Universidad ORT Uruguay

Facultad de Ingeniería

Shake Tool

Sistema de detección de phishing en correos electrónicos

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

Pablo Baccarezza - 120981

Alejandro Ramírez - 120032

Tutora: Mariana Lasarte

2023

Declaración de autoría

Nosotros, Pablo Baccarezza y Alejandro Ramírez, declaramos que el trabajo que se presenta en esta obra es de nuestra propia mano. Podemos asegurar que:

• La obra fue producida en su totalidad mientras realizamos el proyecto de grado de la carrera Licenciatura 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 aportado por otros, y qué fue aportado por nosotros;

• Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes

_____ _____

Pablo Baccarezza **Alejandro Ramírez**

7/4/2023 **7/4/2023**

Agradecimientos

Queremos agradecer a nuestra tutora, Ing. Mariana Lasarte, por el apoyo y guía durante esta enriquecedora etapa.

También a aquellos docentes y compañeros que dedicaron su tiempo a darnos sus consejos y sus experiencias cuando fue necesario.

Agradecer al equipo de Seguridad de la Información del BROU, con las figuras de su gerente Ing. Marcelo Varaldi, y los Analistas especialistas Ing. Alejandro Cao, Ing. Rodrigo Mateos, Ing. Federico Zubbiri. Además, a Ing. Álvaro Lousteau de Arquitectura y Especialista de SI Laura Fontana.

Quiero agradecer a mi esposa y a mis hijas que sin su apoyo no hubiera sido posible el desarrollo de este trabajo.

Pablo Baccarezza

Deseo agradecer a aquellos amigos y familiares que han estado pendiente, mandando su energía positiva, pero especialmente a mi esposa e hijos, que han sido el sostén para poder dedicar el tiempo y esfuerzo que esta tarea amerita, brindando su incondicional apoyo sobre todo en los momentos difíciles.

Alejandro Ramírez

Abstract

El presente trabajo corresponde al Proyecto de Grado de la carrera de Licenciatura en Sistemas de la Universidad ORT Uruguay y fue posible gracias a la colaboración de los diferentes equipos de trabajo del Banco de la República Oriental de Uruguay que participaron del mismo.

Nuestro proyecto involucra una problemática que ha sido y es, transversal a la institución, y que tiene que ver con la recepción de mails con contenido fraudulento, incluso teniendo que dar respuestas a la opinión pública por hechos que la involucran. Esto no solo afecta en términos de reputación, sino que pone en evidencia la dificultad de tratar el tema, de procesar estos casos y de centralizar información para una mejor toma de decisiones de alta gerencia.

Este equipo de proyecto se planteó desarrollar una prueba de concepto con la finalidad de automatizar el proceso de análisis de correos electrónicos, clasificándolos de forma automática, con la intención de facilitar la gestión a los usuarios finales y analistas de seguridad de la información y a su vez proveer de datos que permitan tomar acciones tempranas y eficientes.

Para ello se apoya en Machine Learning, y la capacidad de utilizar la información y conocimiento ya adquirido para entrenar los algoritmos y contar con modelos ajustados a la casuística del banco. También se deja abierta la posibilidad de incorporar varias estrategias a futuro, como la incorporación de nuevos modelos, procesamiento por una combinación de los modelos más eficientes e incluso el procesamiento mediante otras técnicas como el análisis de la Meta Data, que es un gran campo de desarrollo complementario a lo que ya fue implementado.

El resultado ha sido la creación de una prueba de concepto que provee la capacidad de analizar correos electrónicos, brindando al equipo de soporte información de los casos detectados y de aquellos casos que requieran la participación de analistas de seguridad de la información para emitir una segunda opinión.

Palabras clave

Machine Learning, Phishing, Ciclo de vida, aprendizaje automático, Seguridad de la Información.

GLOSARIO

Active Directory: Un directorio es una estructura jerárquica que almacena información sobre objetos en la red. Un servicio de directorio, como los Servicios de dominio de Active Directory (AD DS), proporciona los métodos para almacenar datos de directorio y ponerlos a disposición de los usuarios y administradores de la red. Por ejemplo, AD DS almacena información sobre cuentas de usuario, como nombres, contraseñas, números de teléfono, etc., y permite que otros usuarios autorizados en la misma red accedan a esta información.[1]

A.I: La inteligencia artificial es un campo de la ciencia relacionado con la creación de computadoras y máquinas que pueden razonar, aprender y actuar de una manera que normalmente requeriría inteligencia humana o que involucre datos cuya escala exceda lo que los humanos pueden analizar.[2]

Analista de Seguridad de la Información: Tiene como responsabilidad realizar análisis de riesgos de los activos de información, proponer controles. Realizar el monitoreo de los sistemas críticos.

BaggingClassifier: Es un meta estimador de conjunto que ajusta los clasificadores base cada uno en subconjuntos aleatorios del conjunto de datos original y luego agrega sus predicciones individuales (ya sea por votación o por promedio) para formar una predicción final. [3]

Correo electrónico: Un mensaje es una serie de caracteres que consta de campos de encabezado (denominados colectivamente "el encabezado del mensaje") seguido, opcionalmente, por un cuerpo. El encabezado es una secuencia de líneas de caracteres. El cuerpo es simplemente una secuencia de caracteres que sigue al encabezado y está separado del encabezado por una línea vacía.[4]

Ciber Delincuente: El ciberdelincuente es la persona que buscará sacar beneficio de estos problemas o fallos de seguridad utilizando para ello distintas técnicas como es la ingeniería social o el malware.[5]

Django: Django es un marco web Python de alto nivel que fomenta un desarrollo rápido y un diseño limpio y pragmático.[6]

Dump: El método dump() se utiliza cuando los objetos de Python deben almacenarse en un archivo.

Framework: Un framework es un entorno o marco de trabajo, un conjunto de prácticas, conceptos y criterios a seguir estandarizados. Siguiendo unas reglas, el framework nos obliga a utilizar buenas prácticas para nuestro código. Por otro lado, los frameworks también nos proporcionan una serie de herramientas ya desarrolladas.

GradientBoostingClassifier: Este algoritmo construye un modelo aditivo en una forma avanzada por etapas; permite la optimización de funciones de pérdida diferenciables arbitrarias. En cada etapa, n_classes_los árboles de regresión se ajustan al gradiente negativo de la función de pérdida, por ejemplo, pérdida logarítmica binaria o multiclase. La clasificación binaria es un caso especial en el que solo se induce un único árbol de regresión.[7]

Hiper Parámetros: Son los valores de las configuraciones de los algoritmos de machine learning utilizadas durante el proceso de entrenamiento.

Linux: Es un sistema operativo (SO) open source. El sistema operativo es el software que gestiona directamente el hardware de un sistema y sus recursos, como la CPU, la memoria y el

almacenamiento. Se encuentra entre las aplicaciones y el hardware, y establece las conexiones entre todos los sistemas de software y los recursos físicos que ejecutan las tareas.

Look and feel: Es el conjunto de propiedades y características que le dan una identidad visual única.

Machine Learning: El aprendizaje automático o aprendizaje automatizado o aprendizaje de máquinas (del inglés, machine learning) es el subcampo de las ciencias de la computación y una rama de la inteligencia artificial, cuyo objetivo es desarrollar técnicas que permitan que las computadoras aprendan.

Malware: El malware es un programa informático (software, en inglés) cuya principal característica es que se ejecuta sin el conocimiento ni autorización del propietario o usuario del equipo infectado y realiza funciones en el sistema que son perjudiciales para el usuario y/o para el sistema.

MultinomialNB: Es un clasificador multinomial, adecuado para la clasificación con características discretas.

Naive Bayes: En un sentido amplio, los modelos de Naive Bayes son una clase especial de algoritmos de clasificación de Aprendizaje Automático, o Machine Learning. Se basan en una técnica de clasificación estadística llamada “teorema de Bayes”.

NLP: Es un campo de las ciencias de la computación, de la inteligencia artificial y de la lingüística que estudia las interacciones entre las computadoras y el lenguaje humano.

O.C.R: El reconocimiento óptico de caracteres (ROC), generalmente conocido como reconocimiento de caracteres y expresado con frecuencia con la sigla OCR (del inglés Optical Character Recognition), es un proceso dirigido a la digitalización de textos, los cuales identifican automáticamente a partir de una imagen símbolos o caracteres que pertenecen a un determinado alfabeto, para luego almacenarlos en forma de datos.

ORM: Una de las características más poderosas de Django, Mapeador Relacional de Objetos (ORM), que le permite interactuar con su base de datos, como lo haría con instrucciones SQL.

Pickle: es el proceso mediante el cual una jerarquía de objetos de Python se convierte en una secuencia de bytes

Pipeline: Un pipeline de datos es una construcción lógica que representa un proceso dividido en fases. Los pipelines de datos se caracterizan por definir el conjunto de pasos o fases y las tecnologías involucradas en un proceso de movimiento o procesamiento de datos.

Phishing: El phishing es un tipo de ataque informático. Consiste en un conjunto de técnicas que persiguen el engaño de una persona, imitando la identidad de un tercero de confianza, como podría ser un banco, una institución pública, empresa o red social, con el fin de manipularla y lograr que brinde información sensible (por ejemplo, usuarios y contraseñas, datos personales, entre otros).

Planning Poker: Planning Poker es una forma potente y divertida de mejorar las ceremonias de planificación y estimación para equipos remotos y presenciales. Se discute el alcance y el esfuerzo de cada historia en equipo y, a continuación, compara la estimación anónima de cada

uno. Si el equipo no llega a un consenso se abre un debate para comprender mejor el trabajo y se vuelve a estimar hasta que el equipo llegue a un acuerdo.

POC: Una prueba de concepto o PoC (del inglés proof of concept) es una implementación, a menudo resumida o incompleta, de un método o de una idea, realizada con el propósito de verificar que el concepto o teoría en cuestión es susceptible de ser explotada de una manera útil.

RandomForest Classifier: Es un meta estimador que ajusta varios clasificadores de árboles de decisión en varias submuestras del conjunto de datos y utiliza el promedio para mejorar la precisión predictiva y controlar el sobreajuste.

Reduce to a root: Modificación de una palabra para expresar diferentes categorías gramaticales como tiempo, caso, voz, aspecto, persona, número, género y modo. Una flexión

expresa una o varias categorías gramaticales con un prefijo, un sufijo o un infijo, u otra modificación interna, como un cambio vocálico.

SI: Seguridad de la Información

Sprint: Un sprint es un período breve de tiempo en el que un equipo de scrum trabaja para completar una cantidad de trabajo establecida.

Sprint Planning: es una reunión que se realiza al comienzo de cada Sprint donde participa el equipo Scrum al completo

Sprint Review: El sprint review es el nombre que recibe la reunión que se celebra con el propósito de evaluar los resultados que obtuvo el equipo Scrum luego de un sprint

Story Point: Los puntos de historia son unidades de medida para expresar una estimación del esfuerzo global necesario para implementar por completo un elemento del backlog del producto o cualquier otra pieza de trabajo. Los equipos asignan puntos de historia en función de la complejidad del trabajo, la cantidad de trabajo y el riesgo o la incertidumbre.

Stop words: Se refiere a las palabras más comunes en un idioma. No existe una lista universal de estas palabras.

TDD: Es una práctica de programación que consiste en escribir primero las pruebas (generalmente unitarias), después escribir el código fuente que pase la prueba satisfactoriamente y, por último, refactorizar el código escrito.

Tokenization: Divide los datos no estructurados y el texto en lenguaje natural en fragmentos de información que pueden considerarse elementos discretos.

UNIX: Es un sistema operativo que nace a principios de los años 70, creado principalmente por Dennis Ritchie y Ken Thompson. Sus características técnicas principales son: su portabilidad, su capacidad multiusuario y multitarea, su eficiencia; su alta seguridad y su buen desempeño en tareas de red.

1. INTRODUCCIÓN 17

1.1. OBJETIVOS DEL PROYECTO 17

1.2. DESCRIPCIÓN DEL CLIENTE 18

1.3. DESCRIPCIÓN DE LOS INTEGRANTES DEL PROYECTO 20

1.3.1 Roles del proyecto 21

1.4 MOTIVACIÓN DEL PROYECTO 22

1.5 DESCRIPCIÓN DEL PROBLEMA 22

1.5.1 Problema a nivel global 22

1.5.2 Problema a nivel institucional 24

1.6 CÓMO TRABAJAN HOY LOS ANALISTAS DE SEGURIDAD DE LA INFORMACIÓN CON RESPECTO A ESTE TEMA? 26

1.7 DESCRIPCIÓN DE LA SOLUCIÓN 27

1.8 VISIÓN DEL PRODUCTO 29

1.9 ANÁLISIS DE PRODUCTOS DEL MERCADO 30

2. ANÁLISIS ESTRATÉGICO 33

2.1 ANÁLISIS PESTEL 33

2.2 SITUACIÓN DEL PHISHING EN URUGUAY 35

2.3 ANÁLISIS DE LAS 5 FUERZAS DE PORTER 36

2.4 ANÁLISIS FODA DEL BROU EN RELACIÓN CON EL PHISHING: 38

3. INGENIERÍA DE REQUERIMIENTOS 39

3.1 PLANIFICACIÓN DE LA ETAPA 39

3.2 RELEVAMIENTO DE REQUERIMIENTOS 42

3.3 INTERESADOS 46

3.4 Especificación de requerimientos 48

3.5 Priorización de requerimientos 55

3.5.1 Alcance inicial 55

3.6 Validación de requerimientos 57

3.7 Conclusiones 60

4. ARQUITECTURA 61

4.1 Análisis de marcos para diseño arquitectónico 66

4.2 Descripción de la Arquitectura 68

4.2.1 Validación 74

4.3 Conclusiones 75

5. TECNOLOGÍA 75

6. CONSTRUCCIÓN 81

6.1 Ciclo de vida 82

6.1.1 Elementos en común y diferencias entre ambos ciclos de vida 85

6.2 Modelo de Machine Learning 87

6.2.1 Comprendiendo el problema 88

6.2.2 Comprensión y preparación de datos 90

6.2.2.1 Pre - Procesamiento de los correos 91

6.2.3 Model training and evaluation 95

6.2.4 Deploy del modelo 105

6.3 Metodología de trabajo 108

6.4 Plan de releases 108

6.4.1 Sprints 109

6.5 Técnicas, ceremonias y métricas 112

6.5.1 Técnicas 112

6.5.2 Ceremonias 113

6.5.3 Métricas de los sprints 114

6.6 Conclusiones 115

7. TESTING 116

7.1 Plan de pruebas 116

7.2 Pruebas unitarias 120

7.3 Pruebas funcionales 122

7.4 Pruebas de performance 136

7.5 Pruebas de compatibilidad 138

7.6 Pruebas de los modelos de Machine Learning 140

7.7 Conclusiones 141

8. GESTIÓN DEL PROYECTO 142

8.1 Metodología del proceso de construcción 143

8.2 Seguimiento del plan 145

8.3 Gestión del tiempo 146

8.3.1 Capacidad del equipo 150

8.4 Gestión de la comunicación 150

8.4.1 Comunicación interna del equipo 152

8.5 Análisis y gestión de riesgos 152

8.5.1 Metodología de riesgos 153

8.6 Gestión del Alcance 159

8.7 Evaluación de los Sprint 159

8.8 Conclusiones 164

9. ASEGURAMIENTO DE LA CALIDAD 166

9.1 Objetivos de Calidad 166

9.2 Actividades de Calidad 168

9.2.1 Ingeniería de requerimientos..... 168

9.2.2 Diseño Arquitectónico..... 169

9.2.3 Construcción..... 170

9.2.4 Gestión de Riesgos..... 173

9.2.5 Gestión del proyecto..... 173

9.2.6 Apoyo a las actividades..... 173

9.3 Métricas..... 175

9.3.1 Usabilidad..... 176

9.3.2 Gestión de Incidencias y Bugs..... 180

Conclusiones..... 183

11. CONCLUSIONES Y LECCIONES APRENDIDAS 189

11.1 Lecciones aprendidas..... 189

11.2 Conclusiones finales..... 191

12. PRÓXIMOS PASOS 193

12.1 Autenticación y Autorización mediante Active Directory..... 194

12.2 Desarrollar el procesamiento de mails con imágenes, OCR 194

12.3 Agregar análisis de MetaData y más Algoritmos 195

12.4 Desarrollo de componentes ShakePlugin y ShakeProxy 196

12.5 Lograr una cobertura de pruebas unitarias suficiente 199

12.6 Desarrollo de nuevos dashboards y reportes 200

12.7 Explorar la posibilidad de extender a los clientes 200

REFERENCIAS BIBLIOGRÁFICAS 202

ANEXOS 204

Anexo 1 204

Anexo 2 206

Entrevista a usuarios finales 212

Entrevistas a Analistas de Seguridad de la Información 220

Anexo 3 232

Anexo 5 235

Anexo 5 246

Anexo 6 251

1. Introducción

El proyecto Shake Tool, surge a partir de la necesidad de la institución que hemos detectado, de enfrentar la problemática de la recepción de correos de contenido fraudulento por parte de nuestros clientes y funcionarios, e identificar mecanismos que los prevengan de ser víctimas de estas estafas.

La propuesta es realizar una prueba de concepto, que permita determinar la viabilidad de utilizar mecanismos automatizados de detección eficiente y temprana de mails con contenido fraudulento. Adicionalmente, proveer de información al equipo de Seguridad de la Información al respecto de la evolución de los indicadores relevantes al contexto del banco y en base a estos indicadores, tomar decisiones más acertadas en cuanto a las actividades o estrategias de control que se puedan aplicar.

1.1. Objetivos del Proyecto

Para la construcción de la Prueba de Concepto, es necesario contar con información real que sirva como insumo para los métodos automatizados que se utilizarán, en este caso algoritmos de Machine Learning. Para lograr el objetivo, estos algoritmos deben ser entrenados y testeados hasta obtener los resultados que nos hemos planteado para la POC. A su vez, proveer de mecanismos para la interacción entre los analistas de seguridad de la Información y el algoritmo entrenado para que puedan desarrollar la actividad.

El proyecto debería servir como evidencia de la utilidad de estas tecnologías, facilitando la tarea actual, además de servir de base para una futura implementación y proveer de información útil para la gerencia del área.

Por otra parte, como objetivo añadido, la utilización de esta capacidad de análisis mediante el uso de una interfaz que cumpla con las necesidades detectadas para cada uno de los interesados de este proyecto.

Objetivo	Métrica
Que el sistema detecte correos de intentos de phishing con una exactitud mayor al 80%.	Utilizamos la métrica ***accuracy_score*** en la fase de test de implementación del modelo de Machine Learning.
Que el sistema cumpla con los requerimientos funcionales y no funcionales que se definan dentro del alcance.	Realizaremos una encuesta para conocer si el cliente considera que se contemplaron los requerimientos funcionales. (El promedio de la encuesta deberá ser >= 4)
Brindar al equipo de Seguridad de la Información datos estadísticos sobre la detección de intentos de phishing	Realizaremos una encuesta para conocer si el cliente considera que la información aporta valor para su trabajo. (El promedio de la encuesta deberá ser >= 4)
Que agregue valor al trabajo de ciberseguridad del Banco.	Realizaremos entrevistas con los analistas de seguridad de la información para conocer si este trabajo aporta valor al trabajo de detección de phishing realizado por ellos.

1.2. Descripción del Cliente

El Banco de la República Oriental del Uruguay es el más importante en nuestro país. Desde su creación, ha jugado en forma ininterrumpida, un rol decisivo en el desarrollo económico del País, manteniendo por más de un siglo una sólida imagen avalada por el Estado Uruguayo, así como por su nivel de patrimonio y su reconocimiento internacional.

Hoy día, el Banco República cuenta con 124 dependencias distribuidas en todo el territorio nacional y 2 sucursales en el exterior.

Cuenta con más de 3300 funcionarios, y una cuota de mercado superior a la de los demás bancos. Apuesta activamente a la actualización tecnológica y a la utilización de herramientas avanzadas, a efectos de estar en la vanguardia y preparados para la dinámica de los negocios de esta época.

Foto: Diario El País 02/02/2021

A pesar de todas las medidas de seguridad ya implementadas, las distintas técnicas utilizadas por los ciberdelincuentes hacen muy difícil cubrir todo el espectro de posibilidades, con la dificultad de evitar que los correos de contenido fraudulento lleguen a las cuentas de correo electrónico del banco.

Sumado a ello, los conocimientos al respecto de la problemática del Phishing y sus riesgos dentro del universo de usuarios de correo electrónico del banco, es muy variada, siendo esto un problema a la hora de dar respuesta a estos casos.

Por tal motivo, surge la posibilidad de desarrollar una Prueba de Concepto que pretende utilizar la capacidad de aprendizaje automático y la capacidad de reconocimiento de patrones a través de técnicas de Machine Learning, para que los usuarios tengan una respuesta en menor tiempo frente a aquellos mails que resulten sospechosos o que eventualmente el sistema pueda detectar de manera autónoma.

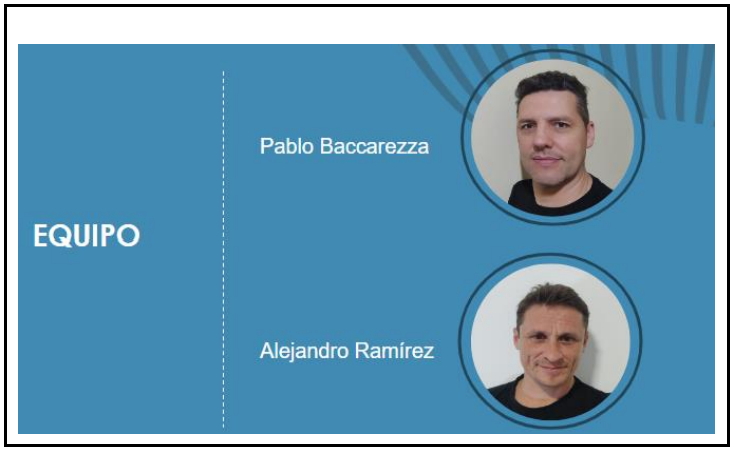
1.3. Descripción de los integrantes del proyecto

Los integrantes de este proyecto somos dos estudiantes de la facultad de ingeniería de la Universidad ORT Uruguay;

Pablo Baccarezza cuenta con conocimientos en el área de RRHH, específicamente en el área de capacitación, desde hace algunos años se desempeña como Analista de Seguridad de la información y es quien cuenta con experiencia de primera mano de la problemática planteada.

Alejandro Ramírez desempeña tareas de administración de sistemas UNIX/Linux perteneciente al departamento de infraestructura del BROU.

En ambos casos contamos con experiencia como participantes de proyectos dentro del banco de diferente tamaño, importancia y alcance.



1.3.1 Roles del proyecto

Debido al tamaño reducido del equipo, tenemos que cumplir múltiples roles. Formalmente hemos definido aquellos que por experiencia y/o preferencia se ajustan a las necesidades del proyecto:

Pablo Baccarezza	Alejandro Ramírez:
● Gestión de Proyecto ● Ingeniería de Requerimientos ● Diseño y Desarrollo	● Aseguramiento de la calidad ● Ingeniería de Requerimientos ● Diseño y Desarrollo

1.4 Motivación del Proyecto

El proyecto surge a partir de la necesidad de encontrar mecanismos que prevengan a los usuarios de correo de ser víctimas de estafas a partir de la llegada de una gran cantidad de correos de intentos de phishing tanto a nuestra institución como a nuestros clientes.

Los analistas de Seguridad de la Información (SI) se ven cada vez más, enfrentados a campañas de phishing. La cantidad de casos reportados, en algunos casos supera la capacidad del equipo que los atiende, además la respuesta a cada caso insume un tiempo considerable, por lo que esta situación se está convirtiendo en un problema.

Nuestra idea pretende dar un alivio a los analistas de SI, automatizando el análisis de cada correo para obtener un veredicto, además de que potencialmente podría ayudarlos con el proceso de responder en cada caso. Sumado a ello, se abre un abanico de posibilidades que incluye la de compartir la información de esas campañas de Phishing con los usuarios finales, tomando acciones preventivas, informándolos, abordando la problemática de la capacitación en el proceso.

1.5 Descripción del problema

1.5.1 Problema a nivel global

La seguridad del correo electrónico se refiere a varias medidas de ciberseguridad para asegurar el acceso y el contenido de una cuenta o servicio de correo electrónico.

La aplicación de medidas de seguridad adecuadas en el servicio de correo electrónico ayuda a proteger la información confidencial en las comunicaciones, prevenir ataques de

phishing, spear phishing y suplantación de identidad por correo electrónico y proteger contra el acceso no autorizado, la pérdida o el compromiso de una o más direcciones de correo electrónico.

Durante la pandemia del COVID-19, se ha observado un aumento significativo en los casos de phishing a través de correos electrónicos. Los ciberdelincuentes han aprovechado la incertidumbre y el miedo asociados a la pandemia para engañar a las personas y obtener información personal y financiera valiosa.

Entre los tipos de phishing más comunes durante la pandemia, se encuentran los correos electrónicos que supuestamente provienen de organizaciones de salud o de gobierno, que solicitan información personal o financiera para acceder a servicios o beneficios relacionados con la pandemia. También se han visto correos electrónicos que parecen provenir de empresas de suministros médicos o de fabricantes de vacunas, ofreciendo productos o servicios relacionados con la pandemia.

Además, se ha observado un aumento en los correos electrónicos de phishing que utilizan tácticas de ingeniería social para engañar a los usuarios. Por ejemplo, los correos electrónicos pueden hacer referencia a noticias o eventos actuales relacionados con la pandemia para aumentar la credibilidad del mensaje y persuadir a los usuarios a hacer clic en un enlace o descargar un archivo adjunto malicioso.

Es importante que los usuarios estén alerta a los correos electrónicos de phishing durante la pandemia y luego de ella, tomando medidas para proteger su información personal y financiera. Algunas medidas que se pueden tomar incluyen verificar siempre la legitimidad del remitente del correo electrónico antes de hacer clic en un enlace o descargar un archivo adjunto, y evitar proporcionar información personal o financiera a través de correos electrónicos no solicitados o de fuentes desconocidas.

1.5.2 Problema a nivel institucional

Como se explicó el BROU es una institución bancaria, por este motivo es un objetivo interesante para los ciber delincuentes que procuran explotar las vulnerabilidades o puntos más débiles. El phishing es una técnica de ingeniería social, que intenta engañar a las personas. Las estrategias usadas son variadas, y el phishing es una de ellas. Con frecuencia, contienen links, que redirigen hacia sitios maliciosos y/o que guían al usuario a la instalación de software con intenciones también maliciosas, conocido como Malware.

En cualquiera de los casos, las víctimas corren el riesgo de exponer información que puede ser explotada en otro momento, o bien sufrir pérdidas económicas. Como el mail que recibieron puede aparentar venir desde el Banco, los usuarios con frecuencia relacionan la institución con estos ataques generando un gran perjuicio en términos de reputación.

El Banco viene realizando campañas de concientización a través de diversos canales de comunicación, con el fin de informar a sus clientes sobre la existencia de esta modalidad de estafa. En el sitio transaccional del Banco e-BROU, se le aclara a los clientes que el Banco nunca solicitará validaciones de seguridad en ventanas emergentes:

En Instagram también se realizan campañas en este sentido:

A pesar de los esfuerzos en realizar comunicados y campañas de comunicación algunos clientes siguen siendo pasibles de este tipo de estafas.

1.6 Cómo trabajan hoy los analistas de Seguridad de la Información con respecto a este tema?

Los analistas de seguridad de la información tienen un procedimiento de acuerdo a como se describe a continuación:

1 – Reciben el correo sospechoso enviado por el centro de contacto.

2 – Revisa el dominio del remitente.

3 – Revisa la presencia de enlaces (link)

4 – Analiza el sitio del enlace.

5 – De ser necesario solicita la baja del sitio

1.7 Descripción de la solución

En la actualidad, el correo electrónico es uno de los principales medios de comunicación utilizados en el ámbito empresarial. Sin embargo, con el aumento del uso de Internet y la sofisticación de los ataques cibernéticos, se han vuelto más frecuentes los intentos de phishing y otros tipos de fraudes en los correos electrónicos. En este contexto, el uso de la Inteligencia Artificial (IA) y en particular, del Procesamiento del Lenguaje Natural (NLP) y del Aprendizaje Automático (Machine Learning), se ha vuelto una solución para proteger los correos electrónicos de estos ataques.

Los mecanismos de protección de correos electrónicos utilizando NLP y Machine Learning se basan en la capacidad de estos sistemas para analizar grandes cantidades de datos, detectar patrones y aprender de ellos. Estos mecanismos permiten detectar correos electrónicos sospechosos y filtrarlos antes de que lleguen a la bandeja de entrada del usuario.

En este sentido, este tipo de soluciones se enfocan en dos aspectos fundamentales: la detección de correos electrónicos maliciosos y la clasificación automática de los correos entrantes. Para la detección de correos electrónicos maliciosos, se utilizan técnicas de análisis de NLP para identificar patrones en el contenido del correo electrónico, como palabras clave, estructuras gramaticales y el tono general del mensaje. Para la clasificación automática de correos entrantes, se utilizan técnicas de aprendizaje automático para categorizar los correos electrónicos en diferentes grupos, según su contenido y otros factores como el remitente y la hora de envío.

En general, los mecanismos de protección de correos electrónicos utilizando NLP y Machine Learning pueden ayudar a mejorar la seguridad en línea y reducir el riesgo de ser víctima de ataques cibernéticos. Además, estas soluciones pueden ser personalizadas según las necesidades de cada usuario y adaptadas a los cambios en el panorama de la ciberseguridad.

Como ya explicamos el phishing es un intento de estafa que se realiza a través de distintos medios, el más frecuente es el correo electrónico. Como control a este intento de estafa proponemos realizar una prueba de concepto de un sistema que analice los correos electrónicos y los clasifique como correos con intentos de phishing o correos fidedignos (ham).

Contamos con una base de correos electrónicos registrados por el equipo de Seguridad de la Información como fuente de datos. Estos correos son valiosos porque contienen características particulares del contexto de nuestro Banco, por lo que son una buena fuente de aprendizaje para nuestros algoritmos.

Entonces nos planteamos crear un mecanismo que pueda tener acceso a esos correos, procesarlos, obtener la información relevante para nuestra casuística, para luego entrenar nuestros algoritmos y testearlos. Una vez completada esta etapa y si los resultados de los indicadores se ajustan a los objetivos que nos hemos planteado, procederemos a realizar el despliegue del modelo resultante para ser utilizado por el sistema.

Entendemos el desafío que implica poder ser precisos y consistentes con la información con la que contamos, pero también hemos procurado utilizar estrategias que nos permitan evaluar los algoritmos con un volumen de datos entendemos razonable.

1.8 Visión del producto

Deseamos apoyar a nuestros usuarios y clientes a trabajar en entornos digitales con más confianza y seguridad, siendo eficientes, adaptables y utilizando los mejores recursos tecnológicos disponibles.

1.9 Análisis de productos del mercado

Analizamos otras soluciones existentes en el mercado con la finalidad de comprender a alto nivel cómo funcionan y que prestaciones proporcionan. [8]

Feature/Solution	Spam filters	Customizable filtering	Malicious file identification	Integration	Report attacks	Malicious URL detection
Cofense	✓	✓	✓	✓	✓	✓
GreatHorn	✓	✓	✓	✓	✓	✓
IRONSCALES	✓	✓	✓	✓	✓	✓

https://www.spiceworks.com/it-security/vulnerability-management/articles/top-10-anti-phishing-software/

Cofense, GreatHorn e Ironscales son empresas de ciberseguridad que ofrecen soluciones antiphishing. Aunque presentan varias similitudes, también existen algunas diferencias entre ellas.

Cofense es una empresa especializada en la detección y respuesta al phishing. Ofrecen una amplia gama de soluciones, como formación en simulación de phishing, respuesta a incidentes de phishing e inteligencia sobre amenazas. Sus soluciones están diseñadas para

ayudar a las organizaciones a detectar y responder a los ataques de phishing de forma rápida y eficaz. Cofense se centra en el compromiso y la educación de los empleados, con el objetivo de crear una cultura de seguridad que ayude a prevenir los ataques de phishing.

GreatHorn es otra empresa de ciberseguridad que ofrece soluciones antiphishing. Proporcionan seguridad de correo electrónico y protección contra phishing a través de su plataforma basada en la nube, que utiliza aprendizaje automático e inteligencia de amenazas en tiempo real para detectar y bloquear ataques de phishing. GreatHorn también ofrece una serie de funciones, como cifrado de correo electrónico, prevención de pérdida de datos y respuesta a incidentes.

Ironscales es una empresa que utiliza inteligencia artificial y aprendizaje automático para ofrecer una plataforma integral de protección contra el phishing. Su solución incluye seguridad automatizada del correo electrónico, respuesta a incidentes e inteligencia sobre amenazas. El enfoque de Ironscales es proporcionar una respuesta proactiva y automatizada a los ataques de phishing, utilizando tecnología avanzada para identificar y prevenir los ataques en tiempo real.

En términos comparativos, Cofense se centra principalmente en la educación y el compromiso de los empleados, mientras que GreatHorn e Ironscales ofrecen soluciones antiphishing más completas que dependen en mayor medida de la tecnología. GreatHorn proporciona seguridad de correo electrónico y funciones adicionales como cifrado y prevención de pérdida de datos, mientras que Ironscales utiliza IA avanzada y aprendizaje automático para ofrecer un enfoque proactivo de la protección contra el phishing.

Cofense [9]

Se basa en un proceso que tiene las siguientes etapas:

- Formar a los empleados para que aprendan a detectar correos con intentos de phishing.
- Ofrecer una herramienta para reportar los correos con intentos de phishing.
- Ofrecer una herramienta que ayuda a los técnicos de monitoreo a detectar intentos de phishing en los correos.

- Buscar y poner en cuarentena correos a través de políticas definidas por el usuario.

GreatHorn [10]

Es una solución que funciona en la nube y tiene un software cliente que se instala en los equipos.

- Proporcionar a los usuarios información sobre correos electrónicos sospechosos para ayudarlos a tomar mejores decisiones.
- Permitir que los empleados etiqueten como spam un correo.
- Advertir y poner en cuarentena cuando detecta un correo sospechoso, tiene la habilidad de mostrar una captura de pantalla del sitio del enlace del correo sospechoso.

IronScales [11]

Está orientada para un entorno de correos en la nube. Permite realizar simulaciones antiphishing y capacitación personalizada basada en datos en tiempo real y situaciones del mundo real para capacitar a su equipo para reconocer, informar y resolver ataques.

Cuenta con un analista de seguridad impulsado por IA que permite que los técnicos de seguridad tomen decisiones más rápidas sobre correos electrónicos sospechosos en tiempo real automatizando los umbrales, el análisis y la cuarentena de amenazas.

A través de IA permite:

- Clasificar y responder automáticamente a los correos electrónicos informados por los empleados
- Agrupar de correos electrónicos sospechosos similares en un solo incidente
- Corrección automática de correos electrónicos ya entregados desde las bandejas de entrada
- Proporcionar sugerencias de incidentes impulsadas por IA para respaldar la toma de decisiones rápida

2. Análisis estratégico

2.1 Análisis PESTEL

El análisis PESTEL es una herramienta útil para examinar los factores políticos, económicos, sociales, tecnológicos, ambientales y legales que pueden afectar a una situación determinada. A continuación, se presenta un análisis PESTEL de la situación del phishing en Uruguay:

Político: Uruguay cuenta con leyes y regulaciones que protegen la privacidad y la seguridad de la información, como la Ley de Protección de Datos Personales y la Ley de Delitos Informáticos. Sin embargo, el gobierno también enfrenta desafíos en la lucha contra el cibercrimen debido a la falta de recursos y capacitación en materia de seguridad digital.

Económico: Uruguay es una economía emergente con un alto nivel de penetración de internet. El aumento de la digitalización y el comercio electrónico han aumentado el riesgo de ataques de phishing dirigidos a empresas y consumidores, lo que puede tener un impacto económico significativo en términos de pérdida de datos y reputación de marca.

Social: La educación en ciberseguridad es fundamental para combatir el phishing en Uruguay. Si bien cada vez más personas están tomando medidas para proteger su información personal y financiera, muchas aún son víctimas de fraudes en línea debido a la falta de conocimiento sobre cómo identificar y prevenir el phishing.

Tecnológico: Los avances tecnológicos han llevado a la creación de nuevas técnicas de phishing, como el spear phishing y el whaling, que se dirigen a individuos específicos o

empresas de alto perfil. Las soluciones tecnológicas, como los sistemas de detección de phishing, son esenciales para contrarrestar estas amenazas.

Ambiental: El cambio climático y los desastres naturales pueden tener un impacto indirecto en la lucha contra el phishing, ya que pueden interrumpir las operaciones empresariales y aumentar la vulnerabilidad de las redes y los sistemas.

Legal: Uruguay cuenta con un marco legal sólido para combatir el phishing, pero aún hay margen para mejorar las leyes y regulaciones en materia de ciberseguridad. Además, la cooperación internacional es esencial para combatir el phishing, ya que los delincuentes pueden operar desde cualquier parte del mundo.

El phishing es una amenaza importante en Uruguay debido al alto nivel de penetración de internet y la falta de educación en ciberseguridad en la sociedad. La adopción de soluciones tecnológicas, junto con una mejor regulación y cooperación internacional, son esenciales para abordar esta amenaza.

2.2 Situación del phishing en Uruguay

Según el último informe de la Unidad de Investigación de Delitos Informáticos de la Policía Nacional de Uruguay, los delitos informáticos aumentaron un 35% en 2020, en comparación con el año anterior. El phishing fue uno de los principales métodos utilizados por los delincuentes para obtener acceso no autorizado a sistemas y datos de las víctimas.

Las pequeñas y medianas empresas (PYMEs) son especialmente vulnerables al phishing en Uruguay. Según un estudio de la Cámara de Tecnologías de la Información y la Comunicación de Uruguay, el 60% de las PYMEs uruguayas no cuentan con medidas de seguridad adecuadas para protegerse contra el phishing.

El sector financiero es uno de los más afectados por el phishing en Uruguay. Las instituciones financieras son un objetivo común para los delincuentes que buscan robar información de tarjetas de crédito y datos bancarios de los clientes.

Las redes sociales también son un objetivo popular para los delincuentes de phishing en Uruguay. Las plataformas de redes sociales son un medio efectivo para los delincuentes para enviar mensajes falsos y engañar a las personas para que revelen información personal y financiera.

El gobierno uruguayo está tomando medidas para abordar la amenaza del phishing. En 2020, la Dirección Nacional de Telecomunicaciones y Servicios de Comunicación (Dinatel) lanzó una campaña de concientización sobre ciberseguridad dirigida a empresas y ciudadanos. Además, el Ministerio del Interior ha creado un grupo de trabajo para combatir el cibercrimen, incluido el phishing.

La amenaza del phishing en Uruguay es real y va en aumento. La educación en ciberseguridad, la adopción de medidas de seguridad adecuadas y la cooperación entre el gobierno y el sector privado son esenciales para proteger a las personas y las empresas contra esta amenaza.

2.3 Análisis de las 5 fuerzas de Porter

El análisis de las 5 fuerzas de Porter es una herramienta útil para examinar la competencia y las fuerzas que afectan a una industria determinada. A continuación, se presenta un análisis de las 5 fuerzas de Porter en el sector financiero uruguayo con respecto al phishing:

Amenaza de nuevos competidores: La amenaza de nuevos competidores en el sector financiero uruguayo es baja. Las barreras de entrada son altas debido a los requisitos regulatorios y de capital, lo que limita la entrada de nuevos participantes en el mercado. Además, las instituciones financieras establecidas tienen una ventaja competitiva en la construcción de relaciones de confianza con los clientes, lo que dificulta que los nuevos participantes ingresen y ganen una participación significativa en el mercado.

Poder de negociación de los proveedores: El poder de negociación de los proveedores es bajo en el sector financiero uruguayo. Los proveedores de servicios y productos de seguridad informática, como los proveedores de software de detección de phishing, son muchos y tienen poco poder de negociación individual. Además, las instituciones financieras tienen opciones de proveedores alternativos y pueden elegir entre diferentes opciones de productos y servicios.

Poder de negociación de los clientes: El poder de negociación de los clientes en el sector financiero uruguayo es moderado. Los clientes tienen una amplia gama de opciones de instituciones financieras y productos, lo que les da cierto poder de negociación. Sin embargo, las instituciones financieras tienen una ventaja en la construcción de relaciones a largo plazo con los clientes y en la protección de sus datos financieros y personales, lo que limita el poder de negociación de los clientes en términos de seguridad.

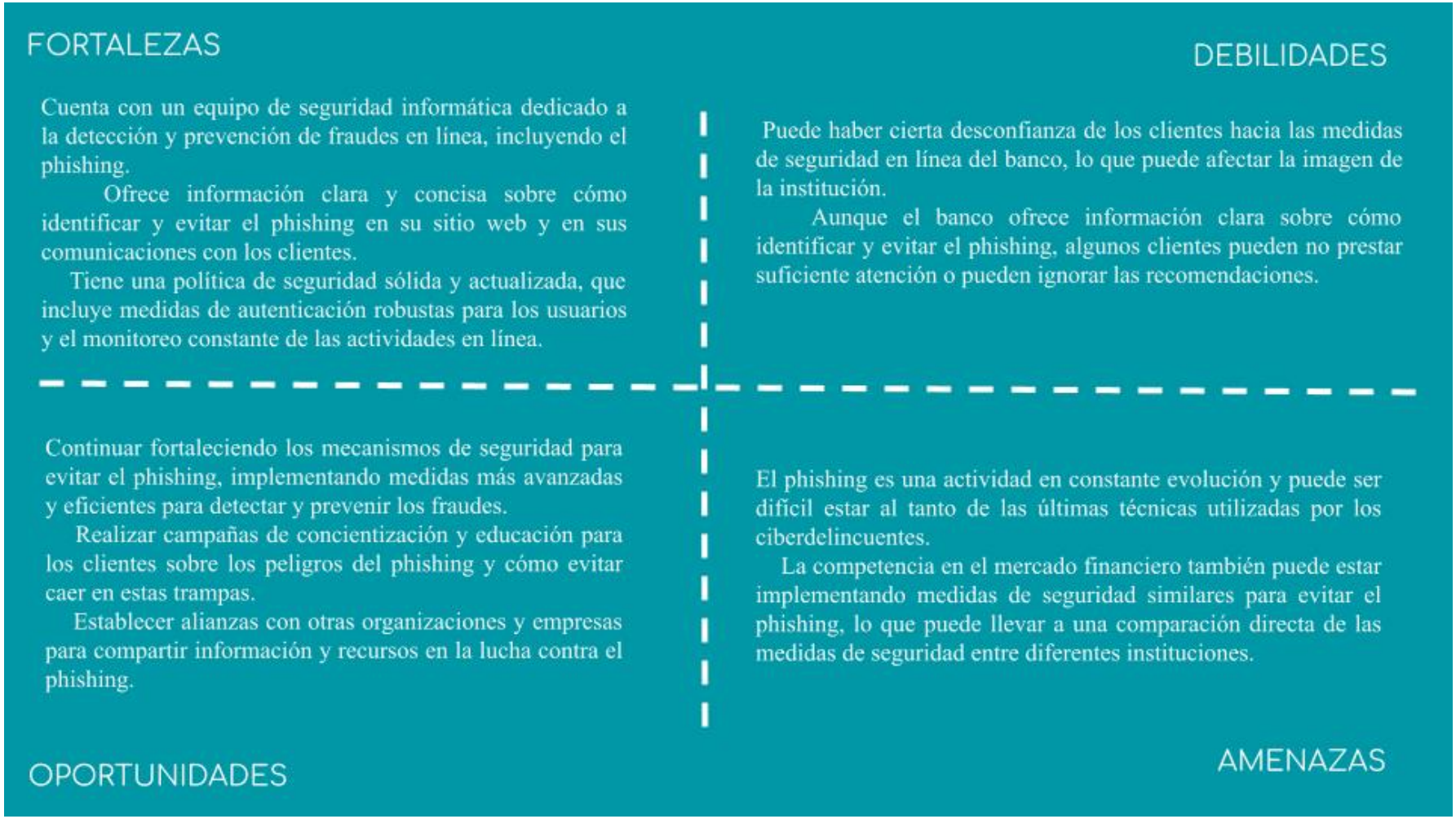
Amenaza de productos o servicios sustitutivos: La amenaza de productos o servicios sustitutivos en el sector financiero uruguayo es baja. Las instituciones financieras proporcionan servicios financieros y productos esenciales para los clientes, lo que limita la capacidad de los clientes para encontrar productos o servicios sustitutivos. Además, los servicios de seguridad informática y detección de phishing son necesarios para proteger a los clientes y a las instituciones financieras, lo que reduce la capacidad de los clientes para sustituir estos productos y servicios.

