

Universidad ORT Uruguay
Facultad de Ingeniería

Looke.Me

Toda la moda en un solo lugar

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

Agustín Alonso – 154212

Máximo Halty – 154840

Tutor: Bruno Delgado Paguas

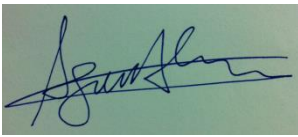
2013

1 Declaración de autoría

Nosotros, Agustín Alonso y Máximo Halty, declaramos que el trabajo que se presenta en esta obra es de nuestra propia mano.

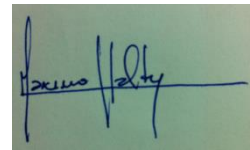
Podemos asegurar que:

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



Agustín Alonso

7 de Marzo, 2013



Máximo Halty

7 de Marzo, 2013

2 Agradecimientos

Agradecemos a nuestras familias, padres y hermanos, parejas y amigos por el apoyo que nos brindaron durante el transcurso de toda la carrera, y con singular énfasis en este último año en que desarrollamos esta tesis. En especial, queremos agradecer a Marcela y Alejandro por prestarnos su estudio, espacio de trabajo durante todo el transcurso del proyecto.

A José Miguel Majó y Juan Manuel Dellepiane, que siendo clientes del proyecto, nos apoyaron en todos los aspectos a su alcance para que el grupo pudiera llevar adelante el mismo. Aun cuando variamos algunas metas, tanto José Miguel como Juan Manuel comprendieron los motivos de los cambios.

A la Diseñadora Gráfica Mercedes Laffitte por su compromiso y dedicación. Ella realizó el diseño Web de la página de los usuarios, integrándose creativamente a la dinámica del equipo.

Queremos agradecer además, en especial, a nuestro tutor Dr (c) Ing. Bruno Delgado, MBF, PMP, que nos guió durante concepción y redacción de esta tesis, transformándose en una pieza fundamental para el equipo.

3 Abstract

Lookea.Me optimiza la experiencia de selección y compra de vestimenta. Lookea.Me une todas las vitrinas de las tiendas en un solo lugar, para mejorar el proceso de toma de decisiones de los consumidores a la hora de realizar la compra de una prenda de vestir. Para ello realizamos un portal Web cuyo desarrollo presenta este documento.

Surge como un emprendimiento privado externo al equipo de proyecto con el objetivo de crear una empresa propia, basada en los servicios que ofrecería Lookea.Me tanto a los consumidores como a las tiendas de moda.

Como vitrina de todas las vitrinas, Lookea.Me cuenta con información actualizada los 365 días del año, centralizando las ofertas más destacadas de la moda, permitiendo tomar decisiones mejor fundamentadas, además de mejorar la experiencia de compra. Los usuarios podrán además vincular Lookea.Me con sus redes sociales, de forma de compartir sus intereses y opiniones respecto a la moda.

Lookea.Me ofrece un nuevo método de publicidad, utilizando Internet como canal. Las tiendas pueden realizar ofertas personalizadas a sus consumidores habituales, debido a que el sistema le proporciona a cada una de ellas información estadística relevante para obtener patrones de comportamiento, logrando un mayor acercamiento hacia el usuario final.

El proyecto inició con la investigación de mercado y la extracción primaria de requerimientos, base que permitió realizar posteriormente una completa especificación de requerimientos. Habiendo realizado la investigación correspondiente en conjunto con los clientes mencionados, se presentó la idea al Centro de Innovación y Emprendedurismo (CIE) de la Universidad ORT, y fue aceptada en Mayo de 2012. Una vez culminada esta primera etapa, en Junio 2012 se comenzó con el diseño arquitectónico y desarrollo de los sistemas utilizando tecnologías Microsoft .NET Framework 4.0, Microsoft SQL Server 2008 R2 y DB40.

Si bien Lookea.Me surgió como un emprendimiento privado externo al equipo de proyecto, la oportunidad de crear un producto exitoso en el mercado llevó a Agustín Alonso y Máximo Halty incorporarse en el equipo de Lookea.Me para futuras oportunidades de desarrollo.

4 Palabras claves

Tienda, moda, Web, consumidor, usuario, cliente, sistema, Shopping, equipo, grupo, artículo, comercial, proyecto, Lookea.Me, vitrina, requerimiento, negocio, diseño gráfico, usabilidad, SCRUM.

5 Índice

1	Declaración de autoría.....	2
2	Agradecimientos	3
3	Abstract.....	4
4	Palabras claves	5
5	Índice	6
6	Glosario	11
7	Introducción	16
7.1	La idea.....	16
7.2	Motivación.....	16
7.3	Clientes y usuarios	17
7.4	Principales objetivos	18
7.4.1	Objetivos académicos	18
7.4.2	Objetivos del proyecto	18
7.5	Organización del documento y CD.....	19
8	El negocio	21
8.1	La oportunidad.....	21
8.2	La solución	22
8.2.1	Beneficios para las tiendas:	22
8.2.2	Beneficios para los consumidores:	22
8.3	Soluciones existentes	23
8.4	Arquitectura del negocio	27
8.4.1	Estructura de las tiendas.....	27
9	Descripción del proyecto	30
9.1	Alcance del sistema	30
9.1.1	Descripción del sistema y su entorno	31
9.1.2	Descripción resumida de las funcionalidades	31
9.1.3	Tiendas	31
9.1.4	Usuarios	32
9.2	Descripción de la forma de trabajo	32
9.3	Proceso de desarrollo	33
9.3.1	Introducción.....	33
9.3.2	Marco metodológico	33
9.4	Metodología tradicional y SCRUM.....	35

9.5	Procesos de ingeniería.....	36
9.5.1	Perspectiva del comportamiento.....	36
9.5.2	Perspectiva funcional.....	37
9.5.3	Perspectiva organizacional.....	38
9.5.4	Perspectiva metodológica.....	39
9.5.5	Perspectiva de resultados.....	40
10	Requerimientos.....	42
10.1	Relevamiento inicial.....	42
10.2	Técnicas utilizadas.....	42
10.2.1	Reuniones con clientes.....	42
10.2.2	Encuestas masivas a potenciales consumidores.....	43
10.2.3	Investigación de sistemas similares.....	43
10.2.4	Investigación de características claves.....	43
10.2.5	Grupo Foco.....	44
10.3	Situación del cliente.....	44
10.3.1	Contexto general.....	44
10.3.2	Requisitos para los locales.....	45
10.4	Situación del usuario.....	45
10.4.1	Contexto general.....	45
10.4.2	Requisitos para los usuarios.....	46
10.5	Especificación de requerimientos.....	46
10.5.1	Requerimientos funcionales.....	46
10.5.2	Requerimientos no funcionales.....	48
10.5.3	Usabilidad.....	48
10.5.4	Mantenibilidad.....	49
10.6	Criterios de priorización.....	49
10.7	Clientes y actores.....	49
10.8	Validaciones y verificaciones.....	50
11	Arquitectura y diseño.....	51
11.1	Componentes del sistema.....	52
11.1.1	Aplicación de usuarios (http://www.lookea.me).....	52
11.1.2	Aplicación de tiendas (http://stores.lookea.me).....	54
11.1.3	Aplicación administradores (http://admin.lookea.me).....	55
11.2	Características de calidad consideradas.....	56
11.2.1	Usabilidad.....	56

11.2.2	Eficiencia.....	56
11.2.3	Disponibilidad	57
11.2.4	Seguridad.....	57
11.2.5	Mantenibilidad.....	57
11.2.6	Escalabilidad	58
11.3	Decisiones Tecnológicas	58
11.4	Herramientas.....	60
11.5	Fundamentos de arquitectura.....	60
11.6	Patrones de diseños utilizados	64
11.6.1	Unit of Work	64
11.7	Vistas de arquitectura	65
11.7.1	Vista de componentes	65
11.7.2	Vista de despliegue	68
11.7.3	Vista de procesos	70
11.7.4	Vista de casos de uso	73
12	Gestión de la configuración	75
12.1	Introducción	75
12.2	Herramientas	75
12.3	Elementos de configuración.....	75
12.3.1	Documentos.....	75
12.3.2	Código fuente	76
12.4	Repositorio.....	76
12.5	Actividades	79
12.6	Control de cambio	80
12.6.1	Circuito de control de cambios de documentos.....	80
12.6.2	Reglas de control de cambios de código fuente	81
12.7	Respaldos.....	81
12.8	Análisis de problemas encontrados	81
12.8.1	Lectura sucia en los documentos	81
12.8.2	Mensajes de actualización constantes	82
12.8.3	Errores por DLLs no importadas	82
13	Gestión de la calidad	83
13.1	Introducción	83
13.2	Actividades de SQA	83
13.2.1	Actividades sobre la gestión.....	83

13.2.2	Actividades sobre el producto	84
13.3	Calidad del proceso de desarrollo	88
13.3.1	Introducción.....	88
13.3.2	PSP.....	88
13.3.3	TSP.....	89
13.3.4	Pruebas	90
14	Gerencia.....	103
14.1	Objetivos.....	103
14.2	Modelo de ciclo de vida y ciclo de vida	103
14.3	Gestión del alcance, tiempo y costo	106
14.4	Cronograma	108
14.4.1	Primeras etapas	108
14.4.2	Planificación.....	109
14.5	Proceso de desarrollo.....	110
14.5.1	Metodología SCRUM	110
14.5.2	División de tareas	111
14.5.3	Principales tareas	114
14.5.4	Control y seguimiento.....	115
14.6	Re planificación	119
14.7	Riesgos.....	124
14.7.1	Identificación.....	124
14.7.2	Principales riesgos	126
14.7.3	Seguimiento y control.....	131
14.7.4	Riesgos ocurridos	132
14.7.5	Solución	132
14.8	Comunicaciones	133
14.8.1	Interna	133
14.8.2	Externa	134
15	Conclusiones	135
15.1	Ingeniería de requerimientos	135
15.2	Arquitectura y diseño	135
15.3	Gestión de la configuración.....	136
15.4	Gestión de la calidad	136
15.5	Gerencia.....	137
15.6	Conclusiones del proyecto.....	138

16 Bibliografía	139
-----------------------	-----

Anexos

- I. Encuestas
- II. Gestión de riesgos
- III. Herramientas
- IV. Lista preliminar de requerimientos
- V. Métricas del producto
- VI. Minuta de reunión
- VII. Plan de pruebas
- VIII. Plan de pruebas de usabilidad
- IX. PSP
- X. SCRUM
- XI. Plan de negocios
- XII. Manual de usuarios
- XIII. Estándares de diseño
- XIV. Plataforma multi-lado
- XV. Investigación
- XVI. ESRE
- XVII. Estándares de codificación
- XVIII. Mock-ups

6 Glosario

AJAX: acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML). Técnica de desarrollo Web para crear aplicaciones interactivas con la utilización de JavaScript y XML.

ASP.NET: *Framework* para aplicaciones Web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios Web dinámicos, aplicaciones Web y servicios Web XML.

Back-end: Portal Web por el cual se administra el perfil de cada una de las tiendas. Únicamente los responsables de las tiendas tienen acceso al mismo.

Base de datos: Conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Browser: Un navegador o navegador Web (del inglés, *Web browser*) es una aplicación que opera a través de Internet, interpretando la información de archivos y sitios Web para que éstos puedan ser leídos.

Bug: Un defecto de software (del inglés, *computer bug*), es el resultado de un fallo o deficiencia durante el proceso de creación de software.

C#: Lenguaje de programación para los proyectos Mono y .Net.

CANVAS: Es una herramienta de gestión estratégica y empresarial, que permite describir, diseñar, inventar, y adaptar el modelo de negocio.

CCC: Circuito de Control de Cambios.

Check-in: Operación asociada al control de versiones de código fuente. Cuando se realiza esta operación se suben los cambios realizados sobre los archivos locales al servidor controlador de versiones.

Check-out: Operación asociada al control de versiones de código fuente. Cuando se edita un archivo automáticamente se ejecuta esta operación permitiendo hacer cambios sobre el mismo localmente, para luego poder impactar los cambios en el servidor ejecutando la operación Check-in.

CIE: Centro de Innovación y Emprendimientos de la Universidad ORT. Promueve y desarrolla la generación de nuevos emprendedores.

CMMI: *Capability Maturity Model Integration* - Integración de modelos de madurez de capacidades. Modelo de mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software.

DAO: *Data Access Object*. Componente de software que facilita el acceso a datos almacenados. Es una implementación creada por Microsoft.

DB4O: *DataBase 4 (for) Objects*. Motor de base de datos orientada a objetos. Desarrollado por db4objects, Inc.

DDD: *Domain Driven Design* - Diseño orientado al dominio.

Drag and Drop: (arrastrar y soltar) Término usado en informática referido a la acción de arrastrar y soltar con el ratón objetos de una ventana a otra o entre partes de una misma ventana o programa.

ESRE: Documento de Especificación de Requerimientos

Facebook: Empresa creada por Mark Zuckerberg en un sitio Web de redes sociales.

Front-end: Sitio Web de Lookea.Me público.

GeneXus: Herramienta de desarrollo de software ágil, multiplataforma, basada en conocimiento, orientada principalmente a aplicaciones Web empresariales, plataformas Windows y dispositivos móviles o inteligentes.

Interfaz de usuario (UI): Medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora. Comprende todos los puntos de contacto entre el usuario y el equipo. Normalmente suelen ser fáciles de entender y fáciles de accionar.

IR: Ingeniería de Requerimientos. Comprende todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer para un software nuevo o modificado, tomando en cuenta los diversos requisitos de los interesados.

Lenguajes de Programación: Idioma artificial diseñado para expresar procesos que pueden ser llevadas a cabo por máquinas como las computadoras.

LISI: Laboratorio de Investigación en Sistemas de Información. Tiene como misión contribuir a generar conocimientos en el área de TI/SI enfocándose a la transferencia de tecnologías entre la academia y la empresa.

Macro: Serie de procedimientos o funciones agrupados en un módulo *vba* (Visual Basic para aplicaciones) que se almacena para poder ejecutarse cuando es invocado.

MER: Modelo Entidad-Relación. Herramienta para el modelado de datos de un sistema de información.

Microsoft SQL Server: Sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional.

Microsoft Visual Studio: es un entorno de desarrollo integrado (IDE) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++,

Visual C#, Visual J#, y Visual Basic .NET, al igual que entornos de desarrollo Web como ASP.NET.

MVC: *Model View Controler*. Modelo Vista Controlador es un patrón o modelo de abstracción de desarrollo de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos.

On-line: en línea (conectado).

Off-line: fuera de línea (desconectado).

ORTsf: *Software Factory* de la Universidad ORT Uruguay.

PSP: Proceso Personal de Software. Conjunto de prácticas disciplinadas para la gestión del tiempo y mejora de la productividad personal de los programadores o ingenieros de software, en tareas de desarrollo y mantenimiento de sistemas.

RAD: *Rapid Application Development* .Desarrollo rápido de aplicaciones. Proceso de desarrollo de software que permite construir sistemas utilizables en poco tiempo.

RF: Requerimiento Funcional. Requisito que define el comportamiento del software explícitamente detallado por parte del cliente y que está asociado a las funciones esperadas que pueda satisfacer el mismo.

RNF: Requisito que define el comportamiento del software explícitamente detallado por parte del cliente y que está asociado a las características de las restricciones que debe cumplir el mismo como rendimiento, mantenibilidad, portabilidad u otras no funcionales.

RUP: *Rational Unified Process*. Proceso de desarrollo de software desarrollado por la empresa “Rational Software”, actualmente propiedad de IBM

SCM: *Software Configuration Management* o Gestión de configuración de *software*, es una actividad de protección que se aplica durante todo el ciclo de vida. Se encarga de identificar, controlar, examinar e informar de las modificaciones que ocurren en el desarrollo del sistema.

SQA: *Software Quality Assurance* o Gestión de la Calidad del *Software*, es un conjunto de actividades sistemáticas y planeadas para asegurar que los procesos y productos de software cumplen con los requerimientos, estándares y procedimientos.

SQL: *Structured Query Language* o lenguaje de consulta estructurado, es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

TSP: *Team Software Process*. Marco de trabajo de procesos definidos diseñado para ayudar a equipos de gerentes e ingenieros a organizar y producir proyectos de software de gran escala.

Twitter: Servicio de *micro-blogging*. Permite enviar mensajes de texto plano de corta longitud, con un máximo de 140 caracteres, llamados *tweets*, que se muestran en la página principal del usuario.

UML: *Unified Modeling Language* o Lenguaje Unificado de Modelado, es el lenguaje de modelado de sistemas de *software* más conocido y utilizado en la actualidad.

XML: *eXtensible Markup Language* ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el *World Wide Web Consortium* (W3C).

7 Introducción

El presente documento presenta el proyecto “Looke.Me – Toda la moda en un solo lugar”, realizado por los estudiantes Agustín Alonso y Máximo Halty, junto con el apoyo de Juan Manuel Dellepiane y José Miguel Majó, cumpliendo así el rol de clientes. El mismo fue realizado con el fin de obtener el título de Ingeniero de Sistemas, en el período de Marzo 2012 y Marzo 2013 dentro del departamento de L.I.S.I. (Laboratorio de Investigación de Sistemas de Información).

7.1 La idea

El objetivo de este proyecto es implementar una herramienta informática que facilite y agilice el proceso de toma de decisiones a la hora de realizar una compra de prendas de vestir, ofreciendo un servicio completo mediante el cual el usuario pueda obtener información específica sobre cierta indumentaria u otros aspectos de la moda, todo en un mismo lugar. El proyecto comprende la creación de un sistema on-line que presente en un solo sitio todas las vitrinas de los locales de moda.

En el sistema propuesto, las tiendas de moda podrán ampliar sus canales de comunicación y publicidad de una forma económica mediante el uso de plataformas tecnológicas tradicionales tales como el PC, y de última generación como *smartphones* o *tablets*, además de pertenecer a un portal Web que integre toda la moda y redes sociales en un solo lugar.

La propuesta no aspira a sustituir la compra física en el local de ropa, sino que el servicio ofrece y agiliza los procesos previos a la compra, tales como evaluar marcas, precios y ubicación de tiendas, entre otros.

7.2 Motivación

La idea surge por parte de José Miguel Majó y Juan Manuel Dellepiane con el objetivo de realizar un emprendimiento propio, que si bien era claro respecto a su idea y motivación, no contaban con el conocimiento o apoyo tecnológico para realizarlo, por lo que consultaron la viabilidad tecnológica del proyecto a Máximo Halty, y posteriormente a Agustín Alonso.

El proyecto fue abordado desde un inicio como una oportunidad para innovar, con potencial para crecer en el mercado. En consecuencia, se propone a José Miguel y Juan Manuel presentar el proyecto a la Universidad ORT Uruguay para realizarlo como tesis de grado.

Posteriormente el proyecto fue presentado y aprobado por el comité del CIE (Centro de Innovación y de Emprendimientos de la Universidad ORT) [33] para formar parte de su proyecto de emprendedores. Esta aprobación fue un factor de gran importancia para estimular la motivación pues surgió la posibilidad de implantar una empresa y comercializar de forma exitosa a Lookea.Me.

7.3 Clientes y usuarios

Lookea.Me surge como una plataforma con la doble finalidad de mejorar y simplificar la experiencia de compra por parte de los consumidores, y a la vez, convertir las tiendas de moda en aliadas. Es así que Lookea.Me cumple como intermediario entre las tiendas y los consumidores (Figura 1), de forma que estos últimos puedan obtener toda la oferta de moda en un solo lugar de forma rápida y sencilla.

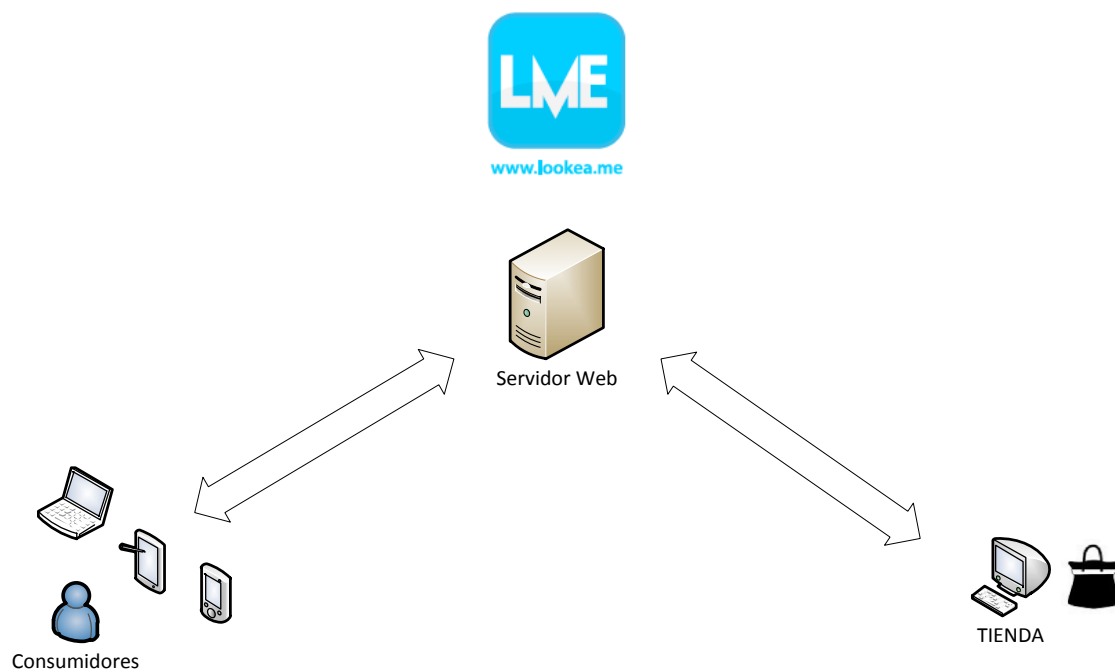


Figura 1 – Usuarios y tiendas

A través de Lookea.Me las tiendas autogeneran su propio contenido y pueden publicar los artículos que deseen, además de obtener información de sus compradores y poder realizar una venta más personalizada a dichos compradores.

Por su lado, los consumidores además de poder acceder a un sitio que publique toda la moda provista por las tiendas, podrán involucrarse con la moda realizando comentarios sobre las prendas y las tiendas, pudiendo asimismo realizar un seguimiento de las vitrinas de las tiendas que le gusten, compartiendo dicha experiencia con otros usuarios.

7.4 Principales objetivos

7.4.1 Objetivos académicos

7.4.1.1 Puesta en práctica de los conocimientos adquiridos:

Poner en práctica los distintos conocimientos adquiridos durante el transcurso de la carrera, en la cual se dictan cursos independientes de cada área del proyecto pero no en forma conjunta para gestionarlo como tal. A su vez, enfrentar el desafío de poner en práctico dichos conocimientos en un proyecto aplicado sobre un producto de índole real.

7.4.1.2 Adquirir conocimientos extra académicos:

Si bien el proyecto es un proyecto académico, el objetivo es también desarrollar conocimientos que no están relacionados directamente con el plan de trabajo de la carrera. Entre ellos es posible encontrar estudios y aspectos de negocio, emprendedurismo y diseño gráfico aplicado a desarrollo de aplicaciones.

7.4.2 Objetivos del proyecto

7.4.2.1 Desarrollar un producto innovador

Se busca que el producto logre satisfacer las necesidades en los consumidores de moda que aún no han sido resueltas por las soluciones existentes. Estas últimas han logrado satisfacer parcialmente dichas necesidades, por lo que Lookea.Me busca ser innovador como solución que satisfaga a una mayor parte de ellas. Como solución, Lookea.Me propone un portal Web en el cual la facilidad de uso y aprendizaje por parte de los usuarios es clave para que dicho producto supere las expectativas de estos últimos.

7.4.2.2 Generar una aplicación que haga uso de tecnologías emergentes y tendencias sociales

En el desarrollo del producto se busca utilizar las tecnologías existentes en el estado de la práctica, de forma de explotar el uso tecnológico que se puede obtener a partir de una aplicación, tanto en una computadora como también en los *smartphones*. Además, alinear el producto de formar que sea fácil de integrar con aspectos sociales existentes hoy en día como lo son las redes sociales, tales como Facebook y Twitter, entre otras.

7.4.2.3 Crear una nueva área de negocio:

Considerando el éxito que puede tener Lookea.Me en el mercado, se apunta a crear un producto con potencial que logre un gran impacto en los usuarios, pudiendo generar nuevas líneas de negocio. De esta forma, las tiendas de moda podrán ampliar los canales de comunicación y publicidad existentes hoy en día en el mercado de la moda, pudiendo además ampliar sus canales de venta y distribución en etapas posteriores.

7.5 Organización del documento y CD

Directorio	Archivos
\entregaDoc	Entrega_LookeaMe_CC.doc
\entregaPDF	Entrega_LookeaMe_CC.pdf
\entregaDoc\Anexos	Entrega_LookeaMe_Anexo_I.doc
	Entrega_LookeaMe_Anexo_II.doc
	Entrega_LookeaMe_Anexo_III.doc
	Entrega_LookeaMe_Anexo_IV.doc
	Entrega_LookeaMe_Anexo_V.doc
	Entrega_LookeaMe_Anexo_VI.doc
	Entrega_LookeaMe_Anexo_VII.doc
	Entrega_LookeaMe_Anexo_VIII.doc
	Entrega_LookeaMe_Anexo_IX.doc
	Entrega_LookeaMe_Anexo_X.doc
	Entrega_LookeaMe_Anexo_XI.doc
	Entrega_LookeaMe_Anexo_XII.doc
	Entrega_LookeaMe_Anexo_XIII.doc
	Entrega_LookeaMe_Anexo_XIV.doc
	Entrega_LookeaMe_Anexo_XV.doc
	Entrega_LookeaMe_Anexo_XVI.doc
	Entrega_LookeaMe_Anexo_XVII.doc
	Entrega_LookeaMe_Anexo_XVIII.doc
\entregaPDF\Anexos	Entrega_LookeaMe_Anexo_I.pdf
	Entrega_LookeaMe_Anexo_II.pdf
	Entrega_LookeaMe_Anexo_III.pdf
	Entrega_LookeaMe_Anexo_IV.pdf
	Entrega_LookeaMe_Anexo_V.pdf
	Entrega_LookeaMe_Anexo_VI.pdf
	Entrega_LookeaMe_Anexo_VII.pdf
	Entrega_LookeaMe_Anexo_VIII.pdf
	Entrega_LookeaMe_Anexo_IX.pdf
	Entrega_LookeaMe_Anexo_X.pdf
	Entrega_LookeaMe_Anexo_XI.pdf
	Entrega_LookeaMe_Anexo_XII.pdf
	Entrega_LookeaMe_Anexo_XIII.pdf

Entrega_LookeaMe_Anexo_XIV.pdf
Entrega_LookeaMe_Anexo_XV.pdf
Entrega_LookeaMe_Anexo_XVI.pdf
Entrega_LookeaMe_Anexo_XVII.pdf
Entrega_LookeaMe_Anexo_XVIII.pdf

Anexos:

Anexo	Temática
Anexo_I	Encuestas
Anexo_II	Gestión de riesgos
Anexo_III	Herramientas
Anexo_IV	Lista preliminar de requerimientos
Anexo_V	Métricas del producto
Anexo_VI	Minuta de reunión
Anexo_VII	Plan de pruebas
Anexo_VIII	Plan de pruebas de usabilidad
Anexo_IX	PSP
Anexo_X	SCRUM
Anexo_XI	Plan de negocios
Anexo_XII	Manual de usuarios
Anexo_XIII	Estándares de diseño
Anexo_XIV	Plataforma multi-lado
Anexo_XV	Investigación
Anexo_XVI	ESRE
Anexo_XVII	Estándares de codificación
Anexo_XVIII	Mock-ups

8 El negocio

8.1 La oportunidad

Los clientes plantearon que hubo dos etapas claves de la historia reciente en la que marcas de moda que eran competencia directa, se convirtieron en aliados estratégicos dentro de un ecosistema de armonía. La primera fue cuando descubrieron que aproximándose geográficamente en una misma calle o cuadra obtenían más flujo de visitas, como es el caso de las Avenidas 18 de julio y 8 de octubre de la ciudad de Montevideo. La segunda etapa fue el modelo de los *Shoppings* o Centros Comerciales [34], grandes concentradores de consumidores de moda.

Desde el surgimiento de Internet en la década de los 90 a la actualidad, concluyeron que no se ha llegado a una tercera etapa de alianza. Esto puede deberse a que cada marca piensa individualmente, desarrollando sitios Web y estrategias no colaborativas, cuando la Web les podría brindar la gran oportunidad de evolucionar hacia un espacio que reúna a todas las marcas y beneficie al conjunto, al igual que lo hicieron las Avenidas y los Centros Comerciales en las etapas anteriores.

En la actualidad, si bien los *Shopping* son sumamente redituables, están entrando en una crisis de disconformidad de los consumidores. Luego de realizar una primera encuesta a 140 posibles consumidores (véase Anexo I), se obtuvo que el 70% de los entrevistados estén disconformes con la saturación de personas que se produce dentro de los *Shopping*, asociando además el problema del estacionamiento vehicular. En consecuencia, un promedio un de 80% de dichos consumidores no concurren más de 2 veces por mes al *Shoppings*, e incluso menos.

De la misma encuesta se desprende que el 34% de los entrevistados realizan la compra en los primeros tres locales que recorren, mientras que el 41% afirma que para realizar la compra que mejor satisfaga su necesidad, debía recorrer toda las tiendas. Si bien el 34% anteriormente mencionado era mayoritariamente integrado por hombres, se puede observar que existe otro 41%, que si bien recorre todas las tiendas, confiesa que este proceso se convierte en una experiencia tediosa y aburrida, lo que teóricamente debiera ser, como experiencia de consumo, una sensación placentera.

En una entrevista con el encargado de una de las tiendas más grandes de Uruguay (véase Anexo I), se desprende que si bien las marcas locales experimentan buenas ganancias durante el transcurso del año, las mismas se ven agobiadas de trabajo a la hora de expandirse, sin poder controlar dicha expansión, pudiéndose afirmar que las tiendas hoy en día experimentan tiempos de estancamiento, sin poder realizar proyecciones más grandes.

Las tiendas buscan expansión y oportunidad de seguir creciendo ubicándose en los lugares de mayor concentración de consumidores. En consecuencia, las etapas

mencionadas anteriormente son complementarias, buscando expandirse en el mercado y no sustituir un local por otro.

8.2 La solución

Lookea.Me nace con la oportunidad de mejorar la experiencia de consumo de los usuarios, estableciendo una tercera etapa de alianza: la unión de tiendas en la nube. La propuesta reúne a las tiendas en un solo sitio con el fin de explotar las posibilidades que ofrece la Web, expandiendo sus canales de comunicación y comercialización.

El objetivo de Lookea.Me es proveer un portal donde las tiendas puedan unirse todas en un solo lugar, de forma de lograr mayor concentración de consumidores que en una tienda física. De esta manera el consumidor podrá obtener la información de las vitrinas de todas las tiendas en un solo sitio, pudiendo hacerlo desde su casa, trabajo o desde cualquier dispositivo móvil en cualquier momento del día sin depender de los horarios de las tiendas.

8.2.1 Beneficios para las tiendas:

- Aumento de cantidad de potenciales compradores observando sus prendas.
- Aumento del tiempo en el cual puede ser visto y vender, siendo 24x7 ya que es Web.
- Venta personalizada a sus clientes habituales.
- Obtención de información estadística de sus compradores.
- Obtención de comentarios e información relevante por parte de los compradores para lograr mejorar y proveer mejores prendas a futuro.
- Aplicación del principio del software como servicio (“SaaS”), ya que las tiendas darían uso a una plataforma propia de Lookea.Me, donde ésta es la responsable de recibir las peticiones de los consumidores y almacenar todos los datos.
- Venta on-line. Tienen la oportunidad de utilizar Lookea.Me y su plataforma para crear un nuevo canal de venta.

