útil del mismo es hasta finalizar la conversión del CMD Montevideo. Por esta razón el uso de los comentarios se restringe a los casos de difícil comprensión, con esto se busca mantener el código “limpio”.

c) Estándares de requerimientos

En nuestro proyecto los requerimientos son estables y fueron definidos al inicio del mismo.

Pero se consideraron los siguientes puntos para evaluar si los mismos cumplían con el estándar requerido:

- ✚ ¿El requerimiento se puede implementar en el nuevo sistema SCADA?
- ✚ ¿El requerimiento está correctamente detallado?
- ✚ ¿El nivel de abstracción es el adecuado?
- ✚ ¿Está describiendo un requerimiento único o se podría sub dividir en más de un requerimiento?
- ✚ ¿Cada requerimiento está alineado con los objetivos del proceso y del producto?
- ✚ Asignarles prioridades, para determinar si el requerimiento es realmente necesario y para priorizar a la hora de implementarlos.
- ✚ ¿Cada requerimiento está acotado y no es ambiguo?
- ✚ ¿Hay requerimientos en conflicto con otros?
- ✚ Luego de implementado cada requerimiento, ¿el requerimiento es probado y validado en las pruebas establecidas?

A la hora de implementar los requerimientos se verificó que se hizo correctamente y todas las veces se verificaron todos los puntos anteriores.

d) Diseño arquitectónico

La arquitectura de la pieza de software se definió y se documentó luego de la definición de los atributos de calidad (requerimientos no funcionales). En dicha arquitectura se basará el desarrollo de las funcionalidades (requerimientos funcionales).

Dicha definición y documentación se detallan en el capítulo 4) Arquitectura.

5.3.3 Estándares de proceso

a) Gestión de lanzamiento y gestión del cambio.

Ambas gestiones están estrictamente relacionadas y nos basamos en ITIL para gestionarlas.

Par el lanzamiento de versiones se detalla el plan ITIL en el capítulo 8) Gestión de Lanzamientos y en Anexo 1 – Políticas de Lanzamiento, donde están descriptas las políticas de lanzamiento.

Para la gestión del cambio, los detalles del Marco ITIL están descriptos en el capítulo 9) Gestión del cambio.

b) Gestión del riesgo

Para la gestión del riesgo por sugerencia de ITIL se utilizó el marco M_o_R de OGC. Esta detallado en el capítulo 7.4) Gestión del riesgo. Se realizaron revisiones periódicas cada dos meses y al final de cada iteración, para analizar la situación de cada riesgo detectado o incluir nuevos riesgos en el caso que fueran detectados.

c) Aprobación de plan de proyecto

Una vez definido el alcance del proyecto y el cronograma de actividades, incluyendo el sistema de *testing*, fue aprobado por UTE.

Para el sistema de *testing* en campo se tuvieron que realizar coordinaciones entre varios grupos dentro de UTE, Hardware, Software, Protecciones y CMD.

5.3.4 Revisiones

5.3.4.1 Revisiones con expertos y de equipo

Se eligió un ciclo de vida incremental, debido a que los requerimientos eran estables y fueron definidos al comienzo. Por lo que al final de cada etapa se realizaron validaciones con expertos en parte técnica de UTE, con el objetivo de recibir un *feedback* que posibilitara objetivos de mejoras desde otro punto de vista. En dichas revisiones se logró conocer oportunidades de mejora y recibir consejos a tener en cuenta en diferentes áreas y complejidades de la tecnología. Las tareas descriptas se pueden ver en Anexo 5.2– Revisiones formales.

Además de las validaciones formales, se aprovechó la disponibilidad y los recursos técnicos del cliente, ya que Andrés trabaja en dicho sector, para realizar muchas

consultas informales que no se documentaron totalmente, o que consistieron en documentaciones informales (Whatsapp). En dichas consultas informales se discutió con expertos en la Arquitectura y con técnicos expertos en los sistemas SCADA; en resumen, se tuvo alcance a muchos expertos en el dominio del problema, que fueron utilizados para consultas en fase de investigación de las herramientas tecnológicas, durante el desarrollo, pero sobre todo en el sistema minucioso de *testing*.

En definitiva, cada una de estas reuniones ayudó mucho para etapas posteriores del proyecto, y se pueden ver registradas en el Anexo 8 – Revisiones informales con técnicos UTE.

Se realizaron semanalmente reuniones entre los integrantes del equipo en las cuales se llevaron a cabo revisiones informales del avance del proyecto. Este tipo de revisiones fueron muy útiles, lograron mayor comunicación y abrieron la posibilidad de discutir las mejoras posibles.

Si bien no existe un registro formal de todas esas instancias, se presentan evidencias de revisiones informales vía correo electrónico y whatsapp en Anexo 9 – Revisiones informales entre equipo.

5.3.4.2 Revisiones de código

Como lo hemos mencionado, nuestro software es apenas una herramienta en el proceso de la migración, luego de esta quedará obsoleto. Si bien se siguió un estándar de codificación, pero no se realizaron revisiones del mismo, se optó por dedicarle más tiempo al *testing* de los unifilares.

5.3.5 Validaciones

Por la metodología empleada y ya comentada, se tuvo la oportunidad de realizar validaciones con el cliente al final de cada etapa.

Entre ellas se pueden mencionar las primeras validaciones de la especificación de los requerimientos y entre las más importantes se destacan las que generaron mayor valor, que fue la validación de cómo se haría el sistema de *testing*. El cliente estuvo de acuerdo con el mismo, como se detalla en minuta de reunión en Anexo 10 – Aprobación de sistema de *Testing*.

5.3.6 Pruebas

El objetivo de las pruebas es validar que las funcionalidades que componen la solución construida satisfacen con los requerimientos especificados. Por otro lado, se espera que una prueba exitosa permita descubrir errores introducidos en el proceso de desarrollo.

Durante la construcción del producto se realizaron diferentes tipos de pruebas sobre los diferentes módulos que componen la solución.

Para comprobar la funcionalidad del producto se definieron pruebas en diferentes niveles: unitarias e integración elementos, integración módulos, pruebas automática, pruebas de enlace y pruebas de campo.

5.3.6.1 Pruebas unitarias

Cada vez que se diseñaba la conversión de un elemento del unifilar se verificaba que el código XML de salida se levantara en el nuevo sistema SCADA, con esto se fueron probando todos los elementos por separado y que la conversión fuera correcta y en los casos en las que no, que devolviera un respectivo mensaje de error.

No se realizaron pruebas unitarias de forma automática utilizando algún *framework* existente, consideramos como lo más adecuado testear el código XML generado en el diseñador SCADA.

El detalle de las pruebas unitarias se puede ver en Anexo 11 – Pruebas Unitarias

5.3.6.2 Pruebas de integración

Dentro de las prueba de integración tenemos dos tipos de pruebas diferentes.

- a) Pruebas de integración de todos los elementos para formar un único diagrama unifilar. Al igual que las pruebas unitarias, a través del XML generado se levantó desde el Sistema SCADA Mirage y se comprobó manualmente que todos los elementos del sistema SCADA viejo fueran traducidos al nuevo sistema. Además, se hacen retoques mínimos en los unifilares. El tiempo insumido en cada prueba y/o revisión depende del tipo de las instalaciones, el tiempo medio estimado se detalla a continuación.

Tipos Instalaciones	Cantidades	Duración prueba unitaria de revisión	Tiempo total conversión (hs)
Estaciones Complejas	10	240 min	40
Estaciones Regulares	60	180 min	180
Sub Estación	19	30 min	9,5
Puestos de Conexión	19	30 min	9,5
Puntos Intermedios	131	15 min	32,8
Totales	239		271,8

Tabla 5-1: Duración en horas para conversión automático.

Si bien no son tiempos bajos, los expertos estiman que en promedio diseñar manualmente dichas instalaciones insumirían los siguientes tiempos:

Tipos Instalaciones	Cantidades	Duración Diseño manual (hs)	Tiempo total manual (hs)
Estaciones Complejas	10	40	400
Estaciones Regulares	60	12	720
Sub Estación	19	4	76
Puestos de Conexión	19	4	76
Puntos Intermedios	131	1	131
Totales	239		1403

Tabla 5-2: Duración en horas para conversión manual.

Nuestro proyecto nos insumió 2122 horas, en el capítulo 7.3 Gestión del Tiempo se mostrará dicho valor, por lo que no justificaría todo el trabajo, si únicamente se considerara el tiempo como variable. Como veremos más adelante, de manera manual no se puede migrar de un sistema a otro, porque es imposible hacer pruebas en campo del 100% de las instalaciones. Además este software será utilizado para la migración en Montevideo, por lo tanto, únicamente con el tiempo ganado en Montevideo se justifica las horas invertidas en la creación de la pieza de software de migración.

Ver los detalles de las pruebas en Anexo 12 – Pruebas de Integración en unifilar.

b) Pruebas de integración entre el sistema gráfico, el *front end* y la base de datos.

Estas pruebas se realizaron ni bien se tuvo la primera conversión de un unifilar y los proyectos migración *front end* y migración base de datos estaban prontos.

La prueba fue un éxito. Un detalle de la misma se puede ver en Anexo 13.28 – Pruebas de Integración de los tres sistemas.

5.3.6.3 Pruebas cruzadas

Cada desarrollador probó tanto los elementos desarrolladas por él mismo como los desarrollados por el otro integrante del equipo, para asegurar que las pruebas no estuvieran condicionadas por el hecho de haber implementado esa funcionalidad.

Esto se llevó a cabo tanto para todas las conversiones de elementos y, además, para verificar que los unifilares luego de los retoques estéticos no tuvieran errores visibles.

5.3.6.4 Pruebas Automáticas

Se construirá una pieza de software, que reciba como entrada el archivo propietario del SCADA X y el XML correspondiente de Mirage. Interpretará los dos archivos y de forma automática irá probando los siguientes puntos:

-  Verificación de coordenadas: Se deberá verificar que una celda se encuentre en coordenadas equivalentes en los dos sistemas y que ambos extremos estén conectados a elementos equivalentes de ambos diagramas. Con esto se evita referencias cruzadas, es decir que cada diseño de celda represente a su celda física correspondiente.
-  Verificar botones de comunicación: Comprobar que los botones de comunicación del protocolo 101 estén bien asociados en el mímico de la instalación correspondiente.
-  Verificar variables: Se deberá identificar que todas las variables correspondan efectivamente a la instalación donde se encuentra diseñada, para evitar que se pueda accionar otra instalación.

- ✚ Verificar instancias mímicas: Comprobar que exista al menos algún botón que instancie a dicho mímico. Existe un mapa de distritos en donde cada zona tiene un enlace a las instalaciones que le corresponden. Cada vez que se dispara una alarma, en dicho mapa se indica con un botón, parpadeando, facilitando la usabilidad del operario. Lo que queremos evitar con esta prueba es que alguna instalación quede en operación, pero no exista un botón para que se pueda acceder de forma rápida, en casos de alarmas.
- ✚ Verificar conectividad: Verificar que cada elemento conectado en SCADA X también lo esté en Mirage y que, además, esté conectado a los mismos extremos.

Estas pruebas se realizan sobre el 100% de las instalaciones. Ver en Anexo 13 – Pruebas automáticas

5.3.6.5 Pruebas en campo

Para el cliente estas pruebas son las más costosas. Las pruebas de apertura y cierre siempre se deben hacer sin energía. Existen dos formas de maniobras para realizar estas pruebas:

- Dejando a clientes sin energía; se eligen zonas donde el corte afecta a menos usuarios y/o los clientes afectados son residenciales.
- Anillando la estación y para el cliente es transparente, le llega corriente de otro punto y no percibe maniobra alguna. Pero no siempre se pueden realizar estos anillados, existen consideraciones técnicas que no lo permiten en algunos casos y en otros casos la potencia no es suficiente para cubrir a todos los clientes afectados.

Cada maniobra se debe coordinar con 4 grupos:

- Software
- Hardware
- Protecciones
- CMD Canelones

Las condiciones climáticas tienen que ser las adecuadas para las pruebas.

Por lo tanto nos quedan un número limitado de estaciones para probar (entre un 5% y un 10% es nuestro objetivo).

La dificultad de probar en campo la totalidad de las instalaciones hace que solo se pueda migrar mediante un software de manera automática, así se puede evitar el error humano en los diseños manuales.

- ✚ Una vez superados estos requisitos previos lo que se probará es lo siguiente:

- ✚ Se realizan aperturas y cierres de elementos comandables y el personal en sitio verifica que el elemento maniobrado es realmente el que se comandó.
- ✚ Se verifican estados de elementos y alarmas, en el plan de pruebas van a estar especificadas las alarmas disponibles. Por ende, si faltaran alarmas por dibujar se marcarán como "no dibujada".
- ✚ Se verifican valores de medida. En caso de que la medida no tenga el valor correcto, se modifica su curva asociada a la misma en la configuración de Mirage.

Ver en Anexo 14 – Pruebas de campo

5.3.6.6 Pruebas de enlace:

Consiste en realizar una captura de pantalla con las medidas en el SCADA X, bajar el enlace de datos, luego subir el enlace en Mirage y luego corroborar que las medidas actuales, sean del orden de magnitud de las anteriores.

Con estas medidas estamos probando que funciona correctamente la comunicación con la estación y que además la migración del *front end* y la base de datos también funcionan correctamente.

Estas pruebas se realizan sobre el 100% de las instalaciones.

Ver en Anexo 15 – Pruebas de enlace.

5.4 Métricas

Una métrica es una medida cuantitativa del nivel en el que un sistema, componente o proceso posee un atributo dado [12]. Por lo tanto las métricas nos sirven para evaluar atributos del producto o del proceso, para asegurar y controlar la calidad mediante el seguimiento de las mismas.

Se recolectaron métricas referentes al producto y al proyecto.

5.4.1 Métricas del producto:

Únicamente mediremos el índice de defectos y la cobertura de las pruebas, no nos interesa medir *performance* ni disponibilidad, ni otras métricas referidas a otro atributo de calidad diferente a testeabilidad.

Índice de defectos: Durante la ejecución de las diferentes pruebas del producto, se detectarán los siguientes índices según la prueba realizada:

- ✚ Pruebas de enlace: En dichas pruebas se registrarán cantidades de bugs encontrados por cada iteración. Entre iteraciones se corregirá el software.

- ✚ Pruebas en campo: En este caso los defectos encontrados son muy importantes, hay que analizarlos y ver si los mismos se pueden replicar en los demás unifilares, ya que la cobertura de estas pruebas solamente será de un 10% máximo.
- ✚ Pruebas automáticas: El software probará realizará 6 tipos de comprobaciones, ya descritas en capítulo “5.3.6.4. Pruebas Automáticas”. El mismo software enumerará la cantidad de bugs encontrados, los cuales se irán corrigiendo en las diferentes pasadas.

Cobertura de las pruebas: Se medirá la cobertura para las diferentes pruebas establecidas para el producto.

- ✚ Pruebas de enlace: Se realizará en el 100% de las migraciones, corrigiendo bugs y volviendo a realizarlas.
- ✚ Pruebas automáticas: Se realizará en el 100% de las migraciones, corrigiendo bugs y volviendo a realizarlas todas las veces que fuere necesario, una vez desarrollado el software serán las pruebas más sencillas, pero a la vez las más potentes.
- ✚ Pruebas en campo: Tendremos una cobertura de máximo un 10% de las instalaciones totales. Son muy costosas en todo sentido, como ya lo hemos explicado.

5.4.2 Métricas del proyecto:

Cumplimiento del cronograma: Mensualmente se medirá cómo estamos comparado al cronograma planificado. Si hay una desviación mayor al 25% se tomarán medidas para corregirlo.

Esfuerzo planificado vs esfuerzo realizado: Si este indicador toma un valor mayor a 25% se deberán efectuar medidas para corregirlo, dedicarle más tiempo, reprogramar el cronograma o priorizar tareas.

5.5 Análisis de las métricas

5.5.1 Análisis de métricas del producto

Índice de defectos

- a. Pruebas de enlace: Consisten en comparar las medidas del sistema actual con las medidas en el nuevo sistema.

En estas pruebas se detectaron dos tipos de problemas y ninguno es referente a la migración gráfica, pero se deben corregir de todas formas.

El primer problema es debido a una configuración en el protocolo de comunicación 101, dicho error es referente al *front end*. Se corrigen configuraciones y el problema se soluciona.

El otro problema es la relación de la curva de medidas, por ejemplo existen relaciones de 100 a 1. En pantalla vemos el valor 1500, cuando en realidad el valor correcto es 15. También se resuelve mediante configuraciones.

Tipos Instalaciones	Cantidades	Protocolo		Curva	
		Cantidad	%	Cantidad	%
Estaciones Complejas	10	2	20%	1	10%
Estaciones Regulares	60	3	5%	1	2%
Sub Estación	19	1	5%	0	0%
Puestos de Conexión	19	2	11%	2	11%
Puntos Intermedios	131	7	5%	1	1%
Totales	239	15	6%	5	2%

Tabla 5-3: Errores detectados en pruebas de enlace

- b. Pruebas en campo: Se han realizado hasta el momento 10 pruebas del total de las 15 que UTE nos habilito para programar.

Se encontraron errores de configuraciones en los comandos; SCADA X internamente los resolvía de otra forma, por lo que a la hora de configurar en campo estas variables no se tenían en cuenta, o directamente el usuario que las programaba no seguía un protocolo determinado. Pero en SCADA Mirage se deben respetar estas configuraciones, porque internamente el objeto es diferente.

Dichos problemas no permiten que se comande a distancia, pero no existe la posibilidad de accionar otro comando debido a esta configuración, simplemente el comando remoto no actúa.

En el 100% de los casos la información que le llega a Mirage del estado del comando (abierto o cerrado) es correcto. Y al accionarlos manualmente el interruptor cambia el estado en pantalla de abierto a cerrado o viceversa de cerrado a abierto.

Conclusiones importantes de estas pruebas:

- ✚ Los problemas de configuración van a existir y se deberá corregirlos. Pero son problemas detectables a la hora del cambio.
 - ✚ Las variables están **bien** referenciadas en el 100% de los casos. Que es el objetivo principal de la prueba.
 - ✚ Para los que tienen problema, la configuración actúa correctamente en campo una vez corregida, realizando sus aperturas o cierres correspondientes.
- c. Pruebas automáticas: El software de prueba reporta los errores directamente, además levanta la totalidad de unifilares de única vez. Es muy sencillo de usar, por lo tanto se realizan la cantidad que se requieran hasta que los errores reportados sean "0", o que se traten de falsos positivos, pero identificados como tales.

A medida que se hicieron las corridas se fueron reportando la siguiente cantidad de errores. Ver Anexo 16 – Ejemplo de tabla de errores detectados en corridas.

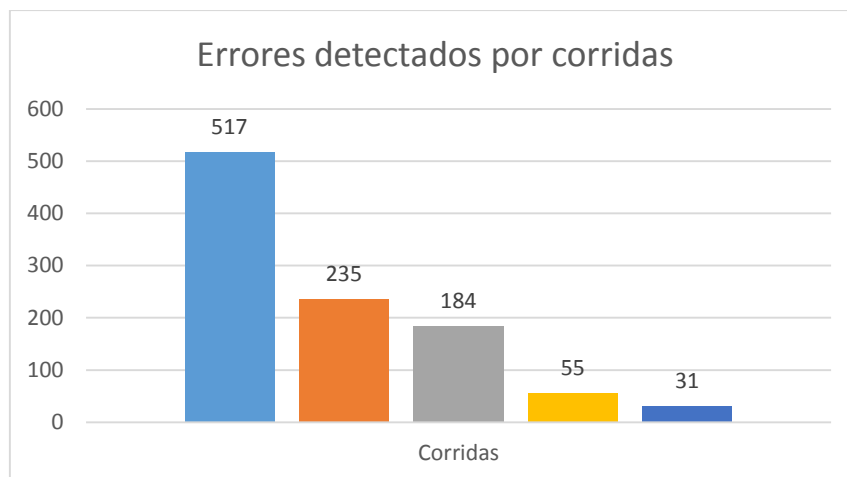


Figura 5.1: Errores detectados por ejecuciones.

Muchos errores fueron falsos positivos, que se fueron exceptuando o corrigiendo los mímicos correspondientes, hasta llegar a cero errores en la última corrida.

5.5.2 Análisis de Métricas del producto

- Cumplimiento del cronograma: Al final de cada etapa se evaluó el tiempo que nos llevó y ver qué desviación existió con respecto a lo planificado.

A continuación se presenta la duración estimada y la real obtenida de cada fase, así como el día de retraso al iniciar cada etapa.

Nombre de tarea	Estimado			Real			Atraso al comenzar las fases
	Duración estimada	Comienzo	Fin	Duración real	Comienzo	Fin	
Fase 0	25 días	29/09/2016	02/11/16	67 días	29/09/2016	05/12/2016	0 días
Fase 1	207 días	03/11/2016	18/08/17	259 días	07/12/2016	23/08/2017	34 días
Iteración 1 - Migrar Elementos	118 días	ju 03/11/2016	lun 17/04/17	165 días	mi 07/12/2016	do 21/05/2017	34 días
Iteración 2 - Migrar Unifilares	17 días	ma 18/04/2017	mié 10/05/17	22 días	lu 22/05/2017	ma 13/06/2017	34 días
Iteración 3 - Pruebas de Enlace	30 días	ju 11/05/2017	mié 21/06/17	48 días	vi 23/06/2017	ju 10/08/2017	43 días
Iteración 4 - Pruebas en Campo	31 días	ju 22/06/2017	jue 03/08/17	62 días	ju 22/06/2017	mi 23/08/2017	0 días
Iteración 5 - Pruebas Automáticas	36 días	ju 11/05/2017	jue 29/06/17	34 días	ju 20/07/2017	mi 23/08/2017	70 días
Iteración 6 - Puesta en Producción	11 días	vi 04/08/2017	vie 18/08/17	-	a definir	a definir	-
Fase 2	14 días	04/08/2017	23/08/17	-	29/07/2016	a definir	-

Tabla 5-4: Duración tareas estimadas vs duración real.

Hubo una estimación errónea de nuestra parte, sobre todo en la duración de esfuerzo del proyecto, lo estimamos en 1520 horas y la duración real hasta el momento es de 2122 horas.

Si bien la mayor desviación estuvo en programación, la fase 0 nos costó mas de lo planificado, causando el retraso del inicio de las demás fases. Sumado a esto el proyecto insumió muchísimas más horas de lo previsto. Para no atrasarnos aún más tuvimos que aumentar considerablemente las horas dedicadas, pero logramos mantener el atraso constante y al final ponernos al día.

Se considera que debería haber sido realizado por tres personas.

- b. Esfuerzo planificado vs esfuerzo realizado: Durante todo el proyecto se evaluaron mes a mes las horas dedicadas, para comparar nuestro grado de compromiso con respecto a lo planificado.

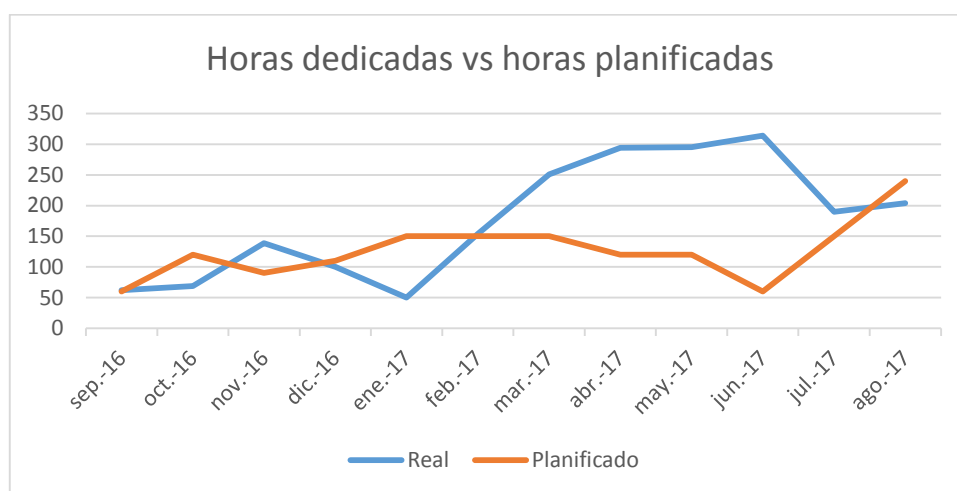


Figura 5.2: Horas dedicados vs horas planificadas

En cuanto a la dedicación, por lo general estuvimos por encima de lo previsto, sobre todo a partir del mes de febrero cuando se tuvo que dedicar muchas horas a la programación y, posteriormente, al *testing*.

Este seguimiento nos sirvió, por ejemplo, en octubre que estuvimos muy por debajo de lo estimado, lo hablamos entre nosotros y se corrigió con un esfuerzo adicional en noviembre.

Otro mes que estuvimos muy por debajo fue en enero, pero principalmente se debió a las licencias, ya que ambos estuvimos sin trabajar en el proyecto los primeros 10 días del mes. Asumimos que ese mes estuvo mal planificado y no se tomaron medidas al respecto.

5.6 Conclusiones y lecciones aprendidas

Como conclusión final se pudo extraer, que el equipo siempre estuvo comprometido para llevar a cabo las actividades que estaban planificadas, no siempre se logró a causa de que hubo momentos que tuvimos que dedicarnos al desarrollo descuidando la documentación.

Se ha logrado el producto esperado, unifilares totalmente funcionales para el nuevo sistema SCADA, testeados y parte ya validados con el cliente. En producción se encuentran unifilares pertenecientes a estaciones de transmisión, dichas estaciones solamente reciben estados y medidas, no se actúa sobre ellas, ya que el mantenimiento no pertenece a Distribución.

Si bien se logró el producto con la calidad deseada, consideramos que dentro de las lecciones aprendidas más importantes está el aprendizaje de la gestión en campo de un proyecto de tal magnitud.

Se tuvieron que realizar muchísimas coordinaciones y reuniones varias, teniendo en cuenta que nuestro producto dependía de que todo funcionara correctamente. Dependía de la migración del front end, de la migración de la base de datos. De que se construyera un centro de cómputos nuevo para alojar el nuevo servicio, debido a que el nuevo corre en plataforma diferente al anterior (en Windows). La construcción de dicho centro de cómputos conlleva un montón de factores más: refrigeración, medidas contra incendios, instalación de la red eléctrica y de la red de comunicación, configuración de la red LAN, importar e instalar servidores, etc.

El trato con un Ente público que maneja otros tiempos, sin dudas, nos aportó mucho en nuestro crecimiento profesional y, también, en el personal.

6 Gestión de la configuración

En esta sección se aborda el proceso de gestión de activos y configuración de servicios (SACM- *service asset and configuration management*). Según ITIL V3 *Service Transition*, pág. 89, ninguna organización puede ser plenamente eficiente o eficaz a menos que gestione bien sus activos, en particular aquellos activos que son vitales para el funcionamiento del negocio del cliente u organización [13].

El propósito del proceso del SACM para ITIL, es asegurar que los activos requeridos para entregar los servicios estén debidamente controlados y que la información exacta y confiable sobre esos activos esté disponible cuando y donde se necesite. Esta información incluye detalles sobre cómo se han configurado los activos y las relaciones entre los mismos.

SACM detallada a continuación que tiene como principal objetivo identificar los elementos de configuración (CI) que durante el ciclo de vida del proyecto estarán bajo su ámbito, y definir los procedimientos y mecanismos que garantice el almacenamiento, la modificación y control de los mismos.

6.1.1 Definiciones

Cada CI debe cumplir con las siguientes definiciones

-  Tener un nombre que lo identifique de manera única.
-  Tener una versión.
-  Tener asignado un estado.
-  Cada vez que se modifica un CI se debe especificar la fecha de dicha modificación y una breve descripción de los cambios afectados.
-  Localización del CI.
-  Cada CI tiene asignado un responsable (*Configuration Item Responsible, CIR*) que asegurar que las definiciones establecidas en el plan de gestión de la configuración se apliquen correctamente.

6.1.2 Identificación de los elementos de configuración

Los CI son todos aquellos activos que son manejados durante el Proyecto y que están sujetos a cambios durante el ciclo de vida del mismo. En la siguiente tabla se especifican los CI encontrados

Elemento	Nombre	Descripción
Código fuente	El nombre de cada archivo corresponde al nombre de las entidades del código.	El nombre de cada archivo corresponde al nombre de las entidades del código.
Documento principal	Proyecto_Migracion_SCADA - Vx.docx	Documentación formal del proyecto.
Gestión de riesgos	Gestión de riesgos - Vx.docx y Riesgos -Vx.xlsx	Registro de la información relativa a la gestión de riesgos y su evolución durante el proyecto.
Gestión de calidad	Gestión de la calidad - Vx.docx	Registro de las definiciones de calidad y control de las mismas.
Plan de comunicación	Plan de comunicación - Vx.docx	Registro de las definiciones relativas a la planificación de la comunicación.
Gestión de Configuración	Gestión de la configuración - Vx.docx	Registro de los elementos de configuración
Gestión de cambios	Gestión de cambios - Vx.docx	Registro de la información relativa a la gestión de los cambios
Gestión de lanzamiento	Gestión de lanzamiento - Vx.docx	Registro de la información relativa a la gestión de lanzamientos.
Cronograma	Cronograma - Vx.mpp	Registro de actividades, estimaciones de esfuerzo y secuencia
Documentos de apoyo	Documentación de ITIL, Documentos de ORT, manuales de SCADA.	No están sujetos a modificaciones.

Tabla 6-1: Elementos de configuración

Nota: Vx identifica la versión del documento. Ejemplo. La versión 8 del Documento principal se identifica como Proyecto_Migracion_SCADA – V8.docx

Tabla 6.1 – Elementos de configuración.

6.1.3 Control de versiones

Se le asigna un numero de versión al final del nombre a cada uno de los CI. Se debe incrementar en 1 cada vez que se suba una nueva versión al repositorio.

6.1.4 Organización de documentos

Para los documentos editables en Word, se utilizó el OneDrive para su organización, se guardaron en la carpeta Documentación Principal las diferentes versiones del documento principal y una carpeta para los otros archivos de gestión, como se pueden ver en las siguientes Figuras.

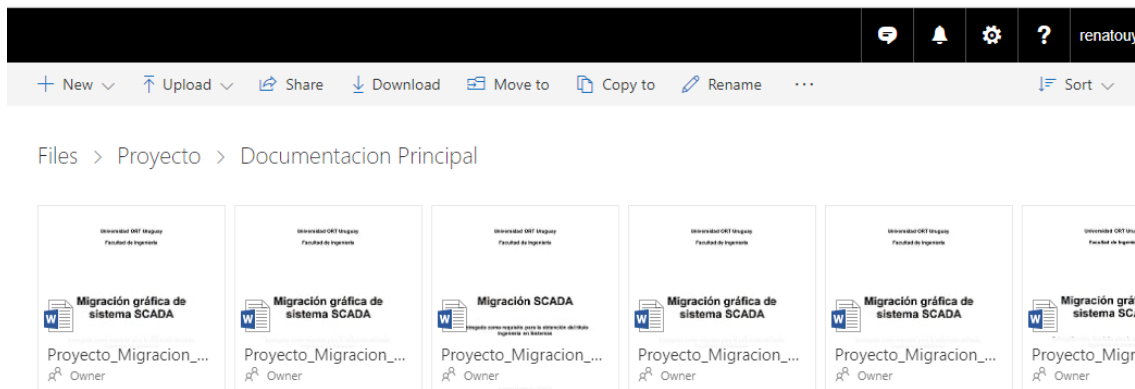


Figura 6.1: Estructura de carpetas con documentación

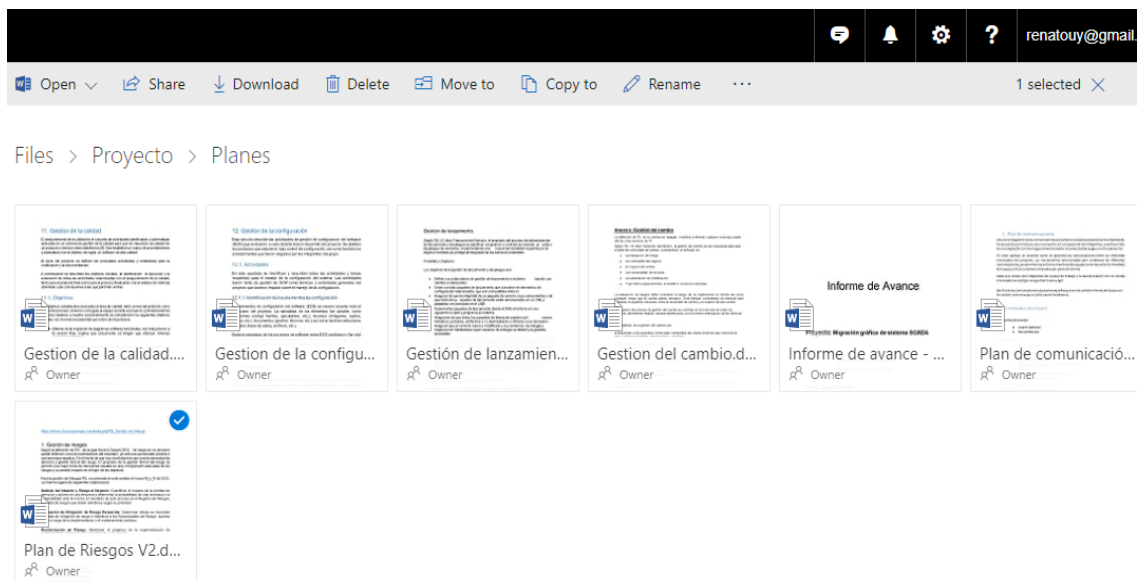


Figura 6.2: Estructura de carpetas con los planes de gestión

6.1.5 Organización del código fuente

El código fuente fue organizado respetando las estructuras predefinidas de cada tecnología utilizada.

Las tecnologías utilizadas en el proyecto sugieren sus estructuras organizacionales del código optimizados para brindar una buena experiencia en configuración del software.

6.2 Herramientas a utilizar

Hoy en día existen muchos repositorios online, los cuales son accedidos desde cualquier dispositivo con acceso a internet y por cualquiera que este habilitado para hacerlo, disponiendo todos de la misma información en todo momento.

Para la documentación se usaron dos repositorios, Google Drive y OneDrive. Mientras que para el código fuente se utilizó el GitHub.

6.2.1 Repositorio para gestión de documentos

En este punto se evaluaron principalmente tres alternativas:

- Dropbox
- Onedrive
- Google drive

Si bien cualquiera de las tres cubre las necesidades básicas como revisión, historial, etc., en un principio optamos por Google Drive, pero ni bien comenzamos a trabajar optamos por OneDrive para los documentos editable en Word. Ya que se pueden sincronizar los documentos utilizando la versión estándar de Microsoft Word, haciendo uso de funciones más avanzadas de edición de texto, que facilita el trabajo en simultáneo.

Para los demás archivos como los cronogramas, archivos de Excel, manuales varios, registros gráficos, materiales de ITIL y de tecnología, etc. continuamos utilizando Google Drive. Trabajando local en cada máquina con sus archivos propietarios. Como somos dos integrantes y estos archivos no eran usados tan asiduamente, no se veía la necesidad de migrarlos a OneDrive.

Dentro del directorio en Google Drive existen varios subdirectorios, como podemos ver en la Figura 12-3

Organizar	NUEVO	ADRI	SE
oyecto Materiales > Documentacion >			
Nombre		Fecha de modifica.	
	Actas y Minutas	28/07/2017 20:41	
	Borradores Proyecto	28/07/2017 22:45	
	Diagramas VISIO	18/06/2017 15:10	
	Documentos ORT (302, 306, 307)	28/07/2017 20:42	
	Gestiones (riesgo, calidad, comunicaciones, release, cambios, etc)	28/07/2017 23:34	
	Imagenes Usadas	30/05/2017 19:48	
	Informe de Avance y propuestas	30/05/2017 19:54	
	ITIL	02/03/2017 15:58	
	ITIL - algunos docs traducidos	28/07/2017 22:19	
	Otros Proyectos	28/07/2017 21:26	

Figura 6.3: Repositorio Google Drive.

6.2.2 Repositorio para gestión de código

Para la elección de una herramienta para gestionar el repositorio de código se compararon los dos modelos de mayor popularidad.

Se compararon dos modelos de repositorios para el código fuente: centralizados (principalmente SVN) y distribuidos (principalmente GIT).

Por mayor conocimiento se decidió por GIT, además porque se puede trabajar sin conexión y comparando las velocidades de ambas en GIT se obtuvo mejor respuesta.

Existen muchos servicios dedicados a ofrecer repositorios GIT, pero optamos por BitBucket, porque se puede mantener el código en privado, mientras que en GitHub no. En nuestro caso es muy importante que el código permanezca privado.

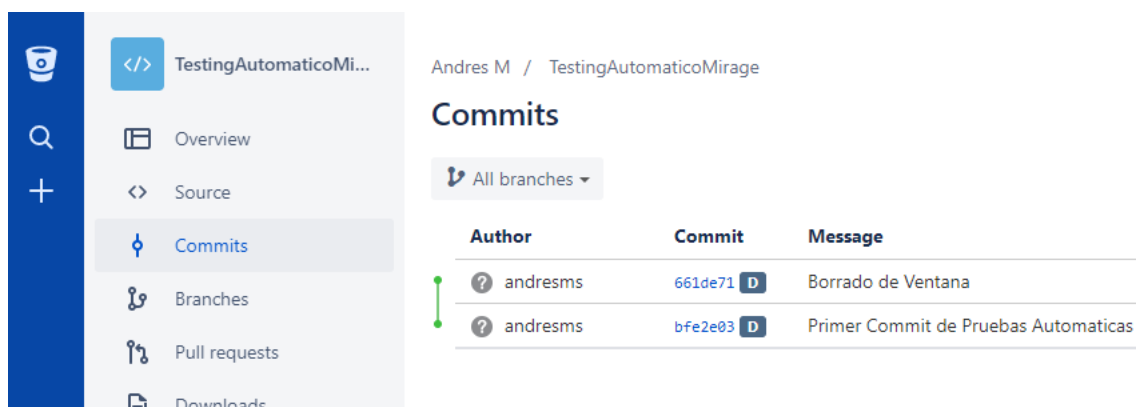


Figura 6.4: Repositorio BitBucket.

Para el manejo de los archivos de Mirage el mismo software tiene su repositorio.

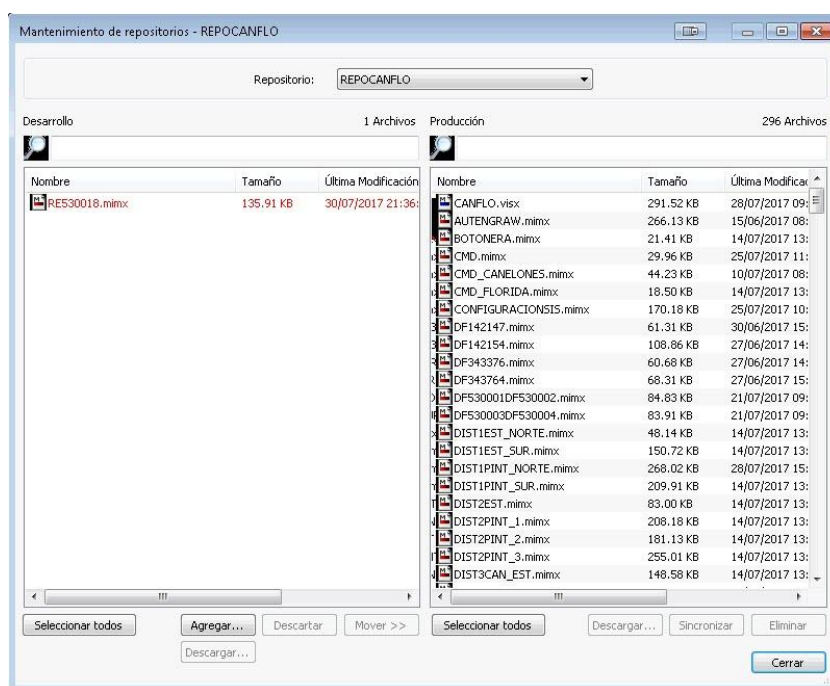


Figura 6.5: Repositorio Mirage.

6.3 Seguimientos

Nombre del CI	Proyecto_Migracion_SCADA - Vx.docx	Gestion de Riesgos - Vx.docx
CIR asignado	Andrés Martínez, Renato Moraes	Renato Moraes
Fecha de Creación	21/09/2016	15/10/2016
Última modificación	01/08/2017	01/08/2017
Versiones creadas	22	5
Versión final	Si	Si
Ubicación del repositorio	Proyecto/Documentación Principal	Proyecto/Planes

Nombre del CI	Gestión de la calidad - Vx.docx	Plan de comunicación - Vx.docx
CIR asignado	Andrés Martínez, Renato Moraes	Renato Moraes
Fecha de Creación	12/12/2016	10/11/2016
Última modificación	01/08/2017	10/11/2016
Versiones creadas	4	1
Versión final	Si	Si
Ubicación del repositorio	OneDrive/Proyecto/Planes	OneDrive/Proyecto/Planes

Nombre del CI	Gestión de la configuración - Vx.docx	Gestión de lanzamiento - Vx.docx
CIR asignado	Renato Moraes	Renato Moraes
Fecha de Creación	01/05/2017	05/10/2016
Última modificación	01/08/2017	22/05/2017
Versiones creadas	2	3
Versión final	Si	Si
Ubicación del repositorio	OneDrive/Proyecto/Planes	OneDrive/Proyecto/Planes

Nombre del CI	Cronograma - Vx.mpp	Riesgos -Vx.xlsx
CIR asignado	Andrés Martínez, Renato Moraes	Renato Moraes
Fecha de Creación	21/09/2016	15/10/2016
Última modificación	01/08/2017	01/08/2017
Versiones creadas	7	5
Versión final	Si	Si
Ubicación del repositorio	Documentacion/Project - Gantts	Documentacion/Gestion

Tabla 6-2: Seguimientos modificaciones

7 Gestión del Proyecto

La gestión de proyectos es la disciplina del planeamiento, la organización, la motivación, y el control de los recursos con el propósito de alcanzar uno o varios objetivos. Un proyecto es un emprendimiento temporal diseñado para producir un único producto, servicio o resultado, con un principio y un final definidos (normalmente limitado en tiempo, en costos y/o entregables), que es emprendido para alcanzar objetivos únicos y que dará lugar a un cambio positivo o agregará valor.

El éxito de un proyecto se corresponde con la consecución de los objetivos de alcance, plazos, coste y calidad, mediante una gestión integrada de los mismos. [14]

El proyecto se estructura en etapas y actividades con objetivos y actividades claramente diferenciadas; las mismas se detallan en el punto 7.4 Gestión del tiempo que describimos más adelante.

Por lo tanto, en este capítulo describimos las diferentes gestiones del proyecto, las metodologías utilizadas en el proceso de gestión y que actividades y herramientas se utilizaron. Para finalizar, se presenta un análisis de las métricas recolectadas, a modo de reflexión sobre la gestión.

7.1 Metodologías de gestión del proyecto.

7.1.1 Proceso

Lo primero que se hizo fue seleccionar un proceso para la gestión del proyecto y que el mismo se adecuara a las características requeridas para el caso.

A continuación se realiza un estudio del producto a ser obtenido, a efectos de justificar el proceso elegido [15].

Nuestro proyecto se trata de una migración de un producto de misión crítica, que funciona en tiempo real, con fuerte regulaciones, que debe estar siempre disponible y que no puede tener error de ningún tipo. Si el sistema deja de funcionar o funciona inadecuadamente es un gran riesgo para el cliente, por ejemplo en el manejo de aperturas y cierres de elementos comandables el incorrecto funcionamiento puede dejar sin energía eléctrica a muchos clientes, ocasionando pérdidas económicas muy grandes.

Si bien el proyecto si bien es de misión crítica, cuenta con requerimientos claramente definidos, por lo que descartamos el paradigma de construcción de prototipos.

Contamos con conocimiento de las herramientas y del producto, pudiendo identificar claramente los riesgos, por lo que se puede tener en cuenta los paradigmas como Espiral, RUP, en V, Ensamblaje de componentes o Métodos formales.

Otro foco muy importante es el de la calidad, esto implica controles estrictos en los procesos de desarrollo y en la gestión de los mismos. Además, el producto para el nuevo sistema SCADA debe presentarse como una opción segura y efectiva, por lo que la calidad de los mismos es un pilar fundamental. Estas condiciones de calidad exigen realizar un sistema de pruebas robustas tanto en validación como en verificación, por ello descartamos el paradigma AS (Agile Software).

Existen limitaciones de tiempo impuestas por tratarse de un proyecto académico final de carrera, que se debe realizar en un año, con fechas de inicio y fin claramente definidas desde un primer momento, además de que se deben validar con anticipación los diagramas obtenidos.

Para ofrecer al cliente un producto temprano podemos optar por un paradigma en Espiral o RUP.

En este caso, en el proyecto se necesitan ver avances, por lo que se descarta el paradigma Lineal Secuencial, del mismo modo que no se recomienda el paradigma en V cuando se desean ver resultados tempranos.

El cliente es altamente exigente, el producto debe ser de calidad y antes de la puesta en producción del producto final se exigirán evidencias de las pruebas del mismo, tomando gran importancia la documentación de todo el proceso.

Estas restricciones en los aspectos formales son otra causa para dejar de lado las metodologías ágiles.

Por lo ya expuesto anteriormente, debemos optar entre los paradigmas en Espiral y RUP. Finalmente elegimos el primero, ya que además de ser software riesgoso que implica el uso de mucha tecnología permite brindar al cliente el producto temprano. Descartamos RUP porque es un método que puede ser muy pesado, por el grado de complejidad, y además no se adaptaba bien a un proyecto que tiene restricciones de tiempo.

En conclusión el paradigma escogido para el proceso es **Espiral**, ya que se adapta bien a un software de misión crítica, con requerimientos estables y restricciones de tiempo.

7.1.2 Ciclo de Vida

En base en lo descrito anteriormente para la elección del paradigma **Espiral**, se instanciará un ciclo de vida **incremental** [16], como lo hemos expuesto por tratarse de software de misión crítica, que se relevarán los requerimientos en la primera instancia y luego se utilizará la construcción en incrementos para asegurar la calidad en cada una de las fases. Además esto nos brinda más estabilidad desde el punto de vista de la arquitectura.

En el modelo de ciclo de vida seleccionado, se identifican claramente 4 pasos. La descripción del sistema es esencial para especificar y confeccionar los distintos incrementos hasta llegar al producto global y final. Las actividades concurrentes

(especificación, desarrollo y validación) sintetizan el desarrollo pormenorizado de los incrementos, que se harán posteriormente. [17]



Figura 7.1: Diagrama genérico del desarrollo evolutivo incremental

7.1.3 Metodología de proceso de construcción de software

Para la construcción del software, al optarse un por un modelo en Espiral con un ciclo de vida iterativo, se establecieron las siguientes tres fases:

7.1.3.1 Fase 0

En esta fase se construye el cronograma, se desarrollan el ESRE, la definición de la Arquitectura y se diseñan los siguientes planes de gestión: lanzamiento, SQA, riesgos, cambios y SCM. Lanzamientos, cambios y riesgos basados en ITIL.

7.1.3.2 Fase 1

Esta es la fase del desarrollo del producto y la misma se dividió en 6 iteraciones. Cada iteración está enfocada en el producto y no en la pieza de software en sí.

Iteración 1 – Migración elementos: En esta iteración es donde se desarrolló la pieza de software, dentro de esta iteración usamos un ciclo de vida iterativo, se iniciaba el análisis, desarrollo y las pruebas unitarias para cada elemento. Durante esta etapa se realizará una prueba piloto de la integración con las otras dos patas del proyecto, como fueron la migración del *front end* y la migración de la BD.

Iteración 2 – Migración unifilares y ajustes manuales: Una vez que el software fue desarrollado para la totalidad de los elementos, se comenzarán a migrar los unifilares, luego de migrado se realizarán ajustes manuales estéticos y para cada unifilar se realizarán pruebas de integración. Todos los bugs detectados se reportan, para modificar el software y que no los vuelva a repetir.

Iteración 3 – Pruebas de enlaces: Para realizar estas pruebas se deben coordinar con el CMD de Canelones, durante la misma se quedarán con sistema off-line mientras dure la prueba.

Esta iteración consiste en capturar los valores de los parámetros eléctricos del SCADA X. Luego se deberá bajar el enlace de comunicación y apuntarlo para el nuevo sistema SCADA Mirage.

Una vez configurado para que reciba las medidas en el nuevo SCADA, las compararemos con las medidas guardadas del sistema SCADA X.

Son aceptables variaciones mínimas en las mismas, debido a que es un sistema vivo en el que varían los parámetros eléctricos. Si los valores son muy distintos, se vuelve a sistema anterior y se compara si no hubo un cambio brusco en la carga. Estas pruebas se realizarán las veces que sean necesarias por cada instalación.