Intensidad de la rivalidad entre competidores existentes: La intensidad de la rivalidad entre competidores existentes en el sector financiero uruguayo es alta. Las instituciones financieras compiten en términos de precio, calidad del servicio y productos, y confianza del

cliente. La seguridad y la protección de datos son un factor crítico de diferenciación entre las instituciones financieras, lo que aumenta la importancia de la seguridad informática y la detección de phishing. Además, el phishing es una amenaza común que enfrentan todas las instituciones financieras, lo que aumenta la intensidad de la rivalidad en términos de seguridad.

El análisis de las 5 fuerzas de Porter sugiere que el sector financiero uruguayo se enfrenta a una amenaza alta de phishing, lo que aumenta la intensidad de la rivalidad entre competidores existentes en términos de seguridad. Las instituciones financieras necesitan tomar medidas efectivas para proteger a sus clientes y datos financieros, y diferenciarse en términos de seguridad para mantener su ventaja competitiva.

2.4 Análisis FODA del BROU en relación con el phishing:



3. Ingeniería de requerimientos

En esta sección, describiremos y desarrollaremos las técnicas y metodologías utilizadas, así como los desafíos y mejores prácticas para la gestión efectiva de los requerimientos del sistema.

La ingeniería de requerimientos es una disciplina fundamental en el desarrollo de software y sistemas. Su objetivo principal es asegurar que el software o sistema a construir satisfaga las necesidades y expectativas de los usuarios finales, así como los objetivos del negocio y los requisitos técnicos. Para lograr este objetivo, la ingeniería de requerimientos se enfoca en identificar, analizar, especificar, validar y gestionar los requerimientos del sistema a lo largo del ciclo de vida del desarrollo de software.

Ilustración 3-1 - Etapas de la ingeniería de requerimientos

3.1 Planificación de la etapa

La etapa de relevamiento de requerimientos es una fase crucial en cualquier proyecto de desarrollo de software, ya que permite definir claramente los objetivos del proyecto, los requisitos y las expectativas del cliente. Para la planificación tuvimos en cuenta los siguientes aspectos:

Objetivos del relevamiento: Establecer qué se espera lograr durante esta fase. ¿Cuáles son los objetivos del proyecto? ¿Qué se espera lograr con el software? ¿Cuáles son las necesidades y expectativas del cliente?

Stakeholders: Seleccionamos un equipo de trabajo que esté compuesto por personas con diferentes habilidades y conocimientos. Esto incluye a los analistas de seguridad de la información, Analistas de operaciones, Ejecutivos de Negocio, Ejecutivos de Crédito Social, Analistas de Finanzas, etc.

Alcance del proyecto: Es importante establecer los límites del proyecto. Esto te permitirá enfocarte en los objetivos del proyecto y evitar desviaciones en el proceso.

Plan de trabajo: Definimos un plan de trabajo que incluye las tareas a realizar, los plazos y los recursos necesarios para llevar a cabo el relevamiento.

Seleccionamos las técnicas de relevamiento: Seleccionamos las técnicas de relevamiento más apropiadas para nuestro proyecto: Entrevistas, cuestionarios, reuniones, prototipos, entre otros.

Relevamiento: Una vez definido el plan de trabajo, comenzamos con el relevamiento.

Procuramos documentar toda la información relevante y mantener una comunicación constante con los stakeholders.

Validación la información: Una vez finalizado el relevamiento, validamos la información obtenida con los stakeholders para asegurarnos de que se hayan entendido correctamente los requerimientos y expectativas.

Elaboración de un informe: Elaboramos un informe con los resultados del relevamiento, incluyendo los requerimientos, las expectativas y las limitaciones del proyecto.

El 5/11/22 comenzamos con la planificación de la etapa de ingeniería de requerimientos, donde definimos las técnicas que emplearemos para el relevamiento:

- Entrevistas con los Analistas de Seguridad
- Entrevistas con los usuarios finales
- Observación de la problemática diaria
- Brainstorming

Una vez relevados los requerimientos comenzamos la etapa de análisis, luego especificamos los requerimientos y la validación la realizamos presentando los prototipos a los analistas de seguridad de la información, y con una reunión con el gerente de seguridad de la información. Ver sección 3.6 Validación de Requerimientos

3.2 Relevamiento de requerimientos

El relevamiento de requerimientos es un proceso crucial en el desarrollo de software, ya que ayuda a entender las necesidades y expectativas de los usuarios finales y a definir qué características y funcionalidades son necesarias para cumplir con sus requerimientos.

En esta etapa, es importante establecer una comunicación clara y efectiva con los stakeholders involucrados, para asegurarnos de que comprendemos completamente sus necesidades y expectativas. Además, debemos considerar tanto los requisitos funcionales como los no funcionales, tales como la usabilidad, la seguridad y el rendimiento.

En este proceso, también es fundamental utilizar herramientas y técnicas adecuadas para recopilar, analizar y documentar los requerimientos, tales como entrevistas, encuestas, diagramas de casos de uso, prototipos y documentación.

Para el relevamiento de requerimientos realizamos entrevistas a los stakeholders. Preparamos una presentación, que nos fue guiando a lo largo de la entrevista:

Durante la entrevista se fueron realizando preguntas a los compañeros, y procuramos captar la mayor cantidad de información posible de la experiencia relatada por ellos. Dentro de la entrevista preparamos un espacio para realizar un brainstorming, donde los compañeros pudieran tratar de vislumbrar una solución a este problema desde sus perspectivas.

La técnica brainstorming permitió que pudieran hablar más abiertamente sobre el tema.

En total realizamos 8, a 11 funcionarios entre usuarios finales y Analistas de Seguridad de la Información:

Entrevista Nro.	Nombre	Cargo	Modalidad
1	Ing. Rodrigo Mateos	Analista de Seguridad de la Información	Presencial
2	Ing. Alejandro Cao	Analista de Seguridad de la Información	Presencial
3	Ing. Federico Zubiri	Analista de Seguridad de la Información	Presencial
4	Laura Fontana	Analista de Seguridad de la Información	Presencial
5	Ticiana Martínez	Ejecutiva de negocios	A distancia
6	Ec. Miguel López	Analista Financiero	A distancia
7	Cr. María Noel Alberro	Analistas de Operaciones	Presencial
7	Ignacio Acuña	Analistas de Operaciones	Presencial
8	Lic. Andrea Delgado	Supervisora de Procesos	A distancia
8	Ec. Michel Godín	Ejecutivo de Gestión Humana	A distancia
8	Javier Casal	Ejecutivo de Crédito Personas	A distancia

3.3 Interesados

Los stakeholders o partes interesadas en la ingeniería de requerimientos son aquellos individuos o grupos que tienen un interés en el software o sistema que se está desarrollando. En general, los stakeholders son cualquier persona que se vea afectada por el software o sistema, ya sea directa o indirectamente.

Algunos ejemplos de stakeholders en la ingeniería de requerimientos pueden incluir:

- Usuarios finales del software o sistema
- Clientes o patrocinadores del proyecto
- Gerentes y miembros del equipo de desarrollo de software
- Analistas de negocio
- Diseñadores de sistemas
- Arquitectos de software
- Especialistas en calidad de software
- Reguladores y organismos gubernamentales que rigen el uso del software o sistema

La identificación y gestión efectiva de los stakeholders es fundamental para el éxito de la ingeniería de requerimientos. Cada stakeholder puede tener diferentes necesidades, expectativas y requisitos del software o sistema, por lo que es importante comprender sus perspectivas y considerarlas en la definición y especificación de los requerimientos del sistema.

La siguiente tabla muestra la lista de interesados y sus responsabilidades dentro del Banco:

Interesado	Responsabilidades
Gerente del Área de Seguridad de la información	Principal interesado, gerencia que pertenece al área de Políticas y control de riesgo
Analista de Seguridad de la información	Encargados de Monitorear, Investigar, y desarrollar lineamientos y políticas de seguridad. Son los encargados de determinar si los correos reportados son casos positivos o no.
Usuario final	Funcionarios del BROU, con cuentas de correo corporativo. Desde atención al cliente hasta gerencias

Dentro de las tareas del funcionario del centro de contacto, se encuentra la de recibir consultas telefónicas o a través del correo electrónico de los clientes. Una de las consultas que reciben es si un correo recibido por un cliente es del Banco o se trata de un intento de estafa. Dentro de la prueba de concepto está prevista una interfaz que le permitirá al funcionario del centro de contacto analizar de forma sencilla un correo recibido por un cliente y que le permitirá rápidamente determinar si el correo se trata de un intento de phishing o no.

Dentro de las tareas de Analista de Seguridad de la Información se encuentra la de analizar si un correo recibido desde el centro de contacto es phishing o no, y en caso de que sea phishing realizar todas las acciones correspondientes.

Dentro de la prueba de concepto está prevista una interfaz que le permite visualizar todos los correos que ya fueron pre analizados por el funcionario del centro de contacto, en esta interfaz podrán ver el remitente, el cuerpo del correo, los links y la evaluación que brindó el sistema, es decir si es phishing o no y con qué exactitud.

El Gerente de Seguridad de la Información si bien no tendrá una interfaz específica, si podrá acceder a los dashboards donde se mostrarán datos estadísticos que describen la evolución del tratamiento de los correos electrónicos.

3.4 Especificación de requerimientos

La especificación de requerimientos es una actividad clave en la ingeniería de software y sistemas, ya que establece las bases para el diseño y desarrollo del software o sistema. En general, la especificación de requerimientos implica identificar, definir y documentar los requisitos funcionales y no funcionales del software o sistema, así como las restricciones y limitaciones a tener en cuenta durante el desarrollo.

Una especificación de requerimientos clara y completa es esencial para garantizar que el software o sistema cumpla con las expectativas de los stakeholders y los objetivos del negocio. Además, una buena especificación de requerimientos ayuda a minimizar los riesgos de errores y malinterpretaciones en el desarrollo, lo que a su vez reduce los costos y tiempos asociados con la corrección de errores en etapas posteriores del proyecto.

En esta sección mostramos cómo realizamos el registro de los requerimientos del usuario y del sistema. Procuramos que tanto los requerimientos del usuario como los del sistema se escribieran de forma clara, tratando de no tener ambigüedades, que sean fáciles de entender, completos y consistentes. Los requerimientos del usuario describen los requerimientos funcionales y no funcionales, de la forma más clara posible para los usuarios del sistema que no cuentan con un conocimiento técnico. Decidimos escribir los requerimientos del usuario en lenguaje natural y en tablas de una forma sencilla.

La especificación de requerimientos está organizada en requerimientos funcionales, requerimientos no funcionales y restricciones.

Requerimientos funcionales

RF 1	Modelos de Machine Learning desplegados.
Descripción:	El sistema clasificará los correos como un intento de phishing o no con una cierta precisión, en función del entrenamiento que ha recibido hasta el momento.
	Mandatorio
RF 2	Distribuir notificaciones.
Descripción:	El sistema en su versión completa, distribuirá mensajes con notificaciones relevantes de nuevos casos, nuevas campañas, etc. a los usuarios.
	Deseable
RF 3	Distribuir nuevas definiciones.
Descripción:	El sistema distribuirá una nueva versión del algoritmo de machine learning una vez que haya sido reentrenado con nuevos casos.
	Deseable
RF 4	Distribuir definiciones previas
Descripción:	El sistema distribuirá una versión previa del algoritmo de machine learning para hacer rollback de una nueva versión del algoritmo de machine learning en caso de que fuera incluida por error.
	Deseable
RF 5	Distribución de definiciones
	Descripción: Se distribuirá a los clientes de correo (Outlook) un archivo con la última definición del algoritmo de machine learning a fin de hacer un análisis con recursos locales. Su ejecución será automática y agendada.

	Deseable
RF 6	Mantener un registro de eventos
Descripción:	Se mantendrá un registro de eventos, tanto de consultas, como distribución y/o rollback de versiones de algoritmos de machine learning.
	Deseable
RF 7	Mantener un registro de accesos al sistema
Descripción:	Se mantendrá un registro de los accesos al sistema por parte de los analistas de seguridad de la información.
	Deseable
RF 8	Reporte de correo sospechoso.
Descripción:	El sistema en su versión final le permitirá al usuario reportar un correo sospechoso en caso de que el sistema no lo detecte, mediante un plugin en Ms Outlook, para que un analista de seguridad de la información realice un análisis más detallado.
	Deseable
RF 9	Feedback hacia el usuario final del caso reportado
Descripción:	El sistema deberá enviar una notificación a través de un correo electrónico al usuario final informando el resultado del análisis del correo que él envió.
	Deseable

RF 10	Inclusión de nuevos correos para analizar.
Descripción:	El sistema permitirá incorporar nuevos correos electrónicos al sistema, ya sea mediante un comando o una interfaz gráfica (web o escritorio).
	Mandatorio
RF 11	Autenticación y autorización.
Descripción:	Para ingresar al sistema, los analistas de seguridad de la información deben autenticarse mediante usuario y contraseña. Luego a través de un control de acceso y de acuerdo a sus permisos o privilegios, podrá acceder a aquellos recursos autorizados.
	Mandatorio
RF 12	Gestión de usuarios
Descripción:	Alta Usuario: El sistema debe permitir a los administradores del sistema agregar un nuevo usuario.
	Baja Usuario: El sistema debe permitir a los administradores del sistema, borrar un usuario existente.
	Modificación Usuario: El sistema debe permitir a los administradores del sistema, modificar datos del usuario: Nombre, Apellido, descripción, contraseña, desbloqueo, etc.
	Mandatorio
RF 13	Datos que se despliegan, listado de links
Descripción	El sistema debe desplegar la lista de links que contiene un mail para facilitar su visualización y detección.
	Deseable

RF 14	Disponer de datos estadísticos específicos
Descripción:	El sistema debe Mostrar: ● %de eficiencia del algoritmo ● % de casos detectados (detectados - reportados) ● de estos últimos, cuáles fueron falsos positivos y cuáles no.
	Mandatorio
RF 15	Ejecución de análisis a demanda
Descripción:	El plug in instalado en el cliente de correo les permitirá a los usuarios finales ejecutar el algoritmo de machine learning a demanda.
	Deseable
RF 16	Bloquear el acceso a links en casos detectados como “positivos”
Descripción:	El plug in instalado en el cliente de correo deberá bloquear los links de los correos que sean detectados como positivos.
	Deseable
RF 17	Los usuarios podrán enviar correos electrónicos sospechosos para su análisis.
Descripción:	El usuario final podrá enviar utilizando el plug in instalado en el cliente de correo, un correo electrónico para su análisis en profundidad en caso de que no sea detectado como phishing automáticamente. ◦ Debe enviar al menos los siguientes datos (de manera automática): ▪ Email del usuario afectado ▪ fecha y hora Descripción por parte del usuario del motivo del envío manual
	Deseable

Requerimientos no funcionales

RNF 1	Machine Learning Accuracy
Descripción:	El sistema debe utilizar un algoritmo de machine learning para la clasificación de correos electrónicos. El algoritmo deberá tener una exactitud mayor o igual al 80%.
RNF 2	Mantenibilidad
Descripción:	El sistema se construirá de forma que sea sencillo su mantenimiento o realización de modificaciones.
RNF 3	Usabilidad
Descripción:	El envío de notificaciones se realizará únicamente cuando sea necesario, para no abrumar a los usuarios finales, procurando que las comunicaciones cumplan el objetivo de concientizar a los usuarios finales sobre los riesgos asociados a los correos electrónicos con intento de phishing.
RNF 4	**Cloud Computing**
Descripción:	Por lineamientos tanto del Banco Central como internos, no es posible actualmente realizar el procesamiento de correos electrónicos en la nube.
RNF 5	Infraestructura definida
Descripción:	Se usará tecnología de virtualización para los servidores, cuyos sistemas operativos serán RedHat en la última versión disponible compatible con la solución. En el momento de redactar este requerimiento no es posible usar plataformas Windows Server.
RNF 6	Infraestructura definida

Descripción:	Se usarán solamente estaciones de trabajo dentro del Banco, con los sistemas operativos estándares y navegadores homologados. Windows 10 y Windows 11, navegadores Edge, IE y Chrome
RNF7	Infraestructura definida
Descripción:	Se usará solamente clientes de correo Microsoft Outlook.
RNF8	Infraestructura definida
Descripción:	No se usará OWA (Outlook Web Application)
RNF9	Infraestructura definida
Descripción:	Es necesario utilizar servidores intermedios (Proxys) para contemplar los requisitos de seguridad y segregación de redes, así mismo como contemplar el costo administrativo.

3.5 Priorización de requerimientos

La priorización de requerimientos la realizamos teniendo en cuenta primero las funcionalidades necesarias para que la prueba de concepto se pueda realizar, es decir aquellas que permitan evaluar si es factible que se pueda implementar en el futuro una solución de software. Para esto realizamos una categorización de los requerimientos funcionales en mandatorio y deseable. La idea de utilizar la palabra mandatario obedece a que es necesario desarrollar estas funcionalidades como se explicó precedentemente para que la prueba de concepto pueda ser evaluada por los analistas de seguridad de la información, determinando su utilidad y/o posibilidad de implementación futura. La categoría deseable tiene que ver con requerimientos que fueron relevados pero que si no se implementan no influyen significativamente en el resultado de la prueba de concepto. Lo explicado anteriormente también obedece a un orden lógico para la etapa de diseño, desarrollo y pruebas, ya que cada iteración se basa en lo construido anteriormente, y para esto las funciones básicas se deben desarrollar primero. Desde el punto de vista de gestión de proyecto esta priorización nos determina el alcance de este proyecto.

Todo lo explicado anteriormente se validó con el cliente al momento de realizar la validación formal de los requerimientos.

3.5.1 Alcance inicial

Debido al tamaño del proyecto y del equipo, al tiempo disponible y los requerimientos relevados, es que se toma la decisión de ajustar el alcance del mismo para que el mismo sea más realista y alcanzable, debiendo centrarse en aquellos aspectos fundamentales sobre los que se basaran las funcionalidades planteadas.

Cuando se recorta el alcance del proyecto, se eliminan algunos objetivos, entregables, tareas o actividades para reducir la complejidad del proyecto y hacerlo más manejable dentro de los recursos disponibles. Este proceso implica la revisión y reevaluación de los objetivos del proyecto y la identificación de aquellos que pueden ser eliminados sin afectar significativamente los resultados finales del proyecto.

Por lo expresado anteriormente, y por la magnitud del sistema planteado en su completitud supera las capacidades del equipo, entonces centraremos los esfuerzos en torno a los mecanismos de despliegue y utilización de los modelos de Machine Learning. Esto deja para instancias futuras lo relacionado a la conectividad directa desde los clientes Outlook y toda su lógica, como el desarrollo e implementación de los componentes Proxy de la solución total. (Ver Ilustración 3.5-1 Alcance propuesto para la prueba de concepto)

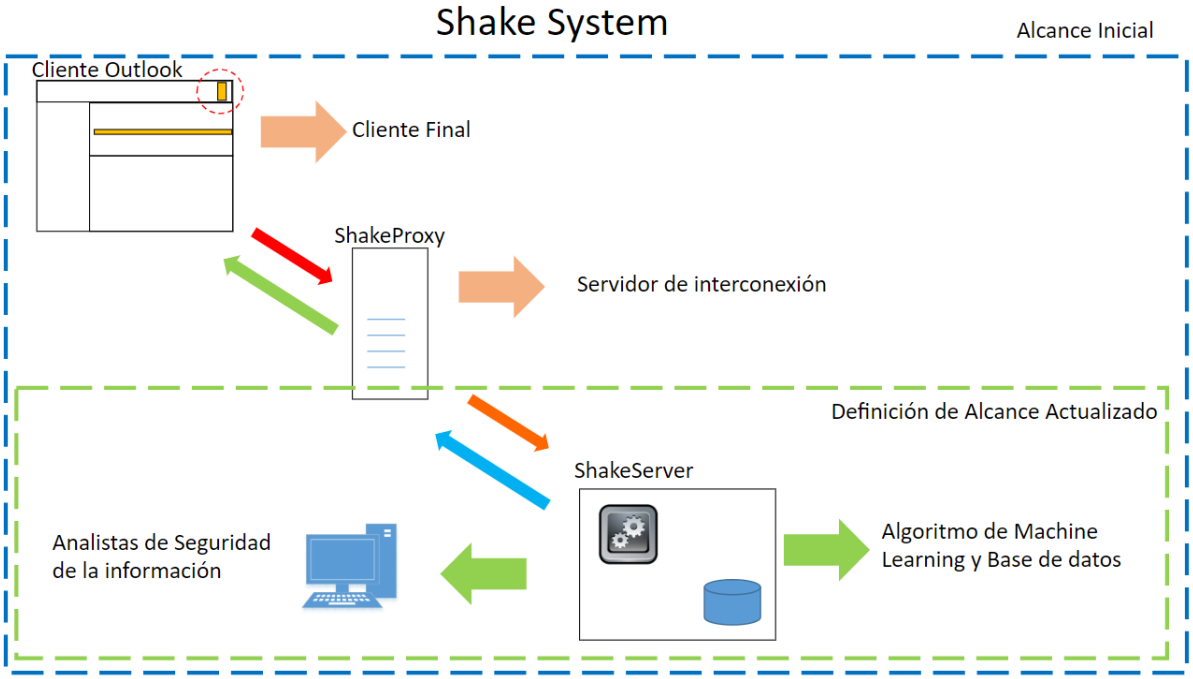


Ilustración 3.5.1-1 Alcance propuesto para la prueba de concepto

Esto implica que los usuarios objetivo de la prueba de concepto se reducen específicamente a los analistas de seguridad de la información, y de la interacción de estos usuarios con el sistema.

En su concepción total, el sistema incluye elementos intermedios de comunicaciones "ShakeProxys", cuya función principal es la de concentrar peticiones de clientes distribuidos, y servir de procesamiento intermedio de ser necesario para evitar tráfico de red y segregar segmentos de red en términos de seguridad. También el desarrollo de componentes integrados en Microsoft Outlook, que permiten un conjunto de acciones automáticas de notificaciones tanto hacia el servidor como notificaciones de avisos hacia los propios usuarios, esto último siendo muy importante dentro del contexto del phishing como problemática de seguridad. Esto quedará entonces para etapas posteriores.

3.6 Validación de requerimientos

La validación de requerimientos es un proceso esencial en el desarrollo de software y otros proyectos de ingeniería. Consiste en verificar que los requerimientos especificados sean correctos, completos, coherentes, realizables y relevantes para el proyecto. La validación de requerimientos ayuda a garantizar que el resultado final del proyecto sea satisfactorio y cumpla con las necesidades y expectativas del cliente y de los usuarios finales. En este proceso, se utilizan diferentes técnicas y herramientas para evaluar la calidad de los requerimientos y asegurar que sean verificables y validables. En este sentido, la validación de requerimientos es una etapa fundamental para el éxito del proyecto para lograr la satisfacción del cliente.

La validación de requerimientos la realizamos en dos instancias, en la primera le mostramos a los analistas de seguridad de la información un prototipo de la prueba de concepto, para conocer su opinión acerca de si lo que relevamos y analizamos es lo que ellos esperaban. Este prototipo inicial, intentaba en una primera instancia reflejar en un sistema semi funcional, las ideas que habíamos capturado.

Algunas de las imágenes del prototipo, que mostramos, en su versión de aplicación de escritorio tenía el siguiente aspecto: Ver ilustraciones 3.6-1, 3.6-2, 3.6-3 y 3.6-4

Ilustración 3.6-1 Prototipo Login

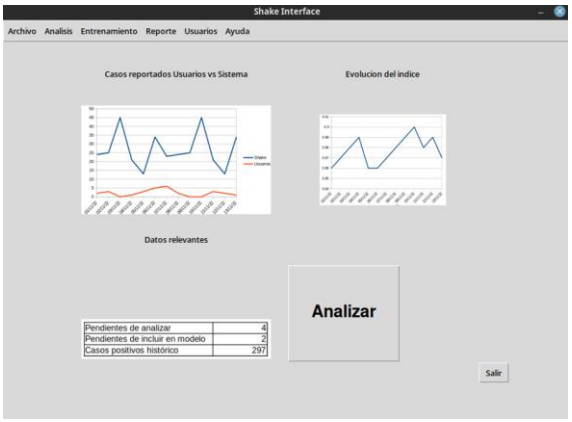


Ilustración 3.6-2 Prototipo Dashboard Inicial

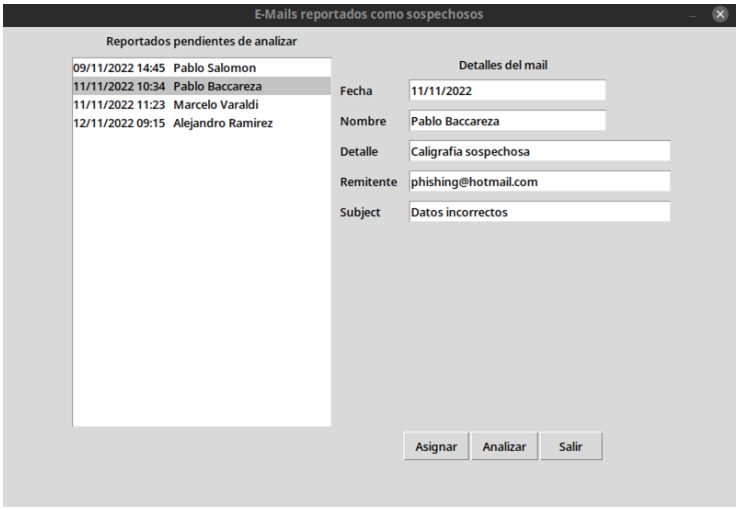


Ilustración 3.6-3 Prototipo listado emails sin procesar

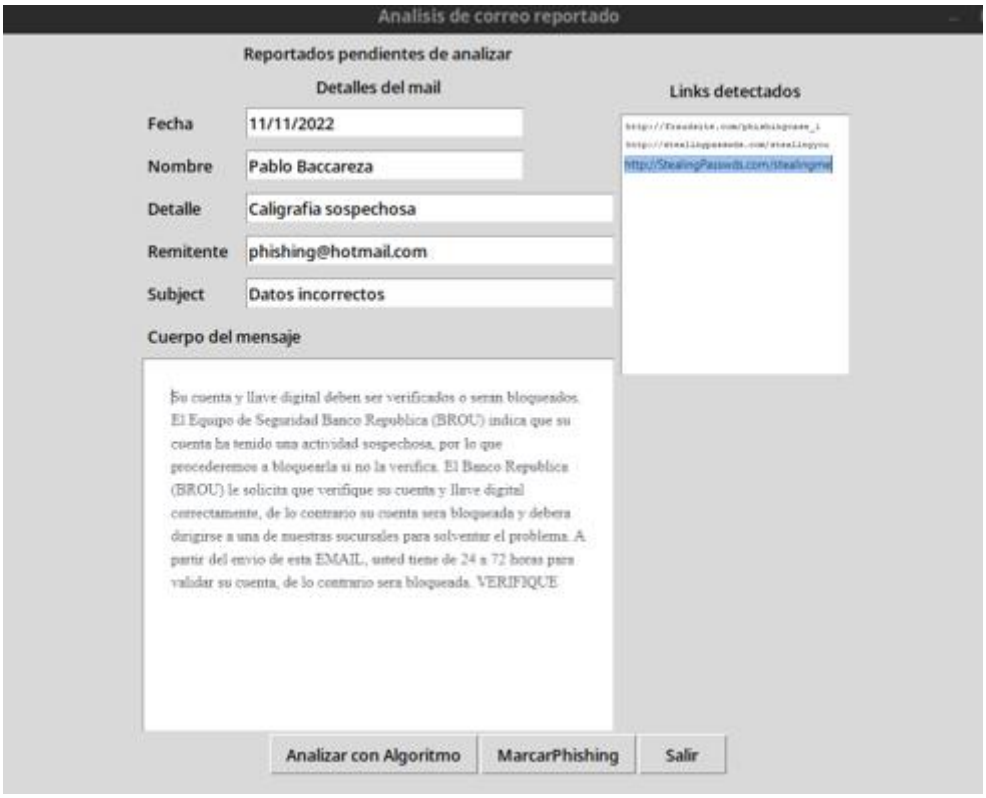


Ilustración 3.6-4 Prototipo detalle de email seleccionado para procesar

Esta presentación del prototipo nos ayudó de gran manera, pues nuestros principales interesados pudieron comprobar de cierta manera lo que estábamos pensando, cual era nuestro rumbo. Además, fue un puntapié para otra gran decisión dentro del proyecto, que implicó dejar de lado la opción aplicación de escritorio, para enfocarse en la construcción de una aplicación de enfoque WEB.

La segunda instancia fue con el Gerente de Seguridad de la Información donde discutimos sobre las distintas etapas de la ingeniería de requerimientos, y finalmente fue él quien validó el documento ESRE.

#	Descripción de la Acción
1	Reunión con los analistas de seguridad de la información para mostrarles un prototipo de la **prueba de concepto** para recoger, modificar o eliminar requerimientos.
2	Reunión con Gerente de seguridad de la información para intercambiar sobre la etapa de ingeniería de requerimientos. Finalmente validar los requerimientos.

Tabla 3.6-1

3.7 Conclusiones

En conclusión, la etapa de relevamiento de requerimientos fue fundamental para establecer una base sólida para el éxito del proyecto. Al identificar los objetivos, limitaciones, funcionalidades y características del producto, se puede crear una solución que satisfaga las necesidades del cliente y asegure su satisfacción al final del proyecto. También ayudó a detectar los principales interesados, restricciones que inicialmente no su hubiesen detectado, pero sobre todo el alcance del sistema, base para la toma de decisiones de acuerdo a los recursos disponibles.

Una vez que determinamos el alcance, las opciones disponibles para continuar con el proyecto nos dirigieron a tomar la decisión de trabajar sobre una prueba de concepto. Para enfocar esfuerzos, dentro de los tres primeros RNF nos concentramos en el RNF1 - Machine Learning, y lo relacionado a lo que sería el corazón de todo lo que luego se construiría.

El problema que encontramos en esta etapa fue la falta de experiencia, ya que se requiere habilidades técnicas y experiencia para recopilar, analizar y documentar los requisitos de manera efectiva. La falta de experiencia en esta área nos llevó a tener que revisar cuidadosamente cada paso de esta etapa, haciendo foco en la comunicación con los interesados para asegurarnos que comprendimos y registramos correctamente los requerimientos.

En cuanto a la interacción con los usuarios, cabe destacar que somos integrantes de áreas relativas a TI, y específicamente relacionadas a Seguridad de la Información. Esto es significativo ya que se conoce muy bien la problemática, pudiendo tener un intercambio fluido con los usuarios finales, comprendiendo cabalmente sus necesidades y en ocasiones sirviendo de nexo para la comprensión del lenguaje técnico y de cómo pensamos abordar el desafío.

4. Arquitectura

El diseño arquitectónico de software es un proceso crucial en el desarrollo de cualquier proyecto de software. Se trata de la creación de una estructura sólida y bien definida que permita la construcción de software escalable, robusto y fácilmente mantenible. El diseño arquitectónico de software se enfoca en la organización de los componentes de software y sus interacciones, para lograr una solución óptima en términos de rendimiento, seguridad, flexibilidad y escalabilidad. La arquitectura del software tiene una influencia significativa en la calidad final del producto, y un diseño arquitectónico sólido puede marcar la diferencia entre un software exitoso y uno que no cumple con las expectativas del cliente.

En nuestro caso, una vez finalizada la etapa de ingeniería de requerimientos, comenzamos con la etapa de arquitectura y modelado de software y para esto identificamos a los usuarios del sistema y la interacción que tendrán tanto con nuestra prueba de concepto, como con los sistemas vinculados al objeto de este trabajo. Los requerimientos relevados en la etapa anterior se utilizaron como insumo para el diseño arquitectónico.

Para diseñar la arquitectura se tuvieron en cuenta los requerimientos no funcionales existentes en el Banco. Estamos en conocimiento que se cuenta con la infraestructura necesaria para la potencial implementación de este sistema.

La instalación de un cliente de escritorio parecía no ser la opción apropiada para este caso, ya que requiere la instalación en cada puesto de trabajo. Decidimos utilizar una arquitectura que contemple el uso de navegadores web para que los funcionarios que requieran utilizar esta aplicación lo puedan hacer desde cada una de las terminales habilitadas.

Se sabe, además, que el banco tiene estándares en términos de tecnologías, ya sea por su homologación como por licenciamiento. Dicho esto, el uso de la tecnología de base de datos para la prueba de concepto fue MySql, que, si bien no es el estándar principal, si existen instalaciones actualmente pero que debe ser debidamente justificadas. En caso contrario, hacer una adaptación de los modelos de datos del sistema a esquemas dentro de las Bases de datos homologadas, como las de Oracle o SQL. Una mención específica a este caso se puede ver en el apartado de Aseguramiento de la calidad (Capitulo 9), donde desde el equipo de Arquitectura Tecnológica del banco, durante el proceso de validación de la arquitectura propuesta, nos hacen notar este punto.

Otra de las consideraciones que tuvimos en cuenta, a pesar de ser una POC, fue que en un futuro los usuarios finales, y no solo los analistas de SI pudiesen consumir esos servicios. El uso de API Rest fue la opción más clara ya que permitiría tanto crear una interfaz web para los analistas, además cómo presentar una interfaz hacia los usuarios finales y una interfaz hacia la aplicación o plugin de Outlook que se plantea desarrollar posteriormente conjuntamente con la API.

La solución planteada utiliza el framework de desarrollo DJANGO, cuyo lenguaje Python facilita una homogénea relación con código que pueda ser creado por fuera del framework.

La estrategia actual para el entrenamiento de los algoritmos, creación de los modelos y despliegue de los mismos se ejecuta de manera independiente. Esto está alineado con las prácticas habituales dentro del despliegue de modelos de Machine Learning, pues los modelos necesitan ser evaluados en ambientes controlados para determinar su performance previo a su

puesta en producción. Uno de los atributos considerados es el Accuracy, pero también se consideró el atributo Recall, (Ver Sección 6.2.3 Model Training and Evaluation) relativo al desbalance de correos de cada tipo, como factor de decisión de elección de algoritmo. Accuracy debe tener una exactitud superior al 80% según lo descrito en el RNF'1, Ver Sección 3.4 Especificación de Requerimientos

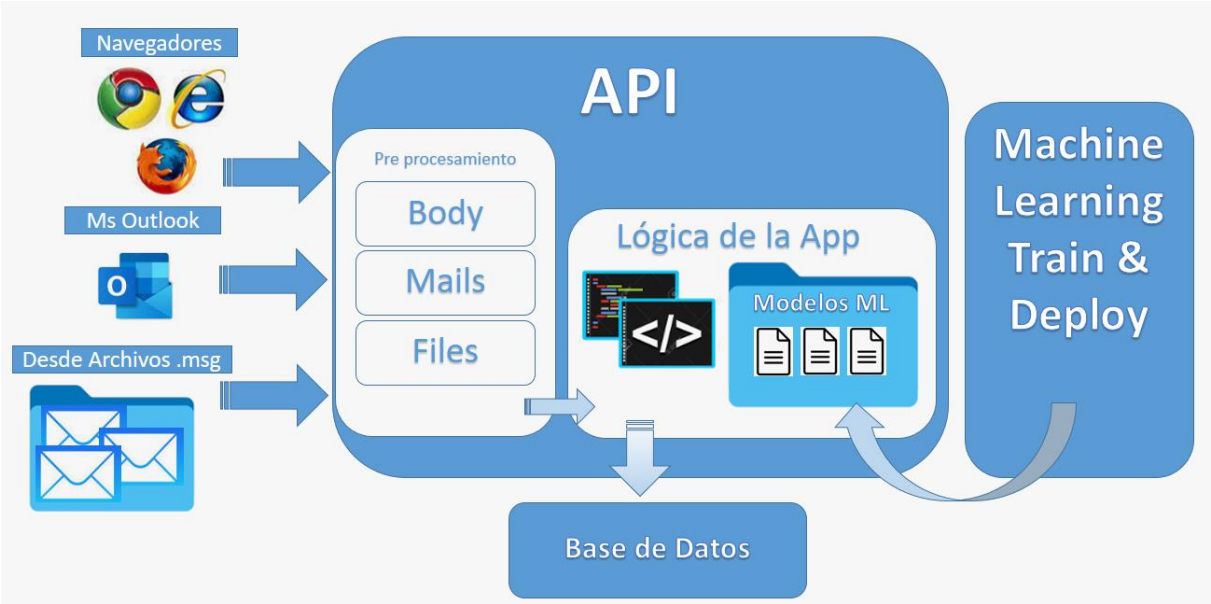


Ilustración 4-1

Los Drivers de arquitectura se resumen en la siguiente lista:

- Acceso vía web por parte de los clientes y equipo de seguridad de la información
- Acceso vía plugin Outlook que consumirá API

- Bases de datos (que pueda justificarse y administrarse por el equipo de Base de datos del BROU)
- Alcance de la solución solamente en la red interna del banco
- Deseable Autenticación de usuarios mediante Active Directory

Para cumplir con estos requerimientos se decidió utilizar una estrategia de capas, la cual propone dividir el sistema en capas en donde se agrupan las piezas de código de forma coherente y desacoplada. Como hemos mencionado, la utilización de un framework de desarrollo de aplicaciones web de código Python, como lo es Django, tiene muchas ventajas en cuanto a las facilidades que brinda. Una de esas facilidades es la utilización de ORM, que es una de las características más poderosas y realiza un mapeo relacional de Objetos (ORM), que le permite interactuar con su base de datos, como lo haría con instrucciones SQL.

Esto provee una abstracción y facilidad de acceder y modificar los atributos y tablas con la ejecución de comandos sencillos. Como desventaja, se pierde un poco de conocimiento de bajo nivel, pero en la modalidad proyecto es sumamente válido el aporte.

Además, Django provee de su propio servidor de aplicaciones, no recomendado para ambientes productivos, pero muy válido para el contexto de la prueba de concepto.

Las características y segregación de las capas anteriormente son:

- **Capa de datos**: Encapsulada por el ORM de Django, pero en una base de datos MySql instalada y administrada de manera independiente.
- **Capa de lógica de Negocio**: Es la capa encargada de recibir los datos de los mails, preprocesarlos y de analizar los textos de los correos mediante mecanismos de procesamiento de lenguaje natural de ML, con los modelos previamente entrenados y probados. Aunque independiente del funcionamiento de la aplicación web, también está

en esta capa el corazón de la solución, aquí está el código de entrenamiento, testing y deploy de los modelos de Machine Learning.

- **Capa de lógica de Interfaz e Interfaz de Usuario**: donde está la lógica de despliegue y de recepción de peticiones para la aplicación. En esta capa es donde se tratan los datos recibidos, pero también se muestran dinámicamente las páginas con los datos resultantes del procesamiento. Lógica inherente al framework DJANGO, pero que también incorpora todos los componentes de HTML, JavaScript y CSS para su correcto funcionamiento.

Cada capa del sistema tiene una responsabilidad en la cadena de pasaje de los datos para su correcto uso, y está diseñada de tal forma de que sea lo más desacoplada posible dentro de las posibilidades.

El módulo específico de entrenamiento, test y despliegue usa como entrada archivos .csv (comma sepparated values), que fue creado a partir del tratamiento de todos los mails que se disponen, tanto los correos que son phishing confirmados, como aquellos que son mails inocuos o HAM. Este tratamiento usa el mismo código para la obtención del contenido principal de cada mail, que el código para obtener la información de todos los mails a ser procesados por el propio sistema. Este reúso mantiene una uniformidad del procesamiento y facilita el mantenimiento del código y detección de errores. Además, cabe aclarar, que, en el proceso de codificación, ya se obtienen los datos necesarios para un eventual uso de una estrategia de análisis de metadatos. Es decir que aplicar una nueva estrategia de análisis, ya sea serializada, o simplemente un cambio de la estrategia, implicaría un costo menor pues es programación que ya está disponible.

Resumiendo, el código que crea los archivos .csv es el mismo que la aplicación usa internamente para procesar los mails ingresados posteriormente durante el funcionamiento.

4.1 Análisis de marcos para diseño arquitectónico

El diseño arquitectónico de software es un proceso crucial en el desarrollo de aplicaciones y sistemas informáticos. Existen diversos marcos y enfoques que pueden ser utilizados para guiar este proceso, a continuación, explicamos los marcos que investigamos:

- Arquitectura Orientada a Servicios (SOA, por sus siglas en inglés): Este enfoque se basa en la creación de servicios independientes que pueden ser utilizados por múltiples aplicaciones. La arquitectura SOA se enfoca en la interoperabilidad, la reutilización y la flexibilidad, lo que permite una mayor escalabilidad y una reducción en los costos de desarrollo.

- Modelo-Vista-Controlador (MVC): Este marco se enfoca en separar la lógica de presentación de la lógica de negocio de una aplicación. El modelo representa los datos y la lógica de negocio, la vista se encarga de la presentación y la interfaz de usuario, y el controlador actúa como intermediario entre el modelo y la vista.

- Arquitectura en Capas: Este enfoque se basa en la separación de una aplicación en diferentes capas, cada una con una función específica. Por ejemplo, una capa de presentación, una capa de negocio y una capa de datos. Esto permite una mayor modularidad y flexibilidad en la aplicación.

- Arquitectura Basada en Eventos: Este marco se enfoca en la comunicación entre los diferentes componentes de una aplicación a través de eventos. Cuando un evento ocurre en un componente, se envía un mensaje a los demás componentes interesados en ese evento. Esto permite una mayor escalabilidad y flexibilidad en la aplicación.

- Arquitectura de Microservicios: Este enfoque se basa en la creación de servicios independientes y altamente especializados que trabajan juntos para formar una aplicación completa. Cada servicio se ejecuta en su propio proceso y se comunica con los demás servicios a través de APIs bien definidas. Esto permite una mayor escalabilidad y flexibilidad en la aplicación, así como una mayor capacidad de manejar cambios en los requisitos de negocio.

- 4+1: Se enfoca en la creación de diferentes vistas de una arquitectura, cada una con un propósito diferente (vista lógica, vista de proceso, vista de implementación, vista de datos y vista escenarios). El objetivo es proporcionar una comprensión completa y detallada de la arquitectura, permitiendo que diferentes partes interesadas comprendan y visualicen diferentes aspectos de la misma.

- C4: Esta estrategia se enfoca en la creación de un modelo visual de la arquitectura, utilizando diferentes diagramas que muestran diferentes niveles de detalle y abstracción. El objetivo principal es proporcionar una manera clara y efectiva de visualizar y comunicar la arquitectura de una aplicación.

Existen diversos marcos y enfoques que pueden ser utilizados para guiar el diseño arquitectónico de software, cada uno con sus propias ventajas y desventajas. Es importante seleccionar el que resulte más adecuado para nuestro proyecto, teniendo en cuenta factores como, la flexibilidad y modularidad requeridas, así como los requisitos de negocio y las limitaciones técnicas y de recursos.

Luego de una evaluación, seleccionamos el marco C4 (Contexto, Contenedor, Componente, Código) debido a que utiliza una interfaz visual, descriptiva y de alto nivel de

documentar la arquitectura de software, además proporciona un lenguaje y una estructura común para que el equipo discuta y documente la arquitectura del sistema de software.

El modelo C4 es flexible y puede adaptarse a distintos niveles de detalle, en función de las necesidades del proyecto. El objetivo principal del modelo C4 es proporcionar una forma clara y concisa de documentar y comunicar la arquitectura de software que sea fácil de entender y pueda ser utilizada eficazmente tanto por los interesados técnicos como por los no técnicos.

El modelo C4 proporciona una visión más clara y concisa de la arquitectura del software que el modelo 4+1. El modelo C4 se utiliza para documentar y comunicar la arquitectura de un sistema de software, mientras que el patrón MVC se utiliza para separar los componentes de una aplicación y promover la separación de preocupaciones. El modelo MVC se utiliza en arquitecturas más complejas.

4.2 Descripción de la Arquitectura

La arquitectura de software es el conjunto de decisiones y principios que guían el diseño y la construcción de un sistema de software. Se enfoca en definir la estructura y la organización de los componentes del sistema, así como en establecer las interacciones entre ellos.

La arquitectura de software es fundamental para garantizar que el sistema cumpla con los requisitos funcionales y no funcionales, tales como la escalabilidad, la seguridad, la eficiencia, la mantenibilidad y la flexibilidad. Además, una buena arquitectura de software permite a los desarrolladores trabajar de forma más eficiente, reducir costos y aumentar la calidad del producto final.

En esta descripción de la arquitectura de software, exploraremos los diferentes aspectos que intervienen en la definición de una arquitectura sólida y eficiente, incluyendo la elección de patrones de diseño, la definición de la estructura de capas y componentes, la selección de tecnologías y herramientas, entre otros.

Utilizando el modelo C4 desarrollamos los 4 niveles antes descritos donde se definió la interconexión entre los distintos artefactos, así como también la tecnología que se utilizará para cada uno de ellos.

Esta metodología propone un enfoque de cuatro niveles para describir la arquitectura de un sistema, que se conocen como los niveles de C4:

Nivel 1: Contexto

El nivel 1 se enfoca en definir el contexto del sistema, es decir, los actores externos y los sistemas que interactúan con él. En este nivel se suele utilizar un diagrama de contexto que muestra el sistema en el centro y los actores externos y sistemas que se relacionan con él en su entorno.