8.2.2 Beneficios para los consumidores:

- Recorrer las tiendas desde su casa, sin necesidad de ir físicamente a ellas y obviarse los problemas de saturación.
- Compra on-line.
- Obtención de un beneficio directo pudiendo comprar on-line desde la casa o cualquier lugar sin tener que ir a la tienda, y realizarlo a cualquier hora y cualquier día, evitando o minimizando el traslado físico.

- Contar con un sitio de vitrinas actualizadas, pudiendo seguir la moda constantemente desde su casa.
- Recibir noticias por parte de las tiendas y futuras prendas a llegar.
- Acceder a las vitrinas de todas las tiendas, siendo de forma indiferente si la tienda pertenece a un *Shopping*, una cuadra de tiendas o asilada en otro lugar.

8.3 Soluciones existentes

Lookea.Me formará parte de la industria de la moda creando una plataforma de servicios, en la cual las tiendas lograrán establecer vínculos con los usuarios y consumidores, previamente desconocidos. En el mundo y la región existen soluciones informáticas que de forma similar, logran solucionar ciertos problemas u oportunidades de mejora detectadas anteriormente (véase Anexo XV). Durante el transcurso de la Ingeniería de Requerimientos, se realizaron estudios detallados de por qué Lookea.Me es diferente a dichas soluciones informáticas, estableciendo las ventajas que tiene Lookea.Me frente a las mismas.

Vitryn [28], de origen chileno, es una comunidad de personas buscando todo tipo de productos considerados “a la moda”. Dichos productos son de tiendas o marcas no oficiales, siendo estos publicados en sus respectivos sitios Web y referenciados por Vitryn (Figura 2).

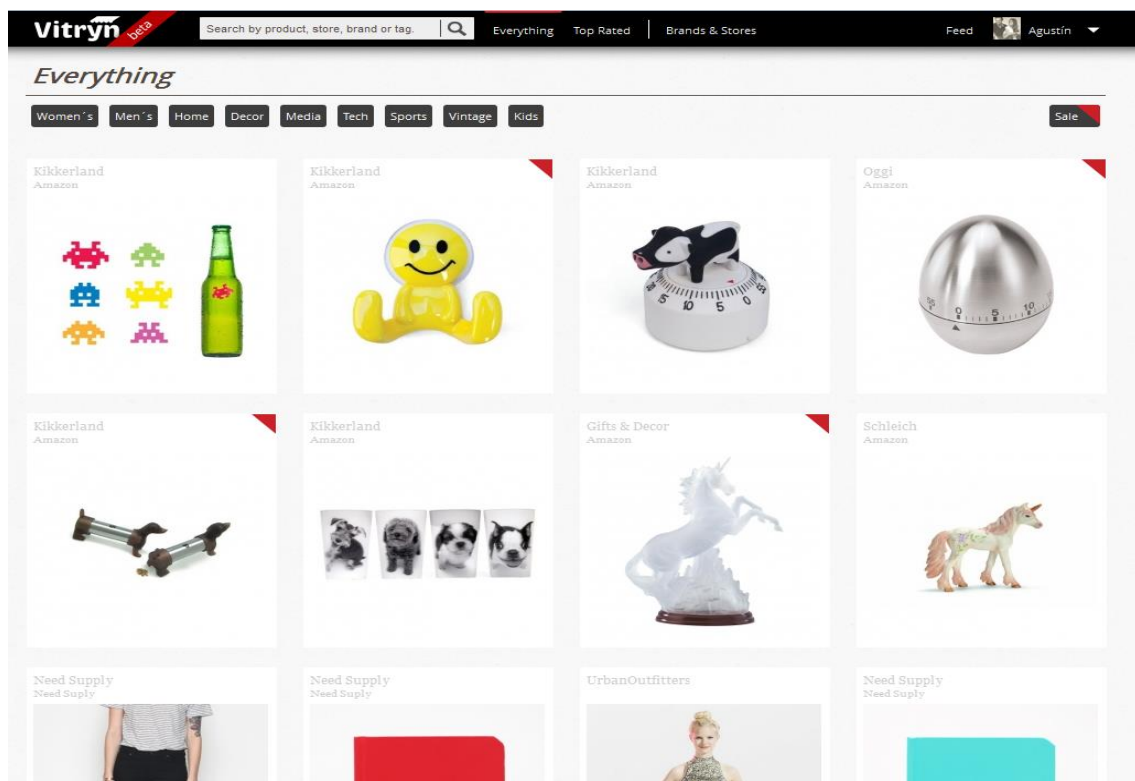


Figura 2 – Página principal de Vitryn

Uno de los sitios más referenciados por Vitryn es Amazon [51], donde se pueden encontrar artículos no solo de moda, sino de hogar, electrónicos y muchos más. Es importante destacar que éstos son previamente filtrados por los integrantes del equipo de Vitryn. De esta forma, los usuarios de Vitryn podrán únicamente buscar artículos que son considerados “a la moda” según el criterio del equipo y no el criterio propio de cada usuario.

Vitryn es un sitio que ofrece una gran cantidad de artículos a nivel mundial, ya que se publican artículos de todas partes del mundo. En el ejemplo anterior se menciona Amazon, pudiendo de esta forma comprar productos del exterior con envío al país correspondiente, además de poder comprar artículos de marcas inexistentes en dicho país.

Uno de los principales sitios a tener en cuenta es el sitio llamado Dafiti [29]. El mismo inició como un proyecto de moda en Brasil, expandiéndose a Colombia, México, Argentina y Chile. Dafiti (Figura 3) opera como una vitrina en línea de marcas líderes mundialmente conocidas, donde su objetivo es poder llegar al público atraído por la moda e Internet de forma conjunta. Dafiti intenta atraer a los consumidores que no desean perder tiempo buscando lugar para estacionar en los *Shoppings* o realizar compras en tiendas saturadas de gente.

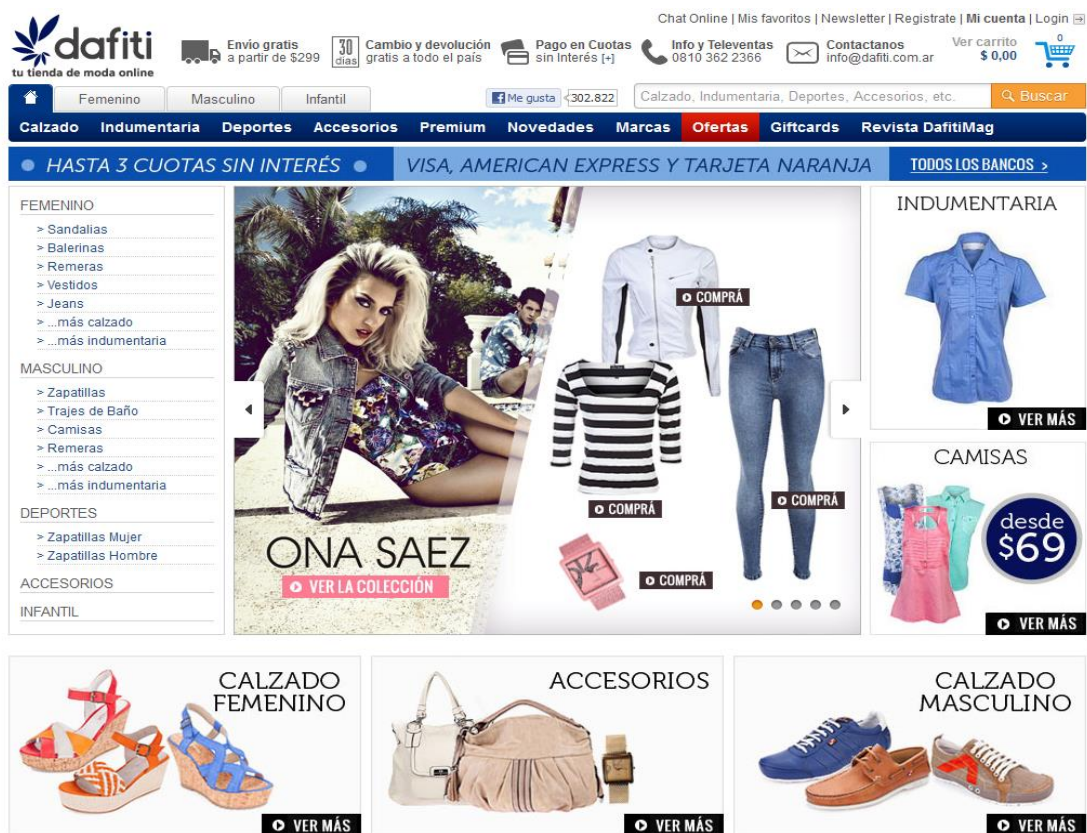


Figura 3 – Página principal de Dafiti Argentina

Sin embargo, Dafiti es un sitio de compra de productos tales como zapatos, ropa u accesorios de moda, estableciendo de esta forma un nuevo canal de ventas. Éste cuenta con un stock propio de artículos a ser vendidos encargándose también de la distribución de los mismos. Provee venta personalizada pero no cuenta con el concepto de comunidad, enfocándose en la venta y distribución masiva, sin formar una comunidad asociada a Dafiti.

Así como existe Dafiti en la región, también existen otros sitios similares en el mundo tales como Dress-For-Less [26] o Etsy [25]. Estos sitios son similares a Dafiti y Vitryn pero orientados al mercado Europeo. Dress-For-Less es de origen Alemán pero rápidamente se expandió a España, Francia e Italia, entre otros. En la actualidad Dress-For-Less se encuentra en más de 20 países, en su mayoría europeos. El modelo que utilizan es similar a Dafiti, en el cual se centran en la venta de productos de moda masivo con poca personalización por parte de los usuarios.

Si bien los sitios anteriormente mencionados son de publicaciones masivas de artículos, también existen otros enfocados a cierto público en particular. Entre estos sitios se pueden encontrar Urban-Outfitters [31] o Forever21 [30], donde ambos tienen un público en específicamente definido, de forma que no se publican todas las marcas sino un conjunto selectivo de las mismas. En el caso de Forever21, además de ser un sitio con cierto estilo de moda definido, el mismo está orientado únicamente a mujeres.

Finalmente, es importante destacar el sitio Web Tienda Nube [27]. El mismo tiene un enfoque de mercado orientado a nuevas empresas o diseñadores independientes y no a la venta de marcas conocidas. A su vez, provee formas de pago y comunicación entre el comprador y el vendedor de forma de no participar en el proceso de envío de mercadería. De esta forma, Tienda Nube es una plataforma en la cual cada diseñador independiente puede crear un sitio Web para vender y distribuir sus artículos.

Si bien Tienda Nube no se asemeja con el modelo de Lookea.Me, igualmente fue investigado ya que resulta importante evaluar cómo funciona la plataforma para la incorporación de artículos por parte de los diseñadores, ayudando a comprender mejor un problema clave: la carga masiva de artículos a ser publicados en la Web de Lookea.Me.

En conclusión, es importante evaluar todos los sitios anteriormente mencionados ya que de alguna manera u otra, Lookea.Me puede verse beneficiado por sus oportunidades de éxito y posibles mejoras que tiene cada uno de ellos. Independientemente de la forma que cada sitio trabaja, todos están vinculados a la industria de la moda y de todos se pueden destacar sus fortalezas y debilidades.

De la investigación de los sitios existentes surgieron requerimientos nuevos asociados a funcionalidades en los dichos sitios. Un ejemplo de esto fue el concepto de “Lo Quiero”/“Lo tengo” del sitio Vitryn, donde cada usuario puede establecer que artículos tiene y cuales quiere, pero aún no lo tiene. Si bien esta es una funcionalidad específica

de Vitryn, existen conceptos similares, entre otros las “Wish List”, donde los usuarios de Lookea.Me podrán establecer un conjunto de artículos que desean para su cumpleaños, festivo o cualquier otro día.

Uno de los principales objetivos de Lookea.Me es proveer un sitio que una la moda y las redes sociales tales como Facebook [35] o Twitter [36]. Algunos de los sitios anteriormente mencionados establecen conexiones con redes sociales, por lo que fue de suma importancia relevar el éxito que tuvo cada sitio con dicho aspecto. A su vez, fue interesante relevar aspectos de experiencias de otros usuarios en los sitios anteriormente mencionados, ajustando ciertos requerimientos según las reacciones y comentarios de los usuarios.

En base al estudio realizado, se tomó como referencia de comparación a Vitryn, ya que dentro de los sistemas analizados, el modelo de negocios de éste es el que más se asemeja al modelo de Lookea.Me, dado que los demás incluyen la compra y distribución de artículos. El análisis comparativo de funcionalidades entre sistemas se presenta en la tabla 1:

Funcionalidad	Vitryn	Lookea.Me
Usuario propio del sistema		X
Integración con Redes Sociales	X	X
Navegación sin registro		X
Reserva		X
Compra online		
Compra con re-dirección a otro sitio	X	X
Perfil propio de usuario	X	X
"Mi Colección"	X	X
"Me Gusta"	X	X
Wish Lists		X
Búsqueda con filtros avanzados		X
Categorización de prendas	X	X
News Letter	X	X
Seguir a otros usuarios	X	X
Seguir a las tiendas/marcas preferidas	X	X
Ofertas de toda la moda		X
Venta cruzada	X	X
Venta de artículos de tiendas locales		X
Reviews de artículos	X	X
Reviews de tiendas/marcas		X
Feedback	X	X
Publicación por parte de las tiendas		X

Tabla 1 - Estudio comparativo de funcionalidades Vitryn vs. Lookea.Me

8.4 Arquitectura del negocio

Como se mencionó anteriormente, el negocio de la moda tuvo dos etapas en las cuales las tiendas de moda se convirtieron en aliadas estratégicas. Tanto en la primera como en la segunda etapa, las tiendas se juntaron con un solo objetivo: aumentar el flujo de visitas. Sin consumidores y sin flujo de visitas, las tiendas no sobreviven en el negocio. Es por este motivo que el objetivo de cada tienda es ubicarse en los lugares donde haya más concentración de consumidores posibles.

Lookea.Me surge como un tercer concentrador de consumidores, y en consecuencia, es una nueva oportunidad para las tiendas de seguir expandiéndose. A diferencia de las cuadras o los *Shoppings*, Lookea.Me es un concentrador virtual, por lo que el límite de tiendas conviviendo en un solo lugar es ilimitado. De esta forma, Lookea.Me tiene el potencial para ser el mayor concentrador de consumidores, publicando la oferta de las tiendas de cualquier lugar físico y siendo visitado por consumidores de cualquier lugar de la región.

8.4.1 Estructura de las tiendas

En la actualidad, las tiendas que se puede encontrar en los *Shoppings* son con frecuencia las mismas que están en las cuadras de moda. Sin embargo, pueden no ser exactamente las mismas tiendas, siendo estas independientes entre ellas donde cada una es una franquicia de la marca, como concesión de los derechos de los productos o la marca otorgados por una empresa a una o varias personas en una zona determinada [37]. En consecuencia, estas pueden ser técnicamente dos tiendas diferentes, de forma que cada una de ellas tenga su estructura y modelo independiente, y por sobre todo, otro stock de artículos.

Es así que Lookea.Me debe considerar ambas estructura de tiendas posibles. Por un lado, una misma marca que tiene varios locales, tanto en los *Shoppings* como en las cuadras, mientras que otras tiendas pueden ser técnicamente diferentes empresas, pero la misma marca, por lo que la sincronización de artículos y publicación de los mismos debe considerar dichos aspectos.

Existen en la actualidad tiendas de moda que facturan y trabajan en computadoras sin acceso a Internet las 24 horas del día. Si bien esto no afecta al negocio en sí mismo, si afecta a la estructura de Lookea.Me, debiéndose adaptar a dicha forma de trabajo. Desde un inicio se obtuvo a partir del análisis que Lookea.Me debía ofrecer una modalidad de trabajo off-line para luego realizar determinada sincronización con los servidores donde se administra la información.

Lookea.Me debería ofrecer todos los servicios para publicar artículos de las tiendas en la Web en modalidad on-line, pero deberá realizarlo de forma tal que no afecte con el trabajo diario que realizan las tiendas. El método de publicación de artículos debería ser

muy eficiente, además de considerar los aspectos del negocio anteriormente mencionados. En consecuencia el trabajo off-line toma un rol importante en Lookea.Me, debiendo ser implementada alguna solución alternativa a la modalidad on-line.

8.4.2 Sector

Lookea.Me participa de la industria de la moda, siendo éste un nuevo canal de comunicación y publicidad para los que los usuarios acceder a las ofertas más destacadas de la moda. En Uruguay, dicha industria mueve aproximadamente U\$600 millones al año, donde 28 de ellos son importados.

En base al análisis de mercado realizado por el grupo comercial de Lookea.Me, se detectaron aproximadamente 160 potenciales clientes, de los cuales 90 se encuentran dentro de los centros comerciales más importantes. El objetivo inicial es lograr el involucramiento de las grandes y medianas tiendas que están por fuera de estos centros comerciales, logrando un rápido reconocimiento y una posibilidad de entrar en el mercado de los *Shoppings*.

En cuanto al público objetivo, Lookea.Me está orientado a mujeres de entre 15 y 50 años de edad de nivel socioeconómico medio y alto, que además se conectan regularmente a Internet. Según el Estudio General de Medios de Equipos Mori [53], la cifra equivale un total de 268.966 mujeres.

Si bien los consumidores muestran interés por la compra on-line de artículos, el sector presenta cierto matiz, dado que la compra on-line de la moda se dificulta en varias zonas del país. Visualizar, sentir y probarse las prendas, son actividades primordiales en el proceso de compra de artículos de moda, causando que la compra on-line no sea una característica clave de la solución que presenta Lookea.Me.

Para evaluar en detalle el modelo de negocios de Lookea.Me, a continuación se presenta el CANVAS (Figura 4) junto con los 9 bloques claves del negocio (véase Anexo XI):

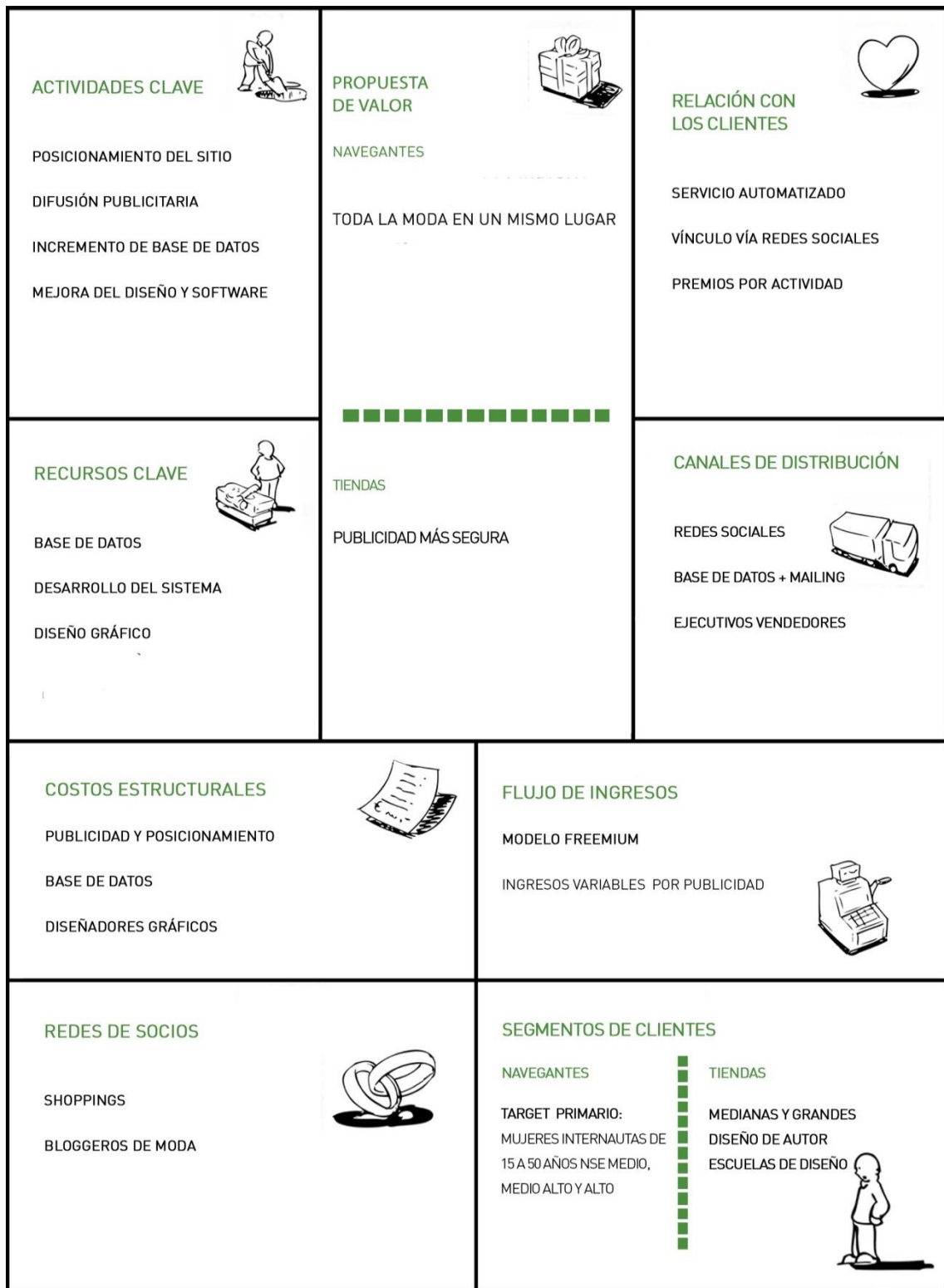


Figura 4 – CANVAS de Lookea.Me

9 Descripción del proyecto

En este capítulo se analizará en detalle la solución propuesta, estableciendo un alcance del sistema para la duración del proyecto. A su vez, se detallarán las características más importantes del sistema y su contexto, describiendo brevemente las funcionalidades más destacadas del producto.

Finalmente, se presentará una descripción resumida de la forma de trabajo del grupo durante los 12 meses del proyecto, realizando un primer análisis del proceso de desarrollo y sus perspectivas.

9.1 Alcance del sistema

Como se mencionó anteriormente, el producto a desarrollar formó parte del emprendimiento de José Miguel Majó y Juan Manuel Dellepiane, donde el alcance era mayor al que se puede obtener en un proyecto académico de doce meses. Por este motivo, se debieron priorizar los distintos requerimientos obtenidos, para luego establecerse cuáles serían tomados para el producto del presente proyecto y cuáles quedarían para futuras etapas.

Si bien ciertas funcionalidades quedaron propuestas para futuras etapas, el diseño del producto debió ser adecuado a que las funcionalidades no incluidas fueran fácilmente desarrolladas en etapas posteriores, sin afectar en gran escala al producto ya desarrollado.

En consecuencia, a continuación se detalla una descripción general del sistema y su contexto, estableciendo aquí algunos aspectos importantes en cuanto al alcance, como por ejemplo, los distintos entregables que se debieron realizar y el contenido planificado de cada uno.

Finalmente se establece una descripción resumida de las funcionalidades más importantes que el sistema debe tener para tener éxito en el mercado, y para que sea además considerado un proyecto de grado completo por parte de la Universidad ORT Uruguay.

9.1.1 Descripción del sistema y su entorno

Lookea.Me es una solución que une todas las vitrinas de moda en un solo lugar para que las mismas puedan ser consultadas por parte de los consumidores vía Web desde su hogar, oficina o dispositivo móvil.

Siendo una plataforma de tipo “multi-lado” (véase Anexo XIV), Lookea.Me es de valor para un grupo de clientes si otros grupos de clientes también están presentes, donde la ausencia de uno de los grupos quita valor al producto. De esta forma, los grupos de clientes mencionados que deben coexistir son las tiendas de moda, y por el otro lado los usuarios de la red.

Como plataforma multi-lado, Lookea.Me crece en valor en la medida en que se atraigan más usuarios y tiendas a la plataforma, aumentando el flujo de información para crear de esta manera un efecto red.

Lookea.Me está concebido como un conjunto de sistemas que interactúan sobre un modelo de datos centralizado. De esta forma, se tendrá un sistema para cada grupo de usuarios de Lookea.Me: uno para las tiendas y otro para los usuarios, donde ambos sistemas se sincronizarán mediante una base de datos centralizada.

9.1.2 Descripción resumida de las funcionalidades

Habiendo establecido un sistema para cada grupo de usuarios, las principales funcionalidades de cada uno son las siguientes:

9.1.3 Tiendas

- Publicar artículos de sus vitrinas.
- Mantener artículos publicados pudiendo modificar sus datos, imágenes e incluso eliminarlo.
- Personalización de su perfil, estableciendo dirección y teléfonos de sus respectivos locales.
- Asociar artículos a los locales donde se pueden encontrar.
- Publicar descuentos colectivos.

9.1.4 Usuarios

- Registrarse a Looke.Me mediante redes sociales como Facebook y Twitter o creándose un usuario propio de Looke.Me.
- Buscar prendas de interés estableciendo filtros personalizados, pudiendo seleccionar sexo, color, talla, tienda o marca, ubicación o rango de precios deseados.
- Seguir a tiendas u otros usuarios para saber cuál es lo último de la moda.
- Mantener un registro de las tiendas y usuarios que un usuario sigue, así como las prendas que son de su interés. Todos estos datos están dentro de una sección llamada “Mi Closet”.
- Realizar reserva de los artículos, seleccionando color y talla.
- Publicar en las redes sociales artículos que su “Closet” o cualquier otro artículo que el usuario desee.

9.2 Descripción de la forma de trabajo

Ciertos aspectos del grupo y su organización son relevantes a la hora de realizar un análisis profundo de la situación y los procesos a utilizar.

El proyecto constaba de dos puntos a ser estudiados de forma independiente, donde uno es el proyecto en sí mismo, y por el otro lado, el emprendimiento junto con Juan Manuel Dellepiane y José Majó. Si bien el proyecto formó parte del emprendimiento, es importante distinguir en qué momento debieron ir alineados y en qué momento debieron tratarse como independientes.

En los comienzos del proyecto se trabajó como un grupo de cuatro integrantes, donde los restantes dos eran los impulsores de la idea: Juan Manuel Dellepiane y José Miguel Majó. Durante estas primeras etapas se hizo énfasis en analizar a fondo la idea y sus características, donde además, el proyecto fue presentado al CIE como idea innovadora [38].

Una vez que el proyecto fue analizado y presentado en el CIE, se generó el primer hito importante definiéndose en detalle el sistema a desarrollar y dividiendo el grupo de tesis por un lado y los impulsores de la idea por otro. De esta manera se definió el grupo de proyecto integrado por Agustín Alonso y Máximo Halty, y los impulsores de la idea pasaron a formar parte del proyecto como clientes. Dichos clientes se transformaron en el grupo comercial de Looke.Me, encargados del proyecto en el CIE.

Habiendo culminado la etapa de investigación de negocio y requerimientos, comenzaron las etapas de programación y desarrollo del producto en sí mismo. En estas etapas, los mencionados clientes tomaron la decisión de apartarse del desarrollo técnico

de la aplicación, enfocándose en el modelo de negocio para luego poder realizar el plan de negocio.

Finalmente, luego de publicar el prototipo en la Web para que el mismo pudiera ser probado y validado por los usuarios, los clientes volvieron a tomar un rol más participativo en el proyecto. Durante la finalización del proyecto, los clientes verificaron los requisitos establecidos en etapas tempranas del proyecto, realizando ciertas críticas constructivas del sitio, mayoritariamente referidas al diseño gráfico de la aplicación. De forma simultánea, el grupo de tesis dedicó las últimas semanas a finalizar con todos los documentos necesarios como requisitos de la entrega final, entre otros: el registro de los resultados de las pruebas por parte de los usuarios, escritura de manual de usuario, manual de instalación y documentación de las proyecto propiamente dicha.

9.3 Proceso de desarrollo

9.3.1 Introducción

A continuación se realizará una descripción del marco metodológico para así comprender en detalle el proceso de desarrollo seleccionado. El mismo se presentará junto con sus con las 5 perspectivas: funcional, organizacional, metodológica, resultados y el modelo de ciclo de vida, junto al ciclo de vida instanciado.

9.3.2 Marco metodológico

9.3.2.1 Grupo

En primera instancia, es importante resaltar que el grupo de proyecto estuvo formado únicamente por dos estudiantes, por lo que el mismo debió manejarse de tal forma que pudiera explotar las ventajas de ser pocos y mitigar los riesgos que surgieron de esta situación.

A su vez, el grupo comercial de Looke.Me contó con dos integrantes que no son estudiantes de ingeniería, y por ende no formaron parte del grupo de proyecto en sí mismo. Aun así, colaboraron con la parte comercial y estratégica del sistema, cumpliendo de esta manera su rol de clientes.

9.3.2.2 Clientes

Cabe resaltar la naturaleza y clasificación de los distintos clientes del proyecto, ya que cada uno de ellos debe debió gestionarse de forma diferente a causa de sus características y cercanías al grupo del proyecto.

El primer cliente y referente para la tesis de grado fue el grupo comercial de Lookea.Me, integrado por José Miguel Majó y Juan Manuel Dellepiane. Dicho cliente a su vez tuvo otros dos clientes por los cuales debió responder: las tiendas de moda y los usuarios. Ambos clientes fueron quienes definieron las características del producto, tanto el portal Web como el back-end.

Si bien para el grupo de proyecto, Majó y Dellepiane fueron los clientes, las tiendas y los usuarios influyeron sobre el proyecto, por lo que no se dejaron de gestionar. El grupo comercial fue quien canalizó esta influencia, representando a dichos clientes sobre el grupo de proyecto (Figura 5).

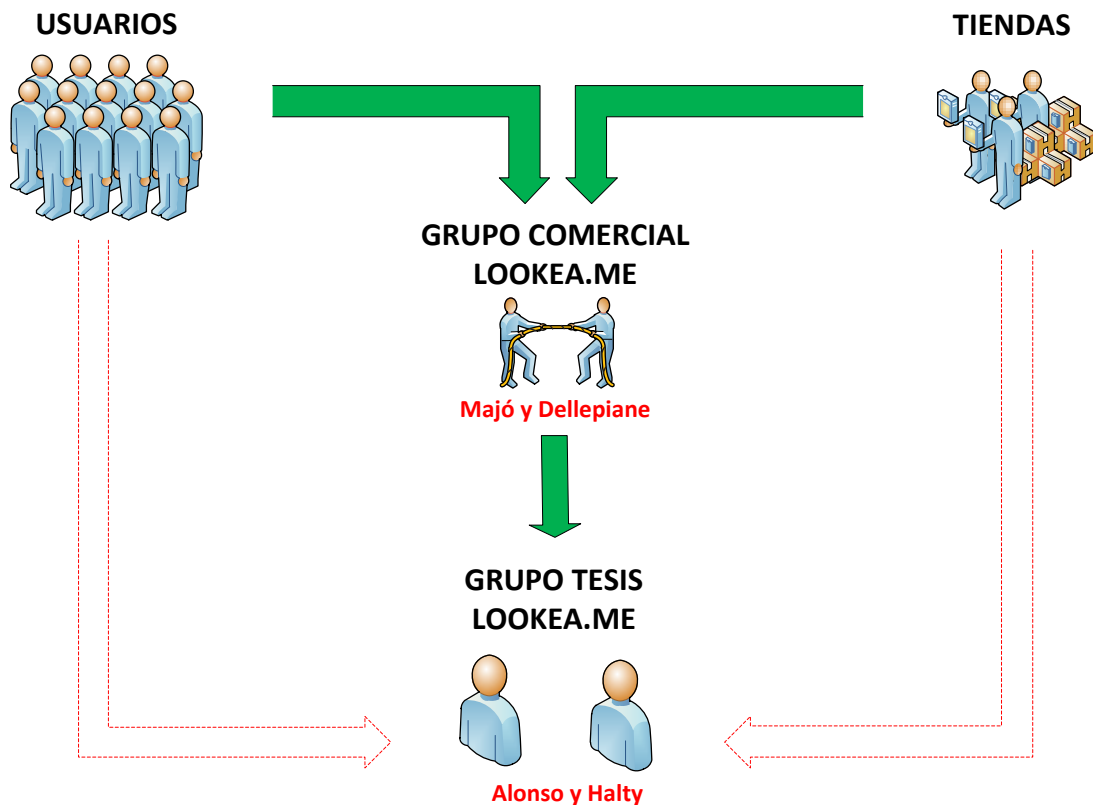


Figura 5 - Influencias sobre el equipo

En cuanto a los usuarios, se contó con facilidad para comunicarse con ellos debido a la cercanía con el grupo de tesis. A su vez, los usuarios no solo eran una cantidad mucho mayor que las tiendas, y por ende más fácil de acceder para obtener información, sino

que muchos de ellos eran conocidos por el grupo. Por este motivo existió una comunicación fluida con los usuarios en todo el transcurso del proyecto, garantizando información confiable y transparente.