Con estas medidas se logran probar los protocolos de comunicación y si las configuraciones están correctas.

Iteración 4 – Pruebas en campo: Por varias razones estas son las pruebas más complejas, por lo que se realizarán entre un 5 y 10 % de las instalaciones.

Este tipo de pruebas es la principal razón de construir un software de migración automático. En todo desarrollo manual, sin excepción, se deben realizar estas pruebas, que en el actual sistema en funcionamiento es imposible.

Para realizar las coordinaciones es necesario coordinar con cuatro gerencias de UTE. Con gerencia de Software, gerencia de Hardware, gerencia de Protecciones y CMD Canelones. Entre ellos determinan cuáles subestaciones pueden alimentar desde otras o cuáles pueden dejar sin energía a los clientes. Las posibilidades son reducidas, se estima poder hacer entre un 5 y 10% máximo.

Iteración 5 – Pruebas automáticas: Para estas pruebas se construirá una pieza de software que debe ser probada. Una vez en producción se ingresará cada unifilar, realizando los grupos de pruebas establecidos, que pueden ser: probar que todos los elementos existan, que esté conectado a los mismos extremos, ubicados en coordenadas equivalentes para eliminar referencias cruzadas, continuidad, etc.

Iteración 6 – Puesta en Producción: Al igual que las pruebas de campo se deberán coordinar con todas la gerencias, con el fin de determinar una fecha en que no existan condiciones climáticas adversas, y que sea durante la mañana (que es cuando trabaja la mayor cantidad de operarios). La tarea en sí es fácil, se baja un sistema y se sube el otro, pero sin margen de error.

7.1.3.3 Fase 2

Esta es la fase de cierre de proyecto, se deberá concluir la documentación del proyecto académico y la documentación a entregar al cliente, se analizarán métricas y por último, es el inicio de la migración del CMD de Montevideo.

A continuación se presenta un diagrama con todas las fases y tareas del proyecto.

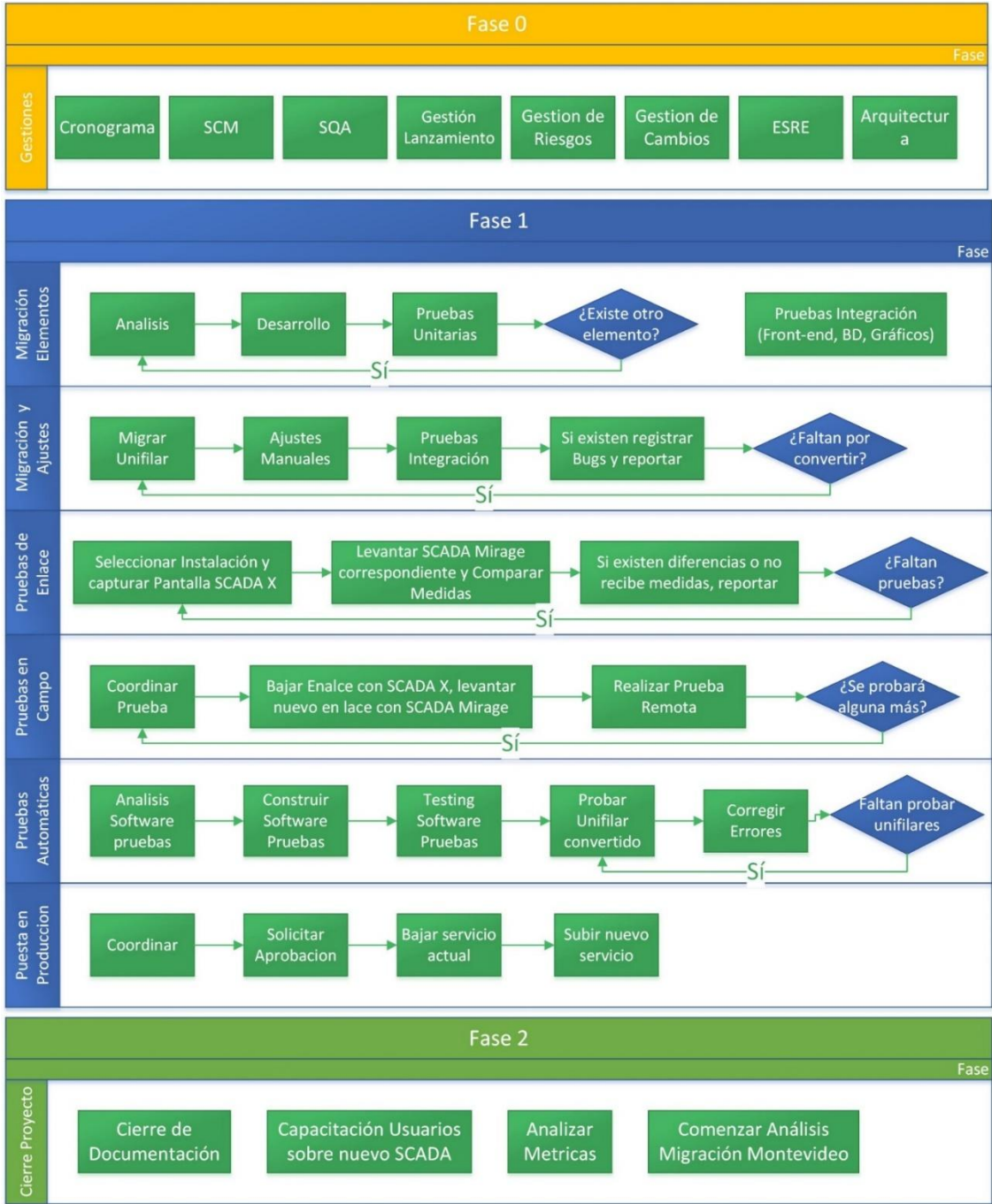


Figura 7.2: Diagrama de Fases y Tareas

7.1.4 Roles

En las metodologías tradicionales se recomiendan ciertos roles a seguir durante todo el proyecto de software. Cada rol está asociado a un área importante del proyecto

A continuación se definen, a grandes rasgos, las tareas y responsabilidades de cada rol:

Rol	Principales tareas
Ingeniero de requerimientos	• Relevar requerimientos. • Especificar requerimientos. • Verificación y validación de requerimientos.
Arquitecto	• Definir y evaluar arquitectura del software Migración y del software para pruebas automáticas. • Integración de la conversión de todos los elementos en un único diagrama por instalación • Guía a los desarrolladores.
Desarrollador	• Analizar funcionalidades. • Diseño de funcionalidades. • Revisión de pares. • Integración de todos los elementos • Realizar pruebas unitarias y de integración de elementos
Gerente	• Elaborar Plan de Gestiones de Proyecto y asegurar que se cumplan los plazos o reajustar cronogramas periódicamente según necesidades. • Comunicar y reportar periódicamente la situación del proyecto. • Tomar decisiones sobre aspectos que tengan impacto en el alcance o plazos establecidos, gestionando los riesgos periódicamente.
Líder de SQA	• Definir de estándares de calidad y procedimientos de trabajo. • Asegurar la calidad del producto final, mediante controles periódicos para asegurar el cumplimiento de lo especificado y mediciones para comprobar calidad de producto y proyecto.
Líder de SCM	• Establecer elementos de configuración y nomenclatura para identificarlos. • Escoger repositorios para trabajar y mecanismos de respaldos, asegurando la disponibilidad de la última versión siempre. • Hacer cumplir el plan SCM con los integrantes del equipo. • Seguir la línea base, controlando las versiones y cambios en ella. • Establecer estándares y procedimientos a seguir con respecto a los cambios, para permitir un control de los mismos.
Tester	• En conjunto con arquitecto, SQA y desarrolladores definir el plan de pruebas. • Realizar pruebas de integración entre <i>front end</i>, base de datos y parte gráfica • Ejecutar pruebas de enlace

	• Coordinar pruebas en campo. • Realizar pruebas automáticas • Reportar resultados de pruebas.
--	--

Tabla 7-1: Responsabilidades por rol

7.1.4.1 Asignaciones de Roles

Luego de haber definido las metodologías de trabajo el equipo cuenta con información sobre los roles necesarios. Para definir la asignación, se tuvieron en cuenta las experiencias y características de cada uno de los integrantes intentando que la tarea de cada uno corresponda con los aspectos personales. Dado que nuestro equipo es pequeño, es necesario combinar varios roles en un mismo integrante y, a su vez, varios roles son asignados a ambos integrantes del equipo.

A continuación se indica el rol asignado a cada uno de los integrantes en el proyecto:

Rol	Andrés Martínez	Renato Moraes
Ingeniero de requerimientos	x	x
Arquitecto	x	
Desarrollador	x	x
Gerente		x
Líder de SQA	x	
Líder de SCM		x
Tester	x	x

Tabla 7-2: Roles asignados

La separación de roles es para asignar responsabilidades a cada integrante y que cada cual lidere su área específica, pero en la práctica sabemos que al ser un equipo de a dos cada integrante hará “de todo un poco” y las responsabilidades de cumplir con las tareas establecidas son de todo el equipo.

7.2 Gestión de la comunicación

Una correcta gestión de las comunicaciones es siempre necesaria para alcanzar los objetivos de los proyectos, por más que sea un proyecto con un equipo de dos integrantes. Cuando se trata de una migración con una alta exigencia en testeabilidad, la comunicación juega un rol fundamental.

En este capítulo se describe cómo se gestionaron las comunicaciones entre los diferentes interesados del proyecto y las herramientas seleccionadas para establecer las diferentes comunicaciones, ya fueran internas entre los miembros del equipo o externas entre los miembros del equipo, el tutor y demás interesados por parte del cliente.

Dada la buena relación con los demás interesados se optó por una gestión liviana y ágil.

Dentro de las comunicaciones separamos aquellas que son de carácter interno de las que son de carácter externo y que a continuación detallamos.

7.2.1 Interesados del proyecto

Integrantes del equipo:

- Andrés Martínez
- Renato Moraes

Tutor del equipo:

- José Szyman

Cliente (UTE):

- Migración del *front end*
- Migración de la base de datos alfanumérica.
- Jefa Lidia Dávila (jefa Andrés)
- Técnicos UTE (compañeros Andrés)
- Grupo Software
- Grupo Hardware
- Grupo Protecciones
- CMD

7.2.2 Herramientas de comunicación

Se dividen las herramientas de comunicación según si son destinadas a comunicaciones internas o externas.

7.2.2.1 Comunicación interna

La comunicación interna es sumamente importante, ya que debe favorecer a que el equipo se mantenga coordinado. Si bien se definieron una serie de roles y responsabilidades, es necesario mantener una comunicación fluida con el fin de evitar desviaciones que generen retrasos o re trabajos.

Herramientas	Uso	Ejemplo
Whatsapp	Comunicaciones informales, temática general. No se debe utilizar para toma de decisiones, si no para contactar y organizar rápidamente tareas con miembros del equipo.	• Cambio de horario de reunión • Aviso de llegada tarde
Celular	Comunicación informal instantánea entre dos miembros del equipo, carece de validez a la hora de toma de decisiones. Utilizado para contactar rápidamente al otro miembro del equipo.	• Consulta de tareas compartidas • Consejos y opiniones • Definir reuniones
Email	Comunicaciones formales e informales, Toma de decisiones o consultas importantes que se requiera mantener registro.	• Adaptación de proceso • Toma de decisiones • Problemas: atrasos, calidad, incumplimientos, etc. • Cambios: arquitectura, gestión de calidad, etc • Definir reuniones
Google drive	Comunicaciones formales. Utilizando google drive como repositorio de información, se definen una serie de planillas donde registrar los avances.	• Artefactos de puesta en producción. • Documentaciones técnicas varias.
One Drive	Comunicaciones formales. Se utiliza One drive para la documentación, ya que nos pareció mejor herramienta para trabajar on-line con los archivos de Word.	• Documentación • Informe de Avance
Reuniones presenciales	Medio de comunicación formal e informal para definir avances del proyecto y posibles problemas que surjan en el proyecto.	• Reuniones semanales de planificación.

Tabla 7-3: Herramientas de comunicación de uso interno.

7.2.2.2 Comunicación externa

La comunicación externa corresponde relacionada principalmente al contacto que el equipo tiene con el tutor y responsable de la cátedra.

Herramientas	Uso	Ejemplo
Email	Consultas generales con los interesados externos del proyecto. Informes de avances, relevamiento de requerimientos. A medida que el equipo vaya generando la documentación necesaria para la entrega del proyecto el tutor recibirá versiones digitales en las que podrá hacer sus comentarios o sugerencias.	• Corrección proyecto por parte del tutor o consultas varias al tutor. • Consulta a responsable a Cliente • Coordinación de reunión con cliente
Reuniones presenciales	Las reuniones presenciales serán necesarias cuando o bien el tutor o responsable de la cátedra quieran recibir un reporte avance del proyecto. En caso de ser necesario se podrán coordinar reuniones presenciales con el tutor para discutir sobre los puntos a desarrollar en el proyecto.	• Definición y validación de requerimientos • Demostración de producto

Tabla 7-4: Herramientas de comunicación de uso externo.

7.2.3 Planificación de la comunicación

En la siguiente tabla se describe la planificación, o cómo se previó llevar a cabo las comunicaciones durante el proyecto.

¿Qué se comunica?	¿Quién lo comunica?	¿A quién lo comunica?	¿Cómo se comunica?	¿Con qué frecuencia se
Planilla de comunicación	Equipo de trabajo	Gerente de proyecto	Reunión presencial con equipo	Al inicio del proyecto
Cronograma	Equipo de trabajo	Gerente de proyecto	Reunión presencial con equipo	Al inicio de cada etapa
Riesgos	Equipo de trabajo	Gerente de proyecto	Reunión presencial con equipo	Inicio del proyecto
Seguimiento de riesgos	Equipo de trabajo	Gerente de proyecto	Reunión presencial con equipo	Al inicio de cada etapa

¿Qué se comunica?	¿Quién lo comunica?	¿A quién lo comunica?	¿Cómo se comunica?	¿Con qué frecuencia se
Avances de Documentación	Equipo de trabajo y tutor	Gerente de proyecto	Via Email	Cuando sea necesario
Reuniones de planificación	Equipo de trabajo	Gerente de proyecto	Reunión presencial con equipo	Al inicio de cada etapa
Seguimiento del proyecto	Equipo de trabajo y tutor	Gerente de proyecto	Reunión presencial con equipo y tutor	En cada reunión con el tutor
Acta de reunión con tutor	Equipo de trabajo y tutor	Gerente de proyecto	Se archiva en repositorio	En cada reunión con el tutor
Acta de reunión con cliente	Equipo de trabajo	Gerente de proyecto	Se archiva en repositorio	En cada reunión con el cliente
Metodología de trabajo	Equipo de trabajo	Gerente de proyecto	Reunión presencial con equipo	Al inicio del proyecto
Proceso del negocio	Equipo de trabajo	Cliente	Reunión presencial, mail, teléfono	Al inicio del proyecto y cuando sea necesario
Dudas sobre requerimientos	Cliente	Equipo de trabajo	Reunión presencial, mail, teléfono	Cuando sea necesario
Cambios en los requerimientos	Equipo de trabajo	Gerente de proyecto	Reunión presencial con equipo	Cuando sea necesario
Defectos encontrados	Equipo de desarrollo	Tester	Se registra en repositorio	En cada etapa de testing
Impedimentos técnicos	Gerente de proyecto	Desarrollador	Reunión presencial con equipo	Cuando sea necesario
Versión final de documento	Gerente de proyecto	Equipo y Tutor	Se archiva en repositorio y se envía vía mail	Cuando haya actualizaciones en documentación

Tabla 7-5: Planificación de la comunicación

Ver Anexo 17 – Reuniones Semanales

7.2.4 Conclusiones y lecciones aprendidas

Por tratarse de un grupo pequeño, con solo dos integrantes, donde Andrés tiene muy buena relación con el cliente se optó por una definición sencilla e informal. Se considera que la misma fue exitosa, siempre existió una comunicación muy fluida que permitió concluir el proyecto sin ningún problema referente al plan de comunicación.

7.3 Gestión del tiempo

En esta sección se describen los procesos requeridos para gestionar en plazo el proyecto. Las actividades fueron definidas y secuenciadas en las diferentes fases del proyecto, faltaría estimar duración y esfuerzo de cada una para poder diseñar el cronograma.

7.3.1 Estimación de recursos y duración

No existía referencia de cuánto tiempo llevaría un proyecto así. Decidimos que de acuerdo a la cantidad de elementos diferentes que se contaron en el SCADA X se estimaría un tiempo para la conversión de toda la pieza de software.

Analizamos en detalle el archivo propietario origen (Anexo 18 – Archivo Propietario), vimos como registraba los elementos y cómo se iba a extraer con un lector. Se tomó el archivo equivalente existente en SCADA Mirage y a partir de ahí se estimó que cada objeto en total nos llevaría 24 hs para la conversión. Contamos 15 objetos diferentes, lo que nos dio un total de 360 horas, para convertir los elementos de software.

Al momento de la estimación ya le dedicamos hora a interpretar el archivo propietario del SCADA X, y sabemos que nos llevará unas 50 horas adicionales.

Y por último tenemos que integrar el software en Mirage, que estimamos 100 horas adicionales.

Según diapositivas de Ingeniería de Software 2, por cada 100 horas de un proyecto pequeño a mediano porte, como el nuestro, se dedicarán 20 hs a la gestión, (ver Anexo 19 – Diapositiva Heurística para estimación.)

Resumiendo estimamos las siguientes horas en programación.

Tareas	Hs
Interpretar archivo propietario	50
Convertir elementos	360
Integrar solución en Mirage	100
Gestión de Proyectos	102
Total	612

Tabla 7-6: Estimación software

Por lo tanto se necesitaran 612 horas para la pieza de software.

De acuerdo a las recomendaciones de Universidad ORT, es necesario dedicarle 15 horas semanales por integrante, un total de 1560 horas que las distribuiremos de la siguiente manera:

	Renato	Andrés	Total
Setiembre	30	30	60
Octubre	60	60	120
Noviembre	45	45	90
Diciembre	55	55	110
Enero	75	75	150
Febrero	75	75	150
Marzo	75	75	150
Abril	60	60	120
Mayo	60	60	120
Junio	30	30	60
Julio	75	75	150
Agosto	120	120	240
		Total	1.520

Tabla 7-7: Horas disponibles por integrante.

Setiembre y octubre serán dedicados para la fase cero, entonces estimamos que entre noviembre y mediados de abril cubriríamos las horas necesarias para el software de conversión.

El resto del tiempo se utilizaría para los sistemas de pruebas, que se necesitarán antes de colocar en producción.

En conclusión, las duraciones estimadas para cada fase del proyecto son las siguientes:

- **Fase 0 - 29/09/2016 al 02/11/2016:** Análisis y relevamiento de necesidades del cliente, especificación de requerimientos; diseño arquitectónico; planificación del proyecto.
- **Fase 1 - 03/11/2016 al 18/08/2017:** Se lleva a cabo el desarrollo del producto. Seguimiento de gestiones y *testing* varios. Se divide en 6 iteraciones:
 - **Iteración 1 - Migrar elementos - 03/11/2016 al 17/04/2017:**
Análisis SCADA X, digerir archivo propietario, construir los diferentes elementos y probar el XML obtenido en SCADA Mirage.
 - **Iteración 2 – Migrar unifilares - 18/04/2017 al 10/05/2017:**
Una vez procesado todos los elementos e integrado en una única solución, se comienzan a convertir los unifilares. Realizando reporte de bugs para corrección del software.
 - **Iteración 3 – Pruebas de enlace - 11/05/2017 al 21/06/2017:**

Consiste en comparar medidas en ambos sistema.

- **Iteración 4 – Pruebas en campo 22/06/2017 al 03/08/2017:**
Coordinaciones para realizar pruebas en campo entre un 5 y 10% del total, son pruebas muy costosas.
- **Iteración 5 – Pruebas automáticas 11/05/2017 al 29/06/2017:** Historial de cambios, seguridad y reportes.
- **Iteración 6 – Puesta en producción - 18/07/2016 al 07/08/2017:**
En esta fecha se estima que se pondrá en producción; previamente debe estar pronto el centro de cómputos nuevos y todos los unifilares probados.

- **Fase 2 - 04/08/2017 al 23/08/2017:** Finalización de documentación.

1.1.1. Desarrollo del cronograma

A continuación se muestra lo expuesto anteriormente en un diagrama de Gantt.

	Nombre de tarea ▼	Duración ▼	Comienzo ▼	Fin
1	▲ Proyecto	235 días	jue 29/09/16	mié 23/08/17
2	▷ Fase 0	25 días	jue 29/09/16	mié 02/11/16
11	▲ Fase 1	207 días	jue 03/11/16	vie 18/08/17
12	▷ Iteración 1 - Migrar Elementos	118 días	jue 03/11/16	lun 17/04/17
22	▷ Iteración 2 - Migrar Unifilares	17 días	mar 18/04/17	mié 10/05/17
34	▷ Iteración 3 - Pruebas de Enlace	30 días	jue 11/05/17	mié 21/06/17
45	▷ Iteración 4 - Pruebas en Campo	31 días	jue 22/06/17	jue 03/08/17
56	▷ Iteración 5 - Automáticas	36 días	jue 11/05/17	jue 29/06/17
70	▷ Iteración 6 - Puesta en Producción	11 días	vie 04/08/17	vie 18/08/17
74	▷ Fase 2	14 días	vie 04/08/17	mié 23/08/17

Figura 7.3: Resumen cronograma

El cronograma detallado se puede ver en Anexo 20 – Cronograma del proyecto.

7.3.2 Métricas relacionadas a la gestión del tiempo.

Para tener un mayor control sobre el proceso, se utilizó una planilla donde ingresábamos las horas que le dedicamos por tarea. La planilla se puede ver Anexo 21 – Planilla de horas semanales.

Al inicio del proyecto realizamos la siguiente estimación de las tareas:

	Estimados
Ante proyecto	38
Arquitectura	20
Conversión³	100
Documentación	80
Gestión proyecto	150
Investigación	80
Programación	612
Requerimientos	60
Riesgos	20
Testing	360
Total	1520

Tabla 7-8: Estimación duración tareas

Al finalizar el proyecto se obtuvieron los siguientes resultados:

	Reales	Estimados	Desviaciones
Ante proyecto	38	38	0
Arquitectura	16	20	-4
Conversión	277	100	177
Documentación	130	80	50
Gestión proyecto	156	150	6
Investigación	101	80	21
Programación	847	612	235
Requerimientos	71	60	11
Riesgos	32	20	12
Testing	454	360	94
	2122	1520	602

Tabla 7-9: Desviaciones en duración tareas

Finalizamos con una desviación del 40%. El principal motivo de esta diferencia estuvo en la programación y el *testing*, se nos extendió mucho la solución, terminó siendo una aplicación con más de 110 clases y 7000 líneas de código.

³Conversión: nos referimos a la etapa en la cual se ingresan los unifilares SCADA X y se obtienen los unifilares en el nuevo sistema SCADA Mirage. Incluye los ajustes gráficos y las pruebas de integración, que consiste en verificar manualmente que los unifilares sean iguales.

En el siguiente gráfico se puede ver la magnitud de la desviación:

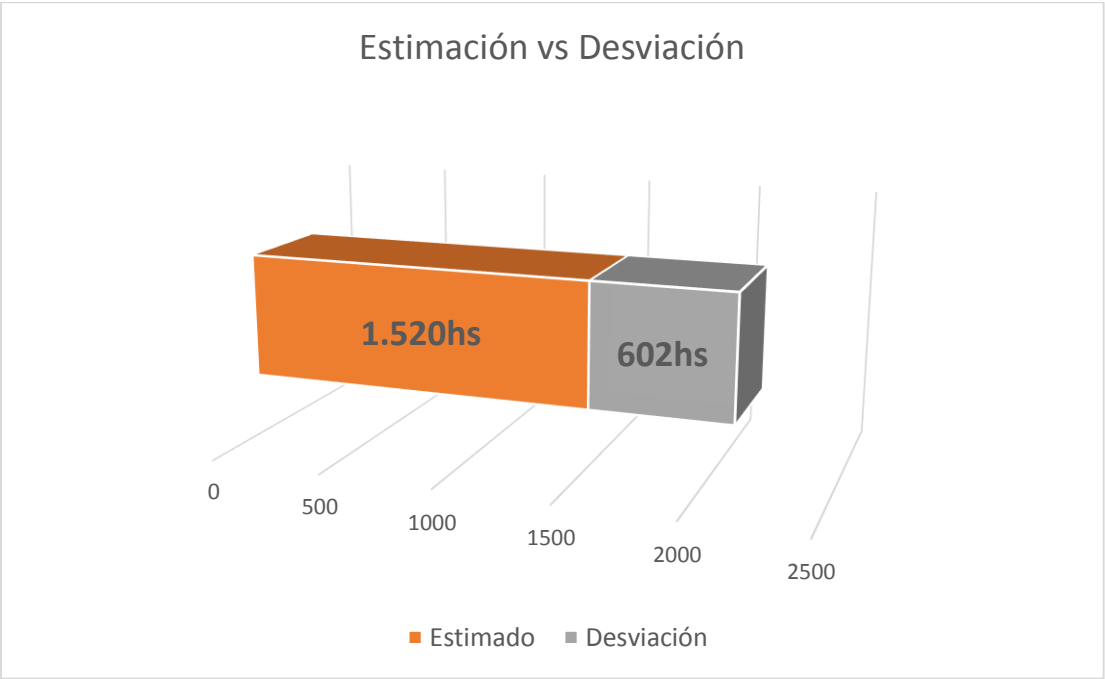


Figura 7.4: Desviación respecto a la estimación

En la siguiente gráfica se pueden diferenciar qué porcentaje de esfuerzo ha insumido cada etapa del proceso:

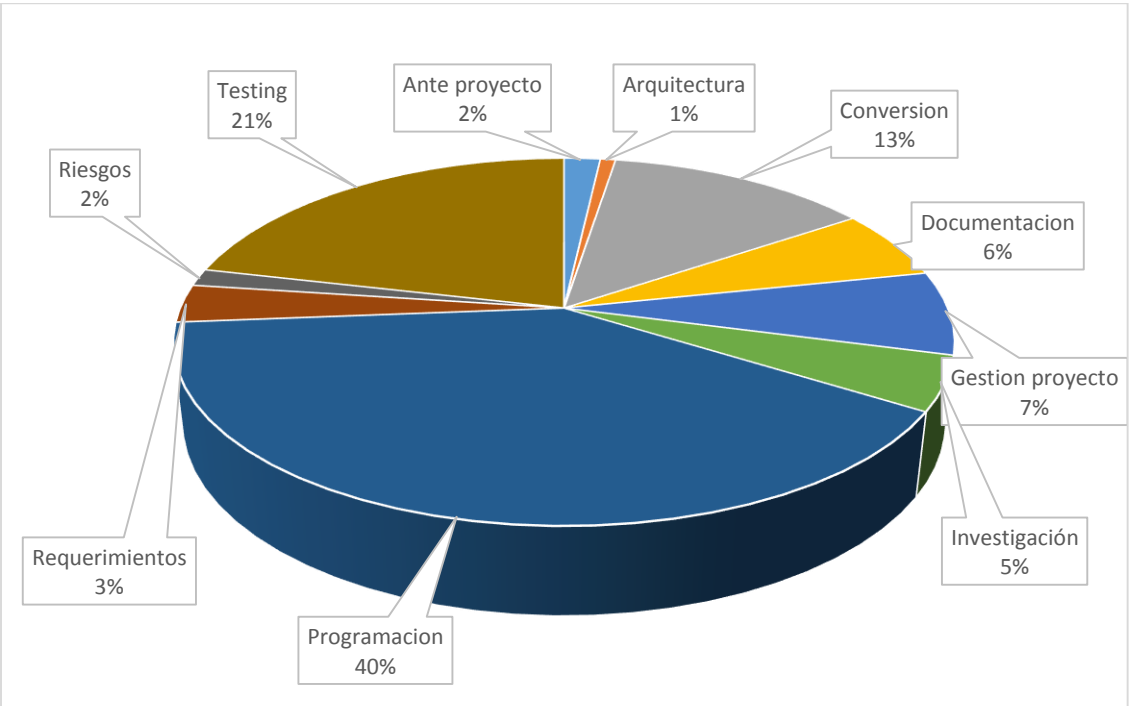


Figura 7.5: Distribución de las horas trabajadas según el proceso.

Dado el perfil del proceso se puede observar que el 61% del tiempo fue utilizado en programación y testing. Dicho 61% fue resultado de un software más largo y complejo de lo que habíamos estimado y, en vista de que la calidad es muy importante para el proyecto, tuvimos la necesidad de dedicarle mucho tiempo a las pruebas.

Con respecto a la relación gestión entre la gestión y el proceso de Ingeniería estamos dentro de los valores esperados, para el tamaño de software. Ver Anexo 19 – Diapositiva Heurística para estimación.

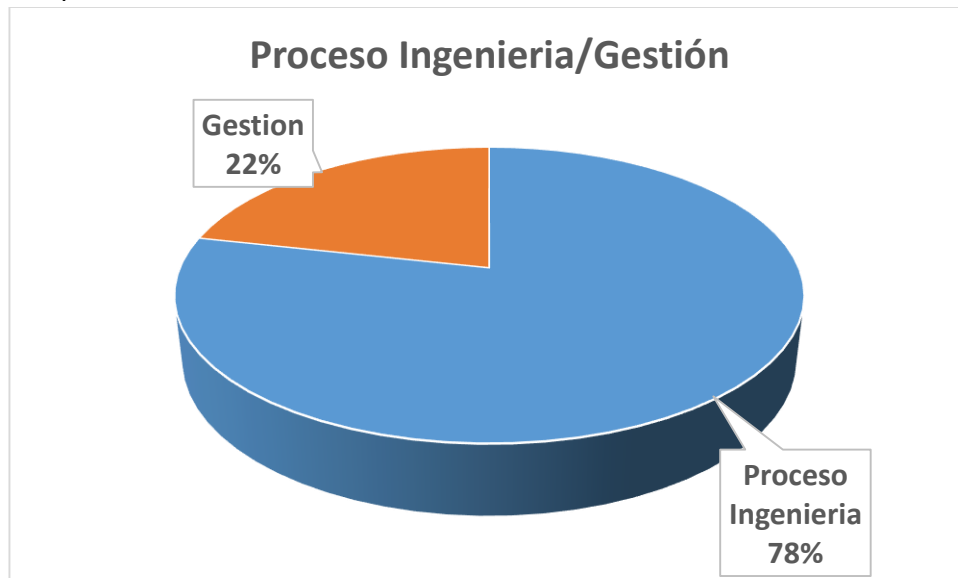


Figura 7.6: Relación entre el tiempo dedicado a gestión y programación.

A continuación se analizará la incidencia de los ítems con mayor esfuerzo y, por ende, mayor desviación en el total del proyecto. Programación implicó un 40% de las horas dedicadas y estuvo un 38% por encima de lo estimado.



Figura 7.7: Tiempo dedicado a programación.

El *testing*, por otro lado, nos insumió el 21% del tiempo y estuvo un 26% por encima de lo estimado:

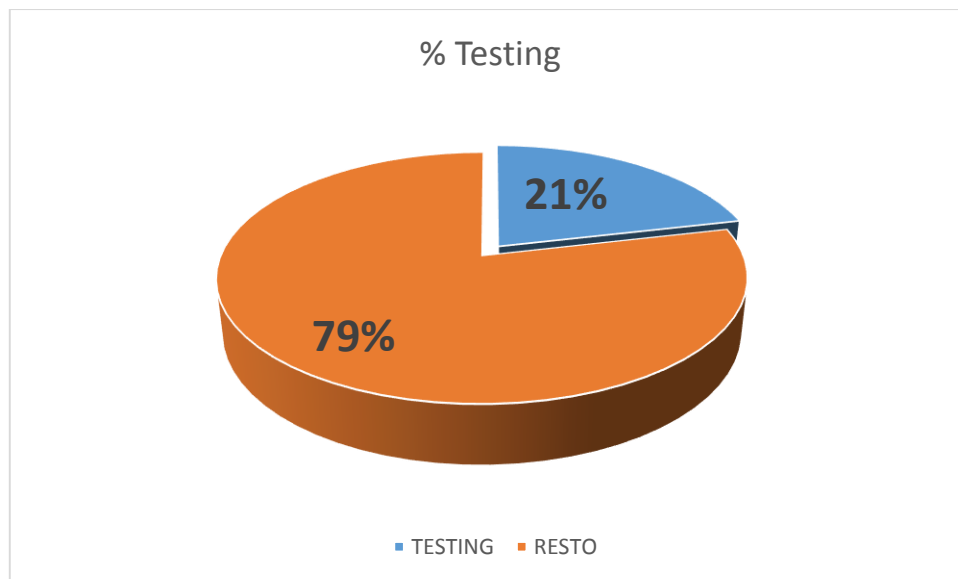


Figura 7.8: Tiempo dedicado a testing.

Y por último vamos a destacar uno de los ítems con mayor desviación, que fue la conversión. Proceso en el que se ingresan los unifilares del SCADA X y se convierten al SCADA Mirage, incluyen correcciones estéticas y pruebas de integración.

La conversión representó el 13% de las horas dedicadas al proyecto.

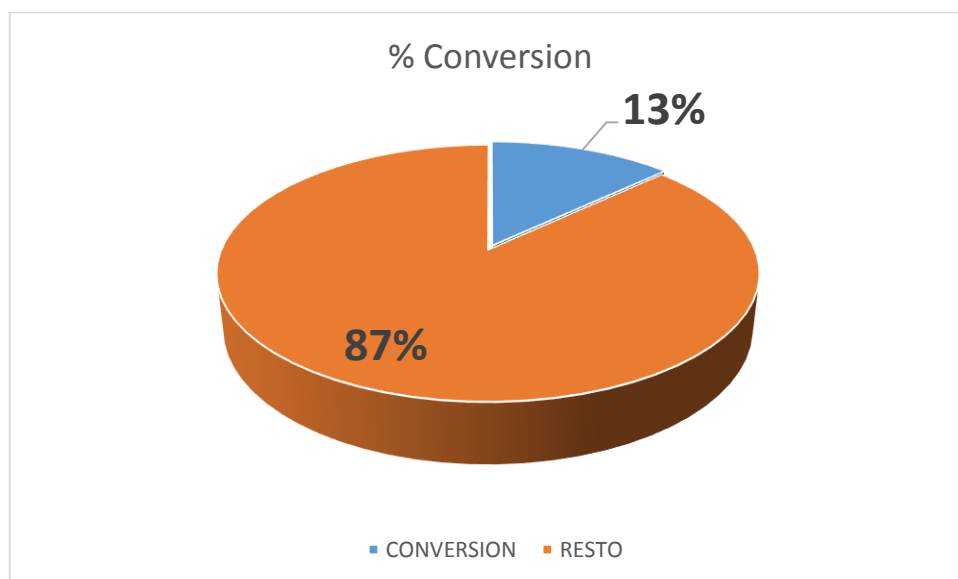


Figura 7.9: Tiempo dedicado a Conversión.

Pero con una desviación de 177% respecto a lo proyectado.

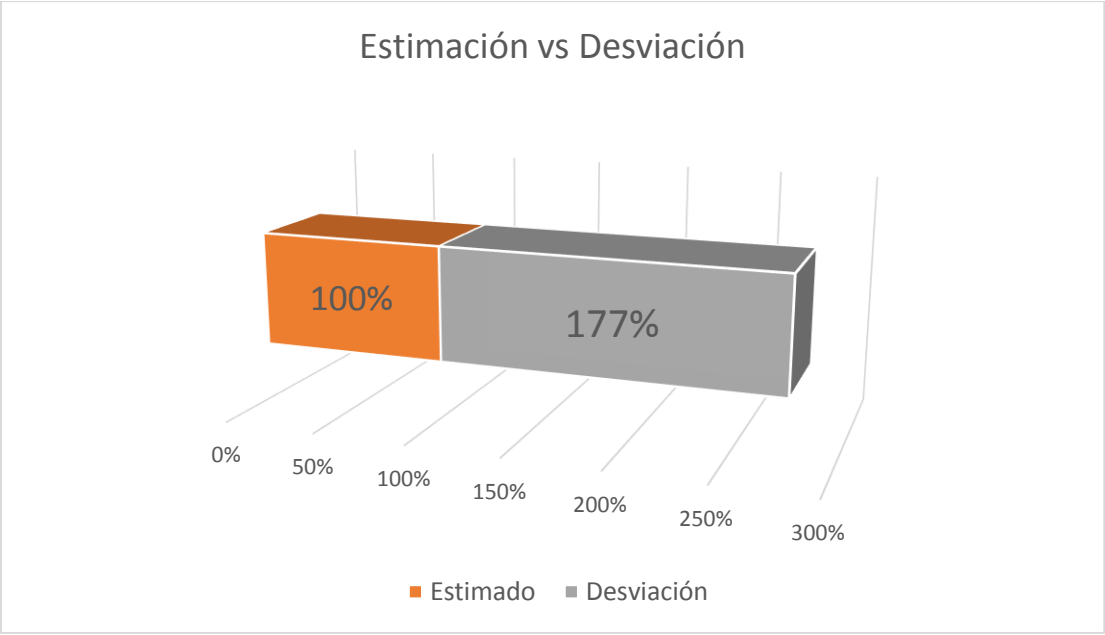


Figura 7.10: Desfasaje entre estimación y el tiempo real.

Del total de las 2122 horas dedicadas al proyecto el mayor esfuerzo estuvo en los meses entre marzo y junio. A continuación se muestra la cantidad total de horas trabajadas en la ejecución del proyecto:

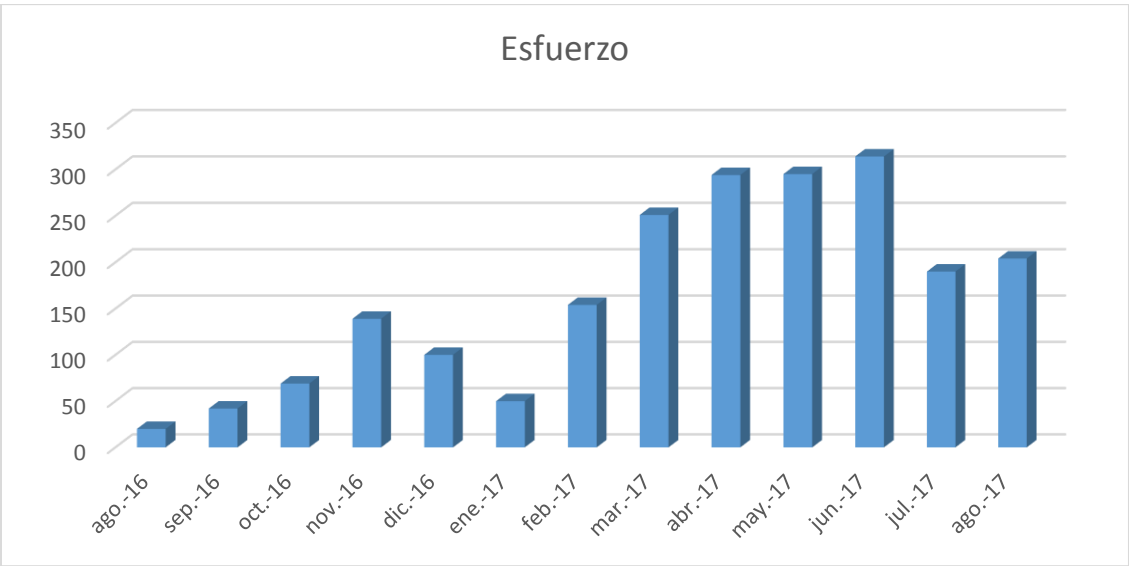


Figura 7.11: Esfuerzo mes total

Se puede observar que a partir de marzo el tiempo destinado al proyecto fue mucho mayor y esto se debe a lo siguiente:

- ✚ Muchas de las horas dedicadas a investigación y a digerir la herramienta ITIL no fue correctamente registrada en los primeros meses del proyecto.
- ✚ A partir de febrero, cuando se comenzó realmente con el desarrollo del software y nos encontramos con varios inconvenientes, que se tuvieron que investigar, el desarrollo del software nos llevó un 40% más del tiempo estimado.
- ✚ A partir de marzo, Andrés comenzó a dedicarle muchas horas de desarrollo en su trabajo, la migración es una prioridad de UTE.

El esfuerzo resultó dispar entre los integrantes del grupo, como se puede ver en la siguiente gráfica.

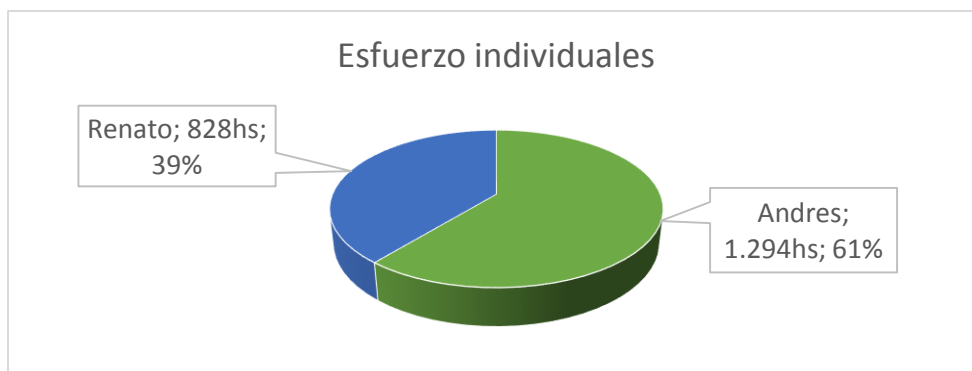


Figura 7.12: Esfuerzos Individuales.

La diferencia, como lo hemos mencionado, se debe a que Andrés comenzó a utilizar muchas horas de su jornada laboral para el desarrollo y *testing*.

7.4 Gestión de riesgos

Según la definición de ITIL V3 *Service Transition* [18], “un riesgo en un proyecto puede definirse como la incertidumbre del resultado, ya sea una oportunidad positiva o una amenaza negativa. Es el hecho de que hay incertidumbre lo que crea la necesidad de atención y gestión formal del riesgo. El propósito de la gestión formal del riesgo es permitir una mejor toma de decisiones basada en una comprensión adecuada de los riesgos y su posible impacto en el logro de los objetivos”.

Para la gestión de Riesgos ITIL recomienda en este sentido el marco *Management of Risk* (M_o_R) de *Open Geospatial Consortium* (OGC)[19][20].

La misma sugiere los siguientes subprocesos:

Análisis del impacto y riesgo al negocio: Cuantificar el impacto de la pérdida de servicios y activos en una empresa y determinar la probabilidad de una amenaza o la vulnerabilidad ante la misma. El resultado de este proceso es el registro de riesgos, una lista de riesgos que deben atenderse según su prioridad.

Evaluación de mitigación de riesgo requerida: Determinar dónde se necesitan medidas de mitigación de riesgo e identificar a los Responsables del Riesgo, quienes están a cargo de la implementación y el mantenimiento continuo.

Monitorización de riesgo: Monitorear el progreso de la implementación de contramedidas, y tomar acción correctiva de ser necesario.

7.4.1 Análisis del impacto y riesgo al negocio

La gestión de riesgos inicia junto con el proyecto. Para identificarlo lo separamos en las siguientes 5 categorías:

- a. Riesgos asociados al cliente
- b. Riesgos asociados al desarrollo
- c. Riesgos asociados al equipo
- d. Riesgos asociados al proyecto
- e. Riesgos asociados a la tecnología.

Luego los ponderamos y se trazaron diferentes estrategias para que los riesgos no se materialicen. Los mismos se controlarán en etapas durante el desarrollo del proyecto con el objetivo de monitorear y controlar la evolución de cada uno de los riesgos identificados. En cada una de estas etapas se analizará si se obtuvieron nuevos riesgos, no identificados hasta ese momento, los mismos deberán ser analizados e incorporados a la gestión de riesgos. Dichas etapas de control se ejecutarán cada dos meses o al final de cada etapa de iteración.

A cada riesgo se le asignaron las siguientes propiedades:

- Id.
- Descripción.
- Probabilidad estimada de ocurrencia.
- Impacto percibido.
- Magnitud del riesgo (como el resultado de la probabilidad x el impacto).
- Medidas adoptadas para mitigar el riesgo.
- Disparador que establece el criterio de medición para identificar cuándo un riesgo determinado se materializa.
- Plan de contingencia en caso de concretarse el riesgo.

Los riesgos son evaluados por los dos miembros del equipo en reuniones cada dos meses.

Se utilizará la Magnitud del riesgo para realizar un análisis cualitativo de los riesgos y determinar sus potenciales consecuencias. Para eso se definieron las siguientes escalas:

Probabilidad ocurrencia	
0.0	No probable
0.2	Poco probable
0.4	Probable
0.6	Muy probable
0.8	Altamente probable
1.0	Un hecho (problema)

Tabla 7-10: Probabilidad de Ocurrencia

Impacto	
0	Ninguno
1	Marginal
2	Poco importante
3	Importante (puede retrasar el proyecto)
4	Crítico (puede detener el proyecto)
5	Catastrófico (fracaso del proyecto)

Tabla 7-11: Impacto de los riesgos

7.4.1.1 Lista de riesgos

La lista de riesgos se confeccionó al comienzo del proyecto, identificando riesgos relacionados a las distintas categorías definidas en la planificación. Luego, fue evolucionando a medida que se detectaron nuevos riesgos.

A continuación se muestran los listados de riesgos según su categoría:

a) Riesgos asociados al desarrollo:

Num	RIESGOS
R1	Que algún integrante no tenga experiencia en alguna actividad puede repercutir en la gestión de algunas áreas.
R2	Si ocurren problemas con el versionado, se pueden perder horas de trabajo realizadas o ser necesarias horas de trabajo para corregir el problema. Esto impacta negativamente en el calendario y el costo del proyecto.
R3	Que los requerimientos se definan mal implicará una carga de trabajo adicional para poder satisfacerlos. Lo que se traduce en un impacto negativo en costos, calendario y alcance del proyecto.
R4	Diseño arquitectónico incorrecto o incompleto
R5	Que a la hora de integrar toda la solución existan complejidades técnicas que no lo permitan o retrase el proyecto.

Tabla 7-12: Riesgos asociados al desarrollo

b) Riesgos asociados a la tecnología:

Num	RIESGOS
R6	Si surgiera una dificultad imprevista de carácter técnico, se deberán invertir horas adicionales para solucionarla. Esto puede afectar negativamente el calendario o alcance del proyecto.
R7	Que surja otro SCADA ofreciendo otras ventajas.
R8	Que SCADA Mirage proponga hacer la migración.
R9	Si bien SCADA Mirage funciona en otros CMD de UTE, puede existir la posibilidad que algún elemento raro no esté en las posibilidades de Mirage.

Tabla 7-13: Riesgos asociados a la tecnología

c) Riesgos asociados al equipo:

Num	RIESGOS
R10	Ausencias no planificadas de los integrantes del equipo.
R11	Falta de experiencia de los integrantes del equipo con las tecnologías seleccionadas para el desarrollo.
R12	Que algún integrante del equipo por motivos personales no tenga tiempo y repercuta en el proyecto.
R13	Problemas personales entre miembros del equipo va a repercutir en las comunicaciones.

Tabla 7-14: Riesgos asociados al equipo

d) Riesgos asociados al cliente:

Num	RIESGOS
R14	Que UTE cancele la migración por intereses de terceros.
R15	No se puedan realizar las pruebas de enlace, debido a que CMD no nos deje bajar el servidor actual.
R16	Que no construyan el centro de cómputos en el año del proyecto.
R17	Que UTE no tenga pronta la migración del <i>front end</i>
R18	Que UTE no tenga pronta la migración de la base de datos alfanumérica.
R19	Por conflictos internos en UTE no se puedan realizar las pruebas en campo

Tabla 7-15: Riesgos asociados al cliente

e) Riesgos asociados al proyecto:

Num	RIESGOS
R20	Falta de experiencia al planificar proyectos de desarrollo de software.
R21	Errores al estimar esfuerzo de actividades.
R22	Pérdida de datos como consecuencia de eliminarlos, extravió o robo de computadoras personales.
R23	Dificultad del diseño de las pruebas automáticas.
R24	No cumplir con la documentación.

Tabla 7-16: Riesgos asociados al proyecto

7.4.2 Análisis del impacto y riesgo al negocio

Para poder mitigar los riesgos críticos, que en caso de materializarse pudieran afectar el desarrollo del proyecto, se realizaron las siguientes actividades:

- Determinar responsables, en este caso como somos dos los integrantes del equipo ambos somos responsables de los mismos.
- Cálculo de la magnitud, entendida como el producto entre la probabilidad de ocurrencia y el impacto del mismo si ocurriese.
- Plan de respuestas.
- Plan de contingencia.
- Disparador.