Nivel 2: Contenedor

El nivel 2 se enfoca en la definición de los contenedores del sistema, es decir, los componentes de alto nivel que conforman el sistema y que se ejecutan en un entorno separado. Los contenedores suelen ser aplicaciones o servicios, y se definen las interfaces y las interacciones entre ellos. En este nivel se suele utilizar un diagrama de contenedores que muestra los contenedores del sistema y sus relaciones.

Nivel 3: Componente El nivel 3 se enfoca en la definición de los componentes del sistema, es decir, los elementos de bajo nivel que conforman los contenedores. Los componentes suelen

ser módulos o librerías que se encargan de realizar una tarea específica. En este nivel se suele utilizar un diagrama de componentes que muestra los componentes del sistema y sus interacciones.

Nivel 4: Código El nivel 4 se enfoca en la implementación detallada de los componentes y en la relación con el código fuente. En este nivel se pueden utilizar herramientas de diagramación de código para mostrar detalles técnicos de la implementación, como clases, métodos y atributos.

Los niveles de C4 proporcionan una manera estructurada de describir la arquitectura de software, desde una vista general del contexto hasta detalles de implementación a nivel de código. Esto permite a los desarrolladores y arquitectos tener una comprensión clara y precisa del sistema, lo que facilita la toma de decisiones y el mantenimiento del sistema a largo plazo

A continuación, se presentan los diagramas de los primeros tres niveles para nuestro proyecto:

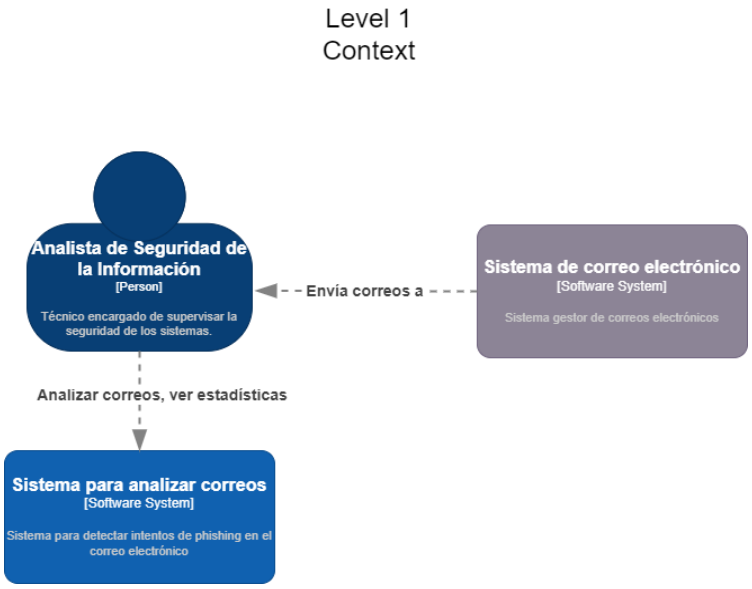


Ilustración 4.2-1

Primero definimos quienes interactuarán con nuestra prueba de concepto. El analista de seguridad de la información es el encargado de procesar los correos para determinar si son intentos de phishing o no. El analista además interactúa con el sistema de correo electrónico. Como puede verse en la Ilustración 4.2-1

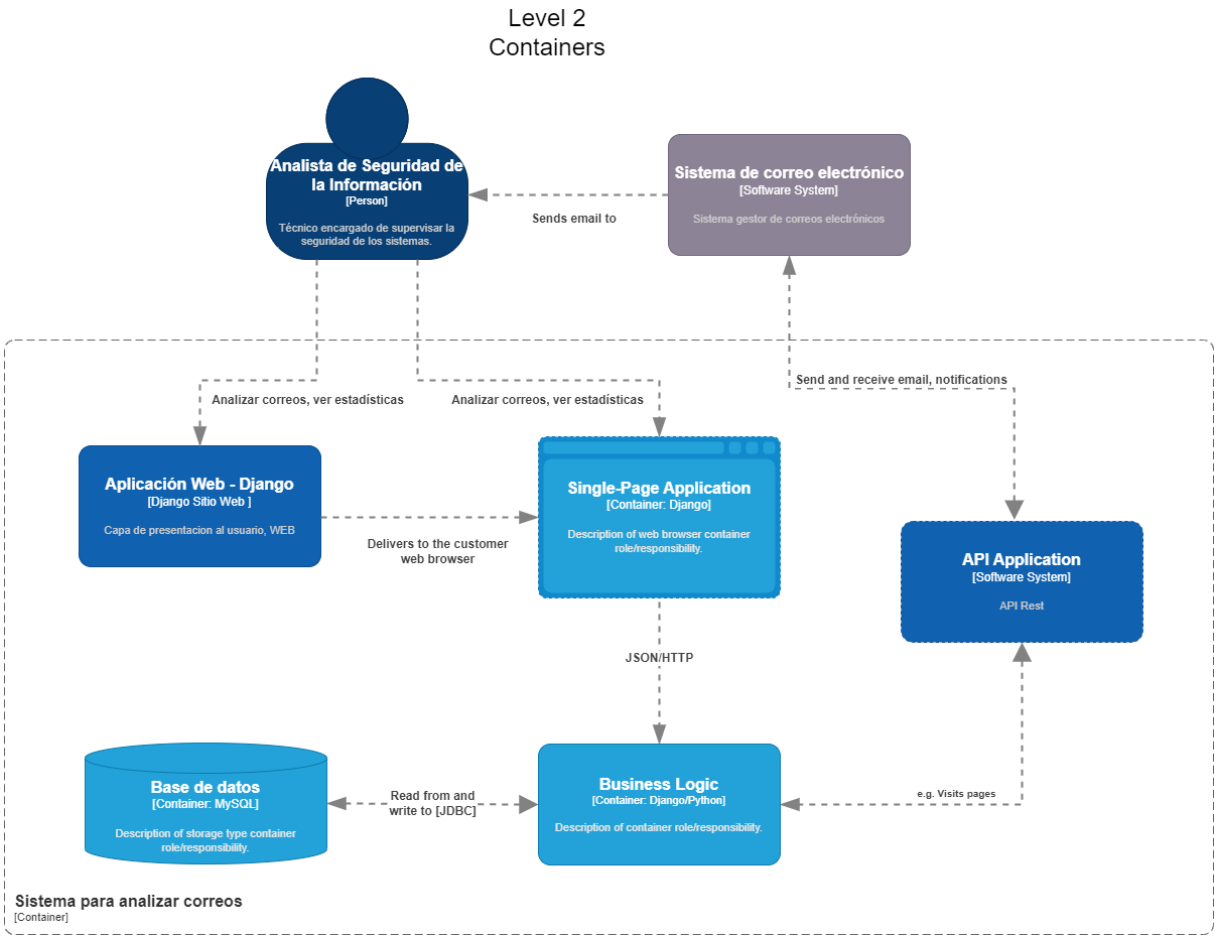


Ilustración 4.2-2

Luego ya se puede observar las instancias de la aplicación, los componentes de lógica de negocio, API y base de datos en una visión muy general. Ver Ilustración 4.2-2

En cambio, en la Ilustración 4.2-3, se puede ver un detalle más preciso de las responsabilidades, y las relaciones entre los componentes de la arquitectura.

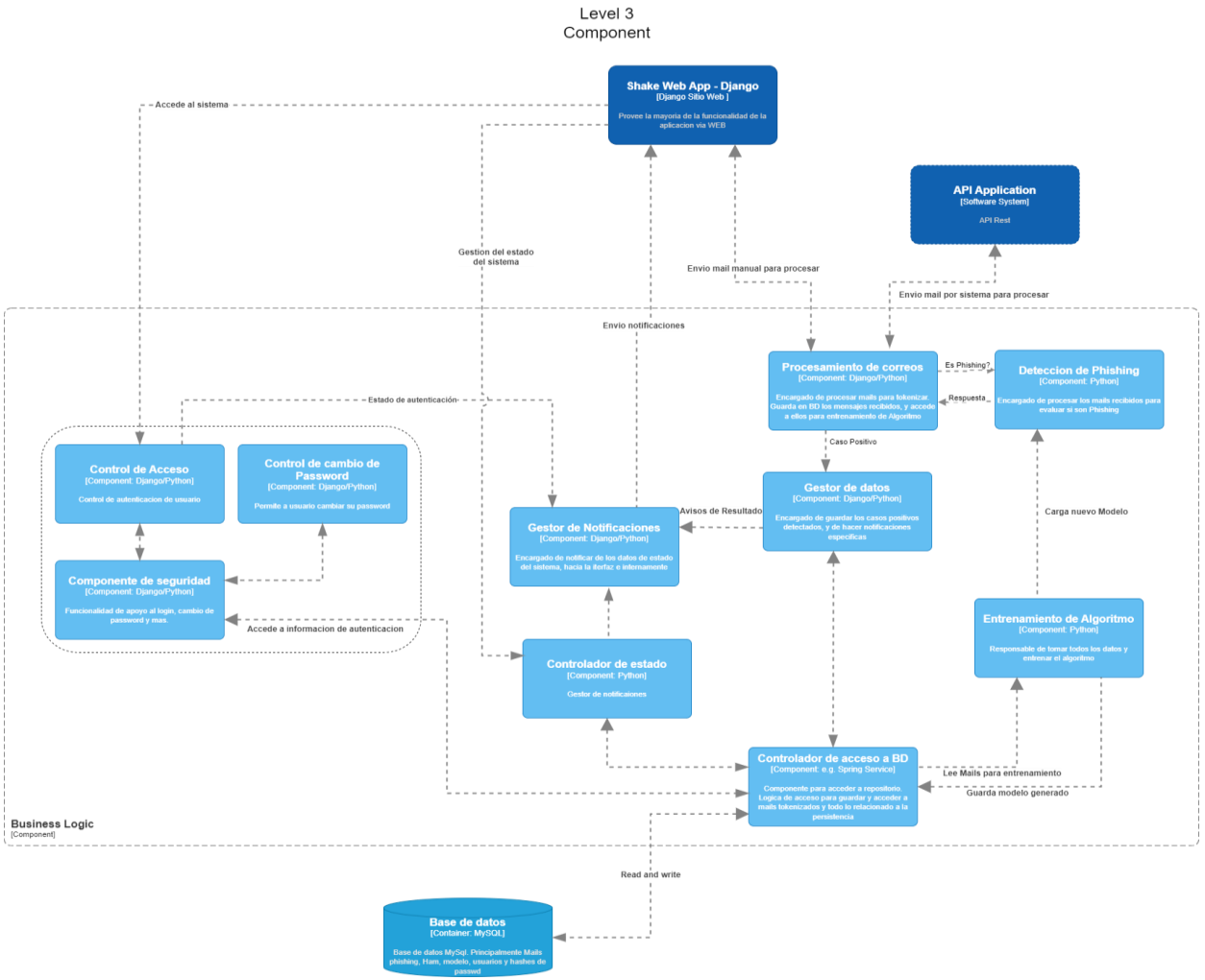


Ilustración 4.2-3

4.2.1 Validación

A efectos de lograr una validación del diseño propuesto, fue preparada una documentación específicamente para el equipo de Arquitectura del banco. En la documentación se incluyó información de contexto para una lectura inicial, con el objetivo de tener un primer acercamiento. Esta tarea de preparación previa se lleva a cabo con la finalidad de complementar un intercambio extraoficial, con la intención de formalizarlo en una reunión coordinada.

Esa reunión es en la que se hace una presentación de la idea, reforzando el contenido de lo ya aportado. Se presenta el avance del proyecto, los diagramas de arquitectura y se solicita su aval para continuar con la prueba de concepto, para lo que obtenemos una devolución positiva, con comentarios al respecto:

De la reunión surge:

- Modificar/agregar en el sistema de notificaciones, una comunicación vía mail hacia los analistas, usando los servidores de mail del banco, cuando haya casos positivos detectados ya sea en los usuarios finales como en el centro de contacto.

También se les solicitó una devolución mediante un formulario, con resultados positivos, y con un comentario en particular, que podría esperarse dentro del contexto de una prueba de concepto:

"*.... ¿Se ajustan a los lineamientos que el Banco tiene en cuanto a los estándares?*

Acá el tema, es que no hablamos de una puesta en producción en el Banco, sino que lo manejamos más como POC, dado que si esto fuera puesto en producción habría que analizar

varios componentes que hoy como está planteado no cumple con los estándares, por ejemplo las bases de datos aceptadas son SQL Server y Oracle, el desarrollo debe poder correr sobre un servidor de aplicaciones JBoss, IIS o JWS.

Luego, habría que analizar la arquitectura de redundancia y balanceo que es capaz de soportar la solución…"

4.3 Conclusiones

Queda definida la base sobre la que trabajaremos, y la forma en la que los diferentes componentes interactuaran, además, que en esta etapa se termina de definir la tecnología que se va a usar, que determina por otra parte la selección de entorno WEB como forma de utilizar los recursos construidos.

En otras palabras, las condiciones son adecuadas para la prueba de concepto y logramos una validación de la arquitectura y diseño, siendo necesaria un análisis en mayor profundidad si esta idea prospera y se pasa a niveles más avanzados. Esto nos asegura la viabilidad técnica, lo que brinda confianza al equipo de desarrollo y los stakeholders.

5. Tecnología

Hay varios criterios que se pueden considerar al decidir qué tecnologías utilizar en un proyecto de desarrollo de software:

Requisitos del proyecto: El conjunto de requisitos del proyecto influyen en las tecnologías que se deben utilizar. Por ejemplo, si el proyecto requiere una gran cantidad de procesamiento de datos, es posible que se necesite una tecnología que pueda manejar grandes volúmenes de

datos. Si el proyecto se enfoca en la creación de una interfaz de usuario, se puede necesitar una tecnología que facilite la creación de interfaces de usuario atractivas y fáciles de usar.

- Escalabilidad: Es importante considerar si la tecnología se puede escalar fácilmente a medida que el proyecto crece. Esto es especialmente importante si se espera que el proyecto crezca rápidamente en un futuro cercano.

- Disponibilidad de recursos: Es importante considerar la disponibilidad de recursos como documentación, herramientas, bibliotecas y frameworks para la tecnología seleccionada. Si hay pocos recursos disponibles, puede ser más difícil y costoso desarrollar el proyecto.

- Experiencia del equipo: La experiencia del equipo de desarrollo es un factor importante para considerar. Si el equipo tiene experiencia en una tecnología específica, es más probable que pueda desarrollar el proyecto con éxito y en menos tiempo.

- Tiempo de desarrollo: El tiempo disponible para desarrollar el proyecto también puede influir en la elección de la tecnología. Si se dispone de poco tiempo, es posible que se deba seleccionar una tecnología que permita desarrollar el proyecto rápidamente.

- Costos: Los costos son un factor importante para considerar en cualquier proyecto de desarrollo de software. Es importante evaluar el costo de la tecnología seleccionada, incluyendo licencias, hardware, recursos humanos, entre otros.

- Mantenibilidad: También es importante considerar la facilidad de mantener la tecnología seleccionada una vez que se haya desarrollado el proyecto. Si la tecnología

se vuelve obsoleta o difícil de mantener, puede aumentar los costos de mantenimiento a largo plazo.

La elección de la tecnología adecuada dependerá de una variedad de factores, incluyendo los requisitos del proyecto, la escalabilidad, la disponibilidad de recursos, la experiencia del equipo, el tiempo disponible para desarrollar el proyecto, los costos y la mantenibilidad a largo plazo.

En nuestro proyecto los factores que determinan las tecnologías que utilizamos son los requerimientos funcionales y no funcionales, que determinan el diseño arquitectónico.

Asimismo, el tiempo de desarrollo es otro factor que tuvimos en cuenta a la hora de seleccionar la tecnología. La experiencia del equipo es otro factor que tuvo peso en la decisión.

El centro del proyecto tiene como base el análisis de información usando Machine Learning, tecnología moderna y que brinda soluciones en muchos campos y aplicaciones. La estrategia para usar Machine Learning es entrenar un algoritmo, usando grandes cantidades de datos. Este proceso se repite usando varios algoritmos, analizando los resultados de cada uno de esos procesos, brindando la oportunidad de seleccionar aquel que se ajusta mejor considerando el contexto como la cantidad de datos o la especificidad de esos datos.

Para el desarrollo de los Modelos de Machine Learning utilizamos Jupyter Notebook que es una herramienta que utiliza Python como lenguaje de programación y tiene la característica que permite escribir código en bloques para ejecutarlos y ver su resultado. De esta forma es sencillo realizar experimentos.

Para el desarrollo de la aplicación web utilizamos Django que es un framework web de alto nivel para crear aplicaciones web en Python. Proporciona un conjunto de herramientas y funciones que facilitan la creación rápida y eficaz de sitios web complejos basados en bases de datos.

Algunas de las características clave de Django incluyen:

- Mapeador objeto-relacional (ORM): Django proporciona un potente ORM que permite interactuar con bases de datos utilizando objetos Python en lugar de escribir consultas SQL directamente.

- Enrutamiento de URL: El sistema de enrutamiento de URL de Django asigna las solicitudes entrantes a la función de vista adecuada, lo que facilita la organización de la lógica y la estructura de su aplicación.

- Sistema de plantillas: Proporciona un motor de plantillas integrado que permite a los desarrolladores definir la estructura y el diseño de sus páginas web utilizando plantillas HTML sencillas.

- Manejo de formularios: Facilita el manejo de datos de formularios, incluyendo la validación y los mensajes de error.

- Interfaz de administración: Proporciona una interfaz de administración incorporada que se puede personalizar fácilmente para proporcionar un potente backend para la gestión de los datos de su aplicación.

- Características de seguridad: Incluye una serie de características de seguridad integradas, incluyendo protección contra ataques de cross-site scripting (XSS), cross-site request forgery (CSRF) e inyección SQL.

Como entorno de desarrollo utilizamos Visual Studio Code de Microsoft, algunas de las razones por las que optamos utilizar Visual Studio Code (VS Code) sobre otros editores de código fueron:

- Flexibilidad y extensibilidad: VS Code es altamente personalizable y tiene una gran cantidad de extensiones disponibles en su marketplace. Esto nos permitió adaptar el entorno de trabajo a nuestras necesidades.

- Integración con Git y otras herramientas: VS Code se integra fácilmente con Git, lo que hace que el control de versiones sea más sencillo. También se puede integrar con otras herramientas comunes de desarrollo, como depuradores y sistemas de construcción.

- Interfaz de usuario intuitiva: La interfaz de usuario de VS Code es clara y fácil de usar. Las características principales, como el editor de texto, el panel de depuración y el explorador de archivos, son accesibles de manera clara y sencilla.

- Multiplataforma: VS Code es compatible con Windows, Mac y Linux, lo que lo hace una buena opción para trabajar con diferentes sistemas operativos.

- Velocidad y rendimiento: VS Code es rápido y ligero, lo que significa que se puede abrir y cerrar rápidamente y ejecuta tareas en segundo plano sin interrupciones notables.

6.Construcción

La etapa de construcción de software es una fase crucial en el ciclo de vida del desarrollo de software. Durante esta etapa, se construye el software a partir del código fuente y se crean los ejecutables y otros componentes necesarios para ejecutar la aplicación en el entorno deseado.

La construcción de software implica la integración de múltiples componentes, incluidas bibliotecas, módulos y dependencias, y la creación de un producto final que se ajuste a los requisitos del usuario. Es un proceso complejo que requiere atención a detalles y un enfoque disciplinado.

La construcción de software también incluye la compilación, pruebas de unidad, pruebas de integración y otras tareas que son esenciales para garantizar que el software construido sea de alta calidad y esté libre de errores y vulnerabilidades.

En esta etapa, se implementan cambios y mejoras en el código fuente y pueden iterar el proceso de construcción varias veces hasta que se obtenga el producto final deseado.

La etapa de construcción de software es fundamental para garantizar que el software funcione correctamente y cumpla con los requisitos del usuario. Es una tarea compleja y disciplinada que requiere un enfoque riguroso para garantizar la calidad y la eficiencia del producto final.

Una vez concluida la etapa del diseño arquitectónico, comenzamos con la etapa de construcción y pruebas.

Para esta etapa detectamos el riesgo “Baja velocidad en la etapa de diseño, construcción y pruebas por ser un equipo reducido”, lo tratamos haciendo seguimiento constante de la planificación de cada sprint.

6.1 Ciclo de vida

En este proyecto se nos presentó el desafío de desarrollar varios modelos de Machine Learning utilizando el ciclo de vida incremental iterativo, que fue el que seleccionamos para construir la prueba de concepto. Existe incompatibilidad entre el proceso de desarrollo utilizando un ciclo de vida de Machine Learning y un ciclo de vida incremental iterativo. Por lo que nos encontramos con la necesidad de utilizar un único proceso que integre ambos ciclos de vida.

En un comienzo no encontramos bibliografía al respecto. Por lo que comenzamos la etapa de desarrollo trabajando en paralelo con el ciclo de vida incremental iterativo para el desarrollo de la aplicación web y por otro lado el desarrollo de modelos de Machine Learning aplicando el ciclo de vida propio para este proceso.

Sabiendo de esta particularidad, y en el proceso de continuar investigando, encontramos, como era de esperarse, que es una problemática que ya se ha dado, ha sido estudiado y que el ciclo de vida resultante ha sido propuesto Ranawana [12].

Según R.Ranawana y A Karunanadda la complejidad de crear e integrar aplicaciones de aprendizaje automático es un desafío para los equipos de desarrollo. Las diferencias inherentes entre el desarrollo tradicional y el aprendizaje automático no permiten aplicar uniformemente las metodologías de ingeniería de software.

El ciclo de vida de Machine Learning es un enfoque que enfoque implica la recopilación y procesamiento de datos, la selección de algoritmos, la creación y evaluación de modelos. El objetivo es interpretar datos y aprender de ellos, para luego poder realizar "predicciones" de situaciones futuras.

El ciclo de vida incremental iterativo es un enfoque para el desarrollo de software en el que se divide el proyecto en pequeñas iteraciones. Cada iteración tiene sus propios objetivos, entregables y plazos. El proceso de desarrollo se divide en fases y cada fase produce un conjunto de entregables.

Según Roger S. Pressman [13] el modelo incremental combina elementos de los flujos de proceso lineal y paralelo, aplica secuencias lineales en forma escalonada a medida que avanza el calendario de actividades. Cada secuencia lineal produce "incrementos" de software que se van entregando al cliente, [McD93] de manera parecida a los incrementos producidos en un flujo de proceso evolutivo.

El desarrollo de software incremental, que es una parte fundamental de los enfoques ágiles, es mejor que un enfoque en cascada para la mayoría de los sistemas empresariales, de comercio electrónico y personales [14]

El desarrollo incremental se basa en diseñar e implementar y entregar incrementos o piezas de software al cliente con determinadas funcionalidades. Por lo general en los primeros

incrementos se entregan las funcionalidades más importantes o que son más urgentes para el cliente. Cada nuevo incremento incorpora nuevas funcionalidades.

Este enfoque tiene la ventaja de que la validación con el cliente se realiza de forma temprana, por lo que permite corregir desviaciones, evitando costos innecesarios. Cada incremento o versión del sistema incorpora algunas de las funciones que necesita el cliente.

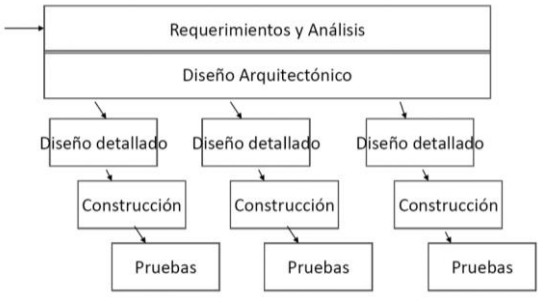


Ilustración 6.1-1. Desarrollo incremental [13] R.S.Pressmann

Según W.J. Murdoch [15] tomado en toda su generalidad, interpretar datos significa extraer información (de alguna forma) de ellos. El conjunto de métodos que se encuentran bajo este paraguas abarca todo, desde el diseño de un experimento inicial hasta la visualización de los resultados finales. En esta forma demasiado general, la interpretabilidad no es sustancialmente diferente de los conceptos establecidos de ciencia de datos y estadística aplicada.

Dado que en gran parte del ciclo de vida se experimenta con diferentes conjuntos de datos, características y algoritmos, el proceso puede ser complejo. Además, no hay garantía de que se pueda crear un modelo que funcione al final del proceso. Factores como la disponibilidad y calidad de los datos, las técnicas de ingeniería de características (el proceso de utilizar para extraer características útiles de los datos brutos) y la capacidad de los algoritmos de aprendizaje, entre otros, pueden impedir que el resultado sea satisfactorio.

David Ping [16] propone el siguiente ciclo de vida:

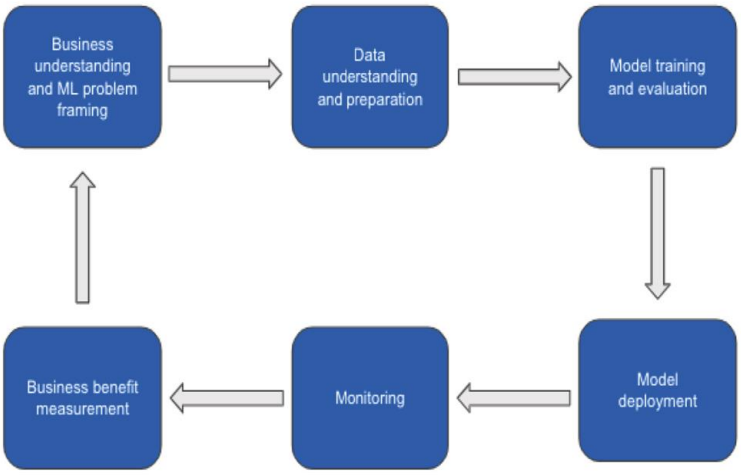


Ilustración 6.1-2. Ciclo de vida de Machine Learning [16]

6.1.1 Elementos en común y diferencias entre ambos ciclos de vida

Tanto el ciclo de vida incremental iterativo como el ciclo de vida de Machine Learning comparten la característica de ser iterativos y adaptativos, lo que significa que permiten ajustar el proceso de desarrollo a medida que se van adquiriendo nuevos conocimientos y se van identificando nuevas necesidades.

Por otro lado, también existen diferencias significativas entre ambos modelos. Mientras que el modelo incremental iterativo se basa en el diseño, desarrollo y prueba de software, el desarrollo de modelos de aprendizaje automático se basa en el diseño, capacitación, evaluación, implementación y monitoreo de datos y modelos. [12]

Los sistemas de aprendizaje automático no son deterministas y, por lo tanto, son difíciles de construir utilizando métodos de desarrollo secuencial [17]. Amershi [18], también reconoce la necesidad de una diversidad de habilidades para llevar un modelo de aprendizaje automático a la producción como uno de los principales obstáculos en el desarrollo de aplicaciones de aprendizaje automático.

Ranawana [12] propone un marco unificado, como se puede observar en la figura 3, llamado Machine Learning Aplication Software Development Life Cycle (MLASDLC) que facilita la planificación, el desarrollo y la implementación de una aplicación de aprendizaje automático a través de procesos paralelos para software e ingeniería de aprendizaje automático.

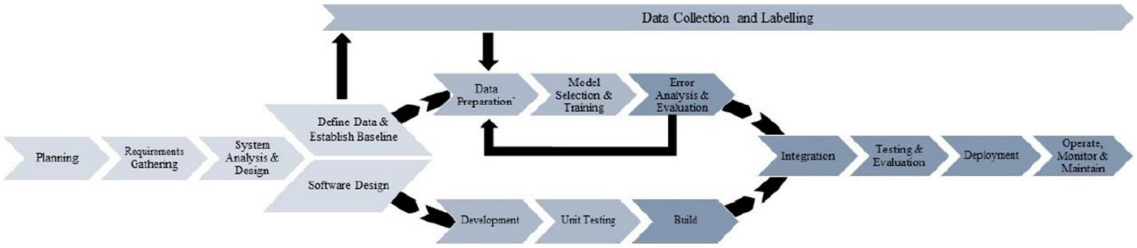


Ilustración 6.1-3 – Modelo MLASDLC [12]

Las primeras tres etapas de este marco: Planificación, recopilación de requerimientos y análisis y diseño, son comunes a ambos ciclos de vida.

A partir de la siguiente etapa el proceso se divide en dos ramas que se desarrollan en paralelo:

En la rama incremental iterativa se llevan adelante las etapas del desarrollo ágil: diseño, desarrollo, pruebas unitarias y construcción. Los procesos se repiten para cada sprint. Estas fases se utilizarán para crear y probar la aplicación de usuario y el sistema de configuración y entrega, las canalizaciones MLOps y DevOps.

En la otra rama se desarrollan los métodos estándar utilizados en la producción de Machine Learning: preparación de datos, selección y entrenamiento de modelos y evaluación.

Las fases de preparación de datos y selección del modelo y entrenamiento se dividen en varias subfases. La recolección de datos y el etiquetado se realiza en un proceso paralelo separado de forma de permitir que el desarrollo del modelo se ejecute sin interrupciones.

Luego de finalizadas las etapas de cada una de las ramas, el software de aplicación y los módulos de aprendizaje automático se integran en un único sistema. A partir de esta etapa se realizan las pruebas y evaluación, despliegue y finalmente operación, monitoreo y mantenimiento del sistema.

6.2 Modelo de Machine Learning

Para construir nuestro modelo de Machine Learning utilizamos etapas descritas por David Ping [16]. Primero comenzamos por comprender el problema de negocio al cual nos enfrentamos, luego comprender los datos asociados a este problema y prepararlos, luego continuamos con la etapa de entrenamiento y evaluación de nuestro modelo, luego se realiza el Deploy (también pude llamarse despliegue) del modelo.

En nuestro caso, tuvimos que adaptar el modelo base, para ajustarlo a nuestra necesidad a la hora de hacer el despliegue, para luego considerar el uso del modelo original para su uso una vez implementado, como puede verse en la Ilustración 6.2-1

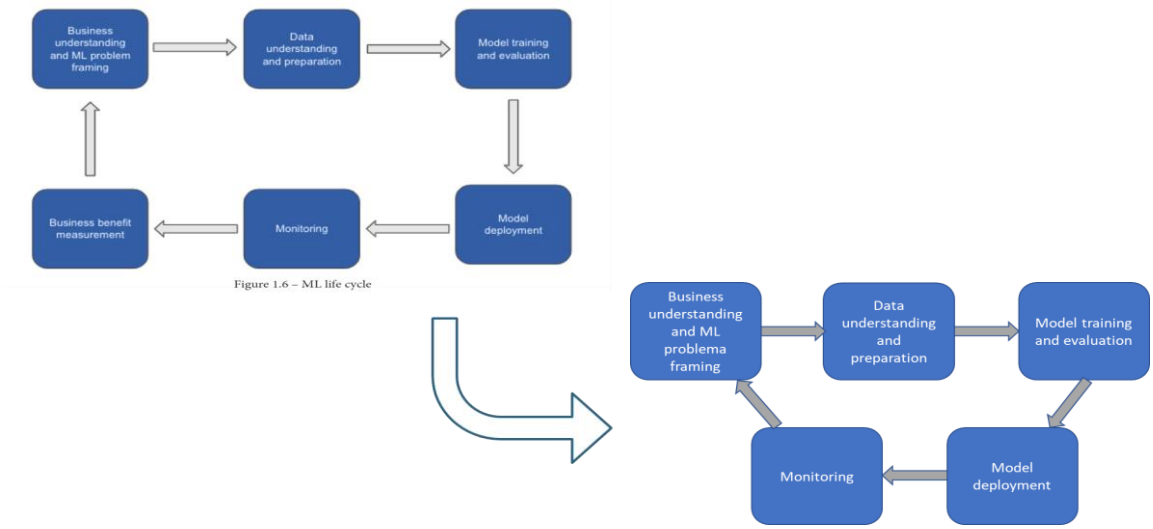


Ilustración 6.2-1. etapas utilizadas del ciclo de vida propuesto por David Ping

6.2.1 Comprendiendo el problema

Para crear un modelo de Machine Learning, lo primero que hicimos es entender el problema que estamos tratando de resolver: determinar si los correos electrónicos entrantes son un intento de phishing o son correos fidedignos. Al analizar los distintos correos con intentos de phishing encontramos que existen distintos formatos, distintas redacciones en el cuerpo de los correos, algunos contienen el logotipo de nuestro Banco, otros no, la mayoría utiliza enlaces (link) para redirigir al cliente a un sitio web falso. Todos o casi todos tienen un mensaje de urgencia donde se expresa que, si no se hace determinada acción inmediatamente, se bloqueará su cuenta bancaria, por ejemplo. Muchos de estos correos electrónicos contienen faltas de ortografía, o contienen algunas palabras en inglés. Todos simulan ser emitidos por la institución bancaria. Debido a la diversidad de textos que incluyen este tipo de mensajes es que decidimos utilizar técnicas de inteligencia artificial para abordar este problema.

En la Ilustración 6.2-2 se puede ver un ejemplo de un correo con intento de phishing.

Analizamos el body y encontramos varios elementos que nos indican que se podría tratar de un intento de phishing: Comienza el correo con una frase que tiene un sentido de urgencia: "***Su cuenta y llave digital deben ser verificados o serán bloqueados.”*** buscando provocar en la persona que lo recibe una sensación de que si no hago lo que me piden se bloqueará mi cuenta. El correo contiene varias faltas de ortografía. Termina con una firma que dice ***“Sincerely, 2022 ?.All rights reserved.”***

Asunto: Verifique sus Productos Banco Republica

De: "Seguridad Banco Republica (BROU)"

Para: xxx@xxxl.com

CC:

Su cuenta y llave digital deben ser verificados o seran bloqueados.

[-EMAIL-],

El Equipo de Seguridad Banco Republica (BROU) indica que su cuenta ha tenido una actividad sospechosa, por lo que procederemos a bloquearla si no la verifica.

El Banco Republica (BROU) le solicita que verifique su cuenta y llave digital correctamente, de lo contrario su cuenta sera bloqueada y debera dirigirse a una de nuestras sucursales para solventar el problema.

A partir del envio de esta EMAIL, usted tiene de 24 a 72 horas para validar su cuenta, de lo contrario sera bloqueada.

VERIFIQUE INMEDIATAMENTE

Sincerely,

2022 ?.All rights reserved.

Ilustración 6.2-2

6.2.2 Comprensión y preparación de datos

Una vez que comprendimos el problema, recopilamos los correos electrónicos que llegaron al Departamento de Seguridad de la Información. Tanto los reportados por los funcionarios como los recibidos por el centro de contacto que es quien atiende a los clientes.

Una vez que tuvimos los correos electrónicos, comenzamos a elaborar una estrategia para determinar de qué forma podríamos obtener la mayor cantidad de información posible para entrenar a nuestros algoritmos de Machine Learning.

Analizando los correos observamos que la mayor parte de los correos contienen gran cantidad de texto, también observamos la forma en cómo están redactados, la información que solicitan, el sentido de urgencia, etc. Todos estos son indicadores que determinan si un correo electrónico es un intento de phishing o no.

Como estrategia decidimos realizar un análisis del body del correo, ya que allí es donde se encuentra la información relevante para determinar si un correo es phishing o no.

6.2.2.1 Pre - Procesamiento de los correos

Como se menciona en el capítulo 4: Arquitectura), el procesamiento de los correos no es parte del sistema desarrollado en cuanto a que sus funcionalidades no son accedidas desde la interfaz de usuario. La motivación de esa decisión está basada en que las tareas de entrenamiento, testing y deploy relativas a Machine Learning, como ya hemos mencionado, deben realizarse offline, para luego si, en un momento acordado y apropiado, colocar los modelos resultantes en la estructura de directorios adecuada (por ejemplo C:\ShakeProject\shake_API\mail_classifier\models) donde van a ser accedidos por el sistema ya en su normal funcionamiento.

También hemos mencionado, que había otra opción que contemplamos inicialmente, que era el análisis de los Metadatos de cada correo, pero nuestra estrategia para analizar iba a enfocarse en procesamiento de lenguaje natural o NLP. Lo cierto es que en ambos casos existía la necesidad de acceder a la información contenida dentro de cada correo, y en NLP específicamente el cuerpo o body del mismo es donde está el contenido realmente importante para este análisis.

Se definió una estructura de directorios para proceder al depósito y posterior procesamiento de los mails, para cada uno de los tipos de ellos, tanto Phishing como HAM como lo muestra la Ilustración 6.2.2.1-1 (Estructura de directorios)

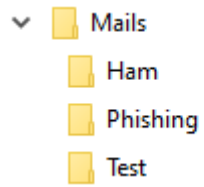


Ilustración 6.2.2.1-1, estructura de directorios

También se destinó un espacio para poder colocar los archivos utilizados como test, y que su posterior procesamiento no le incluiría ninguna etiqueta.

Para todos esos casos, la estrategia de apertura de los correos y procesamiento fue similar, se ejecutaron los siguientes pasos:

1. Se procesa un mail main_process_auto,py()
2. Se valida el mail mailvalidation(mail_seleccionado)
    a. Se verifica existencia
    b. Se verifica Tamaño
    c. Se verifica extensión
3. Se procesa procesomailtipo_msg(mail_seleccionado)
    a. Extracción de email emailextraido = extract_msg.Message(mail)
    b. Tageo de mail lista_mail_sospechoso=colocomarcador(emailextraido)
    c. Se crea un Objeto Mail creomailoriginaltokenizado(lista_mail_sospechoso)

Este procedimiento de alto nivel toma cada uno de los mails, y luego de obtener la Metadata (Remitente, Asunto, fecha, etc.), comienza a procesar el cuerpo del mensaje y lo recorre hasta encontrar el “Body” original. Puede que ese correo no sea la primera vez que haya sido reenviado, por lo que hay un desafío en encontrar el punto exacto desde el que hacer esta extracción.

Una vez encontrado el cuerpo original, el mismo se pone en una lista, que es devuelta por la función correspondiente y se guarda como uno de esos atributos del Objeto Mail antes mencionado. Esta lista, posteriormente será parseada en un archivo .csv, al igual que los demás mails.

Cabe destacar que, en el proceso de preprocesamiento, se quitan muchos caracteres innecesarios para la tarea de procesamiento mediante NLP, se formatean las fechas y direcciones de mail de los remitentes, y se obtienen los links existentes, a su vez que se captura toda la información que es relevante durante el proceso. Esto es importante, pues si bien sólo nos interesa el body, la programación relativa a la obtención de la Metadata ya está hecha, y en caso de decidir recorrer el camino de análisis de Metadata, solo se trata de consumir los datos que ya se obtienen.

Posterior a todo el procedimiento mencionado, para cada uno de los Objetos Mail (de clase *mailtokenizado*, se procede a añadir una línea en el archivo (según sea el caso, con etiquetas o no) correos.csv o test.csv. El procesamiento de los mails para llegar a tener esos archivos ronda las 1000 líneas de código.

Podemos ver en las siguientes ilustraciones (6.2.2.1-2 y 6.2.2.1-3) donde se muestran extracciones de las partes de cada tipo de mail procesado del archivo correos.csv.

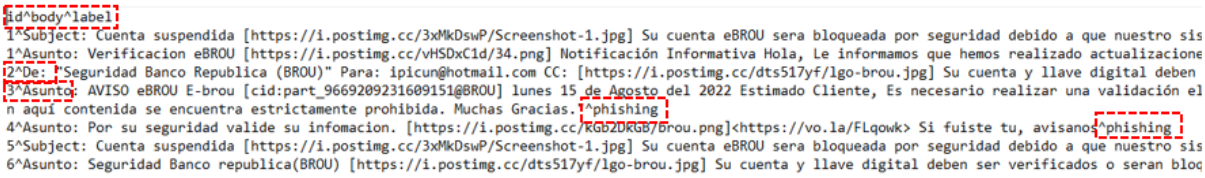


Ilustración 6.2.2.1-2, Extracto de archivo correos.csv, caso Phishing

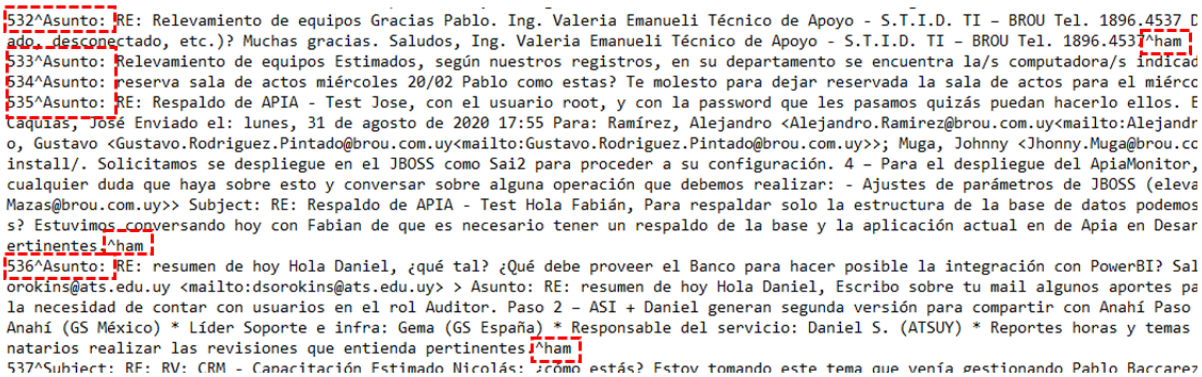


Ilustración 6.2.2.1-3, Extracto de archivo correos.csv, caso HAM

En cambio, podemos observar en la Ilustración 6.2.2.1-4, que las líneas del archivo de Test (test.csv) no contiene las etiquetas al final

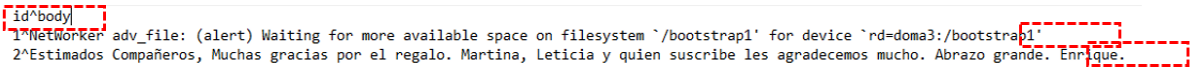


Ilustración 6.2.2.1-4, Extracto de archivo test.csv

Todo este procesamiento previo, o preprocesamiento fue necesario para que los algoritmos de clasificación de Machine Learning tuviesen su ingesta de datos correctamente formateada. En ocasiones aparece en algún mail, en contadas oportunidades algún carácter que nos obliga a tomar acción, o bien en el archivo de forma manual, o en el procesamiento de forma automatizada. Al momento actual, se han tomado las precauciones para que todo fluya hasta el final de manera automática. Una de las razones principales es que solamente estamos admitiendo el formato de archivos *.msg de Outlook como se ve en la Ilustración 6.2.2.1-4, Extracto de código de método mailvalidation, es que Microsoft Outlook es la herramienta standard del Banco, y tuvimos que reducir la amplitud de las extensiones.

```
elif pathlib.Path(mail).suffix != ".msg": #Aca puedo poner mas extensiones
    print("mailvalidation - Extension invalida, tiene que ser 'msg'")
    return False
```

Ilustración 6.2.2.1-5, Extracto de código de método mailvalidation

6.2.3 Model training and evaluation

A partir de lo expresado anteriormente decidimos utilizar procesamiento de lenguaje natural (NLP), por sobre el análisis de la Metadata.

Luego de realizar data cleaning de nuestros correos construimos nuestro primer dataset.

Dataset 1
Cantidad de correos electrónicos: 396
Correos electrónicos con intentos de Phishing:179
Correos electrónicos Ham: 217

Tabla 6.2.3-1

El procesamiento de lenguaje natural lo realizaremos en varias etapas [19], como se puede ver en la figura 6.21: preprocessing, tokenization, stop words removal, reduce to a root form y vectorize.

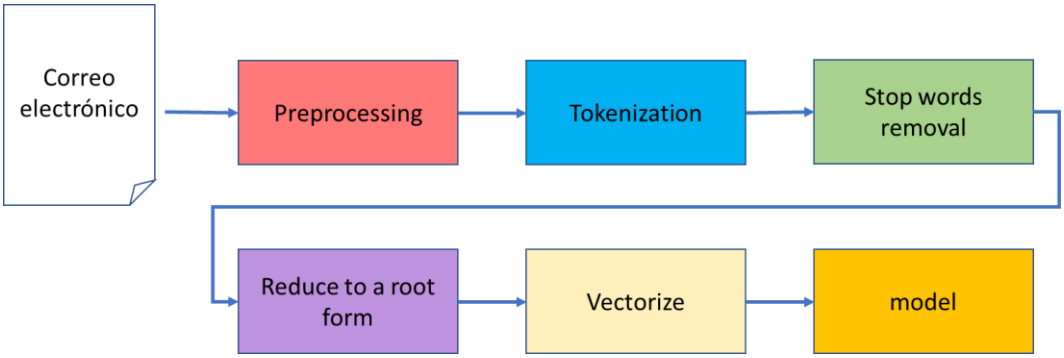


Ilustración 6.2.3-2 [19]

- Preprocesamiento: En esta etapa se eliminan caracteres como por ejemplo espacios o signos de puntuación.
- Tokenization: Se coloca el texto en un vector, donde cada palabra ocupa una posición de ese vector.
- Stop words removal: Se eliminan las palabras “vacías” que son un conjunto de palabras de uso común en cualquier idioma. Por ejemplo, "el", "es" y "y". En NLP se consideran palabras sin importancia, o, mejor dicho, que no aportan información, esto permite centrarse en las palabras “importantes”.
- Reduce to a root form: Modificación de una palabra para expresar diferentes categorías gramaticales, como el tiempo, la voz, el aspecto, la persona, el número, el género y el modo. Una flexión expresa una o varias categorías gramaticales con un prefijo, un sufijo o un infijo, u otra modificación interna, como un cambio vocálico.
- Vectorize: Se convierten las palabras a números para que puedan ser procesadas.