La situación de las tiendas fue opuesta a la de los usuarios, donde no sólo eran una cantidad significativamente menor, sino que el tiempo y la información que podían ofrecer al grupo era escasa para la magnitud del sistema, incluso pudiendo no ser completamente transparente su participación debido a aspectos relacionados con la confidencialidad y beneficios propios de las tiendas.

Si bien la parte visible de Looke.Me era el sitio Web con las vitrinas, es importante el sistema back-end para que las tiendas puedan publicar sus artículos en la Web. Existen dos sistemas separados pero dependientes entre sí, ya que el sitio Web de usuarios toma la información provista por las tiendas en el back-end.

Las funcionalidades de cada sistema debieron ser relevadas con cada cliente, siendo necesario que el proceso considerara que las tiendas no podían ser consultadas en cualquier momento. En consecuencia, se hizo uso de este recurso de forma limitada, lo cual influyó de forma directa sobre la forma de realizar la ingeniería de requerimientos del sistema back-end.

9.4 Metodología tradicional y SCRUM

La aplicación de una metodología tradicional pura sin la consideración de metodologías ágiles para las características del grupo anteriormente mencionada, no sería la adecuada a seleccionar, teniendo en cuenta el apoyo que estas últimas pueden ofrecer para obtener ventaja de los posibles riesgos que puedan ocurrir, entre otras consideraciones relevantes, por ser un grupo pequeño.

El grupo basó su proceso en SCRUM, aplicando su metodología en el proceso de desarrollo de software para grupos pequeños, con alto grado de comunicación, adaptabilidad y colaboración, donde además supone que el cliente está altamente disponible.

El uso de metodologías ágiles tiene sus desventajas, como por ejemplo, no enfocarse en realizar una documentación completa y exhaustiva, aspecto imprescindible para la índole del presente proyecto. Siendo importante considerar que el proyecto debía ser entregado como Tesis de grado para el título Ingeniería de Sistemas de la Universidad ORT Uruguay, donde se exigen estándares específicos, este aspecto resultaba ineludible.

Por esta razón, si bien SCRUM sería la base de la metodología durante el proceso de desarrollo, la misma debió ser adaptada sobre el uso de metodologías tradicionales, de

forma de encontrar un equilibrio adecuado entre las ventajas y desventajas de cada metodología por la conformación descrita anteriormente.

9.5 Procesos de ingeniería

A continuación se detallarán las perspectivas del proceso, incluyendo el ciclo de vida utilizado para el proyecto.

9.5.1 Perspectiva del comportamiento

La disponibilidad por parte de ciertos clientes fue acotada a la hora de realizar una ingeniería de requerimientos, debiendo dicha actividad ser realizada de forma exhaustiva y a fondo únicamente en las primeras etapas, por lo que el ciclo de vida utilizado para el proyecto fue el incremental (Figura 6).

Se realizó en una primera etapa la investigación y la ingeniería de requerimientos, dividiéndose posteriormente en módulos las etapas de desarrollo, implementación y prueba. La ejecución de estas últimas etapas se basó en metodologías SCRUM, adecuando ciertos aspectos a metodologías tradicionales. Luego de culminadas estas tres tareas junto con sus iteraciones, se realizó la tarea de puesta en producción, de forma de dar por terminado el proceso de desarrollo.

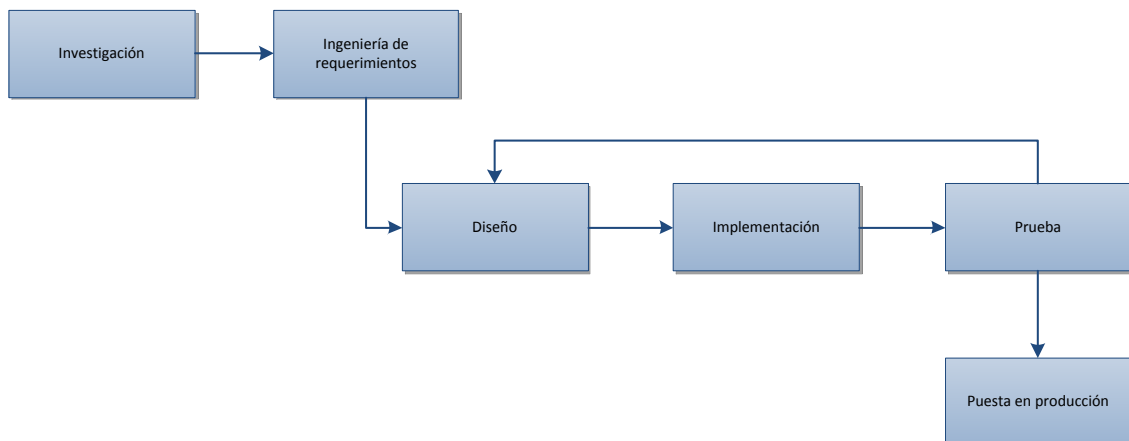


Figura 6 - Ciclo de vida

9.5.2 Perspectiva funcional

El proceso seleccionado incluyó las siguientes actividades:

- Investigación.
- Ingeniería de requerimientos.
- Diseño.
- Implementación.
- Prueba.
- Puesta en producción.

El sistema se basó en un modelo de negocios que ninguno de los dos integrantes del grupo conocía y en consecuencia, la investigación fue una actividad muy importante para comprender el entorno en el cual se trabajó (véase Anexo XV). Para poder modelar un sistema exitoso era necesario conocer el estado de la práctica a nivel del negocio y cómo el sistema influiría en él, transformando ciertos aspectos del mismo. Durante la investigación se emplearon distintos métodos, entre otros, encuestas, entrevistas y análisis de modelos de negocios existentes. Finalmente, en base a los datos obtenidos se logró realizar la especificación de requerimientos, analizando además las prioridades de cada requerimiento.

El diseño de la aplicación fue dividido en dos grandes partes. Por un lado, antes de comenzar con el desarrollo de la aplicación, se realizó un diseño a nivel macro de la arquitectura y el diseño de la base de datos. Luego, el diseño lógico a bajo nivel fue distribuido en las distintas iteraciones del desarrollo, estableciendo el diseño para los requerimientos a ser desarrollados en cada iteración.

En conjunto con el diseño por iteraciones, también se realizó el desarrollo de los requerimientos junto con sus posteriores pruebas. Una vez terminado cada módulo se realizaban pruebas de integración para verificar el correcto funcionamiento de todas las partes juntas.

Posteriormente, se realizó la puesta en producción una vez que el sistema estuvo terminado, donde se migró desde el servidor de prueba a los servidores de producción, permitiendo el acceso público.

9.5.3 Perspectiva organizacional

Si bien para un proyecto de software las buenas prácticas recomiendan tener un responsable por cada área relevante, para la naturaleza del proyecto esto no pudo ocurrir debido a que el grupo estuvo compuesto por 2 integrantes y eran las 7 áreas de proyecto relevantes a gestionar: Gerencia, SCM, Ingeniería de Requerimientos, Arquitectura y Diseño, Desarrollo, Pruebas y SQA (Tabla 2).

Para mitigar posibles problemas en la planificación y ejecución del proyecto, ambos integrantes realizaron el curso “Prácticas de mejora en la Ingeniería de Software”, el cual se enfocó en la metodología PSP propuesta por CMMI para asegurar la calidad en el proceso de desarrollo personal. A su vez, dicho curso tuvo una introducción a la metodología TSP para la gestión de calidad en equipos de desarrollo. Por esta razón, se decidió establecer un plan de calidad con sus respectivas métricas pero sin asignar un responsable del área SQA, ya que ambos integrantes están fuertemente capacitados para gestionar la calidad según PSP y TSP.

Se estableció una división de las restantes áreas del proyecto donde los responsables de cada área se muestran a continuación. Como consecuencia de ser un grupo formado por 2 integrantes, no fue necesario especificar quien era el suplente del área.

Área del proyecto	Responsable
Gerencia	Agustín Alonso
SCM	Máximo Halty
Ingeniería de Requerimientos	Agustín Alonso
Arquitectura y Diseño	Máximo Halty
Desarrollo	Agustín Alonso
Pruebas	Máximo Halty

Tabla 2 - Responsables de cada área

Si bien el grupo estuvo integrado por dos integrantes, a lo largo del proceso aparecieron tres colaboradores claves para el desarrollo del sistema: los dos creadores de la idea y el tutor. Cada una de estas personas cumplió un rol importantísimo en el proceso y por eso fueron importantes al momento de describir la perspectiva organizacional, representada en la figura 7.

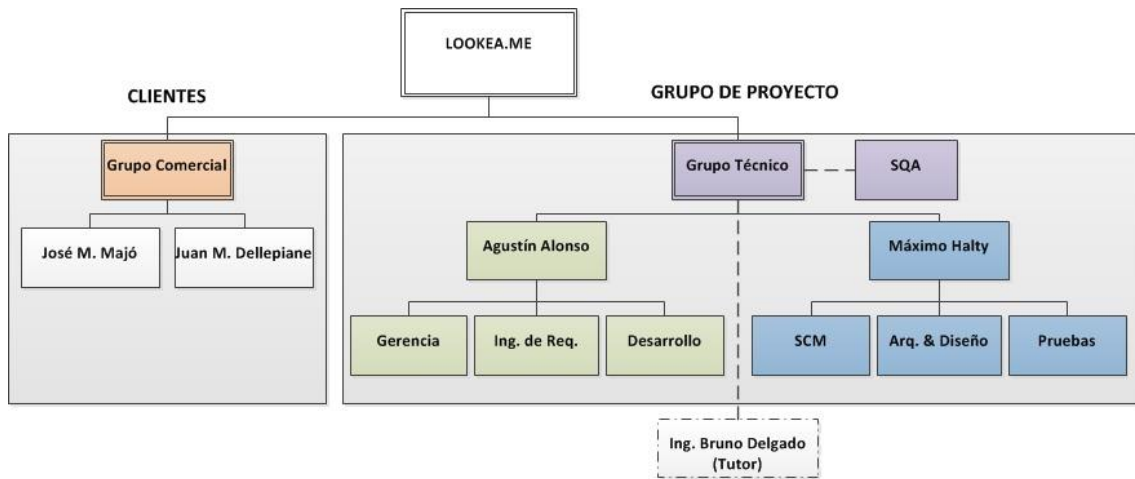


Figura 7 - Perspectiva Organizacional

9.5.4 Perspectiva metodológica

Al comienzo del proyecto, el grupo comenzó trabajando con una herramienta de gestión llamada “ComindWork” [39] (véase Anexo III). En ella, el grupo pudo realizar una planificación en formato Gantt para luego automáticamente poder ingresar el esfuerzo realizado de forma diaria, especificando la tarea del Gantt. Esta herramienta permitía además compartir archivos, y por ende, utilizarla para centralizar los archivos del proyecto. Sin embargo, desde un comienzo se decidió utilizar la herramienta “Dropbox” [40] (véase Anexo III) ya que ambos integrantes tenían previamente una cuenta creada. Finalmente el grupo dejó de usar la herramienta “ComindWork” debido a que la misma quedó sin espacio gratuito, debiendo buscar soluciones alternativas.

En este punto, se decidió exportar el Gantt a formato “Microsoft Project” y el registro de horas en formato “Microsoft Excel”. Uno de los estudiantes realizó la exportación y además realizó un Excel con programación de Macros y tablas dinámicas que permitían realizar el mismo seguimiento de registro de horas que proveía “ComindWork”, que además realizaba gráficas de seguimiento.

La carpeta de documentos en Dropbox fue también compartida con el tutor del proyecto para facilitar correcciones y opiniones, además de poder realizar un seguimiento del proyecto diario. Por su lado, se compartió una nueva carpeta con otros documentos al grupo comercia de Looke.Me, especialmente documentos de negocio y requerimientos del sistema. Finalmente, la misma herramienta fue utilizada para compartir el diseño con la diseñadora gráfica contratada para realizar el diseño del sitio Web de usuarios.

En cuanto a herramientas de comunicación, se utilizaron “Skype” y “Google HangOut” para realizar videoconferencias con los clientes, y de ser necesario, entre los integrantes del grupo de tesis. Por su lado, el uso del mail fue una herramienta utilizada durante el transcurso del proyecto con todos los interesados del mismo. Por este motivo además, se hizo uso de los “Google Groups”, creando uno de ellos para la comunicación entre los integrantes del grupo de tesis y el tutor.

Si bien más adelante se detallarán los aspectos de la tecnología utilizada para desarrollar las aplicaciones, se utilizó como herramienta de desarrollo “Microsoft Visual Studio 2010”, que además se integró con “Microsoft Team Foundation Server (TFS)” para la gestión de versiones del código fuente. A su vez, se utilizó “Microsoft SQL Server 2008 R2” como motor de base de datos.

Para el versionado y la publicación de la Web en etapas de desarrollo se acondicionó una computadora de la casa de uno de los estudiantes para que funcionara como servidor. En etapas de pruebas del sistema, dicho servidor hosteaba las diferentes aplicaciones para que las mismas puedan ser accedidas por URLs públicas y ser probadas desde cualquier lugar.

9.5.5 Perspectiva de resultados

Los sistemas de SW esperados al terminar el proyecto son los siguientes:

- Sitio Web para los usuarios.
- Sitio Web para los administradores del sitio.
- Sitio Web para las tiendas.
- Aplicación de escritorio para la carga masiva de artículos por parte de las tiendas.
- Programas de soporte tales como envíos de correo electrónico y procesos *batch*.

Todos los sistemas utilizan una base de datos centralizada con la salvedad de la aplicación de escritorio de las tiendas, la cual trabaja con una copia local y luego sincroniza los datos al servidor, interactuando las aplicaciones por medio de la base de datos centralizada.

En la figura 8 se presenta el diagrama cómo los sistemas interactúan entre ellos por medio de la base de datos centralizada.

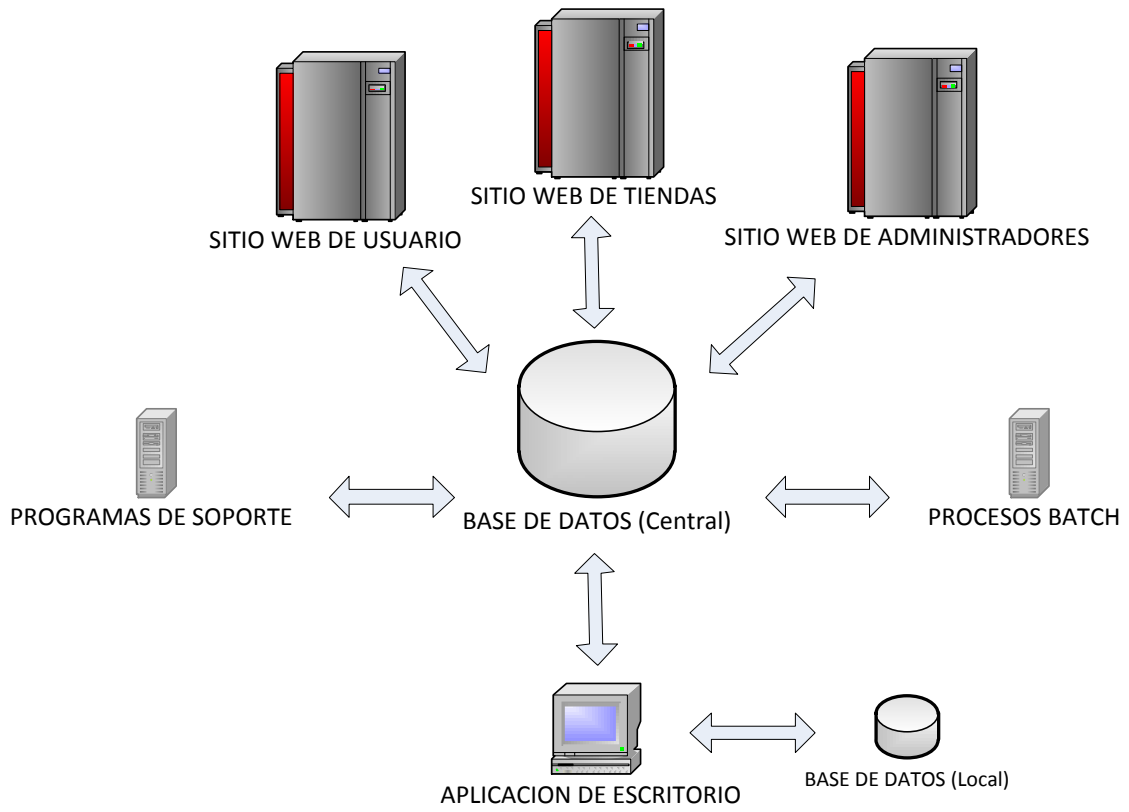


Figura 8 - Arquitectura de alto nivel

10 Requerimientos

10.1 Relevamiento inicial

Desde sus inicios, Looke.Me fue presentada con una lista preliminar de requerimientos basados en una investigación del tipo “*Benchmarking*”, realizada por los propios clientes: José Miguel Majó y Juan Manuel Dellepiane. Durante la presentación de la idea al grupo de proyecto surgieron aspectos inconclusos, entre ellos, el gran problema vinculado al método de subida de contenido por parte de las tiendas de moda, ya que se debía encontrar la manera para que las mismas no fueran consideradas ni realizadas como un trabajo extra.

La presentación del proyecto al grupo se realizó pocos días antes de que el mismo se presentara en el comité de proyectos para que fuera aprobado como tesis de fin de carrera. Por lo que no hubo un relevamiento inicial más del que fue realizado por los mencionados clientes en el anteproyecto.

Luego de analizado el contexto, finalmente se obtuvo que se debía implementar un ciclo de vida incremental, estableciendo una única etapa ingeniería de requerimientos en etapas iniciales del proyecto, realizando posteriormente la investigación y extracción de requerimientos propiamente dicha.

10.2 Técnicas utilizadas

A continuación, se detallarán los distintos métodos utilizados en el proceso de extracción de requerimientos. En este proceso se debieron utilizar diversos métodos debido al contexto y las diferentes características de los usuarios y de los clientes. A su vez, se debe recordar que a esta altura del proyecto, la ingeniería de requerimientos formaba parte de la etapa de investigación, donde además de entender el negocio en sí mismo, había que entender cómo Looke.Me debía actuar para mejorar las distintas cadenas de valor, tanto para las tiendas como para los usuarios.

10.2.1 Reuniones con clientes

Las tiendas de moda son poco accesibles, siendo a su vez dificultoso reunirse con el encargado o supervisor de un local para entender en detalle al negocio. Por este motivo, en general la forma de acceder a las tiendas era por los vendedores, salvo algún contacto directo de tiendas con personas cercanas al grupo.

Se logró contactar con un número reducido de clientes de fuente confiable, cuyos comentarios y expectativas no diferían demasiado entre ellos. Asumiendo este riesgo, el

grupo comercial de Looke.Me optó por validar que las entrevistas realizadas con los clientes eran suficientes para definir los requerimientos de las tiendas.

10.2.2 Encuestas masivas a potenciales consumidores

La mayor diferencia entre el sistema de los usuarios y el sistema de las tiendas, es que el sistema de usuarios es un producto masivo. En consecuencia, el número de usuarios es mucho mayor que el sistema de tiendas, y por ende, los usuarios son más accesibles que las tiendas.

En este escenario, se optó por realizar una encuesta de forma masiva, no solo identificando los requerimientos del sistema a desarrollar, sino también obtener información relevante sobre las características de los potenciales usuarios. Dicha encuesta constó de 10 preguntas de rápida respuesta (véase Anexo 1), en las cuales había entre 3 y 5 posibles respuestas. Los resultados de las mismas superaron las expectativas, no solo en cuanto a la cantidad de respuestas obtenidas, sino también al resultado de las mismas, ya que se pudieron realizar conclusiones basadas en información de fuente confiable.

Se realizaron asimismo algunas entrevistas adicionales con potenciales usuarios para lograr obtener información más detallada y no estadística, como lo es la información obtenida de las encuestas.

10.2.3 Investigación de sistemas similares

Si bien en etapas de anteproyecto fue realizado un estudio del tipo “*Benchmarking*”, el mismo no fue lo suficientemente exhaustivo como para obtener conclusiones seguras y confiables. En consecuencia, se realizó un estudio de sistemas similares existentes, no solo en el Uruguay, sino en la región y en el mundo. A su vez, se identificaron las razones por las cuales cada sistema tuvo éxito o falló en cada país que fue desarrollado (véase Anexo XV).

Se lograron obtener requerimientos basados en las posibles oportunidades de mejora que tuvieron otros sistemas, pero también se obtuvieron ideas exitosas de modo de adaptarlas a Looke.Me. En este punto, se consultaron a ciertos usuarios para obtener sus comentarios sobre qué les gustaba y que no de los sistemas similares.

10.2.4 Investigación de características claves

Uno de los puntos más importantes del sistema de usuarios es la usabilidad que el mismo debe tener, ya que sin un buen “*look and feel*” y la capacidad de auto-aprendizaje, el sistema fracasaría en el corto plazo. Por este motivo, se hizo énfasis en los requerimientos de usabilidad, donde posteriormente se contrató a los diseñadores

gráficos Sebastián Gavary, quien realizó los diferentes logos de Lookea.Me, y Mercedes Laffitte, quien realizó el diseño de la Web de los usuarios. Finalmente se hicieron planes de prueba específicamente sobre la usabilidad del sistema.

Otro de los puntos clave, en este caso el sistema de tiendas, era el método de subida para que las tiendas pudieran publicar sus contenidos en la Web. Si bien el requerimiento en sí mismo fue simple, los aspectos no funcionales del mismo debieron adecuarse a las necesidades de facilidad y rapidez que las tiendas necesitan.

10.2.5 Grupo Foco

Si bien el sistema puede escalar a nivel internacional y el problema se puede adecuar a las necesidades de cada país, para esta primera versión de Lookea.Me se puso énfasis en el mercado uruguayo y de la región.

A su vez, para definir los requerimientos de usuario, tanto usabilidad como requerimientos funcionales, se definió al usuario como mujeres de 15 a 50 años de nivel socioeconómico medio, medio alto y alto, que se conectan a Internet semanalmente.

10.3 Situación del cliente

10.3.1 Contexto general

Según un relevamiento que se llevó a cabo con el grupo comercial de Lookea.Me, se detectó que más de 160 tiendas son clientes potenciales, de las cuales unas 90 se encuentran dentro de los *Shoppings* y 70 por fuera. Cabe resaltar que cerca de un 70% de las 160 tiendas están a su vez en cuadras comerciales, como por ejemplo la Avenida 18 de Julio, ubicada en el centro de Montevideo.

Durante una entrevista con el encargado de una de las tiendas más grandes a nivel local, emergió la dificultad de expansión que muchas de estas grandes tiendas tienen para ampliar sus canales de venta. En muchos casos, dichas tiendas son a su vez empresas familiares, donde un crecimiento aún mayor afectaría la estructura organizacional debido a la necesidad de contratar a más personal, perdiendo de esta forma el carácter de empresa familiar.

En consecuencia, las mismas se ven estancadas en cuanto a crecimiento, sin encontrar aún el método para expandirse. Es por este motivo que Lookea.Me cumple con el mencionado deseo de no contratar personal para expandir su canal de ventas. Esto se debe a que Lookea.Me ofrece una plataforma de publicación de artículos, que no sólo sirve como publicidad más económica y efectiva, sino que además se puede convertir en

el mediano plazo en un nuevo canal de venta, ya que Lookea.Me tiene previsto incluir como nueva funcionalidad la venta de artículos por Internet.

10.3.2 Requisitos para los locales

Los principales requisitos que los locales necesitan son:

- Publicación de artículos en la Web de forma fácil y rápida.
- Informes estadísticos de los usuarios que visitan las prendas de la tienda.
- Informes detallados de las características de los usuarios más frecuentes de la tienda.
- Contacto personalizados con los clientes más compradores de la tienda.
- Publicar artículos destacados de forma de diferenciarse de los demás artículos publicados.
- Obtener comentarios por parte de los usuarios, tanto de los artículos publicados como también de la tienda en sí misma.
- Posibilidad de vender artículos por la Web.

10.4 Situación del usuario

10.4.1 Contexto general

Si bien los comercios tanto dentro como fuera de los *Shoppings* son rentables, la disconformidad es un tema común entre los clientes, ya sea por la mala atención dentro de los locales o por la aglomeración de gente en los mismos. Si bien hay personas, en su mayoría mujeres, que disfrutan de ir de compras, el problema de la mala atención y aglomeración en los locales son elementos desestimuladores del proceso de compra.

Por este motivo muchas personas buscan otros canales para buscar y/o comprar moda, ya sea dentro o fuera del país. Analizando el problema, muchas de ellas lo hacen cómo alternativa para no ir a los locales, no solo por los problemas antes mencionados, sino también por un problema de distancias.

En ciertas entrevistas realizadas con algunos usuarios, el problema de la distancia fue mencionado en repetidas ocasiones, buscando la alternativa de comprar online sin necesidad de ir al local físico con la consecuente pérdida de tiempo, donde en reiteradas ocasiones el cliente llega y se va sin comprar lo que busca.

En base a los problemas anteriormente mencionados, Lookea.Me provee un canal por el cual el usuario puede mejorar el proceso de toma de decisiones antes de la compra, ya sea desde su hogar, trabajo o celular. De esta forma, el usuario al llegar al local no solo

sabe lo que está buscando, sino que tiene la certeza que realizará la compra. A su vez, el tiempo de compra dentro del local es mucho menor, por lo que la aglomeración de gente dentro de los locales será menor.

Si bien la mala atención dentro de los locales puede ser originada por diferentes problemas, la aglomeración de personas dentro de los locales hace que la atención sea poco personalizada y de baja calidad. En consecuencia, si los locales tienen menos personas dentro, se podrá mejorar la calidad de la atención personalizada y como resultado, minimizar la disconformidad asociada a este motivo.

10.4.2 Requisitos para los usuarios

En base al análisis anteriormente propuesto, los principales requisitos de los usuarios son:

- Acceder a la oferta de moda del mercado local de forma on-line sin necesidad de trasladarse a los locales.
- Unificar toda la oferta en una misma página Web.
- Mejorar la experiencia de compra.
- Facilitar toda la información necesaria para que la decisión de compra sea la correcta.

10.5 Especificación de requerimientos

A continuación se presentarán un conjunto de requerimientos para cada sistema, los cuales fueron extraídos de las etapas anteriormente mencionadas. No se presentan la totalidad de los requerimientos, sino un conjunto menor lo suficientemente representativo de la complejidad de cada sistema (véase Anexo IV y XVI).

10.5.1 Requerimientos funcionales

El requerimiento más importante de todo el sistema de las tiendas es la carga de contenidos. Como se puede observar en la tabla 3, los requerimientos que incluyen alguna modificación en cuanto a los artículos publicados tienen prioridad alta. Por otro lado, también se pueden observar otros requerimientos, tales como los reportes, que si bien son requerimientos del sistema, son de prioridad media, ya que no son esenciales para que el sistema de usuarios pueda estar en funcionamiento:

Aplicación	Nombre	Prioridad
Tiendas	Publicar artículo	Alta
	Publicación masiva de artículos	Alta
	Mantenimiento de artículos	Alta
	Reportes por intranet	Media
	Publicar descuento colectivo	Alta
	Vender un artículo reservado	Alta
	Lista de reservas	Alta
	Generar información personalizada a seguidores	Media
	Chat on-line	Alta
	Calificar la experiencia compra-venta	Baja

Tabla 3 - Requerimientos de la aplicación de tiendas

Si bien el requerimiento más importante del sistema de usuarios es la búsqueda de artículos, hay un conjunto de variables que influyen en que éste no sea el único requerimiento necesario para publicar la página. Esto se debe a que las tiendas a su vez necesitan garantizarse que el sistema tenga potencial de uso, imponiendo así la necesidad de guardar los usuarios en una base de datos. De forma que los usuarios se vean atraídos a registrarse en la página, la página debe contener requerimientos tentadores, como por ejemplo, la vinculación con las redes sociales. Los requerimientos del sistema de usuarios se detallan en la tabla 4.

Aplicación	Nombre	Prioridad
Usuarios	Registrarse	Alta
	Login (incluyendo redes sociales)	Alta
	Búsqueda de artículo	Alta
	Opción "Me Gusta"	Alta
	Agregar artículos a mi colección	Alta
	Wish List	Alta
	Publicar en redes sociales	Alta
	Seguir tiendas y/o usuarios	Alta
	Reservar	Alta
	Comprar	Alta
	Visualizar de artículos destacados	Alta
	Visualizar de actividades recientes	Media
	Venta cruzada	Alta
	Armar de recorrido	Media
	Recibir información personalizada	Media

Aplicación	Nombre	Prioridad
Usuarios	Regalar vale de compra	Media
	Comprar por descuento colectivo	Alta
	Calificar atención recibida por el local	Alta

Tabla 4 - Requerimientos de la aplicación de usuarios

Finalmente, los requerimientos del sistema de Administradores (Tabla 5) no tuvieron mayores complicaciones debido a que los requerimientos de dicho sistema se veían influenciados por los requerimientos de los otros dos sistemas. El único requerimiento no influenciado por los demás es la opción de subir contenidos, donde los administradores podrán subir contenidos de tiendas o páginas del estilo “eBay” de forma de aumentar la oferta de artículos en la página.

Aplicación	Nombre	Prioridad
Administradores	Subir Contenidos	Alta
	Monitorear de actividad de las tiendas	Media
	Consultar calificación de usuarios	Media
	Crear news letter	Alta
	Personalizar news letter	Media
	Enviar mensaje de texto	Media
	Dar de alta una tienda	Alta
	Dar de alta personal shopper	Media
	Aprobar ingreso de contenidos	Alta

Tabla 5 - Requerimientos de la aplicación de administradores

10.5.2 Requerimientos no funcionales

10.5.3 Usabilidad

El diseño gráfico de la página debe ser atractivo, sencillo e intuitivo. Los usuarios deben disfrutar la experiencia desde lo visual y además deben poder aprender solos cómo realizar las tareas más importantes del sitio, cómo visualizar y reservar prendas. A su vez, la aplicación de tiendas, si bien no es requisito que el sistema sea atractivo, es necesario que sea fácil y rápido gestionar la subida de los diferentes artículos que las tiendas deseen incorporar, modificar o eliminar.

10.5.4 Mantenibilidad

Los sistemas contienen un alcance mayor al alcance propuesto para este proyecto. No obstante, la arquitectura y código de los requerimientos a desarrollar deben ser tales que la inclusión de los restantes requerimientos no impacte de forma importante lo desarrollado al momento. Cabe resaltar que dentro de este requerimiento se incluye la posibilidad de multi-lenguaje, siendo en un principio para español pudiendo agregar posteriormente otros idiomas, sin mayor impacto.

10.6 Criterios de priorización

Los requerimientos fueron priorizados en tres categorías: Alto, Medio y Bajo. Para adjudicar una prioridad se tomó en cuenta en primera instancia si dicho requerimiento era esencial para que el sistema tuviera éxito. Los de prioridad Alta se los consideró necesarios para el sistema, mientras que los de prioridad Media y Baja eran considerados deseables.