En la siguientes dos tablas se define la probabilidad de ocurrencia, el impacto, la magnitud y también se pueden ver definidos los planes de respuesta que describen las acciones que se deben tomar para mitigarlos, los planes de contingencia con acciones a hacer si ocurren, y los disparadores que indicaban el umbral de cuando estaba por ocurrir el riesgo y así poner en marcha dicho plan de contingencia.

Riesgo	Probabilidad de ocurrencia	Impacto	Magnitud del riesgo	Plan de respuesta
R1	0.4	3	1.2	Cada integrante es responsable de realizar la tarea asignada, por lo que debe previamente debe conocer la actividad o buscar una solución en tiempo y forma.
R2	0.2	3	0.6	Definir el plan de versiones y realizar <i>backup</i> semanales en discos duros de todos los repositorios.
R3	0.6	4	2.4	Validar con el cliente los requerimientos
R4	0.4	3	1.2	Realizar los diferentes diagramas de arquitectura previamente.
R5	0.2	4	0.8	Tener referentes a quien consultar dudas, por ejemplos compañeros de trabajo con más experiencia.
R6	0.4	3	1.2	Tener referentes a quien consultar dudas, por ejemplos compañeros de trabajo con más experiencia.
R7	0	4	0	Realizar un análisis FODA de la migración con Mirage, para tratar de convencer a UTE a continuar con la migración.
R8	0.2	4	0.8	Realizar un análisis FODA de la migración con Mirage, para tratar de convencer a UTE a continuar con la migración.
R9	0.2	5	1	Buscar representarlo con un objeto que cumpla similares responsabilidades
R10	0.2	3	0.6	Comunicarse con el integrante, ver las razones de ausencia y realizar un plan para recompensar las horas perdidas.
R11	0.4	2	0.8	Tener referentes a quien consultar dudas, por ejemplos compañeros de trabajo con más experiencia.
R12	0.4	3	1.2	Cada integrante tiene la capacidad de realizar todas las tareas, por lo que puede reemplazar en dicha ausencia.
R13	0.2	3	0.6	Realizar reuniones informales al inicio y compartir actividades varias.
R14	0.2	5	1	Realizar un análisis FODA de la migración con Mirage, para tratar de convencer a UTE a continuar con la migración.
R15	0.2	5	1	Tener acceso a la alta Gerencia para solicitar una orden para realizar las pruebas.
R16	0.6	4	2.4	Hacer un seguimiento de avances de las obras del centro de cómputos
R17	0.6	4	2.4	Hacer un seguimiento de avance del desarrollo del <i>front end</i>
R18	0.6	4	2.4	Hacer un seguimiento de avance del desarrollo de la base de datos alfanumérica
R19	0.4	5	2	Tener acceso a la alta Gerencia para solicitar una orden para realizar las pruebas.
R20	0.4	3	1.2	Diseñar y cumplir con el cronograma. Volver a definir el cronograma si es necesario.
R21	0.4	3	1.2	Ser conservadores en la estimación de las actividades, aumentando la duración estimada en cada caso en un 50%
R22	0.2	2	0.4	Trabajar en la nube para evitar estos problemas. Por ejemplo Google Drive, Dropbox y otros repositorios para la solución, además hacer un respaldo semanal en un disco externo.
R23	0.2	4	0.8	Tener referentes a quien consultar dudas, por ejemplos compañeros de trabajo con mas experiencia.
R24	0.6	4	2.4	Establecer un cronograma con todas las actividades y planes a documentar.

Tabla 7-17: Plan de respuesta

A continuación el plan de respuesta y el disparador.

Riesgo	Plan de contingencia	Disparador
R1	Resolver dificultad entre ambos integrantes y/o consultar experto.	Retraso en un 50% de lo previsto en una actividad, debido a la falta de experiencia
R2	Cambiar de proveedor	Problemas para acceder a proveedores de repositorios.
R3	Volver a definir los requerimientos	Que se tengan que modificar cuatro requerimientos.
R4	Re definir la arquitectura.	Que se dificulte el desarrollo a partir de la arquitectura seleccionada.
R5	Reunirse con SCADA Mirage e intentar resolver con ellos la complejidad dada.	Transcurrir el 30% destinado a la integración sin obtener avances y no se ven soluciones
R6	Realizar consultas a los referentes	Dificultades sin encontrar solución.
R7	Presentar a UTE el FODA y convencerlos de continuar con la misma.	UTE otorgue al otro SCADA la migración
R8	Presentar a UTE el FODA y convencerlos de continuar con la misma.	UTE autorice a SCADA Mirage hacer directamente la migración.
R9	Contactarse con Mirage para solicitarle que lo agreguen	Elemento encontrado.
R10	De ser constantes las ausencias definir con coordinador y tutor alternativas a esta situación.	3 ausencias no planificadas y/o constantes ausencias planificadas.
R11	Realizar consultas a los referentes	Problemas sin encontrar solución.
R12	Que el otro integrante realice las tareas pendientes.	Que alguno de los integrantes tenga un problema familiar que lo ausente durante mas una semana.
R13	Hacer reuniones para mediar el conflicto.	Surja algún conflicto.
R14	Presentar a UTE el FODA y convencerlos de continuar con la misma.	UTE cancela la migración.
R15	Reunirse con Gerencia para solicitar una orden para realizar las pruebas.	Pruebas no autorizadas
R16	No existe, asumimos este riesgo.	Atraso en la construcción.
R17	Ayudar en el desarrollo.	Atraso en el desarrollo.
R18	Ayudar en el desarrollo.	Atraso en el desarrollo.
R19	Reunirse con Gerencia para solicitar una orden para realizar las pruebas.	Pruebas no autorizadas
R20	Consultar con tutor para que nos oriente en la planificación.	Atraso en más de un 50% de la duración de cada tarea.
R21	Consultar con tutor para que nos oriente en la planificación.	Atraso en más de un 50% de la duración de cada tarea.
R22	Obtener último respaldo de repositorio.	Pérdida de información no disponible en la nube.
R23	Realizar consultas a los referentes	Transcurrir el 30% destinado a la integración y no se obtuvieron avances y no se ven soluciones
R24	Redefinir el cronograma.	Atraso en más de un 50% de la duración de cada tarea a documentar.

Tabla 7-18: Plan de contingencia y disparadores

En las tablas anteriores se priorizaron los riesgos por colores. Los colores se definieron según la siguiente matriz de impacto:

P/I	0	1	2	3	4	5
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.2	0.0	0.2	0.4	0.6	0.8	1.0
0.4	0.0	0.4	0.8	1.2	1.6	2.0
0.6	0.0	0.6	1.2	1.8	2.4	3.0
0.8	0.0	0.8	1.6	2.4	3.2	4.0
1.0	0.0	1.0	2.0	3.0	4.0	5.0

Tabla 7-19: Matriz de Probabilidad e Impacto

Los valores son del momento inicial de evaluación de cada riesgo, habiendo sido revisados cada dos meses o al final de cada etapa de iteración, por el equipo durante el proyecto de forma de mantenerlos actualizados.

Las acciones se toman de acuerdo a la categoría, como se describen en la siguiente tabla.

Rango	Prioridad	Descripción
0.2 a 0.8	Baja	Aceptado
1 a 2	Media	Requiere atención
2.4 a 5	Alta	Prioritario

Tabla 7-20: Niveles de Prioridades

7.4.3 Monitorización del riesgo

Cada dos meses y/o al final de cada iteración se hicieron evaluaciones de todos los riesgos y de acuerdo al análisis cualitativo inicial los siguientes riesgos se atacaron desde el momento en que se detectaron.

Evolución de los riesgos en las diferentes evaluaciones. Ver en Anexo 22 – Tabla evaluación Riesgos.

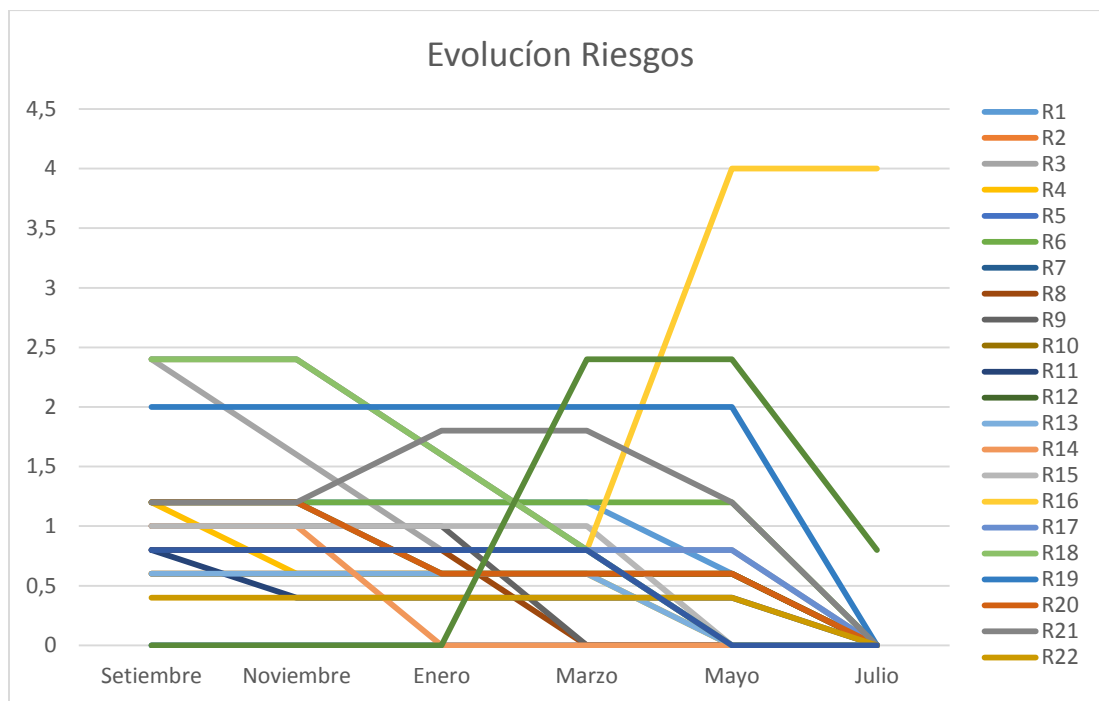


Figura 7.13: Evolución Riesgos

Cada riesgo que estaba en la categoría Alta fue atacado inmediatamente. A continuación se detallan los que en algún momento pasaron por dicha categoría.

Num	Riesgo
R3	Que los requerimientos se definan mal implicará una carga de trabajo adicional para poder satisfacerlos. Lo que se traduce en un impacto negativo en costos, calendario y alcance del proyecto.
R16	Que no construyan el centro de cómputos en el año del proyecto.
R17	Que UTE no tenga pronta la Migración del <i>front end</i> .
R18	Que UTE no tenga pronta la Migración de la base de datos alfanumérica.
R24	No cumplir con la documentación.

Tabla 7-21: Riesgos altos

A continuación la evolución de estos riesgos:

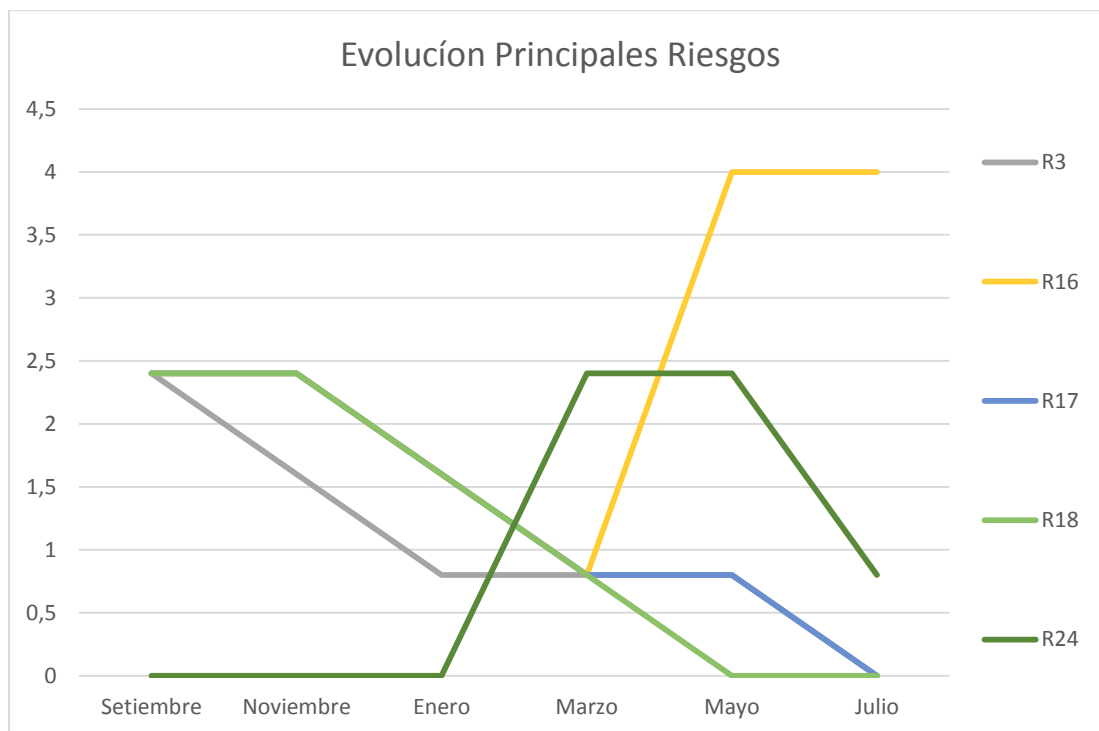


Figura 7.14: Evolución Principales Riesgos

Tres de estos 5 riesgos dependen del cliente, por lo tanto son los más difíciles para tomar medidas y mitigarlos.

7.4.3.1 Requerimientos (R3)

Si bien los requerimientos son estables, una mala definición de los mismos podría ocasionar atrasos, para eso se definió la lista de riesgos de manera temprana y se validaron ante UTE. Si bien se modificaron algunos requerimientos, este riesgo se controló y en la segunda evaluación en noviembre lo bajamos de categoría.

7.4.3.2 Centro de cómputos (R14)

Para el funcionamiento del nuevo SCADA es necesaria, la construcción de un centro de cómputos nuevo ya que el sistema origen corre sobre la plataforma UNIX, y el sistema destino SCADA Mirage corre sobre la plataforma Windows. Fue necesario prever el lugar físico para los nuevos servidores y centros de trabajo. Por tratarse de una inversión importante, que sin ella no era viable la migración, se le asignó un impacto muy alto.

Si bien para UTE este proyecto es de prioridad, los tiempos que el Ente maneja muchas veces no son los mismos que manejamos en el proyecto. Teníamos la certeza de que se iba a construir desde un inicio, pero existía cierto riesgo que no se hiciera en el plazo estipulado, teniendo en cuenta que los procedimientos de compras estatales son demorados.

Realizamos un control periódico de avance en este sentido y en el mes de diciembre ya se había realizado la gestión de compras de servidores y asignado el lugar físico, en la

revisión de riesgos del mes de enero se le bajó la probabilidad de ocurrencia y bajó de categoría, pasó a ser un riesgo que si requiere atención, pero no de tipo prioritario.

Este era, hasta entonces un riesgo controlado, pero al hacer las consultas en la evaluación de mayo, nos comunicaron que existió un inconveniente en la entrega de los servidores y los mismos recién llegarían en setiembre. Por error administrativo no habían incluido nuestros servidores en el pedido previsto para mayo; luego del reclamo pertinente la nueva fecha establecida para la entrega es setiembre.

De todas formas continuamos con el desarrollo para terminarlo antes del 23 de agosto y que únicamente se quede a la espera de los servidores.

Los equipos de red si han sido configurados y acondicionado el centro de cómputos. Existen posibilidades que para mediados de setiembre, antes de la defensa se pueda hacer la migración. Periódicamente se realizaran consultas referentes a la llegada de dichos servidores.

7.4.3.3 Migración front end (R15)

Como ya se ha explicado la migración cuenta de tres partes: el *front end*, la base de datos y la parte gráfica.

Fron tend son los protocolos de comunicación, los protocolos usados son el 104, 101 y DNP. Esta parte de la migración le correspondía a UTE, nosotros monitoreamos periódicamente que se viniera trabajando en dicha migración.

En abril se realizaron las primeras pruebas con la migración de sub-estaciones pilotos y el *front end* funcionó a la perfección, por lo que en nuestra evaluación de mayo se bajó el riesgo a categoría “Requiere Atención”.

Luego de la conversión a finales de mayo se comprobó que la migración del *front end* fue un éxito. Este riesgo pasó a categoría “No probable”.

7.4.3.4 Migración Base de datos (R16)

En cuanto a la base de datos, esta contiene los valores de las señales que llegan de campo, se deben crear las tablas que se almacenarán en SQL Server. Al igual que en el caso del *front end*, esta parte de la migración le correspondió a UTE.

En abril se realizaron las primeras pruebas con la migración de subestaciones pilotos y la migración de la base de datos funcionó a la perfección, por lo que en nuestra evolución de mayo se bajó el riesgo a categoría “Requiere Atención”.

Luego de la conversión a finales de mayo se comprobó que la migración de la base de datos fue un éxito. Este riesgo pasó a categoría “No probable”.

7.4.3.5 Documentación (R20)

Este riesgo no lo consideramos en un principio, pero a partir de las evaluaciones de marzo y de la reunión con el tutor, es que pasó a ser uno de los principales riesgos a tenerse en cuenta. En las evaluaciones de marzo redefinimos los tiempos y pasó a ser prioridad del proyecto. En la evaluación de mayo continuaba siendo un punto crítico del proyecto.

Si bien sabemos que era una de nuestras prioridades, seguimos atrasados en la evaluación de junio. El proyecto es una prioridad para UTE, nosotros nos comprometimos a entregarlo en tiempo y forma, pero debido al atraso en la programación y luego en las pruebas, nos insumió muchísimo más tiempo del estimado al inicio.

Se debía optar por un producto mediocre y una excelente documentación o por un mejor producto. Cuando el sistema esté operativo no pueden haber errores, ya que las pérdidas económicas podrían ser muy grandes para el Ente.

7.4.4 Conclusiones y lecciones aprendidas

Uno de los riesgos no se pudo controlar. Si bien teníamos planes de mitigación y contingencia, que se llevaron a cabo, al final se nos escapó de las manos. Al inicio del proyecto estábamos concientes de que algo así podía ocurrir, pero en el correr del proyecto se veían avances en la infraestructura y todo estaba dado para que sí se construyera el centro de cómputos. Lamentablemente, en mayo nos informaron que hubo cambios y que los equipos se tardarían en llegar. Se intentó, por todos lados, que los equipos llegaran antes, pero no fue posible. Si bien es un interés de UTE, para ellos es lo mismo que se implemente en agosto o en setiembre.

Independientemente de esto, la gestión de riesgo ayudó a identificar los riesgos y a tenerlos presente, buscando planes que permitan minimizar las probabilidades de ocurrencia y las consecuencias de los eventos adversos.

Creemos que la planificación fue buena, nos permitió, por ejemplo, evaluar en el momento de las dificultades técnicas qué medidas tomar para resolverlas y mantener el cronograma dentro de los tiempos esperados, nos ayudó a mantenernos enfocados en la culminación exitosa del proyecto.

Como lección aprendida concluimos que se deben controlar todos los factores externos, pero que existen ciertos tipos de riesgos que van más allá de las medidas de contingencia que se tomen; principalmente con un cliente como UTE, al cual dos meses de atraso no les significa mayores problemas.

8 Gestión de Lanzamiento

Nos basamos en ITIL V3, libro Transición del Servicio para la planificación de la gestión de lanzamiento. [13]

Según ITIL V3, el propósito del proceso de administración de lanzamiento y despliegue es planificar, programar y controlar la creación, prueba y despliegue de versiones, implementando una nueva funcionalidad requerida por el negocio mientras se protege la integridad de los servicios existentes.

8.1 Finalidad y Objetivos:

Los objetivos de la gestión de lanzamiento y despliegue son:

- Definir y acordar planes de gestión de lanzamiento e implementación con clientes e interesados.
- Crear y probar paquetes de lanzamiento que consisten en elementos de configuración relacionados, que son compatibles entre sí.
- Asegurar de que la integridad de un paquete de versión y sus componentes y de que todos los paquetes de lanzamiento estén almacenados en un DML y grabados con precisión en el CMS.
- Implementar paquetes de lanzamiento desde el DML al entorno en vivo siguiendo un plan y programa acordados.
- Asegurese de que todos los paquetes para la puesta en producción puedan rastrearse, instalarse, probarse, verificarse y / o desinstalarse o retirarse si es necesario.
- Asegurar que un servicio nuevo o modificado y sus sistemas, tecnología y organización habilitadores sean capaces de entregar la utilidad y la garantía acordadas.
- Registrar y gestionar las desviaciones, los riesgos y las cuestiones relacionadas con el servicio nuevo o modificado y adoptar las medidas correctivas necesarias.
- Asegurar que haya una transferencia de conocimiento que permita a los clientes y usuarios optimizar su uso del servicio para apoyar sus actividades empresariales.
- Asegurar que las habilidades y conocimientos se transfieren a las funciones de operación del servicio para brindar, y mantener el servicio de manera efectiva y eficiente de acuerdo con las garantías y niveles de servicio requeridos.

8.2 Políticas de gestión de lanzamiento e implementación

Se establecieron políticas de gestión de lanzamiento y despliegue para ayudar a la organización a lograr el equilibrio correcto entre el costo, la estabilidad del servicio y la agilidad. Las mismas se presentan en Anexo 23 – Políticas de Lanzamiento.

En el caso del servicio de UTE lo más importante es la estabilidad del servicio, por lo que el tiempo de pruebas fue mayor.

Las políticas de gestión de lanzamiento y despliegue deben ayudar al personal de administración de lanzamiento y despliegue a tomar decisiones que apoyen los objetivos generales del negocio.

Estas políticas pueden establecerse en un nivel global, en nuestro caso la migración es un lineamiento de la alta gerencia de UTE (una de las metas trazadas para este año), dada la importante reducción de costos en licenciamiento y en funcionamiento general, ventajas que ya fueron enumeradas al inicio del documento.

8.2.1 Componentes.

A continuación el esquema de desarrollo del proceso Migración SCADA:



El proceso migración consiste en construcción de la pieza de software para migrar la parte gráfica, integrar con las soluciones Migradas por el departamento de TI de UTE y

Figura 8.1: Esquema desarrollo proyecto Migración.

realizar las diferentes pruebas definidas. El cliente se encarga de migrar la base de datos y el *front end*.

8.2.2 Unidades de lanzamiento

Dentro de la migración gráfica se planificaron dos lanzamientos, divididos por zonas geográficas, que son las zonas de Montevideo y Canelones. El alcance del proyecto final de carrera es la puesta en producción de la migración de Canelones. Montevideo quedaría para una siguiente instancia.

Id.	Nombre de tarea	2016					2017											
		ago.	sep.	oct.	nov.	dic.	ene.	feb.	mar.	abr.	may.	jun.	jul.	ago.	sep.	oct.	nov.	dic.
1	CONSTRUCCION SOFTWARE																	
2	LANZAMIENTO CANELONES																	
3	LANZAMIENTO MONTEVIDEO																	

Figura 8.2: Unidades de lanzamiento.

Luego de la construcción del software se deben migrar uno a uno los unifilares y de cada estación o subestación y aplicarles las pruebas automáticas, de conectividad e integridad, además de todas la pruebas ya mencionadas en el capítulo Gestión de Calidad.

8.2.3 Opciones de implementación y consideraciones.

El diseño definirá el enfoque para la transición del servicio actual al nuevo servicio. El paquete de diseño de servicios define los componentes de diseño de servicios y soluciones que se van a realizar para entregar el servicio requerido.

Las opciones comunes para el lanzamiento y despliegue que se consideran en el diseño del servicio fueron opciones de despliegue "Big Bang" o despliegue en "Fases".

- Opción 'Big Bang': El servicio nuevo o modificado se despliega en todas las áreas de usuario en una sola operación.
- Enfoque en fases: El servicio se despliega inicialmente en una parte de la base de usuarios, partiendo en dos o más partes la base de usuarios. Luego esta operación se repite para la siguiente parte de la base de usuarios a través de un plan de despliegue programado.

El actual servicio del SCADA X corre bajo plataforma Linux, mientras que el servicio SCADA Mirage corre bajo plataforma Windows, por ello se debe bajar un servicio y levantar el nuevo servicio, eventualmente se podría dejar funcionando ambos sistemas de forma parcial, pero para el operador sería una complejidad extra controlar dos sistemas parciales a la vez. Por lo tanto cada uno de los lanzamientos programados, (Canelones y Montevideo) se harán entonces mediante un despliegue "Big Bang".

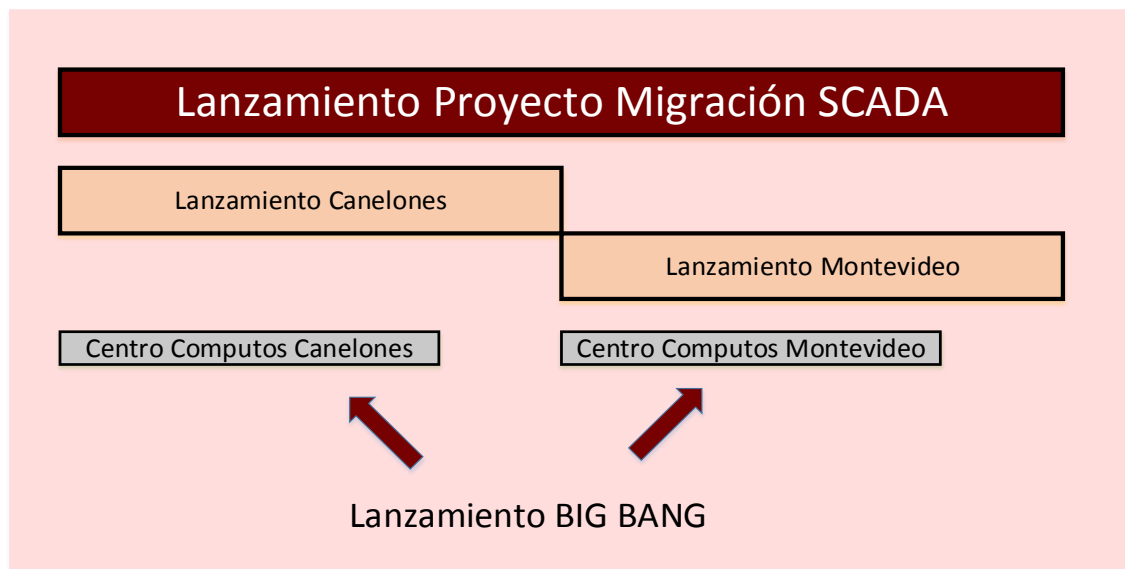


Figura 8.3: Esquema de Implementación BIG BANG.

8.2.4 Modelos de lanzamiento e implementación.

Un servicio puede ser desplegado en el entorno en vivo de varias maneras. El diseño del servicio seleccionará los modelos de lanzamiento y de despliegue más adecuados que incluyan el enfoque, los mecanismos, los procesos, los procedimientos y los recursos necesarios para construir, probar e implementar la puesta en producción a tiempo y dentro del presupuesto.

Para el lanzamiento y despliegue se definieron:

- Estructura de puesta en producción: Se definieron dos paquetes de lanzamiento, que correrán en dos rack de servidores distintos dentro del mismo centros de cómputos, ambos corriendo bajo plataforma Windows, uno para CMD Centro (Canelones y Florida) y el segundo para CMD Montevideo.
- Los criterios de salida y entrada, incluidos los entregables obligatorios y opcionales y la documentación para cada etapa. Ver en Anexo 8.2 - Lista de entregables del Lanzamiento.
- Un entorno controlado para las pruebas de cada lanzamiento.
- Calendarios de lanzamiento y de despliegue.
- Responsables de ejecutar el traspaso y la aceptación de cada puesta en producción.

En la hora del despliegue se tomarán las siguientes consideraciones:

- Verificar que una versión cumple con el paquete de desarrollo del servicio, la arquitectura y las normas relacionadas.
- Garantizar la integridad del hardware y el software durante la instalación, manipulación y puesta en producción.
- Utilizar procedimientos y herramientas de puesta en producción y despliegue estándar.
- Utilizar procedimientos para administrar y controlar componentes durante las actividades de administración de implementación y despliegue, por ejemplo, para verificar los prerrequisitos, requisitos y actividades posteriores a la instalación.
- Documentar los pasos de puesta en producción y administración de despliegue.
- Implementar controles y listas de verificación para asegurar que todas las actividades requeridas se lleven a cabo y que se mantengan registros para apoyar auditorías futuras del proceso.
- Haber definido un procedimiento de rollback, para volver al sistema anterior, ante una eventual falla. Ver Anexo 8.2 – Procedimiento de Rollback.

8.3 Actividades, métodos y técnicas del proceso

Hay cuatro fases para liberar y administrar el despliegue (vea la Figura 9.4):

- Desarrollo y planificación de la implementación: Se crean planes para crear e implementar la versión. Esta fase comienza con la autorización de administración de cambios para planificar un lanzamiento y termina con la autorización de administración de cambios para crear la puesta en producción.
- Creación y prueba de puesta en producción: El paquete de lanzamiento se crea, se prueba y se registra en la biblioteca de medios. Esta fase comienza con la autorización de gestión de cambios para crear el lanzamiento y finaliza con la autorización de administración de cambios para el paquete de lanzamiento. Esta fase solo ocurre una vez para cada puesta en producción.
- Implementación: El paquete de versión se implementa en el entorno en vivo. Esta fase comienza con la autorización de administración de cambios para desplegar el paquete de lanzamiento a uno o más entornos de destino y termina con la transferencia a las funciones de operación de servicio y de inicio soporte vital. Puede haber muchas fases de despliegue separadas para cada lanzamiento, pero en nuestro caso solo habrá una por centro de cómputos.
- Revisar y cerrar: La experiencia y la retroalimentación se capturan, se revisan los objetivos de rendimiento y los logros y se aprenden las lecciones.

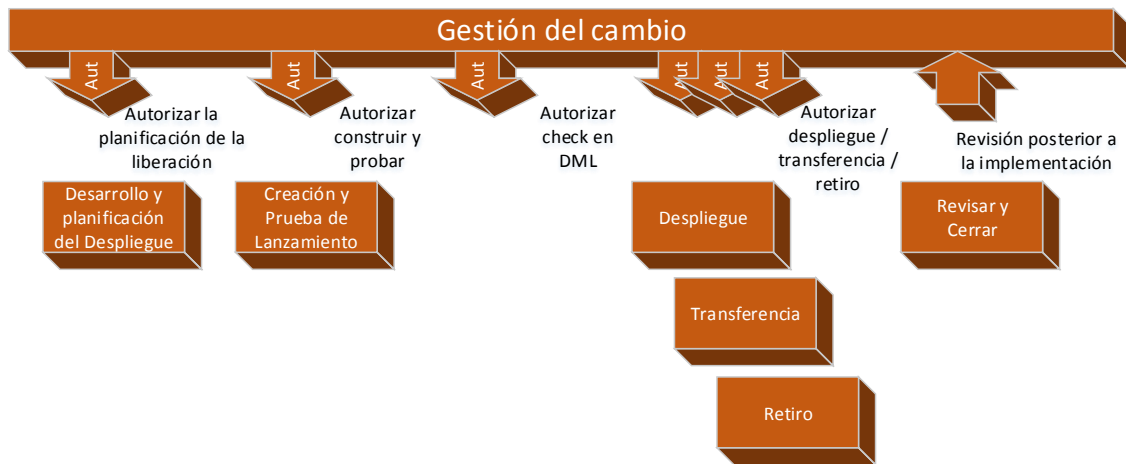


Figura 8.4: Fases de la gestión de lanzamiento y despliegue.

8.3.1 Planificación de lanzamiento y despliegue

La mayor parte de la planificación se lleva a cabo durante la etapa de diseño del servicio. Estas actividades se llevan a cabo como parte del proceso de gestión de lanzamiento y despliegue. Estos planes forman parte de los planes generales de transición de servicios y deben coordinarse con los planes para otras actividades, tales como planes de proyectos, planes de gestión de cambios, etc.

Los planes de lanzamiento y despliegue deben basarse en modelos de servicio y en los requisitos documentados de la utilidad y garantía del servicio, así como en los datos técnicos sobre los componentes que constituyen el nuevo servicio y estarán vinculados al plan general de transición del servicio y adoptarán el modelo de despliegue e implementación seleccionado.

Los planes de lanzamiento y despliegue deben ser autorizados a través de la administración del cambio y deben contener la siguiente información:

- Alcance y contenido del comunicado.
- Evaluación del riesgo y perfil de riesgo para el lanzamiento.
- Organizaciones e interesados afectados por el lanzamiento.
- Las partes interesadas que pueden autorizar la solicitud de cambio para cada etapa del comunicado.
- Equipo responsable del lanzamiento.
- Calendario de implementación para el lanzamiento.
- Estrategia de entrega y despliegue.
- Recursos para la creación, prueba y despliegue de lanzamientos, y para soporte de vida temprana.
- Procedimiento de Rollback definido.

Para efectuar el lanzamiento, la comisión de transición del servicio es responsable de:

- Construir, probar y mantener los ambientes controlados antes del despliegue.

- Verificar que la compilación y prueba deben funcionar con la validación y las pruebas de servicio y garantizar que se llevaron a cabo con éxito todos los planes de prueba.
- El informe de evaluación no muestra riesgos inaceptables.
- Verificar que fuera autorizado por la administración del cambio.

La estrategia de prueba define el enfoque general para validar y probar el servicio. Incluye la organización de actividades y recursos de validación y prueba de servicios y puede aplicarse a toda la organización, un conjunto de servicios o un servicio individual.

Los niveles típicos de configuración para la construcción y las pruebas se muestran en la Tabla 8.1.

Nivel	Requisitos y Construir / Validación y pruebas diseño entregar	
Nivel 1 Necesidades del cliente / negocio	Definición estructurada de los requisitos del contrato	Contrato de cliente Prueba de servicio y evaluación El nuevo servicio ya es utilizado en otros CMD de UTE, por lo que ya sabemos que satisface las necesidades técnicas de UTE.
Nivel 2 Requisitos de servicio	Requerimientos de servicio que se remontan a los requisitos del contrato	Capacidad recursos y los requisitos del servicio Prueba de servicio Prueba que se cumplen los criterios de aceptación del servicio. Incluye la validación del desempeño del servicio en relación con los requisitos de nivel de servicio.
Solución de servicio de nivel 3	Modelo de servicio, entornos de servicio	Solución / sistema requerido para entregar la capacidad de servicio. Prueba de preparación operativa del servicio Evalúa la integración y el funcionamiento de la capacidad y los recursos del servicio. Verifica que la organización de despliegue de destino y las personas están preparadas para desplegar y operar el servicio nuevo en producción. Las funcionalidades son similares a la del actual sistema, de todas formas UTE ha planificado realizar cursos de capacitación.
Lanzamiento de servicio de nivel 4		Puesta en producción del paquete Prueba de puesta en producción del servicio Prueba que los componentes de servicio se pueden desplegar correctamente y que la versión puede instalarse, construirse y probarse en los entornos de destino. Se creará un ambiente para pruebas

			corriendo en la nueva plataforma para minimizar riesgos.
Nivel 5 Componentes y ensamblajes	Especificación de prueba de componentes y ensamblajes	Componente o conjunto de componentes	Prueba de componentes y ensamblaje Se realizarán pruebas unitarias a cada elemento y luego se realizará pruebas de ensamblaje de todos los elementos, además se realizaran comparaciones de resultados con el actual sistema.

Tabla 8-1: Niveles de configuración para construir y probar.

Hay muchas consideraciones de planificación que deben considerarse. En la tabla 8.2 se pueden responder varias preguntas que surgen en los despliegues:

Pregunta de implementación	Respuestas
¿Qué necesita ser desplegado?	Se desplegará el sistema SCADA Mirage corriendo bajo plataforma de Windows, para eso se debe migrar del sistema SCADA anterior.
¿Quiénes son los usuarios?	Los usuarios ya manejan localmente en alguna estación el SCADA Mirage, por lo que de ser necesarias capacitaciones las mismas serán breves y ya están previstas por parte de UTE.
¿Hay consideraciones diferentes según ubicación?	No, son las mismas consideraciones tanto para Canelones como Montevideo.
¿Dónde están los usuarios?	Los usuarios operan desde el CMD de cada región, la infraestructura del nuevo centro está a cargo del departamento de TI de UTE.
¿Quién más necesita estar preparado con suficiente antelación?	Además de los usuarios del CMD también deben estar preparados las unidades Telecomunicaciones, unidades de Hardware y unidades Software.
¿Cuándo es necesario completar el despliegue?	No existen inconvenientes de horarios para efectuar el despliegue, pero se realizará por la mañana en horario comercial, que es cuando hay más operarios en las estaciones y sub estaciones, por cualquier medida de contingencia necesaria, ya que se trata de un servicio básico de la población.
¿Por qué está ocurriendo el despliegue?	El nuevo sistema presenta mejoras, es más usable y además tiene costos de licenciamiento muchos más económicos y todas las ventajas ya nombradas en capítulo 2.2.3 Principales beneficios.
¿Cuáles son los factores críticos de éxito y los criterios de salida?	Se planifica dejar el servidor actual LINUX durante 2 meses para un eventual <i>rollback</i> . En el día del lanzamiento se evaluarán los siguientes criterios para saber si fue exitosa la migración:

	• Se recibirán medidas eléctricas, como corrientes, voltajes y potencias. • No se encenderán alarmas falsas. • Y principalmente no se verá interrumpido el servicio de energía eléctrica a la población. Luego a demanda se probaran las aperturas y cierres a distancia. Una vez transcurrido los dos meses sin eventuales fallas se dará por finalizado el despliegue.
¿Cuál es la capacidad actual del proveedor de servicios?	Actualmente tanto el CMD de Montevideo como de Canelones operan con un sistema SCADA que corre bajo plataforma de Linux, con altos costos en licenciamientos.

Tabla 8-2: Preguntas que se deben responder al planificar la implementación.

8.3.2 Creación y prueba de la versión

Los aspectos clave que se deben gestionar durante las actividades para crear y probar un servicio es el correcto uso de los entornos de compilación y prueba y la gestión de las configuraciones durante las actividades de compilación y prueba, por ejemplo, control de versiones, gestión de líneas de base, control de entradas y salidas desde una etapa de construcción o prueba.

La versión definitiva del paquete de lanzamiento (autorizado por la administración de cambios) debe colocarse en la biblioteca de medios. El paquete de lanzamiento siempre debe tomarse de la biblioteca para desplegarlo para la preparación de la operación de servicio, la aceptación del servicio y los entornos en vivo.

8.3.2.1 Puesta en producción y crear documentación

Procedimientos, plantillas y guías deben ser usados para permitir que el equipo de puesta en producción tome los activos de servicio y los productos de proveedores internos y externos y construya un paquete de lanzamiento integrado de manera eficiente y efectiva.

Los procedimientos y documentos para comprar, distribuir, instalar, mover y controlar activos y componentes que son relevantes para adquirir, construir y probar un lanzamiento.

En este nuestro caso UTE se encarga de:

- Construir el nuevo centro de cómputos, se encargó de toda la gestión, comprar, instalación de hardware y software, licenciamiento de software, darles los soportes de redes, políticas y procedimientos de seguridad, etc.
- Gestión de configuraciones de servicios e infraestructuras.
- Configuración de enlaces.
- Desmantelamiento de instalaciones e infraestructuras.

Nosotros nos encargamos de los procedimientos para:

- Entornos de pruebas y puesta en marcha.
- Validación y pruebas de servicio.
- Gestión del cambio.

Para la puesta en producción previamente se instalará y se probará el sistema SCADA Mirage en el nuevo centro de cómputos, examinando que funcionen correctamente las conexiones remotas y la base de datos. Para que luego, en el momento del lanzamiento únicamente se deban cargar los unifilares migrados.

La solución propuesta debe documentarse para permitir que los conocimientos recopilados durante la fase de construcción y prueba sean las funciones de operación de servicio y la mejora continua del servicio que se mantendrá para futuras versiones.

La documentación de compilación y puesta en producción debe incluir:

- Planes de compilación, instalación y prueba.
- Detalles sobre cómo monitorear y verificar la calidad de la versión y cómo reconocer y reaccionar a los problemas.
- Los procesos y procedimientos automatizados o manuales necesarios para distribuir, desplegar y puesta en producción en el entorno de destino (o eliminarlo según sea necesario).
- Procedimientos para cancelar las unidades de lanzamiento o corregir un cambio en caso de fallo de la puesta en producción (*rollback*).
- Procedimientos para el seguimiento y gestión de licencias de software y derechos digitales.

8.3.2.2 Pruebas de servicio

La prueba tiene como objetivo generar confianza en la capacidad de servicio antes de la aceptación final. Se basará en la estrategia de prueba y en el modelo para el servicio que se está cambiando.

Los criterios de prueba reflejan las condiciones anticipadas en las que se espera que el servicio funcione y proporcione beneficios.

Las pruebas de preparación para la operación de servicio garantizan que un servicio y su infraestructura subyacente de aplicaciones y tecnología se puedan transferir al entorno en vivo de manera controlada.

La prueba de preparación operativa del servicio tiene como objetivo:

- Determinar si un servicio y sus activos de servicio subyacentes pueden liberarse en el entorno en vivo, por primera vez y para implementaciones posteriores.

- Asegurarse de que los procesos de negocio, los clientes, los usuarios y las interfaces de proveedores de servicios son capaces de utilizar el servicio correctamente.
- Asegurarse de que los equipos de servicio son capaces de operar el servicio y utilizar los sistemas de gestión de servicios correctamente.

Las pruebas que se realizan como parte de la prueba de preparación operativa del servicio incluyen:

- Prueba de preparación del despliegue: Para garantizar que los procesos, procedimientos y sistemas de despliegue puedan implementar, instalar, encargar y dismantelar el paquete de lanzamiento y el servicio nuevo o modificado resultante en el entorno de destino.
- Prueba de funcionamiento del servicio: Para asegurar que los equipos de servicio pueden operar el servicio en uso.
- Prueba de nivel de servicio: para garantizar que el servicio nuevo o modificado ofrezca los requisitos de nivel de servicio.
- Prueba de usuario: Para garantizar que los usuarios pueden acceder y utilizar el servicio nuevo.
- Prueba de verificación de implementación: Para asegurarse de que la capacidad de servicio se ha implementado correctamente.

8.3.3 Despliegue

Las actividades de planificación y preparación acondicionan al grupo para el despliegue. Esta es una oportunidad para adecuar a la organización y a la gente para el cambio organizacional.

A la hora de planificar y preparar el grupo o entorno de despliegue se cuenta con las siguientes ventajas:

- Nosotros como agentes de implementación tenemos suficiente confianza en la versión de servicio para implementarla, además de que estamos muy comprometidos con la implementación.
- La alta gerencia de UTE está alineada con la implementación, siendo uno de los principales interesados en que se realice correctamente. Los usuarios aceptan la implementación, ya que el nuevo sistema le proporciona una mejor usabilidad. Y al departamento de TI le simplifica el alta de nuevos servicios a futuro, así como el mantenimiento de los actuales.

La preparación para el despliegue incluye evaluar la disponibilidad de cada grupo de despliegue para recibir e implementar un paquete de lanzamiento, planificar las actividades necesarias para desplegar el servicio.

La evaluación de la preparación para un grupo de despliegue identifica los problemas y riesgos que pueden afectar a la implementación.

Los tipos de riesgo incluyen:

- Falta de recursos internos dedicados.
- Falta de capacitación y habilidades.
- Cambio no planificado o tardío de los requisitos.
- Aspectos legales contractuales con el actual sistema.
- Establecimiento de contratos nuevos, licencias y derechos digitales.
- Capacidad y recursos actuales de administración de servicios, incluyendo:
 - Diferencias con respecto a los requisitos previos para el despliegue. Ejemplos: Acuerdos de licencia inadecuados, ancho de banda de la red.
 - Operaciones actuales y recursos de soporte, por ejemplo, herramientas, personas.
 - Soporte de recursos y cargas de trabajo ya que puede haber un aumento significativo en el número de incidentes, desviándose los recursos para la gestión de incidentes, problemas y arreglos.
- Infraestructura e instalaciones:
 - No se modificara a los usuarios finales, si bien se instala un nuevo centro de cómputos el acceso al mismo es remoto.
 - En cuanto al equipo de TI, sustituirá un centro de cómputos antiguo por otro moderno, lo que le va implicar menos mantenimiento.
 - Si bien al inicio tendrán un trabajo adicional en establecer:
 - Requisitos de espacio y capacidad de los equipos informáticos.
 - Requisitos de potencia y clasificación de los interruptores, fuentes de alimentación ininterrumpida y cargas de generador.
 - Requisitos de temperatura y humedad, salidas de calor y requisitos de aire acondicionado.
 - Requisitos de cableado de red y banda ancha.

Rollback

Se puede tomar una decisión de vuelta atrás en el lanzamiento debido a que:

- Un paso de la implementación falla o se comporta de manera inesperada.
- La verificación muestra que el despliegue no ha tenido éxito.

En estos casos se restaura la base de datos y el *front end* al estado anterior, el centro de cómputos de Linux se mantendrá activo durante 2 meses, únicamente se deberá bajar el enlace desde Windows y volverlo a subir en Linux.

8.3.4 Revisar y Cerrar

Al finalizar el despliegue se deberán realizar las siguientes actividades:

- Capturar experiencias y comentarios sobre la satisfacción de usuarios, por ejemplo, mediante encuestas de retroalimentación.
- Resaltar los criterios de calidad que no se cumplieron.
- Comprobar que todas las acciones, correcciones necesarias y cambios estén completos.
- Comprobar que los problemas, los errores conocidos y las soluciones temporales sean documentados y aceptados por los clientes / empresas y / o proveedores.
- Revisar el registro de riesgos e identificar los elementos que afectan la operación y el soporte del servicio. Abordar los riesgos o acordar una acción, como mover los riesgos al registro de riesgos de transición del servicio.

Cada despliegue debe considerar si se han detectado problemas relevantes que se deben corregir, tales como:

- Errores en los procedimientos detectados.
- Datos o información incorrecta en los registros pertinentes.
- Incidentes y problemas causados por la implementación.
- Dificultades para actualizar los registros.

Una revisión posterior a la implementación de cada despliegue se realiza a través de la administración del cambio.

8.3.4.1 Revisar y cerrar una transición de servicio

Para consumir que se ha completado una transición de servicio, debería realizarse una revisión formal apropiada a la escala y magnitud del cambio. Una revisión de la transición del servicio debe incluir:

- Comprobar que todas las actividades de transición se han completado, por ejemplo: documentación e información, son capturados, actualizados, protegidos y archivados.
- Comprobación de la captura de métricas precisas. La evaluación formal de la versión de servicio utiliza los resultados del despliegue. Esta evaluación de cambios comprueba el rendimiento real y los resultados del servicio nuevo o cambiado frente al rendimiento y los resultados previstos, es decir, el servicio esperado por el usuario o el cliente.

8.4 Conclusiones

La Gestión de Lanzamiento siguiendo los lineamientos de ITIL fue muy importante en el Proyecto.

Si bien no se pudo realizar el despliegue “*Big Bang*” del CMD Canelones se considera exitosa la gestión de lanzamiento, debido a que:

- Se probaron todos los unifilares en el entorno creado para las pruebas.
- Entró en producción un conjunto de unifilares que pertenecen a la gerencia de transmisión y que son únicamente visualizados por gerencia distribución (CMD), sin la posibilidad de efectuar maniobras. Ver alcance gerencias en Capítulo 1.2 UTE como empresa y la utilización del SCADA.

9 Gestión del cambio

La definición de ITIL de un cambio es "agregar, modificar o eliminar cualquier cosa que pueda afectar a los servicios de TI"

Según ITIL V3, libro *Service Transition* [13], la gestión del cambio es una respuesta adecuada a todas las solicitudes de cambio, considerando un enfoque en:

- La evaluación del riesgo.
- La continuidad del negocio.
- El impacto del cambio.
- Las necesidades de recursos.
- La autorización de modificación.
- Y por último, especialmente, el beneficio comercial realizable.

La evaluación de riesgos debe considerar el riesgo de no implementar el cambio así como cualquier riesgo que el cambio pueda introducir. Este enfoque considerado es esencial para mantener el equilibrio necesario entre la necesidad de cambio y el impacto de ese cambio.

El propósito del proceso de gestión del cambio es controlar el ciclo de vida de todos los cambios, permitiendo realizar cambios beneficiosos con la mínima interrupción de los servicios de TI.