Una vez que los correos fueron procesados por NLP serán clasificados utilizando algoritmos de machine learning.

Para esto utilizamos procesamiento de lenguaje natural y algoritmos de Machine Learning de clasificación, debido a que este problema consiste en clasificar los correos en dos categorías: phishing o ham. Se le llama Ham a los correos fidedignos.

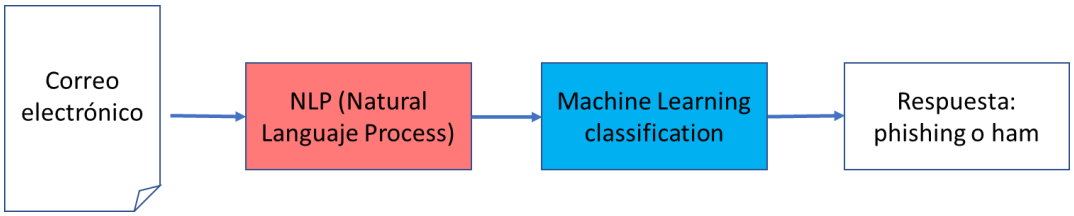


Ilustración 6.2.3-3

Cuando desarrollamos nuestro modelo, realizamos varios experimentos con cuatro algoritmos de clasificación: BaggingClassifier, RandomForestClassifier, GradientBoostingClassifier y MultinomialNB. Probamos realizar ajustes a los hiper parámetros de estos algoritmos, procurando obtener un valor de exactitud mayor al 80%, pero sin llegar al sobreajuste.

Para el entrenamiento de los modelos utilizamos 396 correos entre phishing y ham. El dataset lo dividimos en 80% para train y 20% para validation.

Luego del entrenamiento obtuvimos los siguientes resultados:

Experimento 1		
Algoritmo	**Exactitud**	**Ajuste de hiperparámetros**
BaggingClassifier	0,55	estimator=tree,n_estimators=10,max_samples=6,random_state=42
RandomForestClassifier	0,9375	n_estimators = 5, criterion='entropy', max_depth=2, min_samples_split=10, min_samples_leaf=10,random_state=42
GradientBoostingClassifier	0,9125	n_estimators=5, learning_rate=0.05, max_depth=3, subsample=0.7, random_state=42
MultinomialNB	0,5375	force_alpha=True, alpha=5, fit_prior=False
Experimento 2		
Algoritmo	**Exactitud**	**Ajuste de hiperparámetros**
BaggingClassifier	0,775	estimator=tree,n_estimators=30,max_samples=7,random_state=42
RandomForestClassifier	0,95	n_estimators = 10, criterion='entropy', max_depth=2, min_samples_split=20, min_samples_leaf=10,random_state=42
GradientBoostingClassifier	0,9375	n_estimators=10, learning_rate=0.07, max_depth=3, subsample=0.7, random_state=42
MultinomialNB	0,7375	force_alpha=True, alpha=3, fit_prior=False
Experimento 3		
Algoritmo	**Exactitud**	**Ajuste de hiperparámetros**
BaggingClassifier	0,9625	estimator=tree,n_estimators=65,max_samples=10,random_state=42
RandomForestClassifier	0,9625	n_estimators = 10, criterion='entropy', max_depth=5, min_samples_split=20, min_samples_leaf=10,random_state=42
GradientBoostingClassifier	0,95	n_estimators=12, learning_rate=0.07, max_depth=5, subsample=0.7, random_state=42
MultinomialNB	0,925	force_alpha=True, alpha=2, fit_prior=False
Experimento 4		
Algoritmo	**Exactitud**	**Ajuste de hiperparámetros**
BaggingClassifier	0,9625	estimator=tree,n_estimators=80,max_samples=15,random_state=42
RandomForestClassifier	0,975	n_estimators = 60, criterion='entropy', max_depth=7, min_samples_split=20, min_samples_leaf=10,random_state=42
GradientBoostingClassifier	0,975	n_estimators=700, learning_rate=0.05, max_depth=3, subsample=0.7, random_state=42
MultinomialNB	0,9625	force_alpha=True, alpha=1, fit_prior=True

Ilustración 6.2.3-4

Una vez que realizamos la etapa de train y validation pasamos a la etapa de test. En este caso testeamos los modelos con un DataSet que contiene 144 correos entre phishing y ham. Estos correos no están incluidos dentro del DataSet de train, es decir son otros correos. La idea en este punto es analizar cómo se comporta el modelo con nuevos correos sin etiquetar.

Los resultados obtenidos fueron los siguientes:

Test					
Algoritmo	**Exactitud**	**Ajuste de hiperparámetros**	**Recall**	**Precision**	**F1Score**
BaggingClassifier	0,94	estimator=tree,n_estimators=80,max_samples=15,random_state=42	0,94	0,91	0,94
RandomForestClassifier	0,97	n_estimators = 60, criterion='entropy', max_depth=7, min_samples_split=20, min_samples_leaf=10,random_state=42	0,96	0,99	0,97
GradientBoostingClassifier	0,97	n_estimators=700, learning_rate=0.05, max_depth=3, subsample=0.7, random_state=42	0,97	0,96	0,97
MultinomialNB	0,67	force_alpha=True, alpha=2, fit_prior=True	0,63	0,69	0,66

Ilustración 6.2.3-5

Hasta esta etapa del desarrollo nos concentramos en la métrica accuracy (exactitud) que sin duda es una métrica importante, pero debemos realizar un análisis de otras métricas relevantes como: recall, precision, f1score.

Experimento 3				
Algorítmo	Exactitud	**Recall**	**Precision**	**F1Score**
BaggingClassifier	0,9625	1,00	0,94	0,97
RandomForestClassifier	0,9625	1,00	0,93	0,97
GradientBoostingClassifier	0,95	1,00	0,91	0,95
MultinomialNB	0,925	0,95	0,91	0,93
Experimento 4				
Algorítmo	Exactitud	**Recall**	**Precision**	**F1Score**
BaggingClassifier	0,9625	1,00	0,94	0,97
RandomForestClassifier	0,975	1,00	0,96	0,98
GradientBoostingClassifier	0,975	1,00	0,96	0,98
MultinomialNB	0,9625	0,96	0,98	0,97

Ilustración 6.2.3-6

Las métricas Recall, Precision y F1 Score son comúnmente utilizadas para evaluar el rendimiento de un modelo de clasificación.

Recall (Recuperación) mide la proporción de casos positivos reales que son correctamente identificados por el modelo. Es decir, de todas las instancias que pertenecen a una clase determinada, ¿cuántas de ellas el modelo logró detectar correctamente? La fórmula es:

Recall = TP / (TP + FN)

donde TP (True Positive) representa las instancias positivas que fueron correctamente identificadas, y FN (False Negative) son las instancias positivas que fueron identificadas erróneamente como negativas.

Precision (Precisión) mide la proporción de casos positivos identificados por el modelo que son realmente positivos. Es decir, de todas las instancias que el modelo clasificó como positivas, ¿cuántas de ellas realmente pertenecen a esa clase? La fórmula es:

Precision = TP / (TP + FP)

donde FP (False Positive) representa las instancias negativas que fueron clasificadas erróneamente como positivas.

F1 Score es una medida combinada de Recall y Precision que proporciona una puntuación única que resume el rendimiento del modelo. Es la media armónica entre Recall y Precision. La fórmula es:

F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

El F1 Score proporciona una evaluación más completa del rendimiento del modelo, ya que tiene en cuenta tanto los verdaderos positivos como los falsos positivos y negativos.

Todos los modelos dieron valores muy buenos para las tres métricas.

Debido a que uno de los riesgos detectados era que la cantidad de datos para entrenar los algoritmos podría ser insuficiente, decidimos construir un segundo DataSet. En este punto se nos presentó el desafío de que no contábamos con más correos con intentos de phishing más allá de las 179 iniciales. Por lo que el segundo DataSet lo construimos con la misma cantidad de correos de phishing, pero incorporamos mayor cantidad de correos Ham:

DataSet 2
Cantidad de correos electrónicos:674
Correos electrónicos con intentos de Phishing:179
Correos electrónicos Ham:495

Tabla 6.2.3-7

Esta decisión la tomamos con la intención de que los algoritmos aprendan de los correos Ham para que luego puedan diferenciarlos de los correos con intentos de phishing. Además, basamos nuestra decisión en una investigación realizada por Andronicus et al. [20], quien presentó el artículo “Classification of Phishing Email Using Random Forest Machine Learning Technique” donde entrenó un modelo utilizando el algoritmo Random Forest con 2000 correos electrónicos, 200 correos de phishing y 1800 correos legítimos.

Los resultados obtenidos fueron los siguientes:

Experimento 5					
Algoritmo	**Exactitud**	**Ajuste de hiperparámetros**	**Recall**	**Precision**	**F1Score**
BaggingClassifier	0,837	estimator=tree,n_estimators=65,max_samples=10,random_state=42	1,00	0,52	0,69
RandomForestClassifier	0,948	n_estimators = 10, criterion='entropy', max_depth=5, min_samples_split=20, min_samples_leaf=10,random_state=42	1,00	0,85	0,92
GradientBoostingClassifier	0,977	n_estimators=12, learning_rate=0.07, max_depth=5, subsample=0.7, random_state=42	1,00	0,93	0,97
MultinomialNB	0,852	force_alpha=True, alpha=2, fit_prior=False	0,72	0,93	0,81

Experimento 6					
Algoritmo	**Exactitud**	**Ajuste de hiperparámetros**	**Recall**	**Precision**	**F1Score**
BaggingClassifier	0,807	estimator=tree,n_estimators=80,max_samples=15,random_state=42	1,00	0,43	0,61
RandomForestClassifier	0,985	n_estimators = 60, criterion='entropy', max_depth=7, min_samples_split=20, min_samples_leaf=10,random_state=42	1,00	0,96	0,98
GradientBoostingClassifier	1	n_estimators=700, learning_rate=0.05, max_depth=3, subsample=0.7, random_state=42	1,00	1,00	1,00
MultinomialNB	0,91	force_alpha=True, alpha=2, fit_prior=True	0,82	0,93	0,87

Ilustración 6.2.3-8

Una vez que realizamos la etapa de train y validation con el nuevo DataSet pasamos a la etapa de test. Testeamos los modelos con el mismo DataSet de Test: 144 correos entre phishing y ham. Como ya dijimos estos correos no están incluidos dentro del DataSet de train, es decir son otros correos. La idea en este punto es analizar cómo se comporta el modelo con nuevos correos sin etiquetar.

Los resultados obtenidos fueron los siguientes:

Test					
Algorítmo	**Exactitud**	**Ajuste de hiperparámetros**	**Recall**	**Precision**	**F1Score**
BaggingClassifier	0,736	estimator=tree,n_estimators=80,max_samples=15,random_state=42	0,476	1	0,648
RandomForestClassifier	0,93	n_estimators = 60, criterion='entropy', max_depth=7, min_samples_split=20, min_samples_leaf=10,random_state=42	0,863	1	0,926
GradientBoostingClassifier	0,972	n_estimators=700, learning_rate=0.05, max_depth=3, subsample=0.7, random_state=42	0,945	1	0,9718
MultinomialNB	0,68	force_alpha=True, alpha=2, fit_prior=True	0,671	0,69	0,68

Ilustración 6.2.3-9

De acuerdo a los resultados obtenidos se puede observar que BaggingClassifier es sensible al desbalance del DataSet ya que el Recall bajó de 0,94 a 0,476. El accuracy de MultinomialNB fue magro obteniendo un valor de 68%.

Comparando los resultados podemos decir que GradientBoosting fue menos sensible al desbalance que RandomForest, los dos algoritmos obtuvieron accuracy similares en validation y en test utilizando la configuración del experimento 5.

GradienBoostingClassifier obtuvo una exactitud en Validation y en Test prácticamente iguales, siendo este uno de los aspectos deseables; es decir que no exista una diferencia considerable entre Validation y Test.

Para la prueba de concepto utilizaremos los modelos creados con los clasificadores RandomForest, Gradient Boosting y Nayve Bayes.

6.2.4 Deploy del modelo

El despliegue de un modelo de Machine Learning es el proceso de poner en funcionamiento un modelo entrenado en un entorno de "producción". Una vez que nuestro modelo fue entrenado y validado en nuestro entorno de pruebas, debemos hacerlo disponible para que sea utilizado por nuestra aplicación web.

Existen varias formas de desplegar un modelo de Machine Learning, que dependen de los requisitos específicos del proyecto. En nuestro caso creamos un servicio web que proporciona una API para que pueda interactuar con el modelo.

El despliegue de un modelo de Machine Learning implica no solo la implementación del modelo, sino también la monitorización y el mantenimiento continuo para asegurar que sigue funcionando correctamente en el entorno de producción. Es importante realizar pruebas exhaustivas y garantizar que el modelo se ajuste a los requisitos del negocio y del usuario final.

Una vez que decidimos de qué forma analizaremos nuestros correos y determinamos qué algoritmo de Machine Learning utilizaremos, comenzamos a trabajar en la etapa de deploy de nuestro modelo para que pueda ser utilizado en nuestra aplicación web.

Primero creamos un archivo simplificado de nuestros experimentos tomando lo mejor de cada etapa:

Creamos una función que se encarga del "data cleaning" aplicando NLP

```
def text_cleaning(text, remove_stop_words=True, lemmatize_words=True):
```

Ilustración 6.2.4-1

Luego dividimos nuestro DataSet en Train y Validation.

Para poder realizar el deploy de nuestro modelo es necesario crear un pipeline, donde se especifique qué etapas se ejecutarán y en qué orden.

```
    body_classifier = Pipeline(steps = [('pre_processing',
TfidfVectorizer(lowercase=False)),


('clf',GradientBoostingClassifier(n_estimators=12, learning_rate=0.07,
max_depth=5, subsample=0.7, random_state=42))])
```

Ilustración 6.2.4-2

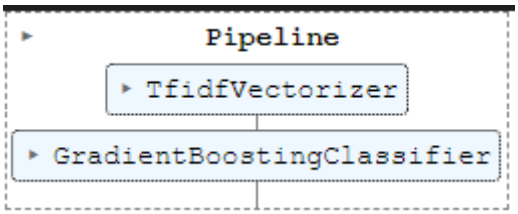


Ilustración 6.2.4-3

Una vez realizado lo descrito anteriormente, evaluamos nuestro modelo.

Por último, realizamos el dump de nuestro modelo, guardándolo en el archivo con formato pickle.

Los archivos pickle son una forma de serializar y guardar objetos de Python en un archivo binario. La serialización es el proceso de convertir un objeto en un formato que pueda ser almacenado o transmitido y luego recuperado para su uso posterior. Los archivos pickle pueden contener cualquier objeto de Python, como listas, diccionarios, clases y funciones, entre otros.

6.3 Metodología de trabajo

Como mencionamos al inicio de este capítulo utilizaremos dos ciclos de vida en paralelo, un ciclo de vida incremental iterativo, utilizando metodologías ágiles y un ciclo de vida de Machine Learning.

Utilizaremos elementos de las metodologías ágiles como el backlog con los requerimientos funcionales, el sprint planning, sprint review, estimación de las historias de usuarios con técnicas como poker planning. Pero no nos regimos por ninguna metodología en particular.

En cada iteración (sprint) realizaremos diseño, construcción y pruebas. Cada cierta cantidad de sprint’s tendremos un release como se verá en la siguiente sección 6.4 Plan de Releases. La estimación del esfuerzo la realizamos utilizando la técnica de poker planner y la unidad de esfuerzo serán story points. A cada story point le asignamos 4 horas de desarrollo sin interrupciones inicialmente, dato que fue variando con el devenir de los sprints y ante la evidencia real del esfuerzo

6.4 Plan de releases

El plan de releases de nuestro proyecto es un apartado que describe la estrategia de liberación de software para el proyecto. El plan de releases detalla cuando se va a entregar y lanzar el software, cuándo se va a realizar cada entrega y qué funcionalidades o características incluirá cada una de ellas. Podemos ver el plan de releases de nuestro proyecto en la Ilustración 6.4-1: Plan de Releases

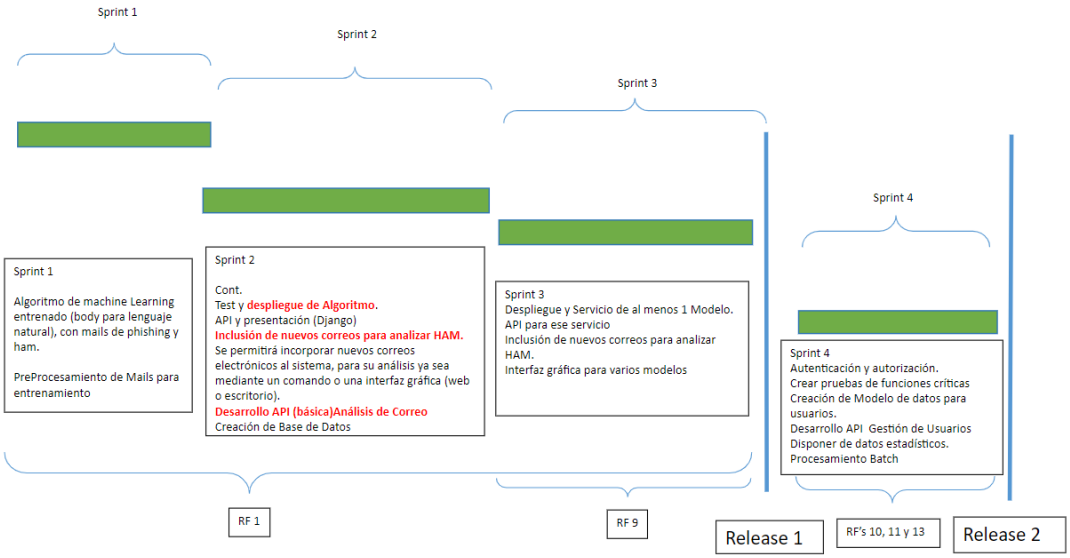


Ilustración 6.4-1: Plan de Releases

En general, un plan de releases se enfoca en la planificación de las entregas y no en el desarrollo del software en sí. Algunos de los elementos que suelen incluirse en un plan de releases son:

6.4.1 Sprints

Basados en el plan de releases antes presentado, y antes del inicio de cada sprint definimos qué tareas se incluirán en ese sprint, teniendo como criterio comenzar por las tareas que son el **Core** (o Centrales) del proyecto. Para poder estimar el esfuerzo del mismo, se usa la técnica de Poker Planning, en la cual sus integrantes estiman de manera independiente y de acuerdo a su experiencia un esfuerzo estimado. En caso de haber diferencias, se debe expresar los motivos que justifican cada una de esas decisiones para lograr llegar a un consenso, un acuerdo en cuanto al esfuerzo de cada tarea en particular.

Sprint 1

En el Sprint planning inicialmente definimos que realizaríamos las tareas MAS-24 a MAS-28. (nomenclatura de Jira, que fue la herramienta utilizada)

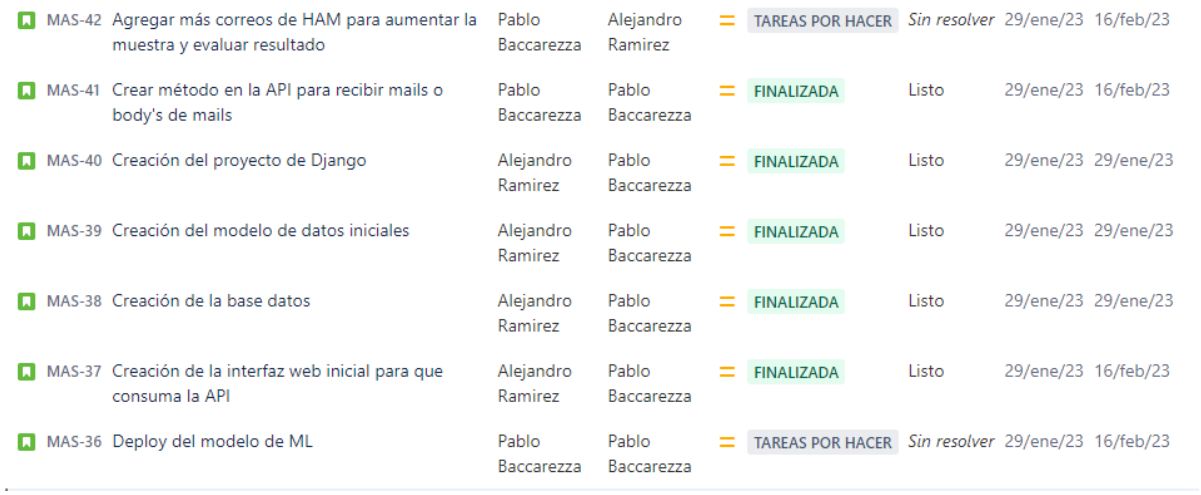
A medida que fue avanzando este sprint decidimos incorporar las tareas MAS-29 a MAS-35.

MAS-35	Crear API inicial	Alejandro Ramirez	Alejandro Ramirez	=	FINALIZADA	Listo	20/ene/23	16/feb/23
MAS-34	Utilizacion de archivo de configuracion	*sin asignar*	Alejandro Ramirez	=	TAREAS POR HACER	*Sin resolver*	20/ene/23	29/ene/23
MAS-33	Model testing	Pablo Baccarezza	Pablo Baccarezza	=	FINALIZADA	Listo	13/ene/23	29/ene/23
MAS-30	Model training and evaluaion - Entrenar y evaluar los clasificadores 1, 2 y 3	Pablo Baccarezza	Pablo Baccarezza	=	FINALIZADA	Listo	13/ene/23	29/ene/23
MAS-29	Data Understanding and preparation - Data cleaning	Pablo Baccarezza	Pablo Baccarezza	=	FINALIZADA	Listo	13/ene/23	29/ene/23
MAS-28	Data Understanding and preparation - NLP (Procesamiento de Lenguaje Natural)	Pablo Baccarezza	Pablo Baccarezza	=	FINALIZADA	Listo	11/ene/23	29/ene/23
MAS-27	Creación de archivo con id, body y label	Alejandro Ramirez	Pablo Baccarezza	=	FINALIZADA	Listo	11/ene/23	20/ene/23
MAS-26	Detección de Mail original	Alejandro Ramirez	Pablo Baccarezza	=	FINALIZADA	Listo	11/ene/23	14/ene/23
MAS-25	Tokenizado de Mail	Alejandro Ramirez	Pablo Baccarezza	=	FINALIZADA	Listo	11/ene/23	12/ene/23
MAS-24	Apertura del Mail	Alejandro Ramirez	Pablo Baccarezza	=	FINALIZADA	Listo	11/ene/23	12/ene/23

Ilustración 6.4.1-1: Tareas del Sprint 1

Sprint 2

Para el sprint 2 definimos realizar las tareas MAS-36 a MAS-42



MAS-42	Agregar más correos de HAM para aumentar la muestra y evaluar resultado	Pablo Baccarezza	Alejandro Ramirez	=	TAREAS POR HACER	*Sin resolver*	29/ene/23	16/feb/23
MAS-41	Crear método en la API para recibir mails o body's de mails	Pablo Baccarezza	Pablo Baccarezza	=	FINALIZADA	Listo	29/ene/23	16/feb/23
MAS-40	Creación del proyecto de Django	Alejandro Ramirez	Pablo Baccarezza	=	FINALIZADA	Listo	29/ene/23	29/ene/23
MAS-39	Creación del modelo de datos iniciales	Alejandro Ramirez	Pablo Baccarezza	=	FINALIZADA	Listo	29/ene/23	29/ene/23
MAS-38	Creación de la base datos	Alejandro Ramirez	Pablo Baccarezza	=	FINALIZADA	Listo	29/ene/23	29/ene/23
MAS-37	Creación de la interfaz web inicial para que consuma la API	Alejandro Ramirez	Pablo Baccarezza	=	FINALIZADA	Listo	29/ene/23	16/feb/23
MAS-36	Deploy del modelo de ML	Pablo Baccarezza	Pablo Baccarezza	=	TAREAS POR HACER	*Sin resolver*	29/ene/23	16/feb/23

Ilustración 6.4.1-2: Tareas del Sprint 2

Sprint 3

Dado que en el Sprint 2 no pudimos completar las tareas MAS-36 y MAS-42 las incluimos nuevamente en este sprint.

Por lo tanto, en el sprint 3 definimos realizar las tareas MAS-36, MAS-42 a MAS-44.

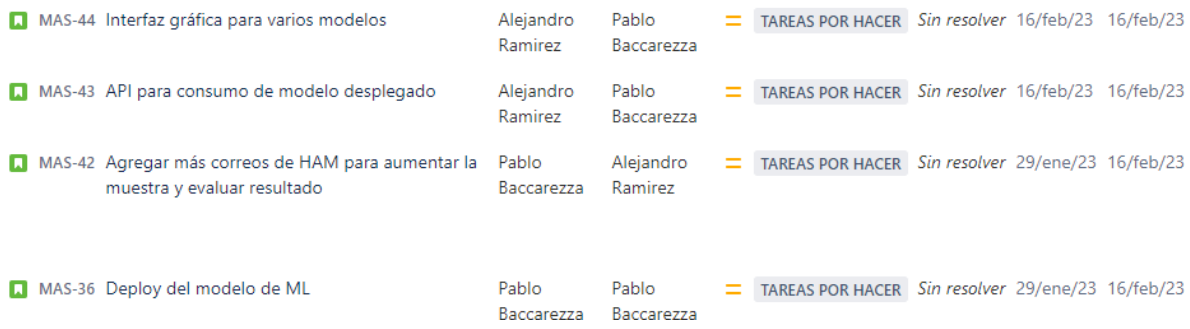


Ilustración 6.4.1-3: Tareas del Sprint 3

6.5 Técnicas, ceremonias y métricas

6.5.1 Técnicas

La técnica de poker planning es una herramienta de estimación de esfuerzo utilizada en la metodología ágil de desarrollo de software, en la que un equipo de desarrollo utiliza cartas de poker para llegar a un consenso sobre el esfuerzo necesario para completar una tarea o historia de usuario.

En esta técnica, cada miembro del equipo recibe un mazo de cartas de “poker” (no son cartas de póker reales, sino cartas con números de la serie Fibonacci del 1 al 13) que tienen valores predefinidos que representan la complejidad o el esfuerzo estimado para la tarea. Luego, el equipo discute la tarea y cada miembro elige una carta que represente su estimación de esfuerzo.

Una vez que todos los miembros han elegido una carta, se revelan simultáneamente y se discute la razón detrás de cada estimación. Si las estimaciones varían ampliamente, el equipo

discute las razones detrás de las diferentes estimaciones y repite el proceso. El objetivo es llegar a un consenso sobre una estimación final para la tarea.

En nuestro caso fue utilizada para la estimación de cada una de las tareas dentro de cada sprint, pudiendo entonces determinar el esfuerzo estimado para cada uno en cuanto a la cantidad de story points. En la medida que se fue progresando y entendiendo mejor la técnica, sumado al entendimiento de las herramientas, las estimaciones fueron bastante más precisas.

6.5.2 Ceremonias

Si bien no hemos utilizado estrictamente las metodologías ágiles en su esencia, si hemos adoptado ceremonias útiles de las mismas. De la metodología ágil SCRUM, hemos utilizado tanto Sprint Planning, que es el evento que se lleva a cabo al inicio del Sprint, en el que se planifica el trabajo a realizar, y también Sprint Review, que es el evento que se realiza al final del Sprint, en el que se revisa el trabajo completado y se obtiene retroalimentación para el siguiente Sprint. Ambos eventos son esenciales para el proceso ágil de desarrollo de software, ya que permiten a los equipos de desarrollo y los stakeholders mantenerse alineados y avanzar en el proyecto de manera efectiva.

Cabe destacar que hemos usado una alternativa acordada de sprints de duración variable. Si bien no es algo usual, nos ha sido útil para agrupar tareas relacionadas y con interdependencia de tareas posteriores. Con esto dicho, la duración de los sprints estuvo determinada por la sumatoria de los esfuerzos planificados para cada una de las tareas incluidas dentro del mismo.

6.5.3 Métricas de los sprints

En el proceso asociado tanto al análisis previo de las tareas de un Sprint, como al proceso de evaluación del mismo, se ha usado terminología específica para determinar ciertos valores y poder entonces hacer mediciones en cuanto a la performance estimada, así también como a la performance real. Es por eso que se han utilizado términos como Horas x story point, Horas promedio semanales dedicadas a construcción, velocidad del equipo, entre otros.

Cuando hablamos Horas por Story Point (SP), debemos referirnos al Story Point como una medida de tiempo necesario "ideal" para poder completar una determinada tarea. Es decir, una tarea X puede llevar *n* Story Points en un escenario favorable, situación que no siempre se da y que además involucra un conocimiento previo de las tareas involucradas si se pretende hacer una estimación adecuada. Inicialmente habíamos estimado 4 horas para cada SP, cuando no contábamos con información tangible, luego con los datos reales comenzó a ajustarse tendiendo hacia las **3 horas**.

Luego, mencionamos Horas promedios semanales, para poder determinar el esfuerzo para cada una de las tareas desglosadas en cada una de las instancias. Es decir, si bien según agenda se está en una etapa de construcción, por ejemplo, también hay tareas de gestión y documentación que hay que contemplar y que significativamente consumen tiempo del comprometido semanal, tanto individualmente como del equipo.

Otra métrica que fue considerada es la velocidad del equipo, como una forma de entender cuánto podemos construir en función de las 2 métricas anteriormente mencionadas.

Una vez que se cuenta con información real, pudimos hacer estimaciones más acertadas al respecto del esfuerzo de los nuevos sprints, y se evidenció una mejora en la capacidad de

estimar pues las tareas incluidas en subsiguientes sprints y su duración se ajustaron de manera aceptable a lo que se había pronosticado.

6.6 Conclusiones

Del proceso de construcción podemos decir que ha sido muy enriquecedor, y que es importante reflexionar sobre los logros y desafíos, para identificar oportunidades de mejora y optimizar el proceso de desarrollo de software en futuros proyectos.

Creemos que se ha completado la implementación del software y se han logrado las funcionalidades requeridas según las especificaciones y requisitos establecidos, se han identificado posibles mejoras o problemas que deben ser resueltos antes de la entrega del software.

El equipo de desarrollo ha adquirido experiencia y conocimientos en el uso de herramientas y tecnologías específicas, lo que puede facilitar la construcción de futuros proyectos.

Se ha logrado una buena comunicación y colaboración entre los miembros del equipo de desarrollo, lo que ha permitido trabajar de manera eficiente y efectiva.

7. Testing

Las pruebas de software son una parte crítica del proceso de desarrollo de cualquier aplicación o sistema de software. Son un conjunto de actividades diseñadas para evaluar la calidad del software y garantizar que cumple con los requisitos especificados por el cliente.

Las pruebas de software pueden incluir desde pruebas manuales realizadas por un equipo de testers hasta pruebas automatizadas que se ejecutan de forma sistemática y programada. En cualquier caso, el objetivo principal de las pruebas de software es asegurar que éste cumpla con los criterios de calidad y funcionalidad requeridos.

Las pruebas de software pueden ser realizadas en diferentes etapas del proceso de desarrollo de software, desde la fase de planificación hasta la implementación y mantenimiento. Existen diferentes tipos de pruebas de software, como pruebas de unidad, pruebas de integración, pruebas de aceptación, pruebas de carga, entre otras.

La implementación adecuada de las pruebas de software puede ayudar a evitar errores y problemas, lo que a su vez puede mejorar la satisfacción del cliente y el éxito del proyecto.

7.1 Plan de pruebas

Para la etapa de testing definimos los objetivos y alcance de las pruebas, así como la estrategia de las pruebas, es decir el tipo de pruebas que se realizaron.

El alcance de nuestras pruebas de software incluye los siguientes objetivos:

- Verificar que se cumplieron los requerimientos funcionales mandatorios descritos en la seccione 3.4
- Detectar errores y defectos en el software y reportarlos para su corrección.
- Evaluar el rendimiento del software en diferentes situaciones y condiciones.
- Evaluar la usabilidad del software y su capacidad para satisfacer las necesidades del usuario.
- Verificar la compatibilidad del software con diferentes browsers.

En la siguiente tabla se repasan los requerimientos funcionales mandatorios:

RF 1 Algoritmo de Machine Learning entrenado con correos etiquetados. Descripción: El sistema clasificará los correos como un intento de phishing o no con una cierta precisión, en función del entrenamiento que ha recibido hasta el momento. **Mandatorio**
RF 10 Inclusión de nuevos correos para analizar. Descripción: El sistema permitirá incorporar nuevos correos electrónicos al sistema, ya sea mediante un comando o una interfaz gráfica (web o escritorio). **Mandatorio**
RF 11 Autenticación y autorización. Descripción: Para ingresar al sistema, los analistas de seguridad de la información deben autenticarse mediante usuario y contraseña. Luego a través de un control de acceso y de acuerdo con sus permisos o privilegios, podrá acceder a aquellos recursos autorizados. **Mandatorio**

RF 12 Gestión de usuarios

Descripción: Alta Usuario: El sistema debe permitir a los administradores del sistema agregar un nuevo usuario.

Baja Usuario: El sistema debe permitir a los administradores del sistema, borrar un usuario existente.

Modificación Usuario: El sistema debe permitir a los administradores del sistema, modificar datos del usuario: Nombre, Apellido, descripción, contraseña, desbloqueo, etc.

Mandatorio

RF 14 Disponer de datos estadísticos específicos

Descripción: El sistema debe Mostrar:

- %de eficiencia del algoritmo
- % de casos detectados (detectados - reportados)
- de estos últimos, cuáles fueron falsos positivos y cuáles no.

Mandatorio

Tabla 7.1-1

7.2 Pruebas unitarias

Durante la etapa de desarrollo incorporamos pruebas unitarias que testean los métodos que se encuentran en el camino crítico de la etapa de procesamiento, previo a la etapa de clasificación de los correos electrónicos.

Realizamos pruebas unitarias a los principales métodos que intervienen en el proceso de validación, procesamiento y clasificación de los correos electrónicos. Concretamente realizamos 10 pruebas a los siguientes métodos:

- Validación de extensión correcta de archivo de correo electrónico
- Validación de extensión máxima de un correo electrónico
- Text cleaning del correo electrónico
- Clasificación de los correos electrónicos (probamos los tres algoritmos de machine learning para casos de ham y phishing).


Unit Test

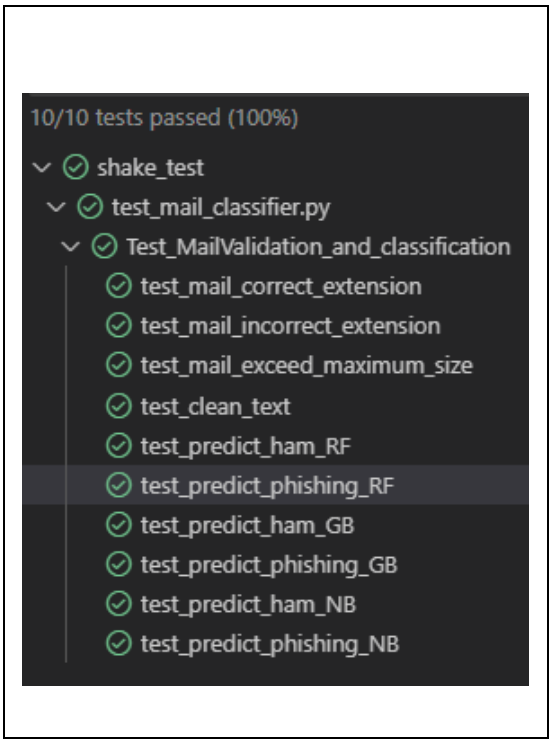


Ilustración 7.2-1

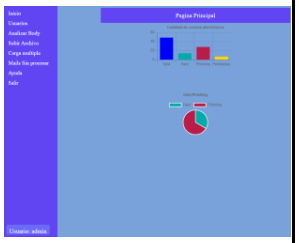
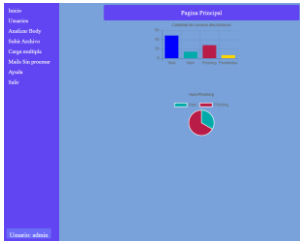
Estas pruebas dan como resultado que los métodos que intervienen en el cumplimiento de los RF1 y RF10 funcionan correctamente. Por otra parte, se procura asegurar que ante cambios en el código no se afecten estos métodos, y se monitoree mediante estas pruebas que los tests no fallen.

7.3 Pruebas funcionales

Para realizar las pruebas funcionales utilizamos la técnica de casos de prueba, utilizando algunos de los elementos descritos en la Norma ISO 29119. Según esta norma un caso de prueba es un conjunto de precondiciones, entradas y resultados esperados, desarrollados para impulsar la ejecución de un elemento de prueba para cumplir con los objetivos de la prueba, incluyendo la implementación correcta, la identificación de errores, el chequeo de la calidad entre otras.

El objetivo de estas pruebas es chequear que se dé cumplimiento a los requerimientos funcionales mandatorios RF1, 10, 11, 12 y 14.

A continuación, se muestra el resultado de las pruebas funcionales, desarrollado en casos de prueba:

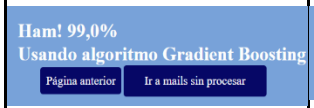
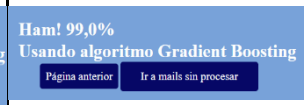
N°	Nombre	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados Actuales	Estado
1	CP-LO	Autenticación	-	-	1- Ingresar Usuario 2 – Ingresar contraseña 3 – Oprimir “Entrar” 	Accedo a la pantalla de Inicio 	Accedo a la pantalla de Inicio 	✓
2	CP-PHI1	Analizar un correo con intento de phishing, cargando el correo electrónico desde el file system, utilizando Random Forest	Estar autenticado	correo (1).msg	1-Seleccionar “Subir Archivo” 2 - Oprimir “Seleccionar Archivo” 3 – Cargar el archivo 4 – Oprimir “Subir Archivo” 4 - Seleccionar el archivo de la lista de correos sin procesar	La respuesta es Phishing con exactitud 99%, utilizando Random Forest 	La respuesta es Phishing con exactitud 99%, utilizando Random Forest 	✓

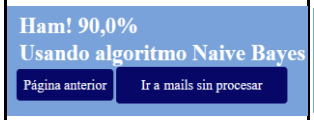
					5 - Oprimir “Procesar”. 6 - Seleccionar el modelo Random Forest para procesar el correo. 7 - Oprimir Analizar.			
Nro.	Nombre	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados Actuales	Estado
3	CP-PHI1	Analizar un correo con intento de phishing,	Estar autenticado	correo (1).msg	1-Seleccionar “Subir Archivo” 2 - Arrastrar y soltar el archivo	La respuesta es Phishing con exactitud 99%, utilizando Random	La respuesta es Phishing con exactitud 99%, utilizando Random Forest	✓

		utilizando drag and drop, seleccionando el algoritmo Random Forest			de correo en la zona punteada. 3 – Oprimir “Subir Archivo” 4 - Seleccionar el archivo de la lista de correos sin procesar 5 - Oprimir “Procesar”. 6 - Seleccionar el modelo Random Forest para procesar el correo. 7 - Oprimir Analizar.	Forest 		
4	CP-PHI2	Analizar un correo con intento de phishing, utilizando drag and drop, seleccionando el algoritmo Gradient Boosting	Estar autenticado	correo (1).msg	1-Seleccionar “Subir Archivo” 2 - Arrastrar y soltar el archivo de correo en la zona punteada. 3 – Oprimir “Subir Archivo” 4 - Seleccionar el archivo de la lista de correos sin procesar 5 - Oprimir “Procesar”. 6 - Seleccionar el modelo Gradient Boosting para procesar el correo. 7 - Oprimir Analizar.	La respuesta es Phishing con exactitud 97%, utilizando Gradient Boosting 	La respuesta es Phishing con exactitud 97%, utilizando Gradient Boosting 	✓

Nro.	Nombre	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados Actuales	Estado
5	CP-PHI3	Analizar un correo con intento de phishing, utilizando drag and	Estar autenticado	correo (1).msg	1-Seleccionar “Subir Archivo” 2 - Arrastrar y soltar el archivo de correo en la zona punteada.	La respuesta es Phishing con exactitud 100%, utilizando Naive Bayes	La respuesta es Phishing con exactitud 100%, utilizando Naive Bayes	✓

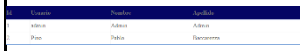
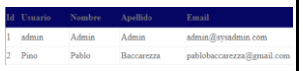
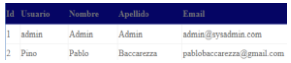
		drop, seleccionando el algoritmo Naive Bayes			3 – Oprimir “Subir Archivo” 4 - Seleccionar el archivo de la lista de correos sin procesar 5 - Oprimir “Procesar”. 6 - Seleccionar el modelo Naive Bayes para procesar el correo. 7 - Oprimir Analizar.			
6	CP-HAM1	Analizar un correo fidedigno (HAM) utilizando drag and drop, seleccionando el algoritmo Random Forest	Estar autenticado	presentismo.msg	1-Seleccionar “Subir Archivo” 2 - Arrastrar y soltar el archivo de correo en la zona punteada. 3 – Oprimir “Subir Archivo” 4 - Seleccionar el archivo de la lista de correos sin procesar 5 - Oprimir “Procesar”. 6 - Seleccionar el modelo Random Forest para procesar el correo. 7 - Oprimir Analizar.	La respuesta es Ham con exactitud 95%, utilizando Random Forest 	La respuesta es Ham con exactitud 95%, utilizando Random Forest 	✓

Nro.	Nombre	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados Actuales	Estado
7	CP-HAM2	Analizar un correo fidedigno (HAM) utilizando drag and drop, seleccionando el algoritmo	Estar autenticado	presentismo.msg	1-Seleccionar “Subir Archivo” 2 - Arrastrar y soltar el archivo de correo en la zona punteada. 3 – Oprimir “Subir Archivo” 4 - Seleccionar el archivo de la lista de correos sin procesar	La respuesta es Ham con exactitud 99%, utilizando Gradient Boosting 	La respuesta es Ham con exactitud 99%, utilizando Gradient Boosting 	✓

		Gradient Boosting			5 - Oprimir “Procesar”. 6 - Seleccionar el modelo Gradient Boosting para procesar el correo. 7 - Oprimir Analizar.			
8	CP-HAM3	Analizar un correo fidedigno (HAM) utilizando drag and drop, seleccionando el algoritmo Naive Bayes	Estar autenticado	presentismo.msg	1-Seleccionar “Subir Archivo” 2 - Arrastrar y soltar el archivo de correo en la zona punteada. 3 – Oprimir “Subir Archivo” 4 - Seleccionar el archivo de la lista de correos sin procesar 5 - Oprimir “Procesar”. 6 - Seleccionar el modelo Naive Bayes para procesar el correo. 7 - Oprimir Analizar.	La respuesta es Ham con exactitud 90%, utilizando Naive Bayes 	La respuesta es Ham con exactitud 90%, utilizando Naive Bayes 	✓

Nro.	Nombre	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados Actuales	Estado
9	CP-DEL1	Agregar un correo a la lista de correos sin procesar y eliminarlo.	Estar autenticado	RE-CRITERIOS 1567.msg	1-Seleccionar “Subir Archivo” 2 - Arrastrar y soltar el archivo de correo en la zona punteada. 3 – Oprimir “Subir Archivo” 4 - Seleccionar el archivo de la lista de correos sin procesar 5 - Oprimir en “Marcar” para eliminar de la lista.	El resultado es que se eliminó el archivo de la lista de correos sin procesar. 	El resultado es que se eliminó el archivo de la lista de correos sin procesar. 	✓
10	CP-EXT1	Agregar un correo con extensión no válida	Estar autenticado	Burn down chart.XLSX	1-Seleccionar “Subir Archivo” 2 - Arrastrar y soltar el archivo de correo en la zona punteada. 3 – Oprimir “Subir Archivo” 4 - Seleccionar el archivo de la lista de correos sin procesar 5 - Oprimir “Procesar”.	No procesa el archivo	No procesa el archivo	✓

Nro.	Nombre	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados Actuales	Estado
11	CP-AU	Dar de alta a un usuario	Estar autenticado	-	1-Seleccionar Usuarios 2-Seleccionar Alta 3-Ingresar datos de usuario: Nombre: Pablo Apellido: Baccarezza UserName: Pino Contraseña: ******** Verificación: ******** Mail: pablo.baccarezza@outlook.com Tipo de usuario: Usuario 4-Oprimir “Crear Usuario” 	El usuario se creó exitosamente	El usuario se creó exitosamente	✓

Nro.	Nombre	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados Actuales	Estado
12	CP-LU	Listar usuarios	Estar autenticado	-	1- Seleccionar “Usuarios” 2- Seleccionar “Listado”	Muestra el listado de usuarios del sistema 	Muestra el listado de usuarios del sistema 	✓
13	CP-MU	Modificar usuario	Estar autenticado	-	1- Seleccionar “Usuarios” 2- Seleccionar “Modificar” 3- Seleccionar de la lista el usuario a modificar 4- Modifico el correo electrónico	Se modificó exitosamente el correo electrónico del usuario 	Se modificó exitosamente el correo electrónico del usuario 	✓
14	CP-VC1	Al ingresar un nuevo usuario validar que la contraseña, coincida con la verificación	Estar autenticado	-	1-Seleccionar Usuarios 2-Seleccionar Alta 3-Ingresar datos de usuario: Nombre: Carlos Apellido: Santana UserName: Violero Contraseña: CsV123 Verificación: CsV123 Mail:	Como la contraseña y la verificación de esta coinciden, se permite Crear el usuario 	Como la contraseña y la verificación de esta coinciden, se permite Crear el usuario 	✓

					Carlos.santana@outlook.com Tipo de usuario: Usuario 4-Oprimir “Crear Usuario”			
Nro.	Nombre	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados Actuales	Estado
15	CP-VC2	Al ingresar un nuevo usuario validar que la contraseña, coincida con la verificación	Estar autenticado	-	1-Seleccionar Usuarios 2-Seleccionar Alta 3-Ingresar datos de usuario: Nombre: Carlos Apellido: Santana UserName: Violero Contraseña: CsV123 Verificación: CsV124 Mail: Carlos.santana@outlook.com Tipo de usuario: Usuario 4-Oprimir “Crear Usuario”	Como la contraseña y la verificación de esta no coinciden, se destaca el campo en rojo y no se permite Crear el usuario 	Como la contraseña y la verificación de esta no coinciden, se destaca el campo en rojo y no se permite Crear el usuario 	✓

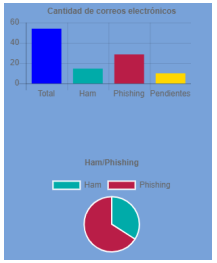
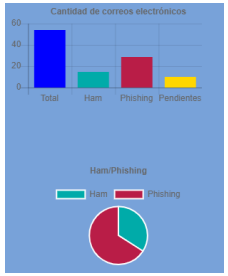
16	CP-CC	Cambiar contraseña de usuario	Estar autenticado	-	1- Seleccionar Usuarios 2- Cambiar contraseña 3 – Escribir las iniciales del UserName en el buscador 4 – Seleccionar al usuario 5 – Escribir la nueva contraseña 6- Verificar la nueva contraseña 7 – Presionar “Ir a modificar”	La contraseña se cambió correctamente 	La contraseña se cambió correctamente	
Nro.	Nombre	Descripción	Precondiciones	Entradas	Pasos	Resultados esperados	Resultados Actuales	Estado
17	CD-DE	Mostrar dashboards con datos estadísticos	Estar autenticado	-	1 – Seleccionar “Inicio”	Se muestran 2 dashboards con datos estadísticos 	Se muestran 2 dashboards con datos estadísticos 	✓

Tabla 7.3-1

Los casos de prueba descritos anteriormente dan cumplimiento a los siguientes requerimientos funcionales catalogados como mandatorios:

Caso de prueba Nro.	Requerimiento Funcional
1	RF11
2 al 8	RF1 y RF10
11 a 16	RF12
17	RF14

Tabla 7.3-2

7.4 Pruebas de performance

Si bien entendemos que la performance en este caso no es un atributo crítico, de todas formas realizamos pruebas para conocer el tiempo que tarda cada modelo, utilizando distintos algoritmos para clasificar los correos electrónicos.