Ciertos requerimientos, como por ejemplo la compra de artículos, si bien eran de prioridad alta, era necesario que estuvieran desarrollados posteriormente a otros, por lo que se debió considerar también el tiempo que insumía desarrollar el requerimiento para luego poder calcular el esfuerzo estimado.

Con estos criterios definidos, se resolvió dejar para etapas posteriores los requerimientos de prioridad Media y Baja. Los de prioridad Alta fueron negociados con el grupo comercial de Lookea.Me para evaluar en qué orden debían ser desarrollados. Una vez definido el orden y estimado el tiempo que insumía cada uno, se obtuvo que el tiempo estimado total excedía el tiempo total estimado del proyecto, por lo que se negoció en una última instancia cuáles requerimientos, aun siendo de prioridad alta, quedarían por fuera del alcance del proyecto.

10.7 Clientes y actores

Si bien existieron varios clientes para el proyecto, no así para la definición de los requerimientos y su priorización. Quien se encargó de definir dichos aspectos de los requerimientos fue el grupo comercial de Lookea.Me, centralizando la comunicación con las tiendas y los usuarios por medio de sus integrantes José Miguel Majó y Juan Manuel Dellepiane. De esta forma, el alcance de los sistemas junto con sus respectivas negociaciones, fueron realizados entre el grupo de proyecto y el grupo comercial.

A continuación se detallan los diferentes actores para cada sistema.

- **Sistema de Administradores:** Grupo comercial y técnico de Lookea.Me
- **Sistema de Usuarios:** Los usuarios del sitio Web. El foco del producto son mujeres entre 15 y 50 años de nivel socioeconómico medio, medio alto y alto, que se conectan a Internet semanalmente.
- **Sistema de Tiendas:** Locales del sector de la moda de Uruguay y la región.

10.8 Validaciones y verificaciones

Las verificaciones, tanto de los requerimientos como de los sistemas desarrollados, fueron llevadas a cabo por parte del grupo comercial de Lookea.Me. Dichas verificaciones fueron realizadas de la siguiente manera:

- **Requerimientos:** Julio 2012, al terminar el proceso de ingeniería de requerimientos.
- **Diseño de la aplicación:** Al terminar el diseño por parte de los diseñadores gráficos en Octubre 2012.
- **Desarrollo de aplicaciones:** Al terminar cada iteración del ciclo de vida, luego de la etapa de pruebas.
- **Aplicaciones publicadas:** Se realizó la publicación en un servidor público en Enero 2013 y los clientes hicieron su última verificación sobre todos los sistemas.

11 Arquitectura y diseño

El presente capítulo tiene la finalidad de presentar todas las decisiones tomadas por el equipo con respecto a la arquitectura del sistema y de cada sistema independiente. Se presentará el diseño arquitectónico realizado, sus principales características de calidad, procesos arquitectónicos y decisiones tomadas para su implementación a alto nivel.

El diseño arquitectónico macro responde a la arquitectura del negocio que Looke.Me busca crear, como se presenta en la figura 9.

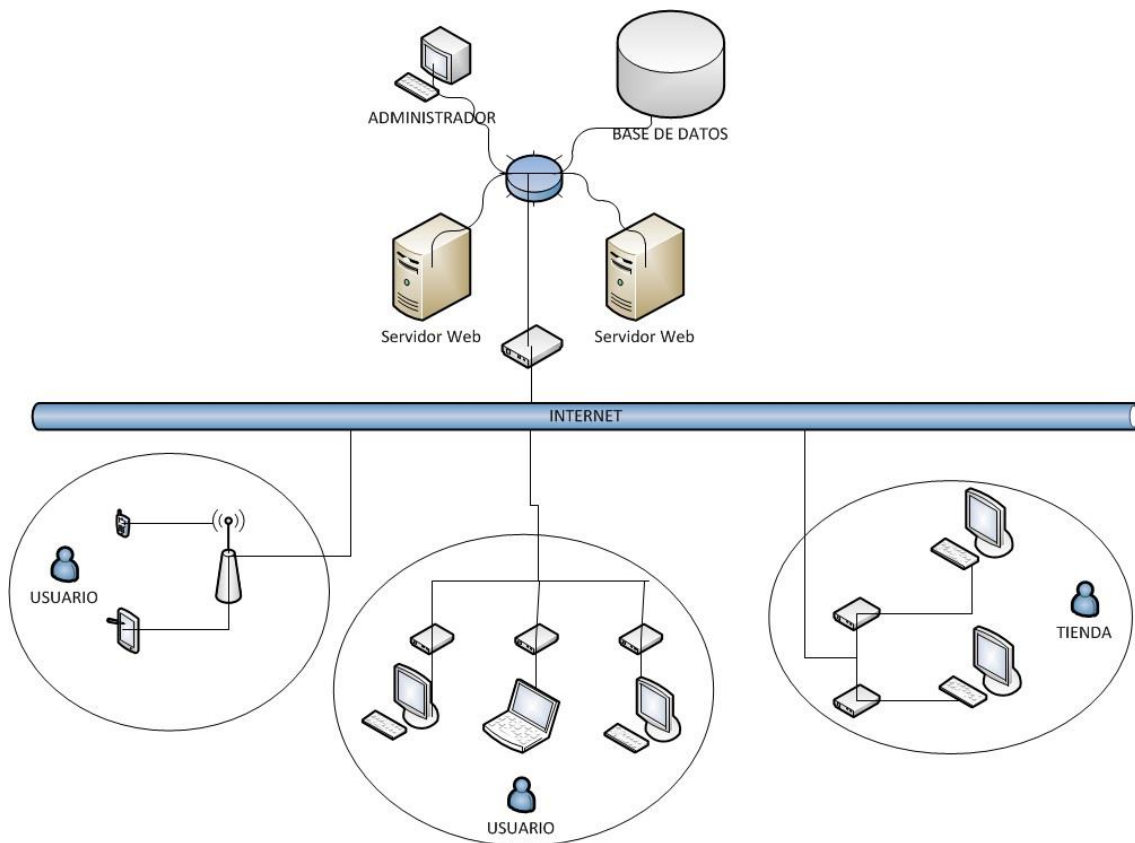


Figura 9 - Diseño arquitectónico a alto nivel

11.1 Componentes del sistema

11.1.1 Aplicación de usuarios (<http://www.lookea.me>)

Es el sitio Web público al cual acceden los usuarios que desean buscar, conocer, comprar y compartir la moda (Figura 10).

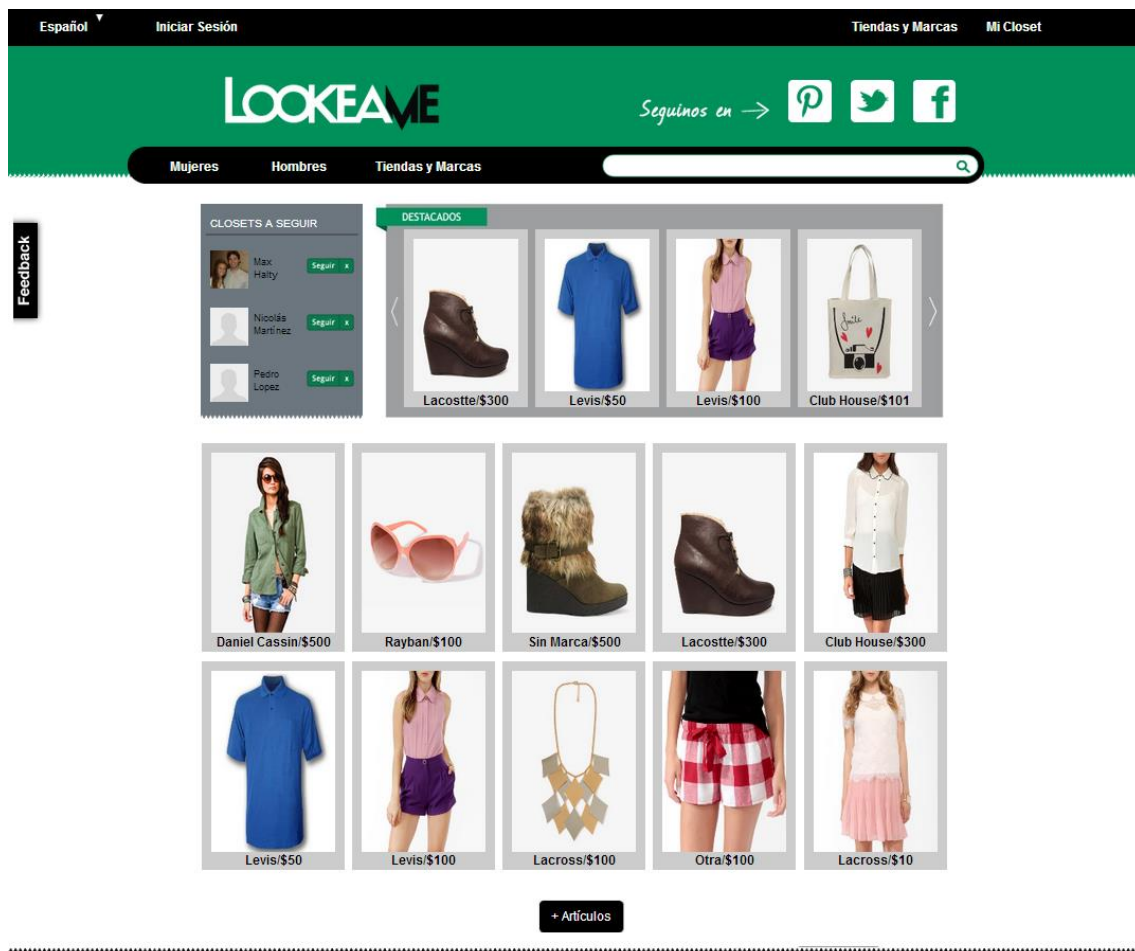


Figura 10 - Sistema de usuarios

El siguiente diagrama (Figura 11) muestra cuáles son las funcionalidades que son responsabilidad de esta aplicación Web:

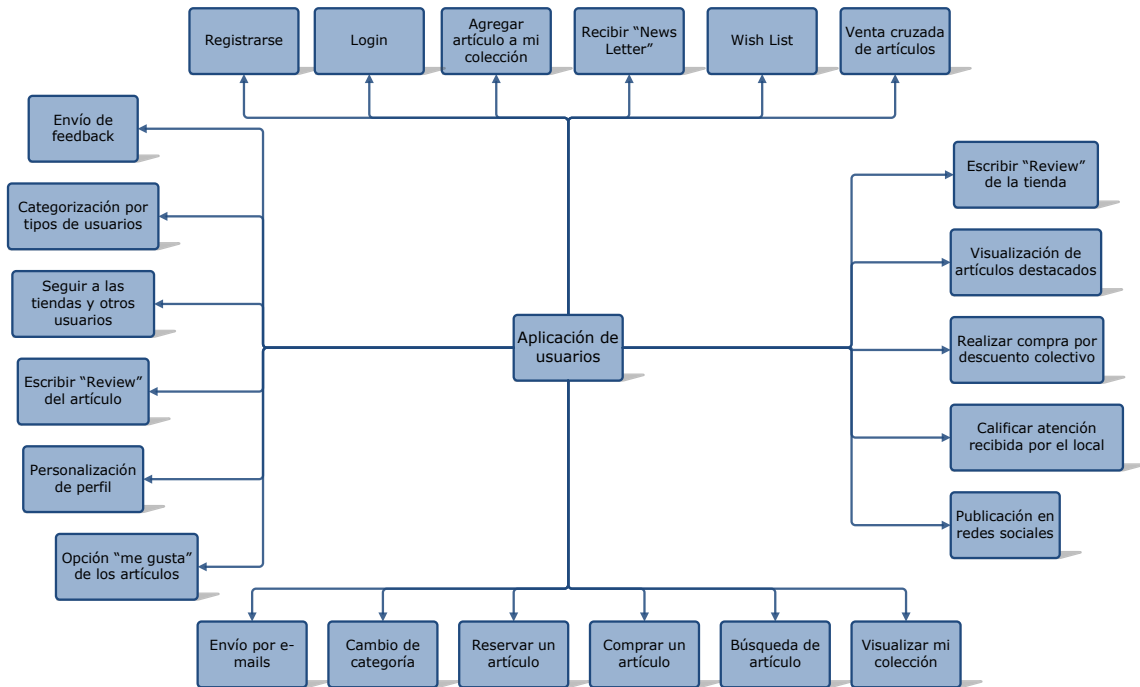


Figura 11 - Funcionalidades de la aplicación de usuarios

11.1.2 Aplicación de tiendas (<http://stores.lookea.me>)

Es la aplicación Web a la cual acceden las tiendas que trabajan con Looke.Me para poder administrar su perfil, los artículos que publican, los datos de la tienda y la interacción con los usuarios (Figura 12).

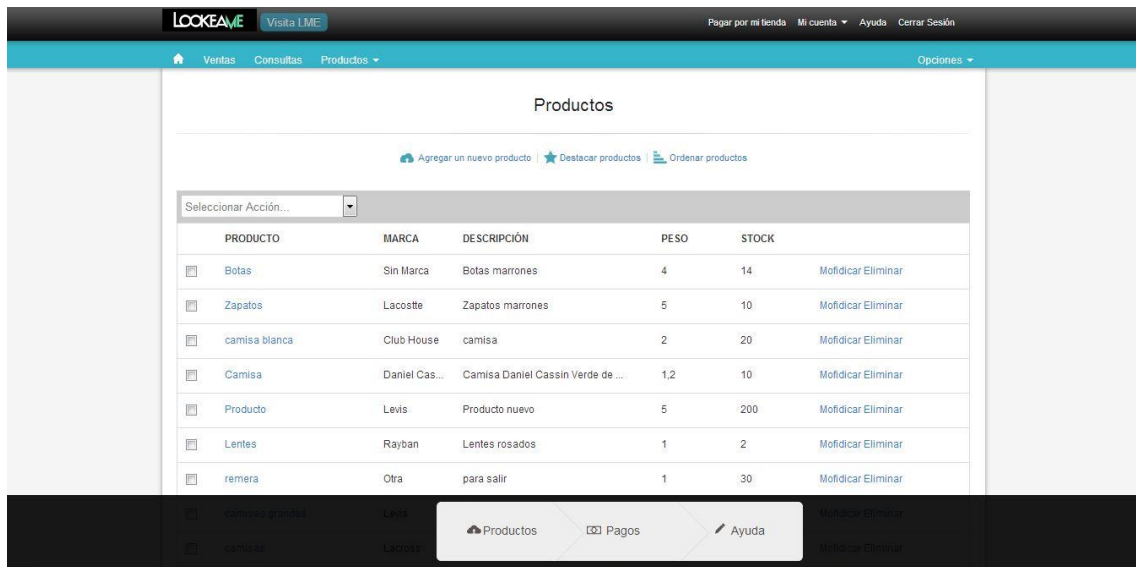


Figura 12 - Sistema de tiendas

El siguiente diagrama (Figura 13) muestra cuales son las funcionalidades que son responsabilidad de esta aplicación Web:

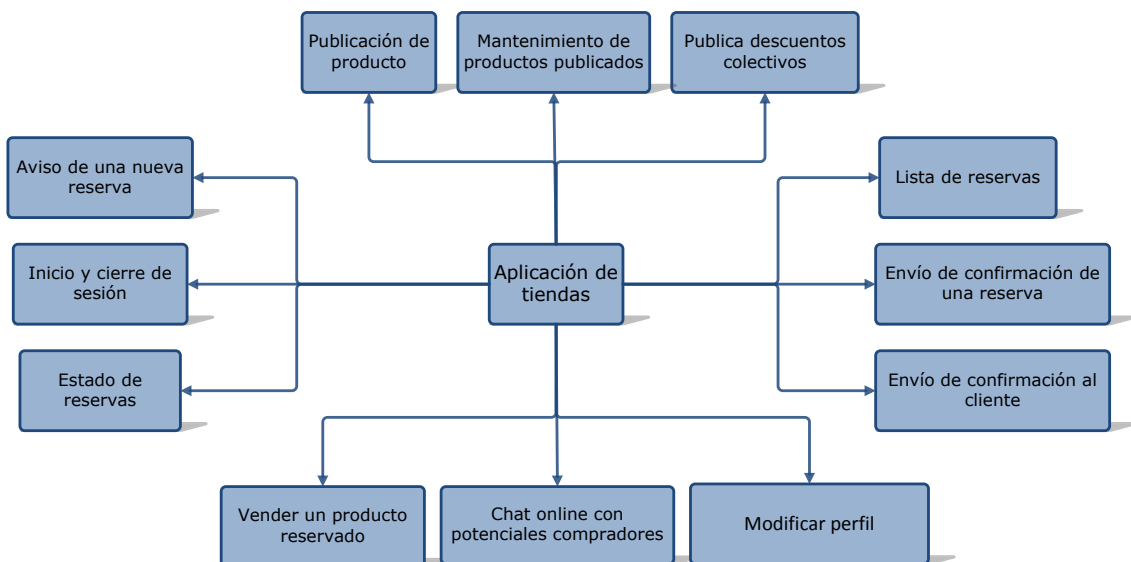


Figura 13 - Funcionalidades de la aplicación de tiendas

11.1.3 Aplicación administradores (<http://admin.lookea.me>)

Es el sitio Web que utilizan los integrantes de Looke.Me para administrar todas las aplicaciones, por motivos de seguridad se necesita centralizar varias funcionalidades en esta aplicación (Figura 14).

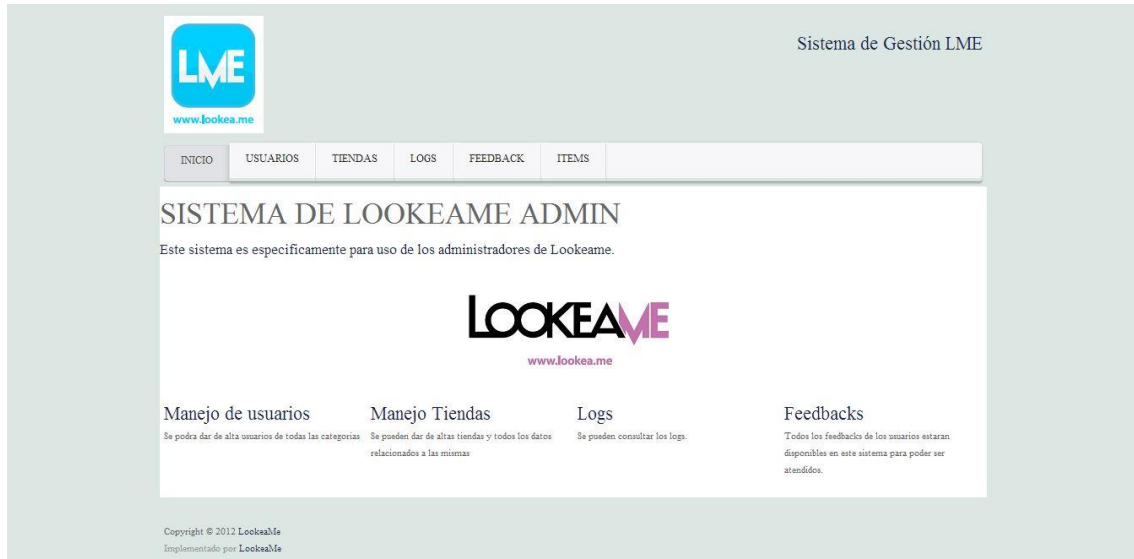


Figura 14 - Sistema de administradores

El siguiente diagrama (Figura 15) muestra cuáles son las funcionalidades que son responsabilidad de esta aplicación Web:

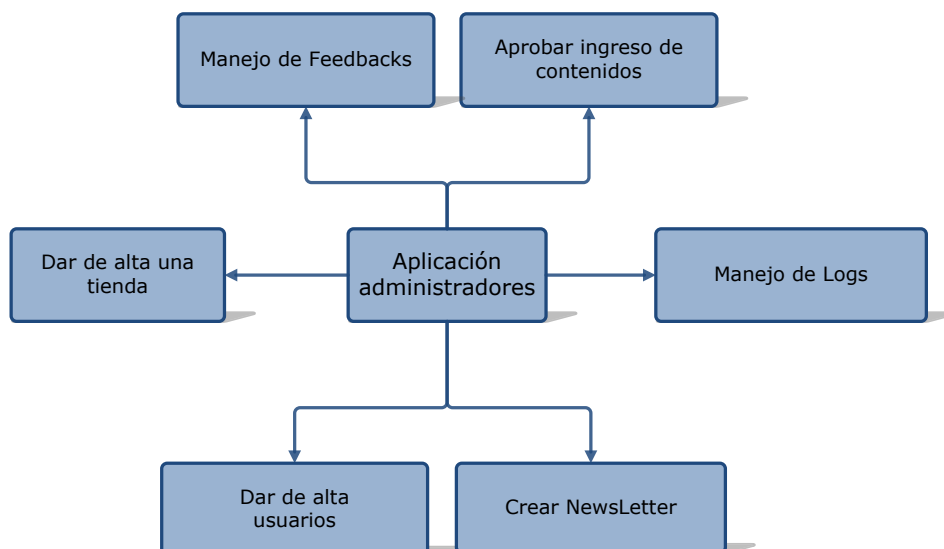


Figura 15 - Funcionalidades de la aplicación de administradores

11.2 Características de calidad consideradas

Los atributos de calidad a ser detallados se basan en los resultados de la Ingeniería de Requerimientos realizada por el equipo y detallada en el Capítulo 8. Los mismos surgen de los requerimientos no funcionales que las aplicaciones a desarrollar deben tener y son los siguientes:

- Usabilidad
- Eficiencia
- Disponibilidad
- Seguridad
- Mantenibilidad
- Escalabilidad

A continuación se presentan los mecanismos de arquitectura utilizados para alcanzar los atributos y los niveles especificados para cada uno.

11.2.1 Usabilidad

La usabilidad es el atributo más importante del sistema, exceptuando la aplicación de administradores, que es utilizada internamente y tiene menor exigencia en este aspecto. Las aplicaciones deben ser extremadamente intuitivas y los usuarios deben poder aprender a utilizar el sistema sin ayuda externa.

El mecanismo que se utilizó para evaluar la usabilidad es la validación con los usuarios finales. Se identificaron perfiles de usuarios y se seleccionaron un grupo representativo de los mismos para realizar pruebas.

Se creó un plan de pruebas de usabilidad (véase Capítulo 11 y Anexo VIII), y se evaluaron tres características principales el “*look and feel*”, “cantidad de errores” y “velocidad de uso” (véase Anexo V).

11.2.2 Eficiencia

El tiempo que demora un usuario desde que entra hasta que encuentra el artículo que está buscando no puede ser mayor a 30 segundos. El sistema se utiliza para que los usuarios encuentren aquellos artículos que desean y la búsqueda debe estar optimizada.

Para lograr esto el sistema tiene dos métodos de búsqueda, la búsqueda libre que permite al usuario ingresar palabras claves con las que buscar los artículos y la posibilidad de utilizar filtros para encontrar productos más fácilmente.

11.2.3 Disponibilidad

Teniendo en cuenta que el sistema busca facilitar una tarea que suele ser parte del día a día de las personas, es importante que en todo momento que un usuario pueda ingresar al sistema el mismo se encuentra disponible.

No es un sistema que ofrezca un servicio que deba estar 99% de las veces disponible pero si se han analizado franjas de horas en donde el sistema debe ofrecer una disponibilidad mayor a 80%, las mismas son de 19:00 a 00:00 en un día entre semana que es el tiempo en que las personas están más libres y entre las 12:00 y la 1:00 am de los fines de semana.

11.2.4 Seguridad

Los dos aspectos de seguridad priorizados fueron la autenticación y autorización. En el plan de pruebas (véase Anexo VII) se definen las pruebas que se realizaran para asegurar que estos requerimientos se cumplan y los niveles de aceptación de los mismos (véase Anexo V).

Se considera que la autenticación debe ser asegurada en todo momento y no debería fallar en circunstancias normales dado que el sistema maneja datos sensibles de usuarios y tiendas. Hay información pública que se podrá consultar abiertamente pero aquella información privada solo podrá ser vista por usuarios identificados. Además solo se deben revelar aquellos datos necesarios para que cada usuario cumpla las funciones a las cuales tiene permisos, por lo que la autorización también es un aspecto importante, aunque en un nivel de prioridad menor.

11.2.5 Mantenibilidad

Considerando el tamaño del grupo de tesis y el hecho de que ya ha programado junto en muchas oportunidades a lo largo de la carrera, este atributo de calidad resultó ser de los primeros en listarse al momento de tomar decisiones arquitectónicas.

No solo se optó por una solución en capaz y con completa independencia de la solución de base de datos sino que además se realizaron pruebas cruzadas por parte de los integrantes del equipo, de modo de asegurar el cumplimiento de los estándares de programación propios del ambiente de desarrollo (véase Anexo VII).

11.2.6 Escalabilidad

Lookea.Me busca ser un sitio referente de moda de la región. Se tiene claro que hay mucha competencia y que es difícil posicionarse, pero sin duda con las muchas barreras que hay para poder lograrlo, no puede ser la arquitectura del sistema una limitante más. Tomando esta directiva, se diseñó un sistema apto para soportar entre 100 y 500 usuarios en forma concurrente, siendo la cantidad de servidores el único limitante. Si fuese necesario se podría aumentar la cantidad de recursos.

11.3 Decisiones Tecnológicas

Luego de la etapa de Ingeniería de Requerimientos el equipo realizó una investigación sobre las tecnologías que utilizaría para desarrollar Lookea.Me. A continuación (Tabla 6) se muestran los datos relacionados a esta decisión y la resolución de la misma:

Lenguaje	Experiencia del equipo	Estudio tecnologías	Certificaciones
Java	Baja	Todos los integrantes cursaron 3 materias en este lenguaje	-
.NET	Alta	Todos los integrantes cursaron 2 materias en este lenguaje	Un integrante tiene certificado del Examen Microsoft “.Net Framework 4.0 Web Applications”
C ++	Media	Todos los integrantes cursaron 2 materias en este lenguaje	-
GeneXus	Media		Un integrante tiene el título “Analista Senior GeneXus”

Tabla 6 - Experiencia de los integrantes del equipo

El equipo decidió utilizar el *framework* .Net 4.0, en particular ASP.Net, ASP.Net MVC 3.0, C# y Microsoft SQL Server como motor de base de datos. También se decidió utilizar la tecnología ADO.Net Entity Framewok 4.0 para elevar el nivel de abstracción de la programación contra fuentes de datos.

La primera decisión arquitectónica que se realizó tuvo que ver directamente con esta última tecnología que se decidió utilizar, ADO.Net Entity Framework 4.0. Un gran punto débil que suele aparecer en los proyectos que utilizan esta tecnología es que las clases del dominio del sistema suelen ser las clases que genera este *framework* a partir del modelo de datos. Esto puede ser una gran ventaja para sistemas pequeños que trabajan con operaciones sencillas de mantenimientos de datos, pero puede ser un gran problema para sistemas grandes dado que si se decide cambiar de motor de base de datos, se debe comenzar el sistema de cero. Por otra parte, resulta muy difícil crear objetos del dominio que no tengan una contraparte en el modelo de datos. Se decidió invertir la dependencia de la solución de base de datos y crear un proyecto “dominio” que contuviera objetos del dominio que fueran las entidades principales del sistema. Lookea.Me debería poder cambiar el modelo de datos, sin necesidad de cambiar ningún aspecto en el dominio del mismo.

La segunda decisión, que acompañó la primera, fue utilizar el patrón arquitectónico en capas, buscando de esa manera potenciar por un lado la mantenibilidad, dado que es más claro entender la responsabilidad de cada capa del sistema si están bien armadas, y por otro la escalabilidad, porque fue posible separar los componentes físicos del sistema más sencillamente. Esto permitió una experiencia de programación para los integrantes del equipo que facilitó la utilización de este patrón.

Por último se decidió utilizar ASP.Net para dos de los tres sistemas Web y ASP.Net MVC para el sistema de usuarios. Esta decisión se basó en que el equipo tenía más experiencia en el primer *framework*, por lo que permitió que el desarrollo fuera más fácil de estimar y de realizar en fecha. Asimismo, el sistema de usuarios debía permitir en un futuro implementar interfaces gráficas para dispositivos móviles o tabletas con un esfuerzo mínimo, lo cual este segundo *framework* permitió, siendo que el modelo MVC está pensado para permitir cambiar las vistas sin necesidad de tocar el código del lado del servidor. Por otra parte, se necesitaba un *framework* que fuera más orientado a lenguajes que corriera del lado del cliente como JavaScript y JQuery, porque estas tecnologías permiten más flexibilidad en el diseño y en la animación de los sitios.

11.4 Herramientas

En base a la experiencia que cada uno de los integrantes del equipo tenía en ambientes de desarrollo de tecnologías Microsoft se decidió utilizar las siguientes herramientas para el desarrollo de Looke.Me:

Entorno de Desarrollo (IDE): Se utilizó Microsoft Visual Studio 2010 como IDE de desarrollo.

Lenguaje de Programación: Se utilizó C# debido al *expertise* que tenían los integrantes del equipo de desarrollo.

Base de Datos: La base de datos utilizada fue MS SQL Server 2008 R2, y para la administración se utilizó el MS SQL Server Management Studio.

Client-Side: Se utilizó HTML y CSS para soluciones de diseño gráfico. Se utilizaron los lenguajes JavaScript y JQuery para animaciones y variadas soluciones de interacción con el lado del cliente en el sitio Web de usuarios.

11.5 Fundamentos de arquitectura

Se detallan a continuación las decisiones arquitectónicas tomadas para alcanzar los atributos de calidad definidos. En algunas de las decisiones se tuvieron en cuenta más de un atributo de calidad, y siempre que se tuvo que optar, se definió que la Usabilidad y la Seguridad eran los atributos prioritarios para el sistema.

La solución fue realizada desde un comienzo aplicando el estilo arquitectónico en capas (Figura 16), obteniendo un menor acoplamiento de la solución. También se diseñó la aplicación en módulos para poder tener una mejor separación de responsabilidades y mejorar así la mantenibilidad.