Los objetivos de la gestión del cambio son:

-  Responder a los requisitos comerciales cambiantes del cliente al tiempo que se maximiza el valor y se reduce incidentes, interrupciones y re-trabajo.
-  Responder a las solicitudes de cambio de negocio y TI que alinearán los servicios con las necesidades del negocio.
-  Asegurar que los cambios sean registrados y evaluados, y que los cambios autorizados sean priorizados, planificados, probados, implementados, documentados y revisados de manera controlada.
-  Asegurarse de que todos los cambios en los elementos de configuración se registran en el sistema de gestión de la configuración.
-  Optimizar el riesgo general del negocio, a menudo es correcto minimizar el riesgo del negocio, pero a veces es apropiado aceptar un riesgo a sabiendas del beneficio potencial.

Para ilustrar el proceso de cambios se diseñaron los siguientes diagramas:

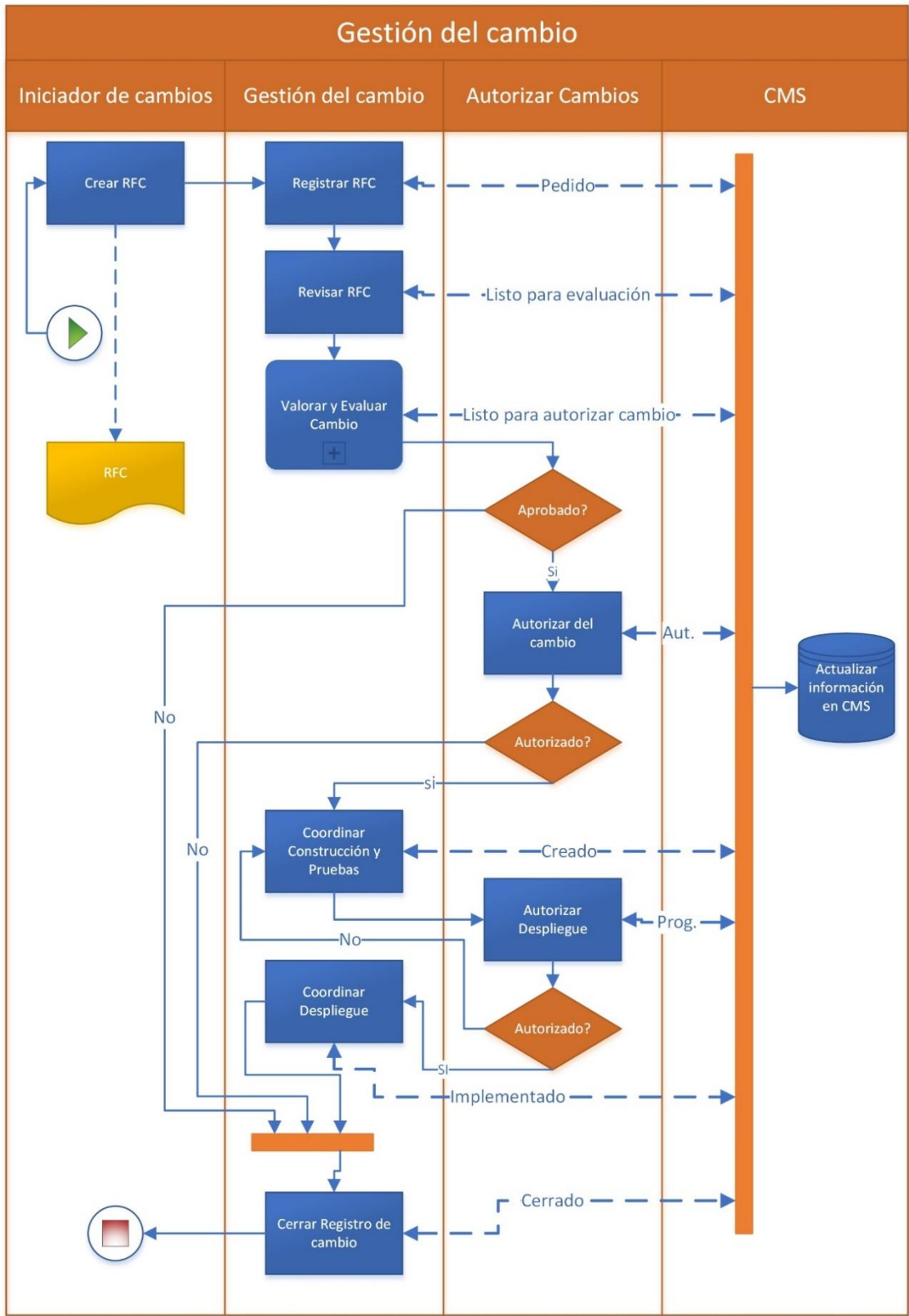


Figura 9.1: Proceso gestión del cambio

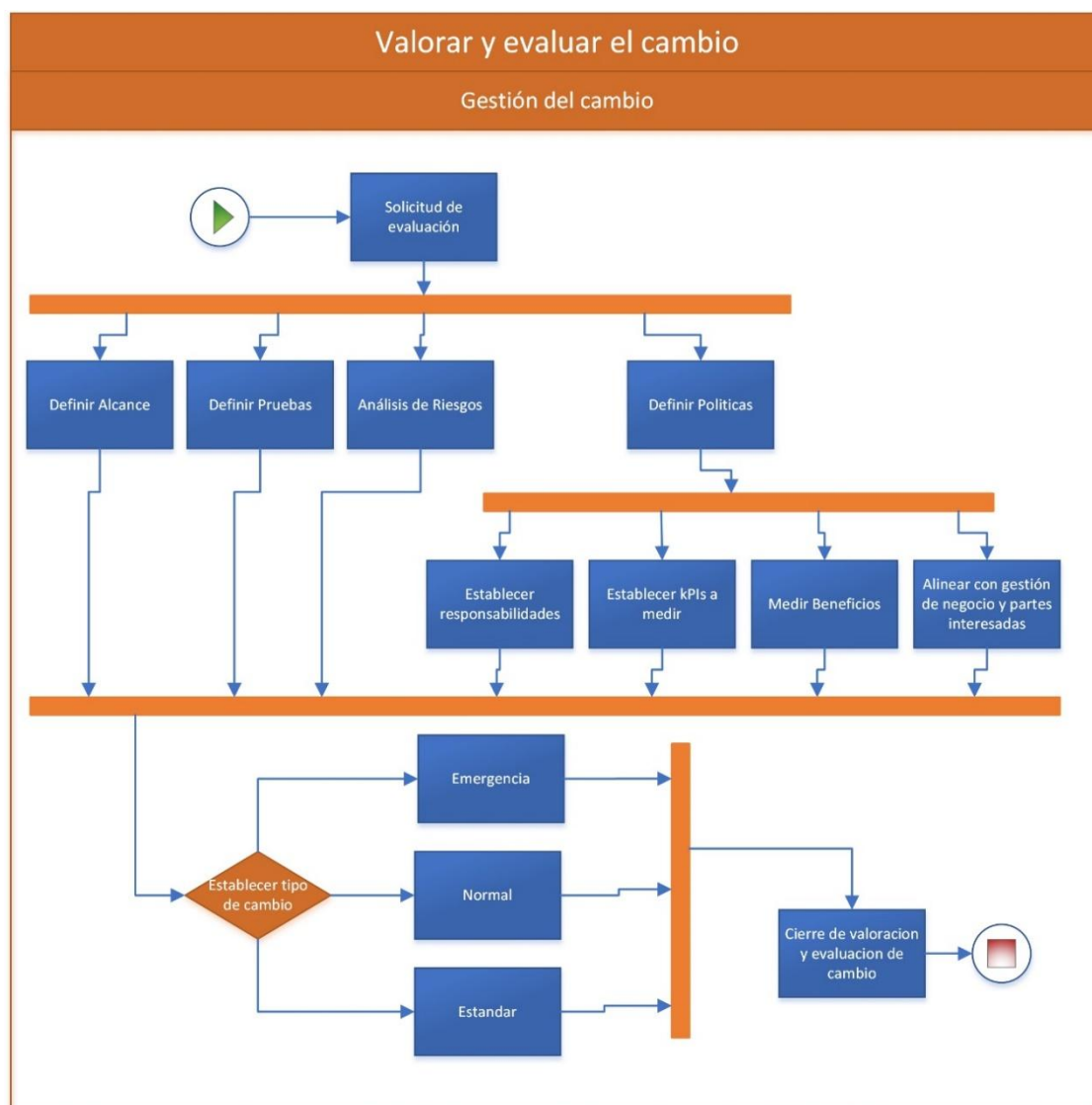


Figura 9.2: Subproceso Valorar y Evaluar el cambio.

Descripción de tareas del proceso de gestión de cambio:

Crear el cambio:

El cambio es planteado por una solicitud del iniciador de cambios, que puede ser un individuo o grupo organizacional que requiere el cambio. Por ejemplo, puede tratarse de una unidad de negocio que requiere instalaciones adicionales o personal de gestión de problemas que instigue una resolución de errores. Para un cambio importante con implicaciones organizacionales y / o financieras significativas se requiere una propuesta de trabajo.

El nivel de detalle depende del tamaño y del impacto del cambio. Se registra cierta información cuando se inicia el documento y luego se puede actualizar la información a medida que el documento de modificación avanza a lo largo de su ciclo de vida.

Los detalles del cambio y las acciones pueden registrarse en otros documentos y referenciarse desde el RFC, por ejemplo, el caso de negocios o el informe de evaluación de impacto.

El registro de cambios contiene el historial completo del cambio, incorporando información del RFC y posteriormente registrando parámetros acordados tales como prioridad y autorización, implementación e información de revisión. Puede haber muchos tipos diferentes de registros de cambios usados para registrar diferentes tipos de cambio. La documentación debe definirse durante la fase de diseño y planificación del proceso.

A medida que un cambio avanza a lo largo de su ciclo de vida, el documento de cambio, los registros relacionados (tales como órdenes de trabajo) y los elementos de configuración relacionados se actualizan en el CMS, de modo que haya visibilidad de su estado y del cumplimiento del proceso. Las estimaciones y los recursos reales, los costos y los resultados (éxito o fracaso) se registran para permitir la presentación de informes de gestión.

Registrar RFC

RFC podrían ser presentados en formularios en papel, a través de correo electrónico o mediante una interfaz basada en web, etc.

Todos los RFC recibidos deben ser registrados y asignados a un número de identificación (en secuencia cronológica). Cuando las solicitudes de cambio se presenten en respuesta a un desencadenante como un registro de problemas (PR), se registrará el número de referencia del documento de disparo se retenga para proporcionar trazabilidad.

Todas las acciones serán registradas, tal como se llevan a cabo, dentro del registro de gestión de cambios. Si esto no es posible por cualquier razón, entonces deben ser grabados manualmente para su inclusión en la próxima oportunidad posible.

Los procedimientos especificarán los niveles de acceso y quién tiene acceso al sistema de registro. Mientras que cualquier personal autorizado puede crear un cambio, o agregar informes de progreso a él, solo el personal de gestión de cambio tendrá permiso para cerrar un cambio.

Valorar y evaluar cambio:

Como bien lo dice el nombre, este es un subproceso para evaluar y darle un valor al cambio. En este proceso se define:

- El alcance,
- las pruebas,
- el análisis de riesgos,
- las políticas, en las cuales se establecen:
 - responsabilidades

- KPI a medir
- Beneficios
- Alinear con gestión de negocio e interesados en el cambio.

Una vez definidos todos los puntos se elige el tipo de cambio:

- Cambio estándar: Un cambio preautorizado de bajo riesgo, relativamente común y que sigue un procedimiento o una instrucción de trabajo. Un ejemplo podría ser incluir una actualización de un PC en una subestación local para que pueda correr SCADA Mirage.
- Cambio de emergencia: Un cambio que debe implementarse lo antes posible, por ejemplo, para resolver un incidente importante.
- Cambio normal: Cualquier cambio de servicio que no sea un cambio estándar o un cambio de emergencia.

Y se finaliza con una evaluación del proceso.

Autorizar el cambio:

Se obtiene autorización formal para cada cambio. Los niveles de autorización para un tipo particular de cambio deben juzgarse por el tipo, el tamaño, el riesgo y el impacto comercial potencial del cambio. Para nuestro proyecto se definieron tres tipos de autoridades de cambio:

- ✚ Junta directiva de UTE: Por ejemplo, si surge una necesidad externa generando un nuevo requerimiento que impacta en el funcionamiento de todas las subestaciones, dicho cambio debe ser aprobado por una Junta directiva de UTE, presentando la autorización formal.
- ✚ CAB (Comité Asesor de Cambio): Por ejemplo, si el sistema necesita un ajuste menor, como corregir la conectividad en un tipo de elemento dado, los mismos son aprobados un comité asesor de cambios (CAB) que en este caso está formado por nosotros y la jefa directa de Andrés.
- ✚ Y por último está el administrador de cambios: Los cambios estéticos que no implican cambios en funcionalidades, pero que son costos en tiempo serán evaluados por el administrador de los cambios, en este caso Renato.

Coordinar construcción y pruebas:

Si este cambio es parte de una puesta en producción, el trabajo de versionar el cambio, crear y probar esta es coordinado por el proceso de administración de lanzamiento y despliegue. Para cambios simples que no forman parte de una versión, el proceso de gestión de cambios coordinará este trabajo.

Los cambios autorizados deben ser contruidos y registrados para que luego se pueda rastrear, por ejemplo, utilizando órdenes de trabajo.

La gerencia del cambio tiene el papel de la supervisión para asegurarse de que todos los cambios que puedan ser se prueban a fondo. Todos los casos que impliquen cambios que no han sido completamente probados, se debe tener especial cuidado durante la implementación.

Las pruebas pueden continuar en paralelo con la migración anticipada en tiempo real de un servicio, buscando situaciones inusuales, inesperadas o futuras para que se puedan tomar medidas correctivas adicionales antes de que se generen errores en la operación en vivo.

Autorizar despliegue:

El diseño, la construcción y las pruebas del cambio deben evaluarse para asegurar que los riesgos se han gestionado. Para cambios significativos, se recibirá un informe de evaluación intermedio del proceso de evaluación del cambio; Para cambios más pequeños el proceso de gestión del cambio llevará a cabo verificaciones adecuadas. El resultado de esta evaluación debe ser pasado a la autoridad de cambio para la autorización formal de la implementación.

Si no se da la autorización en esta etapa, la autoridad de cambio puede solicitar modificaciones en el diseño o en la planificación de implementación. Esto puede conducir a un enfoque iterativo en el que se realizan las etapas de creación y prueba de cambios y de autorización varias veces hasta que se cumple la autoridad de cambio.

Coordinar despliegue:

El trabajo de desplegar una puesta en producción es parte del proceso de administración de implementación. Para cambios simples que no forman parte de una versión, el proceso de gestión de cambios coordinará esta actividad.

La dirección del cambio tiene la responsabilidad de asegurar que los cambios se implementen según lo programado. Esta función se comparte con la gestión de lanzamiento y despliegue.

Algunos cambios, especialmente los que están sujetos a la administración de lanzamiento y despliegue, pueden tener varias etapas de implementación. Cada despliegue debe ser autorizado por una autoridad de cambio apropiada.

Cerrar registro de cambio:

Antes de que se cierre el cambio, se debe realizar una evaluación para garantizar que no hay riesgos inaceptables. Para los cambios significativos se recibirá un informe de evaluación del proceso de evaluación del cambio; para cambios más pequeños el proceso de gestión del cambio llevará a cabo verificaciones adecuadas. Si la evaluación sugiere que el rendimiento real está creando riesgos inaceptables, se le informará a la autoridad de cambio de que es necesario tomar nuevas medidas.

Si la evaluación muestra que el cambio es aceptable, debe presentarse como un cambio completo para el acuerdo de las partes interesadas (incluyendo el cierre de incidentes, problemas o errores conocidos que el cambio haya resuelto). Claramente, para los cambios importantes habrá más participación de la junta directiva de UTE y de las partes interesadas durante todo el proceso.

La evaluación también debe incluir cualquier incidente que surja como resultado del cambio (si se conoce en esta etapa).

Una revisión de cambios, es para confirmar que el cambio ha cumplido sus objetivos, que el iniciador y las partes interesadas están satisfechos con los resultados y que no se han producido efectos secundarios inesperados. Las lecciones aprendidas deben ser retroalimentadas en los cambios futuros.

Cuando un cambio no ha logrado sus objetivos, la dirección del cambio (o la autoridad del cambio) debería decidir qué acción de seguimiento es necesaria, lo que podría implicar elevar un RFC revisado. Si la revisión es satisfactoria o el cambio original es abandonado (por ejemplo, que las circunstancias que requirieron que el cambio ya no esté vigentes y el requisito desaparezca), el cambio debería cerrarse formalmente en el sistema de registro.

10 Conclusiones

10.1 Conclusiones del proyecto

La primera conclusión, y la más importante para nosotros y para el cliente, es que el proyecto resultó exitoso, logrando cumplir con las expectativas de desarrollar un proyecto de índole académico con aplicación a un caso real.

En deuda nos quedó la entrada en producción dentro del año académico, simplemente por adquisición de infraestructura ajena a nosotros que ya fue explicado en la gestión de riesgos. De todas formas, están operativas estaciones de transmisión, en las cuales se reciben medidas y estados.

A continuación exponemos los diferentes criterios por los que se ha considerado un proyecto exitoso:

-  Haber aplicado los diferentes conocimientos adquiridos a lo largo de la carrera, cada estudiante cumplió con las diferentes responsabilidades relacionadas a su rol, para poder lograr un producto final que satisface a las necesidades reales.
-  Cumplimiento del cronograma, se terminó a tiempo, si bien hubo desviaciones de cronograma, los mismos estuvieron controlados.
-  La gestión un proyecto que supera ampliamente la complejidad de un trabajo obligatorio, además con aplicación real y bajo estándares de calidad en producto y gestión exigentes.
-  Las enseñanzas que nos dejó la oportunidad de trabajar en un proyecto de esta magnitud. Hubo un ida y vuelta de consultas constante con técnicos especializados, de cuyo intercambio de opiniones ganamos madurez profesional.
-  Continúa coordinación con diferentes áreas del cliente en diferentes gestiones, toda la coordinación de pruebas en campo fue muy enriquecedora, cada vez que se debía coordinar con cuatro áreas diferentes.
-  También se evalúa como favorable la parte negativa del proyecto. Haber sobrestimado la fase 0, en las investigaciones y en los planes de gestión, al igual que hicimos con el esfuerzo requerido para desarrollar la pieza de software, nos obligó a gestionar mejor la desviación de dichos tiempos. Tuvimos que dedicar muchas horas extras en estos puntos para poder cumplir el cronograma.

Luego de haber culminado este proyecto estuvimos de acuerdo que era un proyecto para realizarlo en tres estudiantes.

10.2 Conclusiones de equipo de trabajo

La sinergia entre nosotros, manteniendo siempre una comunicación fluida nos permitió lograr el éxito, en un proyecto tan desafiante.

Llevamos adelante la gestión interna de tareas de forma exitosa, y la clave para lograrlo fue atacar los problemas en conjunto, esto nos permitió lograr una muy buena organización del proyecto.

Desde el punto de vista personal, se adquirieron conocimientos en las herramientas y tecnologías de gran valor para abrirnos a un campo de desarrollo en la industria, en un área tan importante como es el monitoreo y control de diferentes procesos de producción. Además de las nuevas tecnologías, gracias al tutor, profundizamos conocimientos sobre el marco de trabajo ITIL, que fue excelente herramienta a la hora de planificar el lanzamiento del proyecto.

10.3 Conclusiones de producto

Para evaluar el producto, vamos a presentar algunas métricas.

Para cada diagrama se ha generado archivos XML con las siguientes cantidades de líneas:

Tipos Instalaciones	Cantidades	Cantidad líneas XML	Total líneas
Estaciones Complejas	10	50000	500.000
Estaciones Regulares	60	15000	900.000
Sub Estación	19	10000	190.000
Puestos de Conexión	19	5000	95.000
Puntos Intermedios	131	3000	393.000
Totales	239		2.078.000

Tabla 10-1: Cantidad de líneas XML

Se ha obtenido de manera automática, por intermedio del software más de 2 millones de líneas de código XML.

Se utilizaron 76 clases y algo menos de 7000 líneas de código en el desarrollo del software de conversión.

Se realizaron pruebas en el software y luego en los unifilares, para lograr un producto altamente testeado, cumpliendo los estándares de calidad exigidos por el cliente para que esté habilitado para entrar en producción.

Respaldado por las métricas también concluimos que este trabajo debería ser desarrollado por mínimo tres estudiantes.

La conclusión final es que el producto obtenido fue exitoso.

10.4 Conclusiones de proceso

El proceso en espiral con un ciclo de vida iterativo, nos permitió en cada etapa realizar las pruebas necesarias. Haber desarrollado la aplicación fragmentada en elementos, posibilitó probarlos de manera independiente a todos.

Concluimos que una parte importante del éxito del producto logrado se debe, sin duda, al proceso seleccionado y a las herramientas que este nos brindó, no solo el *testing* a final de cada etapa, sino también la evaluación de riesgos que se debían realizar.

10.5 Lecciones aprendidas

Al finalizar el proyecto las lecciones aprendidas son muchas, hemos adquirido muchísima experiencia que nos aportarán profesionalidad y conocimientos en futuros emprendimientos.

Muchas enseñanzas sin dudas vinieron por la naturaleza crítica de nuestra aplicación, por ejemplo aprendimos a:

-  Gestionar los riesgos periódicamente.
-  Realizar una planificación minuciosa para la puesta en producción bajo un marco de trabajo ordenado y estandarizado como lo es ITIL.
-  Diseñar un sistema de pruebas, que para ello tuvimos que abstraernos del desarrollo de la pieza de software, y diseñar pruebas para el producto obtenido desde una óptica más global.
-  Mantener una comunicación fluida durante el desarrollo.

Creemos que el proceso de Espiral con un ciclo de vida iterativo fue la base del éxito, nos permitió ir probando temprano y gestionando los riesgos en cada fase.

Aprendimos que las pruebas a temprana etapa de desarrollo resultan muy importantes, permitiendo validar el trabajo y evaluar oportunidades de mejora de manera anticipada.

Con el cliente hemos aprendido que la disponibilidad para consultas técnicas y una buena relación son los principales ingredientes para lograr un proyecto exitoso.

Además, hemos asimilado que existen muchas complejidades a la hora de realizar las coordinaciones en conjunto con varias unidades, sobre todo en clientes con las dimensiones de UTE, por ejemplo, al momento de las pruebas se tuvieron que realizar coordinaciones con unidades de software, de hardware, de protecciones y CMD.

Como conclusión final este proyecto hemos aprendido que es necesario mucho esfuerzo para obtener el éxito y que el mismo depende de todas las áreas, requerimientos, arquitectura, desarrollo, testing y sobre todo de las diferentes gestiones.

11 Referencias bibliográficas

- [1] SCADA [Online], Disponible en: <https://es.wikipedia.org/wiki/SCADA> [Último acceso: 22/08/2017]
- [2] UTE [Online], Disponible en: <http://portal.ute.com.uy/institucional/qui%C3%A9nes-somos> [Último acceso: 22/08/2017]
- [3] Red eléctrica [Online], Disponible en: https://es.wikipedia.org/wiki/Red_el%C3%A9ctrica [Último acceso: 22/08/2017]
- [4] Mirage [Online], Disponible en: <http://www.controles.com/espanol/Telecontrol/SCADA.htm> [Último acceso: 22/08/2017]
- [5] Competencia desleal [Online], Disponible en: <http://evamunoz.es/actos-de-competencia-desleal-explotacion-de-la-reputacion-ajena/> [Último acceso: 22/08/2017]
- [6] Organigrama UTE [Online], Disponible en: <http://portal.ute.com.uy/institucional/organigrama> [Último acceso: 22/08/2017]
- [7] Unifilar [Online], Disponible en: https://es.wikipedia.org/wiki/Esquema_unifilar [Último acceso: 22/08/2017]
- [8] Principios y Patrones de diseño [Online], Disponible en: <https://rchavarria.wordpress.com/2010/03/04/principios-y-patrones-de-diseno-2/> [Último acceso: 22/08/2017]
- [9] Aseguramiento de la Calidad [Online], Disponible en: https://es.wikipedia.org/wiki/Aseguramiento_de_la_calidad [Último acceso: 22/08/2017]
- [10] Reglamentos para estudiantes [Online], Disponible en: <http://fi.ort.edu.uy/estudiantes/reglamentos#normas> [Último acceso: 22/08/2017]
- [11] Convenciones de código de C# [Online], Disponible en: <https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/inside-a-program/coding-conventions> [Último acceso: 22/08/2017]
- [12] Métricas [Online], Disponible en: https://aulas.ort.edu.uy/pluginfile.php/173969/mod_resource/content/0/CS-5-%20Mediciones.pdf [Último acceso: 22/08/2017]
- [13] HM Government, ITIL Service Transition, 2011th ed. The Stationery Office; 2nd edition 2011.
- [14] Gestión de Proyectos [Online], Disponible en: https://es.wikipedia.org/wiki/Gesti%C3%B3n_de_proyectos [Último acceso: 22/08/2017]

- [15] Universidad ORT Uruguay <<Ingeniería de Software 2 - Introducción>> [Online], Disponible en: <https://aulas.ort.edu.uy/mod/folder/view.php?id=9141> [Último acceso: 22/08/2017]
- [16] Universidad ORT Uruguay <<Ingeniería de Software 2 - Procesos y ciclos de Vida>> [Online], Disponible en: <https://aulas.ort.edu.uy/mod/folder/view.php?id=9141> [Último acceso: 22/08/2017]
- [17] Modelo iterativo incremental [Online], Disponible en: <https://es.wikipedia.org/wiki/Software> [Último acceso: 22/08/2017]
- [18] HM Government, ITIL Service Design, 2011th ed. The Stationery Office; 2nd edition 2011.
- [19] ITIL Gestión del Riesgo [Online], Disponible en: https://wiki.es.it-processmaps.com/index.php/ITIL_Gestion_del_Riesgo [Último acceso: 22/08/2017]
- [20] Open Geospatial Consortium (OGC) [Online], Disponible en: https://es.wikipedia.org/wiki/Open_Geospatial_Consortium [Último acceso: 22/08/2017]
- [21] Actuador [Online], Disponible en: <http://diccionario.raing.es/es/lema/actuador> [Último acceso: 22/08/2017]
- [22] Alarmas [Online], Disponible en: https://es.wikipedia.org/wiki/Sistema_de_alarma [Último acceso: 22/08/2017]
- [23] Centro de transformación <<Celdas de corte>> [Online], Disponible en: https://es.wikipedia.org/wiki/Centro_de_transformaci%C3%B3n [Último acceso: 22/08/2017]
- [24] Estaciones y Subestaciones [Online], Disponible en: <https://twenergy.com/co/a/que-son-las-subestaciones-electricas-y-para-que-sirven-1759> [Último acceso: 22/08/2017]
- [25] Instalaciones Eléctricas [Online], Disponible en: https://es.wikipedia.org/wiki/Instalaci%C3%B3n_el%C3%A9ctrica [Último acceso: 22/08/2017]
- [26] PAT o Puesta a Tierra [Online], Disponible en: https://es.wikipedia.org/wiki/Toma_de_tierra [Último acceso: 22/08/2017]
- [27] Protocolo 101 y 104 [Online], Disponible en: [Último acceso: 22/08/2017]
- [28] Protocolo DNP [Online], Disponible en: <https://es.wikipedia.org/wiki/DNP3> [Último acceso: 22/08/2017]
- [29] Seccionador [Online], Disponible en: <https://es.wikipedia.org/wiki/Seccionador> [Último acceso: 22/08/2017]

12 Glosario

Actuador: Dispositivo o máquina que activa el sistema de control de un proceso; la naturaleza de esta señal determina el tipo de actuador (hidráulico, neumático, electrónico, etc.).[21]

Alarma: Una alarma es un elemento de seguridad pasiva. Esto significa que no evitan una situación anormal, pero sí son capaces de advertir de ella y reducir el tiempo de ejecución de las acciones a tomar. Por ejemplo en instalaciones eléctricas podemos tener alarmas contra Intrusos, de detección de humo, de baja tensión de baterías, entre muchas más. [22]

Bandera DPF: Representación de detección de paso de falta (DPF) en los sistemas SCADA. Las alarmas de paso de falta detectan un cambio brusco de corriente en la red eléctrica y en SCADA lo representa el dibujo de una bandera que cambia de color de acuerdo a lo alarmado.

Celda de corte: Conjunto de elementos de corte de maniobra y protección. Se pueden diferenciar en las siguientes: [23]

- Celda de entrada de línea: Son celdas que reciben las líneas desde el exterior del centro. Están equipadas con un interruptor o una pareja interruptor-seccionador. A pesar de su nombre, pueden realizar la función de entrada o salida de línea.
- Celda de seccionamiento: En ella se encuentran los elementos de apertura de las diferentes líneas de tensión. En cuyo caso podría observarse un corte visible del mismo, siempre que hiciera falta.
- Celda de remonte: Permite remontar los cables directamente hasta el embarrado formado por el conjunto de celdas.
- Celda de medida: Estas celdas alojan en su interior los transformadores de medida, de tensión e intensidad. Estos transformadores convierten los valores de las magnitudes eléctricas (tensión e intensidad) propias de la línea a valores manejables por los equipos de medida (100 voltios y 5 amperios).
- Celda de protección de transformador: Son celdas dotadas con interruptor-seccionador automático con fusibles. Protege al transformador mediante relés indirectos que actúan sobre la bobina de disparo del interruptor. La protección frente a cortocircuitos la realizan los fusibles.

Celdario: Es un conjunto de celdas, donde cada una se identifica con un número.

CMD: Centro de Maniobras de Distribución, son los responsables realizar cualquier operación sobre la red eléctrica de distribución.

Comando: Instrucción u orden de un usuario que ejecuta una acción sobre un elemento de la red eléctrica, a través del SCADA.

Comando habilitar/deshabilitar relé: Comando que permite habilitar o deshabilitar la comunicación con el dispositivo de campo. Es utilizado en la operación en los casos que los dispositivos estén en falla y/o reportando estados incorrectos.

Conectividad: Refiere a la representación gráfica de la conexión de los elementos de un circuito eléctrico, permitiendo visualizar rápidamente si dicho circuito carece de energía en alguno de sus tramos.

Elemento: Es cualquier entidad de la red eléctrica representable en el sistema SCADA.

Elemento comandable: elementos de cortes que se pueden comandar a distancia de manera remota.

Elemento de corte: Se clasifican en dos grupos, los de maniobra y los de protección. Los de maniobra son todos aquellos grupos y elementos que pueden accionar equipos eléctricos, mientras que los de protección Son aquellos elementos que sirven para proteger circuitos eléctricos, equipos, máquinas, etc. En ambos casos su accionamiento puede ser: Manual, Automático Local o a distancia.

Elemento Dinámico: Es una representación gráficas del dinamismo y acciones (comandos) que poseen los elementos que están en campo. En general representan estados de interruptores, alarmas, y medidas analógicas como tensiones, corrientes, potencias, etc.

Elemento Estático: Es una representación gráficas cuyo valor no cambia en tiempo real conforme al dinamismo de la red, pueden ser textos o figuras gráficas.

Equipo de trabajo de Hardware: Unidad encargada de la configuración de los dispositivos que comunican los elementos de la red eléctrica con el SCADA.

Equipo de trabajo de Protecciones: Unidad encargada de configurar y calibrar los dispositivos que protegen los elementos de la red eléctrica.

Equipo de trabajo de Software: Unidad encargada la administración de sistemas SCADA y del desarrollo y proceso de migración de los mismos. Además provee a otras unidades de UTE información obtenida del SCADA.

Equipo de trabajo del CMD: Operadores que trabajan en CMD.

Estación: Una estación eléctrica es una instalación, o conjunto de dispositivos, que forma parte de un sistema eléctrico de potencia. Su principal función es la producción, conversión, transformación, regulación, repartición y distribución de la energía eléctrica. La estación debe modificar y establecer los niveles de tensión de una infraestructura eléctrica, para que la energía pueda ser transportada y distribuida. [24]

Estado: Situación o modo en el que se encuentra una variable digital, por ejemplo en el contexto de los SCADA un interruptor es representado por una variable digital cuyos estados posibles son abierto y cerrado.

Instalación eléctrica: Una instalación eléctrica es el conjunto de circuitos eléctricos que, colocados en un lugar específico, tienen como objetivo dotar de energía eléctrica a edificios, instalaciones, lugares públicos, infraestructuras, etc. Incluye los equipos necesarios para asegurar su correcto funcionamiento y la conexión con los aparatos eléctricos correspondientes. [25]

Interruptor: Elemento de la red eléctrica que permite interrumpir el suministro de corriente eléctrica.

Medida: Cualquier valor que provenga de los diferentes dispositivos, como pueden ser parámetros eléctricos que miden tensión o potencia.

Mímico: Un mímico es la representación gráfica en pantalla del sistema que controla el SCADA, a los efectos de UTE mímico representa el diagrama de la instalación eléctrica, por lo que es indistinto denominarlo Mímico o Unifilar.

Mímicos Automáticos Mirage o por scripts: Son mímicos generados en tiempo de ejecución, cuyo contenido está dado por alarmas o medidas. En general dicho script está contenido en un botón, inserto en un unifilar de una instalación, que cuando el usuario lo presiona en la consola, este despliega una ventana que contiene alarmas y/o medidas respectivas de la instalación que este visualizando. Estos scripts son programados por UTE y tiene como ventaja prescindir de un archivo físico y son editables por el usuario, solo basta con parametrizar dicho botón con la instalación correspondiente para que despliegue las alarmas.

Mirage SCADA: El SCADA Mirage, es desarrollado por Controles y está orientado principalmente al uso en redes eléctricas incluyendo generación transmisión y distribución.

Su arquitectura modular permite emplearlo en instalaciones sencillas con una sola estación eléctrica o en aplicaciones complejas con decenas de RTU o como Centro de Control de usinas generadoras.

Sus principales características son:

- Arquitectura modular cliente/servidor con comunicación TCP/IP y RPC entre módulos.
- Capacidad de crecimiento agregando módulos adicionales según la aplicación.
- Se puede instalar en una única PC o en múltiples PC en red.
- Diseños de alta confiabilidad con Hot Standby.
- Seguridad de acceso por usuario y por estación.
- Múltiples protocolos con RTU con canal simple o duplicado.
- Comunicación con otros SCADA por IEC60870-5-101.
- Almacenamiento de datos históricos en ODBC y/o texto.

- Completo editor de mímicos con biblioteca y servidor de mímicos.
- Librería.NET para las aplicaciones desarrolladas por el usuario.
- Capacidad de ejecución de lógica local según IEC 61131.
- Sincronización de tiempo por GPS o NTP
- Interfaz OPC esclava
- Plataforma Windows

Se distribuye en dos versiones: Mirage y Mirage CL con igual código y solo difieren en la cantidad de puntos y usuarios que pueden operar y en los módulos que se integran.

La versión Mirage está prevista para ser usada en un Centro de Control y la versión Mirage CL está orientada a ser empleada como consola local de una estación eléctrica. [4]

Objeto Mirage: Un objeto Mirage es una entidad reutilizable que maneja el SCADA a nivel gráfico tanto para representar elementos dinámicos o estáticos de la red eléctrica, además de cualquier dibujo o texto complementario. Dichos objetos pueden ser creados y editados por los administradores del SCADA dándole así más uniformidad a los gráficos.

Panel de Comunicaciones: Es un cuadro perteneciente a los unifilares de SCADA X que contiene información de los paquetes del enlace de comunicación, como lo son paquetes transmitidos, recibidos, con error y reintentos de comunicación.

PAT o Puesta a tierra: Se emplea en las instalaciones eléctricas para llevar a tierra cualquier derivación indebida de la corriente eléctrica a los elementos que puedan estar en contacto, ya sea directa o indirectamente, con los usuarios (carcasas, aislamientos, etc.) de aparatos de uso normal, por un fallo del aislamiento de los conductores activos, evitando el paso de corriente al posible usuario. [26]

Pintar por conectividad: Funcionalidad que permite visualizar gráficamente que tramos de un circuito eléctrico tienen energía y cuáles tramos no. Dicha funcionalidad siempre está activa en el sistema SCADA X, mientras que en el SCADA Mirage se puede activar o desactivar, esta opción nos permite tener dos vistas, la vista de conectividad y una segunda vista que muestra una representación por nivel de tensión del unifilar.

Pintar por nivel de tensión: Funcionalidad propia del SCADA Mirage que permite visualizar rápidamente los niveles de tensión que componen una instalación.

Protocolo 101:(IEC 60870-5-101) es una norma internacional preparada por TC57 para la monitorización de los sistemas de energía, sistemas de control y sus comunicaciones asociadas. Es totalmente compatible con las normas IEC 60870-5-1 y IEC 60870-5-5 y su uso estándar es en serie y asíncrono para el telecontrol de canales entre DTE y DCE. El estándar es adecuado para múltiples configuraciones como la de punto a punto, estrella, *multidropped*, etc. [27]

Protocolo 104: IEC 60870-5-104 (IEC 104) es una extensión del protocolo IEC 101 con cambios en los servicios de la capa de transporte, de la capa de red, de la capa de enlace y de la capa física para satisfacer la totalidad de accesos a la red. El estándar utiliza la interfaz de red TCP/IP para disponer de conectividad a la red LAN (Red de Área Local) con diferentes *routers* instalación (RDSI, X.25, *Frame relay*, etc.) también se puede usar para conectarse a la WAN (*Wide Area Network*). La capa de aplicación IEC 104 se conserva igual a la de IEC 101 con algunos de los tipos de datos y los servicios no utilizados. Existen dos capas de enlace definidas en la norma, que son adecuadas para la transferencia de datos a través de Ethernet o una línea serie (PPP - *Point-to-Point Protocol*). Generalmente para los sistemas de energía se utiliza el protocolo IEC 104 para el centro telecontrol y el protocolo IEC 101 para la interacción con las remotas de campo. [27]

Protocolo DNP: Es un protocolo industrial para comunicaciones entre equipos inteligentes (IED) y estaciones controladores, componentes de sistemas SCADA. Es un protocolo ampliamente utilizado en el sector eléctrico, de gran difusión en Estados Unidos y Canadá, y menor presencia en Europa donde el uso de alternativas como IEC-60870 101 e IEC-60870 104 gozan de mayor popularidad. [28]

Reenganches: Son dispositivos que cuando se abren con la finalidad de proteger las líneas de transmisión, apertura por algún evento externo, los mismos intentan reconectar nuevamente de forma automática, realizando varios intentos.

Relé: El relé es un dispositivo que se usa para proteger los elementos de la red eléctrica o para enviar a distancia sus estados.

Seccionador: Un seccionador es un elemento de la red eléctrica que permite separar de manera mecánica un circuito eléctrico de su alimentación, garantizando visiblemente una distancia satisfactoria de aislamiento eléctrico. [29]

Segmento: Estructura utilizada en el SCADA X para contener los elementos que componen un mímico.

Sistema Parcial 101: Son canales de comunicación del protocolo 101 y se identifican con un número.

Sistema SIO: Ver Anexo 5 – SIO (Sistema integrado de Operación).

Subestación: Se diferencia de la estación por los niveles de tensión, que son más bajos por lo general.

Telecontrol: Según Real Academia Española, telecontrol se refiere a mando de un aparato o sistema, ejercido a distancia. Además en el contexto del proyecto refiere a la unidad de UTE encargada de administrar los sistemas SCADA.

Textos estáticos: son textos que por lo general poseen nombres de instalaciones o elementos de las mismas, una vez dibujados permanecen incambiados.

Unifilar: Un esquema o diagrama unifilar es una representación gráfica de una instalación eléctrica o de parte de ella. Generalmente el esquema unifilar tiene una estructura de árbol, no obstante no es la única. A los efectos de este documento es indistinto nombrarlo mímico o unifilar.

13 Anexos

13.1 Anexo 1 – Framework ITIL v3

En este anexo presentamos una descripción breve de ITIL V3. En nuestro proyecto aplicamos ITIL en la gestión de lanzamiento y en la gestión de cambios, que fueron tratados en los capítulos 8 y 9 respectivamente.

La Biblioteca de Infraestructura de Tecnologías de Información (o ITIL, por sus siglas en inglés) es un conjunto de conceptos y buenas prácticas usadas para la gestión de servicios de tecnologías de la información, el desarrollo de tecnologías de la información y las operaciones relacionadas con la misma en general. ITIL da descripciones detalladas de un extenso conjunto de procedimientos de gestión ideados para ayudar a las organizaciones a lograr calidad y eficiencia en las operaciones de TI. Estos procedimientos son independientes del proveedor y han sido desarrollados para servir como guía que abarque toda infraestructura, desarrollo y operaciones de TI. [17]

ITIL v3 fue publicada en el 2007 y consta de los siguientes 5 volúmenes, el desarrollo de nuestro proyecto, por tratarse de la migración de un servicio estaría enmarcado en el 5 volumen “ITIL v3 *Service Transition*”. En los capítulos *Release & Deployment* y *Configuration Manangement System*.



Figura 13.1: Volúmenes ITIL V3

Este ciclo de vida consta de 5 fases [18]:

1. **Estrategia del servicio:** Esta fase es central al concepto de Ciclo de vida del servicio y tiene como principal objetivo convertir la Gestión del Servicio en un activo estratégico. Para conseguir este objetivo es imprescindible determinar qué servicios deben ser prestados y por qué han de ser prestados desde la perspectiva del cliente y el mercado.
2. **Diseño del servicio:** La principal misión de esta fase es la de diseñar nuevos servicios o modificar los ya existentes para su incorporación al catálogo de servicios y su paso al entorno de producción. Siempre siguiendo las directrices establecidas en la fase de Estrategia.
3. **Transición del servicio:** El objetivo de esta fase es hacer que los productos y servicios definidos en la fase de Diseño se integren en el entorno de producción y sean accesibles a los clientes y usuarios autorizados.
4. **Operación del servicio:** Cubre las mejores prácticas para la gestión del día a día en la operación del servicio. Esta fase es, sin duda, la más crítica, ya que la percepción que los clientes y usuarios tengan de la calidad de los servicios prestados depende en última instancia de una correcta organización y coordinación de todos los agentes involucrados.
5. **Mejora continua del servicio:** Proporciona una guía para la creación y mantenimiento del valor ofrecido a los clientes a través de un diseño, transición y operación del servicio optimizado.

13.2 Anexo 2 – Funcionalidades destacables SCADA X

El sistema X cuenta con innumerables funcionalidades, de las cuales vamos a enfocarnos solamente en las que tienen similitud, puntos de contacto o más valor agregado en el SCADA Mirage, que son las que se detallan a continuación:

1. Cada elemento de corte en un unifilar se representan de manera individual en un objeto gráfico.

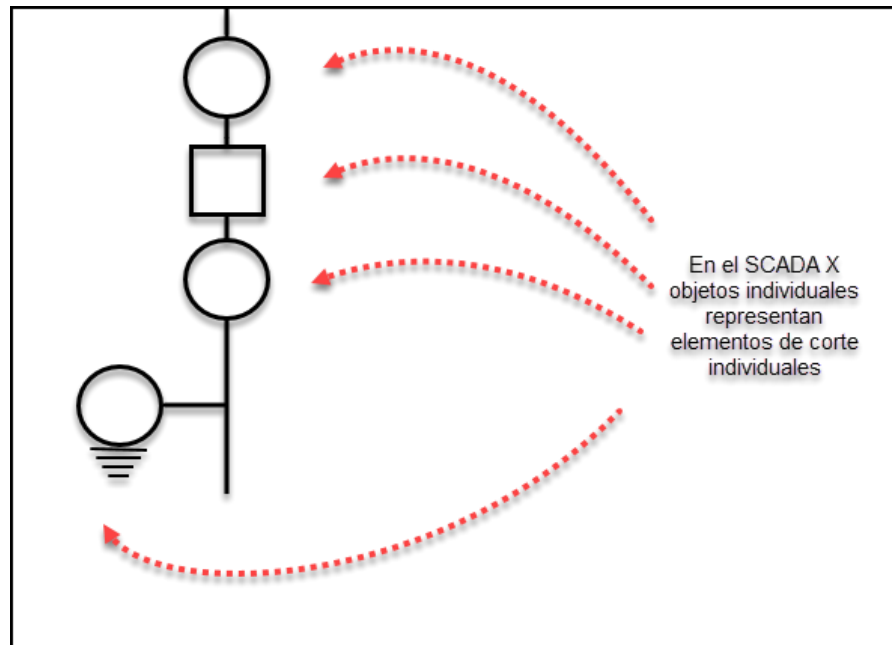


Figura 13.2: Objetos de corte individuales en SCADA X



Figura 13.3: Único Objeto de corte en Mirage

2. Conectividad: Los elementos de corte que componen un unifilar de instalación están conectados de forma tal que visualmente se aprecia la topología de la red

interna de la instalación. Esto es, frente a una apertura de un elemento, se ve rápidamente el tramo de la red sin tensión, en SCADA se visualiza un tramo sin pintar. Esto no aplica a puntos intermedios, donde solo existe un único elemento de corte.

3. Los elementos de corte tienen un mismo color en cualquier nivel de tensión, mientras que el SCADA Mirage sigue una convención de colores por nivel de tensión (véase Anexo 10).
4. Áreas de operación y prueba: El SCADA X maneja tres tipos de áreas, 1, 2 y 3. Área 1 corresponde al área de prueba o desarrollo, donde las instalaciones no pueden ser operadas por ningún CMD; área 2 corresponde al área de operación o explotación del CMD de Canelones; y finalmente el área 3 corresponde al área de operación del CMD de Montevideo.

13.3 Anexo 3 - Variables SCADA Mirage

Las variables SCADA conforman la representación lógica de cualquier elemento telecontrolado de una instalación.

Los unifilares o mímicos contienen objetos animados que representan gráficamente los elementos de instalación eléctrica real, estos objetos hacen referencia a la parte lógica del SCADA, que son las variables.

Estas variables si bien tienen una descripción que se interpreta fácilmente, las mismas se identifican por un nombre compuesto jerárquicamente generado por el CMD, Instalación, nivel de tensión, ubicación relativa a la instalación (celda, acople, trafo, general, etc.), y elemento (interruptor, seccionador, alarma, etc.) Por ejemplo, el interruptor de celda 4 en 6kV de la estación EST5049 del CMD de Canelones, se identifica como **CAEST5049_006_C04_INT** donde:

- CA: Corresponde al CMD actual pertenece la instalación, en este caso Canelones (CA).
- EST5049: Corresponde a la matrícula de la instalación.
- _006: Indica el nivel de tensión de la variable, en este caso 6kV.
- _C04: Indica la ubicación de la variable dentro de la estación EST5049, en este caso corresponde a la celda 04.
- _INT: Indica el elemento representado, en este caso corresponde a un interruptor.

En general esta información es utilizada por los administradores del sistema SCADA, quienes operan la red solo se guían por la descripción de las variables y por la ubicación grafica de los elementos que las representan en el mímico.