			Tiempo en segundos			
Correo para clasificar	Tamaño del correo	Algoritmo utilizado por el modelo	Corrida 1	Corrida 2	Corrida 3	Promedio
correo1.msg	78 KB	Random Forest	0,10	0,11	0,10	0,10
correo1.msg	78 KB	Gradient Boosting	0,15	0,1	0,16	0,14
correo1.msg	78 KB	Naive Bayes	0,11	0,17	0,2	0,16
Correo largo 2.msg	151 KB	Random Forest	0,38	0,27	0,37	0,34
Correo largo 2.msg	151 KB	Gradient Boosting	0,20	0,30	0,18	0,23
Correo largo 2.msg	151 KB	Naive Bayes	0,31	0,32	0,31	0,31

Tabla 7.4-1

Como se puede ver en la tabla 7.4-1 los tiempos de clasificación son similares entre los tres algoritmos. Entendemos que estos tiempos son razonables ya que la clasificación de los correos es casi instantánea, por lo que el usuario no deberá esperar para conocer el resultado, por lo tanto, entendemos que la usabilidad es adecuada en este aspecto.

Utilizamos Postman para comprobar el tiempo de respuesta, realizando un request a la página de inicio:

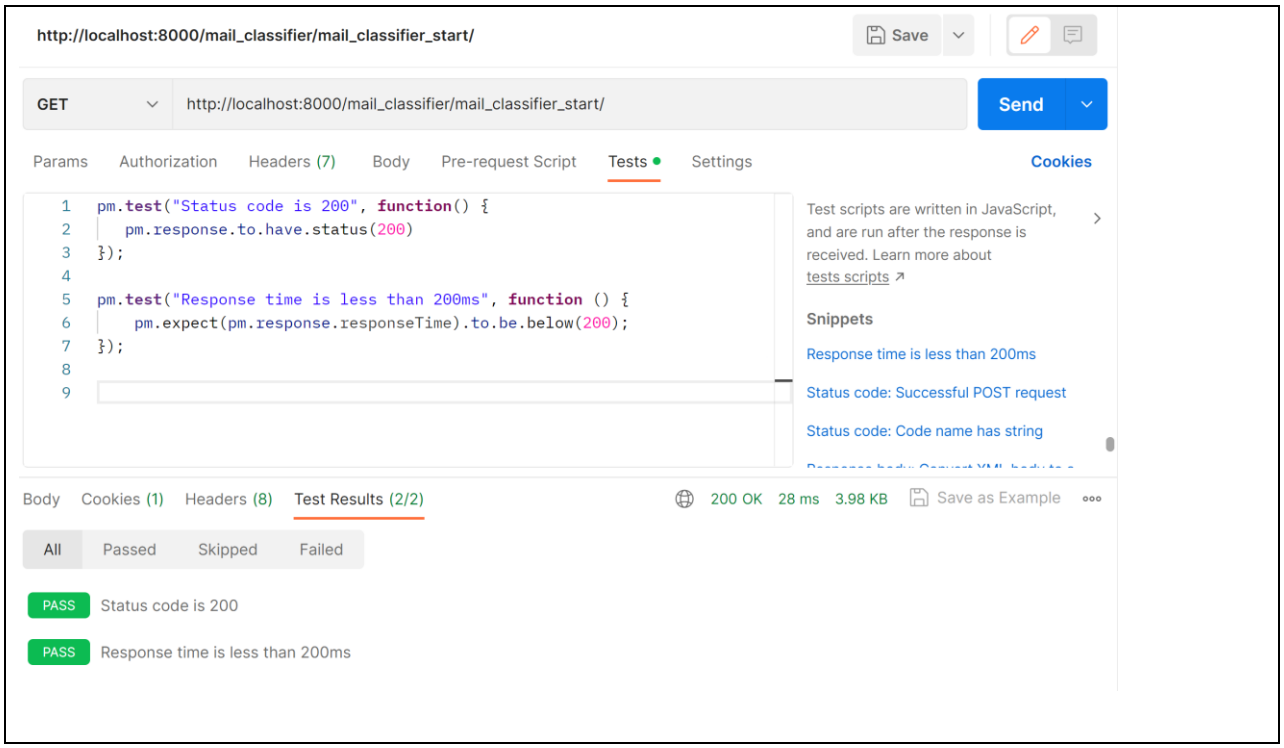


Ilustración 7.4-1

De acuerdo a lo que se puede observar en la ilustración 7.4-1 el tiempo de respuesta de la página de inicio es razonable, según indica postman la respuesta demoró 28ms. Por lo tanto, al igual que en el caso anterior entendemos que la usabilidad es adecuada en este aspecto.

7.5 Pruebas de compatibilidad

Probamos la aplicación web en tres browsers distintos para comprobar que la presentación y la funcionalidad fuera correcta:

Ilustración 7.5-1: Pantalla de inicio

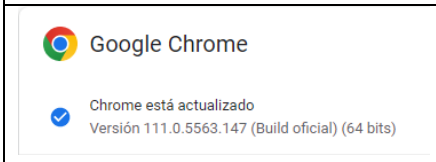


Ilustración 7.5-2: Pantalla de inicio

Ilustración 7.5-3: Pantalla de inicio

Como se puede observar en las ilustraciones anteriores las pruebas de compatibilidad en los tres browsers fue satisfactoria, ya que el diseño luce similar en los tres y la distribución de los elementos en pantalla es similar. La funcionalidad no cambia entre los tres browsers. Por lo tanto, el usuario obtendrá una experiencia similar utilizando indistintamente algunos de estos tres browsers.

7.6 Pruebas de los modelos de Machine Learning

En el capítulo 6, más precisamente en el 6.2.3 Training and evaluation, durante el desarrollo de los modelos de Machine Learning se realizaron varios experimentos donde se evaluaron los resultados de estos, utilizando distintos valores de los hiperparámetros hasta encontrar el modelo que mejor se adaptaba a nuestro proyecto.

7.7 Conclusiones

El resultado de las pruebas fue satisfactorio ya que pudimos comprobar que los principales métodos que intervienen en el proceso de validación, text cleaning y clasificación de los correos electrónicos funcionan correctamente.

Los tiempos de respuesta tanto de los algoritmos de Machine Learning como de los request http fueron satisfactorios, ya que no generaron tiempos de espera para el usuario.

Con las pruebas funcionales comprobamos que se cumple lo especificado en los requerimientos funcionales clasificados como mandatorios.

Si bien podríamos haber realizado mayor cantidad de tests unitarios, obteniendo de esta forma mayor cobertura de pruebas, en este caso por tratarse de una prueba de concepto, nos concentramos en testear los métodos que están en el camino crítico del proceso de validación, procesamiento y clasificación de los correos electrónicos, ya que entendemos que el resultado de esta prueba de concepto debe dejar como corolario que es posible clasificar correos electrónicos en categorías de phishing o ham aplicando algoritmos de Machine Learning.

8. Gestión del Proyecto

La gestión de proyectos de software es esencial para el éxito en la creación y desarrollo de soluciones tecnológicas. La complejidad inherente a la construcción de software requiere de una planificación detallada y una gestión efectiva de recursos, plazos y calidad para garantizar la entrega de un producto de software exitoso.

Abarca desde la definición del alcance del proyecto, la identificación de los requisitos hasta la planificación de las actividades y el control del progreso. En este sentido, es esencial que el equipo de trabajo sea capaz de adaptarse a los cambios del entorno y de aplicar las mejores prácticas para alcanzar los objetivos establecidos.

Esta práctica implica la aplicación de técnicas y herramientas para planificar, organizar, dirigir y controlar los recursos y actividades involucrados en el desarrollo de un software, con el objetivo de lograr los objetivos del proyecto en términos de tiempo y calidad.

Implica trabajar en equipo, definir y documentar los requerimientos del software, planificar y controlar el alcance del proyecto, establecer un calendario de entregas, gestionar los riesgos, administrar los recursos y comunicarse de manera efectiva con los stakeholders.

Es importante destacar que la gestión de proyectos de software no solo se enfoca en el desarrollo del software, sino también en la satisfacción del cliente, la gestión del cambio y el mantenimiento del software a lo largo del tiempo.

El proyecto comenzó el 17/10/22, con la definición de los principales objetivos y el establecimiento de indicadores para medir su cumplimiento. Luego, definimos la propuesta de valor, la visión del producto, realizamos un estudio de Benchmarking para conocer otros productos que ya dan solución a esta misma problemática. Durante el proyecto identificamos y gestionamos los riesgos detectados, y realizamos actividades de aseguramiento de la calidad.

8.1 Metodología del proceso de construcción

La metodología del proceso de construcción de software es el conjunto de pasos y prácticas que utilizamos para desarrollar software de forma sistemática y efectiva. Estos pasos incluyen desde la planificación inicial del proyecto, pasando por el diseño, la implementación y la verificación del software.

Para este proyecto utilizamos un ciclo de vida incremental iterativo con metodologías ágiles. La cual se enfoca en la entrega rápida y continua de software funcional, a través de ciclos de desarrollo iterativos e incrementales. El equipo trabajó de manera colaborativa y adaptativa, respondiendo a los cambios en los requisitos y en el entorno del proyecto.

El proceso de construcción de software incluye pasos como la definición de requisitos, la planificación y programación, la implementación, la prueba y la verificación, el mantenimiento y la documentación. El objetivo final es producir un software calidad que satisfaga las necesidades del usuario y cumpla con los requisitos del proyecto.

Comenzamos la gestión del proyecto con la planificación de las distintas etapas que lo componen, de acuerdo con las prácticas de ingeniería de software. Definimos roles y asignamos responsabilidades. Cada integrante tenía claro cuáles eran sus tareas asignadas y los plazos

definidos. Esto nos permitió monitorear el avance de las etapas y comparar con los plazos establecidos.

La gestión del proyecto la realizamos con el apoyo de la herramienta Jira Work Management, la cual nos permite organizar las tareas, definir plazos y asignar responsables. Una ventaja que nos brindó esta herramienta es la vista general de todo el proyecto, pudiendo visualizar claramente los hitos y el deadline del proyecto.

Ilustración 8.1

8.2 Seguimiento del plan

Durante y al final de cada etapa, realizamos un seguimiento de la planificación inicial, donde verificamos que se cumplieran los plazos establecidos y los objetivos programados.

A continuación, se muestran los gráficos con los días planificados por etapa y los días reales que nos tomó completar cada una.

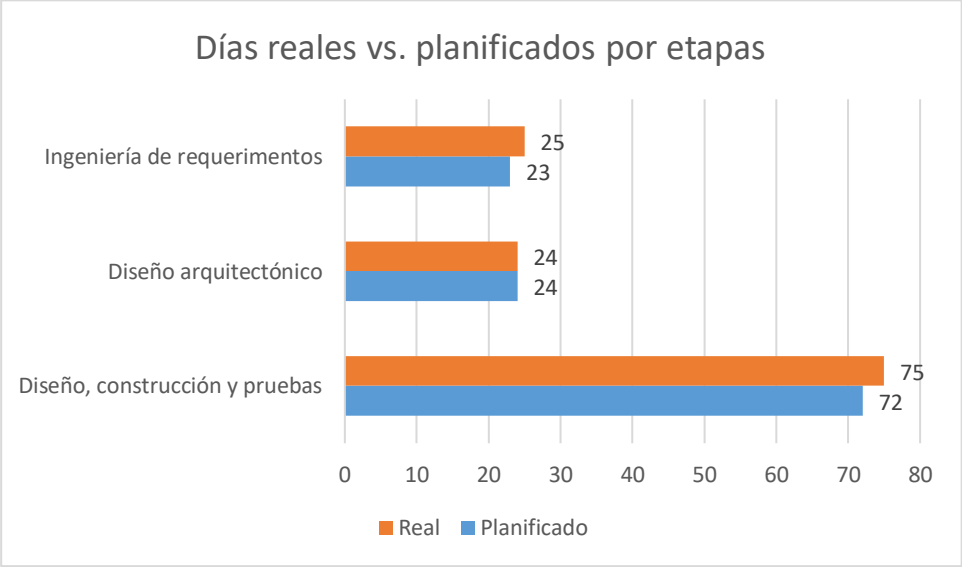


Ilustración 8.2-1

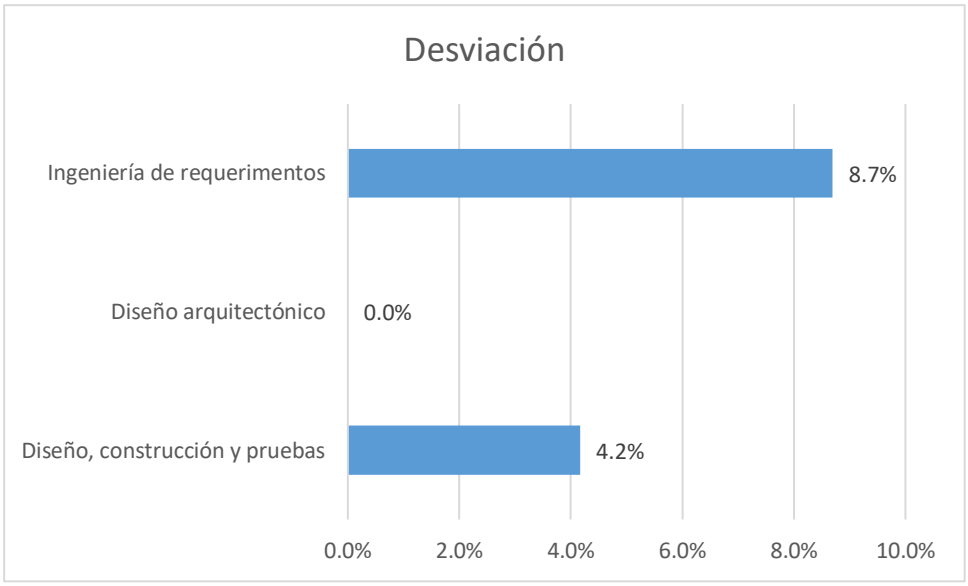


Ilustración 8.2-2

8.3 Gestión del tiempo

El plazo total para la ejecución de este proyecto fue de 176 días.

La etapa de ingeniería de requerimientos nos tomó 23 días desde su planificación hasta la validación, desde el 5/11/22 y hasta el 28/11/22. La etapa de diseño arquitectónico se llevó a cabo entre el 29/11/22 y el 23/12/22, es decir la realizamos en 24 días. La etapa de desarrollo se dividió en iteraciones de tiempo variable:

Sprint	Fecha	Duración
1	11/1/23 – 25/1/23	15
2	26/1/23 – 14/2/23	20
3	17/2/23 / 10/3/23	22
4	11/3/23 – 31/3/23	21

Tabla 8.3-1

El tiempo total de construcción fue de 78 días. Mientras que la documentación se realizó entre el 28/2/23 y el 8/4/23.

Hemos registrado las actividades en cuanto a la cantidad de horas empleadas en cada una de las etapas del proyecto, pero específicamente hemos trabajado sobre el desglose de tareas durante la etapa de construcción. En la ilustración siguiente (8.3-1: Detalle de horas), podemos ver el total de horas de esfuerzo para el proyecto para los integrantes, pero específicamente un control de horas dedicado a diferentes tareas en la fase de construcción por entender que se trata de una etapa de finalización de proyecto donde la optimización y control del esfuerzo es de suma importancia

TOTAL (sprints)	Construcción	Gestión	Documentación	Total x sprint
Sprint 1	67	22	1	100
Sprint 2	54	28	7	106
Sprint 3	81	14	8	111
Sprint 4	99	40	84	225

Ilustración 8.3-1, Detalle de horas durante los Sprints

Relación de Horas dedicadas a las diferentes actividades del proyecto:

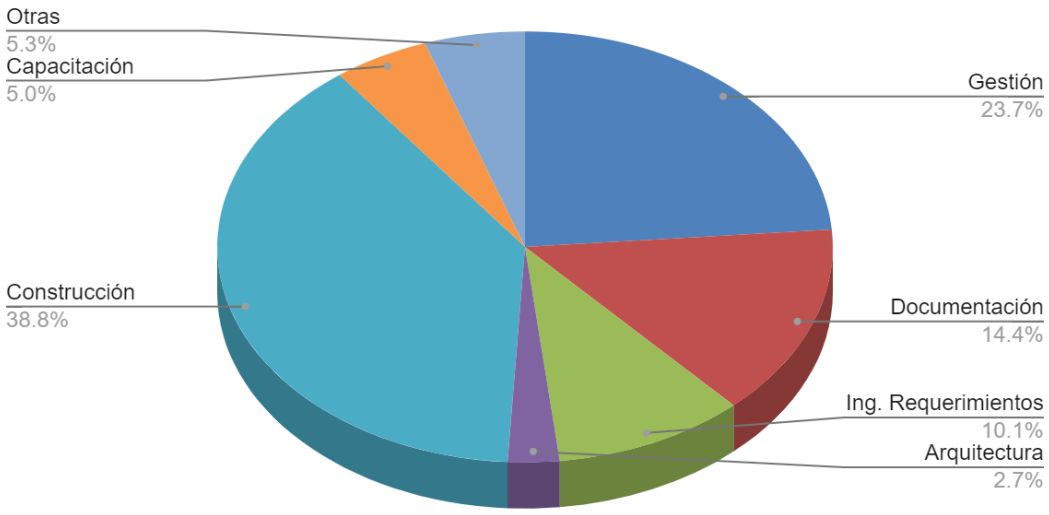


Ilustración 8.3-1

El proyecto tomó en total 800 horas con una dedicación aproximada de 30 horas por semana.

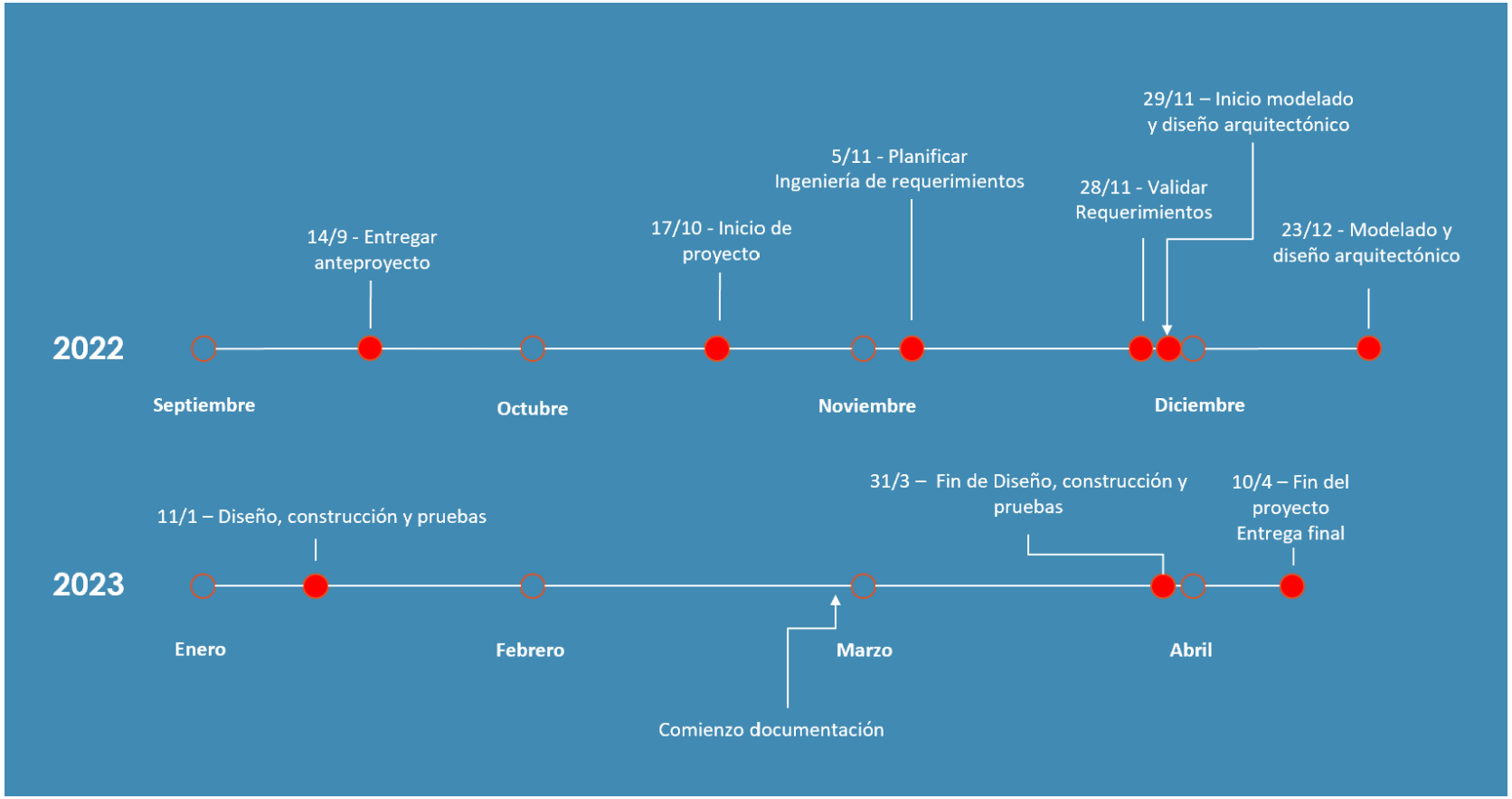


Fig. 8.3-1 Cronograma del proyecto

8.3.1 Capacidad del equipo

Al comienzo del proyecto realizamos un análisis de capacidad de cada integrante y del equipo. Estimamos la capacidad que se podía dedicar al proyecto en horas.

Capacidad	Horas
Alejandro Ramírez	15
Pablo Baccarezza	15
Capacidad semanal	30

Tabla 8.3.1-1

8.4 Gestión de la comunicación

Acción	Canal	Integrantes	Objetivo	Frecuencia
Reunión con Tutora para revisar los avances	Virtual – Microsoft TEAMS	Equipo - Tutor	Mostrar los avances a la Tutora, realizar un análisis junto y corregir posibles desvíos	Semanal
Sprint Planning	Virtual – Google meet	Equipo	Revisar el backlog y planificar el siguiente Sprint	Al principio de cada Sprint
Sprint Review	Virtual –	Equipo	Evaluar el resultado del	Al final de cada

	Google meet		Sprint	Sprint
Documentación del Proyecto, control de versiones	GIT para código fuente. Google Drive para la documentación	Equipo	Centralizar toda la documentación del proyecto, gestión de cambios	Cuando era necesario

Tabla 8.4-1

8.4.1 Comunicación interna del equipo

La comunicación del equipo se realizó por dos canales: WhatsApp y Google Meet. Utilizamos para el pasaje de documentación y archivos Google Drive y OneDrive, y las diferentes plataformas de Email disponibles Gmail, MsOutlook(interna del BROU). Encontramos que Google Drive realizaba ciertas modificaciones en el formato de los archivos, formato que debía ser conservado, por eso usamos para algunos archivos One Drive.

8.5 Análisis y gestión de riesgos

La gestión y análisis de riesgos es una actividad fundamental para tomar decisiones informadas y efectivas. El riesgo es una realidad inherente a cualquier actividad humana, y su gestión adecuada puede marcar la diferencia entre el éxito y el fracaso. El análisis de riesgos implica identificar, evaluar y priorizar los riesgos potenciales, mientras que la gestión de riesgos implica tomar medidas para mitigarlos y reducir su impacto en caso de que ocurran.

Nuestro equipo definió como actividad clave del proyecto, la identificación y seguimiento de los principales riesgos que podrían tener impacto en la correcta evolución de la prueba de concepto. Al comienzo del proyecto identificamos ciertas amenazas a las que estaba expuesto nuestro proyecto. Las clasificamos de acuerdo con la etapa en la que están presentes, la probabilidad de ocurrencia y el impacto que tendrían en caso de materializarse. Inicialmente planteamos controles para mitigar el impacto de materializarse una amenaza. Luego realizamos nuevamente la estimación de la probabilidad e impacto teniendo en cuenta los controles planteados.

8.5.1 Metodología de riesgos

Para el análisis y gestión de riesgos tomamos algunos elementos de la metodología MAGERIT – versión 3.0.[21]. Según [21] la experiencia ha demostrado la utilidad de métodos simples de análisis llevados a cabo por medio de tablas que, sin ser muy precisas, sí aciertan en la identificación de la importancia relativa de los diferentes activos sometidos a amenazas.

Utilizaremos la siguiente escala para calificar la magnitud del impacto del riesgo y la probabilidad de ocurrencia:

Impacto del riesgo	
1	Muy bajo
2	Bajo
3	Medio
4	Alto
5	Muy alto
Probabilidad de ocurrencia del riesgo	
0,0	No probable
0,25	Poco Probable
0,5	Probable
0,75	Muy Probable
1	Altamente Probable

Tabla 8.5.1-1 Impacto y probabilidad

El nivel de riesgo lo calculamos con la fórmula $probabilidad * impacto$

Estos son los riesgos detectados al inicio del proyecto:

Riesgo#	Etapa	Riesgo	Probabilidad	Impacto	Nivel de riesgo	Controles	Probabilidad	Impacto	Nivel de riesgo luego de controles
1	Todas las etapas	Estimación inadecuada del esfuerzo en las diferentes etapas	Muy Probable	Alto	3	Prever un 20% más del tiempo estimado	Probable	Alto	2
2	Todas las etapas	Falta de un integrante del equipo	Probable	Muy alto	2,5	Definir roles y responsabilidades claramente desde el principio	Poco Probable	Medio	0,75
3	Todas las etapas	Baja calidad del producto por falta de controles	Probable	Medio	1,5	Actividades de aseguramiento de la calidad mínimas adecuadas	Poco Probable	Bajo	0,5

Tabla 8.5.1-2

Riesgo detectado en la etapa de Ingeniería de Requerimientos:

Riesgo#	Etapa	Riesgo	Probabilidad	Impacto	Nivel de riesgo	Controles	Probabilidad	Impacto	Nivel de riesgo luego de controles
4	Ingeniería de requerimientos	Construcción de un sistema inadecuado por un incorrecto relevamiento y validación de requerimientos.	Probable	Alto	2	Realizar una correcta validación de requerimientos con el cliente	Poco Probable	Medio	0,75

Tabla 8.5.1-3

Riesgos detectados en la etapa Diseño, construcción y pruebas:

Riesgo#	Etapa	Riesgo	Probabilidad	Impacto	Nivel de riesgo	Controles	Probabilidad	Impacto	Nivel de riesgo luego de controles
5	Diseño, construcción y pruebas	Baja velocidad en la etapa de diseño, construcción y pruebas por ser un equipo reducido	Muy Probable	Muy alto	3,75	1. Definir correctamente el alcance 2. Realizar mediciones de la velocidad del equipo para realizar ajustes sobre el alcance	Probable	Alto	2
6	Diseño, construcción y pruebas	Falta de experiencia con la tecnología que utilizaremos en la etapa de diseño e implementación	Muy Probable	Muy alto	3,75	Realizar una capacitación previa y durante el diseño e implementación	Probable	Muy alto	2,5
7	Diseño, construcción y pruebas	No obtener la exactitud definida en el Algoritmo de Machine Learning por falta de experiencia	Muy Probable	Alto	3	Obtener una opinión experta	Probable	Bajo	1
8	Diseño, construcción y pruebas	Baja cobertura de pruebas redundando en una baja calidad de código	Probable	Alto	2	Realizar una cobertura de pruebas de las principales funciones	Poco Probable	Bajo	0,5
9	Diseño, construcción y pruebas	No obtener los resultados esperados con los modelos de Machine Learning por falta de datos	Muy Probable	Alto	3	Incorporar al dataset mayor cantidad de correos fidedignos (ham)	Probable	Medio	1,5

Tabla 8.5.1-4

Riesgo#	Etapa	Riesgo	Probabilidad	Impacto	Nivel de riesgo	Controles	Probabilidad	Impacto	Nivel de riesgo luego de controles
1	Diseño, construcción y pruebas	Baja velocidad en la etapa de diseño, construcción y pruebas por ser un equipo reducido	Alto	Muy alto	4.5	1. Definir correctamente el alcance 2. Realizar mediciones de la velocidad del equipo para realizar ajustes sobre el alcance	Medio	Alto	3.5
2	Diseño, construcción y pruebas	Falta de experiencia con la tecnología que utilizaremos en la etapa de diseño e implementación	Alto	Muy alto	4.5	Realizar una capacitación previa y durante el diseño e implementación	Medio	Muy alto	4
3	Todas las etapas	Estimación inadecuada del esfuerzo en las diferentes etapas	Alto	Alto	4	Prever un 20% más del tiempo estimado	Medio	Alto	3.5
4	Todas las etapas	Falta de un integrante del equipo	Medio	Muy alto	4	Definir roles y responsabilidades claramente desde el principio	Bajo	Medio	2.5
5	Todas las etapas	Baja calidad del producto por falta de controles	Medio	Medio	3	Actividades de aseguramiento de la calidad mínimas adecuadas	Bajo	Bajo	2
6	Ingeniería de requerimientos	Construcción de un sistema inadecuado por un incorrecto relevamiento y validación de requerimientos	Medio	Alto	3.5	Realizar una correcta validación de requerimientos con el cliente	Bajo	Medio	2.5
7	Diseño, construcción y pruebas	No obtener la exactitud definida en el Algoritmo de Machine Learning por falta de experiencia	Alto	Alto	4	Obtener una opinión experta	Medio	Bajo	2.5
8	Diseño, construcción y pruebas	Baja cobertura de pruebas redundando en una baja calidad de código	Medio	Alto	3.5	Realizar una cobertura de pruebas de las principales funciones	Bajo	Bajo	2
9	Diseño, construcción y pruebas	No obtener los resultados esperados con los modelos de Machine Learning por falta de datos	Alto	Alto	4	Incorporar al dataset mayor cantidad de correos fidedignos (ham)	Medio	Medio	3

Tabla 8.5.1-5 – Riesgos detectados durante todo el proyecto.

El seguimiento de los riesgos lo realizamos al final de la etapa de ingeniería de requerimientos y al final de cada sprint, de acuerdo con lo que se puede visualizar en el gráfico 8.5.1-1.

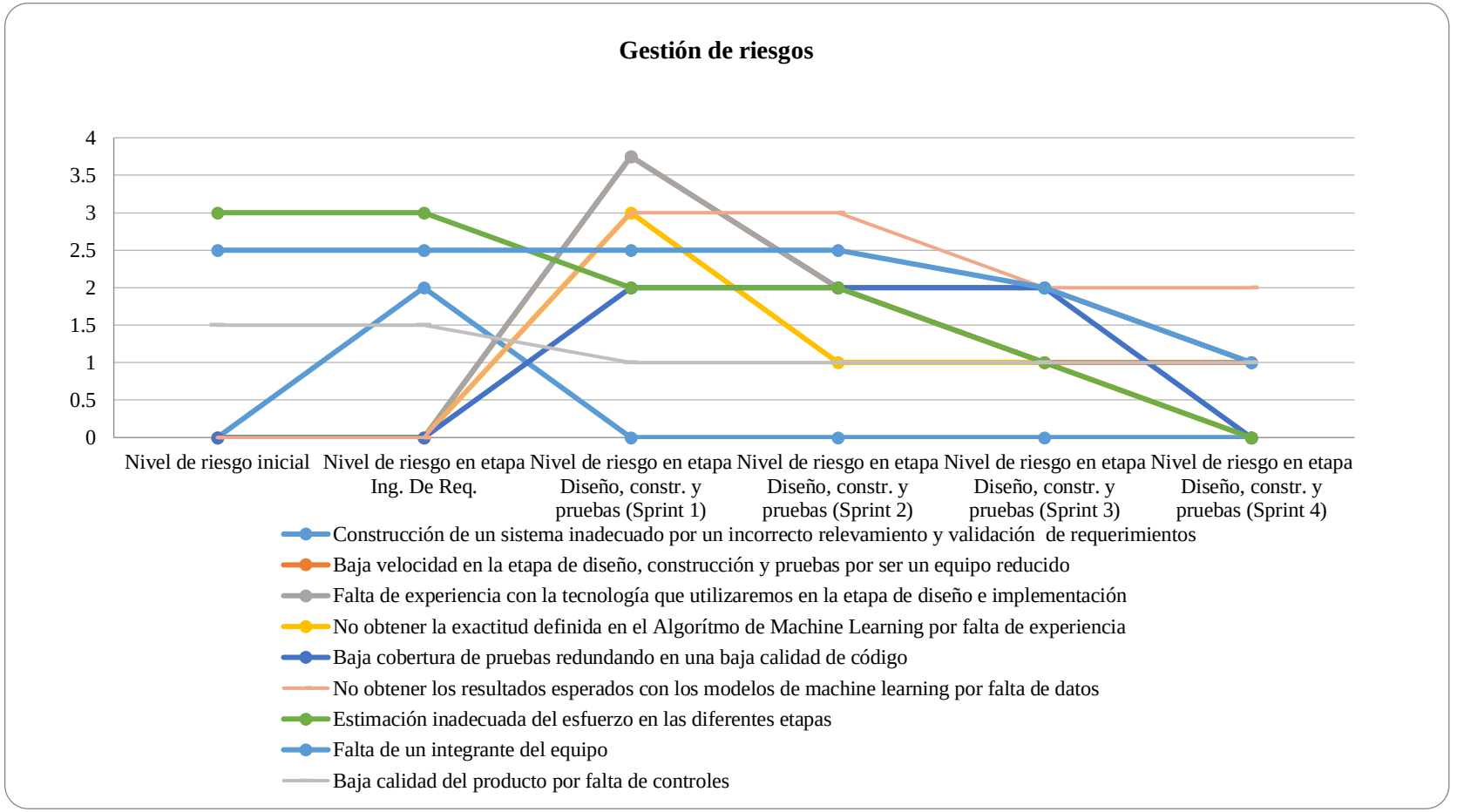


Gráfico 8.5.1-1.

8.6 Gestión del Alcance

La gestión del alcance es una actividad importante en la gestión de proyectos de software, ya que su objetivo es definir, planificar, monitorear y controlar todo el alcance del proyecto, asegurando que se cumplan los objetivos y requisitos definidos por los clientes y partes interesadas.

El alcance del proyecto se refiere a los límites y objetivos específicos que se deben alcanzar durante el desarrollo del software, lo que incluye tanto los productos y servicios que se entregarán, como las actividades y tareas necesarias para producirlos.

Una adecuada gestión del alcance permite tener una visión clara de lo que se debe hacer y de cómo se debe hacer, lo que ayuda a reducir los riesgos, evitar la duplicación de esfuerzos y asegurar que el proyecto se entregue dentro del plazo establecido.

Una vez concluida la etapa de validación de requerimientos, definimos el alcance, seleccionando aquellos requerimientos que fueron clasificados como mandatorios.

8.7 Evaluación de los Sprint

Al final de cada Sprint realizamos el Sprint Review donde analizamos el resultado del Sprint. En esta ceremonia evaluamos si efectivamente se realizaron todas las tareas programadas, además se analiza la diferencia entre el esfuerzo planificado y el esfuerzo real. También se evalúan qué actividades deben mejorar y cuáles salieron bien.

Sprint 1:

Horas planificadas	Horas reales	Diferencia %	Horas reales por Story Points
96	61	-36.5%	2.5

Tabla 8.7-1

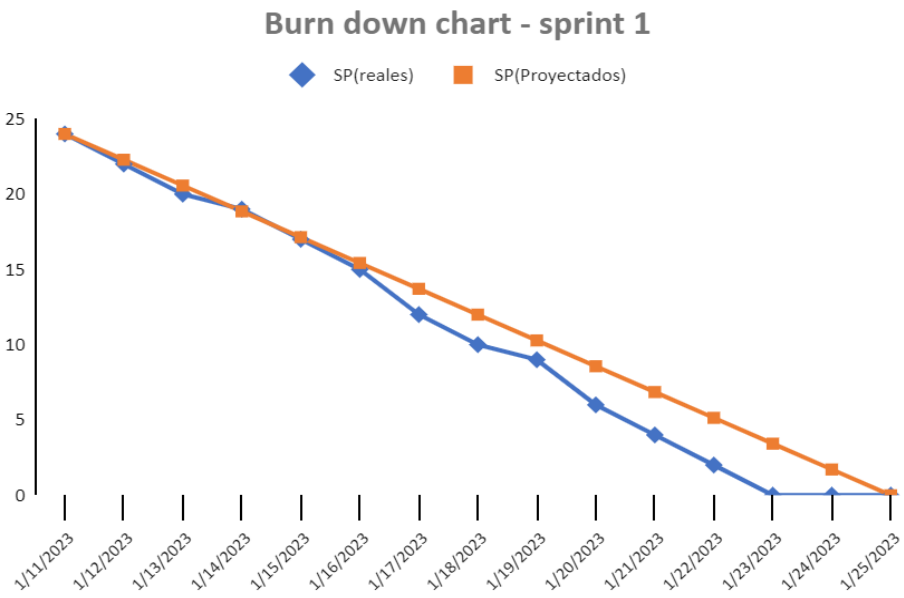


Ilustración 8.7-2: Burndown Chart Sprint 1

Sprint 2:

Horas planificadas	Horas reales	Diferencia %	Horas reales por Story Points
67.5	54	-20%	2.6

Tabla 8.7-2

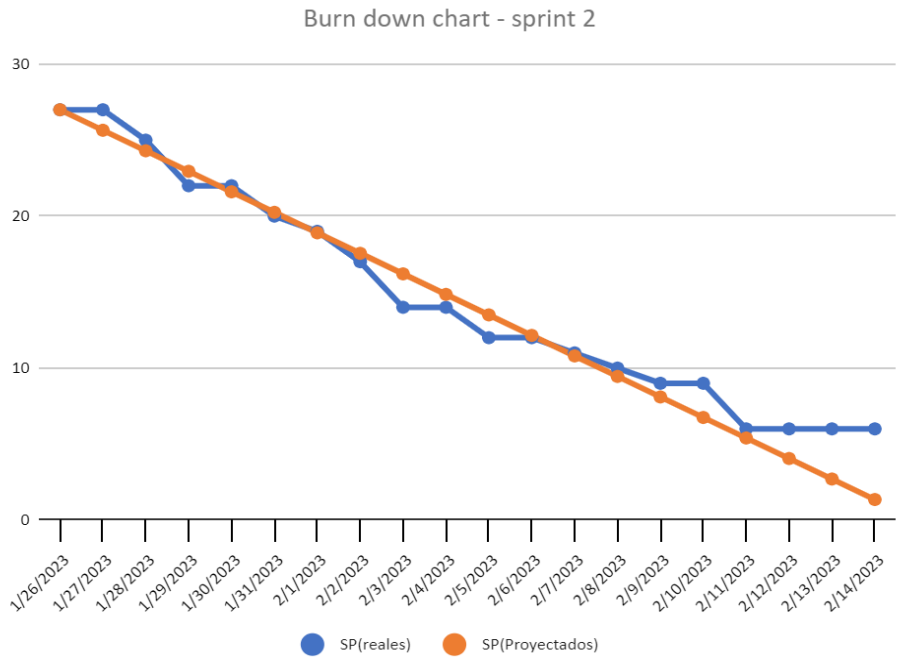


Ilustración 8.7-3: Burndown Chart Sprint 2

Sprint 3:

Horas planificadas	Horas reales	Diferencia %	Horas reales por Story Points
55	56	1.8%	2.5

Tabla 8.7-3

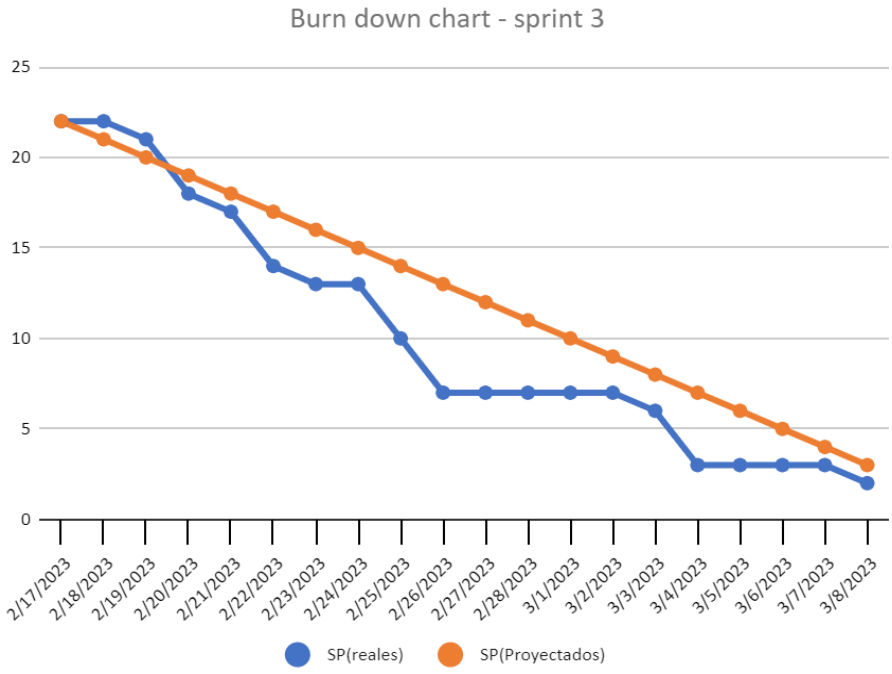
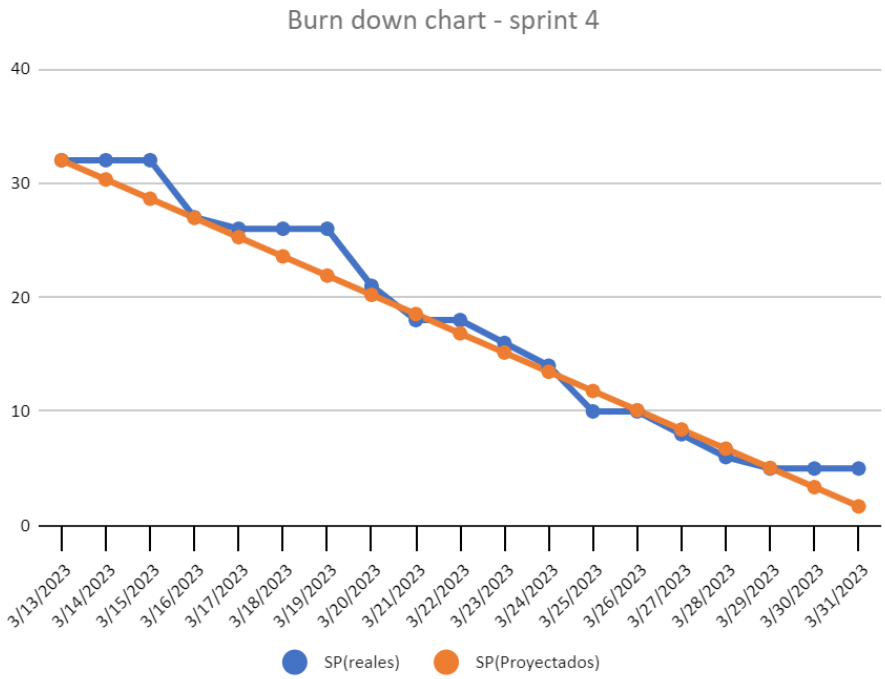


Ilustración 8.7-4: Burndown Chart Sprint 3

Sprint 4:

Horas planificadas	Horas reales	Diferencia %	Horas reales por Story Points
80	86	7.5%	2.7



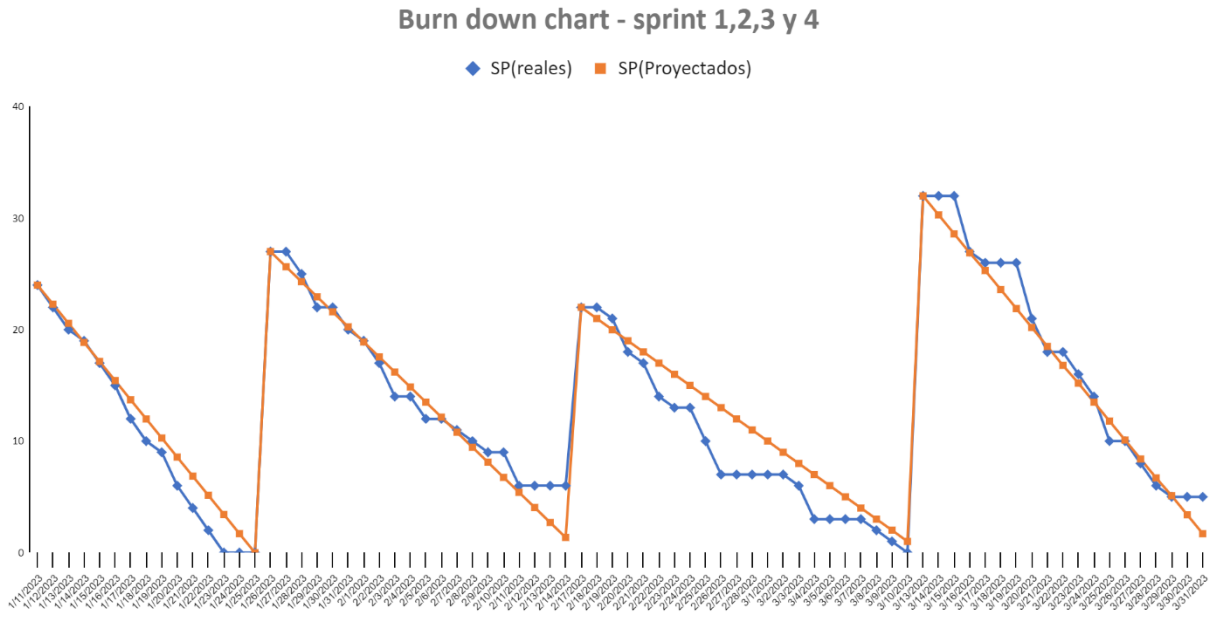


Ilustración 8.7-5: Burndown Chart Sprints 1,2,3 y 4

8.8 Conclusiones

La gestión del proyecto la realizamos durante todo el ciclo de vida de este desarrollo. Esta actividad nos permitió realizar una planificación de las distintas etapas, a través de estimaciones y siguiendo las prácticas de Ingeniería de Software. Realizamos seguimiento de lo planificado, comparando el esfuerzo real versus el planificado, en algunas etapas tuvimos que replanificar debido a que las estimaciones no fueron del todo correctas. Entendemos que la dedicación horaria semanal fue correcta, ya que pudimos cumplir con los objetivos trazados.