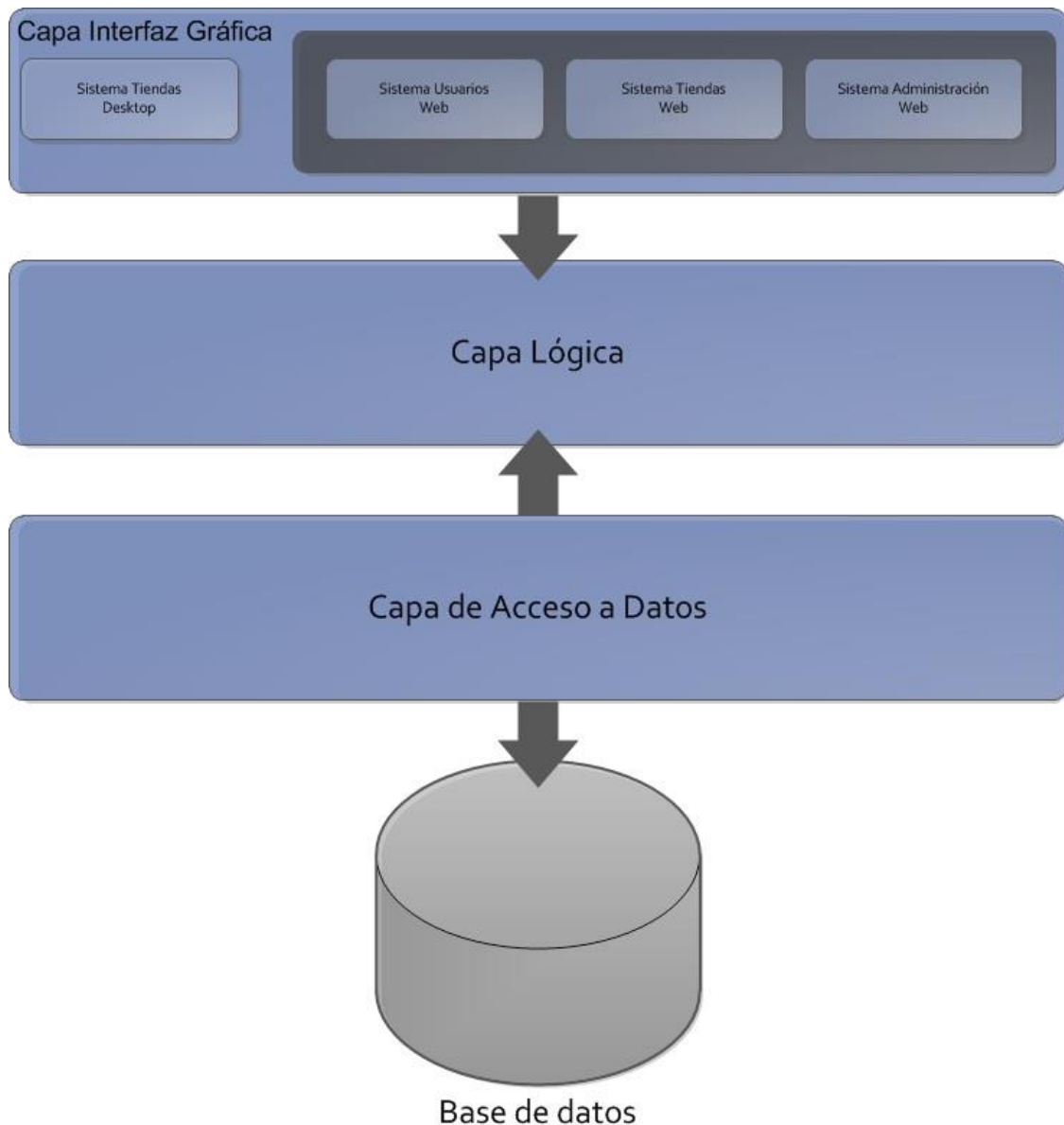


Figura 16 - Arquitectura en capas

En la aplicación de usuarios se buscó potenciar la usabilidad utilizando el patrón arquitectónico MVC (Modelo Vista Controlador) (Figura 17). Lo que se buscó implementando este estilo, fue separar los datos de la aplicación, la lógica de la aplicación y la interfaz gráfica. De esta manera se logró una mejor separación de responsabilidades y se favoreció la mantenibilidad al ser más fácil de entender el sistema, controlándose la seguridad a través del acceso a las páginas por medio de controles en los llamados a las URLs. Antes de que un usuario vea cualquier página accede a un controlador que le responde con la página a la que tiene acceso. Y por último y más importante, se favorece la usabilidad debido a que es un estilo arquitectónico que potencia el uso de lenguajes orientados al lado del cliente como JavaScript y JQuery, obteniéndose mejores resultados en la interacción con el usuario, permitiendo crear varias vistas de una misma página pudiendo controlar cual vista mostrarle al usuario dependiendo de lo que se quiera desplegar.

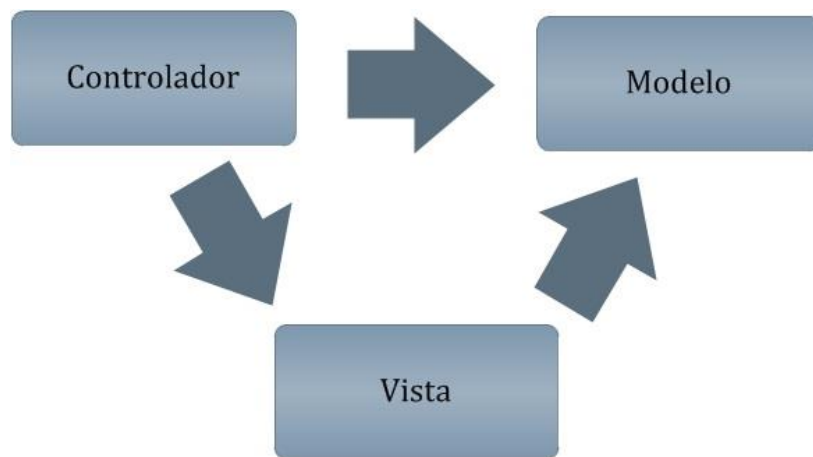


Figura 17 - Arquitectura MVC

La implementación de este patrón en el .Net Framework 4.0 fue ASP.NET MVC 3.0 con el motor de vistas conocido como Razor [52]. Razor no es un lenguaje de programación en sí mismo, pero usa la sintaxis de C# o Visual Basic para tener código dentro de una página sin los delimitadores típicos de ASP.NET: `<%= %>`. La extensión de archivos Razor es 'cshtml' para el lenguaje C#, y 'vbhtml' para Visual Basic.

Este patrón se utilizó únicamente en la aplicación de usuarios, de modo de mitigar el riesgo de utilizar una nueva tecnología dado que el equipo tenía experiencia en ASP.NET pero no en MVC. Se consideró que el riesgo iba a ser mejor gestionado si se asumía en un único sistema, además de que el mayor provecho que se podía sacar de dicho estilo se podía ver únicamente en el sistema de usuarios.

Se utilizó la arquitectura propuesta en ASP.NET para resolver la seguridad del sistema. Dentro de las opciones que ofrecía esta arquitectura se decidió utilizar la seguridad basada en formularios (*cookies*) ya que proporcionan la posibilidad de manejar roles de usuarios y permisos especiales para usuarios particulares. Pero en busca de poder controlar mejor los roles y grupos de usuarios se implementó todo lo relacionado a roles y grupos de usuarios dejando que el *framework* se encargara del resto de responsabilidades. De esta forma, se aseguró una implementación de seguridad probada, con años de estar en el mercado y muy estable, implementándose únicamente lo relacionado a los roles y grupos, de forma de manejar esto desde la base de datos de Looke.Me.

Tal como se puede apreciar en la figura 18, el sistema cumple con una arquitectura distribuida Cliente – Servidor.

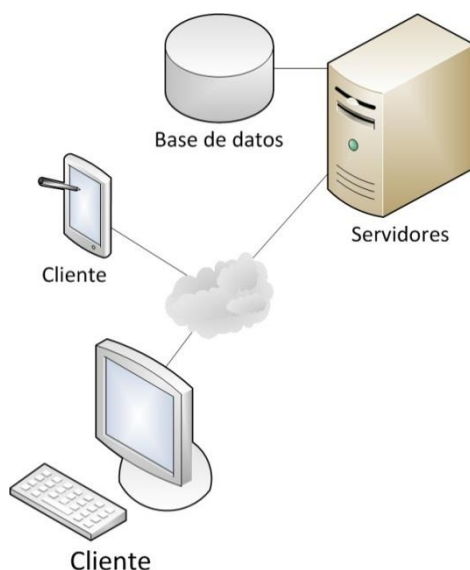


Figura 18 - Arquitectura cliente-servidor

El Cliente, que es básicamente responsable de la capa de presentación, requería servicios del Servidor para actuar como un motor completando sus solicitudes. La comunicación entre el Cliente y el Servidor se realizó utilizando el protocolo *TCP/IP*.

Es importante destacar que existió la posibilidad de utilizar la herramienta GeneXus [41] (véase Anexo III) para el desarrollo del sistema de administradores. Teniendo en cuenta el objetivo y las funcionalidades asociadas al sistema mencionado, GeneXus cumplía técnicamente con todos los requisitos para poder ser utilizado como herramienta de desarrollo.

Como ventaja en comparación con el uso de ASP.NET, el desarrollo con GeneXus de dicho sistema sería bastante más rápido. Por su lado, habiendo utilizado Microsoft SQL Server como motor de base de datos, la integración de las demás aplicaciones con el uso de GeneXus no hubiera sido un problema. Dicha integración debía considerar aspectos de nomenclatura que utiliza GeneXus para poder funcionar, pero en este caso, afectaría solamente el modelo de la base de datos.

Ahora bien, si el desarrollo para los demás sistemas era en Microsoft .NET y para el sistema de administradores se utilizaba GeneXus, los clientes deberían pagar por ambas licencias, la de Microsoft .NET y la de GeneXus. Por este motivo, los clientes decidieron desarrollar todas las aplicaciones con Microsoft .NET, descartándose la utilización de GeneXus, aun cuando ciertos desarrollos pudieron haber sido más eficientes en GeneXus. Cabe resaltar que tomando esta decisión, el alcance del sistema

fue menor que si hubieran optado pagar ambas licencias, ya que el desarrollo back-end pudo haber sido más rápido, logrando mayor énfasis en el desarrollo del sitio Web de usuarios.

11.6 Patrones de diseños utilizados

11.6.1 Unit of Work

Al optar por la tecnología de Entity Framework [42] como solución de acceso a datos se debía encontrar una solución para no perder el Diseño orientado al Dominio o *Domain-Driven Design* (DDD) [43], de modo de reducir la separación entre los expertos del negocio y los desarrolladores.

Dicho *framework* permitía crear las clases necesarias para trabajar con los objetos del dominio, generando que el mismo estuviera completamente acoplado a la solución de base de datos. El objetivo era hacer desaparecer las dependencias que éste generaba.

Se observó que era posible realizar una solución que se basara en un dominio con clases POCO (Plain Old C# Object, simplemente objetos con sus atributos y métodos, sin nada extra que no fuera del dominio específicamente), con una solución de base de datos basada en Entity Framework 4.0, utilizando el patrón de diseño Unit of Work [45]. Este último, “tiene como objetivo tratar como una Unidad todos aquellos objetos nuevos, modificados o eliminados con respecto de una fuente de datos” [44].

Se buscó tener un dominio que no dependiera de ningún otro paquete en la solución, de modo que su única razón de modificación fuera un cambio en el negocio. Para ello, se realizó una inversión de dependencia entre el paquete de acceso a datos y el dominio, haciendo uso del patrón Unit of Work para poder lograr satisfactoriamente esta solución.

Se realizaron pruebas de concepto que tuvieron como resultado una solución que plasmó lo buscado con aprobación de todo el equipo. En la figura 19 se presenta un diseño de alto nivel de la misma:

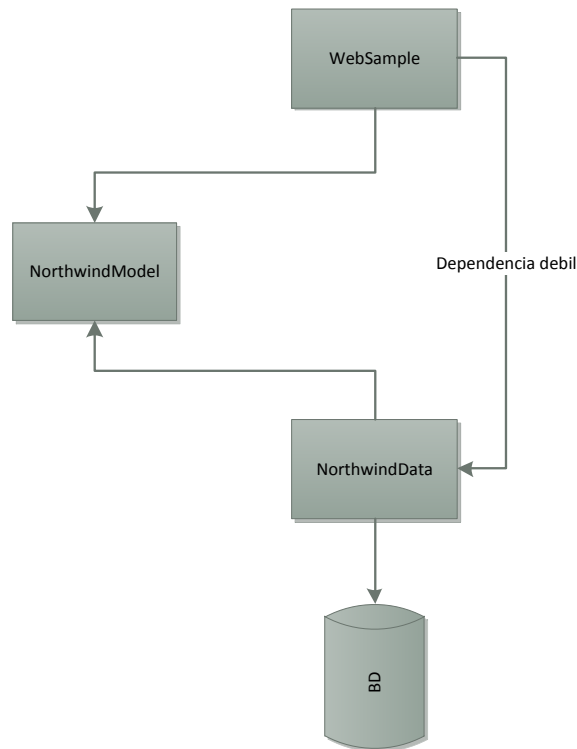


Figura 19 - Aplicación con el patrón Unit of Work

Fachada

Este patrón fue utilizado junto al patrón Unit of Work, dado que las interfaces que permitían invertir la dependencia con la base de datos eran propiamente implementaciones de este patrón de diseño. Permitían al dominio interactuar con la fachada sin importar la implementación que la misma tenga, reduciendo en forma importante el acoplamiento.

11.7 Vistas de arquitectura

A continuación se describirá el diseño arquitectónico del sistema basado en el modelo “The 4 + 1 View Model Software Architecture” de Krutchen [2], modelo basado en el uso de vistas múltiples y concurrentes. A los efectos de este proyecto, se describirán la vista de componentes, la vista de despliegue, la vista de procesos y por último, la vista de casos de uso. No se presente la vista lógica dado que las decisiones de diseño más importantes pueden ser analizadas en la vista de componentes.

11.7.1 Vista de componentes

A continuación se presentará como se estructura físicamente el sistema en componentes de software, incluyendo sus principales dependencias, interfaces y responsabilidades.

En la figura 20 se detallan los principales componentes del sistema. Se omitieron algunos componentes secundarios para favorecer la comprensión y se realizaron agrupaciones por capas en algunos casos para no presentar dependencias que no aportan información relevante.

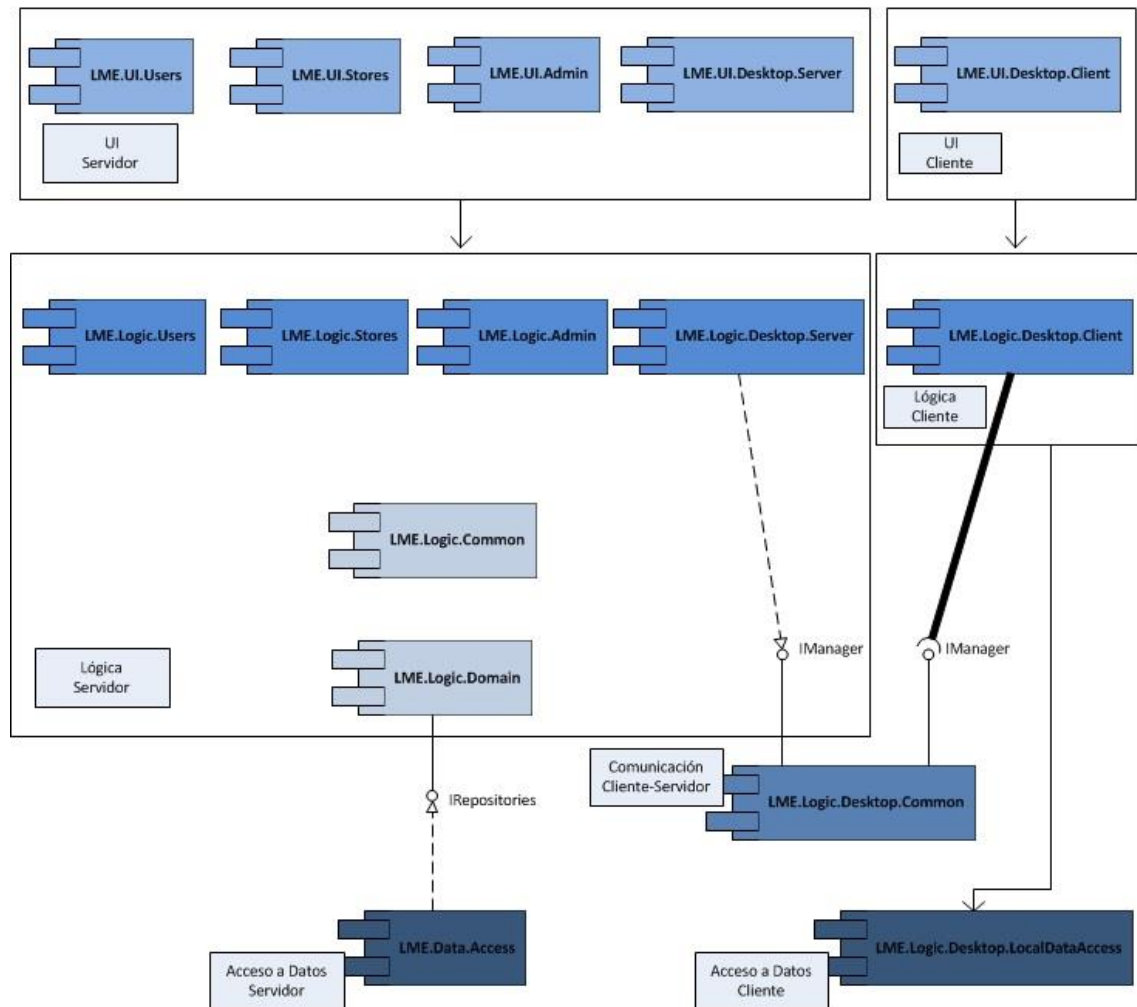


Figura 20 - Vista de despliegue

En el diagrama mostrado en la figura 20 se puede visualizar la arquitectura en capas que se utilizó tanto en las aplicaciones Web como en la aplicación cliente de escritorio, simplificando el esquema seleccionando los componentes principales y las asociaciones más relevantes.

Para comunicarse con las redes sociales se utilizó DLLs públicas e invocaciones a métodos REST, siendo estos servicios Web que no contienen estado y son accedidos únicamente por URLs. Al día de hoy las redes sociales más usadas tienen *frameworks* que hacen sencilla la tarea de comunicación desde aplicaciones externas.

Dentro de las asociaciones que se simplificaron es importante detallar que entre la capa de interfaz gráfica y la capa lógica las asociaciones son completamente independientes

entre cada aplicación Web y aplicación de escritorio. Cada aplicación Web tiene su componente de interfaz gráfica que interactúa únicamente con el componente de lógica de negocio correspondiente a dicha aplicación. De esta forma se redujo el acoplamiento y se favoreció la mantenibilidad dado que cada aplicación es completamente independiente de las demás. Lo único que comparten es el componente de Dominio, que tiene todos los objetos propios del dominio de todas las aplicaciones, y el componente denominado “Common”, que fue creado en un refactorio del código por sugerencia de uno de los integrantes del equipo, dado que algunas funcionalidades eran idénticas en todas las aplicaciones Web y aparecía código duplicado.

El patrón “Unit of Work” propone que cada acceso a datos de los objetos se maneje a través de los denominados Repositorios. Existen varios repositorios asociados al dominio, teniendo un Repositorio por cada objeto de negocio, los cuales fueron representados en el diagrama como “IRepository”. Cada repositorio es una interfaz que define los métodos sobre el acceso a datos pero no la implementación en sí misma. La solución fue independiente de la implementación de la capa de acceso a datos, mediante la utilización de interfaces definidas en el componente de dominio e implementadas por el componente de acceso a datos.

La primera interfaz que se puede observar en la figura 20 cumple la función de independizar la solución de la implementación del componente de acceso a datos. La segunda interfaz (“IManager”) se encuentra en el componente encargado de la comunicación entre la aplicación instalada en el cliente y la aplicación servidor. Fue implementada en el componente que contiene la lógica de la aplicación servidor, y es utilizada por el componente que contiene la lógica de la aplicación cliente. La dependencia con IManager utilizada entre los componentes fue realizada con el fin de implementar *Remoting* como solución de comunicación entre los sistemas. Esta tecnología pudo haber sido sustituida por Web Services, debido a que para el alcance del sistema actual estos últimos cumplían con los requisitos necesarios. Se decidió por *Remoting* teniendo en cuenta futuros requerimientos que no fueron incluidos en esta versión, pero que están especificados para su posterior desarrollo.

Para la solución de acceso a datos de la aplicación cliente no se utilizó el patrón “Unit of Work”, debido a que la base de datos local de dicha aplicación se creó para que estuviese muy acoplada a la solución, buscando obtener una solución sencilla de mantener y con pocas clases. Además se utilizó una solución de base de datos local conocida como DB4O [46] (véase Anexo III), solamente con las funcionalidades que implementa la aplicación de escritorio. En este caso únicamente posee las tablas maestras y las tablas relacionadas a los artículos.

11.7.2 Vista de despliegue

A continuación se presenta el diagrama de despliegue que describe los nodos que conforman la topología de hardware donde se ejecuta el sistema:

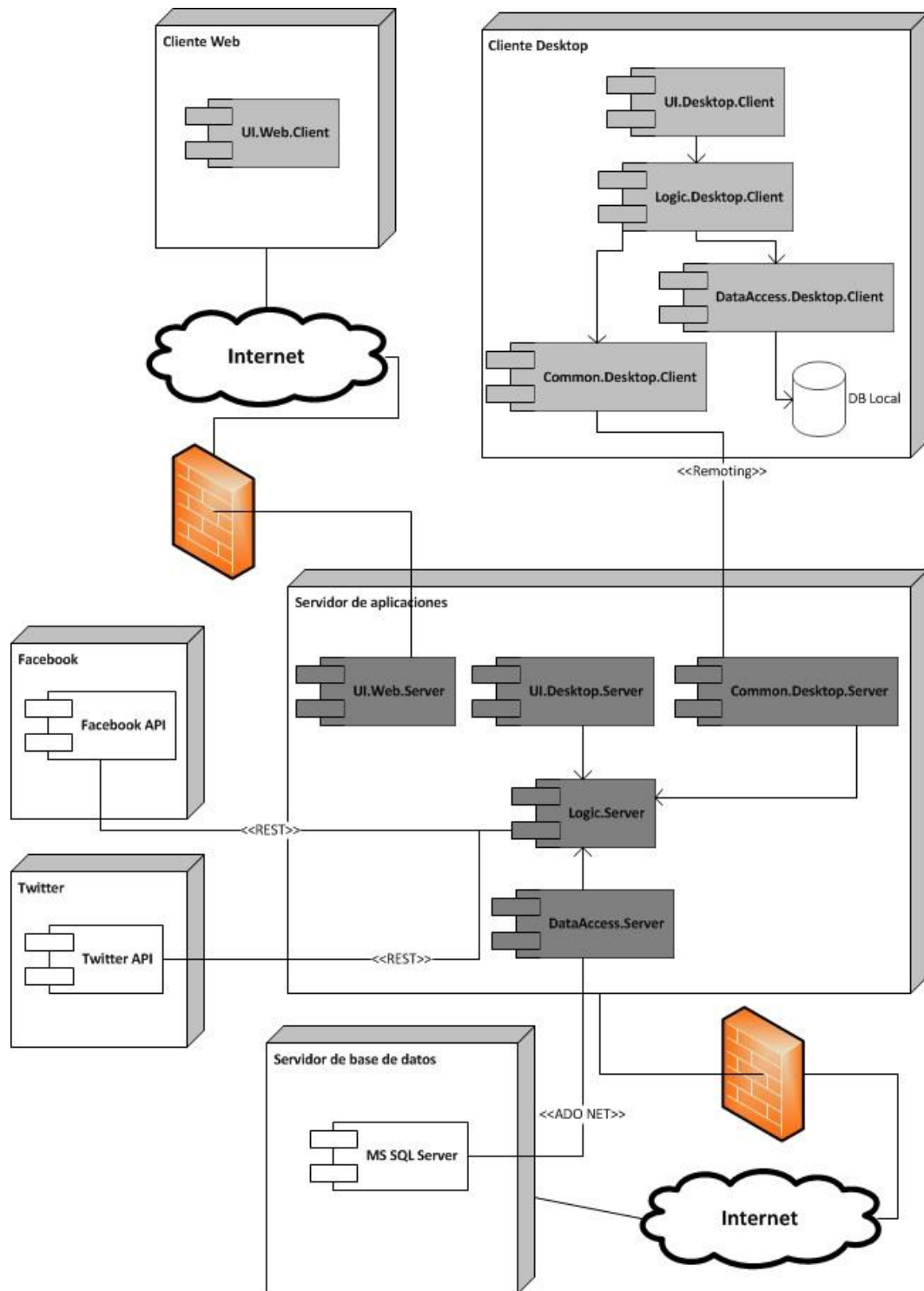


Figura 21 - Vista de despliegue

Como se puede observar en la figura 21, el sistema se encuentra distribuido de forma de poder manejar una cantidad no acotada de nodos clientes (Web y Desktop) que se comunican con un nodo servidor. En la implementación presentada es uno solo, pero si fuese necesario, se podría utilizar un balanceador de cargas y distribuir la aplicación en varios nodos servidores. Este servidor tiene la capa de interfaz, la capa lógica y la capa de acceso a datos, y un nodo servidor de base de datos en donde se centraliza la base de datos del sistema.

Todas las interacciones con las redes sociales se hacen mediante las APIs publicadas por las mismas. La comunicación se realiza mediante un mecanismo de consumo REST. Dicho mecanismo hace que la interacción sea extremadamente sencilla, se utilizan URLs para interactuar con las APIs y el procesamiento de datos es simple y claro.

La base de datos fue pensada para estar alojada en un nodo diferente al de aplicaciones porque se decidió que las imágenes de los productos serían almacenadas en la base de datos por seguridad. Esto implica que por cada artículo agregado se tendrían entre 1 y 5 fotos, las cuales hacen que la base de datos crezca tanto que pueda llegar a afectar el funcionamiento del sistema. Para esto se implementaron dos soluciones:

- La primera solución resuelve que las fotos de los artículos sean almacenadas en la base de datos, pero el sistema está pensado para que solo se levanten las fotos de la base la primera vez que son requeridas. Una vez que son requeridas serán almacenadas en el *file system* y desde entonces serán siempre consultadas desde ahí. De esta manera, la manipulación de imágenes en la base de datos se reduciría drásticamente, mejorando el desempeño del sistema.
- La segunda solución es la de utilizar un nodo servidor de base de datos. Esto permitiría manejar mejor el tamaño y los recursos que el sistema necesitaría en cada momento y facilitaría las tareas de *backup* y de *restore*.

Se identificó que una de las tareas que consumía muchos recursos en el nodo servidor de aplicaciones era el envío de correos electrónicos. Ya sea para la *news letter* que se enviaban a todos los usuarios que así lo hubiesen configurado o para el registro de usuarios u olvido de contraseña. Todos estos correos consumían muchos recursos al momento de ser ejecutados y hacían que el hilo de ejecución del sistema se demorara bastante. Para resolver esto, se tomó la decisión de implementar un componente independiente de todos los demás que se encargara del envío de correos. Dicho componente puede ser colocado (si así fuese necesario en un futuro) en un nodo independiente sin necesidad de modificar el código, simplemente mediante la realización de algunas configuraciones. Asimismo, se implementaron todas las llamadas al componente en hilos de ejecución independientes del hilo principal. Dado que no afecta a la experiencia del usuario que el correo se envíe en el momento o en unos segundos después, se decidió que el hilo principal no debía esperar a que el correo fuese enviado. De ocurrir un error el usuario podría pedir que se le envíe nuevamente el correo o enviar un correo a los administradores de sistema pidiendo que se corrija el problema.

Para el uso de los administradores también se implementó una lista de correos que retorna “errores” de comunicación. Pueden ser correos que estén mal ingresados o que se cerraron o algún tipo de error en especial. Estos errores son documentados y son almacenados automáticamente por el componente de envío de correos electrónicos para su posterior revisión.

11.7.3 Vista de procesos

En la siguiente vista se buscará detallar tres procesos del sistema que implican aspectos de sincronización y concurrencia. Cada uno de estos procesos fueron discutidos por el equipo de tesis para buscar la mejor solución posible con los conocimientos que tenían al alcance.

11.7.3.1 Carga de las imágenes en las aplicaciones Web

En la figura 22 se muestran las interacciones que tienen las diferentes capas de la aplicación para lograr el cargado de imágenes de un artículo por medio de la Web, proceso lanzado por medio de un usuario de tiendas.

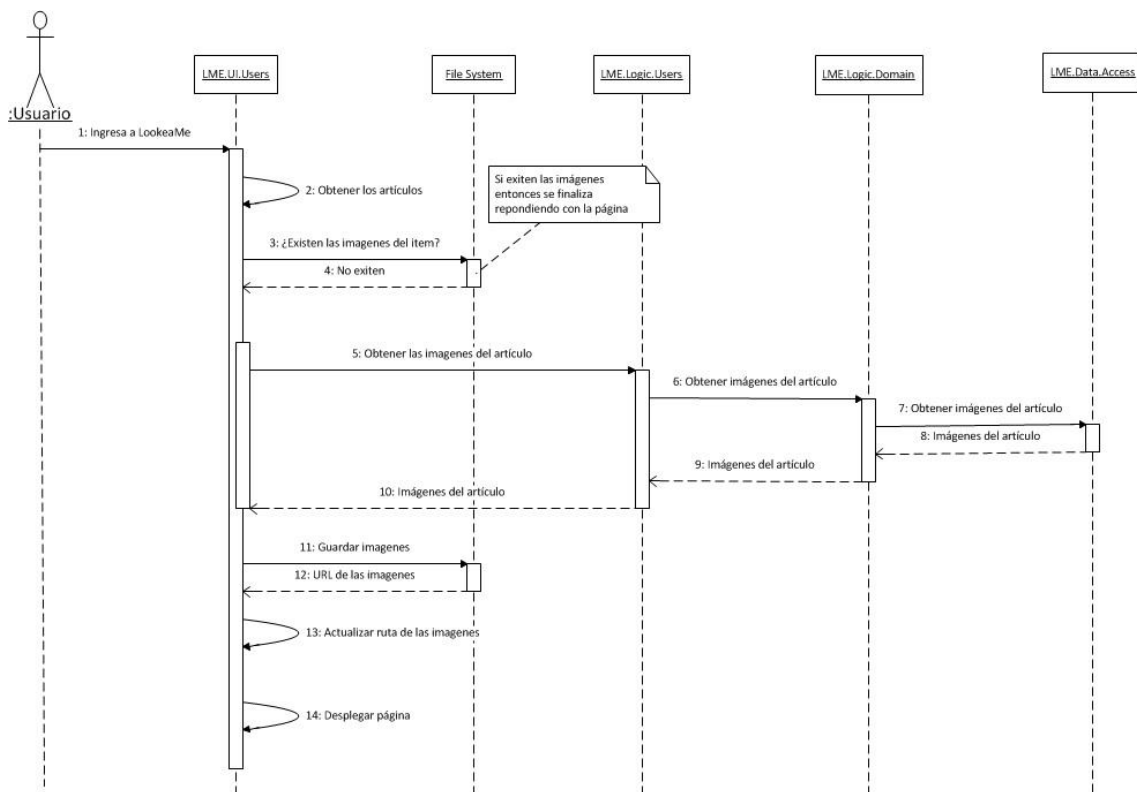


Figura 22 - Diagrama de interacción de carga de imágenes

11.7.3.2 Envío de mails

En la figura 23 se muestran las interacciones que tienen las diferentes capas de la aplicación para realizar el envío de mails por proceso batch.

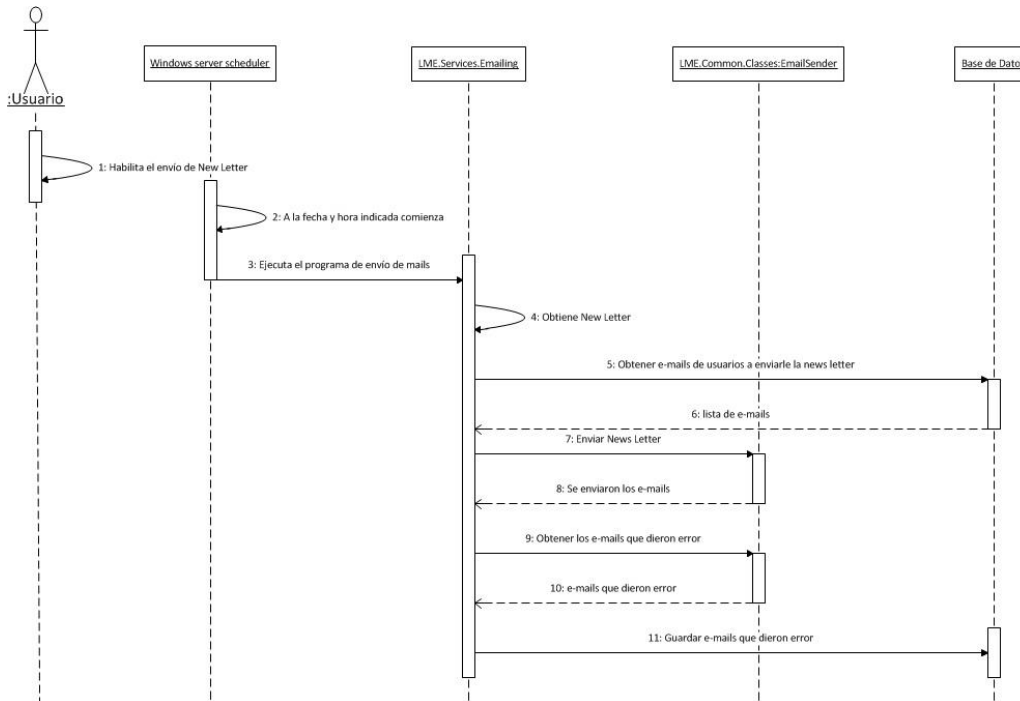


Figura 23 - Diagrama de interacción de envío de mails

11.7.3.3 Sincronización entre cliente y servidor (carga masiva)

En la figura 24 se muestran las interacciones entre la aplicación de escritorio de las tiendas y el servidor con la base de datos centralizada, de forma de cargar masivamente los datos provenientes de este primer sistema.