En la siguiente imagen vemos el cuadro de diálogo que el operador visualiza al momento de seleccionar el elemento anterior desde el mímico de la estación EST5049:

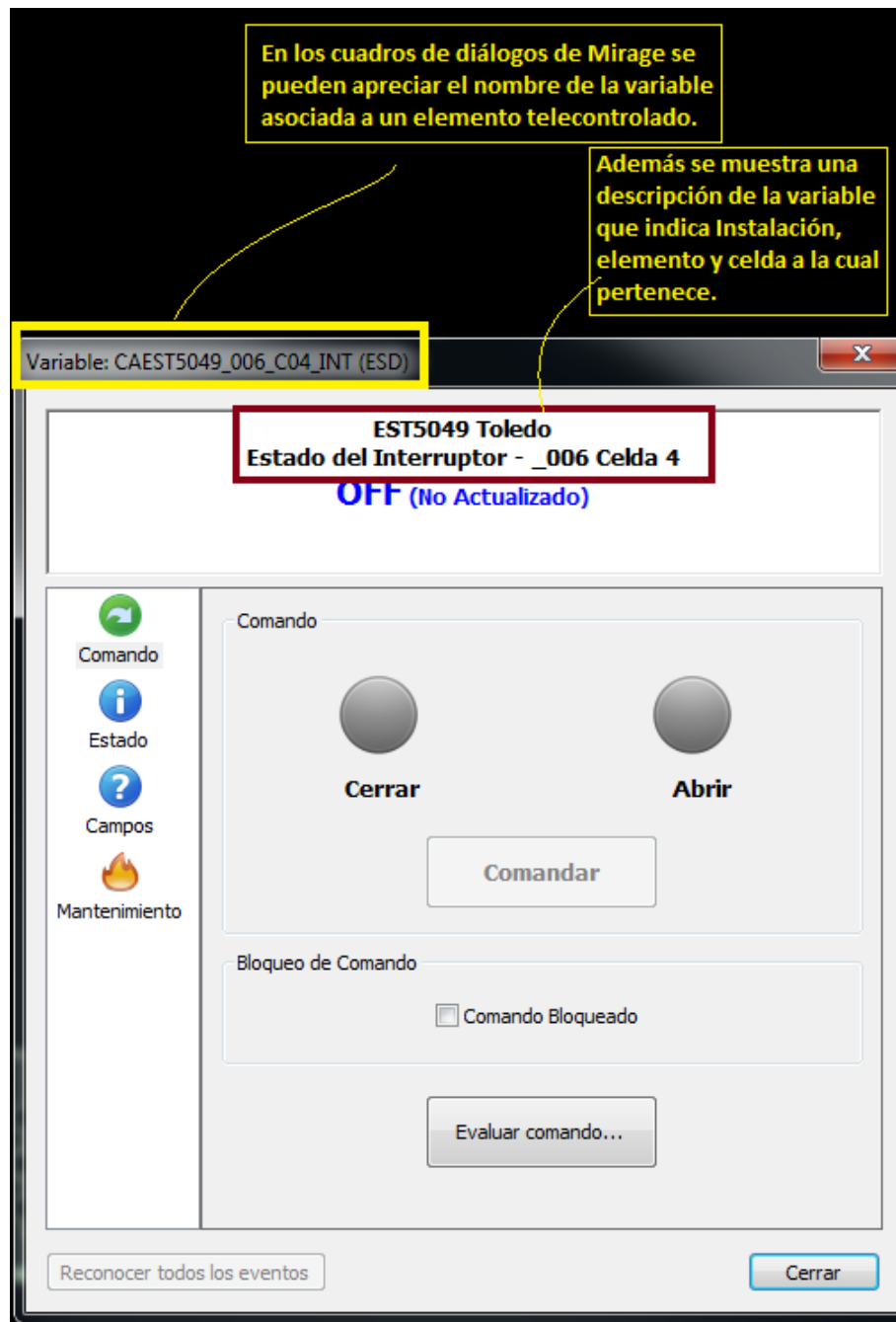


Figura 13.4: Identificación de variables

Editor Mirage

El mapeo entre los elementos y las variables en el mímico, se realizan mediante el editor Mirage, que es el software en el cual se construyen los mímicos.

El editor permite construir “objetos Mirage” y esos objetos son insertados en el mímico y son animados de acuerdo al valor que toman las variables SCADA. Por ejemplo, un objeto que representa una medida de corriente, mostrará en el mímico el valor de la variable que representa esa corriente.

Representación de elementos compuestos

Los objetos que pueden representar a más de una variable, se los llama “objetos compuestos”, y a continuación vemos un ejemplo.

En el editor cuando se selecciona un elemento para mapear las variables se mostrará el cuadro a continuación:

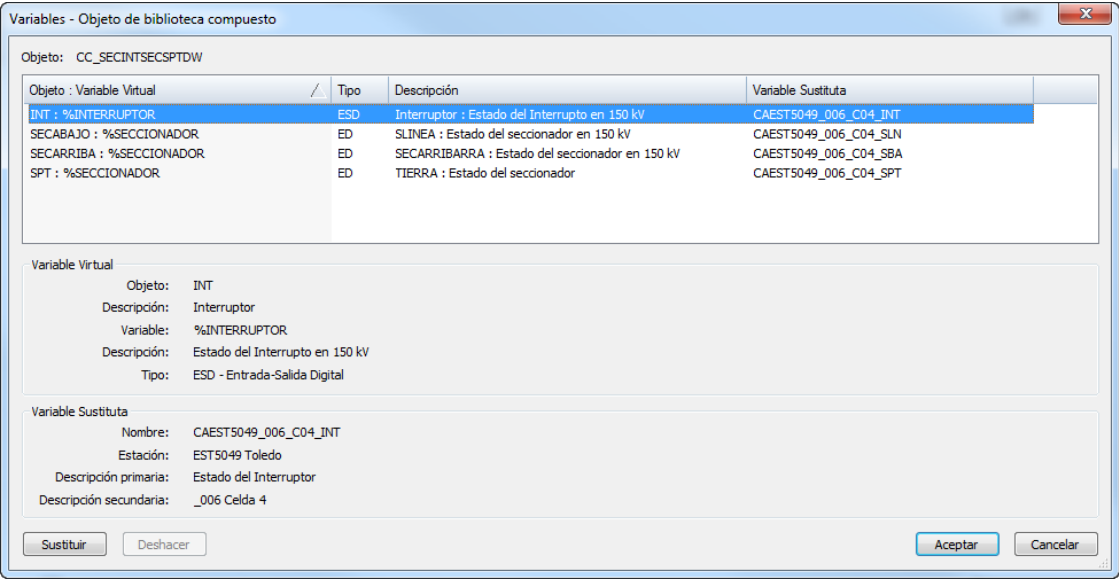


Figura 13.5: Objeto seleccionado CC_SECINTSECSPTDW

A continuación las distintas vistas del objeto seleccionado:



Figura 13.6: Vistas objeto CC_SECINTSECSPTDW

Detalle de mapeo a nivel gráfico:

El elemento compuesto CC_SECINTSECSPTDW contiene las siguientes variables:

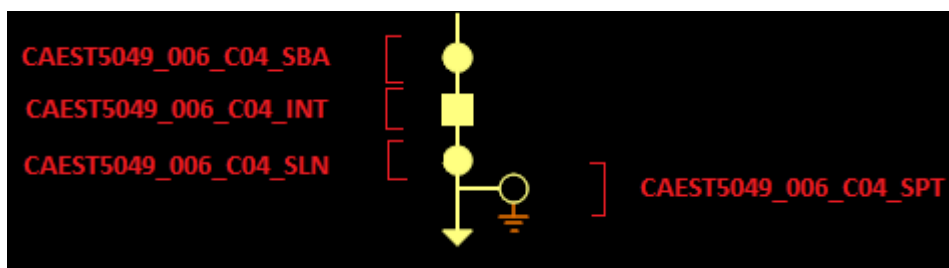


Figura 13.7: Elemento compuesto

13.4 Anexo 4 – Color por nivel de tensión en SCADA Mirage

Nivel de Tensión (kV)	Color	Codificación RGB
6	Amarelo	(255,255,128)
15	Verde	(255,255,255)
30	Ciano	(0,255,255)
60	Naranja	(255,128,0)
150	Purpura	(128,128,255)

Figura 13.8: Color por nivel de tensión

13.5 Anexo 5 – SIO (Sistema integrado de Operación)

El Sistema Integrado de operación, SIO, es una herramienta con la que cuenta UTE para gestionar incidencias y maniobras de toda la red eléctrica del país de manera gráfica y alfanumérica. SIO tiene información de cada elemento de la red eléctrica y de la topología de la misma, así también como la cantidad de clientes que dependen de cada nodo de la red.

Si bien distintas unidades internas de UTE consultan al sistema SIO (por ej. Telegestiones, atendiendo reclamos de clientes con ausencia de energía eléctrica), son los operadores del CMD quienes se encargan de todo el manejo de incidencias en SIO, ya que ellos son los encargados de operar la red. Las operaciones sobre la red pueden ser mediante la asistencia de una brigada en el lugar físico de la red o, a distancia, desde la mesa de operación mediante los sistemas SCADA. Esto implica que los sistemas SIO y SCADA estén integrados a nivel de elementos de corte.

Por temas de rentabilidad no toda la red eléctrica está telecontrolada por los sistemas SCADA, pero si lo están las instalaciones más significativas. Entonces, dichas instalaciones se las integra con SIO, para los efectos de gestionar cualquier incidencia o maniobra que se suscite en sus elementos de corte. Por lo tanto, si ocurre una apertura

de un interruptor en SCADA, de manera controlada o no, dicha apertura provoca ausencia de energía eléctrica en un tramo de la red y se ve inmediatamente reflejada en SIO, indicando entre otras cosas, cuantos clientes están afectados en esa maniobra.

La manera que se integran los elementos de corte, interruptores y seccionadores, entre SCADA y SIO es por medio de un identificador numérico. Ese identificador es asociado al elemento en cada sistema y cuando se da un cambio de estado en SCADA de cualquier elemento, ese estado es enviado a SIO con el fin de registrar una incidencia. Además, como ambos sistemas permiten un manejo gráfico de la red, también se integran los mímicos de las instalaciones que están en SCADA. Esto permite al operador desplegar un mímico en SCADA desde el sistema SIO, facilitando las operaciones. Esta es la razón por la cual en los requerimientos del proyecto se menciona la necesidad de persistir el mapeo entre los identificadores de las instalaciones con el mímico generado.

A continuación vemos algunas imágenes de ejemplo del sistema SIO:

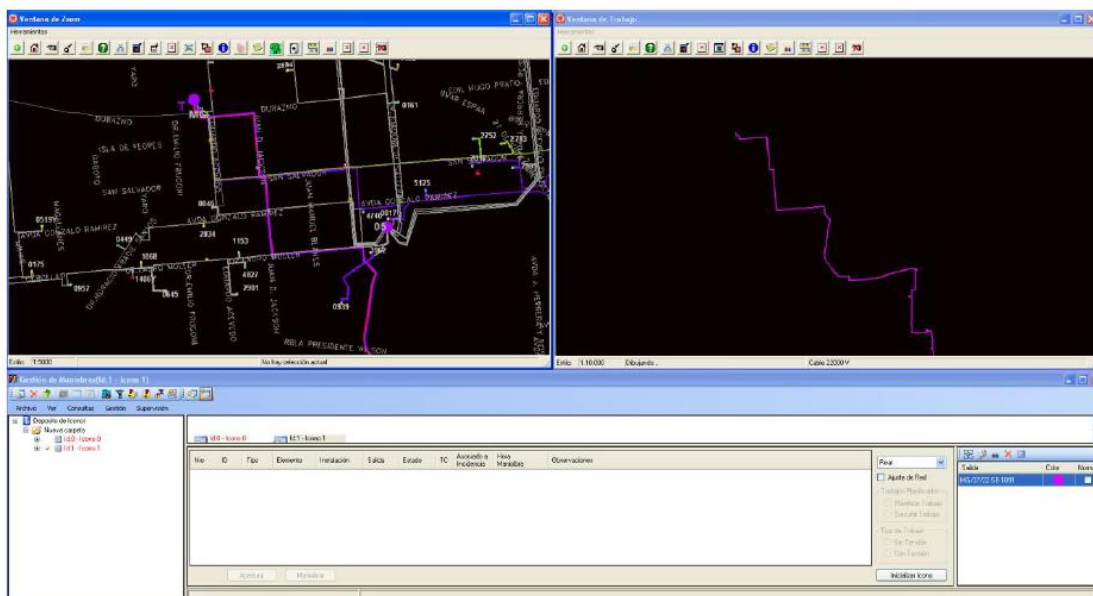


Figura 13.9: Vista de operación y manejo de incidencias en sistema SIO.

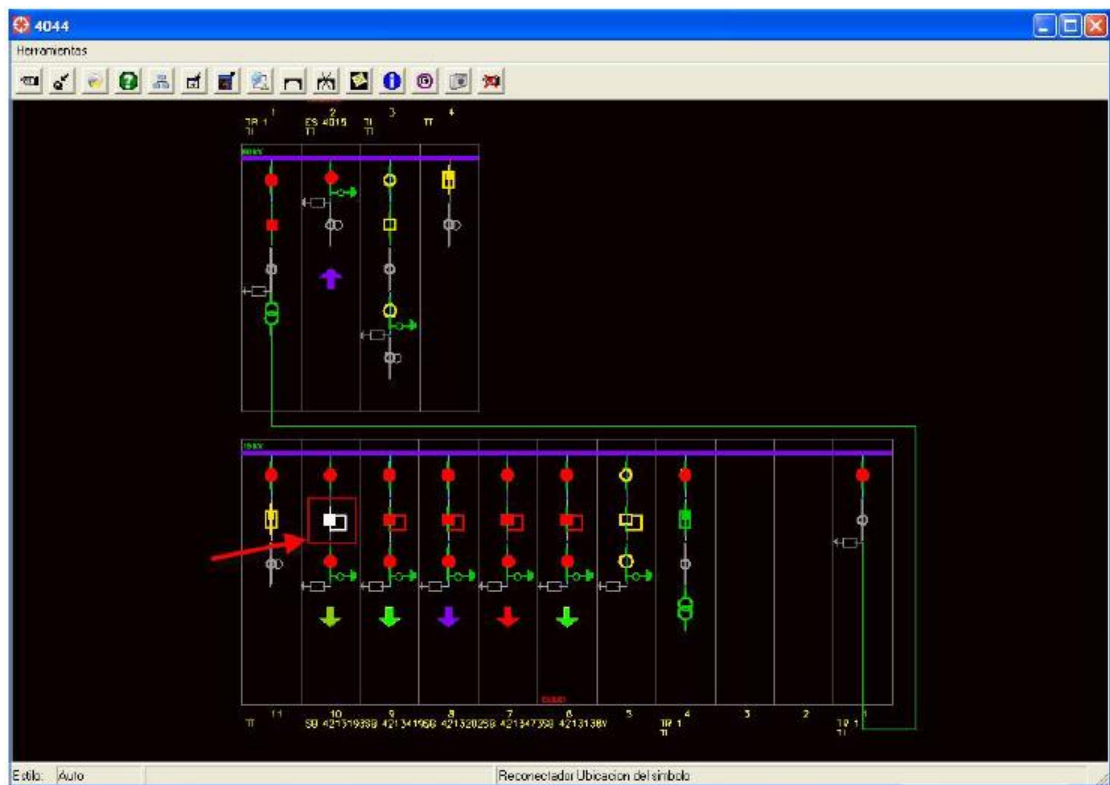


Figura 13.10: Ejemplo de un unifilar de estación en el sistema SIO

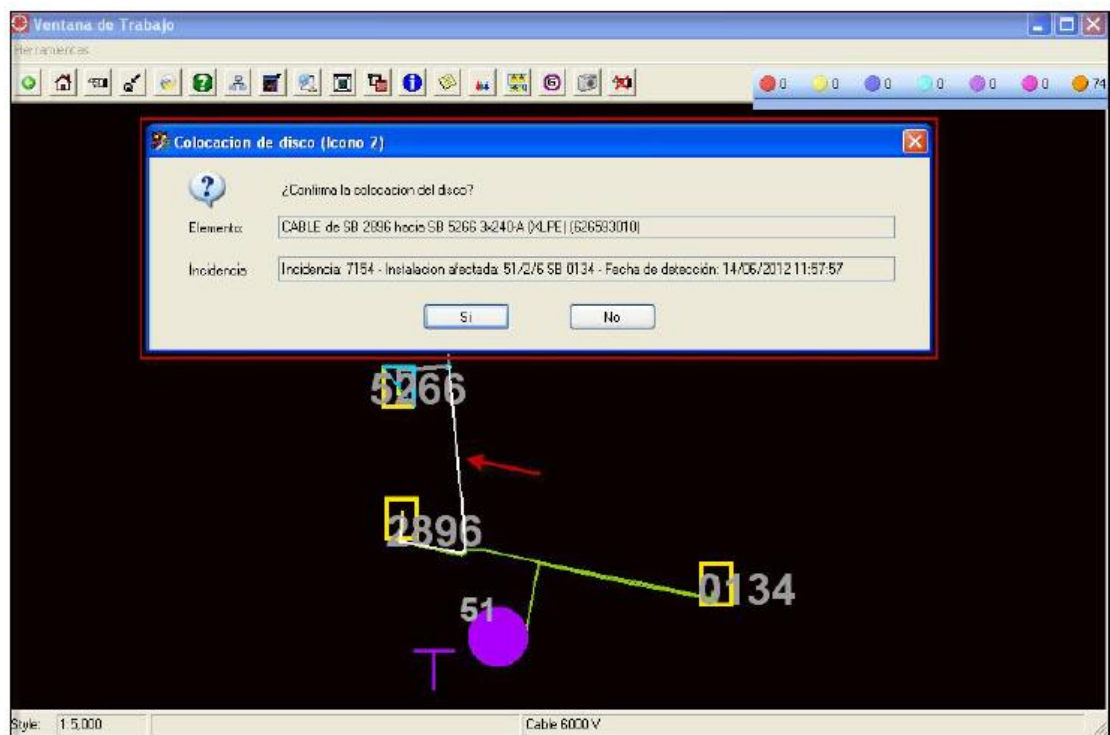


Figura 13.11: Ventana de trabajo del sistema SIO

13.6 Anexo 6 – Plan de Calidad

Se presenta la lista de actividades planificadas y qué se espera de cada actividad, con el objetivo de asegurar la calidad del producto final entregado y puesto en ejecución en CMD Canelones:

Actividad	Descripción	Resultado esperado
Relevar requerimientos	Aprovechamos el conocimiento que se posee en el dominio y una parte técnica accesible en el cliente para relevar los requerimientos.	Informe con lista de requerimiento relevados
Especificar requerimientos	Especificar los requerimientos relevados	Documento con requerimientos definidos.
Validar requerimientos	Validar requerimientos con cliente.	Informe con resultado de validación para evaluar posibles correcciones
Verificar requerimientos	Verificar que las funcionalidades cumplan con los criterios de aceptación especificados	Registrar como finalizada la funcionalidad
Diseñar Arquitectura	Se debe realizar el diseño de la arquitectura de la solución	Documento de arquitectura
Revisar diseño de Arquitectura	Revisión de la Arquitectura y discusión de ventajas y desventajas de la misma.	Documento de arquitectura mejorado.
Codificar	Construir el producto con los lenguajes de codificación seleccionados tomando como referencia los estándares definidos	Código Fuente
Realizar pruebas	En el software se realizaran pruebas unitarias, pruebas de Integración y pruebas cruzadas, mientras que en el producto final. 1) Pruebas automáticas. 2) Pruebas de Enlace. 3) Pruebas en campo.	Un producto testeado, que cumpla con los requisitos de calidad que se necesita para que entre en operación en un ambiente crítico, como es el control remoto de estaciones de UTE.
Definir estándar de codificación	Se seguirá las convenciones de código de C# (Guía de Programación de C#)	Código fuente estandarizado.
Definir estándar de documentación	Se utilizará el estándar definido por la Universidad ORT Uruguay para los proyectos finales de carreras.	Documentación cumpliendo estándares definidos por la Universidad ORT Uruguay.

Actividad	Descripción	Resultado esperado
Recolectar métricas del producto	Recolectar métricas como defectos o bugs detectados, tiempo trabajado en defectos	Informe con métricas descriptas que permitan evaluar y analizar el software, corrigiendo los bugs, para que no se repitan nuevamente en el proceso o que no se repitan en la migración de CMD Montevideo.
Verificar código con estándar definido	Se deberá verificar que el código fuente cumpla con los estándares de codificación definidos previamente.	Detectar incumplimiento de estándar para lograr un código legible que permite un mantenimiento óptimo por parte de cualquier desarrollador
Verificar documentación con estándar definido	Se deberá verificar que la documentación cumpla con los estándares definidos previamente.	Detectar incumplimiento de estándar para lograr una documentación de calidad
Registrar reuniones	Se registrarán las reuniones realizadas con tutor y con expertos del dominio.	Registro de reunión, para poder tener documentado y al alcance en cualquier momento los temas tratados en las reuniones

Tabla 13-1: Plan de calidad

13.7 Anexo 7 – Revisiones formales.

A continuación minuta de reunión y acta de aprobación de requerimientos

Minuta de reunión

Datos de la reunión

Fecha: 03/12/2016

Lugar: Rondeau 2352 1er piso

Objetivo/s: Validación de requerimientos.

Reunión convocada por: Andrés Martínez y Renato Moraes

Participantes

Nombre y apellido	Cargo	Referencia
Andrés Martínez	Desarrollador	AM
Renato Moraes	Desarrollador	RM
Lidia Dávila	Encargada de Unidad Software	LD

Temas tratados

Validación de requerimientos

Se presentó el documento de Especificación de requerimientos, en base a esto se validaron los mismos y se firmó el acta de aprobación de los requerimientos.

Compromisos asumidos

Descripción	Responsable
Definir Arquitectura	AM y RM
Definir <i>Testing</i>	AM y RM

Temas pendientes

1. Validación de la Arquitectura: en base a los requerimientos aprobados se deberá definir la arquitectura que se utilizara en el software para la migración.
2. Validación del *Testing*: Desarrollar *testing* completo para la prueba de los unifilares.

Próxima Reunión: 17/12/17

LISTADO DE REQUERIMIENTOS

Se definieron los siguientes requerimientos funcionales y no funcionales, para el proyecto migración:

Requerimientos Funcionales

1. Agrupar elementos de corte.
2. Identificar configuraciones de celda de corte.
3. Persistir selección de usuario.
4. Elementos dinámicos distintos a elementos de corte.
5. Conectividad en reenganches.
6. Elementos dinámicos.
7. Color de elementos comandables.
8. Mantener la conectividad de las instalaciones.
9. Mantener integración de instalaciones con sistema SIO de UTE.
10. Inyecciones de conectividad.
11. Formato textos estáticos y textos en celdarios (número y destino de celdas)
12. Textos estáticos de unidades de medidas.
13. Migración nombre interno instalación.
14. Comandos habilitar/deshabilitar relé
15. Selección por área.
16. Selección por instalación

17. Variables en bandera DPF.
18. Nombre de variables en proceso de conversión.
19. No migrar panel de comunicaciones.
20. Instalaciones cruzadas.
21. Celdas bien referenciadas.
22. Celdas solapadas gráficamente.
23. Botón de comunicaciones 101.
24. Mímicos invocados.
25. Conectividad de instalaciones.
26. Mímicos con instalaciones migrados automáticamente.

Requerimientos no funcionales

1. Sistema operativo Windows 7
2. Acceso a bases de datos
3. Lenguaje de programación
4. IDE de programación
5. Pruebas automatizadas
6. *Testing* unitarios
7. Auditoría del sistema (log)
8. Utilización del *Framework* ITIL V3

ACTA DE APROBACION DE REQUERIMIENTOS

Los requerimientos descritos en dicho documento y especificados en el documento “Especificación de Requerimientos” fue revisado a la fecha de hoy y aceptados por ambas partes involucradas, desarrolladores y contraparte autorizada por UTE.

13.8 Anexo 8 – Revisiones informales con técnicos UTE.

Se realizaron incontables consultas informales a técnicos de UTE, a continuación el registro por mail:

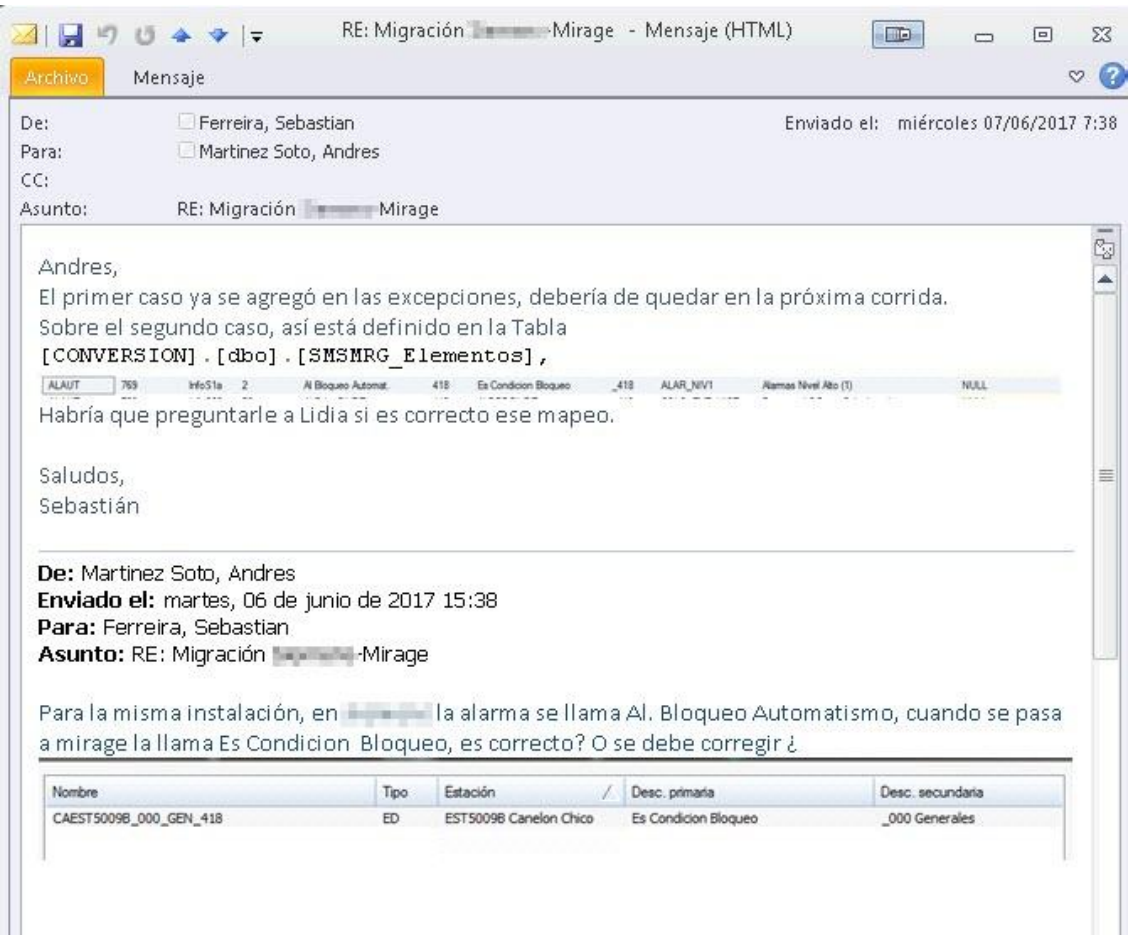


Figura 13.12: Revisión informal

13.9 Anexo 9 – Revisiones informales entre equipo.

Muchas de las reuniones informales se coordinaban por Hangouts o Whatsapp.



Figura 13.13: Coordinación Reunión Informal

13.10 Anexo 10 – Aprobación de sistema de *Testing*.

Montevideo, 2 de marzo de 2017

PRUEBAS

Se definen que se realizarán las siguientes pruebas:

- Pruebas Unitarias
- Pruebas de Integración
- Pruebas cruzadas
- Pruebas de Enlace
- Pruebas automáticas
- Pruebas de enlace
- Pruebas en campo

ACTA DE APROBACION DE TESTING

Las pruebas definidas en este documento fueron revisadas a la fecha de hoy y aceptadas por ambas partes involucradas, desarrolladores y contraparte autorizada por UTE.

13.11 Anexo 11 – Pruebas unitarias

Las pruebas unitarias fueron muy importante durante el desarrollo de la aplicación. En muchos casos existía una complejidad al momento de armar un mímico, ya que los mismos utilizaban objetos compuestos por otros objetos internos. Además, Mirage no usa la referencia a dichos objetos de manera simple, sino que necesita toda la estructura original del objeto interno.

Entonces se confeccionó un programa para diseñar los objetos Mirage convertidos, así de esta manera se iba entendiendo poco a poco la estructura de los mismos. Dicho código de programa luego fue utilizado en el software de conversión, para indicarle al usuario qué objetos se están utilizando.

En la siguiente imagen vemos el dibujo correspondiente al objeto compuesto "CC_SECINTSECSPTDW" mencionado anteriormente:

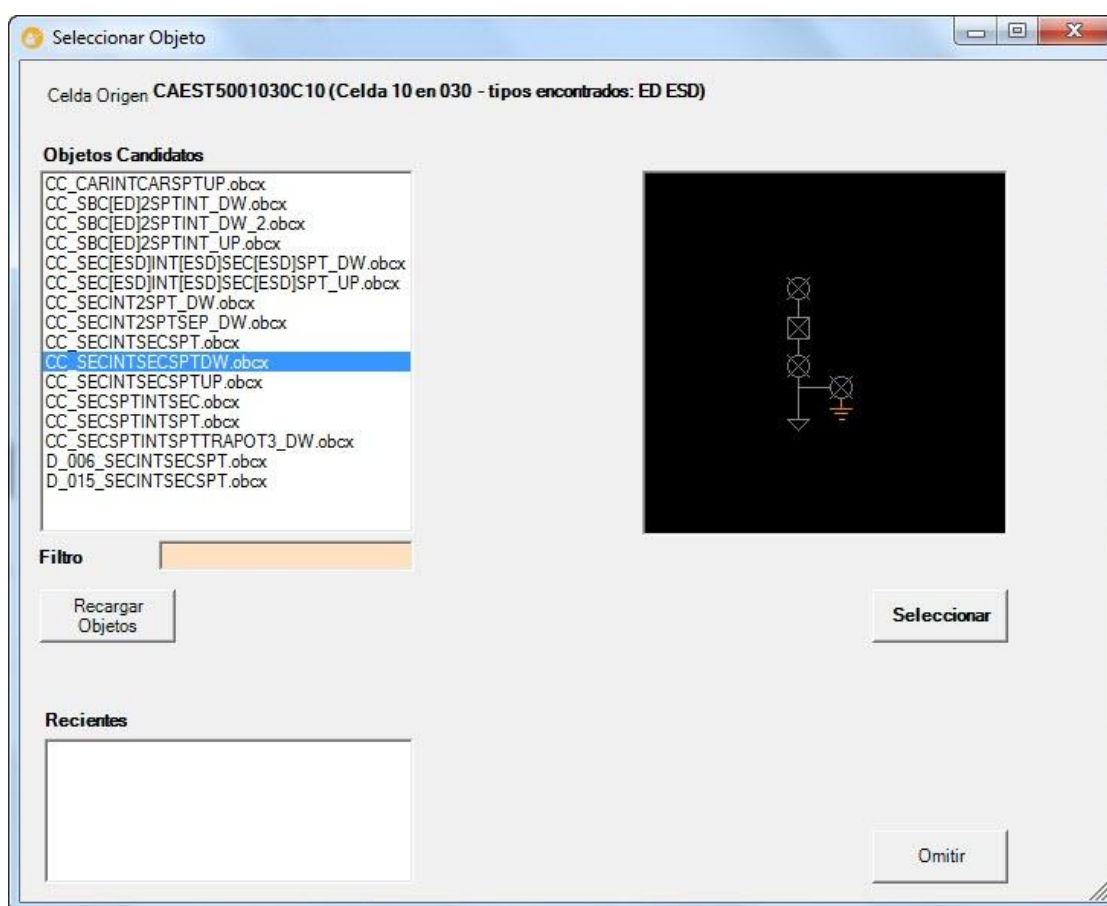


Figura 13.14: Ventana Selección Objetos

13.12 Anexo 12 – Pruebas de Integración en unifilar.

A continuación vemos el mismo unifilar en ambos sistemas, fueron de las primeras conversiones realizadas en forma automática.

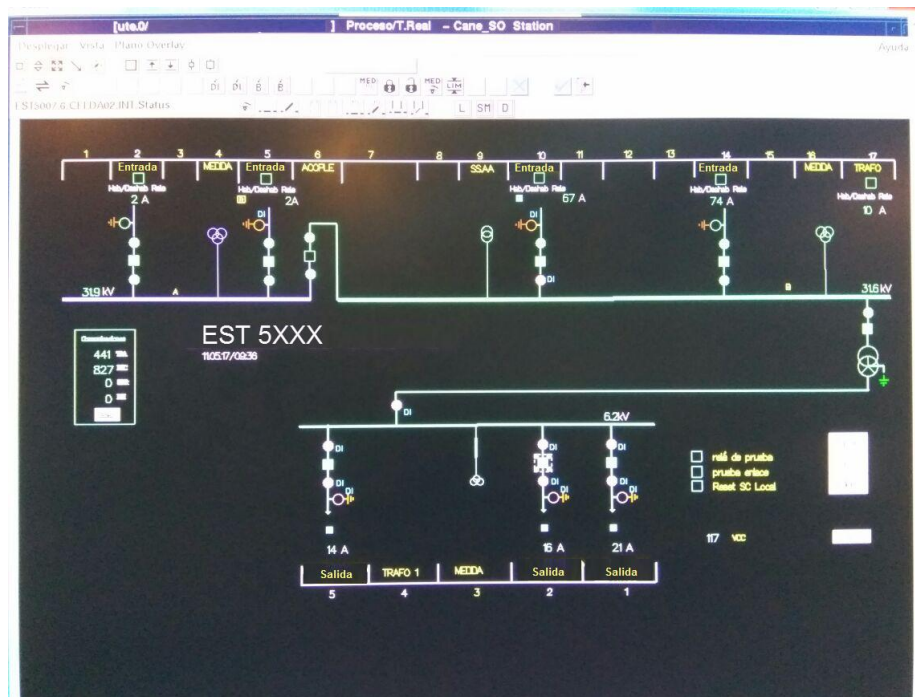


Figura 13.15: Sistema SCADA X

Luego de haber traducido todos los elementos por separados, se debían integrar en un único mímico. Como se puede observar la integración de todos los elementos fue un éxito.

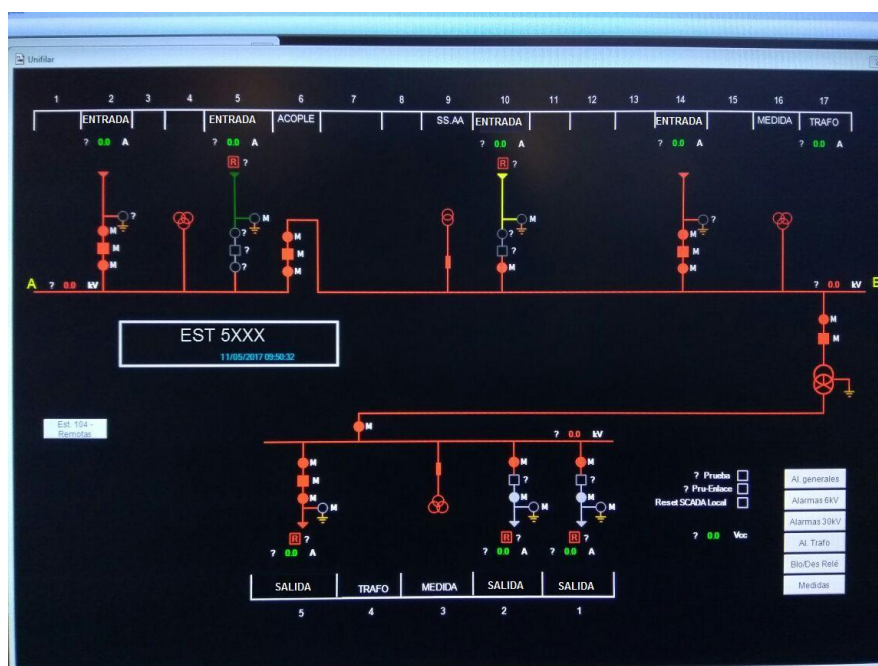


Figura 13.16: Sistema SCADA Mirage

13.13 Anexo 13 – Pruebas Automáticas

Software para realizar las pruebas automáticas.

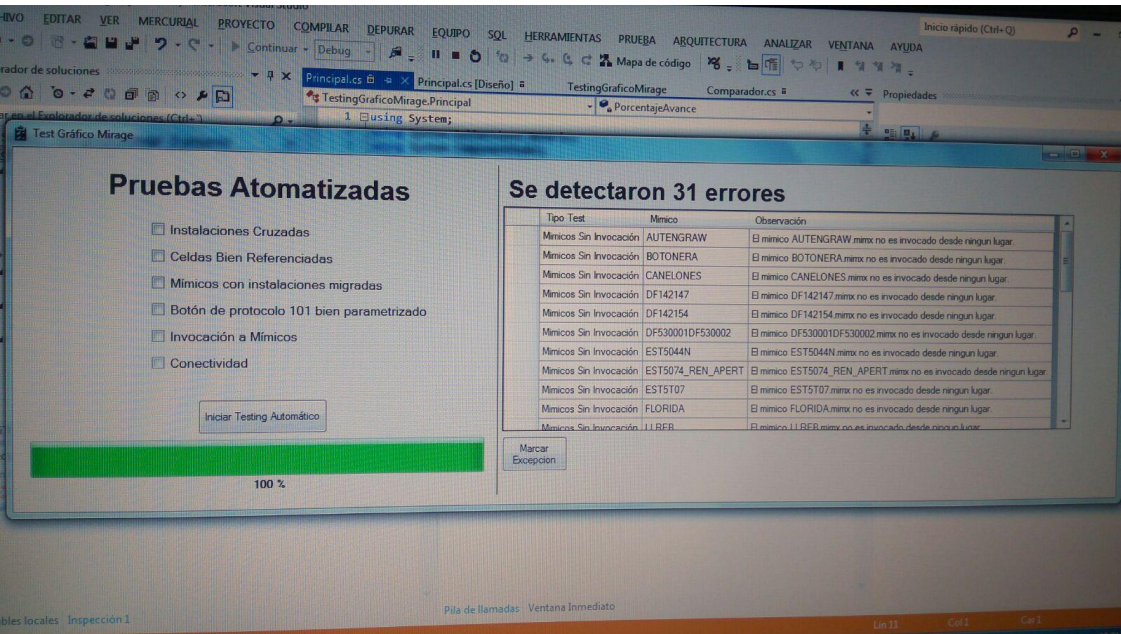


Figura 13.17: Software pruebas automáticas

A continuación se ilustran algunos ejemplos de lo que se busca probar de manera automática.

Botón de protocolo 101: Debido a que el sistema que UTE usa para asignar dicho protocolo lo hace de forma aleatoria, por lo tanto se deben chequear que todos los botones hagan referencia al sistema correcto. En la siguiente imagen se puede ver un ejemplo en el que está cambiada la asignación:

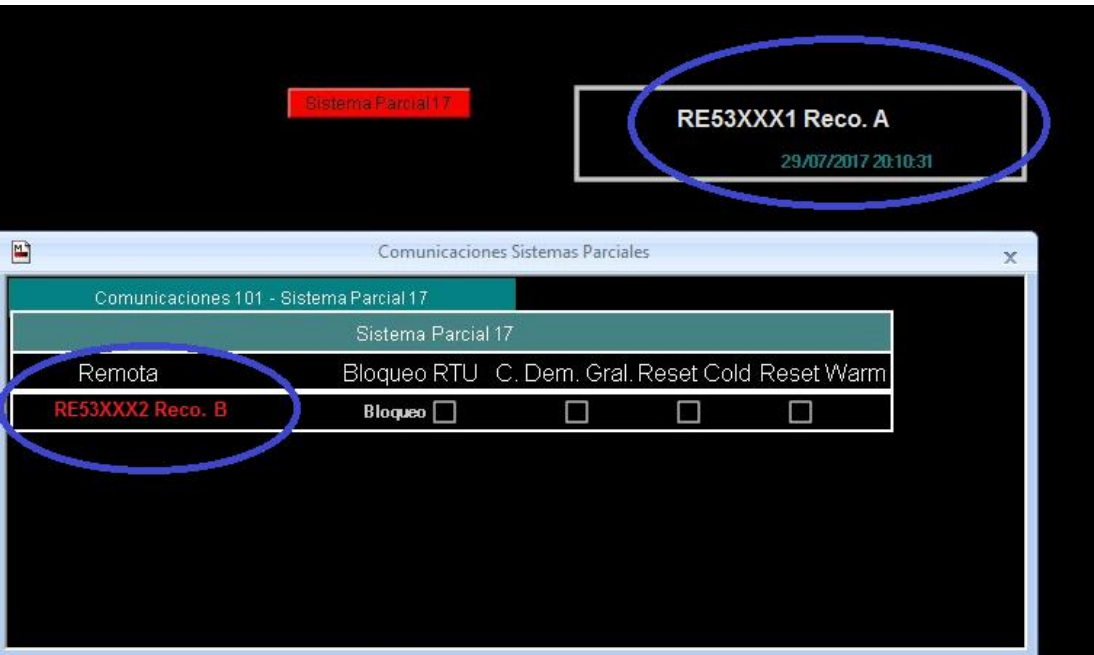


Figura 13.18: Asignación cambiada en Botón Protocolo 101

Invocación a Mímicos: Acá se refiere a que desde el mapa de la zona en que se encuentra la estación existe un link para abrir dicho mímico. A continuación se puede ver una parte del mapa de distrito:

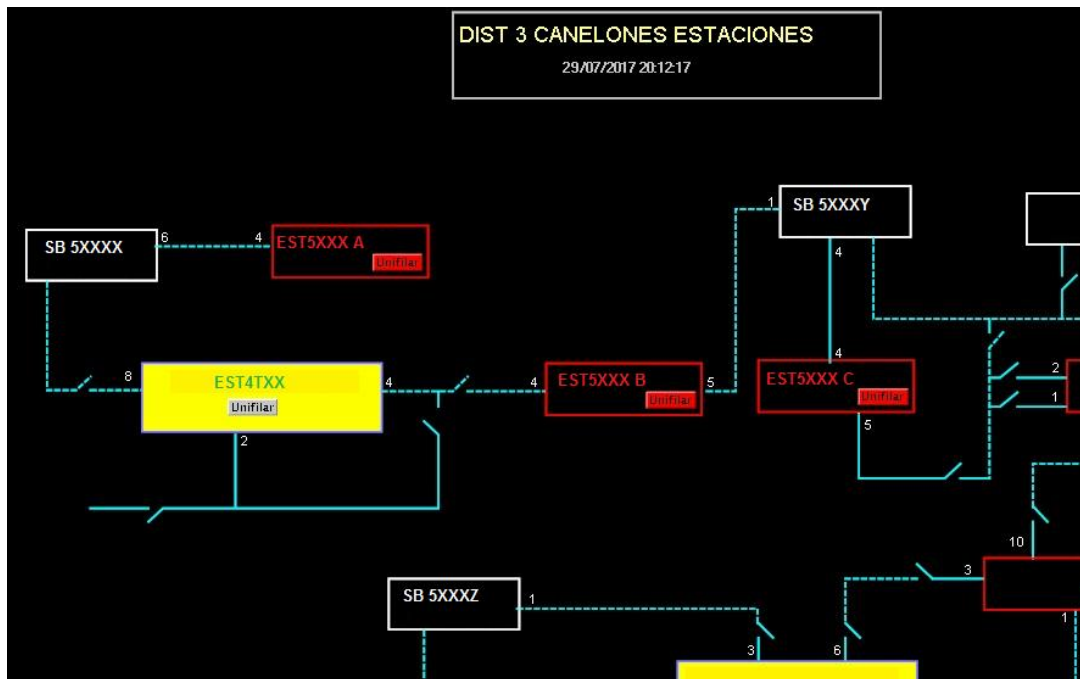


Figura 13.19: Mapa de Distrito

Continuidad: A continuación un ejemplo para ilustrar a qué se hace referencia con continuidad, el primer diagrama sin la opción de continuidad seleccionada.

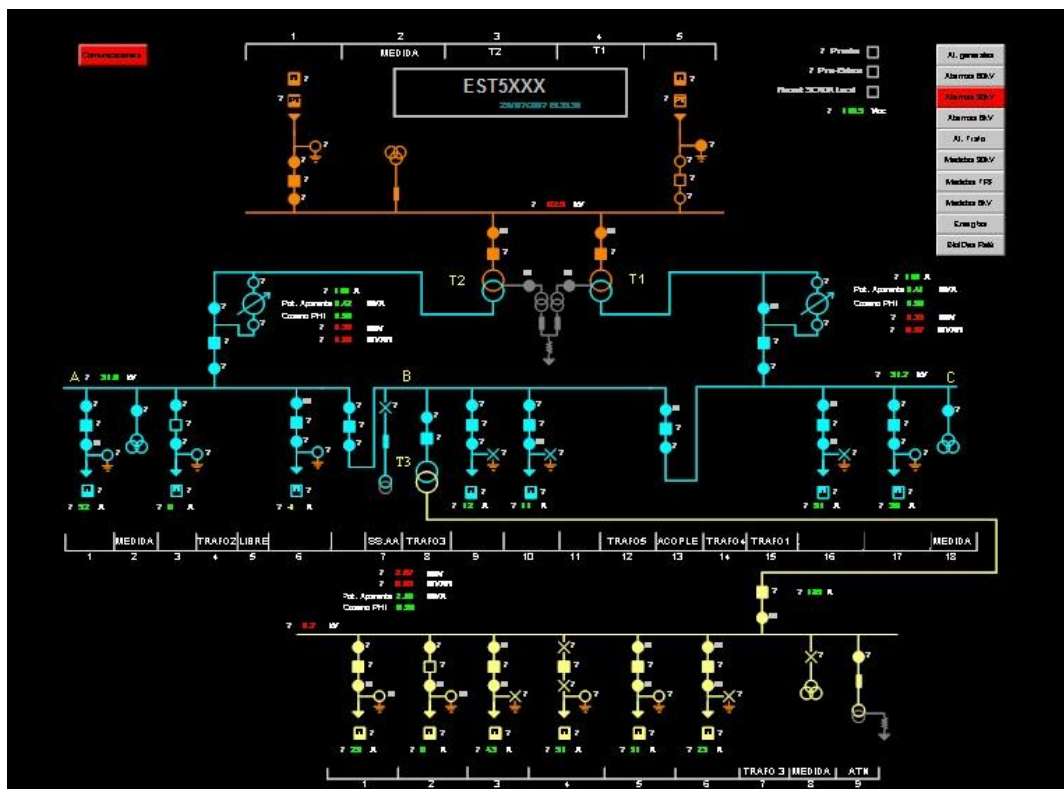


Figura 13.20: Sin continuidad seleccionada

Mientras que en este segundo diagrama la continuidad ha sido seleccionada:

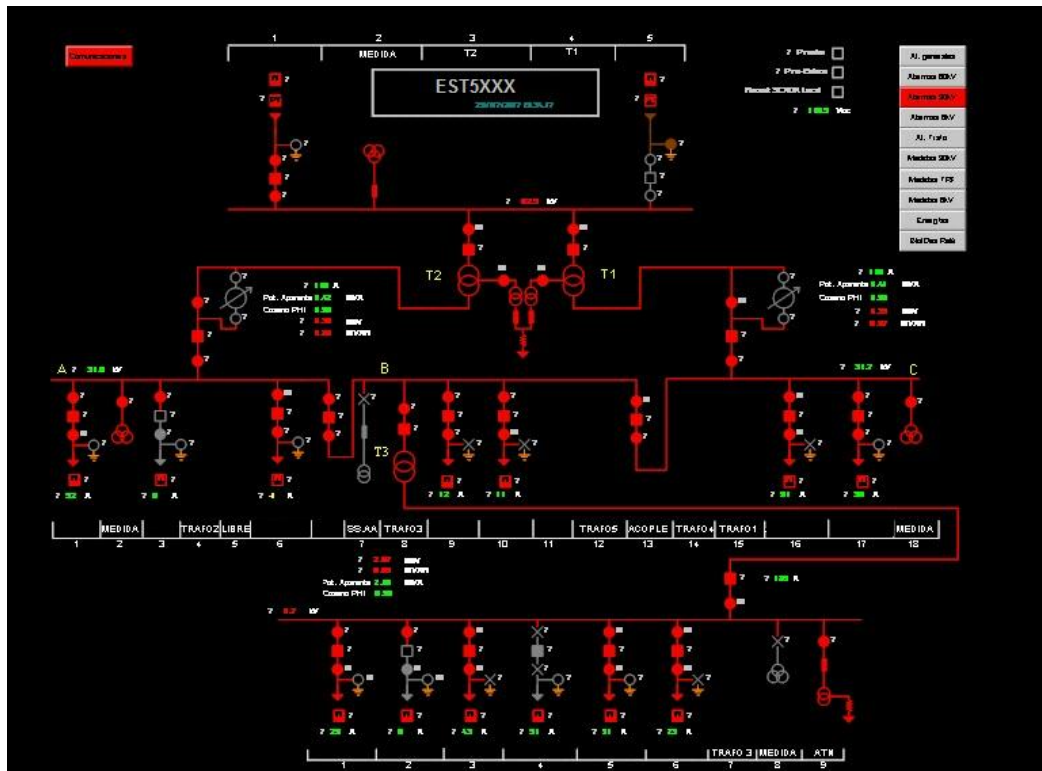


Figura 13.21: Con continuidad seleccionada

Se colorea con el mismo color donde hay corriente circulando, con esto el operario detecta de forma rápida y fácil si existe algún sector desenergizado. En el siguiente diagrama se puede diferenciar la parte amarilla energizada, mientras que la parte en verde está sin corriente:

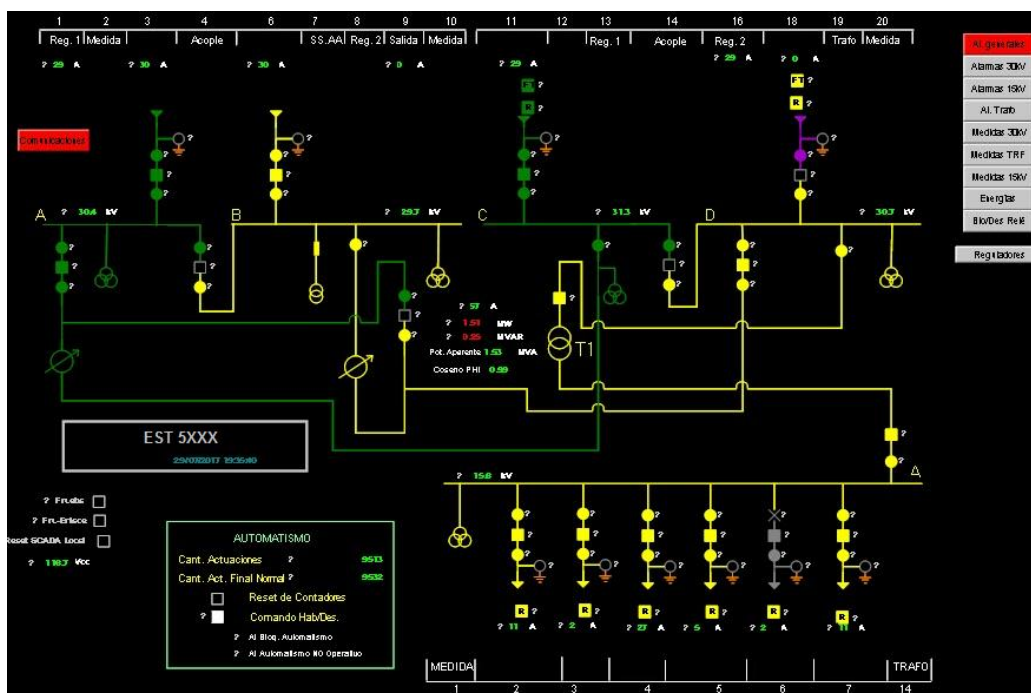


Figura 13.22: Amarillo energizado - Verde sin energizar

13.14 Anexo 14 – Pruebas de Campo

En la siguiente imagen se registró el primer comando probado de manera exitosa. El interruptor se accionó a distancia.