El análisis de riesgo nos permitió tener una mejor idea de las amenazas presentes en cada etapa y el impacto que podría tener la materialización de los riesgos identificados.

9. Aseguramiento de la Calidad

El aseguramiento de la calidad en el desarrollo de proyectos de software es un proceso que se enfoca en garantizar que el software producido cumpla con los requerimientos establecidos y tenga la calidad esperada por los usuarios y los stakeholders.

Este proceso incluye la aplicación de técnicas y herramientas para planificar, supervisar y mejorar continuamente la calidad del software.

A continuación, se describirán los objetivos de calidad del proyecto, planteados por el equipo y que se desprenden de los requerimientos relevados y validados.

Las tareas de aseguramiento de la calidad incluyen actividades que tienen por objetivo lograr validaciones de las diferentes etapas en todo el proyecto. Estas actividades se realizaron desde etapas tempranas como ingeniería de requerimientos, diseño de la arquitectura y modelado, durante el proceso de desarrollo del software y la documentación, y también en la etapa de construcción. Esto lo veremos en la sección 9.2 Actividades de Calidad

9.1 Objetivos de Calidad

Se definieron los objetivos de calidad en las diferentes áreas del proyecto, considerando desde el comienzo que se trata de una Prueba de Concepto y ciertas particularidades de la misma.

Entonces, se desea asegurar un nivel aceptable de calidad desde las siguientes perspectivas:

- **Desde el punto de vista de la Prueba de Concepto**: Que demuestre que la hipótesis planteada para este proyecto es válida, pero que además agregue valor a los usuarios objetivo dentro de un entorno cuya usabilidad sea aceptable.

- **Desde el punto de vista de la viabilidad técnica**: Que sea una propuesta realizable dentro de las capacidades de la infraestructura del banco.

- **Desde el punto de vista del proceso**: Teníamos el desafío de integrar técnicas de Machine Learning con técnicas tradicionales, de la que hay escasas referencias, cuyos ciclos de vida debieron ser adaptados, además de controlar y medir, para determinar si las decisiones tomadas lograron los objetivos exitosamente.

9.2 Actividades de Calidad

Para asegurar un nivel aceptable de calidad de nuestra Prueba de Concepto, se definieron ciertas actividades para las correspondientes a las etapas propias del proyecto y algunas relativas a todo el proyecto:

- Ingeniería de requerimientos
- Diseño Arquitectónico
- Construcción
- Gestión de riesgo
- Gestión del proyecto

9.2.1 Ingeniería de requerimientos

Como actividad de aseguramiento de la calidad, en esta etapa hemos intercambiado un documento con los requerimientos detectados y detallados, que luego posteriormente y de manera presencial fueron validados. Durante la ingeniería de requerimientos se grabaron las entrevistas, se transcribieron para luego validar el contenido con los entrevistados. A posterior, de estas entrevistas surgieron los requerimientos funcionales, que también fueron validados por los interesados entrevistados, también por el gerente de Seguridad de la información en una instancia formal. En este mismo punto, se presentó un prototipo inicial, con ciertas funcionalidades ya creadas para validar si habíamos logrado entender el foco del problema que debíamos resolver.

9.2.2 Diseño Arquitectónico

Validados los requerimientos y el prototipo, diseñamos una arquitectura inicial. Se acordó una reunión con el equipo de arquitectura del Banco para validar nuestro diseño. Debido a que no fue posible coordinar rápidamente una reunión con el equipo de arquitectura del Banco, comenzamos con la etapa de construcción.

En la documentación enviada al equipo de arquitectura incluimos información contextual para la lectura inicial, con el objetivo de tener un primer acercamiento. Esta tarea de preparación previa se lleva a cabo con la finalidad de complementar un intercambio previo a la reunión. En la reunión se presentó formalmente la arquitectura diseñada en la que nos basamos.

Esta arquitectura abarca todos los requerimientos, además de contemplar aquellos aspectos de infraestructura, requerimientos no funcionales y restricciones también detectados. Por ejemplo, las restricciones en términos que implican el procesamiento “on premise” y sistemas operativos homologados y licenciados. La arquitectura propuesta, se basa en un modelo de sitio web, estándar, cuyos diagramas y contexto son presentados, para su validación, obteniendo una muy buena devolución.

Se aprueba el diseño arquitectónico, aprovechando la oportunidad de que teníamos una versión inicial funcionando, lo que fue de utilidad para mostrar el potencial. De ese intercambio surgen sugerencias en el mecanismo de notificaciones, que son anotados y tomados en cuenta para una siguiente modificación. El detalle de esta tarea puede verse en la sección **4.2.1 Validación**

Como fue mencionado oportunamente, las condiciones son adecuadas para la prueba de concepto, contamos con el aval del equipo de Arquitectura del Banco. La arquitectura propuesta es estándar de la industria y no es una innovación en términos de componentes o capas, tampoco en cuanto a necesidades de infraestructura, pero por supuesto siendo necesaria un análisis en mayor profundidad si esta idea prospera y se pasa a niveles más avanzados.

9.2.3 Construcción

Durante la construcción de un software, se deben llevar a cabo una serie de actividades de aseguramiento de la calidad para garantizar que el software sea confiable, seguro y cumpla con los requisitos del cliente. Las actividades de aseguramiento de la calidad que se llevaron a cabo son las siguientes:

- Control de Versiones
- Revisión de código
- Control de tareas
- Pruebas
- Control de Horas

9.2.3.1 Control de Versiones

El control de versiones mediante el uso de Github y SourceTree, que permiten una integración del código entre los miembros del equipo para trabajo colaborativo, asegurando de esta manera la correcta utilización de las versiones adecuadas. (Ver Ilustración 10-1: Gestión de la Configuración, SourceTree).

9.2.3.2 Revisión de código

La Revisión de código se lleva a cabo de manera Ad-Hoc, por parte de cada uno de los integrantes para identificar y corregir errores de programación, asegurando que el software sea de la mejor calidad posible y se adhiera a las mejores prácticas de codificación. Los resultados de estas revisiones generalmente tienen pequeñas modificaciones en el código que no son registradas como pare de constante refactoring de código, pero aquellos cambios que implican un cambio importante se registran en la planilla de tareas que veremos en la sección 9.2.3.3. Esto nos ha permitido ampliar el rango de mejoras y un mayor control de los tiempos de respuesta, sobre todo en cuanto a los errores que ameritan atención preferencial.

También se utilizaron controles de excepciones en aquellos sitios identificados donde una falla en la ejecución de código compromete el funcionamiento de todo el sistema.

(Ver Ilustración 8.1-1: Control de Excepciones).

```
try:
    # Marcar como analizado y mostrar mas datos
    # Pasar ademas el algoritmo y el resultado
    if result == 'ham':
        mark_as_procesed_auto(idmail, False, model_selected_str, probability)
    else:
        mark_as_procesed_auto(idmail, True, model_selected_str, probability)

    return render(request, 'files/result.html', {'result': result, 'probability':probability, 'model_used':model_selected_str})
except:
    return render(request, 'files/analyzed_error.html')
```

Ilustración 9.1.3-1: Control de Excepciones

9.2.3.3 Control de Tareas

En la etapa de construcción se implementó un control de tareas, con la finalidad de poder llevar un backlog de tareas, modificaciones y bugs a ser corregidos. Los plazos de tiempo disponible dificultaron utilizar una herramienta de nivel profesional, por lo que hemos decidido usar una planilla en Excel customizada para esta tarea. (ver Ilustración 8.1-2: Control de Tareas).

H1 fx Solucionado

	A	B	C	D	E	F	G	H
1	Fecha	Tipo	Estado	Nivel	Detalle	Modulo	Funcion	Solucionado
2	16/2/2023	Bug	Terminado	3	Token elegido "A" presente en algunas palabras, se crea el archvo csv, pero falla su procesamiento	mail_process_manual	abrir_un_mail.py	2/3/2023
3	27/2/2023	Mejora	Terminado	1	Pasaje de body de mail externo para analisis	Mail_classifier	views/processing_mail	1/3/2023
4	28/2/2023	Bug	Terminado	2	No refresca mails borrados de la tabla (con comando desde workbench y luego refresco)	views	uploaded_mail_display	2/3/2023
5	28/2/2023	Mejora	Iniciado	3	Pasar estilos de pagina de evaluacion de body a CSS	index.html	css	
6	28/2/2023	Mejora	Terminado	3	Alineado de los botones en index.html	index.html	css	7/3/2023
7	1/3/2023	Mejora	Iniciado	3	Mejor extraccion de Body para procesar en index.html	mail_process	abrir_un_mail.py	
8	1/3/2023	Mejora	Terminado	1	Solucionar lo de drag and drop en http://localhost:8000/mail_classifier/mail_upload_file/	mail_upload_file/		4/3/2023
9	2/3/2023	Mejora	Creado	2	Diseñar y crear tabla e interfaz para mostrar datos estadisticos			
10	2/3/2023	Mejora	Terminado	2	Agrupar tareas de procesado de mails en una aplicacion para que no queden sueltas en la estructura	mail_process, mail_process_manual		2/3/2023
11	2/3/2023	Mejora	Terminado	2	Opcion para borrar mails sin procesar o marcar mails como procesados	views		8/3/2023
12	3/3/2023	Mejora	Creado	2	Marcar mails procesados y anotar estadisticas			
13	4/3/2023	Mejora	Creado	3	Verificar estado de BD al iniciar el sistema			
14	6/3/2023	Mejora	Creado	2	Procesamiento batch para presentacion			
15	7/3/2023	Mejora	Terminado	2	Ordenar Salida de mails subidos (primero los ultimos subidos)	views	uploaded_mail_display	8/3/2023
16	7/3/2023	Mejora	Terminado	2	Filtrar Salida de mails subidos (Solo los NO procesados)	views	uploaded_mail_display	8/3/2023
17	7/3/2023	Mejora	Terminado	1	Agregar mas modelos de nuevos algoritmos a la evaluacion	views		11/3/2023
18	7/3/2023	Mejora	Creado	3	Controlar que no se entra un archivo con el mismo nombre y el mismo usuario			

Ilustración 9.2.3.3-1: Control de Tareas

Los resultados de los controles que se desprenden del control de horas pueden verse con más detalle en la sección siguiente:

9.2.4 Gestión de Riesgos

Se definen riesgos detectados inicialmente, asignándoles atributos de “probabilidad” e “impacto”.

Se aplican controles que ayudan a minimizar esos riesgos. Existen riesgos propios de cada etapa que caducan al finalizar dicha etapa. Ver Sección 8.5 Análisis y gestión de riesgos

9.2.5 Gestión del proyecto

Como uno de los mecanismos de gestión del proyecto, se deja registro de las horas trabajadas y un detalle del tipo de tarea para su posterior categorización. De esa manera podemos saber la cantidad de esfuerzo del equipo para cada una de las etapas, evaluar las desviaciones en cuanto a las estimaciones y tomar acciones correctivas para su ajuste.

Ver capítulo 8 Gestión de Proyecto.

9.2.6 Apoyo a las actividades

Desde etapas tempranas del proyecto utilizamos como apoyo una planilla de actividades de aseguramiento de la calidad con la finalidad de tener un listado de tareas que creíamos indispensable realizar para mantener ciertos criterios de calidad aceptables. Si bien sabemos que el proceso es mejorable, esta ayuda nos permitió contar con cierta forma de trabajo homogénea.

Adjuntamos la tabla de tareas mencionada:

Actividades

Etapa	Actividad	Entrada	Salida	Herramienta	Encargado
Investigación	Investigar herramientas y tecnologías	Artículos, libros, videos, Consulta a expertos, Proyectos anteriores	Documentos de investigación para proyecto	Reuniones, Videos, Libros	Equipo de proyecto
	Capacitación en nuevas tecnologías y herramientas	Artículos, libros, videos, Consulta a expertos	Documentos, ejemplos de Pruebas y desarrollo	Videos, Libros, artículos, sitios técnicos	Equipo de proyecto
Requerimientos	Identificación de Requerimientos	Descubrimiento de requerimientos, Entrevistas	Listado de Requerimientos	Reuniones, y conocimiento experto del caso	Equipo y Stakeholders
	Aprobación o Validación de Requerimientos	*Listado de Requerimientos Funcionales*	Listado de Requerimientos confirmados	Reuniones, Mail	Equipo y Stakeholders
	Priorización de Requerimientos	Listado de Requerimientos Funcionales	Listado de RF's priorizados	Reuniones entre equipo de proyecto y Stakeholders	Equipo y Stakeholders
	Definición del alcance del proyecto	Listado de Requerimientos Funcionales priorizados. Backlog del proyecto	Definición del alcance y backlog con pendientes que quedaron fuera	Reuniones entre equipo de proyecto y Stakeholders	Equipo y Stakeholders
	Estimación de Requerimientos	Listado de Requerimientos Funcionales Priorizados y alcance definido.	Confirmación del alcance y los RF's comprometidos	Cálculos de horas disponibles del equipo. Estacionan Lineal (que tecnica usar)	Equipo de proyecto
Análisis y diseño	Diseño de la arquitectura	Diseño preliminar del sistema y requerimientos funcionales y no funcionales	Diagramas de arquitectura general y particular	Reuniones del equipo de proyecto, consultas con expertos	Equipo de proyecto
	Validar Arquitectura	Diagramas de arquitectura general y particular	Arquitectura validada	Reuniones del equipo de proyecto, consultas con expertos y con Stakeholders	Equipo de proyecto, Equi de Arquitectura del banco
	Diseño de la solución	Arquitectura validada, RF's y RNF's	Casos de Uso e historias de usuario?	Reuniones del equipo de proyecto y con Stakeholders	Equipo de proyecto
Construcción ML	Planificación	Tareas relevantes, fechas de evaluacion, diponibilidad de tiempo	Planificacion inicial	Reuniones, plan tentativo inicial	Equipo de proyecto
	Planificacion de releases	Plan inicial, RF's y RNF's	Plan de Releases	Reuniones de planificacion y coordinacion	Equipo de proyecto, Stakeholders
	Planificacion de sprints	Plan de Releases	Lsitado de tareas del sprint	Plan de releases, Jira	Equipo de proyecto
	Sprint review	Tareas realizadas, Tareas pendientes del sprint	Plan de releases ajustado con tareas que hayan quedado	Jira y reunion de coordinacion	Equipo de proyecto
	Registro de Bugs	Planilla de tareas, Bugs y Mejoras	Planilla con el ingreso de tareas pendientes y realizadas	Planilla de Tareas y herramientas de construccion	Equipo de proyecto
	Diseño de casos de prueba	Listado de funcionalidades construidas y RF's	Pruebas documentadas de cada caso	UnitTest, Django Unit Test, Planillas internas	Equipo de Proyecto
	Pruebas de usabilidad	Listado de funcionalidades construidas y lsitado de mejoras para usabilidad	Evaluacion de cada una de las pruebas de cada caso de uso	Planilla de control interno y documentacion final	Equipo de Proyecto y Analistas de ASI

Pruebas	Pruebas unitarias	Listado de Funciones Criticas	Test validados o no	UnitTest, Django Unit Test, Planillas internas	Equipo de Proyecto
	Pruebas Funcionales	Listado de funcionalidades construidas y RF's	Pruebas documentadas de cada caso	Aplicación construida	Equipo de Proyecto
	Pruebas de Performance	Ingreso de registros al sistema	Tiempos de respuesta para cada algoritmo	Aplicación construida y los registros de tiempo	Equipo de Proyecto
	Pruebas de Satisfacción	Encuestas a Analistas	Encuestas respondidas	Encuestas especificas y Mails	Analistas de ASI,Ingenieros de Arquitectura y Equipo de Proyecto
Proceso de gestión	Gestión de la documentación	Planillas, documentos y versiones actualizadas del software	Documentos actualizados en repositorios conocidos	Google Drive, OneDrive, Estructura de Directorios de documentacion	Equipo de Proyecto
	Registro de horas dedicadas al proyecto	carga de horas ingresadas por cada miembro	Planilla de control de horas	Planilla Google Drive	Equipo de Proyecto
	Gestión del Riesgo	Identifiaccion de riesgos iniciales y riesgos emergentes	Categorizacion de riesgos y medidas de mitigacion	Magerit, reuniones de equipo y de evaluacion periodica	Equipo de Proyecto

Tabla 9.2.6-1

9.3 Métricas

En esta sección mostraremos las métricas recopiladas durante todo el proyecto con la intención de poder medir la evolución y performance del equipo en cada una de las instancias.

Nuestro equipo consideró algunas de las métricas obtenidas como críticas, esas son las relativas a la exactitud de los algoritmos y modelos, en tanto que, por tratarse de una prueba de concepto, otras métricas inicialmente no eran de preocupación. Estas métricas se describen con profundidad durante el capítulo 6, en 6.2.3 Model Training and Evaluation

De todas maneras, se trabajó teniendo en mente esta posibilidad y aquí se presentan los resultados.

9.3.1 Usabilidad

En cuanto a los criterios de usabilidad, y los objetivos, propusimos una encuesta para determinar cómo se había comportado el sistema en términos de facilidad de uso y la información que el sistema brinda. Si bien los resultados son en ocasiones subjetivos, se obtuvo una buena evaluación, cuyo promedio es de 8+, en una escala del 1 al 10. Ver resultados de las encuestas en el ANEXO 6. Aquí se muestra el resultado de una de las entrevistas. Ver ilustración 9.3.1-1

Para evaluar la usabilidad tuvimos en cuenta sus dimensiones:

- Eficiencia
- Eficacia
- Satisfacción

Los resultados de las encuestas nos brindaron un panorama razonable en cuanto a la **Satisfacción** de los usuarios, así como también la **Eficiencia** en la que fueron dispuestos los objetos dentro de la página.

Encuesta: Rodrigo Mateos

Pregunta (del 1 a- 10, siendo 1 el minimo, 10 el maximo)	Respuesta
¿Es facil encontrar la opcion buscada?	7
¿La opcion seleccionada, cumple con su objetivo?	8
¿Se brinda informacion relevante de acuerdo al contexto del problema?	9
¿El funcionamiento del login, es el esperado?	9
¿Es sencillo acceder a crear un usuario?	7
¿Es sencillo bloquear/desbloquear un usuario?	9
¿Es sencillo cambiar la contraseña un usuario?	7
¿Califique la sencillez de subir un archivo para su analisis?	8
¿Es facil encontrarlo luego para su procesamiento?	7
¿El resultado del procesamiento, es claro? (archivo)	9
¿Califique la sencillez de analizar un texto en particular?	7
¿El resultado del procesamiento, es claro? (texto)	7

Calificación Promedio	8

Ilustración 9.3.1.-1

El promedio de los resultados de satisfacción para cada uno de los ítems consultados se puede ver en la Ilustración 9.3.1.-2

Pregunta (del 1 a- 10, siendo 1 el minimo, 10 el maximo)	Promedio
¿Es facil encontrar la opcion buscada?	7,8
¿La opcion seleccionada, cumple con su objetivo?	8,8
¿Se brinda informacion relevante de acuerdo al contexto del problema?	9,0
¿El funcionamiento del login, es el esperado?	8,8
¿Es sencillo acceder a crear un usuario?	8,3
¿Es sencillo bloquear/desbloquear un usuario?	7,8
¿Es sencillo cambiar la contraseña un usuario?	8,0
¿Califique la sencillez de subir un archivo para su analisis?	8,5
¿Es facil encontrarlo luego para su procesamiento?	8,3
¿El resultado del procesamiento, es claro? (archivo)	8,5
¿Califique la sencillez de analizar un texto en particular?	8,0
¿El resultado del procesamiento, es claro? (texto)	8,5

Ilustración 9.3.1.-2

Para reforzar lo mencionado, podríamos hacer notar que el menú lateral es transversal a todo el sitio, que permite al usuario una experiencia similar independientemente de donde se ubique.

Con esto se consigue tener un mínimo de clics para acceder a cada funcionalidad, permitiendo además generar modificaciones de manera granular en cada una de las opciones que deseemos agregar o quitar.

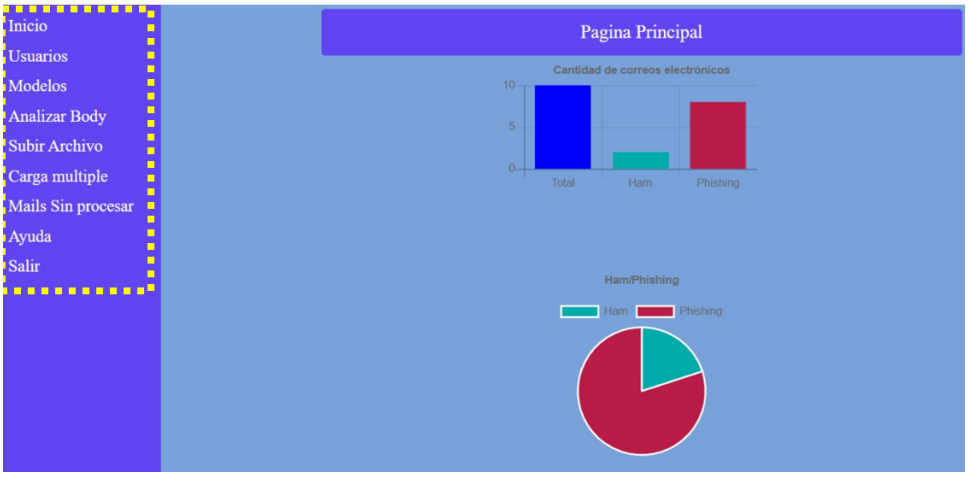


Ilustración 9.3.1-3 Pantalla principal, con foco en el menú lateral

Además, en aquellos casos que fue necesario, se implementó submenús, cuyo despliegue es automático, para minimizar la cantidad de clics, así también como filtros para las búsquedas de las tablas donde la cantidad de registros podría dificultar alguna búsqueda

Ilustración 9.3.1-4 Submenú implementado

Con esto hemos logrado reducir significativamente el promedio de clics necesarios para acceder a cada funcionalidad, teniendo un promedio de **1,2** clics por funcionalidad para

todo el sitio. Esto esta alineado con nuestro enfoque en cuanto a **Eficiencia**, pues no es costoso acceder a las funcionalidades del sistema. Aquellas funcionalidades que tienen una complejidad más alta fueron separadas en diferentes paginas para que el promedio (hasta la fecha) no fuese más de 2 clics. Ver ilustración 9.3.1-5

Funcionalidad		Cantidad de Clics para acceder	Campos
Login		1	2
Ir a dashboard principal		1	0
Ir Dar de alta un usuario		1	0
Dar de alta un usuario		1	7
Listado de usuarios		1	0
Modificar un Usuario		1	5
Analizar Un body		1	2
Subir un archivo de mail		1	4
Ver mails sin procesar		1	0
Procesar un mail	Ver lista mails sin procesar	1	0
	Seleccionar mail	2	
	Procesar, selección algorit.	2	
Acceso a ayuda		1	
Logout		1	
	Visualizar el mail y procesar	2	

Promedio	1,2

Ilustración 9.3.1-5

9.3.2 Gestión de Incidencias y Bugs

Como se puede ver en la Sección 9.2.3.3, se implementó una planilla de tareas, donde se toman nota de aquellos incidentes, mejoras y bugs reportados. Los mismos son tratados mediante el control de cambios (Ver Sección 10.1). En esta planilla se puede ver la fecha del reporte de la incidencia, y la fecha de finalización, de esta manera logramos obtener mediciones y promedios de las resoluciones de los incidentes.

Actualmente tenemos 40 incidentes registrados a lo largo de toda la fase de construcción (Ver Ilustración 9.3.2-1), con 12 incidentes sin resolver, de los cuales 2 están en progreso. De todos ellos, 7 fueron Bugs detectados.

Incidencias	
Iniciado	2
Creado	12
Terminado	26
Total	**40**
Bugs	**7**

Ilustración 9.3.2-1

El promedio de resolución de los incidentes es de casi 6 días, 5,8 para los incidentes en general, pero de 4,6 días para la resolución de los Bugs. Si bien se agrupó la información para simplificar, los defectos están catalogados en función de su criticidad, dato que puede ser relevante si consideramos en qué momento del proyecto es que surgen.

Promedio resolucion General	5.8
Promedio resolucion Bugs	4.6

Ilustración 9.3.2-2

Se ha observado una aparición de más incidentes (sobre todo Bugs) sobre el final de la etapa de construcción, y esto puede ser explicado por la necesidad de integrar diferentes componentes.

Esfuerzo en dias para resolucion	
Bugs	32
Todos	152

Ilustración 9.3.2-3

De la ilustración 9.3.2-3 se puede ver que la cantidad de días **acumulados** totales en la resolución de los incidentes, los Bugs implican el 20% del esfuerzo, siendo este un factor muy importante a resolver para reducir los tiempos promedio de solución. Cabe destacar que una característica de nuestro proyecto fue el esfuerzo dedicado a otras etapas concurrentemente, sobre todo en las etapas finales. Se podría implementar una herramienta o planilla que permitiese registrar el esfuerzo real en horas, a efectos de contabilizar de manera más eficiente el esfuerzo dedicado al retrabajo.

9.3.2 Pruebas

Para lograr obtener un producto que cumpla con las expectativas funcionales del cliente, se definieron y realizaron distintos tipos de pruebas:

- Pruebas Unitarias
- Pruebas Funcionales
- Pruebas de Performance
- Pruebas de Compatibilidad

El desarrollo de todas estas pruebas puede consultarse en el Capítulo 7

Conclusiones

En general, la prueba de concepto demostró con éxito la viabilidad de la idea del nuevo producto. Los requerimientos exigidos por el Cliente se cubren en su mayoría, y los que han sido observados son perfectamente ajustables en la infraestructura del banco. Por otro lado, los datos y la evidencia recabada en el análisis de los algoritmos/modelos utilizados evidencian que el mecanismo propuesto es válido. Ver resultados de pruebas en la Sección 6.2.3.

Sin embargo, para asegurar que el producto sea de calidad aceptable o de alta calidad, se hacen las siguientes recomendaciones:

- Recopilar continuamente los comentarios de los usuarios: Estos son fundamentales para identificar problemas de usabilidad en el producto. Para garantizar que el producto final satisfaga las necesidades de los usuarios, se recomienda que el equipo de desarrollo recopile continuamente estos comentarios y realice mejoras en el producto.

- Establecer un proceso formal para control de calidad: si bien el equipo pudo probar con éxito el producto, el proceso fue algo **ad hoc**. Sabemos que los proyectos de Prueba de Concepto brindan ciertas libertades en cuanto a los niveles de exigencia en torno a QA, es por ello por lo que se debería establecer un proceso que garantice la calidad, con roles y responsabilidades claramente definidos, esto puede ayudar a garantizar que el producto sea de calidad y cumpla con todos los requerimientos.

10. Gestión de la configuración

La gestión de la configuración de software es un proceso clave en el desarrollo y mantenimiento de cualquier sistema de software. Se trata de un conjunto de prácticas y herramientas que permiten controlar y gestionar los cambios en el software a lo largo de todo su ciclo de vida, desde la planificación y diseño hasta el mantenimiento y actualización.

Es esencial para garantizar la calidad del software, asegurando que se cumplan los requisitos del cliente y que se mantengan los estándares de calidad y seguridad. Además, permite trabajar de manera colaborativa y eficiente, manteniendo un registro de todas las versiones del software y facilitando la identificación y resolución de problemas.

Para llevar un control de las versiones del software, utilizamos el repositorio en la nube GitHub, donde mantenemos sincronizados los repositorios locales de cada desarrollador con el repositorio remoto. También usamos SourceTree como herramienta de sincronización entre repositorio local y repositorio remoto.

Desde la perspectiva de Sourcetree podemos ver el registro de actividades, un detalle de la fecha y hora del upload, y el comentario del commit al respecto del cambio o los cambios realizados. (Ver Ilustración 9-1: Gestión de la Configuración, SourceTree)

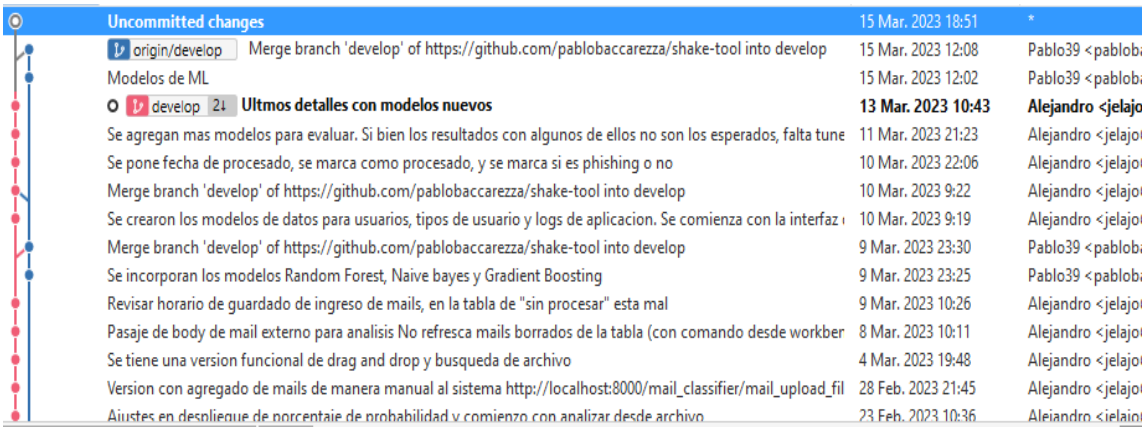


Ilustración 10-1: Gestión de la Configuración, SourceTree

Es importante en este punto que los registros de los comentarios sean claros y suficientes, de esa manera el equipo puede contar efectivamente con la versión adecuada a la hora de hacer el pull, o descarga. Al momento llevamos unos 70 commits, y ya hemos finalizado el sprint 4, habiendo llegado al hito del release 2.

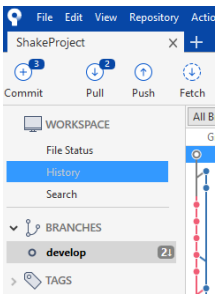


Ilustración 10-2: Gestión de la Configuración, Status

La herramienta notifica de los cambios en el repositorio local que deben ser impactados en el repositorio remoto, pero también nos avisa de diferencias en el sentido inverso. Tiene una gestión de diferencias que permite seleccionar cuáles cambios son los que se deben mantener, permitiendo elegir las modificaciones más recientes o convenientes. (Ver Ilustración 9-3: Gestión de la Configuración, Aviso de desfase de repositorios)

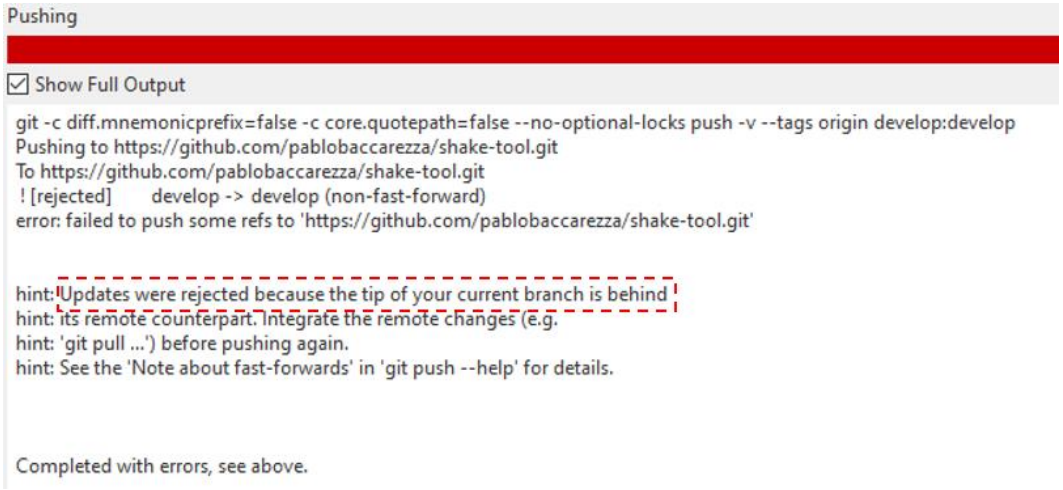


Ilustración 10-3: Gestión de la Configuración, Aviso de desfase de repositorios

De esta manera, mediante una secuencia de pasos predefinidos, se resuelve el conflicto o conflictos, y se puede proceder a hacer la subida de la versión del repositorio local.

(Ver Ilustración 9-4: Gestión de la Configuración, Desfase o conflicto resuelto)

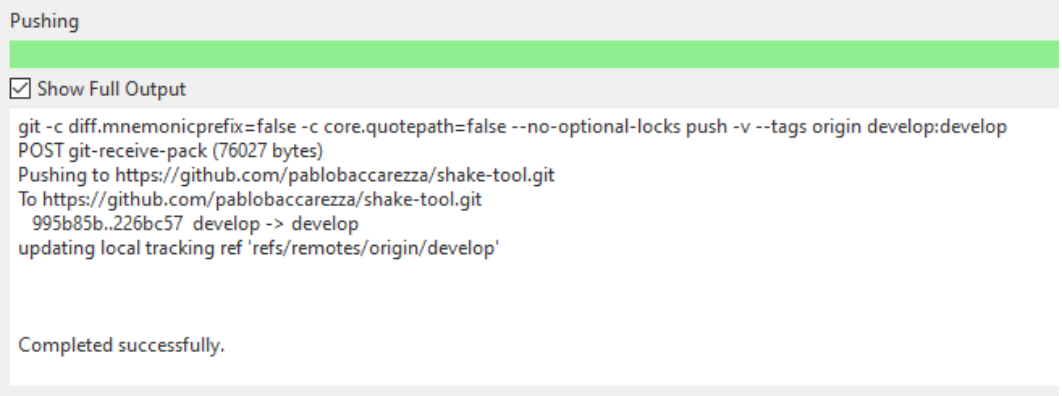


Ilustración 9-4: Gestión de la Configuración, Desfase o conflicto resuelto.

10.1 Gestión de Cambios

Durante la etapa de construcción se utilizó un proceso de control de cambios, sobre los elementos de configuración del software definidos anteriormente.

El procedimiento que utilizo fue simplificado, para poder responder rápidamente ante un cambio y lograr un consenso la conveniencia y oportunidad del mismo. Ver Ilustración 10.1-1 Proceso de Gestión de Cambios

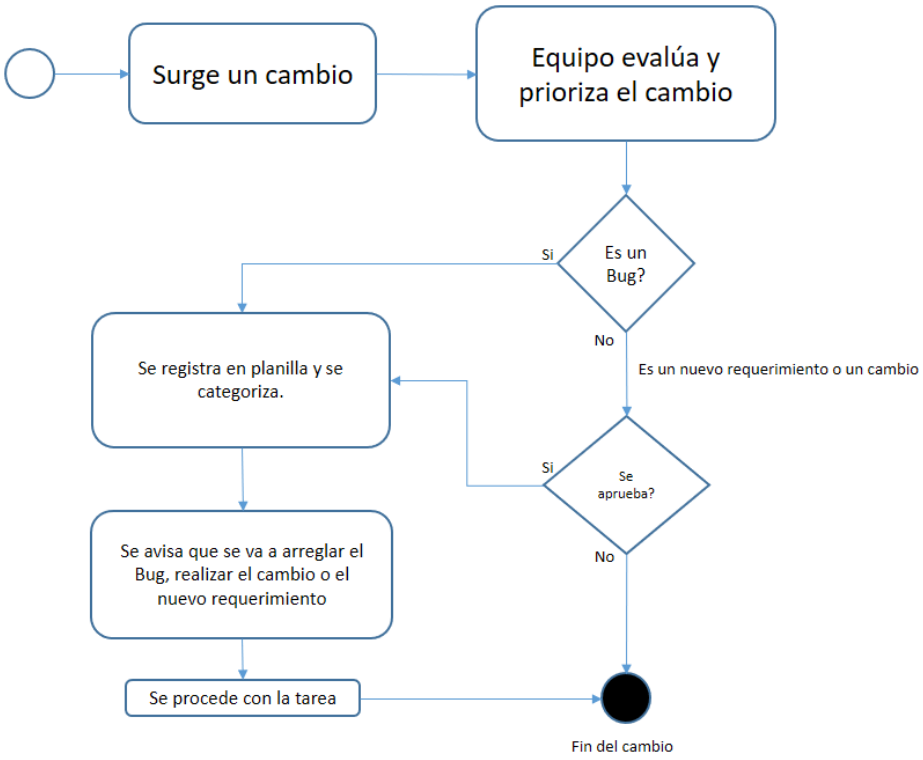


Ilustración 10.1-1 Proceso de Gestión de Cambios

El proceso comienza cuando un miembro del equipo detecta un error, el mismo ya se registra en la planilla de control (Ver Ilustración 9.1.3-2: Control de Tareas) y queda registro del día de descubrimiento, detalle del error, módulos y función relacionada. En el caso de una solicitud de cambio o nueva funcionalidad se evalúa y en caso de aprobarse en el equipo, se registra, con su categoría correspondiente, dejando los mismos datos, fecha, detalle, módulos y función (actual o nuevo). Los motivos por los que no se aprobaría serian

por cuestiones de alcance o conveniencia y para aquellas que no entren en este ciclo pero que tengan valor para futuro, se agregaran en la lista de tareas a futuro. Ver Capitulo 12 – Próximos Pasos

Cuando se finaliza una tarea “x”, se deja registro de la fecha de resolución, se cambia el estado de “Iniciado” a “Terminado” y se agrega comentario pertinente. Luego se vuelve a revisar la lista de tareas por resolver en búsqueda de ítems que tengan alta prioridad y que además estén relacionados con la instancia de construcción actual para estar alineados con los tiempos planificados de dicha etapa. En este caso, se notifica al equipo que se va a tomar la tarea, y se cambia el estado de “creado” a “iniciado”.

11. Conclusiones y lecciones aprendidas

11.1 Lecciones aprendidas

- Se logró construir una herramienta que avala el concepto que queríamos probar. Abre un sinfín de posibilidades de desarrollo y mejoras, de automatismos para facilitar aún más el objetivo. Tenemos claro cuál es el estado actual del desarrollo que hemos logrado, lo que resulta muy auspicioso en cuanto a las mejoras que se le pueden hacer, las expansiones, y las mejoras de diseño. También, como mencionaremos en la sección de Próximos Pasos (Ver Capitulo 12 – Próximos Pasos), las posibilidades de hacer llegar ese potencial a todos los usuarios del banco, y por qué no, ayudar a nuestros clientes brindando una interfaz para una comprobación de primera línea.

- Es crítico para el proyecto hacer un exhaustivo proceso de ingeniería de requerimientos, serio, detallado. Pero a pesar de lo profesional y dedicado de ese proceso, no es posible sustituir la interacción con los usuarios finales. Se tiene que mostrar la evolución, de manera periódica, pues la visión que ellos pueden aportar al uso de la herramienta es invaluable. como seres humanos, estamos sujetos al sesgo de nuestra percepción, pudiendo cometer el error de presentar un producto que no cumple con la satisfacción del cliente, poniendo gravemente en riesgo la aceptación del mismo.

- La planificación. Poder imaginar y plasmar formalmente cuál va a ser el camino a recorrer, marcando los hitos, los desafíos y los riesgos es un ejercicio no solo para tener una noción del trabajo al que nos vamos a enfrentar. La planificación nos marca los plazos y el esfuerzo para lograrlo, pero también es la vía de comunicación con el cliente para que sepa cómo vamos a trabajar y cuando esperar resultados.

- El aprendizaje en cuanto a las herramientas con las que se enfrenta un proyecto, tanto las de gestión como las propias herramientas de software. La elección de dichas herramientas y la comprensión de sus capacidades nos ayudan a comprender nuestras deficiencias, pudiendo mejorar para futuros proyectos.

- Logramos gestionar el proyecto en una organización pública, y tratar con profesionales altamente especializados en la rama de la seguridad informática, e interpretar correctamente sus necesidades. También interactuar con usuarios con mucho conocimiento de negocio y problemática diaria, pero con conocimientos técnicos escasos.