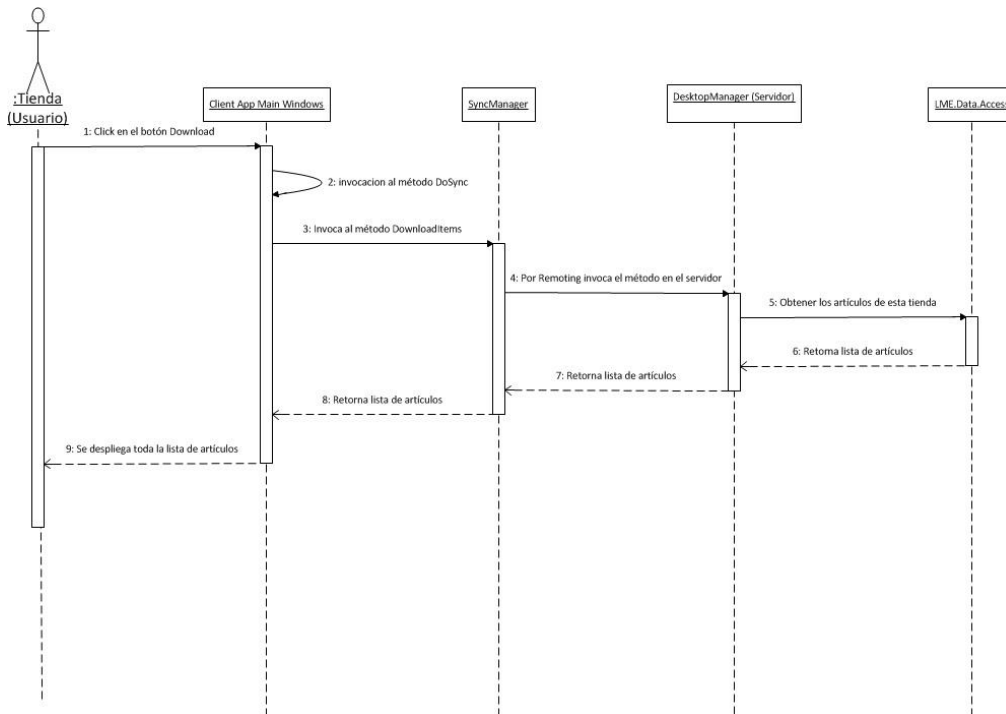


Figura 24 - Diagrama de interacción de la sincronización

11.7.4 Vista de casos de uso

En la Vista de casos de uso se detallan los correspondientes a cada usuario del sistema.

El actor “Usuario” es quien interactúa con la aplicación central de Lookea.Me, es decir, quien visita la página de usuarios, busca artículos, invita amigos, comparte en redes sociales, reserva y compra artículos, entre otras funcionalidades que se detallan en el diagrama a continuación. Su vista de casos de uso se puede observar en la figura 25.

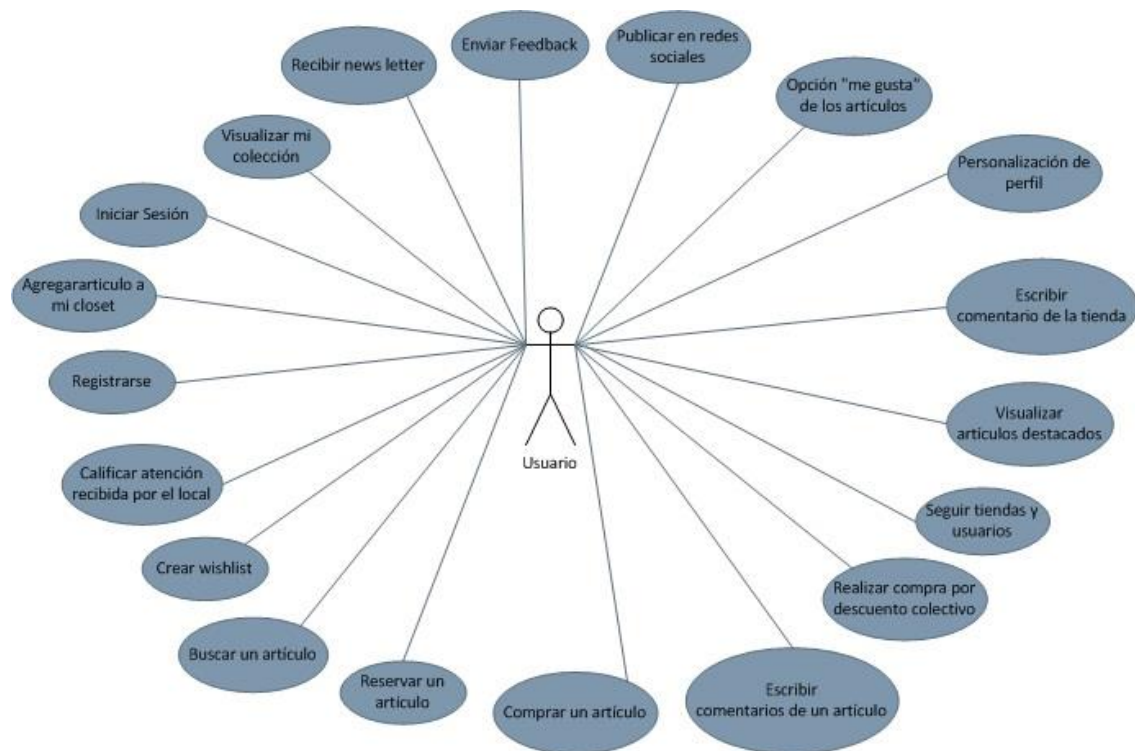


Figura 25 - Casos de uso del actor Usuario

El actor “Tienda” es quien interactúa con el usuario de la tienda a la cual corresponde. Dicho actor ingresa al sistema con su nombre y contraseña particular del local para recibir las reservas, actualizar artículos, administrar la tienda, entre otras funcionalidades que se detallan en el diagrama a continuación (Figura 26).

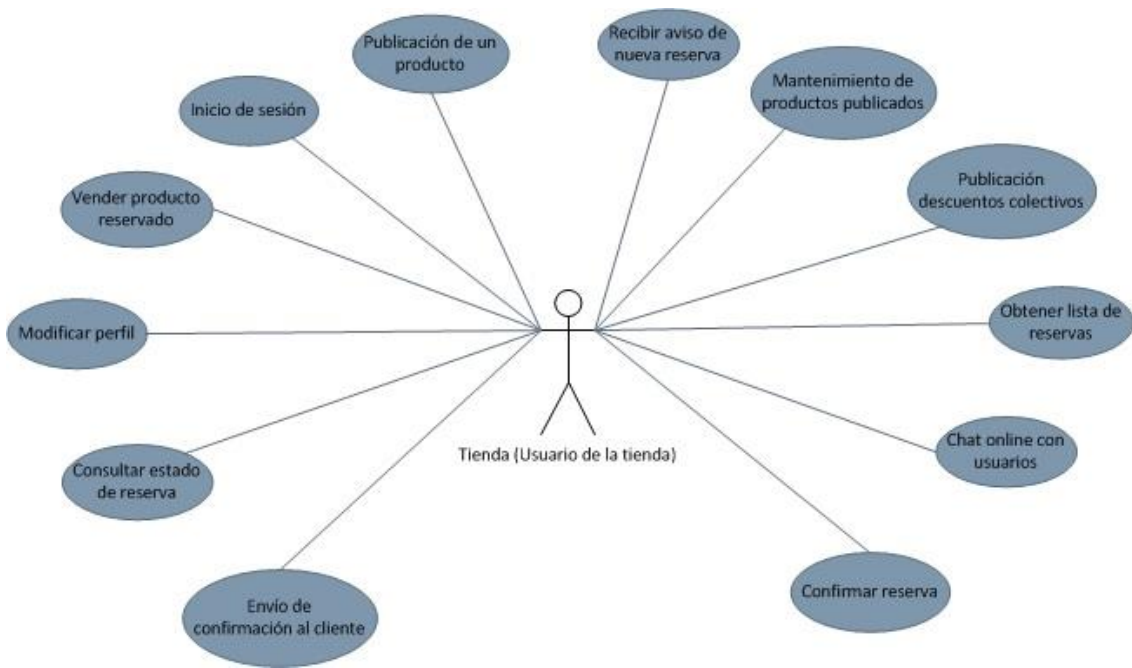


Figura 26 - Casos de uso del actor Tienda

El actor ‘administrador’ es el responsable del sistema en general. Es el único que puede dar de alta una tienda, revisar los *logs* del sistema, los *feedbacks* de los usuarios, entre otras funcionalidades que se detallan en el diagrama a continuación (Figura 27).

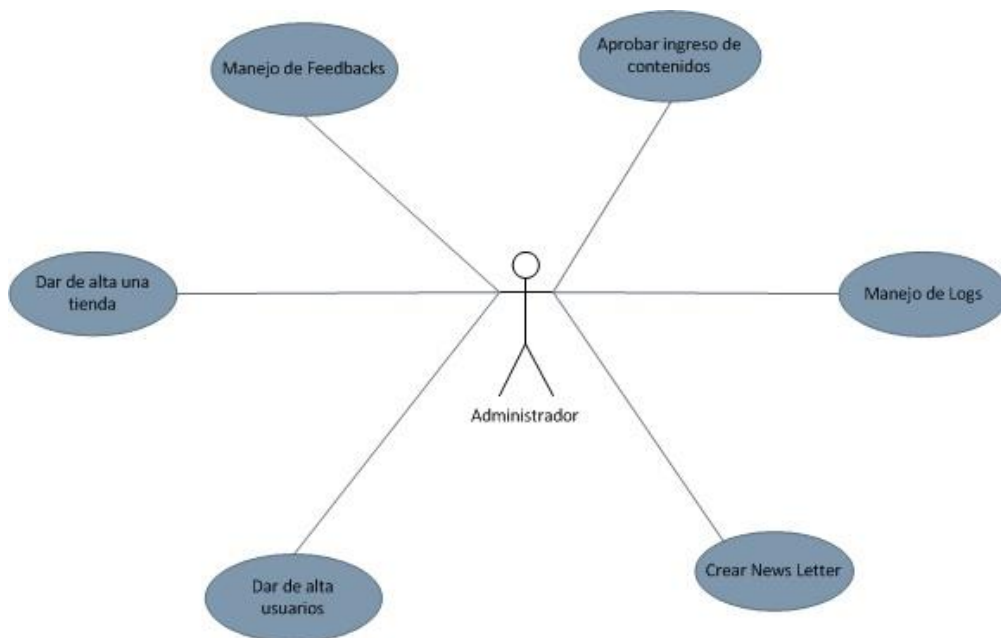


Figura 27 - Casos de uso del actor administrador

12 Gestión de la configuración

12.1 Introducción

En el siguiente capítulo se detallará el proceso utilizado para la gestión de la configuración del software (SCM – Software Configuration Management). El mismo abarca las herramientas, los elementos de configuración, actividades, respaldos realizados y los circuitos de control de cambio definidos por el equipo.

12.2 Herramientas

En la etapa de investigación, se tomó la decisión de realizar toda la etapa de desarrollo con un servidor propio, dado que se tendría mayor control sobre el mismo, mayor confidencialidad sobre el código creado y sería gratuito, asumiéndose por parte del equipo riesgos como la posible falla del equipo, la lentitud de la red en algunos momentos del día y el trabajo de mantener el servidor en buen funcionamiento.

El servidor de desarrollo fue una computadora de escritorio de la casa del responsable de SCM. Se le instaló Windows Server 2008 R2 dado que era la versión de Windows Server con la que los integrantes del equipo tenían experiencia de trabajo.

Para la gestión de los ECS (Elementos de Configuración de Software) de tipo código fuente se utilizó Microsoft Visual Studio Team Foundation Server, lo cual se instaló en el servidor de desarrollo y no necesitó capacitación por parte del responsable, dado que todo el equipo ya tenía experiencia con esa herramienta. Para los ECS de tipo documentos se utilizó Dropbox, debido a que el equipo ya había trabajado en otros trabajos dentro de la Universidad con esta herramienta y resultaba muy eficiente.

12.3 Elementos de configuración

A continuación se detallan los elementos de configuración que se identificaron en el proyecto:

12.3.1 Documentos

Dentro de los documentos identificados como ECS tenemos aquellos generados para el cliente, como por ejemplo la lista de requerimientos, como así también los generados para cumplir con las diferentes instancias académicas, como las revisiones, la entrega final y la defensa del proyecto.

A continuación se listan todos los documentos del proyecto identificados como elementos de configuración del software:

- Plan de negocio.
- Plan de gestión de la configuración (SCM).
- Minutas de reunión.
- Presentaciones para las revisiones.
- Plan de gestión de riesgos.
- Documento de especificación de requerimientos (ESRE).
- Estándares de codificación.
- Documentos de diseño de alto nivel.
- El plan de pruebas.
- El plan de pruebas de usabilidad.
- Datos del proyecto.
- Manuales de usuario.
- Documento de registro de horas.
- Documento de control de avances interno con las funcionalidades de cada sprint.
- Datos de encuestas.
- Datos de prueba.

12.3.2 Código fuente

A continuación se listan todos los elementos de configuración que categorizamos como código fuente:

- Carpeta de *deploy* del sistema para publicar en el servidor.
- Código fuente del programa.

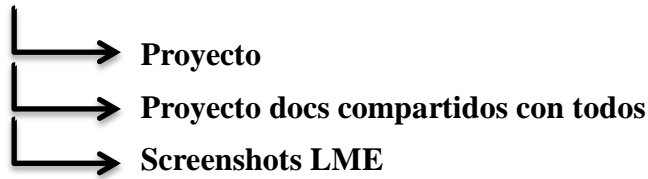
12.4 Repositorio

El repositorio de documentos en Dropbox se manejó en tres carpetas:

- Una carpeta para todos los documentos internos del equipo que era compartido con el tutor para que pudiera realizar correcciones en los diferentes documentos,
- Una segunda carpeta para compartir documentos con los clientes internos del equipo,
- Una última carpeta para compartir documentos y archivos con la diseñadora que colaboró con el equipo en la etapa de diseño.

La estructura de Dropbox quedó de la siguiente manera:

Dropbox



Dentro de la carpeta “Proyecto” se crearon carpetas para poder mantener mayor control y orden a lo largo de todo el proceso, con la estructura que muestra la figura 28.

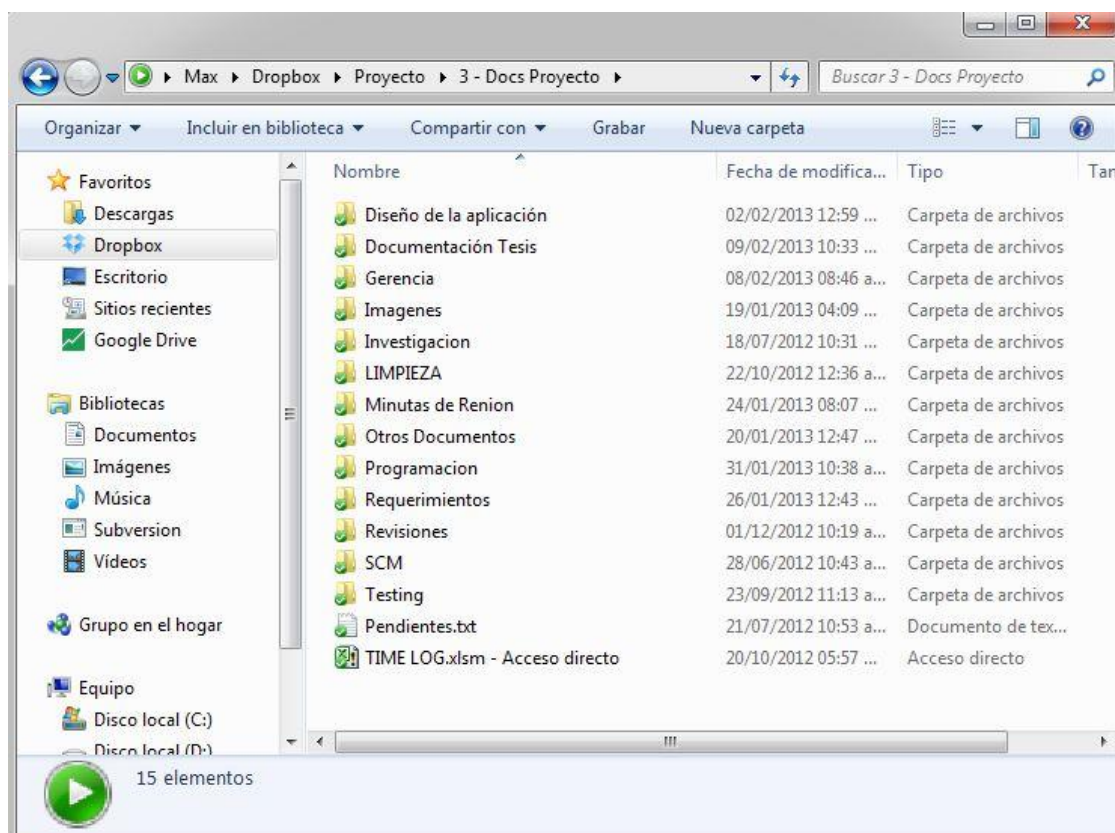


Figura 28 - Repositorio de documentos

Dentro de todos los documentos, el más utilizado por el equipo fue el “TIME LOG” y para hacerlo más accesible, se creó un acceso directo a dicho documento.

En cuanto al versionado del código fuente, se trabajó sobre un *team project* con nombre LME. Dentro de este proyecto fueron creadas y versionadas las soluciones desarrolladas con Visual Studio.

La estructura física del Team Project LME se muestra en la figura 29.

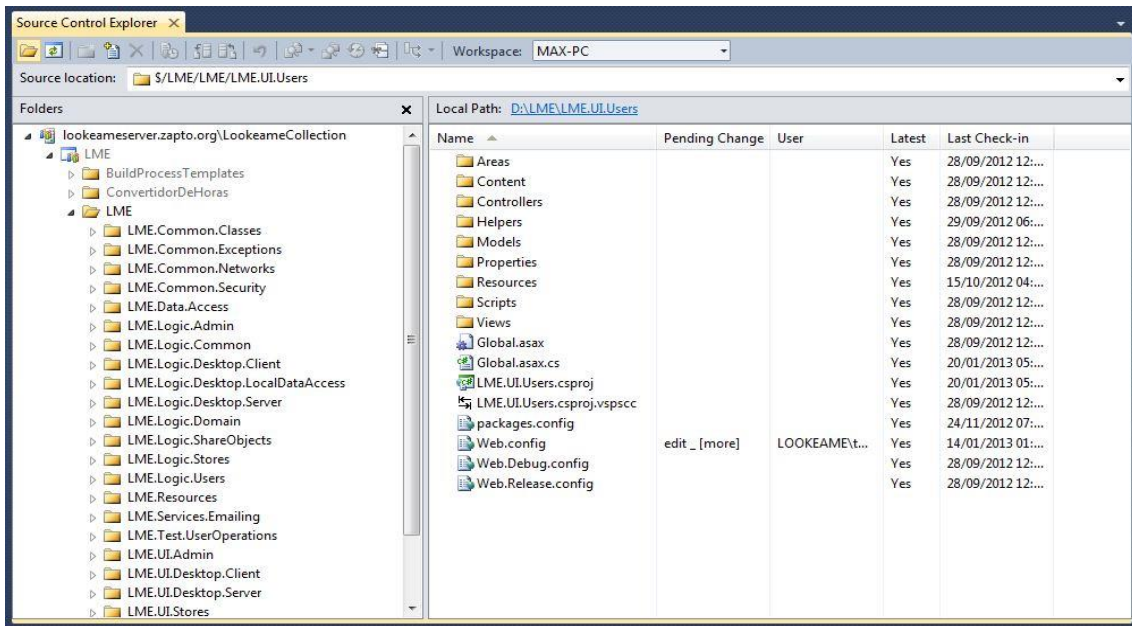


Figura 29 - Repositorio de código fuente

01-LME.UI: En esta carpeta se encuentran todos los proyectos de interfaz gráfica, los tres proyectos Web y los proyectos cliente y servidor del sistema de escritorio. Estos proyectos incluyen únicamente lo que compete a la interfaz gráfica.

02-LME.Logic: Esta carpeta contiene todos los proyectos con la lógica de los diferentes sistemas que integran Lookeame.

03-LME.Services: Esta carpeta contiene los proyectos denominados de servicio, como el proyecto de envío de correos electrónicos. Está definida para que se alojen todos los proyectos que en un futuro cumplan la función de prestar servicios al sistema.

04-LME.Data: Esta carpeta tiene el proyecto de acceso a datos. Se incluyó un único proyecto que maneja el acceso a datos, dado que se es una única base de datos en el servidor que unifica toda la información del sistema.

05-LME.Common: La carpeta Common contiene los proyectos de uso común entre los diferentes sistemas, como por ejemplo el proyecto de excepciones del sistema, que es utilizado todo a lo largo de los diferentes sistemas de Lookeame.

06-LME.Resources: Todas las DLLs que son externas al *framework* .Net 4.0 y ASP.Net MVC 3.0 utilizadas por Lookeame se encuentran en esta carpeta.

10-LME.Test: Los proyectos de prueba unitaria que se crearon a lo largo del proceso de desarrollo están alojados en esta carpeta.

12.5 Actividades

El objetivo primordial del proceso de SCM en Looke.Me fue lograr una correcta configuración de software de forma de mitigar los defectos del sistema, asegurar el acceso y disponibilidad de los componentes, y evitar la pérdida de información.

Para asegurar todas estas características mencionadas anteriormente se definieron las siguientes actividades:

- **Identificación de los ECS:** Antes de poder realizar tareas de gestión de la configuración fue necesario identificar los elementos de la configuración del *software*.
- **Selección y configuración de herramientas:** Se seleccionaron las herramientas para utilizar en la gestión de la configuración. Las mismas se seleccionaron según la experiencia de los integrantes del equipo. La configuración estuvo a cargo del encargado de SCM.
- **Definición de CCC y reglas de control de cambios de código:** Una de las ventajas del pequeño tamaño del equipo de desarrollo, fue que la definición de reglas a seguir al momento de desarrollar código y de modificar documentos se realizaban de tal manera que todos los integrantes se sentían cómodos con los procesos, pudiendo ejecutarlos sin inconvenientes y así asegurando una mejor calidad de productos finales.
- **Respaldos:** Los respaldos necesitaban ser realizados rutinariamente para poder asegurar que no se perdiera información o valor en el producto si ocurría algún error o problema importante con el servidor local, pudiendo recuperarse la información fácilmente.

12.6 Control de cambio

Se definió un CCC (Circuito de Control de Cambios) y sus reglas asociadas de modo de controlar los cambios de código fuente. A continuación detallamos cada uno de ellos:

12.6.1 Circuito de control de cambios de documentos

Dada la naturaleza del equipo, cada vez que un integrante del mismo modificaba un documento o lo creaba, el otro integrante del equipo lo validaba. Si el documento necesitaba ser corregido por el tutor, se le informaba luego de que tuviese la aprobación de todo el equipo. En la figura 30 se muestra el diagrama de actividad que describe dicho proceso.

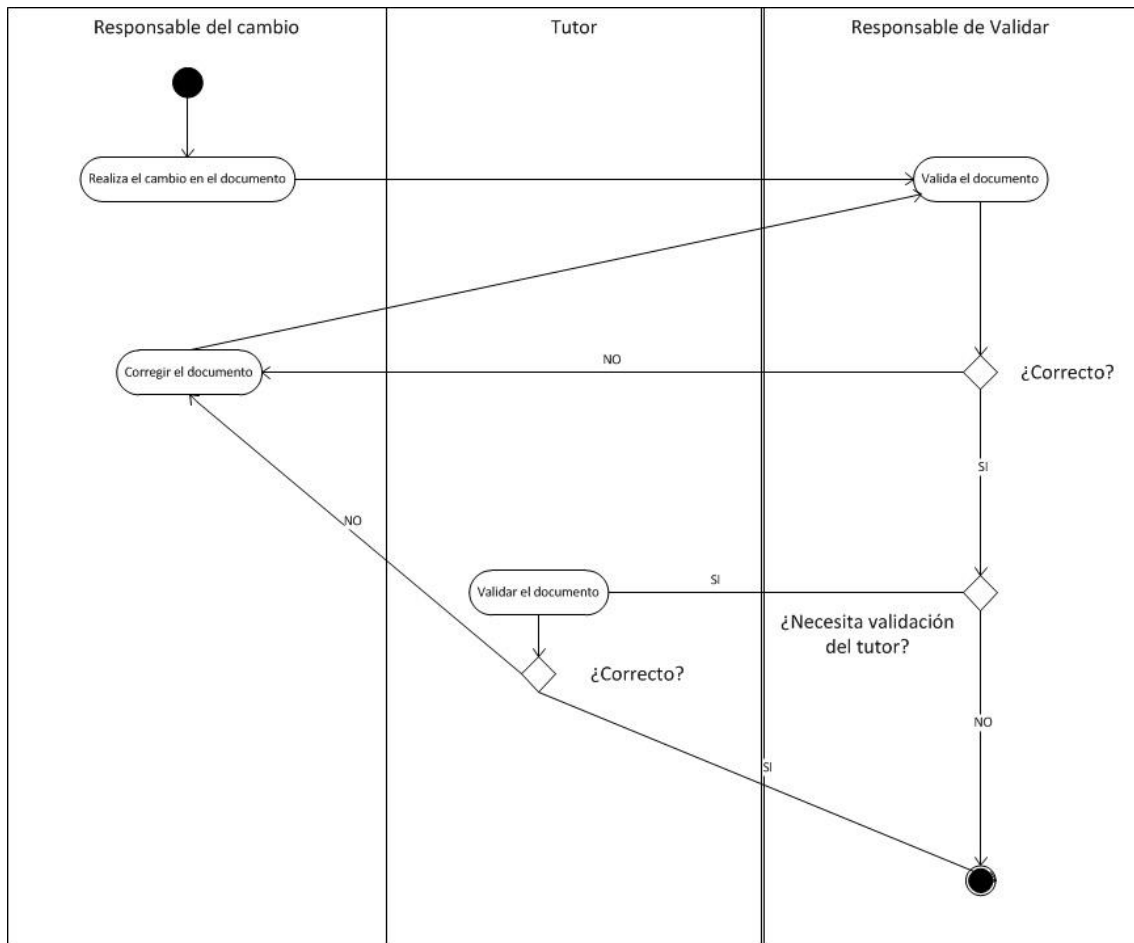


Figura 30 - Circuito de control de cambios de documentos

12.6.2 Reglas de control de cambios de código fuente

Significaba mucho esfuerzo el plantear que cada vez que un integrante del equipo realizaba un cambio sobre el código el otro debía validarlo, por lo que el esfuerzo de validar el código se realiza al finalizar de cada sprint (véase Anexo VII). Lo que se definió para controlar el cambio del código fue un conjunto de reglas que cada uno de los integrantes del equipo debía verificar antes de subir código nuevo al repositorio.

Las reglas definidas por el encargado de SCM eran las siguientes:

- Asegurarse que se estaba trabajando sobre la última versión del código.
- El código no debía tener errores de compilación.
- Si el nuevo código afectaba alguna de las funcionalidades para las cuales se habían definido pruebas unitarias, se debían correr las pruebas nuevamente hasta que las mismas obtengan los resultados esperados.

12.7 Respaldos

Uno de los riesgos más significativos para el encargado de SCM era que al tener un servidor de desarrollo manejado localmente, podría llegarse a perder todo lo que se hubiese desarrollado si ocurría alguna catástrofe con el servidor.

Para mitigar este riesgo, el encargado de SCM realizaba un respaldo del servidor una vez por mes y un respaldo del Team Foundation Server una vez cada tres semanas. Los documentos no necesitaban ser respaldados dado que estaban respaldados en Dropbox, herramienta que automáticamente guarda las versiones anteriores de cada uno de los documentos y al ser un servicio mundialmente usado, y con copias locales en cada una de las estaciones participantes, la probabilidad de que se perdiera información era prácticamente nula.

12.8 Análisis de problemas encontrados

12.8.1 Lectura sucia en los documentos

Uno de los mayores problemas que presentaba Dropbox como herramienta de versionado de documentos es que si dos o más miembros del equipo editaban el mismo documento al mismo tiempo, no veían en tiempo real lo que los demás estaban haciendo sobre el documento. Esto generaba un problema conocido en los sistemas de sincronización como “lectura sucia de datos”. Para evitar esto, se asignaron encargados para cada documento y si uno de los integrantes del equipo necesitaba editar un documento del cual no era encargado, debía notificar al encargado previamente para que no lo editaran al mismo tiempo.

12.8.2 Mensajes de actualización constantes

Podía resultar un problema menor pero al estar trabajando mucho tiempo, los mensajes que Dropbox realizaba a todos los integrantes del equipo cuando otro integrante estaba actualizando documentos constantemente eran muy molestos. Para resolver esto, lo que se decidió fue que cada integrante hiciera además una copia local del documento cuando estaba trabajando en el mismo, y lo subiera a Dropbox cuando terminara de editarlo.

12.8.3 Errores por DLLs no importadas

El error más común relacionado al código fuente era que al agregarse una DLL al sistema que no pertenecía al *framework* de desarrollo, el desarrollador que la agregaba se olvidaba de copiar dicha DLL dentro de la carpeta de recursos que tenía la solución, lo que ocasionaba que al subirlo al TFS se perdía la referencia y para todos los demás integrantes del equipo, el sistema dejaba de compilar.

13 Gestión de la calidad

13.1 Introducción

La gestión de la calidad no fue planteada desde el comienzo del proyecto como un rol que un integrante del equipo debía cumplir, sino que se definió como un conjunto de buenas prácticas y actividades que todo el equipo debía comprometerse a cumplir, incluidos los clientes internos.

La gestión de la calidad tenía como meta la calidad de Looke.Me como producto de software, así como de los procesos orientados al producto y orientados a la gestión.

Detallaremos a continuación las actividades que se realizaron a lo largo del proyecto para gestionar la calidad.

13.2 Actividades de SQA

13.2.1 Actividades sobre la gestión

La mejora continua del desempeño global del equipo fue un objetivo permanente del proyecto. La gestión de la calidad fue centrada en el modelo de mejora conocido como PDCA (Plan Do Check Act), propuesto por Deming Juran [47].

En la figura 31 se muestra cómo interactúan las diferentes partes que componen la gestión de la calidad del proyecto.