Figura 13.23: Primer comando probado en campo

Imagen de CMD Canelones durante las pruebas:



Figura 13.24: CMD Canelones

13.15 Anexo 15 – Pruebas de Enlace



Figura 13.25: Pruebas de Enlace

Imagen 1: Unifilar SCADA X – Zoom en Medidas SZ5XXXX2

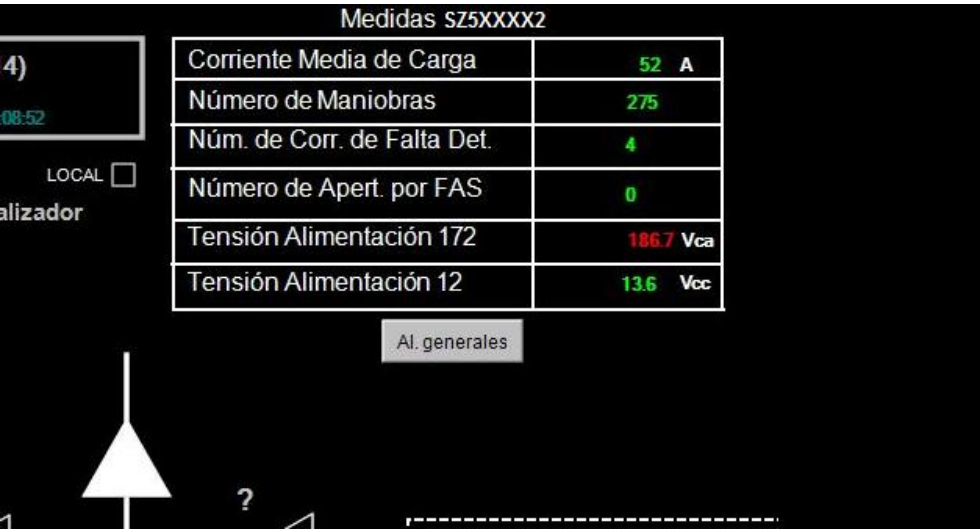


Figura 13.26: Medidas SCADA X

Imagen 2: Unifilar SCADA MIRAGE – Zoom en Medidas SZ5XXXX2

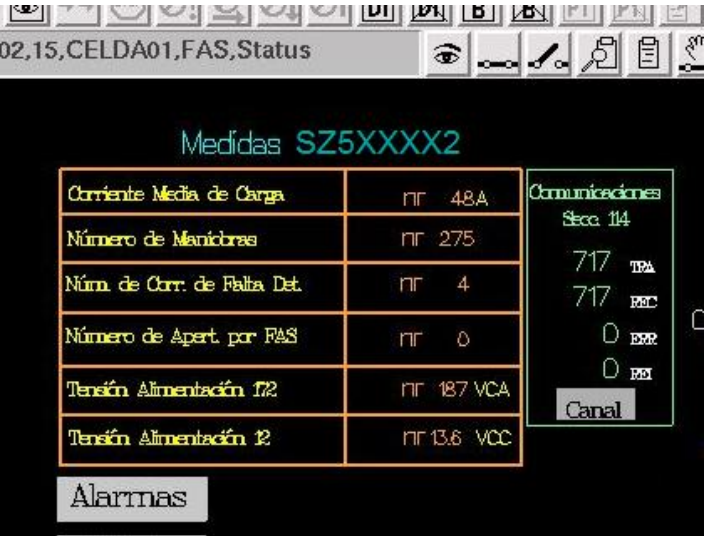


Figura 13.27: Medidas en SCADA Mirage

Como se pueden corroborar todos los valores son iguales, con la excepción de la corriente, primero registra 52A y luego registra 48A. Esto se debe a que la corriente varía todo el tiempo, con la entrada y salida de equipamientos en clientes, pero igual son valores aceptables.

13.16 Anexo 16 – Ejemplo de tabla de errores detectados en corridas

Tipos de Test		Mímico	Observación
Mímicos	Sin Invocación	AUTENGRAW	El mímico AUTENGRAW.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	BOTONERA	El mímico BOTONERA.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	CANELONES	El mímico CANELONES.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	DF142147	El mímico DF142147.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	DF142154	El mímico DF142154.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	DF530001DF530002	El mímico DF530001DF530002.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	EST5044N	El mímico EST5044N.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	EST5074_REN_APERT	El mímico EST5074_REN_APERT.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	EST5T07	El mímico EST5T07.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	FLORIDA	El mímico FLORIDA.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	LLREB	El mímico LLREB.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	MONTA150	El mímico MONTA150.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	MONTB150	El mímico MONTB150.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	NEPTUNIA2	El mímico NEPTUNIA2.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	PRUEBA	El mímico PRUEBA.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	RE510012	El mímico RE510012.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	RE510013	El mímico RE510013.mimx no es invocado desde ningún lugar.

Tipos de Test		Mímico	Observación
Mímicos	Sin Invocación	RE520006	El mímico RE520006.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	RE520010	El mímico RE520010.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	RE530015	El mímico RE530015.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	REG_EJEMPLO	El mímico REG_EJEMPLO.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	SB5111146	El mímico SB5111146.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	SB5111481Y	El mímico SB5111481Y.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	SB5111493Y	El mímico SB5111493Y.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	SB5111497Y	El mímico SB5111497Y.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	SB5111529Y	El mímico SB5111529Y.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	SB5122140Y	El mímico SB5122140Y.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	SZ510004	El mímico SZ510004.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	SZ510007	El mímico SZ510007.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	SZ520007	El mímico SZ520007.mimx no es invocado desde ningún lugar.
Mímicos	Sin Invocación	TAHONA	El mímico TAHONA.mimx no es invocado desde ningún lugar.

Tabla 13-2: Errores Detectados

13.17 Anexo 17 – Reuniones Semanales

Interesados en el proyecto con los cuales nos hemos reunido.

Nomenclatura	Cargo
AM = Andrés Martínez	Estudiante
RM = Renato Moraes	Estudiante
LD = Lidia Dávila	Jefa Departamento Software UTE
PS = Pedro Salvetto	LISI
RB = Rafael Bentancur	Laboratorio de Software
JS = José Szyman	Tutor
TU = Técnico UTE	Técnico disponible UTE

Tabla 13-3: Interesados Proyecto

Listado de Reuniones:

Fecha	Motivo	Participantes
20/07/2016	Información sobre la modalidad de proyecto LISI	PS, AM, RM
28/07/2016	Información sobre la modalidad de proyecto de laboratorio de SF.	RB, AM, RM
01/08/2016	Definimos hacer por LISI	AM, RM
10/08/2016	Se definió que íbamos a presentar Proyecto Migración SCADA, segunda opción software gestión para una ESCO, y tercera opción proyecto LISI	AM, RM
25/08/2016	Evaluar los anteproyectos	AM, RM
09/09/2016	Se presentó bosquejo anteproyectos a Salvetto	Mail
15/09/2016	Realizar anteProyectos en formato LISI y presentarlos	AM, RM
21/09/2016	Nos aprobaron anteproyecto	-
22/09/2016	Primera reunión con Tutor	JS, PS, AM, RM
28/09/2016	Analizar problema en detalle, bosquejo tareas	AM, RM
05/10/2016	Requerimientos	AM, RM
12/10/2016	Alcance proyecto, reunión con UTE.	LD, AM, RM
19/10/2016	Requerimientos	AM, RM
26/10/2016	Estudio factibilidad y bosquejo cronograma	AM, RM
02/11/2016	Gestiones discutir como las haríamos.	AM, RM
09/11/2016	Arquitectura, discusión como se haría el programa	AM, RM
16/11/2016	ITIL - aplicado a gestiones de Lanzamiento y Cambios	AM, RM
23/11/2016	interpretación de archivo propietario de SCADA X	AM, RM
30/11/2016	Planes de GESTION y Modificación de algunos requerimientos	AM, RM
03/12/2016	Validación requerimientos en UTE	LD, AM, RM
07/12/2016	interpretación de archivo propietario de SCADA X	AM, RM
09/12/2016	Encontrar solución a lector SCADA X	TU, AM
14/12/2016	Avance Software	AM, RM
11/01/2017	Se definió hacer Ciclo de Vida y se rearmó cronograma	AM, RM
18/01/2017	Puesta a punto, software y doc	AM, RM
01/02/2017	Avance Software (problemas con celdas poco común)	AM, RM
08/02/2017	Documentación para entregar a Tutor	AM, RM
15/02/2017	Documentación avances a presentar	AM, RM

Fecha	Motivo	Participantes
22/02/2017	Finalizar DOC	AM, RM
02/03/2017	Aprobar plan de <i>testing</i>	LD, AM, RM
08/03/2017	DOCUMENTACION, SEGUIMOS CON PLANES DE GESTION	AM, RM
09/03/2017	Puesta a punto proyecto	JS, PS, AM
16/03/2017	Resolver problemas de integración elementos convertidos	AM, RM
23/03/2017	Resolver problemas en PAT	AM, RM
31/03/2017	definir que detalles estéticos, se corregirán manuales, tipo de fuente superpuesta	AM, RM
07/04/2017	Avances de pruebas de integración, diagramas en Mirage	AM, RM
15/04/2017	Primeros diagramas convertidos	TU, AM
16/04/2017	Pruebas integración con <i>front end</i> y BD	AM, RM
21/04/2017	Resolución botón protocolo 101	AM, RM
29/04/2017	Botones alarma	AM, RM
06/05/2017	Programación Software	AM, RM
14/05/2017	Últimos detalles software	AM, RM
21/05/2017	Definir como empezar la conversión, software pronto	AM, RM
29/05/2017	BUGS detectados -Correcciones Software	AM, RM
05/06/2017	Continuar con la conversión, ajustes manuales en diagramas	AM, RM
13/06/2017	Resumen conversiones, fue un éxito toda la conversión	AM, RM
20/06/2017	Arrancan las pruebas enlace y las coordinaciones de pruebas de campo	TU, AM
21/06/2017	Primeras pruebas de campo, abre y cierra comando	AM, RM
28/06/2017	Pruebas de enlace	AM, RM
05/07/2017	Pruebas de campo	AM, RM
12/07/2017	Comenzar a definir software de pruebas automáticas	AM, RM
19/07/2017	DEFINIR CIERRE PROYECTO	AM, RM
27/07/2017	VER COMO HACER DOC, ENTREGAR EL 1/08/16	AM, RM
29/07/2016	DOCUMENTACION	AM, RM
31/07/2016	DOCUMENTACION	AM, RM

Tabla 13-4: Reuniones Semanales

13.18 Anexo 18 – Archivo Propietario



```
segment{
  did=4192;
  comment="";
  local_figure{
    object{
      type=POLYGON;
      special=NORMAL;
      spuxname="";
      magn_min=-1.0;
      magn_max=-1.0;
      color=3;
      style=SOLID;
      hatch=H_IRRELEVANT;
      border=NO;
      bwidth=2;
      bcolor=8;
      btexture=SOLID;
      xy{ x=0.612262309; y=0.509484589; };
      xy{ x=0.61586231; y=0.509484589; };
      xy{ x=0.61586231; y=0.51128453; };
      xy{ x=0.612262309; y=0.51128453; };
    };
    object{
      type=DYN_KEY;
      id=761;
      keyboxtype=FIXKEY;
      functionkey=0;
      keytype=18;
      reptype=LOCFIG;
      keynr=3955;
      position=0;
      figgrp="0";
      figdt=0;
      attrgrp="202";
      attdt=1;
      ta4sel=NOTA4SEL;
      keymark=NORMKEY;
    };
  };
}
```

Figura 13.28: Archivo Propietario

13.19 Anexo 19 – Diapositiva Heurística para estimación.

Las tareas de gestión sumadas están entre 13% y 29%. Para nuestro desarrollo se estima un 20%.

Algunas Heurísticas					
	10 PF	100 PF	1000 PF	10.000 PF	100.000 PF
Requerimientos	5	5	7	8	9
Diseño	5	6	10	12	13
Codificación	50	40	30	20	15
Testing	27	29	30	33	34
SCM	1	4	6	7	8
Documentación	4	6	7	8	9
Gestión del proyecto	8	10	10	12	12
	100	100	100	100	100

% del esfuerzo por actividad (Estimating Software Costs - pag.179)

Ingeniería de Software II: Estimación 66

Figura 13.29: Heurísticas para estimación

13.20 Anexo 20 – Cronograma del proyecto.

Fase 0 – Inicio Proyecto.

1	▲ Proyecto	235 días	jue 29/09/16	mié 23/08/17
2	▲ Fase 0	25 días	jue 29/09/16	mié 02/11/16
3	ESRE	15 días	jue 29/09/16	mié 19/10/16
4	Gestión de Riesgos	10 días	jue 20/10/16	mié 02/11/16
5	Gestión de la configuración (SCM)	2 días	jue 20/10/16	vie 21/10/16
6	Aseguramiento de la Calidad (SQA)	5 días	jue 20/10/16	mié 26/10/16
7	Gestión de Lanzamiento	15 días	jue 29/09/16	mié 19/10/16
8	Gestión de Cambios	10 días	jue 20/10/16	mié 02/11/16
9	Arquitectura	6 días	jue 20/10/16	jue 27/10/16
10	Cronograma	5 días	jue 20/10/16	mié 26/10/16
11	▲ Fase 1	207 días	jue 03/11/16	vie 18/08/17
12	▷ Iteración 1 - Migrar Elementos	118 días	jue 03/11/16	lun 17/04/17
22	▷ Iteración 2 - Migrar Unifilares	17 días	mar 18/04/17	mié 10/05/17
34	▷ Iteración 3 - Pruebas de Enlace	30 días	jue 11/05/17	mié 21/06/17
45	▷ Iteración 4 - Pruebas en Campo	31 días	jue 22/06/17	jue 03/08/17

Figura 13.30: Cronograma Fase 0

Fase 1 – Iteración 1 – Migrar Elementos

	Nombre de tarea ▼	Duración ▼	Comienzo ▼	Fin
1	▲ Proyecto	235 días	jue 29/09/16	mié 23/08/17
2	▷ Fase 0	25 días	jue 29/09/16	mié 02/11/16
11	▲ Fase 1	207 días	jue 03/11/16	vie 18/08/17
12	▲ Iteración 1 - Migrar Elementos	118 días	jue 03/11/16	lun 17/04/17
13	Análisis Elemento	40 días	jue 03/11/16	mié 28/12/16
14	Desarrollo Elementos	60 días	jue 29/12/16	mié 22/03/17
15	Pruebas Unitarias Elemento	5 días	jue 23/03/17	mié 29/03/17
16	Integración	12 días	jue 30/03/17	vie 14/04/17
17	Pruebas de Integración 1	1 día	lun 17/04/17	lun 17/04/17
18	▲ Gestión	1 día	jue 30/03/17	jue 30/03/17
19	Riesgos	1 día	jue 30/03/17	jue 30/03/17
20	SQA	1 día	jue 30/03/17	jue 30/03/17
21	SCM	1 día	jue 30/03/17	jue 30/03/17
22	▷ Iteración 2 - Migrar Unifilares	17 días	mar 18/04/17	mié 10/05/17
34	▷ Iteración 3 - Pruebas de Enlace	30 días	jue 11/05/17	mié 21/06/17
45	▷ Iteración 4 - Pruebas en Campo	31 días	jue 22/06/17	jue 03/08/17
56	▷ Iteración 5 - Automáticas	36 días	jue 11/05/17	jue 29/06/17
70	▷ Iteración 6 - Puesta en Producción	11 días	vie 04/08/17	vie 18/08/17

Figura 13.31: Cronograma Fase 1 – Iteración 1

Fase 1 – Iteración 2 – Migrar Unifilares

	Nombre de tarea ▼	Duración ▼	Comienzo ▼	Fin
1	▲ Proyecto	235 días	jue 29/09/16	mié 23/08/17
2	▷ Fase 0	25 días	jue 29/09/16	mié 02/11/16
11	▲ Fase 1	207 días	jue 03/11/16	vie 18/08/17
12	▷ Iteración 1 - Migrar Elementos	118 días	jue 03/11/16	lun 17/04/17
22	▲ Iteración 2 - Migrar Unifilares	17 días	mar 18/04/17	mié 10/05/17
23	Migrar Unifilares	2 días	mar 18/04/17	mié 19/04/17
24	Ajustes Manuales	15 días	jue 20/04/17	mié 10/05/17
25	Pruebas de Integración 2	1 día	mar 18/04/17	mar 18/04/17
26	Registro de Bugs	1 día	mié 19/04/17	mié 19/04/17
27	▲ Desarrollo	4 días	jue 20/04/17	mar 25/04/17
28	Resolución Bugs	4 días	jue 20/04/17	mar 25/04/17
29	Liberar nueva versión software	1 día	jue 20/04/17	jue 20/04/17
30	▲ Gestión	1 día	jue 20/04/17	jue 20/04/17
31	SCM	1 día	jue 20/04/17	jue 20/04/17
32	SQA	1 día	jue 20/04/17	jue 20/04/17
33	Riesgos	1 día	jue 20/04/17	jue 20/04/17
34	▷ Iteración 3 - Pruebas de Enlace	30 días	jue 11/05/17	mié 21/06/17

Figura 13.32: Cronograma Fase 1 – Iteración 2

Fase 1 – Iteración 3 – Pruebas de Enlace

	Nombre de tarea ▼	Duración ▼	Comienzo ▼	Fin
1	▲ Proyecto	235 días	jue 29/09/16	mié 23/08/17
2	▷ Fase 0	25 días	jue 29/09/16	mié 02/11/16
11	▲ Fase 1	207 días	jue 03/11/16	vie 18/08/17
12	▷ Iteración 1 - Migrar Elementos	118 días	jue 03/11/16	lun 17/04/17
22	▷ Iteración 2 - Migrar Unifilares	17 días	mar 18/04/17	mié 10/05/17
34	▲ Iteración 3 - Pruebas de Enlace	30 días	jue 11/05/17	mié 21/06/17
35	Coordinar Prueba	10 días	jue 11/05/17	mié 24/05/17
36	Realizar Prueba	10 días	jue 25/05/17	mié 07/06/17
37	Registro Bugs	5 días	jue 08/06/17	mié 14/06/17
38	▲ Desarrollo	5 días	jue 15/06/17	mié 21/06/17
39	Resolucion Bugs	4 días	jue 15/06/17	mar 20/06/17
40	Liberar nueva version software	1 día	mié 21/06/17	mié 21/06/17
41	▲ Gestión	1 día	jue 08/06/17	jue 08/06/17
42	SCM	1 día	jue 08/06/17	jue 08/06/17
43	SQA	1 día	jue 08/06/17	jue 08/06/17
44	Riesgos	1 día	jue 08/06/17	jue 08/06/17
45	▷ Iteración 4 - Pruebas en Campo	31 días	jue 22/06/17	jue 03/08/17

Figura 13.33: Cronograma Fase 1 – Iteración 3

Fase 1 – Iteración 4 – Pruebas en Campo

	Nombre de tarea ▼	Duración ▼	Comienzo ▼	Fin
1	▲ Proyecto	235 días	jue 29/09/16	mié 23/08/17
2	▷ Fase 0	25 días	jue 29/09/16	mié 02/11/16
11	▲ Fase 1	207 días	jue 03/11/16	vie 18/08/17
12	▷ Iteración 1 - Migrar Elementos	118 días	jue 03/11/16	lun 17/04/17
22	▷ Iteración 2 - Migrar Unifilares	17 días	mar 18/04/17	mié 10/05/17
34	▷ Iteración 3 - Pruebas de Enlace	30 días	jue 11/05/17	mié 21/06/17
45	▲ Iteración 4 - Pruebas en Campo	31 días	jue 22/06/17	jue 03/08/17
46	Coordinar Prueba	10 días	jue 22/06/17	mié 05/07/17
47	Realizar Prueba	15 días	jue 06/07/17	mié 26/07/17
48	Registro Bugs	1 día	jue 27/07/17	jue 27/07/17
49	▲ Desarrollo	5 días	vie 28/07/17	jue 03/08/17
50	Resolucion Bugs	4 días	vie 28/07/17	mié 02/08/17
51	Liberar nueva version software	1 día	jue 03/08/17	jue 03/08/17
52	▲ Gestión	1 día	jue 27/07/17	jue 27/07/17
53	SCM	1 día	jue 27/07/17	jue 27/07/17
54	SQA	1 día	jue 27/07/17	jue 27/07/17
55	Riesgos	1 día	jue 27/07/17	jue 27/07/17
56	▷ Iteración 5 - Automáticas	36 días	jue 11/05/17	jue 29/06/17

Figura 13.34: Cronograma Fase 1 – Iteración 4

Fase 1 – Iteración 5

	Nombre de tarea ▼	Duración ▼	Comienzo ▼	Fin
45	▷ Iteración 4 - Pruebas en Campo	31 días	jue 22/06/17	jue 03/08/17
56	▲ Iteración 5 - Automáticas	36 días	jue 11/05/17	jue 29/06/17
57	▲ Desarrollo Software Pruebas	19 días	jue 11/05/17	mar 06/06/17
58	Analisis	5 días	jue 11/05/17	mié 17/05/17
59	Desarrollo	12 días	jue 18/05/17	vie 02/06/17
60	Pruebas	2 días	lun 05/06/17	mar 06/06/17
61	Correr Pruebas Automáticas	10 días	mié 07/06/17	mar 20/06/17
62	Reportar Bugs	1 día	mié 21/06/17	mié 21/06/17
63	▲ Desarrollo	6 días	jue 22/06/17	jue 29/06/17
64	Resolucion Bugs	5 días	jue 22/06/17	mié 28/06/17
65	Liberar nueva version software	1 día	jue 29/06/17	jue 29/06/17
66	▲ Gestión	1 día	jue 22/06/17	jue 22/06/17
67	SCM	1 día	jue 22/06/17	jue 22/06/17
68	SQA	1 día	jue 22/06/17	jue 22/06/17
69	Riesgos	1 día	jue 22/06/17	jue 22/06/17
70	▷ Iteración 6 - Puesta en Producción	11 días	vie 04/08/17	vie 18/08/17
74	▷ Fase 2	14 días	vie 04/08/17	mié 23/08/17

Figura 13.35: Cronograma Fase 1 – Iteración 5

Fase 1 – Iteración 6

	Nombre de tarea ▼	Duración ▼	Comienzo ▼	Fin
1	▲ Proyecto	235 días	jue 29/09/16	mié 23/08/17
2	▷ Fase 0	25 días	jue 29/09/16	mié 02/11/16
11	▲ Fase 1	207 días	jue 03/11/16	vie 18/08/17
12	▷ Iteración 1 - Migrar Elementos	118 días	jue 03/11/16	lun 17/04/17
22	▷ Iteración 2 - Migrar Unifilares	17 días	mar 18/04/17	mié 10/05/17
34	▷ Iteración 3 - Pruebas de Enlace	30 días	jue 11/05/17	mié 21/06/17
45	▷ Iteración 4 - Pruebas en Campo	31 días	jue 22/06/17	jue 03/08/17
56	▷ Iteración 5 - Automáticas	36 días	jue 11/05/17	jue 29/06/17
70	▲ Iteración 6 - Puesta en Producción	11 días	vie 04/08/17	vie 18/08/17
71	Coordinar	2 días	vie 04/08/17	lun 07/08/17
72	Solicitar Aprobación	7 días	mar 08/08/17	mié 16/08/17
73	Migrar Servicio	2 días	jue 17/08/17	vie 18/08/17
74	▷ Fase 2	14 días	vie 04/08/17	mié 23/08/17

Figura 13.36: Cronograma Fase 1 – Iteración 6

Fase 2 – Cierre Proyecto

	Nombre de tarea ▼	Duración ▼	Comienzo ▼	Fin
1	▲ Proyecto	235 días	jue 29/09/16	mié 23/08/17
2	▷ Fase 0	25 días	jue 29/09/16	mié 02/11/16
11	▲ Fase 1	207 días	jue 03/11/16	vie 18/08/17
12	▷ Iteración 1 - Migrar Elementos	118 días	jue 03/11/16	lun 17/04/17
22	▷ Iteración 2 - Migrar Unifilares	17 días	mar 18/04/17	mié 10/05/17
34	▷ Iteración 3 - Pruebas de Enlace	30 días	jue 11/05/17	mié 21/06/17
45	▷ Iteración 4 - Pruebas en Campo	31 días	jue 22/06/17	jue 03/08/17
56	▷ Iteración 5 - Automáticas	36 días	jue 11/05/17	jue 29/06/17
70	▷ Iteración 6 - Puesta en Producción	11 días	vie 04/08/17	vie 18/08/17
74	▲ Fase 2	14 días	vie 04/08/17	mié 23/08/17
75	Cierre Documentación	14 días	vie 04/08/17	mié 23/08/17
76	Analizar Metricas	1 día	vie 04/08/17	vie 04/08/17

Figura 13.37: Cronograma Fase 2

13.21 Anexo 21 – Planilla de horas semanales.

Fecha inicio	Fecha fin	PROCESO	RENATO HORAS	PROCESO	ANDRE S HORAS
13-7-16	20-7-16	Ante proyecto	1	Anteproyecto	1
20-7-16	28-7-16	Ante proyecto	1	Anteproyecto	1
28-7-16	1-8-16	Ante proyecto	2	Anteproyecto	2
1-8-16	10-8-16	Ante proyecto	8	Anteproyecto	7
10-8-16	25-8-16	Ante proyecto	1	Anteproyecto	1
25-8-16	9-9-16	Ante proyecto	3	Anteproyecto	4
9-9-16	15-9-16	Ante proyecto	3	Anteproyecto	3
15-9-16	22-9-16	Reunión tutor	3	Reunión tutor	3
22-9-16	28-9-16	Investigación	6	Investigación	12
28-9-16	5-10-16	Requerimientos	5	Requerimientos	5
5-10-16	12-10-16	Reunión UTE	2	Reunión UTE	2
12-10-16	12-10-16	Gestión proyecto	4	Investigación	7
12-10-16	12-10-16	Investigación	2		
12-10-16	19-10-16	Requerimientos	8	Requerimientos	8
19-10-16	26-10-16	Documentación	8	Documentación	4
26-10-16	26-10-16	Cronograma	14		
26-10-16	2-11-16	ITIL	10	SQA	8
2-11-16	2-11-16	Riesgos	10	ITIL	10
2-11-16	2-11-16	Lanzamiento	12	Arquitectura	16
2-11-16	9-11-16	Cambios	12	Requerimientos	12

Fecha inicio	Fecha fin	PROCESO	RENATO HORAS	PROCESO	ANDRES HORAS
9-11-16	16-11-16	Lanzamiento	10	Investigación	12
16-11-16	23-11-16	Investigación	4	Investigación	12
23-11-16	30-11-16	Requerimientos	11	Requerimientos	10
30-11-16	3-12-16	Requerimientos	6	Requerimientos	6
3-12-16	3-12-16	Riesgos	2		
3-12-16	7-12-16	Investigación	4	Programación	12
7-12-16	9-12-16	Programación	4	Investigación	4
9-12-16	14-12-16	Gestión proyecto	6	Programación	16
14-12-16	11-1-17	Cronograma	12	Programación	14
11-1-17	18-1-17	Documentación	12	Programación	12
18-1-17	1-2-17	Programación	12	Investigación	5
1-2-17	1-2-17	ITIL	5	Programación	12
1-2-17	8-2-17	Documentación	12	Documentación	5
8-2-17	15-2-17	Riesgos	4	SQA	4
15-2-17	22-2-17	Gestión proyecto	12	Documentación	5
22-2-17	2-3-17	Programación	20	Testing	20
2-3-17	8-3-17	Documentación	12	Programación	52
8-3-17	16-3-17	Investigación	4	Programación	48
16-3-17	16-3-17	Gestión proyecto	4		
16-3-17	16-3-17	Programación	12		
16-3-17	23-3-17	Programación	24	Programación	46
23-3-17	31-3-17	Programación	24	Programación	35
31-3-17	7-4-17	Programación	24	Programación	40
7-4-17	7-4-17	Riesgos	2	Riesgos	4
7-4-17	15-4-17	Programación	24	Programación	24
15-4-17	16-4-17	Programación	24	Programación	52
16-4-17	21-4-17	Programación	24	Programación	32
21-4-17	29-4-17	Programación	24	Programación	40
29-4-17	6-5-17	Programación	24	Programación	50
6-5-17	14-5-17	Programación	24	Programación	24
14-5-17	21-5-17	Conversión	30	Conversión	45
21-5-17	21-5-17	Riesgos	2		
21-5-17	29-5-17	Conversión	30	Conversión	32
29-5-17	29-5-17	Testing	10	Programación	24
29-5-17	5-6-17	Conversión	25	Conversión	45
5-6-17	13-6-17	Conversión	25	Conversión	45
13-6-17	20-6-17	Testing	25	Testing	45
20-6-17	21-6-17	Gestión proyecto	12	Testing	45
21-6-17	28-6-17	Testing	12	Testing	45
28-6-17	5-7-17	Gestión proyecto	10	Testing	20
5-7-17	12-7-17	Investigación	4	Programación	40
12-7-17	19-7-17	Gestión proyecto	10	Testing	20

Fecha inicio	Fecha fin	PROCESO	RENATO HORAS	PROCESO	ANDRE S HORAS
19-7-17	27-7-17	Riesgos	4	Programación	10
27-7-17	29-7-16	Documentación	12	Documentación	12
29-7-16	31-7-16	Documentación	24	Documentación	24
31-7-16	7-8-16	Testing	30	Testing	50
7-8-16	14-8-16	Testing	30	Testing	45
14-8-16	21-8-16	Riesgos	4	Testing	45

Tabla 13-5: Horas Semanales

13.22 Anexo 22 – Tabla evaluación Riesgos.

	Setiembre			Noviembre			Enero			Marzo			Mayo			Julio		
Num	PO	I	MR	PO	I	MR	PO	I	MR	PO	I	MR	PO	I	MR	PO	I	MR
R1	0.4	3	1.2	0.4	3	1.2	0.4	3	1.2	0.4	3	1.2	0.2	3	0.6	0	3	0
R2	0.2	3	0.6	0.2	3	0.6	0.2	3	0.6	0.2	3	0.6	0.2	3	0.6	0	3	0
R3	0.6	4	2.4	0.4	4	1.6	0.2	4	0.8	0.2	4	0.8	0.2	4	0.8	0	4	0
R4	0.4	3	1.2	0.2	3	0.6	0.2	3	0.6	0.2	3	0.6	0	3	0	3	0	0
R5	0.2	4	0.8	0.2	4	0.8	0.2	4	0.8	0.2	4	0.8	0	4	0	4	0	0
R6	0.4	3	1.2	0.4	3	1.2	0.4	3	1.2	0.4	3	1.2	0.4	3	1.2	0	3	0
R7	0	4	0	0	4	0	0	4	0	0	4	0	0	4	0	0	4	0
R8	0.2	4	0.8	0.2	4	0.8	0.2	4	0.8	0	4	0	0	4	0	0	4	0
R9	0.2	5	1	0.2	5	1	0.2	5	1	0	5	0	0	5	0	0	5	0
R10	0.2	3	0.6	0.2	3	0.6	0.2	3	0.6	0.2	3	0.6	0.2	3	0.6	0	3	0
R11	0.4	2	0.8	0.2	2	0.4	0.2	2	0.4	0.2	2	0.4	0.2	2	0.4	0	2	0
R12	0.4	3	1.2	0.4	3	1.2	0.2	3	0.6	0.2	3	0.6	0.2	3	0.6	0	3	0
R13	0.2	3	0.6	0.2	3	0.6	0.2	3	0.6	0.2	3	0.6	0	3	0	0	3	0
R14	0.2	5	1	0.2	5	1	0	5	0	0	5	0	0	5	0	0	5	0
R15	0.2	5	1	0.2	5	1	0.2	5	1	0.2	5	1	0	5	0	0	5	0
R16	0.6	4	2.4	0.6	4	2.4	0.4	4	1.6	0.2	4	0.8	1	4	4	1	4	4
R17	0.6	4	2.4	0.6	4	2.4	0.4	4	1.6	0.2	4	0.8	0.2	4	0.8	4	0	0
R18	0.6	4	2.4	0.6	4	2.4	0.4	4	1.6	0.2	4	0.8	0	4	0	0	4	0
R19	0.4	5	2	0.4	5	2	0.4	5	2	0.4	5	2	0.4	5	2	0	5	0
R20	0.4	3	1.2	0.4	3	1.2	0.2	3	0.6	0.2	3	0.6	0.2	3	0.6	0	3	0
R21	0.4	3	1.2	0.4	3	1.2	0.6	3	1.8	0.6	3	1.8	0.4	3	1.2	0	3	0
R22	0.2	2	0.4	0.2	2	0.4	0.2	2	0.4	0.2	2	0.4	0.2	2	0.4	0	2	0
R23	0.2	4	0.8	0.2	4	0.8	0.2	4	0.8	0.2	4	0.8	0	4	0	0	4	0
R24	-	-	-	-	-	-	-	-	-	0.6	4	2.4	0.6	4	2.4	0.2	4	0.8

Tabla 13-6: Evaluación Riesgos

A continuación mostramos a que hace referencia cada nomenclatura y la descripción de cada riesgo.

Nomenclatura	Detalles
PO	Probabilidad de Ocurrencia
I	Impacto
MR	Magnitud del riesgo
R1	Riesgo 1 = Que algún integrante no tenga experiencia en algún actividad puede repercutir en la gestión de algunas áreas.
R2	Riesgo 2 = Si ocurren problemas con el versionado, se pueden perder horas de trabajo realizadas o ser necesarias horas de trabajo para corregir el problema. Esto impacta negativamente en el calendario y el costo del proyecto.
R3	Riesgo 3 = Que los requerimientos se definan mal implicará una carga de trabajo adicional para poder satisfacerlos. Lo que se traduce en un impacto negativo en costos, calendario y alcance del proyecto.
R4	Riesgo 4 = Diseño arquitectónico incorrecto o incompleto
R5	Riesgo 5 = Que a la hora de integrar toda la solución existan complejidades técnicas que no lo permitan o retrase el proyecto.
R6	Riesgo 6 = Si surgiera una dificultad imprevista de carácter técnico, se deberán invertir horas adicionales para solucionarla. Esto puede afectar negativamente el calendario o alcance del proyecto.
R7	Riesgo 7 = que surja otro SCADA ofreciendo otras ventajas
R8	Riesgo 8 = que SCADA Mirage proponga hacer la migración
R9	Riesgo 9 = Si bien SCADA Mirage funciona en otros CMD de UTE, puede existir la posibilidad que algún elemento raro no esté en las posibilidades de Mirage
R10	Riesgo 10 = Ausencias no planificadas de los integrantes del equipo.
R11	Riesgo 11 = Falta de experiencia de los integrantes del equipo con las tecnologías seleccionadas para el desarrollo.
R12	Riesgo 12 = Que algún integrante del equipo por motivos personales no tenga tiempo y repercuta en el proyecto.
R13	Riesgo 13 = Problemas personales entre miembros del equipo va a repercutir en las comunicaciones.
R14	Riesgo 14 = Que UTE cancele la migración por intereses de terceros.
R15	Riesgo 15 = 0
R16	Riesgo 16 = Que no construyan el centro de cómputos en el año del proyecto
R17	Riesgo 17 = Que UTE no tenga pronta la Migración del <i>front end</i>
R18	Riesgo 18 = Que UTE no tenga pronta la Migración de la base de datos alfanumérica.
R19	Riesgo 19 = Por conflictos internos en UTE no se puedan realizar las pruebas en campo
R20	Riesgo 20 = Falta de experiencia en la planificación de proyectos de desarrollo de software.

R21	Riesgo 21 = Errores al estimar esfuerzo de actividades.
R22	Riesgo 22 = Perdida de datos como consecuencia de eliminarlos, extravió o robo de computadoras personales.
R23	Riesgo 23 = Dificultad del diseño de las pruebas automáticas
R24	Riesgo 24 = No cumplir con la documentación

Tabla 13-7: Nomenclaturas Riesgos

13.23 Anexo 23 – Políticas de Lanzamiento.

Revisiones

Registro de modificaciones pasadas

A continuación se muestra un registro de todas las modificaciones hechas anteriormente a este documento.

Versión	Fecha	Nombre	Detalle
1	4/02/17	Creación	1 versión del documento
1.1	11/02/17	Lanzamiento	Cambios en los tipos de lanzamiento
1.2	19/02/17	Horarios	Definición horarios lanzamiento
1.3	20/05/17	Revisión	Cambios varios.
1.4	03/08/17	KPI	Se agregaron KPI'S

Tabla 13-8: Revisiones

Propietario del Documento

La persona mencionada a continuación es el propietario de este documento. No se pueden hacer modificaciones al documento sin pasar por este individuo. Todas las modificaciones hechas al documento serán registradas por el propietario del documento en la sección "Registro de modificaciones anteriores".

< Nombre del propietario del documento >	
Nombre	Renato Moraes
Título profesional:	Ayudante de Ingeniero
Empresa:	Universidad ORT Uruguay
Departamento:	Estudiante
Teléfono:	096 430 010
Ubicación:	
Email:	renatouy@gmail.com

Tabla 13-9: Propietario DOC Lanzamiento

Abreviaturas

Acrónimo	Descripción
CAB	<i>Change Advisory Board</i> (Consejo de Cambios)
CI	<i>configuration item</i> (Ítem de configuración)
CMDB	<i>configuration management database</i> (Base de datos de gestión de la configuración)
COE	<i>Center of Excellence</i> (Centro de excelencia)
DML	<i>definitive media library</i> (Biblioteca de medios definitiva)
CSF	<i>critical success factor</i> (factor crítico de éxito)
ITIL	<i>Information Technology Infrastructure Library</i> (Biblioteca de Infraestructura de Tecnología de Información)
KEDB	<i>known error database</i> (Base de datos de errores conocida)
KPI	<i>key performance indicator</i> (indicador clave de rendimiento)

NER	<i>New Environment Request</i> (phase of the RLC) (Solicitud de nuevo entorno (fase del RLC))
OR	<i>Operational Readiness</i> (Disponibilidad operacional)
PM	<i>project manager</i> (gerente de proyecto)
PMO	<i>project management office</i> (Oficina de gestión de proyectos)
RFC	<i>requests for change</i> (Solicitudes de cambio)
QA	<i>Quality Assurance</i> (phase of the RLC) (Garantía de calidad (fase del RLC))
RLC	<i>release lifecycle</i> (Ciclo de vida de lanzamiento)
RACI	<i>responsible, accountable, consulted, and informed</i> (Responsables, responsables, consultados e informados)
RM	<i>release management</i> (gestión de la puesta en producción)
SDLC	<i>software development lifecycle</i> (Ciclo de vida del desarrollo de programas)
SLM	<i>service-level management</i> (Gestión de nivel de servicio)
SM	<i>Service Management</i> (abreviatura de ITIL Service Management)
UAT	<i>user acceptance testing</i> (Pruebas de aceptación del usuario)

Tabla 13-10: Abreviaturas

A.1 Visión de conjunto

A.1.1 Propósito

La política de lanzamiento sirve como guía para la estrategia general de puesta en producción de la organización. Se ha desarrollado para apoyar la Declaración de Misión de la organización para atender las necesidades de los clientes y usuarios empresariales. En un nivel alto, define las funciones y responsabilidades de la administración de liberaciones dentro del proceso de desarrollo de aplicaciones y sistemas y aclara cómo el ciclo de vida de la versión (RLC) ayuda a asegurar implementaciones listas para la producción. La política de puesta en producción también definirá los estándares de puesta en producción que deben ser utilizados:

- Tipos de lanzamientos y frecuencia.
- Lanzar convenciones de nomenclatura.
- Describe restricciones de tiempo de puesta en producción.
- Entregas de proyectos críticos.

La interrelación con otros procesos de la Biblioteca de Infraestructura de Tecnologías de la Información (ITIL), la base de datos de gestión de la configuración (CMDB) y la biblioteca de medios definitiva (DML).

A.1.2 Alcance

La gestión de lanzamientos es el proceso reconocido que se utilizará para la entrega de todas las versiones de la aplicación y del sistema. Los artefactos de control clave identificados dentro del proceso de puesta en producción se usan para gobernar el proceso de puesta en producción. Estos artefactos de control se alinean con aquellos identificados con el ciclo de vida de puesta en producción acordado. Todas las

aplicaciones bajo el control de la gestión de versiones, incluido el software suministrado por el fabricante, están sujetas a la política de lanzamiento.

El software desarrollado y entregado también está sujeto al proceso de administración de la versión.

Todas las entregas se deben gestionar desde el desarrollo, personalización y configuración, pruebas e implementación hasta la operación en el entorno de producción.

A.1.3 Misión de Administración de Lanzamientos

La administración de lanzamientos lleva a cabo la planificación, diseño, construcción, configuración y pruebas de software y hardware para crear un conjunto de componentes de puesta en producción para el entorno en vivo. Las actividades también cubren la planificación, preparación y programación de un lanzamiento para la migración de Montevideo y otro para Canelones.

La misión de la administración de puesta en producción es establecer un proceso que logre un nivel de eficiencia que permita a la organización desarrollar y entregar productos de una manera que sea:

Repetible: permite medir el proceso.

Controlable: mide lo bien que estamos y las mejoras que podemos hacer.

Escalable: permite el soporte de todos los proyectos futuros.

Esto lleva a:

- Coherencia de la calidad de la publicación,
- Reducción del número de incidentes,
- Mayor eficiencia en la planificación y coordinación de cambios en el entorno de la organización.

Al hacerlo, la administración de lanzamiento permitirá ser capaz de hacer frente al lanzamiento de software de Canelones y Montevideo sin sacrificar la calidad del servicio. Los mecanismos de control dentro de la gestión de puesta en producción ayudan a apoyar este objetivo de una manera eficiente.

Los beneficios son:

- Entrega de una migración de calidad, obteniendo un producto de calidad, sin errores, manteniendo todos los objetos o sus equivalentes de una migración a otra.
- Políticas y procedimientos documentados que aseguren que todas las partes interesadas comprendan los requisitos de lanzamiento y cambio.
- Procesos eficientes, consistentes y repetibles

A.1.4 Estrategia

La gestión de lanzamientos desempeña un papel integral para garantizar la integridad de todo el desarrollo de aplicaciones y sistemas siguiendo un proceso consistente y repetible. Existe una estrecha vinculación entre la gestión de las emisiones y la gestión de proyectos (PMO). Sin embargo, la gestión de lanzamientos no es el ciclo de vida del desarrollo. El PMO proporciona directrices para el ciclo de vida del desarrollo dentro de la gestión de proyectos. El papel de la administración de puesta en producción es ayudar a la administración del proyecto con la planificación e implementación en el entorno de producción. El PMO es responsable de rastrear e informar sobre el estado de los proyectos. La gestión de lanzamientos permite el ciclo de vida del desarrollo, lo hace proporcionando la estructura del proceso y definiendo los procedimientos relacionados. También proporciona plantillas de artefactos que permiten al equipo del proyecto realizar eficientemente las tareas de desarrollo necesarias relacionadas con los entregables de lanzamiento. El ciclo de vida de lanzamiento (RLC) define los artefactos que proporcionan la base para el gobierno del proyecto.

A.1.5 Planificación de Lanzamiento

El proceso de planificación garantiza que se disponga de los recursos necesarios, se hayan realizado pruebas correspondientes y se haya completado la documentación sobre el volumen de negocios a fin de garantizar una ejecución sin problemas. La gestión de lanzamientos desempeña un papel clave, junto con la gestión de proyectos, en el proceso de planificación de recursos. La utilización del ciclo de vida de puesta en producción (RLC) guía el proceso de planificación de la puesta en producción. Las interrelaciones entre la puesta en producción, el cambio y la gestión de la configuración son esenciales para la población efectiva y el uso de la CMDB. La planificación de la puesta en producción efectiva asegura una población precisa de la CMDB, la cual tiene un efecto sobre otros procesos de soporte y entrega de servicios.

A.2 Gestión de Lanzamiento — Contexto Organizacional

A.2.1 Dirección

Una de las principales actividades de la gestión de lanzamientos es definir y administrar el RLC, el cual proporciona guía durante el desarrollo de liberaciones. La definición del RLC y la consulta con los equipos de diseño a través del proceso garantizan la puesta en producción exitosa de las aplicaciones y el hardware asociado. La gestión de lanzamientos trabaja en estrecha colaboración con el PMO para proporcionar capacitación a los procesos del gerente de proyectos y los productos clave a través del ciclo de vida.

A.2.2 Gobernanza

Si bien la función principal de la gestión de la puesta en producción no es la gobernanza, desempeña un papel de gobernanza conjunta con la Gestión de proyectos. El papel primordial es asegurar que la documentación de RLC predefinida se utilice y complete como se requiera. La documentación de RLC proporciona las plantillas y define los

procedimientos para asegurar que la preparación apropiada, la programación y la preparación de la operación se hayan completado antes de la implementación de la versión.

A.2.3 Objetivos

Asegurar que no existan interrupciones del servicio de UTE, asegurando la coordinación eficiente de los cambios en la infraestructura a través de la planificación de la puesta en producción.

Establecer un proceso que eleve la eficacia del proceso de puesta en producción para mantener un nivel consistente de calidad para la entrega de sistemas y software.

Asegúrese de que el proceso de gestión de lanzamiento esté totalmente integrado en todos los grupos de desarrollo y que haya funciones y responsabilidades claramente definidas para cada grupo involucrado en el proceso.

A.3 Tipos de Lanzamientos

A.3.1 Unidad de Lanzamiento

Una unidad de lanzamiento es el segmento de software o hardware que normalmente se suelta en conjunto. Estas unidades pueden variar dependiendo del tipo y nivel de elementos de configuración (CI) dentro de la infraestructura. Las unidades de puesta en producción pueden definirse en forma diferente para sistemas o infraestructuras debido a las desigualdades en las interrelaciones. El propósito es definir el nivel de unidad de puesta en producción más apropiado para cada aplicación o tipo de aplicación.

Al decidir el nivel apropiado para unidades de puesta en producción, el equipo de desarrollo debe considerar lo siguiente:

-  La cantidad de cambios necesarios en cada nivel,
-  la cantidad de recursos y tiempo requerido,
-  la facilidad de implementación,
-  la complejidad de las interfaces con la infraestructura,
-  y la capacidad disponible de los diferentes entornos.

A.3.2 Lanzamiento completo

Todos los CIs de una unidad de puesta en producción son desarrollados, probados e implementados juntos. Dado que una versión completa considera todas las IC dentro de una unidad de puesta en producción, hay menos posibilidades de tener conflictos ya que todas las interfaces de CI y las interfaces serán probadas, aunque no cambien durante el desarrollo de la versión.