- Los proyectos de prueba de concepto son exploratorios, y pueden surgir nuevos requisitos a medida que el proyecto avanza. Es importante ser flexible y estar dispuesto a adaptarse a los cambios necesarios para alcanzar los objetivos del proyecto.

- Fue crucial definir el alcance del proyecto de prueba de concepto claramente y asegurarse de que se cumpla. Esto ayuda a evitar cambios constantes en los requisitos del proyecto que podrían retrasar el progreso y aumentar el costo del proyecto. También pone en conocimiento a los clientes de cuáles deben ser las expectativas y los motivos por los que se toman las decisiones a la hora de definir el alcance.

11.2 Conclusiones finales

Como se mencionó en el Capítulo 1, el objetivo principal de este proyecto es probar la viabilidad de utilizar Algoritmos de clasificación y Machine Learning para brindar una herramienta que ayude a los usuarios de correo electrónico del banco, a los analistas de seguridad de la información y a la gerencia del departamento de Seguridad de la información a la toma de decisiones en cuanto al tratamiento de los mails de contenido fraudulento. Cada uno de ellos con decisiones diferentes de acuerdo con su rol, pero sin dudas que, con un mismo foco, disminuir la cantidad de fraudes que actualmente se llevan a cabo mediante ese mecanismo.

Que el producto resultado sea una prueba de concepto, no significó dejar de lado los objetivos de calidad. Durante todo el proceso se han tenido en cuenta ciertas características que podrían asegurar o no la aceptación del desarrollo. En términos de arquitectura, el diseño permitiría escalar el sistema e incluir aquellos componentes que no fueron parte del alcance inicial. Las presentaciones de los prototipos han ayudado a entender realmente las necesidades, y a descubrir opciones que no habían sido previstas, lo que garantizan la eficiencia requerida por el usuario y la facilidad de uso lo que permitirá una rápida adopción por parte de los actores involucrados.

Durante este proceso, también se visualizó una posibilidad sumamente prometedora, la de aportar una interfaz donde nuestros clientes externos puedan probar los contenidos de correos que dicen venir del Banco, y que resulten sospechosos. Esto tiene una gran ventaja en términos de reputación, y de sensibilidad social, pero también en cuanto a la información que podemos recibir para nutrir nuestro sistema y mantenerlo actualizado, haciendo de cada una de esas interacciones un punto de entrada de información valiosa, algo de lo que hemos comentado como una flaqueza inicial (Ver Sección 1.7 Descripción de la solución), podríamos convertirlo en una gran oportunidad.

Comenzar un proyecto Machine Learning fue un gran desafío para este equipo, ya que contábamos con diferentes niveles de experiencia previa en este tema y las tecnologías

involucradas. Somos estudiantes de generaciones donde estos conceptos no estaban incluidos en los planes de estudio, por lo que fue de mucho valor que en las subsiguientes modificaciones se hayan incluido materias opcionales que permitieron tener conocimiento al menos a uno de los integrantes, y que a la postre fue la génesis de la idea.

Al comenzar el proyecto, el equipo se encontró con la dificultad de conocer la realidad parcialmente de Centro de Contacto, pero si el pleno conocimiento del funcionamiento de los procesos internos del equipo de Seguridad de la Información.

Los procedimientos, los formalismos, pero sobre todo los tiempos de respuesta a incidentes verdaderos lejos están de ser los ideales, y muchas veces por fuera del alcance de los profesionales que atienden cada uno de los casos.

Se tiene mucha confianza en que podemos entregar una herramienta que supere las expectativas, lo que podrá ser comprobado después de un proceso de aceptación, implementación y uso, para un posterior análisis de los datos recopilados.

Además, se es muy optimista en cuanto a las posibilidades que pueden ser tratadas a futuro, en caso de poder continuar el desarrollo.

Queda por delante, por parte de las Autoridades de Tecnologías de la Información y del Área de Seguridad de la Información, determinar recursos para asegurar esa continuidad en el desarrollo, la implantación tecnológica necesaria y definir los responsables técnicos que darán soporte y mantenimiento evolutivo. Si bien la organización cuenta en estos momentos con equipos que están incursionando en las técnicas de AI, y Machine Learning, lejos se está de lograr una sinergia para poder contar con conocimiento unificado y aún más lejos de contar con equipos de especialistas en estas materias dentro del banco. Creemos que nuestro proyecto puede

ser un gran catalizador de esa idea, pero, sobre todo, de gran ayuda en la reducción de los fraudes vía correos con contenido fraudulento.

12. Próximos pasos

En esta sección comentaremos algunas de las mejoras que pensamos que se pueden implementar a futuro y que podrían ser de mucho valor para aumentar las capacidades de la prueba de concepto.

Habitualmente, después de una prueba de concepto en el desarrollo de software, el siguiente paso suele ser el desarrollo de un prototipo funcional. Un prototipo funcional es una versión preliminar del software que permite a los desarrolladores probar la viabilidad técnica y la funcionalidad del software en un entorno más realista. En nuestro caso, el prototipo funcional ya está desarrollado, aunque con limitaciones, por lo que resta continuar iterando y mejorando el software para asegurarse de que cumpla con todos los requisitos y objetivos del proyecto. Esto puede implicar la realización de pruebas de usuario, la identificación y corrección de errores, la implementación de nuevas características y funcionalidades, y la optimización del rendimiento y la escalabilidad del software.

Dicho esto, podemos pensar que las siguientes serían opciones interesantes para continuar con el desarrollo:

- Autenticación y Autorización mediante Active Directory (standard BROU)
- Desarrollar el procesamiento de mails con imágenes, mediante OCR
- Agregar análisis de MetaData y más Algoritmos
- Desarrollo de componentes ShakePlugin y ShakeProxy
- Lograr una cobertura de pruebas unitarias suficiente

- Desarrollo de nuevos dashboards y reportes a partir de la información recopilada
- Explorar la posibilidad de extender a los clientes parte de la funcionalidad.

12.1 Autenticación y Autorización mediante Active Directory

Una funcionalidad que tendrá un apoyo muy grande por parte de las diferentes áreas operativas del Banco, pues simplifica mucho a nivel administrativo en cuanto a la centralización de objetos usuario y objeto grupo. Esto también proporciona autenticación de usuarios para acceder a los recursos de la red, lo que ayuda a garantizar que solo los usuarios autorizados puedan acceder a los recursos. Permite crear políticas de grupo que se pueden aplicar a usuarios y equipos en la red, lo que facilita la implementación y gestión de políticas de seguridad y configuración en la red.

En resumen, Active Directory es una herramienta muy útil para la gestión de identidad y acceso en entornos de red de Microsoft, y proporciona una serie de ventajas importantes para la administración y seguridad de la red.

12.2 Desarrollar el procesamiento de mails con imágenes, OCR

OCR (Optical Character Recognition) es una tecnología que permite la detección y reconocimiento de caracteres de texto en imágenes o documentos escaneados. Permite la automatización del proceso de detección de texto, lo que reduce significativamente el tiempo y el costo de la detección manual de texto, pero sobre todo permite la digitalización y la indexación del texto, lo que facilita la búsqueda y recuperación de información en documentos y archivos de texto.

En ciertas ocasiones la estrategia de los ciber delincuentes consta de copiar una notificación oficial del Banco, sobre todo las más novedosas, las que contienen noticias recientes o campañas recientes, donde capturan esa imagen y simplemente le adjuntan un link a su sitio malicioso a dicha imagen. Las posibilidades de manipular el comportamiento del destino de ese link son infinitas, desde redirigirlo a un sitio fraudulento con un look and feel similar al del Banco para intentar robar sus datos, como descargar software malicioso, etc.

El manejo de OCR nos daría la potencia de detectar el texto dentro de la imagen, para poder analizarlo con NLP, como hemos hecho en esta prueba de concepto, para determinar si es un caso de phishing, pero también podríamos determinar mediante el análisis de Metadata cuál es el destino del link de la imagen. Esto será mencionado en la subsección siguiente.

12.3 Agregar análisis de MetaData y más Algoritmos

Como hemos visto en la sección 6.2.2.1, la programación para obtener la MetaData del mail cuando este es procesado es una tarea para nada trivial. En nuestro caso se aprovechó la oportunidad de dejar la programación para la obtención de la Metadata de cada mail ya disponible, pensando en la futura implementación de un mecanismo alternativo, o mejor aún, la sumatoria de las dos estrategias de análisis, tanto Procesamiento del Lenguaje Natural como análisis de MetaData. En lugar de utilizar solo una de las técnicas, utilizar ambas, ya sea en serie o en paralelo.

Cuando hablamos de análisis de MetaData nos referimos al conjunto de patrones que Machine Learning puede analizar cuando cuenta con una cantidad de datos aceptables al respecto de los emails con los que se cuenta. Por ejemplo, si en un mail aparecen palabras como “Bloqueo”, “Cuenta”, “Validar”, pueden alimentar una matriz de datos, entre

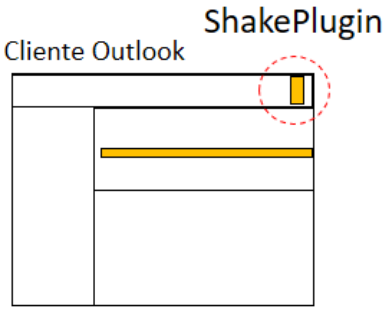
muchos otros campos de dicha matriz, para luego ser analizado por ML. Dicha matriz contempla aproximadamente 150 factores en su versión más exhaustiva, con lo que representa un gran desafío y una gran oportunidad de mejora si se decide recorrer ese camino. Es por ello que hemos decidido dedicar el esfuerzo solamente a procesamiento de Lenguaje Natural, pues no había manera de lograr en tiempo este proyecto implementando ambas estrategias, y que contábamos con experiencia previa y podíamos optimizar el tiempo, algo sumamente valioso en proyectos de corta duración.

12.4 Desarrollo de componentes ShakePlugin y ShakeProxy

Como se mencionó en el capítulo 3 sección 5, el alcance de esta prueba de concepto tuvo que ser ajustado en un par de oportunidades, inmediatamente después del proceso de ingeniería de requerimientos. (ver Ilustración 3.5-1 Alcance propuesto para la prueba de concepto) En este punto es donde se tiene una noción del sistema en su completitud, pudiendo entender el esfuerzo que implica su construcción, pero también sabiendo con los recursos con los que se cuenta. Dicho esto, el sistema completo cuenta también con dos (2) componentes importantísimos que podrían ser desarrollados e incluidos en el futuro, ShakeProxy y ShakePlugin.

ShakePlugin habla de un desarrollo específico para Microsoft Outlook, cuya función principal es brindar ayuda a los usuarios de correo mediante el análisis local de los correos.

Este análisis se basa en una correcta distribución de los últimos modelos generados por ShakeSystem, para que la capacidad de procesamiento vaya a la primera línea. A su vez, permite tener un canal de comunicación con los clientes, fortaleciendo lo que es la flaqueza más grande que explota el Phishing, la falta de conocimientos.



Cliente:
- Reglas actualizadas enviadas por servidor
- Capacidad de pre procesamiento predictivo para nuevos casos
- Configura el Proxy
- Suspende temporalmente análisis
- Marca visualmente los ítems con análisis positivo
- Despliega información para educar al cliente

Ilustraciones 12.4-1, 12.4.2 Extractos de la propuesta inicial y algunas de sus características

ShakeProxy se trata de un servidor intermedio, cuyas funciones principales serían:

•Interfaz de recepción de mails sospechosos enviados x cliente

•Unifica comunicación con el servidor mediante standard

•Habilita la posibilidad de incorporar otros clientes

•Sirve de pasarela y caché de nuevas actualizaciones

•Capacidad de procesamiento offline con últimos modelos a demanda

•Comunicaciones mediante API's

•Lista de clientes del proxy

Además de estas funciones descritas, tiene como agregado la segregación de la red en términos de seguridad informática, ajustándose al estándar de segregación por segmentos de red dependiendo de las tareas y riesgo de cada segmento. Los clientes de una subred solo hablan con 1 servidor proxy, y el proxy con el servidor principal. esto evita un costo

administrativo de mantenimiento muy grande en cuanto a las reglas de firewall que hay que mantener, entre otros beneficios.

12.5 Lograr una cobertura de pruebas unitarias suficiente

La cobertura de pruebas se refiere a la cantidad y calidad de pruebas que debemos realizar en el software para garantizar que se han verificado todas las funcionalidades y características importantes.

La cobertura de pruebas suficiente es importante para garantizar la calidad del software y reducir los errores y problemas que puedan surgir después de su lanzamiento.

Algunos de sus beneficios son:

1. Mejora la calidad del software: La realización de pruebas exhaustivas en un software puede ayudar a identificar y corregir errores y defectos, lo que mejora la calidad del software y reduce el riesgo de fallos en el futuro.

2. Reducción de costos: La detección temprana de errores y defectos en el software puede reducir significativamente los costos asociados con la corrección de estos errores después del lanzamiento del software.

3. Incremento de la confianza: La realización de pruebas adecuadas puede aumentar la confianza en la calidad del software por parte de los usuarios, lo que puede llevar a una mayor adopción y satisfacción del cliente.

4. Cumplimiento de los requisitos: La cobertura de pruebas suficiente puede garantizar que se han probado todas las funcionalidades y características importantes del software para garantizar que cumple con todos los requisitos especificados.

Dentro del contexto de nuestra prueba de concepto, y como ya se ha mencionado previamente, la generación de pruebas exhaustivas no estaba comprendido dentro de nuestro camino crítico. El foco de nuestro desarrollo estaba en validar la idea de que era

posible detectar emails de contenido fraudulento mediante Machine Learning, y dicho objetivo se cumplió. Esto no significa que no se tenga en cuenta como un pendiente de esta primera etapa, por eso está aquí mencionado y creemos que amerita el esfuerzo, como parte del compromiso asumido según la continuidad del proyecto a futuro.

12.6 Desarrollo de nuevos dashboards y reportes

Es evidente que las decisiones deben ser tomadas basadas en información. En la medida que se mejoren y se recopilen métricas del funcionamiento, mejor preparados para tomar decisiones. Entonces el desarrollo de una capacidad de consumir esos datos, y de que estén disponibles cuando sean necesarios parece ser una característica muy interesante a desarrollar.

Por ello, la creación de pantallas de despliegue que se ajusten a las necesidades de cada uno de los interesados, y la generación de reportes con información suficiente y relevante forma parte de aquellas tareas que son ineludibles.

Solo podemos imaginarnos el potencial del consumo de esos datos, pues la evidencia y el devenir del tiempo podrán demostrar la validez del sistema y la información que pueda brindar.

12.7 Explorar la posibilidad de extender a los clientes…

Una de las ideas que habíamos tenido desde un principio, era la de ayudar a nuestros clientes, a los usuarios de servicios del Banco.

Entendemos que debemos recorrer un camino, y que comienza dentro del banco. Nosotros tenemos herramientas, conocimientos y equipos especializados para enfrentar esta problemática. Los usuarios del Banco NO, y han sido “empujados” a la bancarización obligatoria, a la digitalización en múltiples aspectos de su vida cotidiana, pero no fueron formados para ello. Entonces, las condiciones están dadas para que sufran estafas, ellos dudan, no saben cómo resolver ciertos desafíos cuando se trata de interacciones con el mundo digital. Esta diferencia se hace más evidente según las franjas de edad sean más avanzadas.

Queda en evidencia, en los noticieros, en los diarios y en las radios la secuencia de estafas de las que son víctimas, y por ser correos que dicen venir de nuestra institución, se asume cierta responsabilidad.

Es por ello que pensamos en una implementación especial, utilizando los mismos mecanismos, pero con la finalidad de proveer de una interfaz donde los usuarios puedan probar por sus propios medios aquellos correos que dicen venir del banco, para de alguna manera, ayudarlos en ese aspecto. Esto tiene como beneficio adicional, la posibilidad de capturar ese contenido y utilizarlo en nuestro beneficio, nutriendo al sistema con los propios usuarios, y eventualmente con aquellos que también pretendan tener malas intenciones; solo que, en ese caso, también sabremos cuál será la estrategia.

Si bien exponer este servicio conlleva un riesgo, podría ser interesante analizar el costo beneficio, y cuanto es que está en juego en términos de reputación. podría ser un gran apoyo y dar las señales correctas de que el banco está de su lado.

Referencias bibliográficas

[1] I. Foulds, “Active Directory”, ago. 2022, [En línea]. Disponible en: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview

[2] Google Cloud, “Inteligencia Artificial”, [En línea]. Disponible en: https://cloud.google.com/learn/what-is-artificial-intelligence?hl=es-419

[3] Scikit Learn, “Bagging Classifier”, [En línea]. Disponible en: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html

[4] “Correo Electrónico”, abr. 2001, [En línea]. Disponible en: https://www.rfc-editor.org/rfc/rfc2822#section-2

[5] INCIBE, “Ciber Delincuente”, [En línea]. Disponible en: https://www.incibe.es/aprendeciberseguridad/hacker-vs-ciberdelincuente#:~:text=El%20ciberdelincuente%20es%20la%20persona,ingenier%C3%ADa%20social%20o%20el%20malware.

[6] Django,”Django”, [En línea]. Disponible en: https://www.djangoproject.com/

[7] Scikit Learn, “GradientBoostingClassifier”, [En línea]. Disponible en: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

[8] B. Chiradeep, «Top 10 Anti-Phishing Software in 2021», ago. 2021, [En línea]. Disponible en: https://www.spiceworks.com/it-security/vulnerability-management/articles/top-10-anti-phishing-software/

[9] Cofense, “Cofense”, 2021, [En línea]. Disponible en: https://cofense.com/

[10] Greathorn, “Greathorn”, 2021, [En línea]. Disponible en: https://www.greathorn.com/

[11] ironscales, “ironscales”, 2021, [En línea]. Disponible en: https://ironscales.com/

[12] R. Ranawana y A. Karunananda, “An Agile Software Development Life Cycle Model for Machine Learning Application Development”, 2021, [En línea]. Disponible en: https://ieeexplore.ieee.org/document/9664736

[13] R. S. Pressman, *Ingeniería del software*, 7a. 2010. México: M. G. Hill. Ch 3, Ch 5-9, Ch 14

[14] I. Sommerville, *Ingeniería de software*, 9a. México: Add. Wesley 2011. Ch 3, Ch 4, Ch 5 y Ch 6

[15] W. J. Murdoch, “Definitions, methods, and applications in interpretable machine learning”, oct. 2019, [En línea]. Disponible en: https://www.pnas.org/doi/full/10.1073/pnas.1900654116

[16] D. Ping, “*The Machine Learning Solutions Architect Handbook – Create machine learning platforms to run solutions in an enterprise setting*.” UK: Packt Publishing 2022.

[17] G. Giray, “A Software Engineering Perspective on Engineering Machine Learning Systems: State of the Art and Challenges”, vol. abs/2012.07919, 2020.

[18] S. Amershi *et al.*, “Software Engineering for Machine Learning: A Case Study”, Canada: IEEE 05, 2019.

[19] Universidad ORT, “Machine Learning para sistemas inteligentes”, 2022.

[20] Andronicus, “Classification of Phishing Email Using Random Forest Machine Learning Technique”, 2014.

[21] Magerit, ”Magerit - versión 3.0. Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información”.

Anexos

Anexo 1

Formulario respuestas Alejandro Cao

	Muy de acuerdo		NsNc		Nada de acuerdo
Contexto	5	4	3	2	1
¿Fue presentado adecuadamente el contexto del problema que se pretende ayudar a resolver?	◉	○	○	○	○
¿Entiende que es un tema quedebe ser abordado?	○	◉	○	○	○
¿El probelma del Phishing, vino para quedarse?	○	◉	○	○	○
¿Afecta la reputacion del banco esta problemática?	◉	○	○	○	○
¿Afecta los usuarios internos y externos del banco?	◉	○	○	○	○
¿Puede ser valioso para la Gerencia y los analistas de seguridad de la informacion?	○	◉	○	○	○
¿Puede ser valioso para los usuarios finales?	○	◉	○	○	○
Los usuarios objetivo, ¿cuentan con los conocimentos adecuados para enfrentarse al Phishing?	○	○	○	◉	○

	Muy de acuerdo		NsNc		Nada de acuerdo
Solucion					
¿Le parece oportuna la implementacion de una herramienta que apoye este tema?	◉	○	○	○	○
¿Las tecnologias presentadas, parecen adecuadas?	○	◉	○	○	○
¿Cree que Machine Learning sea de ayuda para este contexto?	◉	○	○	○	○

Usabilidad					
¿Facil de usar la version pre-release 1 presentada?	○	◉	○	○	○
¿Tiene potencial de ser util?	◉	○	○	○	○
¿La estetica general es adecuada?	○	◉	○	○	○
¿La herramienta podria ser util para los Analistas de Seguridad?	◉	○	○	○	○
¿La herramienta podria ser util para los compañeros de Centro de Contacto?	◉	○	○	○	○
¿La herramienta podria ser util para los Usuarios del banco?	○	◉	○	○	○
¿La herramienta podria ser util para los Usuarios finales?	○	○	◉	○	○

Notas

Se toma nota de las propuestas de casos de uso y de diseño de interfaz propuestas de acuerdo a la reunion mantenida el viernes 10 de marzo del 2023

Formulario de respuestas de Rodrigo Mateos

	Muy de acuerdo		NsNc		Nada de acuerdo
Contexto	5	4	3	2	1
¿Fue presentado adecuadamente el contexto del problema que se pretende ayudar a resolver?	◉	○	○	○	○
¿Entiende que es un tema quedebe ser abordado?	◉	○	○	○	○
¿El probelma del Phishing, vino para quedarse?	◉	○	○	○	○
¿Afecta la reputacion del banco esta problemática?	○	◉	○	○	○
¿Afecta los usuarios internos y externos del banco?	◉	○	○	○	○
¿Puede ser valioso para la Gerencia y los analistas de seguridad de la informacion?	◉	○	○	○	○
¿Puede ser valioso para los usuarios finales?	◉	○	○	○	○
Los usuarios objetivo, ¿cuentan con los conocimentos adecuados para enfrentarse al Phishing?	○	◉	○	○	○

	Muy de acuerdo		NsNc		Nada de acuerdo
Solucion					
¿Le parece oportuna la implementacion de una herramienta que apoye este tema?	○	○	○	○	○
¿Las tecnologias presentadas, parecen adecuadas?	◉	○	○	○	○
¿Cree que Machine Learning sea de ayuda para este contexto?	○	◉	○	○	○

Usabilidad					
¿Facil de usar la version pre-release 1 presentada?	○	◉	○	○	○
¿Tiene potencial de ser util?	◉	○	○	○	○
¿La estetica general es adecuada?	○	◉	○	○	○
¿La herramienta podria ser util para los Analistas de Seguridad?	◉	○	○	○	○
¿La herramienta podria ser util para los compañeros de Centro de Contacto?	○	◉	○	○	○
¿La herramienta podria ser util para los Usuarios del banco?	○	◉	○	○	○
¿La herramienta podria ser util para los Usuarios finales?	◉	○	○	○	○

Notas
Se toma nota de las propuestas de casos de uso y de diseño de interfaz propuestas de acuerdo a la reunion mantenida el viernes 10 de marzo del 2023

Formulario respuestas Álvaro Lousteau

	Muy de acuerdo		NsNc		Nada de acuerdo
Contexto	5	4	3	2	1
¿Fue presentado adecuadamente el contexto del problema que se pretende ayudar a resolver?	◉	○	○	○	○
¿Entiende que es un tema quedebe ser abordado?	○	◉	○	○	○
¿El probelma del Phishing, vino para quedarse?	◉	○	○	○	○
¿Afecta la reputacion del banco esta problemática?	◉	○	○	○	○
¿Afecta los usuarios internos y externos del banco?	◉	○	○	○	○
¿Puede ser valioso para la Gerencia y los analistas de seguridad de la informacion?	◉	○	○	○	○
¿Puede ser valioso para los usuarios finales?	○	◉	○	○	○
Los usuarios objetivo, ¿cuentan con los conocimentos adecuados para enfrentarse al Phishing?	○	◉	○	○	○

	Muy de acuerdo		NsNc		Nada de acuerdo
Solucion					
¿Le parece oportuna la implementacion de una herramienta que apoye este tema?	○	◉	○	○	○
¿Las tecnologias presentadas, parecen adecuadas?	○	◉	○	○	○
¿Cree que Machine Learning sea de ayuda para este contexto?	◉	○	○	○	○

	Muy de acuerdo		NsNc		Nada de acuerdo
Arquitectura					
¿La selección del framework para diseñar la arquitectura, fue adecuada? C4	◉	○	○	○	○
Los diagramas presentados, ¿fueron claros?	◉	○	○	○	○
Los diagramas, ¿Abordan la problemática de manera adecuada?	○	◉	○	○	○
¿Se ajustan a la infraestructura que posee el banco?	○	◉	○	○	○
¿Se ajustan a los lineamientos que el banco tiene en cuanto a los standards?	○	○	○	○	○

Notas
En la entrevisata personal, se toma nota de la recomendación en cuanto a notificar via mail (SMTP) a los analistas cuando un caso es detectado como positivo, para poner en conocimiento de manera inmediata a los especialistas.

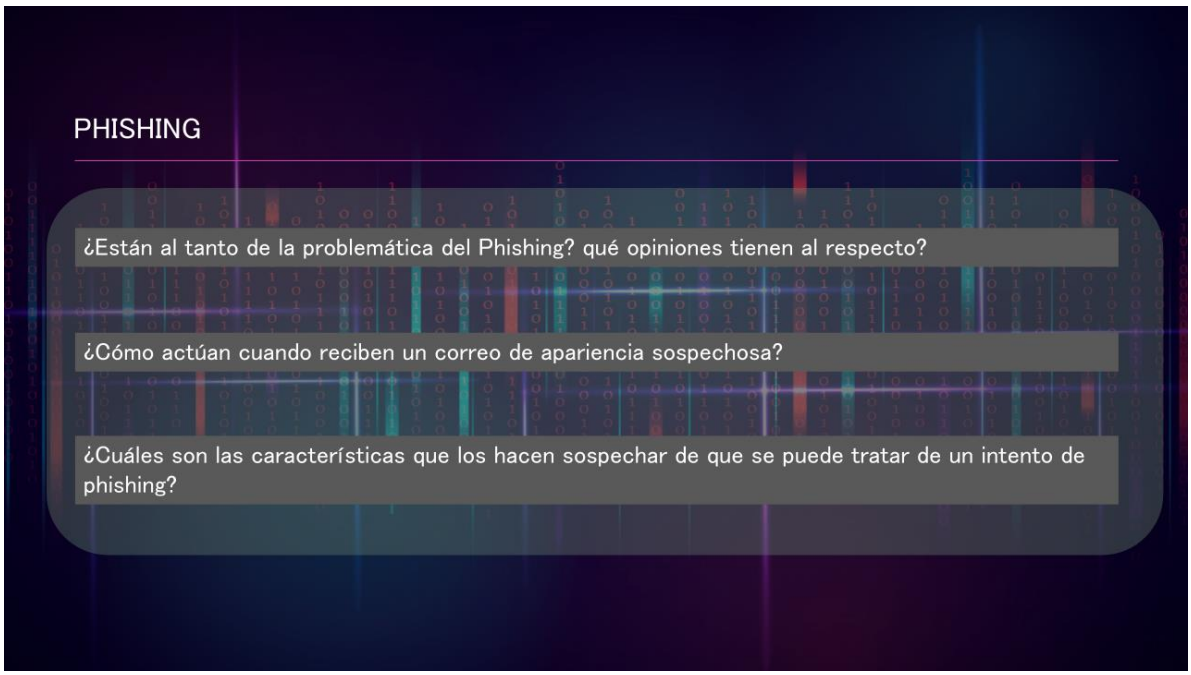
Anexo 2

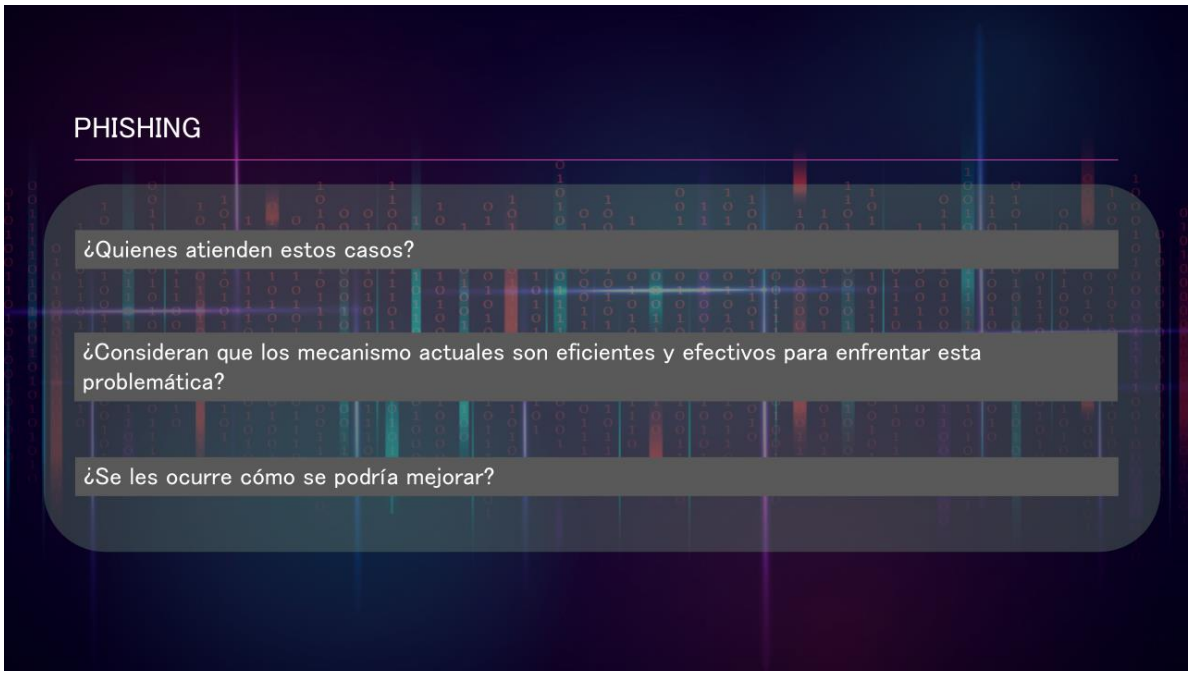
Para realizar las entrevistas durante el relevamiento de requerimientos preparamos una presentación que nos sirvió como guía para las entrevistas, la entrevista tuvo una parte guiada por preguntas y un espacio libre para que los participantes pudieran expresar libremente su punto de vista sobre esta temática:

Contexto

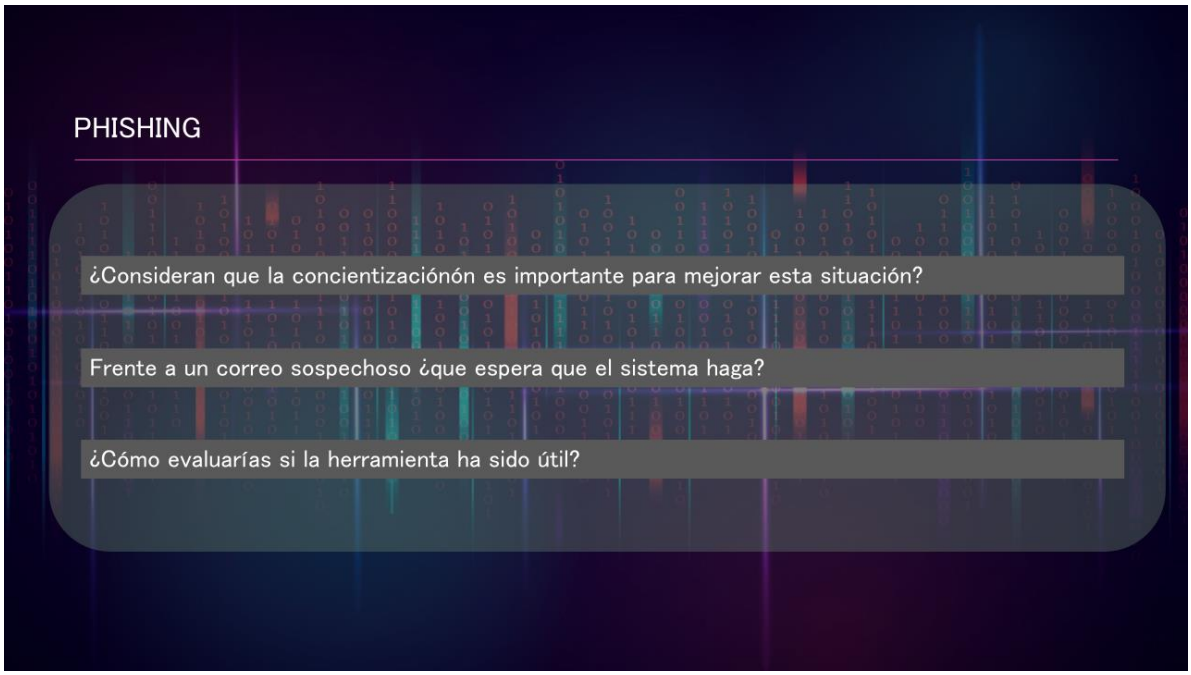
Este es un trabajo académico solicitado como requisito para aprobar el proyecto de fin de carrera de Licenciado en Sistemas

¿Qué es un correo con un intento de phishing?


PHISHING
¿Están al tanto de la problemática del Phishing? qué opiniones tienen al respecto?
¿Cómo actúan cuando reciben un correo de apariencia sospechosa?
¿Cuáles son las características que los hacen sospechar de que se puede tratar de un intento de phishing?


PHISHING
¿Quienes atienden estos casos?
¿Consideran que los mecanismo actuales son eficientes y efectivos para enfrentar esta problemática?
¿Se les ocurre cómo se podría mejorar?


BRAINSTORMING
¿Se les ocurre? ¿cómo se podría mejorar?


PHISHING
¿Consideran que la concientizaciónón es importante para mejorar esta situación?
Frente a un correo sospechoso ¿que espera que el sistema haga?
¿Cómo evaluarías si la herramienta ha sido útil?

Casos reales

-------- Mensaje original --------
Asunto: Verifique sus Productos Banco Republica
De: "Seguridad Banco Republica (BROU)"
Para:
CC:

BANCO REPUBLICA

Su cuenta y llave digital deben ser verificados o seran bloqueados.

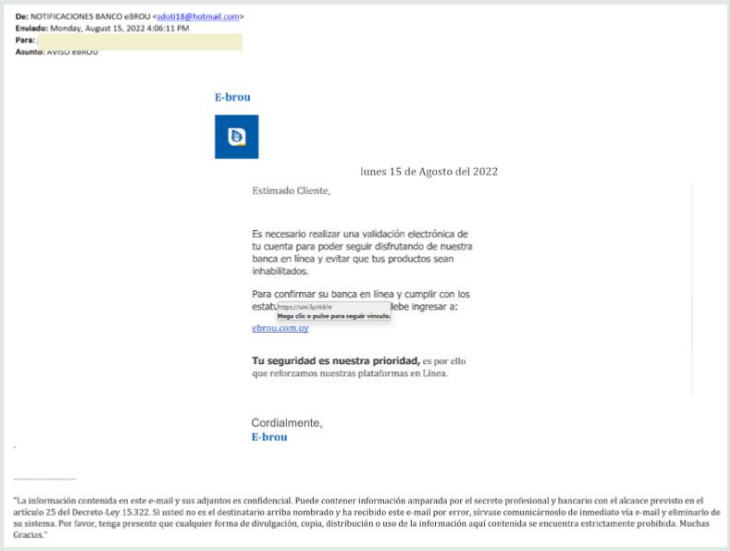
[-EMAIL-],

El Equipo de Seguridad Banco Republica (BROU) indica que su cuenta ha tenido una actividad sospechosa, por lo que procederemos a bloquearla si no la verifica.
El Banco Republica (BROU) le solicita que verifique su cuenta y llave digital correctamente, de lo contrario su cuenta sera bloqueada y debera dirigirse a una de nuestras sucursales para solventar el problema.
A partir del envio de esta EMAIL, usted tiene de 24 a 72 horas para validar su cuenta, de lo contrario sera bloqueada.

https://li1.su/g4hzu
Haga clic o pulse para seguir vinculo.
VERIFIQUE INMEDIATAMENTE

Sincerely,
2022 ?.All rights reserved.

Casos reales



De: NOTIFICACIONES BANCO eBROU <sdoti18@hotmail.com>
Enviado: Monday, August 15, 2022 4:06:11 PM
Para:
Asunto: AVISO EBROU

E-brou

lunes 15 de Agosto del 2022

Estimado Cliente,

Es necesario realizar una validación electrónica de tu cuenta para poder seguir disfrutando de nuestra banca en línea y evitar que tus productos sean inhabilitados.

Para confirmar su banca en línea y cumplir con los estat[illegible]ebe ingresar a:

https://uni.la/nkkrv
Haga clic o pulse para seguir vinculo.

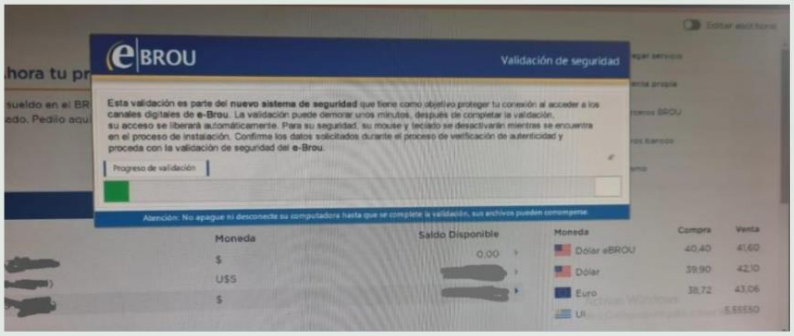
ebrou.com.uy

Tu seguridad es nuestra prioridad, es por ello que reforzamos nuestras plataformas en Línea.

Cordialmente,
E-brou

"La información contenida en este e-mail y sus adjuntos es confidencial. Puede contener información amparada por el secreto profesional y bancario con el alcance previsto en el artículo 25 del Decreto-Ley 15.322. Si usted no es el destinatario arriba nombrado y ha recibido este e-mail por error, sírvase comunicárnoslo de inmediato vía e-mail y eliminarlo de su sistema. Por favor, tenga presente que cualquier forma de divulgación, copia, distribución o uso de la información aquí contenida se encuentra estrictamente prohibida. Muchas Gracias."

Casos reales



Entrevista a usuarios finales.

Modalidad: virtual
Entrevistado: Cr. Miguel López
Cargo: Analista Financiero
¿Qué es un correo con un intento de phishing? Llega un correo electrónico que simula una página habitual, luego cuando entras te pide algún dato. Se muestran casos reales **¿Están al tanto de la problemática del Phishing? ¿Qué opiniones tienen al respecto?** Si lo explicó **¿Cómo actúan cuando reciben un correo de apariencia sospechosa?** Lo borro **¿Cuáles son las características que los hacen sospechar de que se puede tratar de un intento de phishing?** Me llamaría la atención si un Banco pide información confidencial, si tiene un link más aún. (Se fija en el cuerpo del correo) Busco antecedentes en Google poniendo frases del texto de correo para detectar si es un caso conocido de estafa (posible detección por lenguaje natural) **¿Quién atiende estos casos?** La gente de TI Aclaramos que es SI, explicamos el mecanismo, ¡no es conocido por todos!

Sugiere colocarlo en Intrabrou, “Si usted recibe un correo de phishing reportarlo…”

Sugiere capacitación, avisos recordatorios de que el Banco nunca le va a pedir datos confidenciales.

¿Consideran que los mecanismos actuales son eficientes y efectivos para enfrentar esta problemática?

¿Se les ocurre cómo se podría mejorar?

Estar avisados de no poner datos personales en ningún enlace, concientización, una campaña de sensibilización/información al respecto, hay que actualizar la campaña porque esto evoluciona

Se sugiere la posibilidad de que el sistema proporcione un aviso a los usuarios dando aviso de que el correo posee elementos que lo convierten en una estafa

Frente a un correo sospechoso ¿qué espera que el sistema haga?

Se sugiere la incorporación de un botón en Outlook que permita al usuario reportar un caso de phishing, permite que el sistema sea dinámico retroalimentando el algoritmo de ML

¿Cómo evaluarías si la herramienta ha sido útil?

El usuario opina que sería bueno definir una métrica, del tipo performance y que vaya evolucionando con el tiempo.

Modalidad: virtual
Entrevistada: Ticiana Martínez
Cargo: - Ejecutiva de negocio asistente en Sucursal Rocha
¿Qué es un correo con un intento de phishing? Recibir un correo que tiene un link que lo abro y quedan cosas en mi computadora o que puede llevarme a una página que no es verdadera. Se muestran casos reales **¿Están al tanto de la problemática del Phishing? ¿Qué opiniones tienen al respecto?** La gente se vulnera mucho. Tenemos que hacer una autocrítica que estamos enviando a trabajar con estas herramientas a mucha gente que no está preparada o capacitada, y cae en estos engaños porque no se da cuenta o no sabe. Hay una falta de información muy importante. No solo con eBROU también con las tarjetas, La gente tiene mucha confianza y comparte mucho la información. **¿Cómo actúan cuando reciben un correo de apariencia sospechosa?** Cuando recibo un correo sospechoso, en Outlook hago clic derecho y bloqueo al remitente. Me han llegado varios de estos correos. (Se comenta cuál es el procedimiento para reportar un intento de phishing) **¿Cuáles son las características que los hacen sospechar de que se puede tratar de un intento de phishing?** De entrada los mensajes llegan medios borrados por así decirlo, no sé si es por características de seguridad, las imágenes adjuntas no se pueden ver. Me fijo en el dominio del remitente. Me fijo si el nombre del remitente me suena, si el tema por el que me escriben me resulta familiar, si es por algo en particular que me escriben. Esto para diferenciarlo de si es un correo promocional. Si he notado que cuando un cliente me muestra un correo en su celular no se lee el dominio, tengo que “clicar” para ver el dominio para ver quien se lo está enviando. Es importante porque lo único que ven es el cuerpo del mensaje.

¿Quién atiende estos casos?

Supongo que seguridad dentro de TI.

¿Consideran que los mecanismos actuales son eficientes y efectivos para enfrentar esta problemática?

Este mecanismo no es preventivo. Actúa después que ya sucedió el hecho. Porque si le doy clic a un enlace ya el daño está hecho.

¿Se les ocurre cómo se podría mejorar?

No porque no se me ocurriría bloquear remitentes de Hotmail o Gmail, porque son dominios que utilizan los clientes para enviarnos documentos. Se sugiere la idea de preclasificar los correos, notificando al usuario en el propio correo que se trata de un intento de estafa, mostrándole el dominio del remitente o a donde lo lleva el link. Luego de esto sugiere la idea de pre clasificarlo en categorías como lo hace Gmail, promociones, redes sociales, etc.

La idea de sensibilizar a través de mostrar los elementos sospechosos de un correo fue compartida

Adicionalmente se sugirió la posibilidad de incorporar un botón en Outlook para reportar el phishing.

Manifestó que hay gente que continúa utilizando el OWA. Interfaz web de Outlook.

¿Consideran que la concientización es importante para mejorar esta situación?

Es importante.

Frente a un correo sospechoso ¿qué espera que el sistema haga?

Que avise que el correo es sospechoso. Ponerle una categorización, ponerlo en rojo o en naranja.

¿Cómo evaluarías si la herramienta ha sido útil?

Si no sucedió nada. O algo que dejó de suceder, si había inconvenientes con el phishing que pare o disminuya. Si se implementa a nivel del Banco quizás también tenga un impacto para con los clientes.

Modalidad: Presencial
Entrevistados: Cr. María Noel Alberro e Ignacio Acuña
Cargos : Analista de Operaciones
¿Qué es un correo con un intento de phishing? Es un intento de sustracción de información que puede ser de la órbita personal o laboral. La idea es intentar sustraer, que el mismo usuario, dueño de la información aporte los datos a esa base o ese programa que generan. Se muestran casos reales **¿Están al tanto de la problemática del Phishing? ¿Qué opiniones tienen al respecto?** Es más difícil caer en estas cosas a quienes somos nativos digitales, aunque está más asociado a la formación que ser nativo digital o no. Al principio se asoció a la demografía de los clientes que tiene el Banco. Luego se comentó que también está asociado a la distracción **¿Cómo actúan cuando reciben un correo de apariencia sospechosa?** Ignacio conocía el mecanismo de reportarlo a través del portal de TI, María Noel no. Se comentó que una vez hubo una comunicación al respecto pero que se perdió en el mar de comunicaciones que emite el Banco internamente. Ignacio recibió solo una vez un correo de apariencia sospechosa en 13 años y María nunca en 9 años. Se comentó que hay una baja percepción del riesgo cuando nosotros nos protegemos en encasillar ese correo en "como no deseado" y decir ya está saldado. Si bloqueamos al correo en Outlook nos estamos protegiendo nosotros pero no estamos contribuyendo a reportarlo y proteger a aquellos compañeros que no lo reconocen como un riesgo. (Se comenta cuál es el procedimiento para reportar un intento de phishing) **¿Cuáles son las características que los hacen sospechar de que se puede tratar de un intento de phishing?**

La solicitud de actualización de información ya es una primera señal de alarma. Cuando se solicitan datos que en ese caso sabemos que es la página.