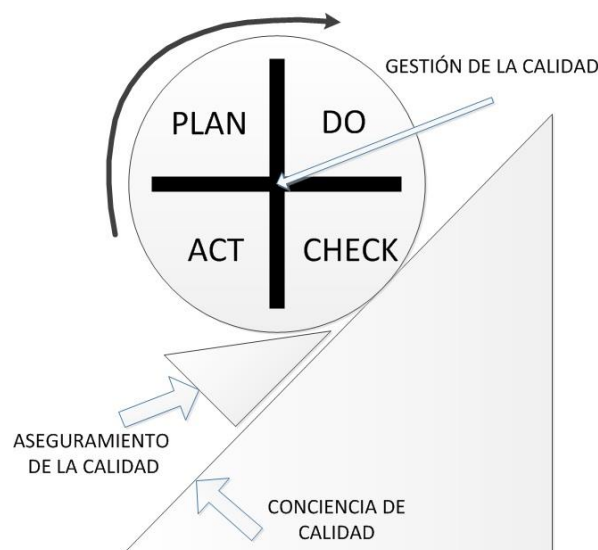


Figura 31 - Modelo de mejor PDCA

La aplicación del concepto de aseguramiento de la calidad en Lookea.ME fue lo que permitió asegurar que cada vuelta del ciclo PDCA implicaba mejorar en la calidad del proceso y del producto.

13.2.2 Actividades sobre el producto

13.2.2.1 Circuito de control de cambios de requerimientos

En Lookea.Me el riesgo de que un requerimiento cambiara o que aparecieran nuevos requerimientos era una preocupación constante por parte del responsable de la ingeniería de requerimientos. Pudiendo aparece nuevos requerimiento porque el entorno y las necesidades habían cambiado o porque el cliente necesitaba algo diferente a lo originalmente relevado. Algunas de estas razones podían ser mitigadas pero otras debían simplemente aceptarse, y por lo tanto, incorporarse adecuadamente a través del Circuito de Control de Cambios.

Lookea.Me desde un comienzo mostró ser un sistema con una naturaleza cambiante, por muchas razones, siendo las principales:

- Un área del negocio desconocida
- Una arquitectura de negocio muy desafiante (multi-lado, debe contemplar ambos usuarios) (véase Anexo XIV)
- Un cliente interno con pocos conocimientos del proceso de software

Para el equipo simplemente aceptar este riesgo de cambio de requerimientos no era una posibilidad porque no se iba a poder llegar a un producto con la calidad de acuerdo a la definida como objetivo. En ese contexto, se decidieron dos acciones:

- La primera acción fue realizar una ingeniería de requerimientos extremadamente detallada, con la intención de conocer bien a fondo el negocio con reuniones extensas con el cliente interno para entender la naturaleza del mismo.
- La segunda acción fue realizar un circuito de control de cambios de los requerimientos, por el cual se pretendió evitar que los posibles cambios afectaran el cronograma y el producto final. (Figura 32)

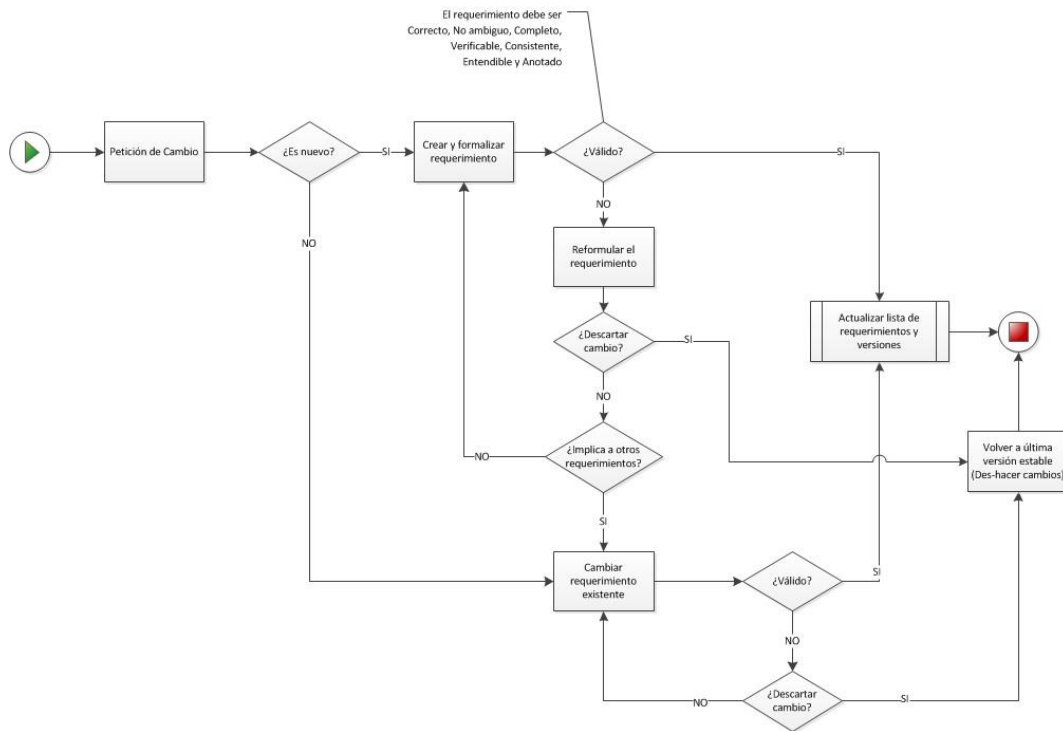


Figura 32 - Circuito de control de cambios de requerimientos primera parte

Cuando una petición de cambio llegaba al equipo se seguían los pasos detallados en el circuito de control de cambios. En caso de que el cambio se determinara como válido, se debía actualizar la lista de requerimientos y versiones para lo que se había creado otro circuito que sirvió de guía para el equipo a lo largo del proyecto. El mismo se muestra en la figura 33.

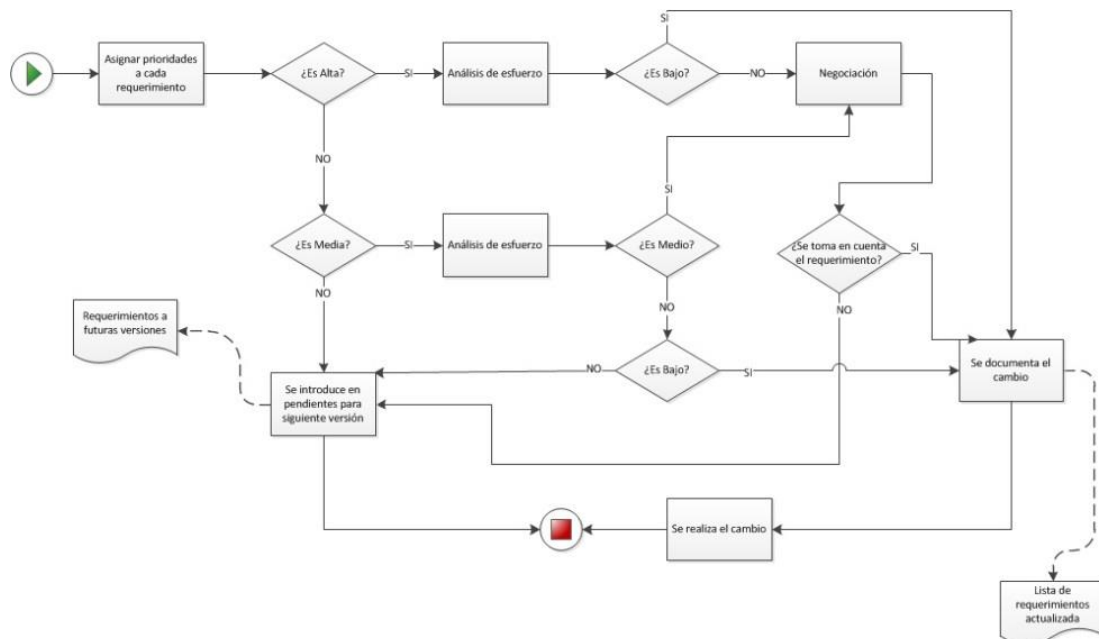


Figura 33 - Circuito de control de cambios de requerimiento segunda parte

Para simplificar la toma de decisiones el equipo se basó en el proceso detallado anteriormente y creó una tabla para saber cuáles requerimientos se implementan y cuáles se agregan a versiones posteriores del sistema Lookea.Me.

En la figura 34 se ejemplifica cómo el equipo decidía el futuro de los requerimientos que eran solicitados por parte de los clientes, siendo estos nuevos o modificaciones sobre alguno existente. Esta tabla facilitaba hacerle comprender al cliente cómo funcionaba la toma de decisiones interna al equipo de tesis:

Prioridad	Esfuerzo	Acción
Alta	Alto	Negociar
Alta	Medio	Negociar
Alta	Bajo	Versión Actual
Media	Alto	Versión Futura
Media	Medio	Negociar
Media	Bajo	Versión Actual
Baja	Alto	Versión Futura
Baja	Medio	Versión Futura
Baja	Bajo	Versión Futura

Figura 34 – Toma de decisiones

Por último, se diseñó un grafo de dependencias entre los requerimientos del sistema, de modo que hubiese una forma más gráfica de explicar a los clientes internos que un cambio en un requerimiento, por más pequeño que fuese, podía llevar mucho esfuerzo por la dependencia que dicho requerimiento tenía con otros.

Este grafo fue creado en base a una sugerencia que se le presentó al equipo en la primera revisión y resultó de gran utilidad para el proyecto (Figura 35).

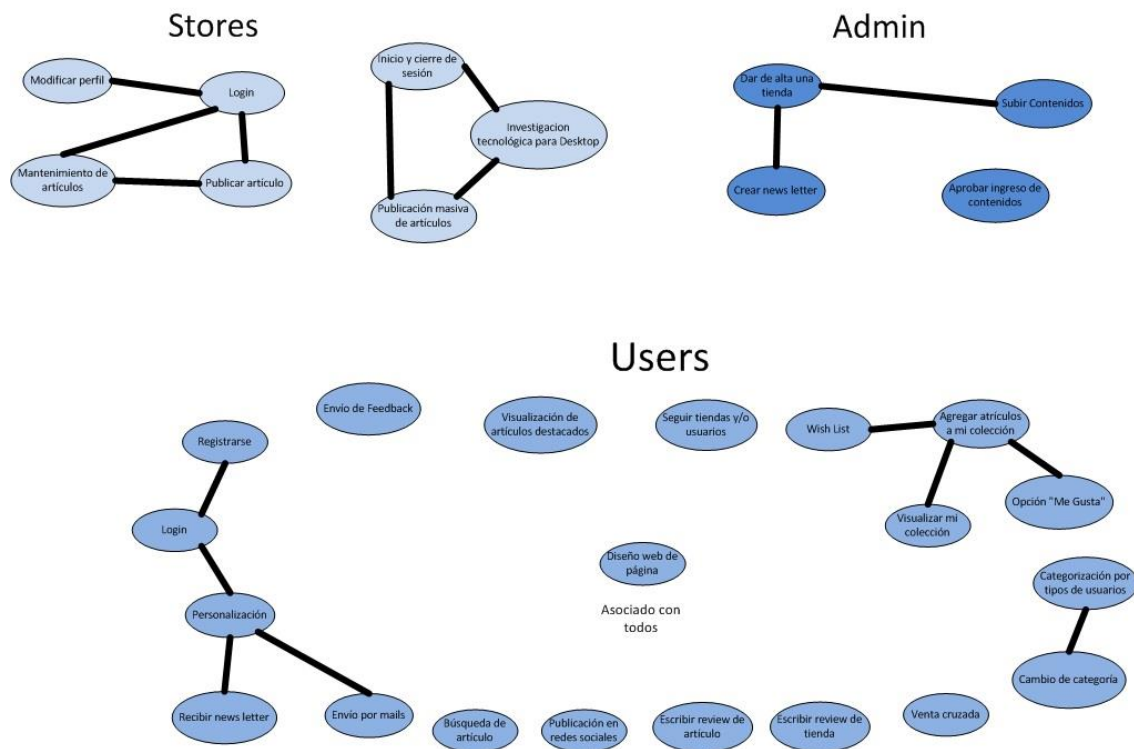


Figura 35 - Grafo de dependencias

13.2.2.2 Revisión del código fuente

Se confeccionó un documento de estándares de código fuente y base de datos (véase Anexo XVII) basado en los definidos por los *frameworks* utilizados, de manera de explicitar los mismos en un documento, y utilizarlo para realizar las revisiones de código.

Las revisiones de código se hacían una vez finalizado cada *sprint* y eran ejecutadas por el desarrollador que no había escrito el código. De esta forma se podía asegurar mayor objetividad. Esta actividad fue realizada para poder asegurar la calidad en el código que luego aseguraba que el producto fuera mantenible y escalable.

13.2.2.3 Plan de Pruebas

Se realizó un plan de pruebas completo (véase Anexo VII), buscando seleccionar las pruebas que fuesen más relevantes para lograr los objetivos que el equipo se había planteado y siempre teniendo en cuenta los recursos con los que se contaba para realizar las mismas.

Se realizaron pruebas unitarias para las funcionalidades básicas, pruebas de integración, pruebas de sistema y pruebas de validación, definiéndose para cada una de estas categorías los respectivos planes de prueba.

13.2.2.4 Pruebas de Usabilidad

La usabilidad, la facilidad con la que un usuario aprende a utilizar el sistema, lo agradable que le resulta la interfaz y lo intuitivo que le parece el sistema, eran características claves para el éxito del Lookea.Me.

Para esto se creó un plan de pruebas de usabilidad (véase Anexo VIII) que detalla todos los pasos que se siguieron para realizar las pruebas de usabilidad que se definieron como más convenientes.

13.3 Calidad del proceso de desarrollo

13.3.1 Introducción

Buscando utilizar todo el potencial que se puede obtener de un equipo que trabaja junto hace muchos años y que comparte buenas prácticas de programación, se decidió utilizar PSP y TSP como guías de buenas prácticas de programación para poder asegurar la calidad del producto.

Considerando la naturaleza del producto final, se tuvo presente la necesidad de construir una aplicación que resultara intuitiva a todo usuario y que hiciera más sencillo la ejecución de diferentes procesos, sin agregar complejidad ni problemáticas adicional a los mismos. Por lo tanto, se construyó un sólido plan de pruebas (véase Anexo VII) haciendo hincapié en las pruebas de usabilidad.

13.3.2 PSP

Personal Software Process está diseñado para ayudar a los desarrolladores profesionales a utilizar consistentemente buenas prácticas de ingeniería. Muestra cómo planificar y controlar el trabajo personal, utilizar un proceso definido y medible, establecer metas medibles, controlar el desempeño comparando contra las metas, cómo manejar la calidad desde el comienzo de un trabajo, cómo analizar los resultados de cada trabajo y cómo usar los resultados para mejorar el proceso para un próximo proyecto.

Al basarse en los principios de planificación y calidad de PSP según el libro *PSP Lite* de Watts S. Humphrey [24], el equipo tomó una lista de principios a los cuales todos los integrantes del equipo debían comprometerse:

- “Para poder producir productos de calidad, los ingenieros deben sentirse personalmente responsables de sus productos. Excelente productos no se producen de causalidad, los ingenieros deben esforzarse por hacer trabajo de calidad.”
- “Cuesta menos encontrar y solucionar defectos en los comienzos de un proyecto que en el final.”
- “Es más eficiente prevenir un defecto que buscar y solucionar defectos.”
- “La forma correcta es siempre la más rápida y barata de hacer el trabajo.”
- “Para poder hacer un trabajo de ingeniería de software correctamente, los ingenieros deben planificar su trabajo antes de comprometerse o comenzar un trabajo, y deben utilizar un proceso definido para planificar su trabajo. Para entender su desempeño personal deben primero medir el tiempo que les consume hacer cada uno de los pasos de su trabajo, los defectos que ingresan al trabajo y los que solucionan, y los tamaños de los productos que realizan.”

Para lograr utilizar PSP fue fundamental que todos los integrantes del equipo cursaran la materia “Prácticas de mejora en ingeniería de software”, que le permitió a cada uno definir su proceso de desarrollo, medir los tiempos que le lleva realizar cada tarea y tener una base de datos con la que se pueda comprar a futuro.

PSP puede ser utilizado para muchos fines. Uno de los más utilizados es para obtener una mejor planificación, pero en el caso de este proyecto, la misma no era la motivación principal para utilizar PSP, sino la mejora continua del proceso por el cual se creaba el producto. Cada integrante debía buscar auto superarse al momento de desarrollar y los principios de PSP servían de guía para lograr ese objetivo. Por esta razón fundamental el equipo decidió utilizar PSP.

13.3.3 TSP

“El *Team Software Process* guía a equipos de ingenieros en el desarrollo de productos de software. La experiencia con TSP muestra que mejora la calidad y productividad de los equipos de ingenieros mientras los ayuda a lograr cumplir con las metas de costo y cronograma.” [5]

“TSP está pensado para grupos de entre 2 y 20 integrantes. Y además asume que cada uno de esos integrantes está correctamente capacitado y tengan un proceso de desarrollo maduro. Por eso mismo se dice que es necesario que cada integrante de un grupo TSP tenga claras nociones de PSP dado que es importante que todos partan de una misma base de conocimiento en buenas prácticas de desarrollo de software.” [5]

La forma en la que este equipo utilizó TSP fue en definir un ciclo de desarrollo que siguiera las siguientes prácticas:

- Reunión de Lanzamiento
 - Revisar objetivos
 - Revisar compromisos
- Desarrollar
- Post mortem
 - Reunir experiencias
 - Revisar objetivos de la iteración
 - Marcar avances

Cada desarrollador ajustó su proceso personal de desarrollo (PSP) a los hitos que el equipo definía como “momentos de entrega” o “fechas de revisiones de avances”, y luego podía controlar su trabajo utilizando los principios de PSP propuestos.

Para poder llevar adelante TSP fue necesario definir roles en el equipo, los roles que se definieron fueron los siguientes:

- Gestor de desarrollo – Asignado al responsable del área de desarrollo
- Gestor de planificación – Asignado al responsable de gerencia
- Gestor de actividades de soporte – Asignado al responsable de SCM

Por su parte, dado el tamaño del equipo y el tiempo que insume cada rol que propone TSP, se decidió no incluir los roles de Líder de Equipo y Gestor de Calidad, considerando que estos roles podrían ser abarcados directamente mediante el uso de PSP.

13.3.4 Pruebas

13.3.4.1 Introducción

Junto a las pruebas planeadas en el plan de pruebas (véase Anexo VII), se crearon métricas para poder evaluar los resultados y tomar decisiones. Las pruebas de usabilidad fueron consideradas desde un comienzo como las más importantes dado que la validación del usuario final era un factor central para el éxito de Looke.Me por eso se creó un plan de pruebas de usabilidad (véase Anexo VIII).

13.3.4.2 Métricas

A lo largo del proyecto se decidió construir métricas para los atributos de calidad más importantes definidos por el equipo. El primero y más relevante era la usabilidad, para lo que se definieron métricas de tres tipos:

- Velocidad de Uso
- Cantidad de errores
- “*Look and Feel*”.

El segundo atributo importante era la seguridad, para el cual se definieron métricas de

- Autenticación
- Autorización.

Las métricas relevadas a lo largo del proyecto fueron tomadas como resultado de las pruebas de sistema y las pruebas de validación. El equipo trabajó con los resultados de todas las pruebas, pero se priorizaron éstas debido a que no era posible atender todos los aspectos del sistema, dado el tamaño del equipo.

A continuación se listan las métricas definidas por el equipo (véase Anexo V):

13.3.4.2.1 Autenticación

Las medidas involucradas eran el número de pruebas fallidas. El conjunto de pruebas creadas para cada funcionalidad debía ser exitosamente ejecutado sin dar ningún resultado negativo. La autenticación era un punto muy importante en el sistema y como había sectores del sistema que permitían que el usuario navegara sin estar autenticado, en los lugares que era necesario estarlo, el sistema no podía fallar en restringir la entrada a usuarios anónimos.

13.3.4.2.2 Autorización

Las medidas involucradas eran el número de pruebas fallidas. Del conjunto de pruebas que se crearon para cada funcionalidad, debían ser exitosamente ejecutadas todas sin dar más de un 20% de resultados negativos.

13.3.4.2.3 Velocidad de Uso

Las medidas involucradas eran el tiempo que le toma a cada usuario realizar las tareas planificadas. Dado que no es simple medir el tiempo exacto que lleva a un usuario hacer una tarea, el valor más importante era el total. Se tomó como una prueba exitosa aquella que le permitía al usuario terminar en tiempo y forma todas las funcionalidades. Para profundizar en los datos, el encargado de *testing* anotaba cualquier dificultad fuera de lo común que presentase un usuario con alguna tarea.

En la tabla 7, se presentan las métricas para el usuario de perfil masculino.

Tarea	Tiempo (segundos)	Comentarios
Registro	120	Se registra correctamente
Iniciar Sesión	30	Inicia sesión correctamente
Olvido de contraseña	150	Logra restablecer su contraseña e iniciar sesión
Buscar artículo	25	Encuentra el artículo buscado
Encontrar donde lo venden	20	Identifica donde es que se vende dicho artículo
Ver artículos relacionados	20	Ingresa a los artículos relacionados
Agregar a artículos que querés para tu cumpleaños	30	Agrega un artículo a su lista de deseados para el cumpleaños
Mandar un comentario a LookeaMe	15	Realizar un Feedback correctamente

Tabla 7 - Métricas para el perfil masculino

En la tabla 8, se presentan las métricas para el usuario de perfil femenino.

Tarea	Tiempo (segundos)	Comentarios
Registro	120	Se registra correctamente
Iniciar Sesión	30	Inicia sesión correctamente
Te gusta un artículo	15	Agregar un artículo a la lista de artículos que le gustan
Querés un artículo para tu cumpleaños	30	Agrega un artículo a su lista de deseados para el cumpleaños
Seguir usuarios en LookeaMe	30	Empieza a seguir un usuario

Tarea	Tiempo (segundos)	Comentarios
Consultar tu closet	90	Ingresa al closet y revisa toda la información existente
Encontrar donde se vende una prenda que te gusta	20	Identifica donde es que se vende dicho artículo
Publicar en Facebook un artículo	40	Publica en Facebook y se asegura de que se publique correctamente

Tabla 8 - Métricas para el perfil femenino

13.3.4.2.4 Cantidad de errores

Esta métrica evalúa la cantidad y magnitud de los errores generados por el usuario al momento de realizar la prueba. El sistema, idealmente, debería evitar que los malos usos de las funcionalidades ocasionen problemas posteriormente. Es así que los máximos tenidos como referencia en esta métrica eran:

- 10 errores predecibles
- 1 error aceptable
- 0 error catastrófico

13.3.4.2.5 Look and Feel

Esta métrica era la más difícil de medir y una de las más importantes, dado que lo que buscaba es medir la opinión de los usuarios sobre Lookea.Me. Buscaba saber cómo se evaluaba el sitio, su diseño, lo intuitivo que era y si gustaba o no a los usuarios. Es una métrica que es sensible a defectos en el sistema y a dificultades para entender el sitio.

Cada usuario tenía, al final de su plan de pruebas, una encuesta con la tabla 9 que detallaba el método de evaluación sobre algunos puntos particulares.

Valor	Calificación
1	Muy malo – Muy difícil
2	No me convence – difícil
3	Está bien – Normal
4	Me gusta – Fácil
5	Excelente – Intuitivo

Tabla 9 - Valores para la evaluación del sistema

Los usuarios debían evaluar los diferentes puntos que se les presentaban y luego el equipo debía analizar esos valores. A los usuarios masculinos se les asignó prioridad diferente porque, sobre todo en diseño, las mujeres tenían más importancia para la toma de decisiones en Looke.Me. Por lo que al momento de analizar los resultados se tuvo en cuenta que una baja valoración por parte de un usuario femenino tenía mayor prioridad para ser resuelta que una baja valoración de un usuarios masculino.

Una vez calculado el total de puntaje obtenido se dividía en dos grupos de resultados. El primero era el grupo de las evaluaciones negativas que daban un total menor al 80%, el segundo grupo era el de las evaluaciones positivas que daban un total mayor o igual al 80%.

Con el primer grupo se realizaba un análisis más detallado intentando obtener los principales incidentes que ocasionaban ese bajo puntaje por parte de los usuarios. Posteriormente se analizaba con el equipo de modo de definir cuales medidas se podían tomar para solucionar esos incidentes rápidamente. Con el segundo grupo se analizaba que no hubiera valores bajos en algún punto en particular. Dado que el resultado global de este grupo es aceptable lo primordial fue trabajar aquellos puntos más débiles dentro de la evaluación.

13.3.4.3 Pruebas unitarias

Las pruebas unitarias fueron diseñadas sobre los métodos que implementan las funcionalidades básicas del sistema. De esta forma se buscó asegurar que a lo largo del proceso de desarrollo, cada vez que se realizaba un cambio que podía impactar sobre alguna de estas funcionalidades, el equipo contaba con un conjunto de pruebas que lograba asegurar su correcto funcionamiento.

Las pruebas eran ejecutadas al final de cada *sprint* cuando se agregaba una nueva funcionalidad. El responsable de ejecutar dichas pruebas era el que no la había desarrollado.

Para diseñar las pruebas se utilizó principalmente la estrategia de partición equivalente. Los casos de prueba se codificaron utilizando las soluciones de pruebas unitarias que el *framework* proporciona para poder luego ejecutar las pruebas las veces que fuera necesario, y mantener un control sobre las pruebas que se ejecutan.

13.3.4.4 Pruebas de integración

Las pruebas de integración se basaron en dos tipos: las pruebas exploratorias y las pruebas basadas en fallos.

Para realizar las pruebas basadas en fallos, el equipo desarrolló un componente orientado al registro de *Logs*, el cual permitía categorizarlos de la siguiente manera:

- Información
- Precaución
- Falla

Al menos una vez por día el encargado de *testing* debía hacerse responsable de que la sección de *Logs* fuera consultada y que se tomaran acciones sobre los diferentes incidentes que aparecían. No necesariamente eran resueltas, pero al menos debían ser asignadas para su posterior solución.

Las pruebas exploratorias que se realizaron son las denominadas de “estilo libre”. Eran procedimientos de pruebas informales donde el *tester* no debía llevar un reporte de ejecución, ni un reporte de defectos. Estas pruebas no tenían etapas de diseño y de ejecución porque era la informalidad una característica propia de las mismas. Consistían en navegar por la página revisando que todas las secciones de la misma estuvieran comportándose de acuerdo a lo esperado funcionalmente, que los estilos se respetaran según el manual de estilos del sitio (véase Anexo XIII), que los *links* funcionasen correctamente y que la página se comportase igual en todos los navegadores seleccionados para las pruebas.

Estas pruebas eran ejecutadas principalmente por el encargado de *testing* y al menos debían ser realizadas dos veces por cada *sprint* terminado.

13.3.4.5 Pruebas de sistema

Las pruebas de sistema buscaban verificar los requerimientos no funcionales y conocer más detalladamente las limitaciones del mismo, con el objetivo de mejorar aquellos puntos que se consideraban más importantes.

Para el diseño de las pruebas se consideraron los siguientes requisitos del sistema:

- Seguridad
- Escalabilidad
- Mantenibilidad

A su vez, las pruebas se dividieron en dos grupos: controlar las buenas prácticas de programación y probar la seguridad del sistema en toda su extensión.

Al comenzar el desarrollo del sistema el equipo decidió seguir un conjunto de buenas prácticas de programación que se conocen del *framework* utilizado (véase Anexo XVII), y se dedicó un tiempo considerable al diseño arquitectónico del sistema para que el mismo fuera escalable.

Por último, la seguridad se probó principalmente bajo las dos características más significativas para el alcance de este sistema: autenticación y autorización. Era responsabilidad del encargado de *testing* realizar pruebas sobre estos dos puntos cada vez que se terminaba un *sprint*, y de encontrarse algún incidente, se asignaba con prioridad alta.

13.3.4.6 Pruebas de validación

Las pruebas de validación eran las pruebas que el equipo consideró de más peso a lo largo del proceso de desarrollo. Dado que LookeaMe es un sistema que tiene un alto grado de innovación en aspectos como el negocio y el conjunto de funcionalidades que ofrece, se consideró que la validación por parte de los usuarios finales era clave para poder evaluar el producto como exitoso.

Dentro de todas las posibilidades de pruebas de validación se decidió realizar pruebas de usabilidad para poder obtener comentarios del sitio por parte de los usuarios que lo usarían.

13.3.4.6.1 Pruebas de usabilidad

Los objetivos buscados por medio de estas pruebas eran:

- Evaluar el nivel de aceptación y velocidad de aprendizaje de un grupo de usuarios representativos.
- Validar que las decisiones de diseño tomadas fueran compartidas por los usuarios finales
- Conseguir información suficiente por parte de los usuarios para poder conocer mejoras que se podían introducir en el sistema para llegar a satisfacer las necesidades de los mismos.

Los perfiles que se definieron para seleccionar los usuarios con los que se realizarían las pruebas fueron los siguientes:

- Mujeres entre 18 y 25 años que no eran económicamente independientes pero si con un poder adquisitivo medio, medio-alto o alto, y que gustaban de comprar online.
- Hombre entre 25 y 50 años de poder adquisitivo medio-alto o alto, que compraban ropa al menos una vez cada dos meses.
- Mujeres de familia de poder adquisitivo medio-alto o alto que realizaban las compras de productos de vestimenta para su familia y que usaban la computadora en el día a día.

Dentro de los perfiles más desafiantes en cuanto a la exigencia en los detalles de diseño, se identificaron los dos perfiles de mujeres, por lo que se decidió que se seleccionarían dos usuarios de dichos perfiles y un usuario del perfil restante.

Se preparó un documento para cada uno de los perfiles, y un documento que fue entregado a todos los usuarios explicando la motivación del sistema y sus funcionalidades más destacadas (véase Anexo VIII). Con este último documento, se buscaba que los diferentes usuarios que realizaban la prueba, se acercaran a ella con la misma información que podría tener un usuario luego de escuchar alguna campaña publicitaria sobre Looke.Me, no pretendiéndose explicar ningún punto en profundidad, debido a que se podía perder información de la prueba.

Se coordinó con los usuarios para realizar la prueba en los entornos más cotidianos posibles, dado que se intentaba evaluar la experiencia del usuario al utilizar el sistema en los ambientes en que lo utilizarían normalmente.

Para realizar esta prueba se utilizó la herramienta Zoom.us [48] (véase Anexo III) que le permite al encargado de *testing* hablar, escuchar y ver al usuario en todo momento. Además le permite ver lo que el usuario está viendo por medio de la funcionalidad “Compartir escritorio”, y por ultimo le permite grabar las sesiones para poder analizarlas posteriormente.

Utilizando dicha herramienta se coordinó la conferencia con los 5 usuarios al mismo tiempo y se les proporcionó los documentos para poder realizar la prueba. El encargado de *testing* cumplió el rol de motivador en todo momento respondiendo consultas pero aportó la menor cantidad de datos posibles, dado que un usuario normalmente no tendría una fuente de información en todo momento.

13.3.4.7 Resultados de las pruebas

Los errores detectados durante el proyecto se reportaron y se les realizó el seguimiento a través de la herramienta Mantis [49] (Figura 37).

En la figura 36 se muestran los estados que se utilizaron para el seguimiento de los mismos.



Figura 36 - Estados para el seguimiento de incidentes

Los incidentes se ingresaban como activos y se asignaban a un desarrollador. El desarrollador realizaba las correcciones pertinentes y una vez que finalizaba, cambiaba el estado del incidente para resuelto y se lo asignaba al otro integrante del equipo. Al final, el otro integrante del equipo verificaba que el incidente hubiera sido solucionado correctamente y cambiaba su estado a cerrado.