A.3.3 Lanzamiento delta

Un lanzamiento delta también se conoce como una puesta en producción parcial o una que sólo incluye uno o muchos pero no todos los elementos de configuración (CI) dentro de una unidad de puesta en producción. Un lanzamiento delta puede tener un riesgo mayor que un puesta en producción completa debido a la falta de pruebas de regresión y vinculación a todas las IC dentro de la unidad de lanzamiento. En algunos casos, un lanzamiento delta es más apropiada que un lanzamiento completo debido al tiempo, la urgencia o la justificación de costos. La evaluación cuidadosa de las IC / interfaces sin cambios o mayores debe hacerse cuando se considera un lanzamiento delta.

A.3.4 Paquete de lanzamiento

Los lanzamientos de paquetes se utilizan cuando hay un enlace entre una unidad de lanzamiento y otros sistemas, interfaces o unidades de lanzamiento. Un lanzamiento de paquete puede consistir en una versión completa y varias versiones delta. Cuando sea apropiado, un paquete de lanzamiento puede reducir el riesgo de software anticuado o incompatible que se utiliza. También puede garantizar la comprobación de compatibilidad entre sistemas e interfaces.

A.3.5 Tipos de lanzamientos aplicados

Se utilizará un lanzamiento completo para la migración del centro de mando de Canelones, luego de ser necesario se utilizara lanzamientos deltas para corrección de errores previos a la migración de Montevideo.

A.3.6 Categorías de lanzamientos

Principales lanzamientos de software y actualizaciones de hardware: grandes cantidades de nueva funcionalidad. Por lo general, una versión principal reemplaza a todas las actualizaciones menores anteriores, versiones y arreglos de emergencia. Estos tipos de lanzamientos pueden incluir cambios en la arquitectura, rediseños parciales y reescrituras.

Versiones de software y actualizaciones de hardware menores: Consiste en pequeñas mejoras, cambios funcionales, arreglos de problemas y actualizaciones técnicas.

Soluciones de hardware y software de emergencia: pequeñas mejoras o mejoras normalmente implementadas en respuesta a un incidente o error conocido. Incluye administración de parches y arreglos de problemas que no pueden ser pospuestos o requisitos personalizados que no pueden ser pospuestos (por ejemplo, cambios regulatorios).

A.3.5 Categorías de lanzamientos aplicadas

Nuestro proyecto constará de una puesta en producción principal y actualizaciones menores posiblemente entre una migración y otra.

A.4 Convenciones de nombre de Lanzamiento

A.4.1 Esquema de nombres

Un esquema de nomenclatura identifica de manera única el CI que representa e incluye un sistema de numeración de versiones que sigue una nomenclatura universal que es consistente para todas las emisiones y en todas las áreas funcionales.

A.4.2 Convenciones estándares de nombres

Todos los lanzamientos seguirán una convención de nomenclatura estándar para todos los componentes lanzados en la infraestructura para asegurar la consistencia y la identificación fácil en cuanto al tipo de puesta en producción y asegurar el control de la versión. Las convenciones de denominación de lanzamiento incluirán el nombre de CI, el número de lanzamiento principal, el número de lanzamiento menor y el número de lanzamiento de emergencia.

Ejemplos de esta convención son: [nombre de la aplicación] X.Y.Z; Por ejemplo, CIMS 1.0.0

Categoría	Número de versión que cambia	Unidad de Lanzamiento Propuesta
Lanzamiento principal	X	Full
Lanzamiento menor	Y	Full
Emergencia	Z	Delta

Tabla 13-11: Convención Lanzamiento

Lanzamiento principal: CRM v1.0, v2.0, etc.

Edición menor: correo electrónico v1.1, v1.2, etc.

Lanzamiento de emergencia: CRM v1.1.1, v1.1.2, etc.

Cuando se desarrolla una nueva versión de una aplicación, el número de versión se debe actualizar inmediatamente para el software en el área de desarrollo. Es responsabilidad del equipo de desarrollo determinar qué tipo de lanzamiento se está desarrollando; Esto determinará la convención de nomenclatura.

En nuestro caso, como mencionamos anteriormente tendremos una puesta en producción principal para la migración de Canelones y luego posiblemente lanzamientos menores antes de la puesta en producción principal del CMD Montevideo.

A.5 Programación de horarios de Lanzamiento

A.5.1 Estrategia

Un lanzamiento puede ser visto como muchos cambios a un servicio. La organización de estos cambios en un horario regular de lanzamiento permite una implementación más controlada y de calidad. Deberá establecerse un horario regular para cada aplicación que permita un tiempo adecuado para planificar a lo largo de todo el RLC. El tipo de puesta en producción y sincronización debe basarse en la complejidad y urgencia de los cambios en la unidad de lanzamiento. Un calendario de lanzamiento mensual para los lanzamientos menores permitirá la recopilación de cambios, así como un método controlado de puesta en producción. Además, un programa de lanzamiento predeterminado establecerá la expectativa de cuándo los usuarios pueden anticipar actualizaciones a sus sistemas.

A.5.2 Programación

Se debe establecer una estrategia de lanzamiento preestablecida para asegurar una planificación y pruebas adecuadas. Se debe designar un calendario de puesta en producción estándar para cada aplicación y debe ser cumplido por todos los equipos funcionales. Cada lanzamiento debe ser designado por la convención de numeración de lanzamiento estandarizada mencionada en la Sección A.4.2.

Es responsabilidad de cada grupo de aplicación determinar el número y la frecuencia de cada tipo de versión. Este programa de lanzamiento pasará a formar parte de la política

de lanzamiento y registrará las versiones de esa aplicación. Esta información se proporcionará para cambiar la gestión y se utilizará para evaluar y conceder la autorización para solicitudes de cambio (RFC).

Para las nuevas aplicaciones, el establecimiento del programa de lanzamiento debe hacerse durante el diseño y desarrollo inicial del sistema y documentarse en el documento de preparación para operaciones. Al establecer el calendario de puesta en producción se deben revisar varias consideraciones.

¿Con qué frecuencia habrá que hacer cambios menores en la solicitud?

¿Cuánto tiempo se tarda en desarrollar y probar los cambios?

¿El cronograma de puesta en producción estándar deja tiempo para la evaluación de la última versión?

¿Con qué frecuencia debe realizarse una revisión completa de la solicitud y qué justificaría un cambio importante?

Las aplicaciones existentes también deben crear un calendario de lanzamiento y pasar a la metodología de lanzamiento en lugar de realizar cambios menores a medida que surjan. Esto permite un mejor seguimiento histórico y documentación de los cambios.

A.5.3 Restricciones de horarios

Al planificar un horario de lanzamiento, debe considerarse los tiempos críticos de negocio para evitar, así como otras restricciones dentro del ambiente. El proceso de gestión del cambio desempeña un papel crítico en la programación de los lanzamientos para asegurar que estos tiempos se eviten. Los calendarios de gestión de lanzamientos y gestión de proyectos deben sincronizarse para reflejar el mismo de fechas de lanzamiento.

En nuestro caso por tratarse de un servicio 365x24 se hará en horario de oficina, horario en el cual la mayor cantidad de operarios están disponibles. Las fechas de puesta en producción se adjuntan en un diagrama de Gantt.

A.6 Biblioteca de medios definitiva (DML)

La biblioteca de medios definitiva (DML) es el término usado para describir una localización segura en la cual las versiones autorizadas definitivas de todos los CI de software son almacenadas y protegidas. En realidad, esta área de almacenamiento puede consistir en una o más bibliotecas de software o áreas de almacenamiento de archivos que deben estar separadas de los entornos de desarrollo, prueba y producción. Contiene las copias maestras de todo el software controlado en una organización.

El DML debe incluir copias definitivas del software adquirido (junto con la documentación de licencia o información relacionada), así como software desarrollado

internamente. Las copias maestras de la documentación del sistema controlado también se almacenarán en el DML en forma electrónica.

Todos los lanzamientos serán controlados a través de la administración de configuración de software y protegidos en la biblioteca a lo largo de sus ciclos de vida.

Título	Bloqueo
Declaración de política	Todos los lanzamientos deben ser controlados a través del procedimiento de gestión de configuración del software y protegidos en la biblioteca a lo largo de sus ciclos de vida. El código fuente sólo puede ser abordado a través de herramientas que protejan la integridad de las fuentes. No se permite el acceso directo a los repositorios. Todo el código fuente está etiquetado y versionado.
Propósito	Proteger los activos de software Asegurar los activos y el IP Asegurar la integridad del código fuente limitando el acceso a los repositorios Evitar problemas de continuidad de negocio relacionados con el desarrollo de software
Alcance de la política	Todo TI
Beneficios	Un bloqueo: Mejorar todo el software desarrollado mediante procedimientos claros de <i>check-in</i> y <i>checkout</i> Limitar el acceso no autorizado a cero, aumentando así la integridad Mejorar las capacidades de continuidad del negocio a través de copias de seguridad centralizadas Permitir un mejor seguimiento de las migraciones de código a través de su ciclo de vida, saber quién está tocando qué activo a qué hora y por qué
Procedimientos y modelos relacionados	Herramientas: <Enumere todas las herramientas si es aplicable>
Fecha de vigencia	TBD

Tabla 13-12: Administración de configuración

A.7 Ciclo de vida de Lanzamiento

A.7.1 Visión de conjunto de RLC

Planificar, probar y documentar estas actividades son los principales objetivos de la gestión de lanzamientos. Estos objetivos pueden lograrse siguiendo un proceso riguroso, consistente y repetible que no sea involuntario. Esto ayuda a garantizar que se recopilan los requisitos adecuados, que la aplicación se comprueba, que esté operacionalmente lista para la producción y tenga una implementación fluida en el entorno de producción.

El RLC no es un ciclo de vida de desarrollo de sistemas de software (SDLC). Más bien, el RLC complementa el SDLC y guía la puesta en producción del software / sistema asegurando que todas las versiones hayan sido planeadas, probadas y programadas.

A.7.2 Componentes de RLC

El RLC está alineado con el ciclo de vida del desarrollo. El PMO desempeña un papel activo en el RLC desde el punto de vista de la integración, a través de los directores de proyectos y los equipos de desarrollo.

El RLC consta de siete fases diferentes; Cada fase se relaciona con las otras fases que conducen a un lanzamiento listo para la producción. Las fases RLC son:

Iniciación

Nueva Solicitud de Entorno (NER)

Aseguramiento de la Calidad (QA)

Configuración y construcción de la puesta en producción.

Preparación Operacional (OR)

Implementación

Post-implementación

Cada fase requiere la creación de entregables específicos que ayudarán a asegurar que cada puesta en producción esté diseñada, construida, probada y operativamente preparada antes de su implementación. Cada fase concluye con una revisión final que debe ser aprobada antes de pasar a la siguiente fase. Esta revisión coincide con la fase descrita en el ciclo de vida del desarrollo.

Además, el RLC se basa en las siete principales actividades definidas en el marco de gestión de lanzamiento de ITIL. El RLC adapta los conceptos teóricos de ITIL e incorpora la aplicación práctica de esos conceptos en el entorno. Cada una de estas fases está relacionada con las otras fases, y juntas, conducen a una puesta en producción.

Las actividades de gestión de lanzamiento de ITIL son:

Planificación

Desarrollo y diseño

Construir y configurar

Liberar la aceptación

Planificación del despliegue

Comunicación, preparación y capacitación

Distribución e instalación

A.7.3 Planificación del Lanzamiento

La administración de lanzamiento juega un papel integral en asegurar la integridad de todos los desarrollos de aplicaciones y sistemas siguiendo un proceso de planificación consistente y repetible. El proceso de planificación garantiza que los recursos necesarios están disponibles, se han completado las pruebas y se ha completado la documentación sobre el volumen de negocios.

Un plan de alto nivel es requerido para cada lanzamiento y debe incluir todos los planes de prueba, criterios de aceptación y entregables de lanzamiento.

A.7.3.1 Iniciación

La iniciación es la fase de entrada en el RLC y comienza después de que el proyecto haya sido aprobado. Varios documentos, tales como una propuesta de concepto, *business case*, análisis de factibilidad, la carta de proyecto, y plan de proyecto y requisitos de negocio se completará antes de entrar en esta fase. Se utiliza una lista de verificación de iniciación para entrar en esta fase. El propósito principal de la fase de Iniciación es asegurar que la aprobación del proyecto y presupuesto se ha dado y que las necesidades de negocio y los entregables han sido claramente definidos.

A.7.4 Diseñar, Desarrollar, Construir y Configurar

A.7.4.1 Nuevo proceso de solicitud de entorno (NER)

El nuevo proceso de solicitud de entorno garantiza que la arquitectura de la versión se configure para proporcionar el uso más eficiente de los recursos de hardware, red, almacenamiento y base de datos. Define un mecanismo para determinar si la infraestructura existente puede ser utilizada o si la adquisición de nuevos componentes es necesaria. El proceso define un procedimiento para navegar por el proceso de adquisición. Las salidas de proceso adicionales incluyen flujo de datos, flujo de proceso y diagramas de vista general de la arquitectura, que pueden utilizarse en el soporte futuro de la aplicación.

Los lanzamientos que no están dentro del proceso NER también están bajo el control de los procesos de cambio y lanzamiento.

Durante el proceso NER, el ciclo de vida de desarrollo de software (SDLC) está en curso en paralelo con el proceso NER.

A.7.4.2 Instalación de Aplicaciones

La instalación es el procedimiento que describe la interfaz entre el ciclo de vida del desarrollo de software y la gestión de lanzamientos. Es responsabilidad de la administración de lanzamiento asegurarse de que todos los componentes de código fuente y lanzamiento confiables estén protegidos en la biblioteca de medios definitiva (DML) antes de la implementación. Cuando el entorno se construye de acuerdo con los requisitos y se ha validado, la aplicación se puede instalar con seguridad.

El proceso NER continúa en paralelo con la instalación de la aplicación.

A.7.5 Aceptación del Lanzamiento

Los procedimientos de Aceptación y Prueba de Lanzamiento están cubiertos bajo los procesos de QA y OR.

A.7.5.1 Aseguramiento de la calidad (QA)

Con el fin de proporcionar el mejor producto preparado para la producción, la puesta en producción necesita pasar por un proceso de QA definido. Los casos de prueba llevados a cabo necesitan correlacionarse directamente con los requisitos del negocio. El proceso de control de calidad también debe incluir pruebas de rendimiento, así como pruebas de aceptación del usuario (UAT). El proceso de prueba puede identificar errores dentro de la puesta en producción. El equipo del proyecto debe revisar estos errores y determinar las soluciones. El RLC requiere que el Reporte de Aceptación de QA sea revisado para asegurar que el grupo de QA ha endosado la puesta en producción o ha identificado apropiadamente bugs potenciales que pueden causar problemas en la producción. El informe también asegura que el equipo del proyecto ha reconocido el riesgo asociado.

A.7.5.2 Preparación de Operaciones (OR)

La Preparación Operacional (OR) es el proceso final antes de implementar la puesta en producción. El enfoque del proceso es asegurar que todos los componentes de la puesta en producción han sido probados, revisados y documentados. Esto se logra con la revisión final de la preparación de las operaciones. La revisión sirve como una lista de comprobación de los artefactos necesarios y ayuda a asegurar que el lanzamiento está listo para su promoción al entorno de producción. La documentación incluye información de soporte de puesta en producción, todos los errores y soluciones conocidos, medidas de continuidad del servicio, documentación de nivel de servicio, planes de capacitación e implementación y pruebas técnicas de preparación operativa.

Las técnicas de preparación operativa son un proceso riguroso diseñado para probar y asegurar que la infraestructura, la red, el *front end* y las bases de datos funcionen de acuerdo con los requerimientos de UTE. También asegura que las herramientas de monitoreo están en su lugar y que todas las actualizaciones de los servicios y las ICs están incluidas en la base de datos de la configuración.

A.7.6 Plan de Comunicación, Preparación y Capacitación

El objetivo del Plan de Comunicación es informar a todas las partes interesadas de todas las cuestiones pertinentes a la publicación. Es responsabilidad del equipo del proyecto asegurarse de que las entregas listadas a continuación se incluyen con el comunicado. Estos artefactos se incluyen como parte de los procesos RLC.

Plan de Soporte y escalonamiento / Documento de cambio (proceso OR)

Planes de implementación y comunicación (proceso de implementación)

A.7.7 Instalación

El procedimiento de Instalación debe ser planificado y documentado para implementar la puesta en producción. Una planificación adecuada de la instalación es crucial para el éxito de la puesta en producción.

A.7.7.1 Implementación

La gestión de lanzamiento trabaja de la mano con la administración del cambio para asegurar que los pasos requeridos para una implementación exitosa se completan antes de recibir la aprobación para implementar. Estos pasos incluyen la comunicación, la implementación, la validación, los criterios para el éxito y los planes de retroceso. Estos pasos se capturan dentro de la RFC. Una vez que el RFC ha sido completado y aprobado por la Junta de Asesoría de Cambios (CAB), el lanzamiento puede pasar a la producción de acuerdo con el plan de implementación.

A.7.7.2 Post-Implementación

El proceso posterior a la implementación es el proceso final dentro del RLC. El objetivo principal de este proceso es el cierre del proyecto. Durante esta fase, todos los documentos del proyecto se recopilan, revisan y archivan. Además, una reunión posterior a la revisión de implementación (PIR) puede ser necesaria para recopilar las lecciones aprendidas y obtener conocimiento para mejorar futuras versiones.

A.7.8 Retención de la documentación de *Release*

La documentación RLC generada durante el proceso de desarrollo se almacena en el repositorio de documentos. Esto permite un repositorio central para esta documentación.

A.8 Roles y Responsabilidades

Se suministra una matriz RACI (encargado, responsable, consultado e informado) para proporcionar las funciones y responsabilidades de la administración de la puesta en producción.

Rol			Descripción
R	<i>Responsible</i>	Encargado	Este rol corresponde a quien efectivamente realiza la tarea. Lo más habitual es que exista sólo un encargado (R) por cada tarea; si existe más de uno, entonces el trabajo debería ser subdividido a un nivel más bajo, usando para ello las matrices RASCI.
A	<i>Accountable</i>	Responsable	Este rol se responsabiliza de que la tarea se realice y es el que debe rendir cuentas sobre su ejecución. Sólo puede existir una persona que deba rendir cuentas (A) de que la tarea sea ejecutada por su responsable (R).
C	<i>Consulted</i>	Consultado	Este rol posee alguna información o capacidad necesaria para realizar la tarea.
I	<i>Informed</i>	Informado	Este rol debe ser informado sobre el avance y los resultados de la ejecución de la tarea. A diferencia del consultado (C), la comunicación es unidireccional.

Tabla 13-13: Roles y Responsabilidades.

Ocupaciones	Gerente de Servicio	Gerente de lanzamiento	Coordinador de lanzamiento	Gerente de Proyecto
Diseño general y mantenimiento continuo del proceso, incluyendo el RLC, los artefactos y la política de lanzamiento	R	A	C	I
Realizar una evaluación continua del proceso de oportunidades de mejora	C	A/R	C	I
Garantizar que las personas, los procesos y la tecnología estén alineados	A	R	I	
Establecer indicadores clave de desempeño (KPI) que midan los factores críticos de éxito	R	A	C	I
Garantizar el objetivo clave de RM es utilizar métodos y artefactos estándar en la	R	A	C	

preparación, construcción, pruebas e implementación de cambios con un impacto mínimo.				
Producir y monitorear KPIs y factores críticos de éxito (CSFs) para asegurar las eficiencias maximizadas	C	A/R	C	I
Actuar como punto focal operacional para las propuestas de modificaciones para liberar las actividades de gestión	R	A/R		
Actuar como punto central de contacto para el equipo de proceso		A/R		
Soporta el rol de Gestor de lanzamiento en la realización de actividades	C	I	A/R	
Mantener y rastrear la información relativa a las emisiones en todo el RLC, incluyendo el estado, los artefactos de lanzamiento y el cierre		A/R	R	I
Producir informes por tipo de publicación		A/R	R	
Producir y mantener comunicaciones apropiadas a todas las partes interesadas con respecto a los lanzamientos	C	R	C	A
Coordinar las actividades en todo el RLC	C	R	R	A
Soporte administrativo para el gestor de versiones y el proceso	C	C	A/R	
Asegúrese de que los requisitos para el RLC específico se planifiquen y ejecuten para cumplir con el calendario de puesta en producción.		I	C	A/R
Finalización de los artefactos de lanzamiento descritos en RLC	I	R	C	A/R
Cerrar sesión en todos los puntos de control RLC	I	A	C	A/R
Asegura que las copias maestras de todo el software estén protegidas en el DML	I	A	I	R

Asegura la puesta en producción exitosa de todo el software y hardware	I	A	I	R
Planificar el lanzamiento y la implementación	I	A	I	R
Ejecutar la transferencia de los entregables (soporte, capacitación y actualización de la información del servicio) al <i>Service Desk</i>	I	A	I	R
Comunicar, preparar y capacitar a los usuarios finales	I	A	I	A/R
Verificar los resultados de las pruebas y los entregables	A/R	C		

Tabla 13-14: Matriz RACI.

Note: R – Encargado, A – Responsable, C – Consultado, I – Informado

A.9 Relaciones de proceso

A.9.1 Gestión de problemas

Los errores conocidos documentados y las soluciones alternativas asociadas ayudarán a resolver incidentes. Esta información puede utilizarse para justificar la identificación y el estudio ulterior del problema. La documentación de los resultados de las pruebas y la información sobre la preparación de las operaciones también se pueden revisar durante el proceso de resolución de problemas y utilizarse para análisis de causa raíz e identificación de errores conocidos.

A.9.2 Gestión del cambio

La gestión del cambio desempeña un papel clave en el proceso de gestión de lanzamiento al ayudar a garantizar que la programación de la puesta en producción se complete de manera expedita pero controlada. Todas las fases y procesos del RLC conducen a la RFC y la aprobación del cambio. La administración del cambio también ayuda a garantizar que las IC modificadas estén documentadas en el CMDB.

Aunque la gerencia de lanzamiento supervisa los detalles de la implementación de un cambio, permanece bajo el control y la autoridad de la gerencia del cambio.

A.9.3 Gestión de la configuración

Antes de que se pueda desarrollar una puesta en producción, debe haber una comprensión de la aplicación existente, el sistema, los ítems de configuración asociados, y la relación con otras unidades de lanzamiento. Esta información se almacena en la CMDB y se gestiona mediante la gestión de la configuración. A medida que se desarrolla

la nueva aplicación, la configuración de la aplicación se captura y documenta a través del proceso RLC. Esta información se utiliza posteriormente para rellenar el CMDB.

En colaboración con la administración de nivel de servicio, el catálogo de servicios también se mantiene dentro de la CMDB (como CI) y se debe actualizar a medida que cambia el servicio o la aplicación.

A.9.4 Gestión del nivel de Servicio

En el desarrollo de la versión, los requisitos de disponibilidad y rendimiento del sistema, los niveles de servicio de soporte y las ventanas de mantenimiento son determinados por el cliente. Estas consideraciones deben entenderse por lo que el tamaño adecuado y la arquitectura de la aplicación se pueden diseñar para satisfacer los niveles deseados. La administración del nivel de servicio mantiene los niveles de disponibilidad y servicio de base, así como los horarios y costos asociados con las variables no básicas. La gestión del nivel de servicio también funcionará con la gestión de la configuración para garantizar que el catálogo de servicios se actualice para reflejar los cambios en los servicios o niveles de servicio.

A.9.5 Gestión de seguridad informática

Una de las actividades requeridas durante el RLC es asegurar la seguridad de la empresa mediante la comprensión completa de si hay riesgos o vulnerabilidades relacionados con la aplicación o la conectividad. En el RLC, se requiere una evaluación y auditoría de seguridad para asegurar el cumplimiento de las políticas corporativas de seguridad y para determinar vulnerabilidades potenciales.

A.9.6 Gestión de Continuidad de Servicios de TI

La continuación y restauración del servicio está documentada dentro del RLC. Esta documentación se convierte en parte del Plan general de continuidad de servicios de TI.

A.10 Indicadores de rendimiento (KPIs)

El atributo principal de nuestro proyecto es la testeabilidad, nuestro proceso es basado fuertemente en las pruebas, por lo tanto los indicadores que se han trazado referentes a las pruebas.

KPIs
Cantidad de pruebas realizadas en campo, dadas las dificultades técnicas a la hora de las coordinaciones, queremos testear entre un 5 y un 10% de las instalaciones.
Cantidad de errores en cada corrida de pruebas automáticas, este KPI debe estar en 0.
Cobertura de las pruebas automáticas y de enlace, este KPI debe estar en el 100%

% de errores resueltos de las corridas automáticas, este KPI tiene que estar el 100%, antes de la puesta en producción.

Tabla 13-15: KPIs

A.11 Informes de gestión

Los informes de gestión de puesta en producción sirven para medir y monitorear la calidad del proceso en comparación con las métricas establecidas como KPIs para asegurar que sean eficientes, eficaces y adecuadas para su propósito.

A.11.1 Estructuras de reuniones y control

A.11.1.1 Consejo Consultivo sobre Cambios (CAB)

El CAB autoriza la puesta en producción y su contenido. Revisa la preparación de la planificación, el contenido de la versión y el análisis de impacto. Decide avanzar al otorgar la aprobación para su implementación.

A.11.1.2 Revisión de Lanzamiento

Esta revisión examina la preparación final antes de la implementación de un comunicado.

Todos los artefactos de puesta en producción requeridos se evalúan para su finalización. Los resultados serán reportados al gerente de cambios, quien finalmente autorizará el lanzamiento para su implementación.

A.11.1.3 Revisión Pos lanzamiento

Esta revisión está diseñada para abordar todas las cuestiones y atar cabos sueltos en la producción, y para mejorar la preparación para futuras versiones.

El resultado incluirá un documento de revisión con elementos de acción y planes de mejora. Este documento de revisión se enviará a la dirección del cambio para incluirlo en la revisión posterior a la implementación para la gestión del cambio.

Luego del lanzamiento de la migración en Canelones será muy importante esta revisión, para mejorar el proceso de la migración mayor en Montevideo.

Anexo A: Indicadores clave de rendimiento (KPI)

1. Software y hardware de mejor calidad:

- a. Porcentaje de reducción en el uso de software y hardware que no han superado los controles de calidad requeridos
- b. Porcentaje de reducción en el software instalado no tomado del DML
- c. Reducción porcentual de la reversión no autorizada a versiones anteriores

3. Implementación de lanzamientos rápidamente (necesidades impulsadas por las empresas) y con precisión:

- a. Reducción porcentual en los errores de compilación
- b. Porcentaje de implementación de lanzamientos en todos los sitios, incluidos los remotos, a tiempo
- c. Reducción porcentual del número de emisiones urgentes
- d. Reducción porcentual de las emisiones causantes de incidentes
- e. Reducción en el porcentaje de liberaciones implementadas sin ser probado
- e. Reducción del porcentaje de emisiones urgentes o de alta prioridad solicitadas sin la justificación apropiada del caso de negocio

4. Diferencias costo-efectivas

- a. Mayor porcentaje de lanzamientos contruidos e implementados según el calendario
- b. Porcentaje de emisiones creadas e implementadas dentro del presupuesto
- c. Reducción de la indisponibilidad del servicio causada por los lanzamientos
- d. Reducción porcentual de las emisiones liberadas
- e. Reducción porcentual de las liberaciones fallidas
- f. Reducción porcentual de las horas extraordinarias de puesta en producción debido a una mejor planificación
- g. Reducción del "coste" de las liberaciones fallidas

Anexo 8.2 - Lista de Verificación de Entregables del Lanzamiento

Fase deliverable	Gerente de lanzamiento -Checklist	Notas
Iniciación		
Requerimientos de Estimación	x	
Requisitos de la empresa	x	
Lista de Verificación de Entregables de Puesta en producción	x	
Liberar el plan del proyecto	x	
Configuración y construcción de lanzamiento		
Requerimientos funcionales	x	
Plan de prueba maestro	x	
Acuerdo de nivel de servicio (SLA)	x	
Solicitud de nuevo entorno (NER)		
Diseño de Arquitectura de Aplicaciones	x	
Diseño de Detalle Infraestructural	x	
Especificaciones funcionales	x	
Procedimiento de construcción	x	
Plan de ejecución de pruebas	x	
Aseguramiento de la Calidad / Pruebas		
Informe de progreso sobre la garantía de calidad (QA)	x	
Aceptación de la garantía de calidad (QA)	x	
Disponibilidad operacional		
Prueba de Preparación para la Operación	x	
Plan de Continuidad del Servicio de TI	x	La pieza de software será usada en puesta en producción CMD MDEO.
Implementación		
Plan de implementación y respaldo	x	
Notificación de Implementación del Proyecto		
Control de Cambios (Formulario FRC / Online, Consejo Asesor de Cambios [CAB], Programa)	x	
Post-implementación		
Revisión posterior a la implementación y entrega de los resultados del proyecto		
Lecciones aprendidas		

Tabla 13-16: Verificación Entregables.

Nota: en la documentación del proyecto están cumplidos todos los puntos seleccionados. Si bien los diagramas están migrados y probados, no se ha efectuado aun la puesta en producción, se está a la espera de los servidores. Una vez en producción el nuevo sistema se hará las revisiones post-implementaciones.

13.24 Anexo 24 – Procedimiento de *Rollback*

La puesta en producción se hará mediante un lanzamiento *Big Bang*, para el CMD de Canelones y Florida, esto quiere decir que una vez generados y probados todos los unifilares para el sistema Mirage, se los pondrá en producción todos en la misma instancia. Esto implica redireccionar el enlace de los servidores del SCADA X a los servidores del SCADA Mirage

Para eso se debe tener ambos centros de cómputos operativos, en la siguiente imagen se puede ver un diagrama de la situación antes del cambio:

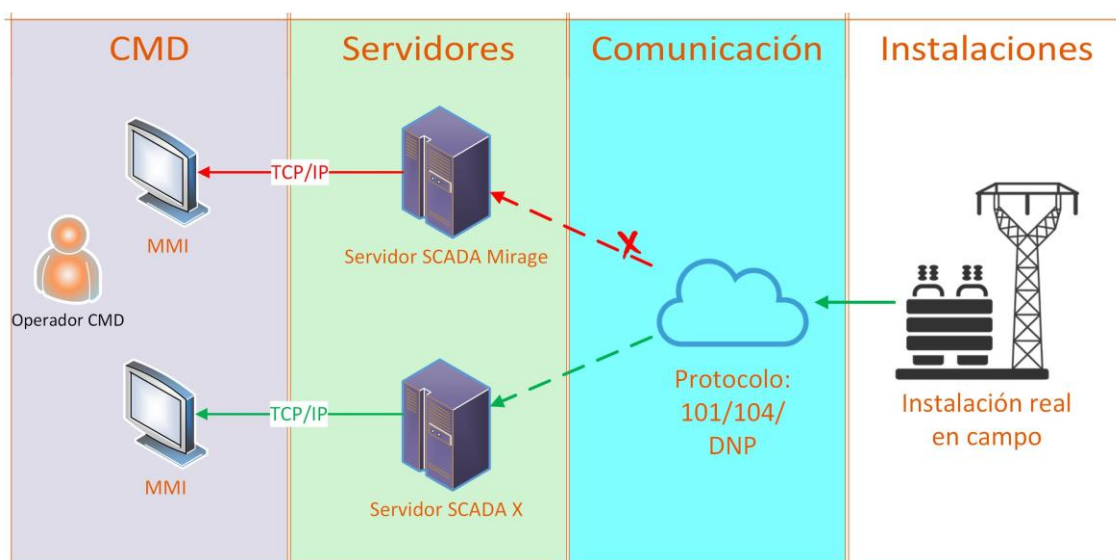


Figura 13.38: Situación actual, SCADA X funcionando.

En la imagen 13.38 se observa que el enlace en esta situación es direccionado a los servidores LINUX, donde corren el SCADA X.

En la migración lo que se hará es bajar el enlace y levantarlo desde el nuevo centro de cómputos con servidores Windows.

En la figura 13.39 se puede observar un diagrama de la situación luego de la migración, donde el sistema SCADA Mirage está a funcionamiento a pleno y el SCADA X está de *backup*.

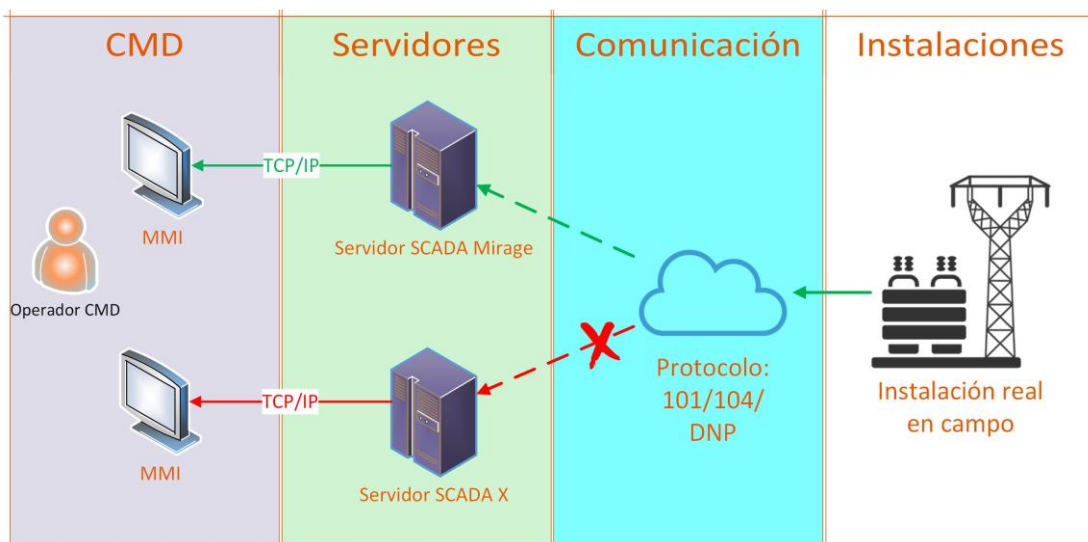


Figura 13.39: Situación futura, funcionamiento a pleno del CMD Canelones en SCADA Mirage con el SCADA X de Backup

Inmediatamente a levantar el estará operativo el nuevo sistema.

Se verificara que dicho sistema reciba medidas, alarmas y estado de las variables. Mientras que los elementos comandables a distancia serán verificados que funcionen correctamente durante la operativa normal del CMD, o sea que se hará a demanda de los operadores durante el uso de los mismos.

Se dará como exitoso la migración en esta primera instancia si los parámetros recibidos son los correctos, no se efectuó la baja involuntaria de suministros eléctricos y las alarmas que reporta el sistema son correctas. Pero recién se dará por finalizado la migración cuando haya pasado dos meses con un funcionamiento normal.

Por lo tanto durante dos meses permanecerán activos los servidores del SCADA X para una eventual marcha atrás.

Se planifican dos tipos de medidas de contingencia, según las características de los problemas, para ello se plantea dos escenarios.

Escenario 1: Se detectan errores en pocas instalaciones. Se bajará el enlace en Mirage y se levantará en SCADA X únicamente los enlaces de las instalaciones con problemas, se verificara si los problemas ya existían o si son propios de la migración. Se disparará la solicitud de cambio para corregir dichos problemas, independiente del origen de los mismos.

En la figura 13.40 se contempla el escenario 1:

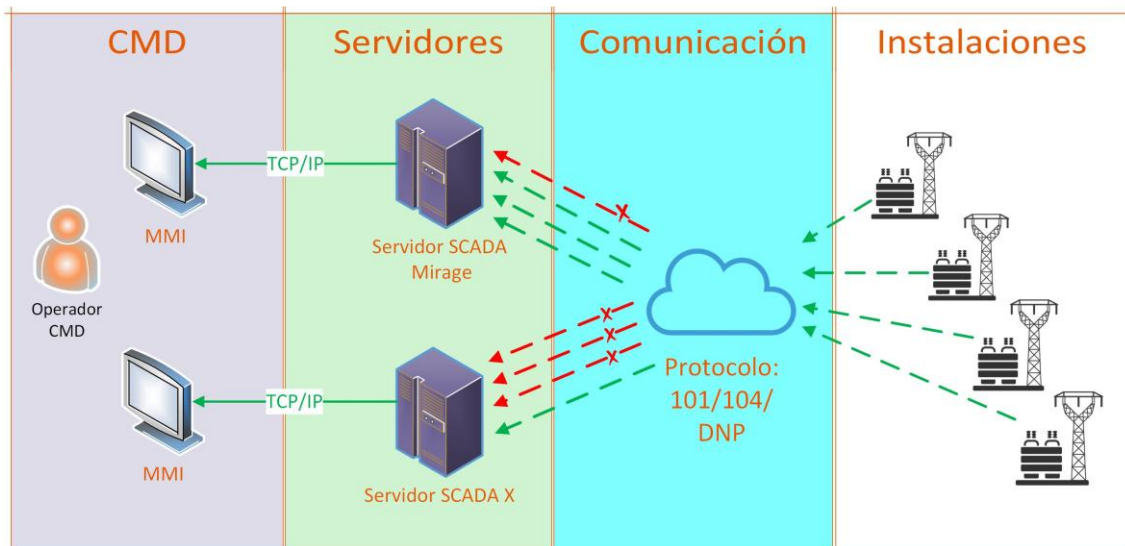


Figura 13.40: Sistema funcionando de forma parcial en ambos SCADA.

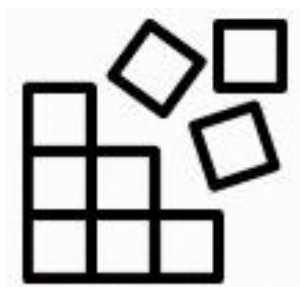
Escenario 2: Los errores están replicados en muchas instalaciones y podrían llegar a causar problemas en la red. En este caso se volvería a la situación inicial en la cual solamente funciona el SCADA X, ver figura 13.38. Y se disparará la solicitud de cambio pertinente.

En ambos escenarios el procedimiento de *rollback* es sencillo, ya que ambos sistemas únicamente operan la red, no persisten datos, entonces no es necesario realizar migración de datos de ningún tipo.

13.25 Anexo 25 – Manual de Usuario SegmentToMimx

SegmentToMimx

Manual de usuario



Introducción

SegmentToMimx se trata de un software construido para migrar mímicos de instalaciones eléctricas entre sistemas SCADA, más precisamente del sistema SCADA X al SCADA Mirage. El nombre del software se debe la correspondencia entre las estructuras con las cuales los SCADA representan los mímicos, segmentos en SCADA X y archivos de extensión .mimx por Mirage. Estos archivos generados son posteriormente modificados por el usuario mediante otro software, el Editor Mirage, con el fin de hacer correcciones estéticas o ajustes necesarios para el SCADA Mirage que no se pueden hacer de forma automática. Por eso se dice que la migración es semiautomática.

Esta migración grafica está enfocada a los administradores de los sistemas SCADA de Distribución de UTE, más precisamente a los integrantes de la unidad de Software de Telecontrol. Dichos usuarios tienen presente los pormenores a la hora de realizar el diseño gráfico de una instalación eléctrica, es más, serán quienes posean el código de este software para sus futuras modificaciones.

Finalmente este manual intentará cubrir aspectos de la instalación, configuración y ejecución del software SegmentToMimx.

Instalación

Requisitos:

- Windows 7 o superior.
- *Framework* .Net 4.0 o superior.
- Cliente Oracle 8 o superior.

Una vez cumplidos con los requisitos previos, alcanza con ubicar en un mismo directorio los archivos *SegmentToMimx.exe*, *SegmentToMimx.exe.config* y *QUERY.Acceso.dll*, como lo muestra la siguiente imagen:

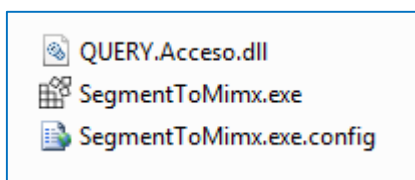


Figura 13.41: Archivos necesarios para la ejecución

- **QUERY.Acceso.dll**: Librería provista por UTE para facilitar el acceso a las base de datos SQL Server y Oracle.
- **SegmentToMimx.exe**: Ejecutable del software de migración.
- **SegmentToMimx.exe.config**: Archivo de configuración del software.

Configuración

A continuación se describe el archivo *SegmentToMimx.exe.config* para su correcta configuración:

- Cadenas de conexiones a bases de datos:
 - **name="PIWEB1"**: Define la cadena de conexión a la base SQL Server, dicha base se encuentra información provista por UTE para la conversión y datos persistidos por el proceso de migración gráfica.
 - **name="ORACLEEMP1"**: Define la cadena de conexión a la base Oracle, donde persiste información SCADA X.
- Directorios necesarios:
 - **key="DIRLOG"**: Directorio donde se guardaran los logs de ejecución, con los posibles errores que puedan ocurrir.
 - **key="ArchivosIDDUG"**: Directorio donde se encuentra la estructura de carpetas y archivos con información de los mímicos del SCADA X.
 - **key="ArchivosMimx"**: Directorio donde el programa guarda los archivos .mimx, correspondientes a los mímicos generados.
 - **key="ArchivosObjetosMir"**: Directorio donde se encuentran los archivos correspondientes a los objetos de biblioteca Mirage. Estos archivos se pueden descargar directamente desde el Editor Mirage.
- Parámetros ajustables a datos de entrada:
 - **key="B1_IN"**: Instalación o instalaciones, separadas por punto y coma (;), para las guals se quiere generar su mímico. Tener en cuenta que debe corresponder al nombre de la instalación definido en SCADA X. Si este parámetro está vacío, toma todas las instalaciones pertenecientes al área definida en el parámetro **B1_AREA**.
 - **key="B1_OUT"**: Instalación o instalaciones, separadas por punto y coma (;), para las guals no se quiere generar su mímico
 - **key="B1_AREA"**: Área o áreas de SCADA X, separadas pro punto y coma (;), donde pertenecen las instalaciones a migrar. Este parámetro no puede estar vacío, y las áreas corresponden a 1,2 y 3, instalaciones en prueba, en explotación de Canelones y en explotación de Montevideo respectivamente.
 - **key="SEGMENTOS_A_EXCLUIR"**: Lista de ids de segmentos a excluir en la migración, separadas pro punto y coma (;). En ocasiones una misma instalación está dibujada en varios segmentos, esto provoca que se genere un mímico desproporcionado, por esta causa se excluye uno. Los ids de los segmentos se obtienen desde la consola SCADA X.
 - **key="B1_SEG_A_EXCLUIR"**: Lista de pares [INSTALACIONX, SEGMENTOX], separadas por punto y coma (;), a excluir en la migración. Existen casos donde un segmento es válido para la migración de una instalación, pero no lo es para otra, entonces se hace la exclusión específica para la instalación y el segmento.

- `key="B1_EXCEPCION_LLAVE"`: Lista de instalaciones del SCADA X, separadas por punto y coma (;), que pertenecen a lo que se llama un mímico de llave (más de un punto intermedio en el mímico), para las cuales se dibuja un objeto Mirage especial para sus elementos de corte, distinto al que llevaría si se representaran solos en un mímico y no en llave.
 - `key="PALABRAS_EXCLUSION"`: Listado de palabras o textos a excluir en la conversión, que no se insertan en el mímico Mirage. Las mismas deben ir separadas por punto y coma (;).
 - `key="ACT_PLANO_POTENCIA_COMPLETO"`: Si vale 1, migra todos los elementos del plano potencia de la instalación, si vale 0 solo migra las medidas que en él se encuentren. El plano potencia del SCADA X, auspicia de capa, donde se pueden mostrar u ocultar algunos elementos telecontrolados.
- Parámetros de mapeo:

El resto de las entradas del archivo de configuración corresponden al mapeo de colores y de objetos entre SCADA X y Mirage, dicho mapeo lo utiliza el software para dibujar de forma automática elementos en mímico, prescindiendo de la ayuda del usuario.

A modo de ejemplo, para las entradas:

- `<add key="SEC_15_251" value="D_015_SECBC_S_AP_UP*.obsx" />`

El sistema traduce directamente los seccionadores (SEC) en 15kV cuyo dibujo corresponde a un triángulo doble (id 251 en SCADA X), al objeto Mirage "D_015_SECBC_S_AP_UP", representado por el archivo D_015_SECBC_S_AP_UP.obsx. Se pone asterisco (*) para evitar versionado de archivos de los objetos de biblioteca.

- `<add key="COLOR_9" value="r='210' g='0' b='0'" />`

En este caso el sistema traduce el color de id 9 de SCADA X al sistema RGB (210, 0, 0), que es el usado por Mirage.

Ejecución

Una vez configurados los parámetros del programa, se ejecuta el archivo *SegmentToMimx.exe* para dar inicio al mismo.

Recomendación: Para realizar la migración de un unifilar de manera más cómoda, se recomienda la utilización de dos monitores. Esto facilita al usuario al momento de seleccionar los objetos Mirage que representan las celdas, en un monitor podemos tener el programa donde dibuja el objeto seleccionado y en otro monitor una conexión remota a la consola SCADA X. En la siguiente figura vemos un ejemplo, en el monitor izquierdo se está ejecutando el software de migración gráfica y en el derecho una conexión remota a una consola del SCADA X con la instalación que se está migrando:

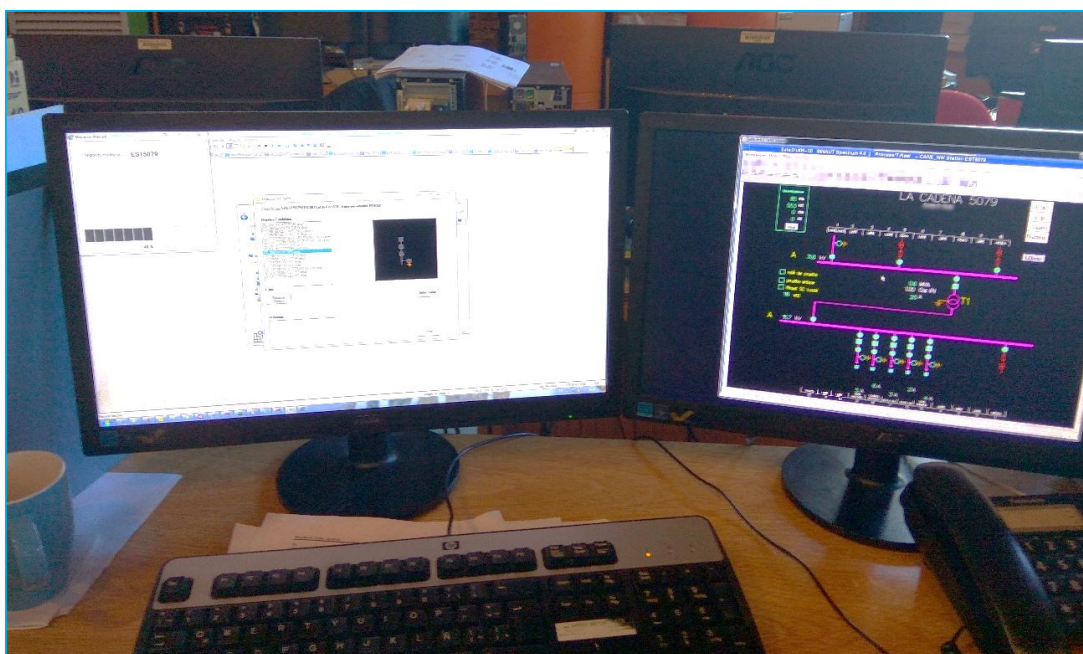


Figura 13.42: Ámbito de trabajo recomendado, dos monitores para el proceso de migración gráfica.

Una vez en ejecución, se podrá apreciar la ventana principal indicando la instalación actual que se está migrando y el porcentaje de avance de la migración.

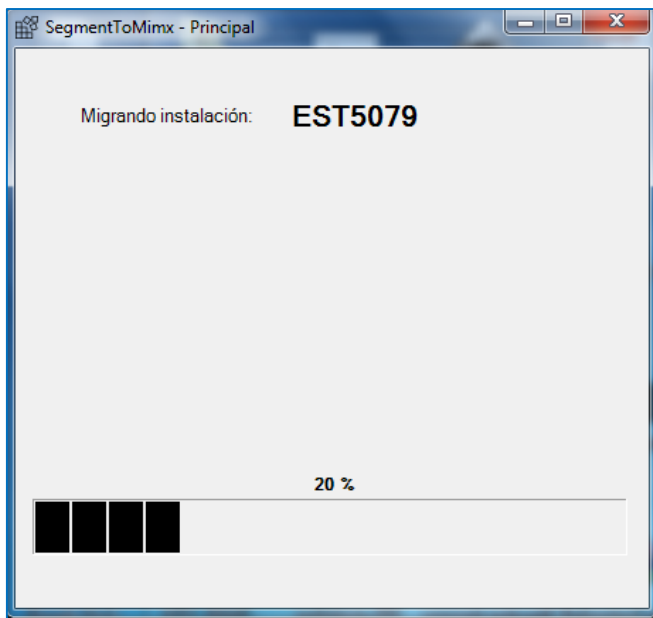


Figura 13.43: Ventana principal de SegmentToMimx, indicando progreso e instalación actual.