Algo interesante que se sugirió es que si no tengo una casilla de correo definida como canal de comunicación por ejemplo no tengo un correo asociado a mi cuenta del Banco y me llega un correo del Banco a esa casilla está mal.

Se comentó lo de revisar el dominio y el enlace.

¿Quién atiende estos casos?

¿Consideran que los mecanismos actuales son eficientes y efectivos para enfrentar esta problemática?

Sería el principio de toda acción. Lo más eficiente es comunicar y divulgar. Comunicación en prensa. Hay que educar a toda la población. Dar el mensaje para que la gente pueda detectar que es un phishing. Ignacio habló de un recambio generacional dentro del Banco y que la modalidad de phishing así como la conocemos dejará de ser efectiva y mutará a otra modalidad. Hoy es efectiva porque es una red de pesca. Hoy la tiras y caen algunos, pero en algún momento no van a caer, y se va a actualizar o va a tener una mutación, va a migrar hacia otro estilo.

La educación tiene acotado su rango de alcance. Está quien le da bolilla al tema, quien lo va a leer y lo va a absorber. Va a funcionar con la misma lógica del phishing, es un ruido para un montón de gente, no le va a dar bola y otras personas le van a dar bola y toma una actitud consciente sobre lo que se les está comunicando.

Nacho hace la pregunta que ante una mutación de la técnica cómo se debería abordar y ahí plantea que hay que pensar cosas nuevas, salirse de la caja y es ahí donde lleva a repensar. Plantea como decirle a alguien que está en peligro, en la calle le pongo un cartel en rojo que dice peligro, en informática como hago. Mencionó a Mozilla monitor, en donde colocó la casilla de correo y te informa si estuvo expuesta. En el caso de los funcionarios que utilizan su correo institucional en cuentas no relacionadas al trabajo, en caso de que fuera vulnerado un sitio donde una persona utilizó su correo y una contraseña para ese sito, ese correo pasó a estar en una base de datos de un ciberdelincuente.

Utilizando la idea del Mozilla Monitor le avisa al compañero en caso de que su cuenta haya sido expuesta. Hoy la lista negra te evita de toparse con un correo no deseado, pero hay que decirle al compañero que lo están atacando. Si vos le avisas habría un cambio de actitud, te

tienes que sentir víctima para estar alerta. Si el Banco te cuida y no te avisa, asume una actitud paternalista y hace que estés más vulnerable al momento de un descuido.

Sugieren hacer gamificación para concientizar a los usuarios.

¿Se les ocurre cómo se podría mejorar?

Cuando inicias sesión que se muestre un pop up como el que se ideó para la firma donde te avise que recibiste un correo sospechoso.

¿Consideran que la concientización es importante para mejorar esta situación?

Es importante. Ya se habló

Frente a un correo sospechoso ¿qué espera que el sistema haga?

Desde el punto de vista de la seguridad espera que no le lleguen correos. Lo que espero es no estar en riesgo. Hoy funciona bastante bien porque me llegó un único correo en 13 años. Pero no me avisa si pasa más seguido de lo que yo he visto.

María Noel comentó que cuando se envía un correo desde Gmail al Banco con adjuntos, el correo del Banco le avisa que este correo puede ser sospechoso

¿Cómo evaluarías si la herramienta ha sido útil?

La utilidad está muy relacionada a si se presenta el evento, si se presenta el evento puedo evaluar si fue satisfactorio o no. Que te salte una alerta y te diga cómo reportarlo.

De repente si me dicen en esta época del año te llegaron 100 correos, bloqueamos 99 y pasó 1, ahí puedo tener una idea de que tan bien funcionó. Si me dan un histórico.

Estaría bueno para saber en qué páginas estuviste hurgando, donde pusiste tu información para prevenir.

Se sugiere evaluar si la herramienta ha sido útil la idea de avisar al usuario que el correo es phishing destacando con color rojo los elementos que lo hacen sospechoso y de esta forma concientizar. Se sugiere la idea de incorporar un botón en la cinta de opciones de Outlook para reportar phishing

Opinan que el desconocimiento te lleva a cometer errores. La percepción del riesgo está atada al conocimiento de los mismos.

Opinan que cuanto más fácil sea el mecanismo de reporte mayor será la tasa entre el evento y la denuncia.

Ahí te dará un gap entre cuánta gente recibió la alerta y no denunció. O cuánta gente entendió que esa alerta era suficiente para no denunciar

Modalidad: virtual

Entrevistados: Lic. Andrea Delgado, Ec. Michel Godín y Javier Casal.

Cargos: Analista de Procesos, Ejecutivo de Gestión Humana y Ejecutivo de crédito personas

¿Qué es un correo con un intento de phishing?

Se muestran casos reales

¿Están al tanto de la problemática del Phishing? ¿Qué opiniones tienen al respecto?

Javier comenta que es posible que mucha gente caiga en este tipo de engaño y pone como ejemplo que hay muchos clientes que tienen dificultades para utilizar el cajero automático. Estima que hay mucha gente que no sabe diferenciar entre un mail real y uno fraudulento.

Michel plantea que uno de los problemas es utilizar el correo como canal de comunicación, ya que el correo es muy fácil de conseguir. Plantea que no está de acuerdo con que se haya tomado el correo como una fuente de validación de un montón de cosas.

Frente a un correo sospechoso ¿qué espera que el sistema haga?

Le planteamos opciones: Que el sistema filtre el correo y no les llegue, que les avise que se trata de un intento de estafa o que lo coloque en una carpeta como correo sospechoso.

Andrea opina que es importante enterarse de que está llegando un correo con intento de estafa para tener mayor precaución.

Javier opina que está bueno que te alerte. ~~También que se alerte a quien realmente debe intervenir para que no pase por el usuario que tenga que dar el aviso~~. Que se separe en una carpeta en el correo dando aviso que el correo se separó por un intento de fraude. El funcionario en caso de entender que no es un intento de estafa lo clasificará como bueno. Y en caso de que sea un caso de estafa de ahí mismo podrá confirmar esta situación sin tener que ingresar al portal de TI.

Michel sugiere que se implemente algo similar a lo de mesa de ayuda con un dedo para arriba o para abajo para confirmar si es phishing o no.

Investigar los incentivos que tienen los entrenados para que se tomen su tiempo para clasificar. A Javier se le ocurre hacer un simulacro enviando correos de prueba con elementos sospechosos para medir cómo reacciona la gente y determinar qué comportamiento tienen.

Entrevistas a Analistas de Seguridad de la Información

Modalidad: Presencial

Entrevistado: Ing. Alejandro Cao

Cargo: Especialista en Seguridad de la Información

Al respecto del Phishing

¿Está al tanto de la problemática del Phishing? ¿Y qué opinión personal tiene al respecto?

Si, están al tanto. *Es una forma de ataque la que el atacante envía un correo engañoso con alguna temática conocida o genera interés y de forma engañosa lo redirige a un sitio para robarle información personal o valiosa. Es un ataque que va cambiando y se han tenido bastantes y sigue teniendo mucho efecto. De hecho se utilizan los próximos mecanismos que el banco utiliza, a los días de la campana vinieron mails con el mismo formato. Quienes envían son más sofisticados y quienes lo reciben no están tan preparados. Ingeniería Social*

En caso afirmativo:

1. ¿Cómo lo resuelven ahora? ¿Es adecuado?

1. *SMG, relay de correo. Tiene reglas de spam y detecta correos maliciosos, pero no específicamente de phishing.*

2. *Usuarios que reportan, y se bloquean los remitentes(puede que la regla no sea tan efectiva. Se bloquea por contenido, por palabras clave sospechosas.*

2. ¿Quiénes atienden estos casos?

    1. *Normalmente el equipo de monitoreo y el equipo de Ciber. Dependiendo de los casos, puede ser phishing al banco o a una empresa de terceros.*

3. ¿Consideran que son eficientes y efectivos para enfrentar esta problemática?

    1. *Es difícil de saber, puede que haya mails que son filtrados. Siempre es una acción reactiva*

4. Estas conforme con el mecanismo actualmente

    1.

5. ¿Alguna idea de cómo lo resolvería si pudiesen mejorar?

    1. *Detección y entrenamiento parece interesante*

6. ¿Cuáles son las características en las que se fija para confirmar un caso positivo?

    1. *Asunto*
    2. *Remitente*
    3. *Sumatoria de variables para el algoritmo*
    4. *Ortografía*
    5. *idioma (Copyright en inglés, cuerpo en español)*
    6. *Redirecciones en los links (Bytly)*

7. Como podría ayudarte una herramienta para resolver y mitigar?

    1. *Marcar los correos como phishing*
    2. *Bloquear los links*
    3. *Entrenamiento, ayuda a mitigar*

Líneas generales

¿Qué funcionalidades se imagina?

1. *Advertencia*

2. *Umbral para detección precisa o certeza alta al respecto de un mail*

2. ¿Considera que están bien preparados?

    1. *Hay de todo, hay algunos, sobre todo los más jóvenes que lo incorporan mejor, no es una generalización. los más veteranos ya depende de lo actualizados que estén. Esto relativo a los funcionarios BROU.*

3. ¿Es importante que reciban ayuda regularmente?

    1. *Hay herramientas, que hacen campanas que envían correos de ejemplo, para ver si la gente hace clic, o lo reporta como phishing, herramientas que no tenemos. Capacitar de manera constructiva, y no represiva.*

4. ¿Cuáles serían los tiempos de respuesta adecuados? ¿Minutos, horas, días?

    1. *Actualmente es reactivo, cuando el correo es visualizado ya está identificado como phishing*

5. ¿Cuál es el tiempo de respuesta actual?

    1. *En menos de 1 hora, dependiendo de otros factores*

6. ¿Cómo evaluarías si la herramienta ha sido útil?

    1. *De los que detecto como phishing cuales fueron acertados % y no%*

    2. *de los que no detecto %*

7. ¿Cree usted que es un momento adecuado para contar con una herramienta del estilo?

    1. *Claro que sí, cree que es el momento adecuado*

RF identificado

Bloqueo de links en los mails sospechosos o detectados (ingresado)

% de eficiencia, detectados positivos, detectados falsos y % de los que no detecto y fueron reportados por los usuarios (ingresado)

Modalidad: Presencial

Entrevistado: Ing. Federico Zubiri

Cargo: Especialista en Seguridad de la Información

Al respecto del Phishing

¿Está al tanto de la problemática del Phishing? ¿Y qué opinión personal tiene al respecto?

Si estamos al tanto. Cuando una víctima recibe un correo que dice ser de una persona pero que en realidad no es, está intentando usurpar la personalidad para inducir a la víctima a hacer clic o ir a una página que no es la real. Estamos bajando de una ola que hemos tenido, pero que generan problemas sobre todo de reputación y "obliga" a las autoridades a salir a los medios para aclarar situaciones. Es un problema de ingeniería social. Esencial la educación, que hacer y que no hacer...y cuando

En caso afirmativo:

1. ¿Cómo lo resuelven ahora? ¿Es adecuado?
    1. *Es un problema que afecta a nivel del usuario. Hay organizaciones públicas y privadas que están en campanas, enfocadas en la educación. No podría resolverlo el banco, el origen es externo.*
    2. *Se atiende luego que llego, significa que paso las herramientas ya implementadas. Se reporta a csirt@brou. Se analiza y si se confirma se solicitan bloqueos, para que los dispositivos de borde no permitan que el correo entre. En ese caso el correo se descarta y el cliente no se entera*
2. ¿Quiénes atienden estos casos?

1. *Lo atiende el SOC, el servicio de monitoreo 7x 24 que está en ASI. Se hace un análisis preliminar y se realizan las acciones (Ticket a TSI, bloqueos o nuevas políticas).Si es un caso menos común, o más complejos elevan los casos hacia el 2do nivel de atención del Banco(más gente y más dedicado) sería ideal poder contar con una herramienta que permita ingresar un caso que aún no afecto al banco.*

3. ¿Consideran que son eficientes y efectivos para enfrentar esta problemática?

    1. *Siempre se puede ser más eficiente y más efectivo. Las herramientas que tenemos o que hay disponibles siempre están corriendo de atrás. Es un deseo ser más efectivo en términos de la prevención*

4. Estas conforme con el mecanismo actualmente

    1. Hay mejores mecanismos, pero actualmente cumple con el propósito

5. ¿Alguna idea de cómo lo resolvería si pudiesen mejorar?

    1. *Hay margen para herramientas que utilicen aprendizaje externo. Sería ilógico no utilizar el conocimiento de los demás.*

6. ¿Cuáles son las características en las que se fija para confirmar un caso positivo?

    1. *Asunto*
    2. *Contenido del mensaje, sobre todo (la gente cae por el contenido)*
        1. *ver el link*
    3. *Sender es importante (pero a veces no es efectivo)*

7. Como podría ayudarte una herramienta para resolver y mitigar?

    1. Que resuelva más rápido que el mecanismo actualmente
    2. y que ese aprendizaje se útil para los demás
    3.

Líneas generales

¿Qué funcionalidades se imagina?

1. *1.*

2. ¿Considera que están bien preparados?

    1. *Están mejor que antes, es subjetivo. Se sabe que hay respuestas hacia los clientes finales que quizás no sean las más acertadas. Una actividad que hay que ayudar permanentemente*

3. ¿Es importante que reciban ayuda regularmente?

    1. *Si, por supuesto, los modus operandi van cambiando y hay que estar acornándose constantemente.*

4. ¿Cuáles serían los tiempos de respuesta adecuados? ¿Minutos, horas, días?

    1. *1.*

5. ¿Cuál es el tiempo de respuesta actual?

    1. *Algunas horas, y depende de la actividad. Pero han mejorado por la actividad que han tenido, han afinado los mecanismos.*

6. ¿Cómo evaluarías si la herramienta ha sido útil?

    1. *Esa estadística se espera que este en algún lado. Si tiene un % de acierto alto y un % de error baja*

7. ¿Cree usted que es un momento adecuado para contar con una herramienta del estilo?

    1. *Si, esto vino para quedarse*

    2. *este ano exploto y hay que tratar de frenar las olas*

RF's

Hay que moderar el FeedBack. Hay que cuidar ese tema para que no genere "Repudio" (ingresado)

Datos estadísticos (ingresado)

Modalidad: Presencial
Entrevistada: Laura Fontana
Cargo: Analista de Seguridad de la Información
Al respecto del Phishing ¿Está al tanto de la problemática del Phishing? ¿Y qué opinión personal tiene al respecto? *Alguien que se hace pasar por una entidad legitima que trata de engañar. Falta comunicación hacia los clientes (los usuarios de correo)- hay que educar* En caso afirmativo: 1. ¿Cómo lo resuelven ahora? ¿Es adecuado? 1. *Lo resuelven cuando ya está el problema sobre la mesa, se bloquean las IP's, es reactivo y no es un mecanismo adecuado*. Me comenta además que hubo momentos en los que no se daba abasto con la cantidad de casos. 2. ¿Quiénes atienden estos casos? 1. *Es atendido por el equipo de Ciberseguridad* 3. ¿Consideran que son eficientes y efectivos para enfrentar esta problemática? 1. *Ya hay un cliente afectado, por lo que no considera que sea muy efectivo o eficiente. Además, sabe que hay en marcha o ya está implementado algún tipo de "solución".* 4. ¿Alguna idea de cómo lo resolvería si pudiesen mejorar? 1. *Educar a los clientes, en que cosas son las que tienen que ver y tener cuidado* 5. ¿Cuáles son las características en las que se fija para confirmar un caso positivo? 1. *Origen*

2. *Contenido del mensaje*

3. *Ortografía*

4. *Look and feel*

6. ¿Está conforme con el mecanismo actual?

    1. *No está conforme*

7. Como analista de SÍ, ¿cómo podría ayudarlo una herramienta para resolver o mitigar este problema?

    1. …

Líneas generales

1. ¿Qué funcionalidades se imagina?

    1. *Escaneo de correo adjunto*

    2. *Bloqueo de links en mails sospechosos*

    3. *PopUps informativos (nuevas definiciones, tips, etc.)*

2. ¿Considera importante que los usuarios finales reciban ayuda regularmente al respecto de esta problemática?

    1.

3. ¿Quiénes podrían obtener un beneficio?

    1. SR

4. ¿Cuáles serían los tiempos de respuesta adecuados? ¿Minutos, horas, días?

    1. *Lo antes posible, hablamos de minutos, eso podría ser aceptable*

5. ¿Cuál es el tiempo de respuesta actual?

    1. *No sabe, pero depende del caso*

    2. *Surgió que el conocimiento ante un intercambio cliente-analista queda en esa orbita y con esta herramienta, todos se pueden beneficiar de manera tangible de un nuevo caso positivo*

6. ¿Cómo evaluarías si la herramienta ha sido útil?

    1. *Se discutió de las diversas formas en las que la herramienta podría ser evaluada, teniendo como principal característica la capacidad de empezar a registrar interacciones, y de ahí poder medir para evaluar*.

    2. *Si los usuarios tienen la herramienta y la capacidad de detectar x sus propios medios, disminuiría la cantidad de casos en los que ASI debe trabajar, pudiendo dedicar más tiempo a los casos que el sistema no pudo detectar, y en consecuencia ser más efectivos*.

7. ¿Cree usted que es un momento adecuado para contar con una herramienta del estilo?

    1. *Es "**el**" momento*

8. Están preparados los usuarios para hacer una correcta detección de Phishing

    1. *Pensaba que los funcionarios estaban mejor preparados, pero no. Si considera que están más preparados que los clientes*

9. ¿Es importante que reciban ayuda regularmente al respecto de esta problemática?

    1. *Es importante que sepan lo que no deben hacer*

Hay una casilla especial para que los casos que actualmente sean sospechosos puedan ser derivados para su análisis, pero no todos lo saben. Y muchos de los mails que se envían con recomendaciones no son tenidos en cuenta

RF's

Bloqueo de Links en texto (ingresado)

Modalidad: Presencial

Entrevistado: Ing. Rodrigo Mateos

Cargo: Analista de Seguridad de la Información

Al respecto del Phishing

¿Está al tanto de la problemática del Phishing? ¿Y qué opinión personal tiene al respecto?

Es un correo que simula ser una empresa para intentar robar información. La información puede ser variada. puede ser directamente preguntando información o redirigirte a un sitio. Vemos muchos casos de phishing que tienen una redacción con problemas de semántica, problemas de ortografía.

Opino que la mejora manera de enfrentarlo es educar. La gente es engañada porque no lo sabe, no está educada.

1. ¿Cómo lo resuelven ahora? ¿Es adecuado?
    1. *Se ha comunicado en circulares, hay una dirección especifica. Cuando se recibe un mail (interno o externo) se analiza la posibilidad de bajar el sitio. Sitios de Phishing al BROU. Se está en contacto con otros bancos para poder estar comunicados, y se intercambian esta información*
2. ¿Quiénes atienden estos casos?
    1. *El correo, el aviso, llega a SI(seguridad de la información). Hay casos en el que los clientes han ingresado información, y allí entra en juego el departamento de Fraudes. Como hay mucha variedad en los asuntos, y las campanas de Phishing son donde se nota*
3. ¿Consideran que son eficientes y efectivos para enfrentar esta problemática?
    1. *Se trabaja luego que el problema arribo, y se busca actuar antes. Son eficientes en el caso que los sitios se logran dar de baja (o los suspenden temporalmente). Idealmente bajarlos enseguida seria lo deseado, pero no es así, no se puede. Contactan a los ISP para dar de baja los sitios.*

4. Estas conforme con el mecanismo actual

    1. *Me gustaría que tuviese un tiempo más reducido, que fuese más fácil de hacer lo que se necesita*

5. ¿Alguna idea de cómo lo resolvería si pudiesen mejorar?

    1. *SR*

6. ¿Cuáles son las características en las que se fija para confirmar un caso positivo?

    1. *Asunto (Simula ser del brou)*
    2. *Siempre hacen referencia a ciertas palabras (desbloqueo, password, etc.)*
    3. *Direcciones que simulan ser del brou, cuyo dominio no es del BROU*
    4. *Texto del mensaje, simula ser un funcionario o un correo automático del banco con link que redireccionan a webs muy similares, casi idénticos*
    5. *A veces los textos tienen imágenes, pero a veces todo el texto y las imágenes son una imagen completa que tienen un link hacia el sitio fraudulento*

7. Como podría ayudarte una herramienta para resolver y mitigar?

    1. *Si comprueba que hay un link, no corresponde. Pero si lo tiene, verificar que sea link valido. Hay ocasiones donde comunicaciones del Banco tienen links. por ejemplo cuando uno se conecta desde un nuevo dispositivo, y el tiempo en el que llega la comunicación vía mail (con link)...este es un caso de falso positivo.*

Líneas generales

¿Qué funcionalidades se imagina?

1. *El análisis del texto y frases comunes*
2. *Hay correos idénticos, pero con links diferentes*

2. ¿Considera que están bien preparados?

    1. *Como clientes, rodrigo describió correctamente cuales eran los stakeholders, y donde estaba el cómo analista de SI*
    2.

3. ¿Es importante que reciban ayuda regularmente?

    1. *Si, tienen que recibir, entiende que sí. Hay variedad de casos para identificar y fueron conversados*

4. ¿Cuáles serían los tiempos de respuesta actual y cuál sería el adecuado? ¿Minutos, horas, días?

    1. *Dentro de una hora sería aceptable, cuanto menor sea el tiempo sería muy bueno*

5. ¿Cómo evaluarías si la herramienta ha sido útil?

    1. *Si mejora los tiempos en cualquier sentido, es utilizan*
    2. *si no genera problemas nuevos*
    3. *que agregue valor, algo de valor al menos*

6. ¿Cree usted que es un momento adecuado para contar con una herramienta del estilo?

    1. *Si, es adecuado, hay que educar, y mejorar el proceso. Definitivamente de acuerdo.*

RF's identificado:

- Usar los mails que se reciben de clientes para alimentar el sistema y reentrenar el algoritmo de ML (ingresado)
- Bloquear links y comparar con links oficiales (ingresado)
- Falsos positivos ya conocidos para tratarlos apropiadamente (mails oficiales con links validos)(ingresado)
- Listar los links que hay en el mail, para ver mejor a donde apuntan (ingresado)

Anexo 3

Sprint 1 - Review

Tareas	Story points	Horas planificadas	Horas reales		
Apertura de mail	1	4	1		
Tokenizado de mail	3	12	5		
Detección de mail original	5	20	15		
Creación de archivo con id, body y label	3	12	10		
Data understanding and preparation - Data cleaning	3	12	10		
Data understanding and preparation - NLP (Procesamiento de Lenguaje Natural)	3	12	10		
Model training and evaluation - Entrenar y evaluar los clasificadores 1, 2 y 3	3	12	7		
Model testing	3	12	3		
	24	**96**	**61**	**Horas reales por SP**	2.5

Tabla 8.7-1: Detalle de esfuerzo estimado vs real Sprint 1

Sprint 2 - Review

Tareas	Story points	Horas planificadas	Horas reales		
Creación de la interfaz web inicial para que consuma la API	5	12.5			
Crear API inicial	3	7.5			
Deploy del modelo de ML	5	12.5			
Crear método en la API para recibir mails o body's de mails	8	20			
Creación del proyecto de Django	1	2.5			
Creación del modelo de datos iniciales	1	2.5			
Creación de la base de datos	1	2.5			
Agregar más correos de HAM para aumentar la muestra y evaluar resultado	3	7.5			
	27	**67.5**	**54**	**Horas reales por SP**	2.6

Tabla 8.7-3: Detalle de esfuerzo estimado vs real Sprint 2

Sprint 3 - Review

Tareas	Story points	Horas planificadas	Horas reales		
Interfaz gráfica para varios modelos	3	7.5			
API para consumo de modelo desplegado	8	20			
Agregar más correos de HAM para aumentar	3	7.5			

la muestra y evaluar resultado					
Deploy del modelo de ML	8	20			
	22	**55**	**56**	**Horas reales por SP**	2.5

Ilustración 8.7-5: Burndown Chart Sprin

Anexo 5

RCF 822 - STANDARD FOR THE FORMAT OF ARPA INTERNET TEXT MESSAGES

3. LEXICAL ANALYSIS OF MESSAGES

3.1. GENERAL DESCRIPTION

A message consists of header fields and, optionally, a body. The body is simply a sequence of lines containing ASCII characters. It is separated from the headers by a null line (i.e., a line with nothing preceding the CRLF).

3.1.1. LONG HEADER FIELDS

Each header field can be viewed as a single, logical line of ASCII characters, comprising a field-name and a field-body. For convenience, the field-body portion of this conceptual entity can be split into a multiple-line representation; this is called "folding". The general rule is that wherever there may be linear-white-space (NOT simply LWSP-chars), a CRLF immediately followed by AT LEAST one LWSP-char may instead be inserted. Thus, the single line

To: "Joe & J. Harvey" <ddd @Org>, JJV @ BBN

can be represented as:

To: "Joe & J. Harvey" <ddd @ Org>,
JJV@BBN

and

To: "Joe & J. Harvey"
<ddd@ Org>, JJV
@BBN

and

To: "Joe &
J. Harvey" <ddd @ Org>, JJV @ BBN

The process of moving from this folded multiple-line representation of a header field to its single line representation is called "unfolding". Unfolding is accomplished by regarding CRLF immediately followed by a LWSP-char as equivalent to the LWSP-char.

Note: While the standard permits folding wherever linear-

white-space is permitted, it is recommended that structured fields, such as those containing addresses, limit folding to higher-level syntactic breaks. For address fields, it is recommended that such folding occur between addresses, after the separating comma.

3.1.2. STRUCTURE OF HEADER FIELDS

Once a field has been unfolded, it may be viewed as being composed of a field-name followed by a colon (":"), followed by a field-body, and terminated by a carriage-return/line-feed. The field-name must be composed of printable ASCII characters (i.e., characters that have values between 33. and 126., decimal, except colon). The field-body may be composed of any ASCII characters, except CR or LF. (While CR and/or LF may be present in the actual text, they are removed by the action of unfolding the field.)

Certain field-bodies of headers may be interpreted according to an internal syntax that some systems may wish to parse. These fields are called "structured fields". Examples include fields containing dates and addresses. Other fields, such as "Subject" and "Comments", are regarded simply as strings of text.

Note: Any field which has a field-body that is defined as

other than simply <text> is to be treated as a structured field.

Field-names, unstructured field bodies and structured field bodies each are scanned by their own, independent "lexical" analyzers.

3.1.3. UNSTRUCTURED FIELD BODIES

For some fields, such as "Subject" and "Comments", no structuring is assumed, and they are treated simply as <text>s, as in the message body. Rules of folding apply to these fields, so that such field bodies which occupy several lines must therefore have the second and successive lines indented by at least one LWSP-char.

3.1.4. STRUCTURED FIELD BODIES

To aid in the creation and reading of structured fields, the free insertion of linear-white-space (which permits folding by inclusion of CRLFs) is allowed between lexical tokens. Rather than obscuring the syntax specifications for these structured fields with explicit syntax for this linear-white-space, the existence of another "lexical" analyzer is assumed. This analyzer does not apply for unstructured field bodies that are simply strings of text, as described above. The

analyzer provides an interpretation of the unfolded text composing the body of the field as a sequence of lexical symbols.

These symbols are:

- individual special characters
- quoted-strings
- domain-literals
- comments
- atoms

The first four of these symbols are self-delimiting. Atoms are not; they are delimited by the self-delimiting symbols and by linear-white-space. For the purposes of regenerating sequences of atoms and quoted-strings, exactly one SPACE is assumed to exist, and should be used, between them. (Also, in the "Clarifications" section on "White Space", below, note the rules about treatment of multiple contiguous LWSP-chars.)

So, for example, the folded body of an address field

":sysmail"@ Some-Group. Some-Org,
Muhammed.(I am the greatest) Ali @(the)Vegas.WBA

is analyzed into the following lexical symbols and types:

:sysmail quoted string

@ special

Some-Group atom

. special

Some-Org atom

, special

Muhammed atom

. special

(I am the greatest) comment

Ali atom

@ atom

(the) comment

Vegas atom

. special

WBA atom

The canonical representations for the data in these addresses are the following strings:

":sysmail"@Some-Group.Some-Org

and

Muhammed.Ali@Vegas.WBA

Note: For purposes of display, and when passing such structured information to other systems, such as mail protocol services, there must be NO linear-white-space between <word>s that are separated by period (".") or at-sign ("@") and exactly one SPACE between all other <word>s. Also, headers should be in a folded form.

3.2. HEADER FIELD DEFINITIONS

These rules show a field meta-syntax, without regard for the particular type or internal syntax. Their purpose is to permit detection of fields; also, they present to higher-level parsers an image of each field as fitting on one line.

field = field-name ":" [field-body] CRLF

field-name = 1*<any CHAR, excluding CTLs, SPACE, and ":">

field-body = field-body-contents
[CRLF LWSP-char field-body]

field-body-contents =
<the ASCII characters making up the field-body, as defined in the following sections, and consisting of combinations of atom, quoted-string, and

specials tokens, or else consisting of texts>

4. MESSAGE SPECIFICATION

4.1. SYNTAX

Note: Due to an artifact of the notational conventions, the syntax indicates that, when present, some fields, must be in a particular order. Header fields are NOT required to occur in any particular order, except that the message body must occur AFTER the headers. It is recommended that, if present, headers be sent in the order "Return-Path", "Received", "Date", "From", "Subject", "Sender", "To", "cc", etc.

This specification permits multiple occurrences of most fields. Except as noted, their interpretation is not specified here, and their use is discouraged.

The following syntax for the bodies of various fields should be thought of as describing each field body as a single long string (or line). The "Lexical Analysis of Message" section on "Long Header Fields", above, indicates how such long strings can be represented on more than one line in the actual transmitted message.

message = fields *(CRLF *text) ; Everything after

```
                              ;  first null line
                              ;  is message body

fields      =    dates                ; Creation time,
                 source               ;  author id & one
                1*destination         ;  address required
                 *optional-field      ;  others optional

source      = [  trace ]              ; net traversals
                 originator           ; original mail
               [  resent ]            ; forwarded

trace       =    return               ; path to sender
                1*received            ; receipt tags

return      =  "Return-path" ":" route-addr ; return address

received    =  "Received"    ":"     ; one per relay
                 ["from" domain]      ; sending host
                 ["by"   domain]      ; receiving host
                 ["via"  atom]        ; physical path
                *("with" atom)        ; link/mail protocol
                 ["id"   msg-id]      ; receiver msg id
                 ["for"  addr-spec]   ; initial form

                  ";"    date-time    ; time received

originator  =   authentic             ; authenticated addr
```

```
            [ "Reply-To"   ":" 1#address] )


authentic   =   "From"      ":"   mailbox  ; Single author
          / ( "Sender"    ":"   mailbox  ; Actual submittor
              "From"      ":" 1#mailbox) ; Multiple authors
                                         ;  or not sender


resent      =   resent-authentic
            [ "Resent-Reply-To"  ":" 1#address] )


resent-authentic =
          =   "Resent-From"     ":"   mailbox
          / ( "Resent-Sender"   ":"   mailbox
              "Resent-From"     ":" 1#mailbox  )


dates       =   orig-date                ; Original
            [ resent-date ]              ; Forwarded


orig-date   =  "Date"       ":"   date-time


resent-date =  "Resent-Date" ":"   date-time


destination =  "To"         ":" 1#address  ; Primary
          /  "Resent-To"    ":" 1#address
          /  "cc"           ":" 1#address  ; Secondary
          /  "Resent-cc"    ":" 1#address
          /  "bcc"          ":"  #address  ; Blind carbon
          /  "Resent-bcc"   ":"  #address
```

```
optional-field =
        /  "Message-ID"        ":"   msg-id
        /  "Resent-Message-ID" ":"   msg-id
        /  "In-Reply-To"       ":"  *(phrase / msg-id)
        /  "References"        ":"  *(phrase / msg-id)
        /  "Keywords"          ":"  #phrase
        /  "Subject"           ":"  *text
        /  "Comments"          ":"  *text
        /  "Encrypted"         ":" 1#2word
        /  extension-field             ; To be defined
        /  user-defined-field          ; May be pre-empted

msg-id     =  "<" addr-spec ">"            ; Unique message id

extension-field =
           <Any field which is defined in a document
            published as a formal extension to this
            specification; none will have names beginning
            with the string "X-">

user-defined-field =
           <Any field which has not been defined
            in this specification or published as an
            extension to this specification; names for
            such fields must be unique and may be
            pre-empted by published extensions>
```

Anexo 5

Ejemplos de correos con intentos de Phishing que recibieron los clientes del Banco:

-------- Mensaje original --------
Asunto: Verifique sus Productos Banco Republica
De: "Seguridad Banco Republica (BROU)"
Para: [redacted]
CC:

BANCO REPUBLICA

Su cuenta y llave digital deben ser verificados o seran bloqueados.

[-EMAIL-],

El Equipo de Seguridad Banco Republica (BROU) indica que su cuenta ha tenido una actividad sospechosa, por lo que procederemos a bloquearla si no la verifica.
El Banco Republica (BROU) le solicita que verifique su cuenta y llave digital correctamente, de lo contrario su cuenta sera bloqueada y debera dirigirse a una de nuestras sucursales para solventar el problema
A partir del envio de esta EMAIL, usted tiene de 24 a 72 horas para validar su cuenta, de lo contrario sera bloqueada.

VERIFIQUE INMEDIATAMENTE

Sincerely,
2022 ?.All rights reserved.

De: e.bankig-Brou.com.uy <cristianlario@hotmail.com>
Enviado: viernes, 12 de agosto de 2022 10:48
Para: [redacted]
Asunto: Verificacion eBROU

De: NOTIFICACIONES BANCO eBROU <sdoti18@hotmail.com>
Enviado: Monday, August 15, 2022 4:06:11 PM
Para: [redacted]
Asunto: AVISO eBROU

E-brou

lunes 15 de Agosto del 2022

Estimado Cliente,

Es necesario realizar una validación electrónica de tu cuenta para poder seguir disfrutando de nuestra banca en línea y evitar que tus productos sean inhabilitados.

Para confirmar su banca en línea y cumplir con los estatus legales requeridos solo debe ingresar a:

ebrou.com.uy

Tu seguridad es nuestra prioridad, es por ello que reforzamos nuestras plataformas en Línea.

Cordialmente,
E-brou

De: e-BrouAtencion al Cliente <marcelo_i@hotmail.com>
Enviado: Monday, August 15, 2022 10:45:11 AM
Para: [redacted]
Asunto: Por su seguridad valide su infomacion.

BANCO REPUBLICA

Estimado/a Cliente:

Informamos que su cuenta ha sido bloqueada temporalmente debido a un intento de ingreso desde una ubicación no autorizada realizada a través del servicio 24 horas en Internet.

Si no reconoces esta actividad tu cuenta será limitada automáticamente después de 24 horas, para verificar esta información ingresa a continuación.

CONTINUAR

Ante cualquier duda, contacta con nuestro Servicio de Atención al Cliente en el correo electrónico

Si fuiste tu, avisanos

-------- Mensaje original --------
De: "E-BROU (1) *Atencion*" <yerq01@hotmail.com>
Fecha: 12/8/22 12:01 (GMT-03:00)
Para: [redacted]
Asunto: Cuentas-Deshabilitada, Acceso denegado #882

Confirma, Deshabilitamos tu cuenta hasta recibir confirmacion. cuenta con 1 dia para desbloquear. Solicitaremos su llave digital junto con otros datos para verificar

Desbloquea en el siguiente enlace que hemos dispuesto:

Restablecer mi cuenta

De: Aviso@e-brou@atención.uy . <jennyprincez@hotmail.com>
Enviado: Friday, August 12, 2022 2:52:01 PM
Para: [redacted]
Asunto: Actualizar su Llave Digital e-brou !!!

Estimado Dignado/a:

Hemos puesto restricción en su Cuenta e-brou el día 12-08-2022

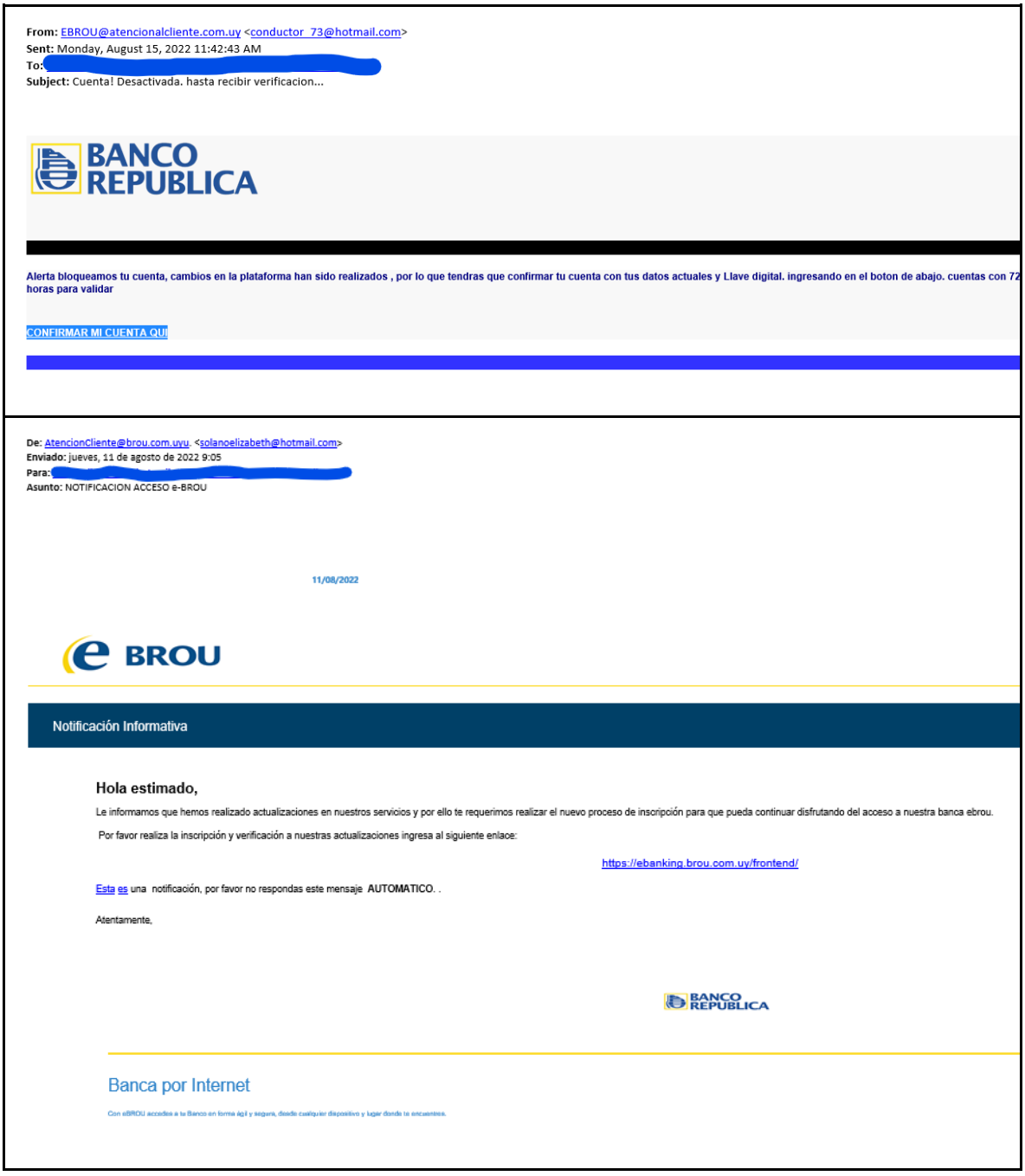
Ya que por su seguridad el banco ha puesto una Actualización de Validación y Confirmación de su Llave Digital para no permitir esa restricción debe Confirmar sus Credenciales en el siguiente enlace:

Confirmar Su Llave Virtual e-brou

Información Importante:

Debe realizar la verificación si no tomaremos el riesgo de Bloquear sus accesos y no podrá realizar transacciones ni pagos a terceros

Gracias por utilizar nuestro Servicios



From: EBROU@atencionalcliente.com.uy <conductor_73@hotmail.com>
Sent: Monday, August 15, 2022 11:42:43 AM
To: [redacted]
Subject: Cuenta! Desactivada. hasta recibir verificacion...

BANCO REPUBLICA

Alerta bloqueamos tu cuenta, cambios en la plataforma han sido realizados , por lo que tendras que confirmar tu cuenta con tus datos actuales y Llave digital. ingresando en el boton de abajo. cuentas con 72 horas para validar

CONFIRMAR MI CUENTA QUI

De: AtencionCliente@brou.com.uyu. <solanoelizabeth@hotmail.com>
Enviado: jueves, 11 de agosto de 2022 9:05
Para: [redacted]
Asunto: NOTIFICACION ACCESO e-BROU

11/08/2022

BROU

Notificación Informativa

Hola estimado,

Le informamos que hemos realizado actualizaciones en nuestros servicios y por ello te requerimos realizar el nuevo proceso de inscripción para que pueda continuar disfrutando del acceso a nuestra banca ebrou.

Por favor realiza la inscripción y verificación a nuestras actualizaciones ingresa al siguiente enlace:

https://ebanking.brou.com.uy/frontend/

Esta es una notificación, por favor no respondas este mensaje **AUTOMATICO**. .

Atentamente,

BANCO REPUBLICA

Banca por Internet

Con eBROU accedes a tu Banco en forma ágil y segura, desde cualquier dispositivo y lugar donde te encuentres.

Anexo 6

Encuesta:Federico Zubbiri

Pregunta (del 1 a- 10, siendo 1 el minimo, 10 el maximo)	Respuesta
¿Es facil encontrar la opcion buscada?	9
¿La opcion seleccionada, cumple con su objetivo?	10
¿Se brinda informacion relevante de acuerdo al contexto del problema?	8
¿El funcionamiento del login, es el esperado?	10
¿Es sencillo acceder a crear un usuario?	10
¿Es sencillo bloquear/desbloquear un usuario?	7
¿Es sencillo cambiar la contraseña un usuario?	8
¿Califique la sencillez de subir un archivo para su analisis?	10
¿Es facil encontrarlo luego para su procesamiento?	10
¿El resultado del procesamiento, es claro? (archivo)	8
¿Califique la sencillez de analizar un texto en particular?	9
¿El resultado del procesamiento, es claro? (texto)	9

Calificación Promedio	9

Encuesta: Alejandro Cao

Pregunta (del 1 a- 10, siendo 1 el minimo, 10 el maximo)	Respuesta
¿Es facil encontrar la opcion buscada?	7
¿La opcion seleccionada, cumple con su objetivo?	8
¿Se brinda informacion relevante de acuerdo al contexto del problema?	9
¿El funcionamiento del login, es el esperado?	7
¿Es sencillo acceder a crear un usuario?	7
¿Es sencillo bloquear/desbloquear un usuario?	8
¿Es sencillo cambiar la contraseña un usuario?	9
¿Califique la sencillez de subir un archivo para su analisis?	8
¿Es facil encontrarlo luego para su procesamiento?	7
¿El resultado del procesamiento, es claro? (archivo)	9
¿Califique la sencillez de analizar un texto en particular?	9
¿El resultado del procesamiento, es claro? (texto)	9

Calificación Promedio	8

Encuesta: Laura Fontana

Pregunta (del 1 a- 10, siendo 1 el minimo, 10 el maximo)	Respuesta
¿Es facil encontrar la opcion buscada?	8
¿La opcion seleccionada, cumple con su objetivo?	9
¿Se brinda informacion relevante de acuerdo al contexto del problema?	10
¿El funcionamiento del login, es el esperado?	9
¿Es sencillo acceder a crear un usuario?	9
¿Es sencillo bloquear/desbloquear un usuario?	7
¿Es sencillo cambiar la contraseña un usuario?	8
¿Califique la sencillez de subir un archivo para su analisis?	8
¿Es facil encontrarlo luego para su procesamiento?	9
¿El resultado del procesamiento, es claro? (archivo)	8
¿Califique la sencillez de analizar un texto en particular?	7
¿El resultado del procesamiento, es claro? (texto)	9

Calificación Promedio	8

Encuesta: Rodrigo Mateos

Pregunta (del 1 a- 10, siendo 1 el minimo, 10 el maximo)	Respuesta
¿Es facil encontrar la opcion buscada?	7
¿La opcion seleccionada, cumple con su objetivo?	8
¿Se brinda informacion relevante de acuerdo al contexto del problema?	9
¿El funcionamiento del login, es el esperado?	9
¿Es sencillo acceder a crear un usuario?	7
¿Es sencillo bloquear/desbloquear un usuario?	9
¿Es sencillo cambiar la contraseña un usuario?	7
¿Califique la sencillez de subir un archivo para su analisis?	8
¿Es facil encontrarlo luego para su procesamiento?	7
¿El resultado del procesamiento, es claro? (archivo)	9
¿Califique la sencillez de analizar un texto en particular?	7
¿El resultado del procesamiento, es claro? (texto)	7

Calificación Promedio	8