El programa convierte automáticamente todos los mímicos asociados a puntos intermedios, reconectores y seccionalizadores, claro está que posteriormente el usuario podrá realizar las modificaciones graficas en el editor Mirage.

Para el resto de las instalaciones, las que tiene celdas de corte, el programa solicitará al usuario que seleccione el objeto que más se ajuste a la configuración grafica de las celdas, a partir de un listado que el propio software se encargó de inferir. Dicha selección es persistida para próximas corridas, pero eventualmente el usuario podrá eliminar esa selección de objetos. Esta eliminación se permite hacer antes de la migración, mediante esta ventana:



Figura 13.44: Borrado de selección de objetos de celdas

Cuando el programa solicita seleccionar un objeto Mirage para representar la celda procesada, muestra la siguiente ventana:

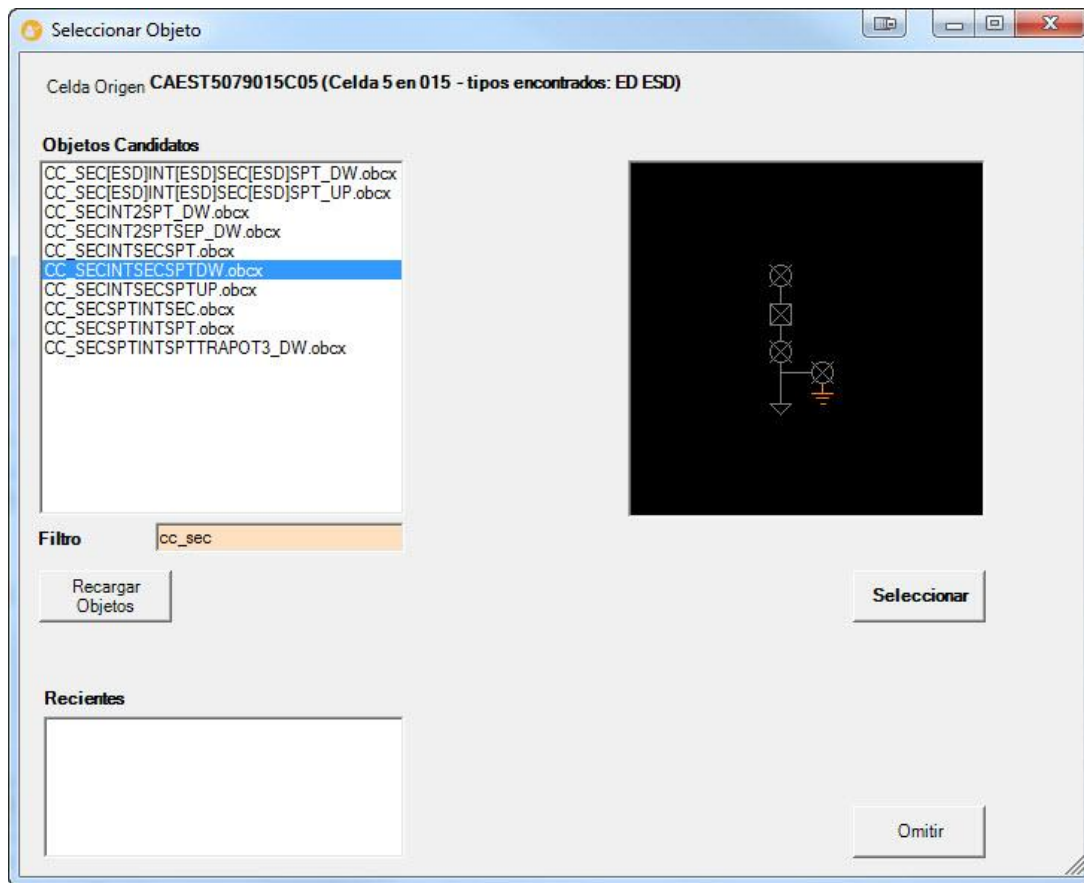


Figura 13.45: Ventana de selección de objetos Mirage.

La ventana de selección de objeto contiene los siguientes controles gráficos:

- **Celda Origen:** Muestra información al usuario sobre la celda a la cual se le debe asignar un objeto. En caso de que sea una celda con una única variable indica nombre y descripción de la variable.
- **Objetos Candidatos y Panel de vista previa:** El usuario puede seleccionar de los objetos candidatos el que crea más conveniente, y mientras selecciona puede ver la configuración grafica del objeto en el panel de vista previa.
- **Filtro:** Permite al usuario filtrar los objetos candidatos por nombre de objeto, en caso de saber de antemano cual necesita.
- **Recargar Objetos:** El botón de recargar objetos permite traer un listado con todos los objetos Mirage disponibles, en caso de que ninguno de los sugeridos por el software se ajusten a la configuración de la celda.

Nota: Si no existe objeto adecuado para la celda, se puede omitir la selección actual, luego crear el objeto necesario, se copia el archivo correspondiente al objeto en el directorio de objetos, y se vuelve a correr la migración para la misma instalación para tener disponible el nuevo objeto creado.

- **Recientes:** Listado de objetos usados recientemente en el proceso.
- **Omitir:** Se omite la selección para la celda actual, con lo cual dicha celda no será agregada en el mímico que se genere.

- **Seleccionar:** Confirma la selección del objeto para dicha celda. De esta manera el sistema dibujará la celda con el objeto seleccionado en el mímico que se genere, ya demás guarda la selección en la base de datos para posteriores ejecuciones.

Una vez finalizada la migración, el sistema propone un nombre para el mímico generado, tomando el nombre de la o las instalaciones contenidas en el mímico. En caso que dicho nombre no sea el que pretende utilizar el usuario, podrá modificarlo en la misma ventana. Este paso en el proceso no es menor, ya que el nombre que se defina para el mímico será utilizado para la integración con el sistema SIO. Es por esta razón que el sistema lanza una advertencia de reemplazo, preguntando al usuario si desea o no reemplazar el mímico anteriormente integrado.

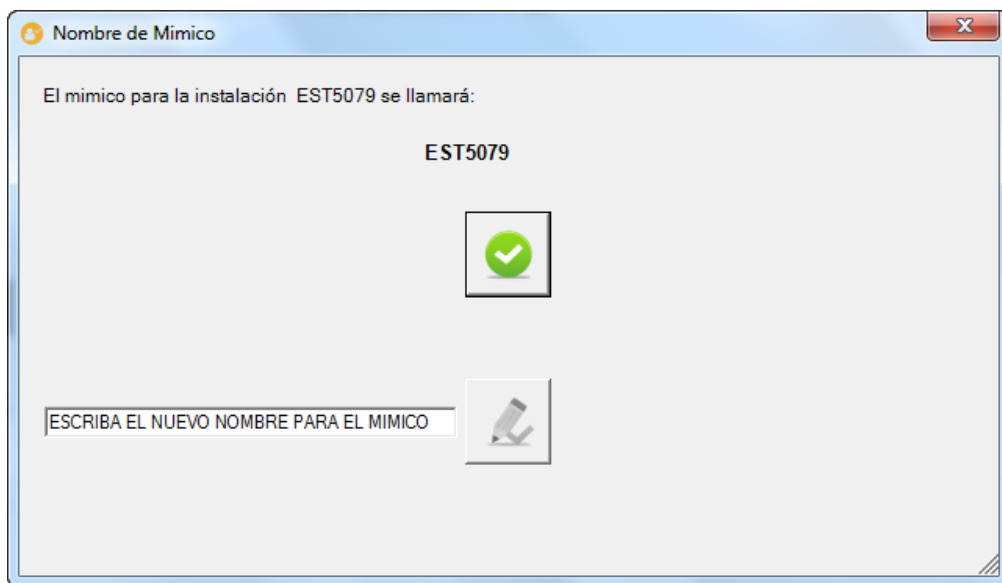


Figura 13.46: Ventana para nombrar Mímico

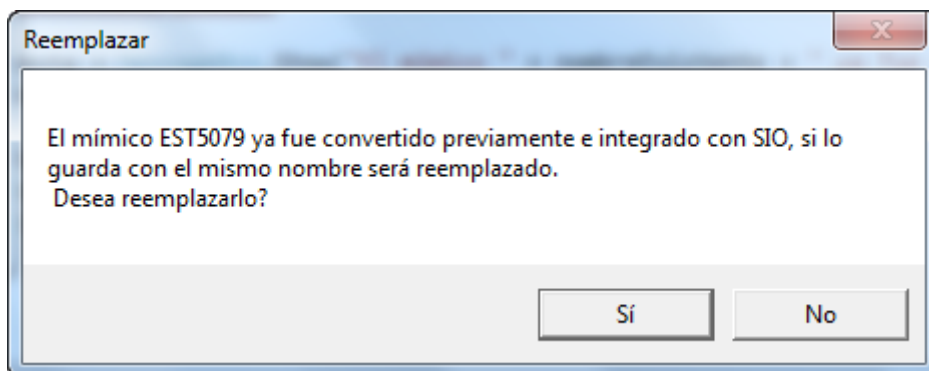


Figura 13.47: Ventana de remplazo de mímico.

Cuando se genera el archivo .mimx asociado al mímico, el usuario ya puede subirlo al sistema Mirage. Esto lo hace mediante el editor Mirage, y una vez en modo de edición se le podrán hacer las modificaciones estéticas que se crea conveniente.

En este punto tambien se recomienda usar dos monitores, uno para editar el nuevo mímico y otro para visualizar el mímico original en SCADA X. Además se confeccionaron

una serie de acciones a realizar en el editor para la primera vez que se abra el mímico que se detallan a continuación:

- Ctrl+A, para seleccionar todos los objetos y comprobar que todos están dentro del área del mímico.
- Actualizar objetos de biblioteca y verificar animaciones del mímico.
- Solucionar errores que muestra el editor.
- Insertar objeto trafo y/o conectarlo cuando sea necesario.
- Verificar que los objetos incrustados correspondan a la disposición origen.
- Verificar que todos los objetos estén conectados.
- Centrar los celdarios a las celdas.
- Revisar que existan inyecciones/entrada línea, cuando corresponda.
- Eliminar textos y figuras estáticas innecesarias.
- Hacer ajustes estéticos tales como tamaños de textos, mover y alinear botones, mover objetos para mantener ciertas proporciones, ajustar barras, etc.
- Chequear los bloqueos de RELE cuando corresponda, se agregan por scripts y no se dibuja en el unifilar.
- Chequear conectividad entre reenganches y barras.

Soporte Técnico

El soporte técnico corre por cuenta de la unidad de Software de Telecontrol-DIS, quién contará con el código fuente del software.

13.26 Anexo 26 – Manual *Testing* Grafico Mirage

***Testing* Gráfico Mirage**

Manual de usuario



Introducción

Para facilitar y automatizar las pruebas en el proceso de conversión se desarrolló el software *Testing Grafico Mirage*.

Lo que pretende la automatización de pruebas es validar que los mímicos generados mantienen la estructura de origen, identificar errores humanos y del proceso de migración automático, validar la configuración gráfica de las instalaciones eléctricas en SCADA y detectar errores presentes en SCADA X para eliminarlos en la migración.

Instalación

Requisitos:

- Windows 7 o superior.
- *Framework .Net* 4.0 o superior.
- Cliente Oracle 8 o superior.

Alcanza con ubicar en un mismo directorio los archivos *Testing Grafico Mirage.exe*, *Testing Grafico Mirage.exe.config* y *QUERY.Acceso.dll*, como lo muestra la siguiente imagen:

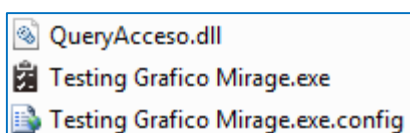


Figura 13.48: Archivos a ser utilizados en el *testing*

- **QUERY.Acceso.dll**: Librería provista por UTE para facilitar el acceso a las base de datos SQL Server y Oracle.
- **Testing Grafico Mirage.exe**: Ejecutable del software de pruebas automáticas.
- **Testing Grafico Mirage.exe.config**: Archivo de configuración del software.

Configuración

A continuación se describe el archivo *Testing Grafico Mirage.exe.config* para su correcta configuración:

- Cadenas de conexiones a bases de datos:
 - **name="PIWEB1"**: Define la cadena de conexión a la base SQL Server, dicha base se encuentra información provista por UTE para la conversión, datos persistidos por el proceso de migración gráfica y excepciones de usuario del software de pruebas.
 - **name="ORACLEEMP1"**: Define la cadena de conexión a la base Oracle, donde persiste información SCADA X.
- Directorios y archivos necesarios:
 - **key="DIRLOG"**: Directorio donde se guardaran los logs de ejecución, con los posibles errores que puedan ocurrir.
 - **key="DIR_MÍMICOS"**: Directorio donde se encuentran los archivos .mimx correspondientes a los mímicos Mirage.
 - **key="SISTEMAS_PARCIALES"**: Archivo de configuración del protocolo 101 de Mirage. Este archivo es utilizado para identificar qué sistema parcial 101 utiliza cada instalación con enlace 101.
- Tipo de variables:
 - **key="TIPOS_VARIABLES_MIRAGE"**: Listado de tipos de variable Mirage separados por punto y coma (;). Estos tipos de variables son los que se leen desde los mímicos en las pruebas automáticas. Los tipos a tener en cuenta son: **ENT_ANA;MEM_ANA;ENT_DIG;ENS_DIG,MEM_DIG**.

Ejecución

Una vez configurados los parámetros del software de pruebas, se ejecuta el archivo *Testing Gráfico Mirage.exe* para dar inicio al mismo.

Al iniciar la aplicación, veremos la ventana principal donde se puede ver los tipos de chequeos que se realizan, su progreso y un botón para visualizar las excepciones añadidas por el usuario.

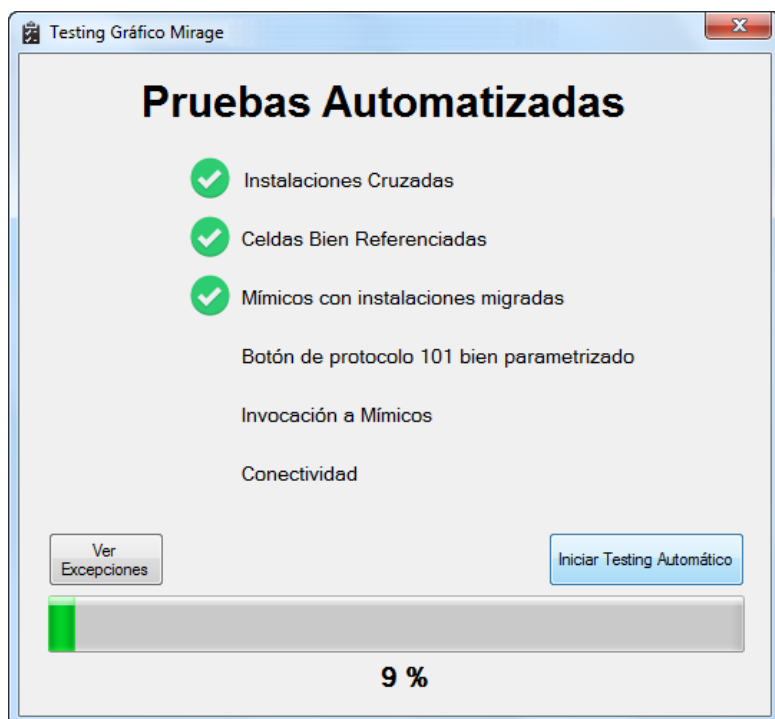


Figura 13.49: Ventana principal de *Testing Gráfico Mirage*.

Tipos de Pruebas:

- **Instalaciones cruzadas:** Verifica que no hayan elementos de una instalación en mímicos de otra instalación. Previene una maniobra errónea en una instalación.
- **Celdas bien referenciadas:** Verifica que las celdas estén dibujadas en la ubicación correcta en el celdario y además que no hayan elementos de una celda en otra celda de la misma instalación. Previene una maniobra errónea en una celda.
- **Mímicos con instalaciones migradas:** Verifica que todos los mímicos contengan instalaciones migradas con el software de migración gráfica. Garantiza integridad en el proceso de migración y asegura la integración con el sistema SIO.
- **Botón de protocolo 101 bien parametrizado:** Verifica que las instalaciones con enlace 101 tengan bien parametrizado el botón correspondiente a su sistema parcial en su mímico. Corrige modificaciones a nivel del protocolo 101 Mirage y los cambios constantes de sistemas parciales.
- **Invocación a mímicos:** Verifica que todos los mímicos del CMD, los que se encuentran en el directorio definido, sean invocados desde algún otro mímico.

Esto evita que el operador no pueda acceder a un mímico que contenga una alarma encendida.

- **Conectividad:** Verifica que la conexión de los elementos de una instalación se mantengan igual a como están en SCADA X. Previene un coloreo erróneo en la conectividad del circuito de la instalación.

Una vez culminado el proceso de *testing* automático se despliega un listado con los errores encontrados. Muchos de esos errores son falsos positivos, con lo cual el usuario los puede agregar como excepción para que no sean tomados en cuenta en futuras pruebas.

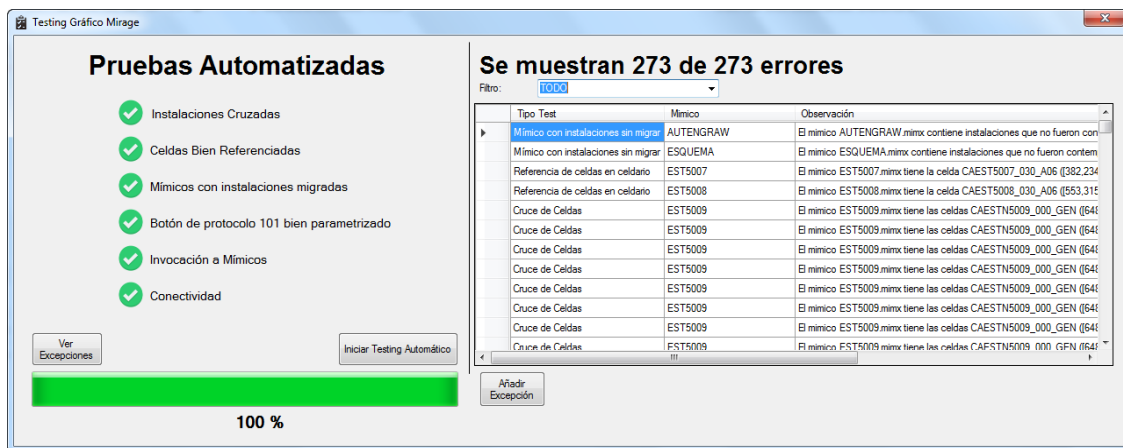


Figura 13.50: Ventana de pruebas con los errores encontrados.

Si se selecciona un registro, y se da clic en el botón “Añadir Excepción”, se muestra un formulario para agregar excepción, dónde se define usuario que agrega la excepción del error y una descripción de la misma.

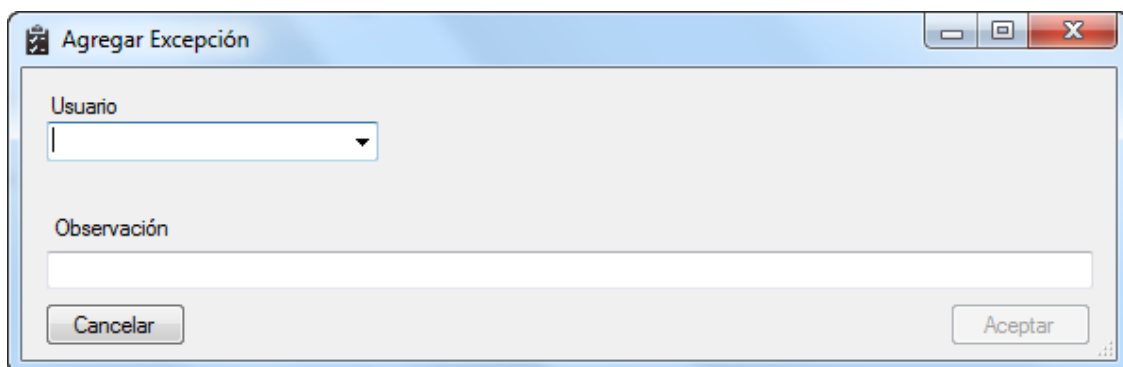


Figura 13.51: Formulario para agregar excepción.

En la ventana principal, desde el botón “Ver excepción”, se podrán visualizar las excepciones añadidas por los usuarios.

Por ejemplo, se puede dar que se haya creado un mímico manualmente, el sistema lo identifica como que no se tuvo en cuenta para la migración automática, pero el usuario es consiente que no se trata de un error y lo marca como excepción.

13.27 Anexo 27 – Editor Gráfico Mirage

A continuación describiremos los aspectos destacables del editor gráfico de mímicos, software que permite editar mímicos del SCADA Mirage. La idea no es entrar en profundidad en el software sino más que nada mostrar la perspectiva que tiene el usuario al momento de agregar y editar un mímico Mirage.

Todos los mímicos obtenidos a partir de la conversión fueron procesados posteriormente con el editor Mirage en mayor o menor medida.

El sistema Mirage unifica los mímicos correspondientes a un mismo CMD en una entidad lógica llamada vista. Esta vista se materializa físicamente en un archivo .visx, este archivo contiene internamente las referencias a los mímicos que engloba.

Para el caso del CMD Centro, se creó la vista “CANFLO.visx”, que contiene todos los mímicos pertenecientes a las instalaciones de Canelones y Florida.

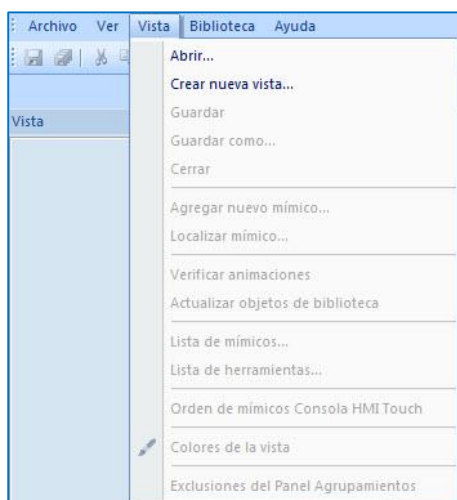


Figura 13.52: Menú Vista en el Editor Mirage.

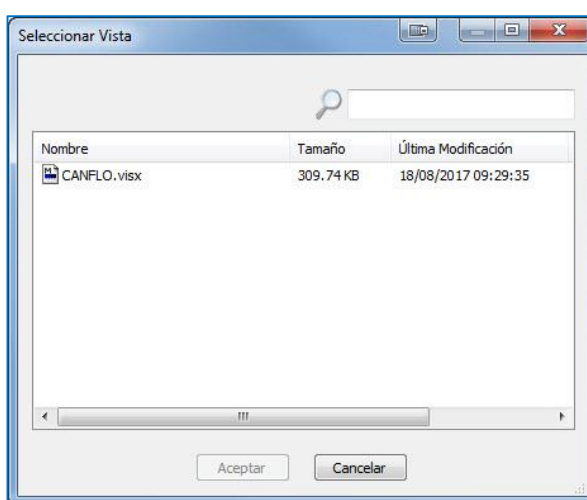


Figura 13.53: Selección de la vista CANFLO.visx

Cuando se abre una vista, podemos visualizar el listado de mímicos que contiene, en el panel de Vista, como se observa en la siguiente imagen:

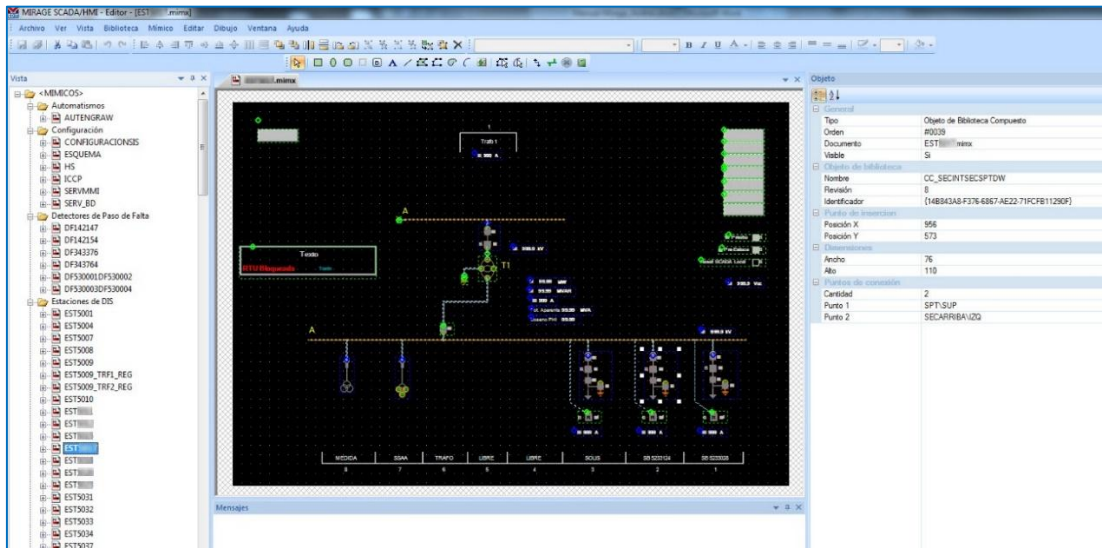


Figura 13.54: Ventana Principal del Editor

Mirage cuenta con un repositorio de mímicos para trabajar en el ámbito de producción, al mismo se puede acceder desde el menú Archivo. Es necesario agregar a ese repositorio el archivo .mimx correspondiente al mímico, previo a agregarlo a la vista.

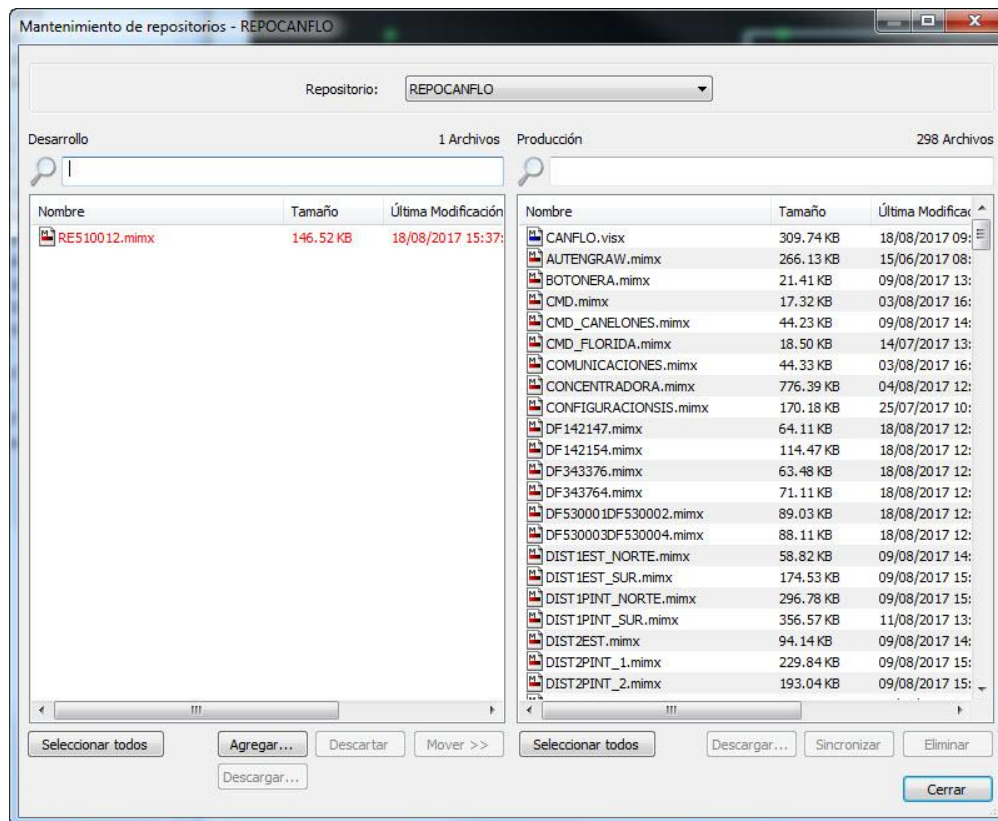


Figura 13.55: Ventana de gestión de archivos en el repositorio Mirage.

Cuando el archivo del mímico ya está en el repositorio, se puede agregar a la vista, mediante la opción Lista de mímicos del menú Vista.

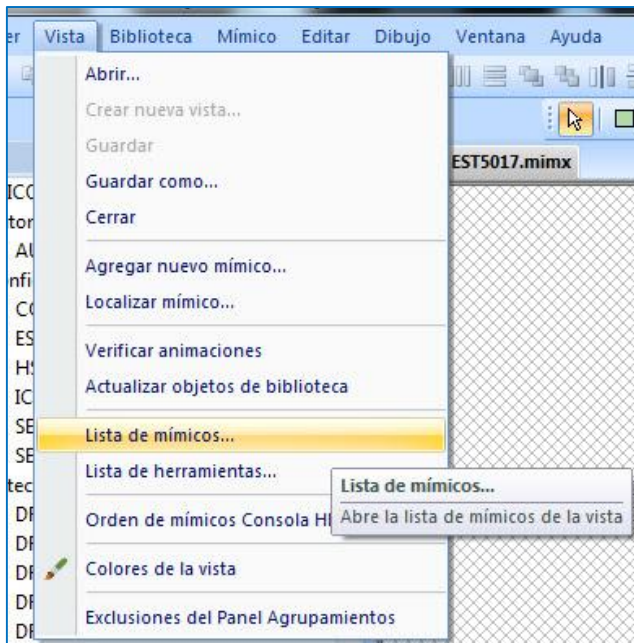


Figura 13.56: Menú Vista en el contexto de la vista CANFLO

Luego se despliega una ventana que muestra un el listado de mímicos pertenecientes a la vista, para agregar el mímico que ya se encuentra en el repositorio hay que presionar el botón “Agregar nuevo”.

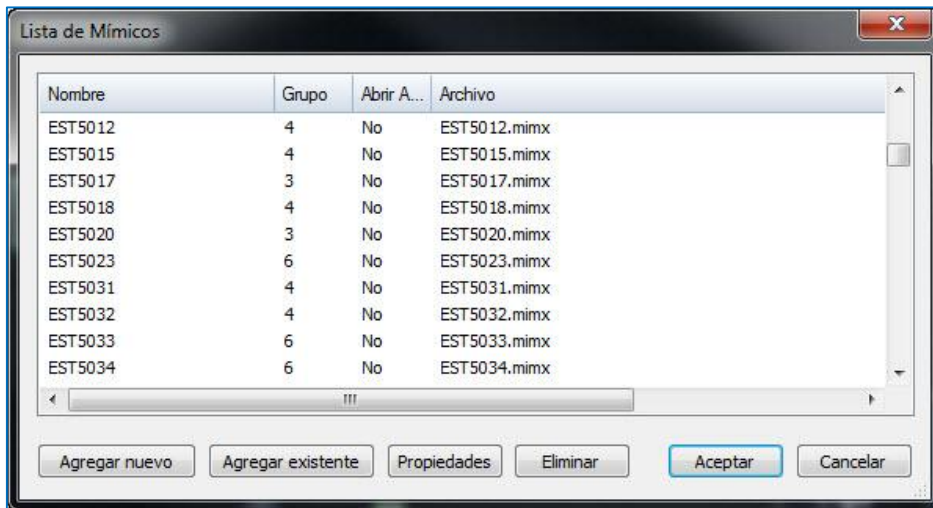


Figura 13.57: Lista de Mímicos con opción de agregar y eliminar.

Luego se despliega un formulario, donde, entre otros parámetros, podemos seleccionar el archivo correspondiente al mímico a agregar.

Figura 13.58 Formulario de agregado de Mímico.

Cuando el mímico queda agregado a la vista, ya queda disponible para su edición.

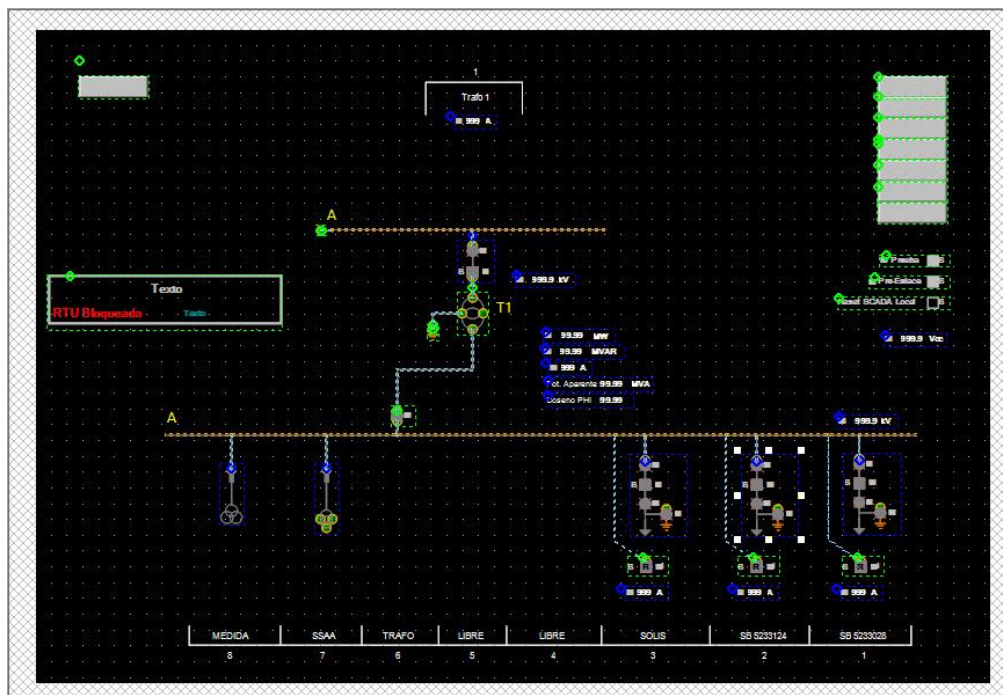


Figura 13.59: Mímico en modo edición.

Las siguientes son barras de herramientas del Editor:



Figura 13.60: Barra de herramientas para inserción de figuras, textos, conectores y objetos de biblioteca.



Figura 13.61: Barra edición de textos.



Figura 13.62: Barra de agrupamiento de elementos gráficos.

En la figura 13.60, se hace mención a los objetos Mirage a los que tanto se hace referencia en la documentación, los mismos se pueden insertar mediante el icono , o mediante el menú el Biblioteca.

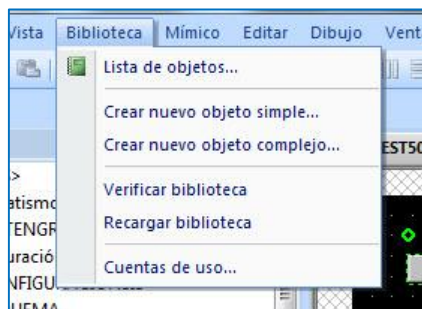


Figura 13.63: Menú Biblioteca en el editor.

Dentro de la biblioteca Mirage, no solo se puede ver un listado de objetos, sino además al seleccionar se muestra una vista previa del mismo.

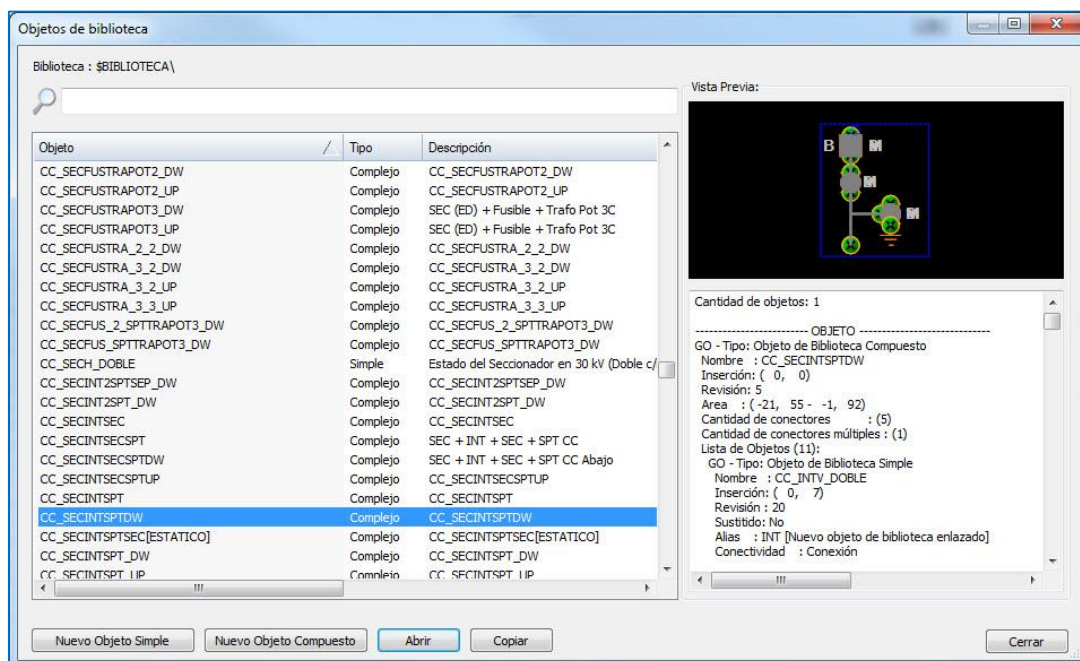


Figura 13.64: Biblioteca de objetos Mirage.

Los objetos Mirage, permiten estandarizar los dibujos y facilitar la edición y mantenimiento de los mímicos.

Estos objetos son mapeados a variables SCADA para que, en función de los estados y valores de esas variables, se animen dinámicamente. Por ejemplo, una variable SCADA que representa un interruptor en campo que puede eventualmente abrir o cerrar, se debe representar esa apertura o cierre en el mímico, y ese dinamismo viene dado por el objeto que representa a la variable.

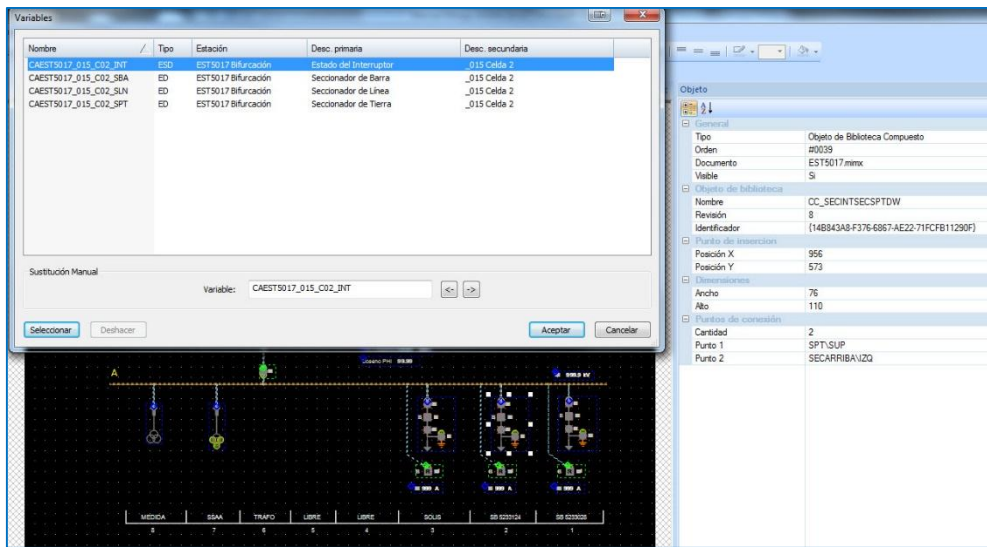


Figura 13.65: Mapeo de variables Mirage a objetos.

Para finalizar, mostramos como se visualiza en la consola Mirage el mímico de la figura 13.59 que estaba en modo edición. Notar que en la figura 13.66 se ven los elementos de los circuitos pintados por nivel de tensión según estándar de UTE-Distribución⁴.

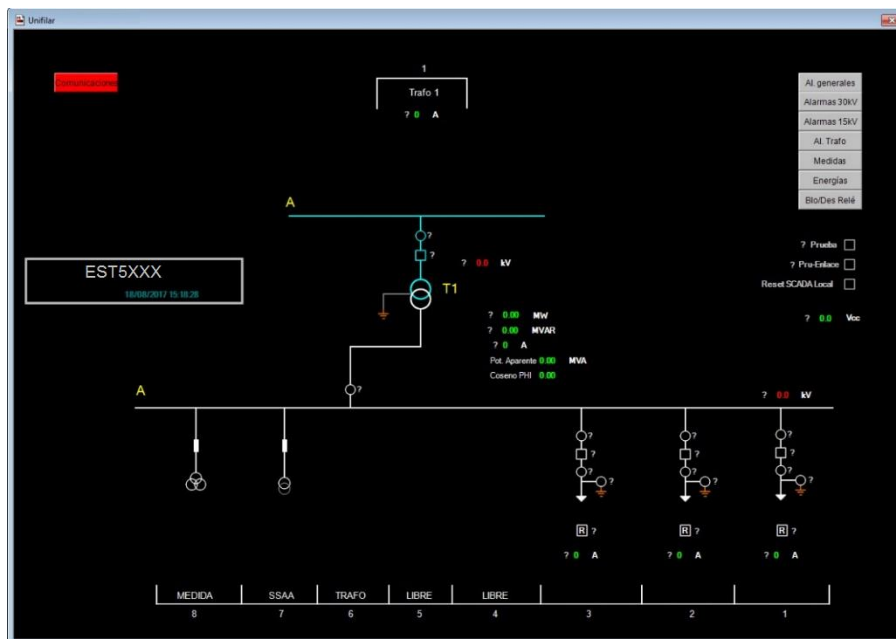


Figura 13.66: Mímico de la estación EST5XXX en la consola pintado por nivel de tensión.

⁴ Estándar de colores en tensiones para UTE-Distribución: Blanco corresponde a 15kV y celeste a 30kV.

Mientras en la segunda muestra el pintado por la opción de conectividad, donde se aprecia elementos con tensión en rojo, solo la barra en alta, y en gris la parte del circuito que no tiene tensión, dado que están abiertos los elementos del mismo.

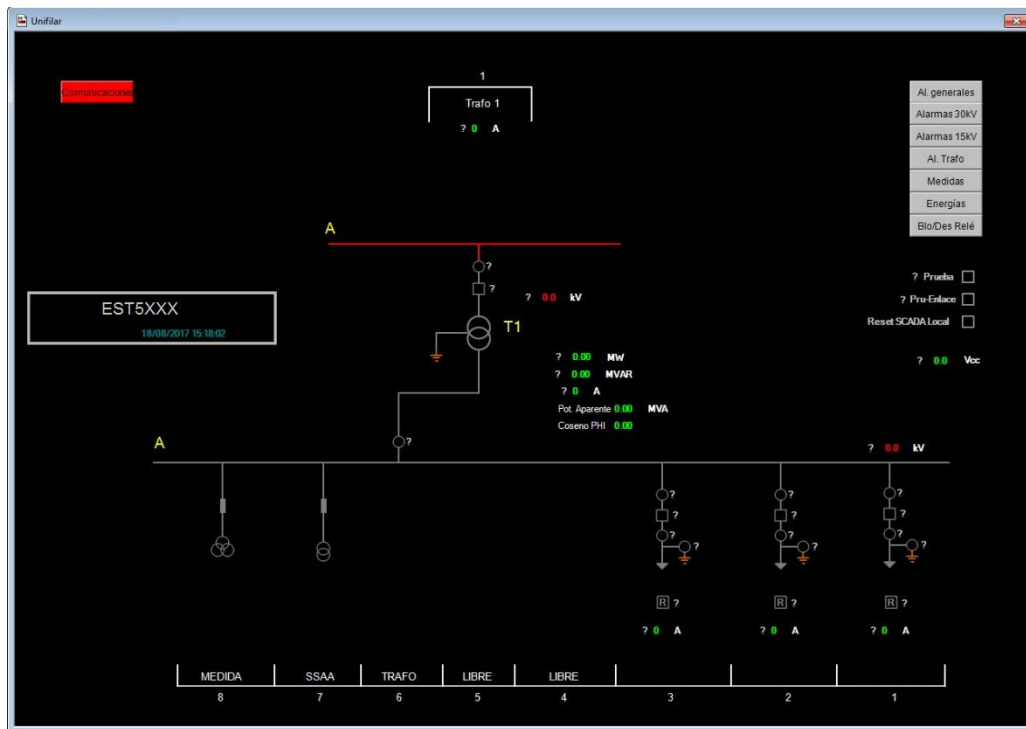


Figura 13.67: Mímico de la estación EST5XXX en la consola pintado por conectividad.

13.28 Anexo 28 – Pruebas de Integración de los tres sistemas.

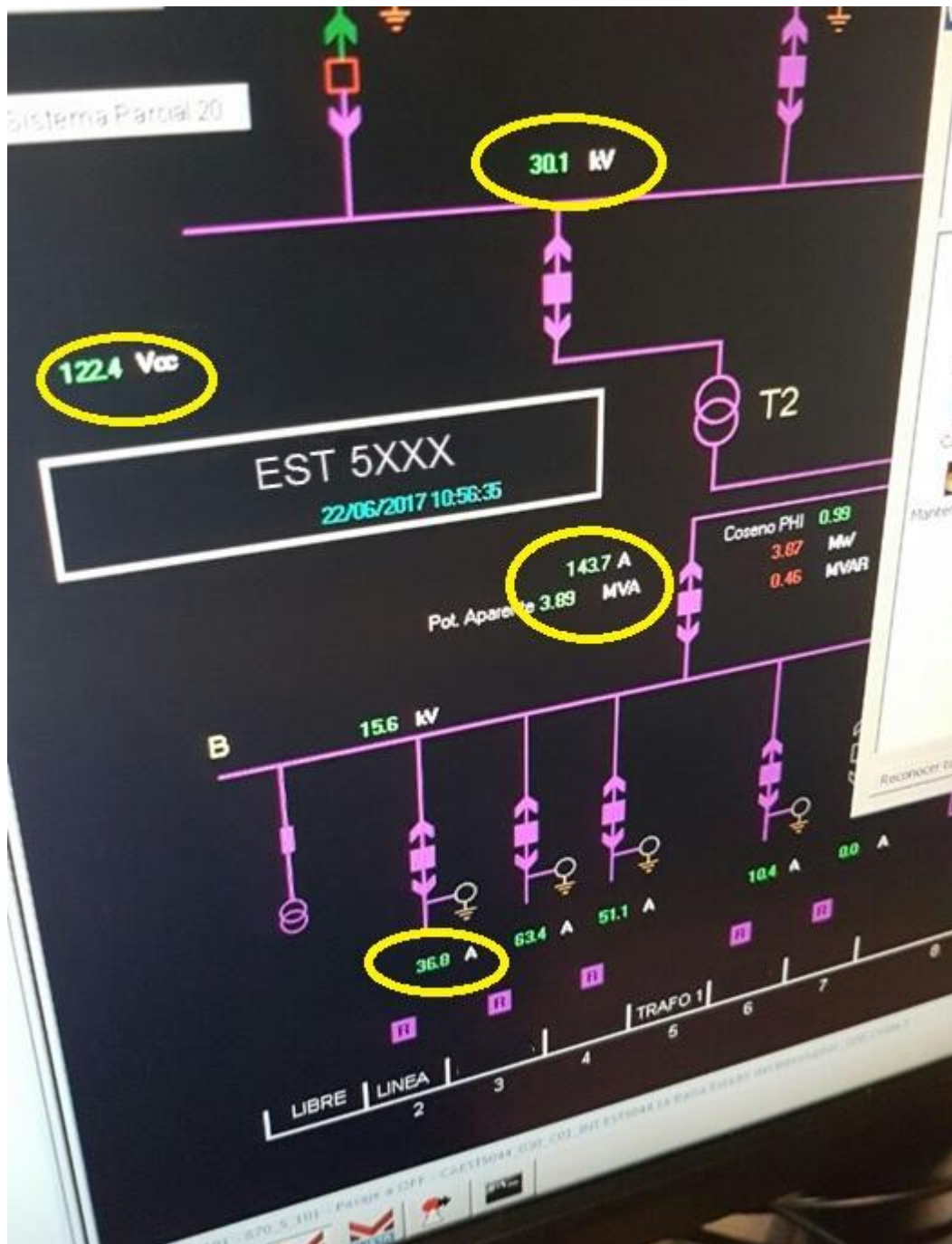


Figura 13.68: Unifilar Mirage conectado EST 5XXX.

En las elipses se pueden ver diferentes medidas y estados, que llegan en tiempo real, para que estas medidas se reporten, los tres componentes del SCADA deben funcionar correctamente.