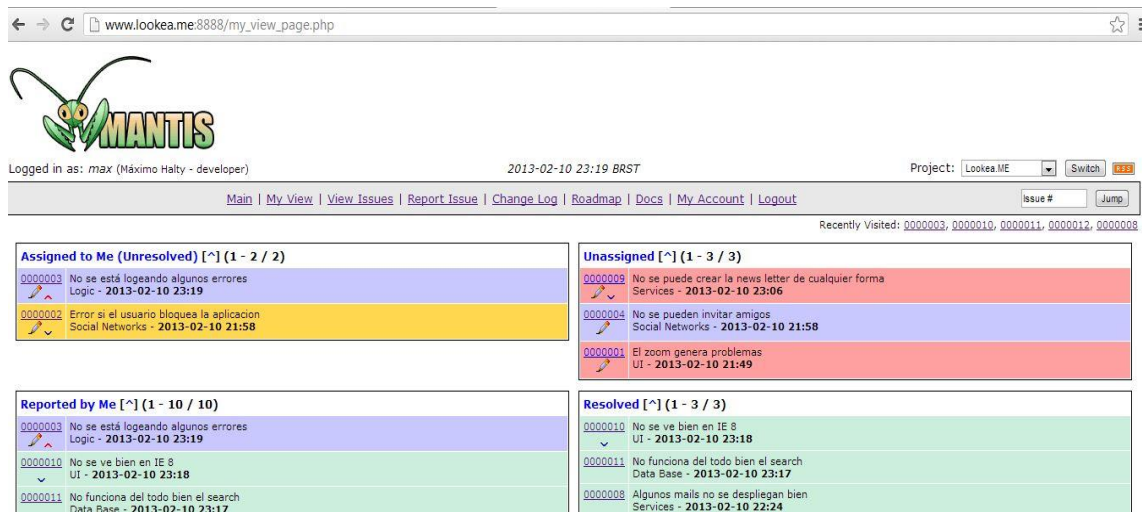


Figura 37 – Mantis

La mayoría de las pruebas se ejecutaron cada vez que se daba por terminado un *sprint*. Se estimaba dentro del proyecto un tiempo de re trabajo que implicaba la corrección de los incidentes encontrados y si ese tiempo no alcanzaba dado el tamaño del problema encontrado, se agregaba la solución del mismo al comienzo del *sprint* que seguía.

Con esta metodología el equipo se aseguró de que todos los incidentes identificados en las pruebas fueron resueltos. Esto no tomaba en cuenta las encuestas y defectos recolectados en las pruebas de usabilidad dado que éstas fueron ejecutadas al final de proceso de desarrollo y por otro lado, porque no era posible estimar el tiempo suficiente

para solucionar todos los problemas. Se solucionaron los más urgentes, los que ocasionarían más daño al sistema si este se pudiese en producción, y se documentaron los demás para trabajar sobre ellos antes de poder sacar una versión estable.

En la tabla 10 se muestran los resultados que se obtuvieron al ejecutar las pruebas al final de cada *sprint*. Algunas pruebas no retornan resultados que se puedan mostrar en tablas, por lo que se verán únicamente los resultados que aportan más información al proyecto.

	Unitarias	Exploratorias	Basadas en fallos	De autenticación	De autorización	
Sprint 1	7	5	0	8	1	21
Sprint 2	10	8	2	0	0	20
Sprint 3	6	7	6	3	0	22
Sprint 4	1	9	5	0	0	15

Tabla 10 - Tablas de resultados

Se puede observar que las funcionalidades relacionadas a la autorización fueron afectadas únicamente en el primer *sprint*, momento en el que fueron creadas. Las relacionadas a la autenticación fueron afectadas en el primer y tercer *sprint*. En el primer *sprint* fueron creadas las funcionalidades de *login* y registrarse, y en el tercer *sprint* fueron implementadas las funcionalidades de integración con las redes sociales, que afectaron la autenticación. Por lo tanto, era esperable el número de defectos que se encontraron en ambas pruebas.

En las pruebas basadas en fallos, se puede observar que fue aumentando la cantidad de defectos encontrados. Esto se puede asociar con el hecho de que el sistema fue creciendo y a medida que crecía, había más funcionalidades que ingresaban *logs* al sistema al momento de fallar. Cuantas más posibilidades de que se produjera una falla, era entendible que creciera la cantidad de fallas encontradas.

Las pruebas de usabilidad retornaron resultados dentro del margen de lo esperado definido anteriormente en las métricas para estas pruebas. Desde esta perspectiva, se esperaba tener mejores resultados pero los obtenidos no se desvían mayoritariamente de las métricas definidas.

El primer resultado a analizar de las pruebas de usabilidad era el relacionado a la “velocidad de uso”. No todos los usuarios lograron terminar las pruebas en el tiempo estipulado de 10 minutos. Los dos usuarios femeninos que eran madres de familia no lograron terminar todas las funcionalidades propuestas. A una de estas usuarias le llevó más de lo esperado realizar el registro, y su encuesta mostraba que una de sus críticas

era la gran cantidad de campos que se solicitaban. La otra usuaria no logró ejecutar la funcionalidad de agregar un artículo a su lista de cumpleaños. No resultaba intuitivo lo que se buscaba hacer, por lo que le costó comprender el objetivo. Para esto se consideró importante poder nutrir el sistema de guías sobre cómo utilizar las diferentes funcionalidades del mismo.

El segundo resultado a analizar es el de la “cantidad de errores”. El equipo intentó evaluar por qué se obtuvieron más errores de los esperados, cuáles de ellos era posible resolver, y cuáles quedarían pendientes para futuras versiones.

La tabla 11 muestra la cantidad de errores totales de la siguiente manera:

	Errores Predecibles	Errores Aceptables	Errores Catastróficos
Sofía Arrambide	7	1	0
Mercedes Laffitte	12	2	0
Juan Francisco Mackinnon	9	0	0
Florencia Lariau	20	0	1
Ximena Montes	19	1	0
Promedio	13	1	0

Tabla 11 - Cantidad de errores

Los valores obtenidos como promedio no estuvieron tan lejos de los límites esperados. La única cantidad de errores que fue mayor fueron los errores predecibles, ya que se esperaba una cantidad máxima de 10 y el promedio fue 13. Se considera que la mejor forma de solucionar estos incidentes es que en el futuro la página tenga más tutoriales.

Los errores aceptables se mantuvieron dentro del margen de lo esperado, sobre todo teniendo en cuenta que uno de los errores fue obtenido por cada uno de los usuarios. El problema fue que el botón de registro no funcionaba bien, aunque el equipo había verificado que dicha funcionalidad se comportaba correctamente, al momento de realizar las pruebas el botón no funcionaba. El error fue identificado y solucionado inmediatamente. Pero las encuestas se vieron marcadas por este hecho.

Errores catastróficos solo ocurrió uno y a uno de los usuarios, luego de intentar realizar el registro sin éxito, al cerrar la ventana *popup* el sistema se quedó congelado. Tuvo que ingresar a la página nuevamente para poder seguir con las pruebas. No se pudo reproducir dicho error y se asume que podría ser por un problema con el equipo del usuario dado que no le sucedió a ningún otro usuario.

Los resultados de las encuestas de satisfacción más relevantes fueron los siguientes:

Usuarios femeninos

Dos de las encuestas que aportaron más información fueron las que detalla la tabla 12.

Diseño y apariencia de:		
Lookea.Me en su totalidad	4	5
Detalle de los productos	3	3
Mi Closet	4	4
Dificultad de:		
Registro	1	1
Inicio de Sesión	5	3
Publicar en Facebook	5	5
Encontrar la información de los artículos	4	4
Poner Me gusta a un artículo	5	5
Poner Lo quiero a un artículo en una lista de deseados	5	2

Tabla 12 - Resultados de encuestas a usuarios femeninos

Comentarios:

“¡Me encanto la página! es muy divertida como se les ocurrió??? Es algo re original”

“Problemas q encontré:

- botón registro no se podía hacer click
- no puedo volver desde mi closet
- si estoy adentro de un producto y quiero buscar otro tema de ropa no me deja salir de la página donde estaba tuve que clickear sobre cartel de LOOKEAME como cuando estaba en closet”

Usuario masculino

Diseño y apariencia de:

- LookeaMe en su totalidad: 4
- Detalle de los productos: 2

Comentarios:

“No me convence, faltan más datos en mi opinión como material por ejemplo, talles disponibles no sé...etc.”

Dificultad de:

- Registro: si anduviera el botón registro te doy un 5
- Inicio de Sesión: No lo use
- Búsqueda: 5
- Encontrar la información de los artículos: 4
- Poner Lo quiero a un artículo en una lista de deseados: 3

El total de la evaluación retornada por el usuario masculino es de un 77% de satisfacción con el sistema según las pruebas que se le solicitaron realizar. Por otro lado las otros dos resultados, asociados a las usuarias femeninas, dieron 80% y 71% respectivamente.

Como se mencionó en la sección de métricas (véase Capítulo 11.3.4.2) estos resultados se agrupan en dos, por un lado los resultados que dieron un total menor a 80% que fueron analizados por el equipo de desarrollo para evaluar qué acciones se podían tomar para solucionar los puntos más débiles en la experiencia de estos usuarios con Lookea.Me y por otro lado el resultado que dio igual a 80% que fue analizado para ver si se plasmaba un punto débil que fuese importante tener en cuenta.

Dentro de los errores y sugerencias encontradas se pudo ver que el “detalle del producto” es una funcionalidad a diseñar mejor y que el registro tenía algunos problemas que fueron posteriormente solucionados. Con estos dos puntos trabajados el equipo pudo ver que las razones que afectaron a los resultados de la prueba más negativamente fueron solucionadas y que además se obtuvieron sugerencias para futuras versiones de Lookea.Me.

14 Gerencia

En el presente capítulo se presentarán los aspectos de gestión utilizados a lo largo del proyecto: como fueron definidas las estrategias, planes, seguimientos y correcciones que se debieron hacer para lograr los objetivos. Para las distintas áreas se definieron procesos, donde a su vez se utilizaron herramientas de soporte y se realizaba el seguimiento correspondiente.

14.1 Objetivos

El principal objetivo del área es planificar y controlar la ejecución de los procesos definidos, de forma de asegurar el correcto desarrollo del proyecto. Según PMBOK [17], la gestión de proyectos es: “la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto con el objetivo de satisfacer o superar las necesidades y expectativas de los inversores del proyecto.”

De esta forma, todas las actividades desarrolladas durante el transcurso del proyecto formaron parte de uno o varios procesos definidos, los cuales a su vez fueron planificados y controlados para asegurar su correcta ejecución. Es importante resaltar en esta instancia la metodología de mejora continua propuesta por Edwards Deming (citada anteriormente en capítulo Gestión de Calidad), estableciendo 4 actividades claves para asegurar la calidad en un proyecto, ya que la misma fue utilizada durante el transcurso del mismo.

14.2 Modelo de ciclo de vida y ciclo de vida

Considerando la magnitud y naturaleza del proyecto en conjunto con las características del grupo, la definición de la metodología de trabajo a utilizar no sería una sola, sino la conjunción de varias de ellas. De esta forma, las metodologías utilizadas debían adecuarse de forma de aprovechar las ventajas y mitigar las desventajas tanto del proyecto como del grupo.

El proyecto constó de un producto cuyo alcance previsto era superior al alcance del proyecto, en contexto de un área de negocio desconocida para el equipo, requerimientos con prioridades inestables y un importante foco en el diseño gráfico. El cliente estaba muy cercano y disponible, y en algunos casos, dentro del equipo de trabajo. Sin embargo, la inexperiencia en proyectos tecnológicos por parte del cliente tomó un rol importante, ya que desconocían aspectos claves como el tiempo, costo y alcance para proyectos de estas características.

El grupo de proyecto estaba integrado por 2 personas altamente cohesivas, que además trabajaron juntos durante toda la carrera. Ambos integrantes tenían trabajos de entre 7 y 9 horas cada uno, limitando el trabajo en el transcurso de la semana, pero haciendo

énfasis en los sábados y domingos. A su vez, ambos integrantes tenían la experiencia y el conocimiento necesario para desarrollar el producto en la tecnología anteriormente propuesta: Microsoft .Net, C#.

Según las características del proyecto mencionadas, el proyecto tuvo un desarrollo riesgoso a causa del área de negocio desconocida, recursos escasos ya que era un grupo de 2 personas y un foco especial en calidad a causa de la importancia de la experiencia del usuario con el diseño gráfico de la aplicación. A su vez, el proyecto en sí mismo requería cierta documentación estándar que no podía faltar por buenas prácticas y requisitos académicos

En consecuencia, a los efectos de las grandes definiciones a nivel de proceso, la aplicación de los paradigmas en cascada y R.A.D. (Rapid Application Development) no eran aplicables al presente proyecto debido a su poca adaptación a los cambios, generando riesgos adicionales e innecesarios. Por su lado, la construcción de prototipos fue descartada debido a la naturaleza del negocio, ya que los requerimientos fueron extraídos en conjunto con el grupo comercial de Looke.Me con estrategias comerciales previamente definidas.

De esta manera, el paradigma elegido tanto al proyecto como al grupo, fue el paradigma en espiral. Este paradigma está orientado a riesgos, optimizando los recursos y acelerando el proceso de desarrollo. En cambio, tiende a producir requerimientos y arquitectura inestables, donde además requiere mayores costos de integración y planificación. Estos aspectos fueron considerados en el balance de selección del paradigma e incluidos en la gestión de riesgos del proyecto.

A causa de los problemas que ocurrieron a partir del paradigma en espiral, se optó por basarse también en el meta-paradigma R.U.P. (Rational Unified Process). Si bien las ventajas y desventajas se asemejan al paradigma en espiral, R.U.P. permitió definir y ejecutar en Looke.Me un proceso centrado en la arquitectura, donde se establecieron refinamientos sucesivos de una arquitectura ejecutable, construida como un prototipo evolutivo.

Si bien el paradigma en espiral tiende a generar requerimientos inestables, esta conjunción permitió la estabilidad de los mismos. Como fue mencionado en capítulos anteriores, el proyecto requirió establecer una única y completa etapa de Ingeniería de Requerimientos, a causa de la poca disponibilidad de las tiendas de moda.

Los problemas relacionados a la arquitectura fueron mitigados bajo un único diseño arquitectónico de alto nivel, para luego iterar sobre un diseño más detallado y de bajo nivel. Este último aspecto colaboró con la mitigación de los costos de integración, ya que el diseño a alto nivel permitió establecer una base sólida sobre la cual trabajar. En cuanto a los costos de planificación, el mismo formó parte de los principales riesgos del proyecto.

En conjunto con los paradigmas anteriormente adoptados en el proyecto, se incluyeron aspectos basados en paradigmas de metodologías ágiles. Básicamente aquellos aspectos dinámicos relacionados al trabajo en grupo que permitió optimizar los recursos y obtener provecho de la cercanía de los clientes, así como también proveyó un conjunto de buenas prácticas para trabajar colaborativamente y en equipo, con lo que se logró el mejor resultado posible para este proyecto.

En base al análisis realizado, el ciclo de vida instanciado fue el incremental tal como lo establece el capítulo 7.5.5, dado que tanto el ciclo de vida en cascada como el evolutivo no se adaptaban a las características del proyecto y grupo. De esta forma, se instanció el ciclo de vida incremental que se detalla en la figura 38:

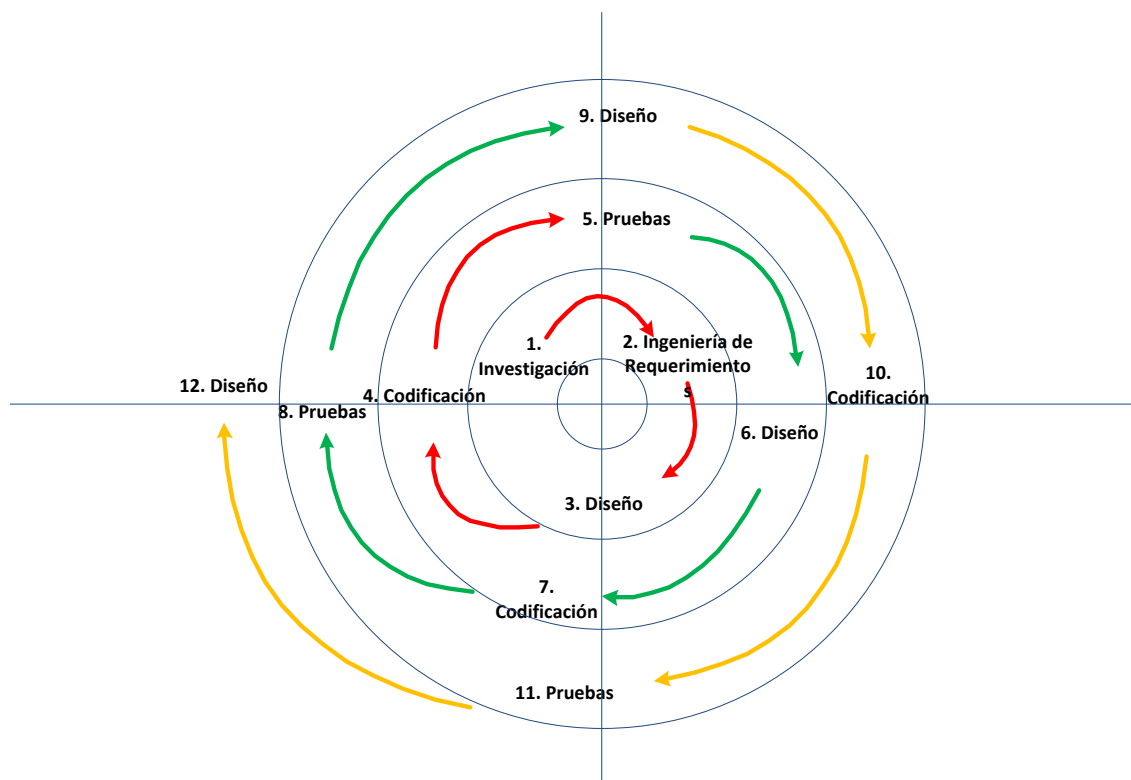


Figura 38 - Instancia del ciclo de vida incremental

En una primera instancia se realizaron la investigación y la ingeniería de requerimientos, para luego iterar sobre el diseño, codificación y pruebas. Se puede observar el ciclo de vida incremental instanciado, basándose en una metodología en espiral.

Por su parte, el modelo R.U.P. permitió establecer que en cada iteración se enfocaba en entregables con agregación de valor al cliente. Esto no significó que al término de cada iteración necesariamente se entregara un producto de software terminado, sino que el valor agregado podía ser un documento, un resultado o un prototipo. A su vez, R.U.P. permitió aplicar una metodología basada en los “*workflows*” o “*disciplinas*”, para que luego los “*workers*” o “*trabajadores*” del equipo pudieran realizar las tareas necesarias

para obtener el o los “artefactos” establecidos, donde cada artefacto era el resultado parcial o final de cada iteración.

Al ser R.U.P. un proceso centrado en arquitectura y dirigido por los casos de uso, al ser combinado con un ciclo de vida incremental se obtuvieron los datos necesarios para realizar un documento de requerimientos completo al comienzo del proyecto. Por este motivo la arquitectura fue realizada en los comienzos del desarrollo, estableciendo de esta manera los distintos componentes de cada sistema y sus respectivas interacciones. Lo mismo ocurrió con la base de datos, ya que de la misma forma que la arquitectura, se tenían los datos necesarios para realizar el M.E.R. completo antes de comenzar el desarrollo en sí mismo.

Finalmente, si bien las metodologías ágiles no fueron plasmadas en el diagrama del ciclo de vida, estas últimas formaron parte de la planificación de las iteraciones y la metodología de trabajo utilizada en el proceso de desarrollo.

14.3 Gestión del alcance, tiempo y costo

La gestión del alcance fue uno de los aspectos más importantes del proyecto. Esto se debe a que el alcance del producto en sí mismo era mayor que el alcance del proyecto, debiendo adaptar el alcance del producto al del proyecto.

Si bien los requerimientos eran estables de acuerdo al proceso de Ingeniería de Requerimientos realizado, no así la prioridad asociada a cada uno de ellos. En varias ocasiones los clientes desearon cambiar los requerimientos a desarrollar en el proyecto, modificando de esta manera el alcance del mismo, y por ende, impactando directamente sobre el tiempo, el costo y la calidad del proyecto.

Para abordar esta problemática, se tomó ventaja que el cliente estaba cerca del grupo, estableciendo el alcance del proyecto por etapas. En la primera etapa del proyecto, donde se realizó la investigación y la Ingeniería de Requerimientos, el alcance fue limitado por la fecha de la primera revisión. De esta manera, se estableció realizar una investigación que abarcara todos los aspectos del negocio a ser estudiados, independientemente si los requerimientos posteriormente extraídos fueran o no desarrollados en el transcurso del proyecto. El desafío de acotar el alcance surgió en la segunda etapa del proyecto, en la cual se definieron los requerimientos a ser desarrollados para luego comenzar con el desarrollo del software.

A su vez, es importante resaltar el efecto que tuvo el costo, el tiempo y la calidad en el presente proyecto y cómo estos afectaron el alcance del mismo, sobre todo en la segunda etapa.

Dentro de los costos del proyecto se encontraron los costos asociados a software y/o hardware adquirido, servicios en Internet y el diseño gráfico de la página Web. No fue necesario adquirir nuevos programas de software debido a que cada integrante contaba con el software necesario para desarrollar el sistema. Por su parte, se centralizó el desarrollo y manejo de versiones en un equipo ajeno a los equipos de cada integrante del grupo, adecuándose una PC antigua de uno de los integrantes, y adquiriendo algunas piezas para lograr un correcto funcionamiento.

El costo anteriormente mencionado no influyó de ninguna forma en el desarrollo del proyecto, por lo que no impactó en el alcance, tiempo y/ calidad del mismo. Pero no así ocurrió con el diseño gráfico, que se transformó en un costo a ser gestionado, no solo por el dinero, sino también porque el mismo fue tercerizado. De esta forma, se debía gestionar dicho esquema de trabajo para que no impactara negativamente en el proyecto.

El diseño gráfico produjo un impacto en la planificación importante pues se debía incluir el diseño como ítem a ser planificado. A su vez, el diseño realizado por la diseñadora fue realizado en formato HTML puro. Por este motivo, dicho diseño debía ser transformado al formato de la tecnología utilizada por el grupo, la cual en este caso fue ASP.NET MVC. Si bien la inclusión del diseño gráfico en la planificación no fue un grave problema para el grupo, tuvo impacto sobre alcance del producto pues el diseño gráfico se convirtió en un nuevo requerimiento a ser desarrollado.

En cuanto al tiempo asociado al proyecto, el mismo también era acotado. Como se mencionó anteriormente, la primera etapa tuvo fecha de fin 3 de Julio de 2012, siendo esta la fecha de la primera revisión. Por este motivo, la investigación y la Ingeniería de Requerimientos debía terminar en dicha fecha. En cuanto a la segunda etapa, la fecha de finalización también fue fija, estableciendo fecha de publicación pública de la Web a fines de Diciembre 2012/principios de Enero 2013.

Cada sistema a desarrollar tenía sus características de calidad asociadas que no necesariamente eran las mismas para cada sistema. De esta manera, el foco de calidad de cada uno era independiente de los restantes, por lo que la gestión de la calidad debía ser gestionada de forma independiente.

En cuanto a la calidad del sistema de usuarios, el atributo de calidad más importante eran la experiencia del usuario, la usabilidad y el diseño gráfico. Se hizo especial énfasis en el diseño gráfico, lo cual derivó en que cada requerimiento a desarrollar fuera más complejo que simplemente desarrollar el requerimiento en sí mismo. Por su parte, el sistema de administración por parte de las tiendas no tenía foco en la experiencia del usuario, sino en el manejo de los artículos a publicar. De esta forma, se hizo especial hincapié en el proceso de publicación de artículos, el cual debía proveer una forma fácil y rápida para poder manipular todos los datos de los artículos. Finalmente, el sistema de administradores no debía tener un “*look and feel*” interesante ni ser rápido en el manejo de los datos, sino que debía incluir todas las herramientas posibles para lograr ofrecer un servicio de soporte adecuado tanto para los usuarios como para las tiendas de moda.

De forma conjunta, la segunda etapa del proyecto debía terminar a fines de Diciembre 2012, junto con el diseño gráfico realizado y transformado en la tecnología utilizada, y cumpliendo con las pautas de calidad establecidas anteriormente. En consecuencia, el alcance se vio reducido en cantidad de requerimientos a ser desarrollados dentro de los plazos del proyecto.

El alcance del proyecto estuvo dado por el equilibrio entre los costos, el tiempo y la calidad asociados al mismo (Figura 39).

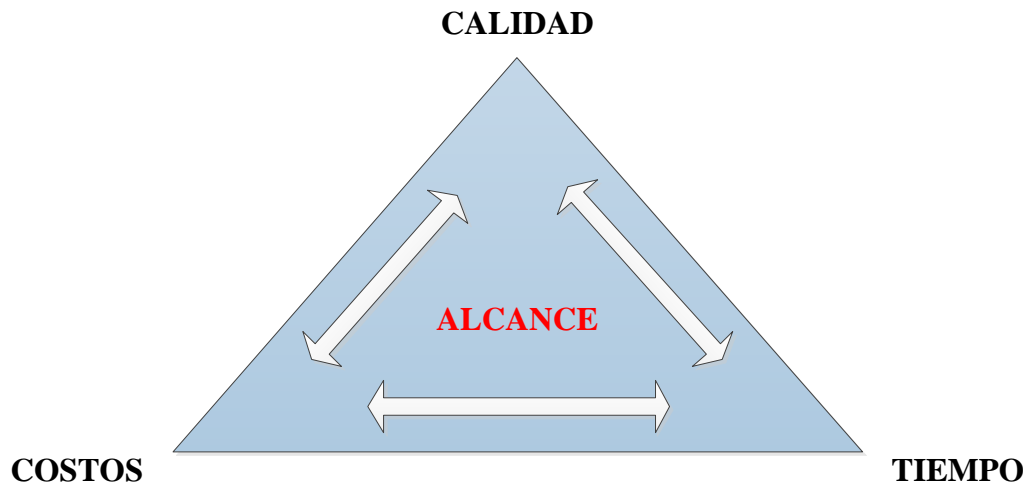


Figura 39 - Costo, tiempo y calidad

La tercer y última etapa del proyecto tuvo fecha de finalización 7 de Marzo 2013, estableciendo entregas intermedias de la documentación final del proyecto en Febrero 2013. A su vez, el alcance de dicha documentación fue establecido de forma conjunta con el tutor, estableciendo claramente los capítulos que debían integrar el documento final.

14.4 Cronograma

14.4.1 Primeras etapas

En las primeras etapas del proyecto, el grupo se puso como objetivo realizar un análisis detallado del negocio para luego realizar la ingeniería de requerimientos. A su vez, se realizaron procesos de planificación y estimativos de alcance para estimar la fecha de comienzo de desarrollo del producto, documentando los aspectos necesarios del proyecto.

Las tareas realizadas fueron:

- Investigación del negocio
- Estimación de alcance preliminar
- Ingeniería de requerimientos
- Documentación acorde a las tareas realizadas

Durante el transcurso de estas primeras etapas, el tutor Ing. Bruno Delgado propuso realizar una primera revisión formal del proyecto con el Dr. Pedro Salvetto. De esta forma, se puso como objetivo culminar las primeras etapas del proyecto para dicha presentación el día 3 de Julio, 2012. Cabe resaltar que a esta altura del proyecto ya se habían definido las estrategias más importantes de gerencia, SCM y SQA, incluyendo el ciclo de vida y las metodologías utilizadas.

14.4.2 Planificación

Una vez realizada la revisión, y habiendo realizado las correcciones correspondientes, se propone una primera planificación del proyecto.

Tarea	Descripción	Principales Requerimientos	Fecha de Finalización
ESRE	Documento de Especificación de Requerimientos	-	30-jun-12
Diseño Inicial	Diseño de arquitectura, MER y diseño alto nivel	-	18-jul-12
Sprint 1	Principales de User y backoffice para tiendas	Registro, Login, Búsqueda de Artículos	10-sep-12
Sprint 2	Publicación masiva por tiendas y aplicación red social	Redes sociales, wish list, mi colección	25-oct-12
Sprint 3	Reserva y compra de artículos. Ambas aplicaciones	Reserva, compra y reviews	10-dic-12
Puesta en producción	Subir la aplicación pública desde el servidor	Todas	10-dic-12
Pruebas de usuarios	Validaciones por parte de usuario + Testing	Todas	14-feb-13
Documentación Final	-	-	7-mar-13

Figura 40 - Primera planificación del proyecto

En base a las principales tareas propuestas en la figura 40, se decidió dividir el proyecto en 4 etapas:

1. Primeras etapas: Investigación, requerimientos y diseño arquitectónico inicial.
2. Etapa de desarrollo: *Sprints*.
3. Puesta en producción y prueba final con usuarios.
4. Preparación del documento final para la presentación de la tesis.

En esta primera planificación resalta la división por *Sprints*. Si bien más adelante se detallará la metodología utilizada para dicha división, cada *sprint* estuvo compuesto de 3 tareas claves: diseño, codificación y pruebas. De esta forma, se podía ver claramente cómo se instanciaba el ciclo de vida incremental, estableciendo 3 iteraciones sobre el diseño, la codificación y las pruebas, habiendo realizado en primera instancia la Ingeniería de Requerimientos.

Finalmente, existe una tercera y cuarta etapa donde se realizaron las pruebas finales del proyecto, junto con el cierre del mismo y su documentación asociada destacando asimismo la importancia de la primera revisión de la tesis, ya que esta fue tomada como preparación para la defensa final. En consecuencia, fue preparada y ejecutada en forma de presentación final, debiendo terminar la primera etapa el día 30 de Junio del 2012 para luego preparar y realizar la revisión el día 3 de Julio. Finalmente se estableció una segunda y última revisión formal a fines de Noviembre 2012 utilizando la misma metodología que la primera revisión.

14.5 Proceso de desarrollo

14.5.1 Metodología SCRUM

Durante el transcurso del proceso de desarrollo, las tareas, las estimaciones y división de tareas fueron realizadas en base a la metodología SCRUM. Ésta es una metodología ágil que provee un conjunto de buenas prácticas que son fuertemente aplicadas a proyectos con las siguientes características:

- El equipo debe estar 100% comprometido con el proyecto.
- Se basa en la experiencia del equipo.
- Los requisitos son cambiantes.
- El cliente y/o empresa desea hacer todo lo posible para que el proyecto prospere.
- El cliente está cerca del equipo. Si está adentro, mejor.

A su vez, esta metodología no es recomendable usarla cuando:

- El entorno no es flexible.
- El o los grupos de desarrollo tienen más de 10 integrantes.
- La temática Costos es muy importante para el proyecto.

Las características propuestas por SCRUM se adecuaban perfectamente con las características del proyecto, salvo por la importancia de los costos del mismo. Si bien el proyecto no tenía requerimientos inestables, si lo eran sus respectivas prioridades, por lo que según SCRUM, los mismos se consideraron como cambiantes.

Ser un equipo reducido pudo ser un riesgo negativo y afectar en gran parte el desarrollo del proyecto. Sin embargo, con SCRUM se pudo obtener provecho del tamaño del grupo obteniendo los mejores resultados basándose en el *expertise* de cada integrante. Además, SCRUM recomienda realizar *sprints* de corta duración, donde el resultado del mismo debe ser un producto funcional. De esta manera, con iteraciones cortas se lograron productos o prototipos funcionales que pudieron ser validados y verificados,

mitigando los problemas originados por la inestabilidad del negocio y los requerimientos.

Hubo otros aspectos de la metodología SCRUM muy relacionados con el éxito de Looke.Me:

- Facilidad de ajustar el producto de SW a las necesidades del negocio.
- Visualización del proyecto día a día.
- Facilidad de responder al factor imprevisible de los requerimientos

Si bien SCRUM es un solucionador de problemas en muchos casos, esto también pudo haber traído ciertas desventajas, siendo estas mitigadas de la siguiente manera:

- Los *sprints* no contemplan todas las actividades necesarias para el desarrollo, las cuales fueron agregadas dentro de la propuesta de cada *sprint*.
- El proceso no considera documentación estándar, la cual puede no satisfacer las necesidades del cliente, por lo que se realizó documentación estándar en etapas anteriores al desarrollo.
- Si bien establece un conjunto de buenas prácticas, muchas veces se debieron aplicar otras metodologías para adaptarse 100% a las características del proyecto, tal cual lo indicado anteriormente.

Es por esta última razón por la cual el equipo utilizó a la metodología como guía y soporte, basándose en metodologías tradicionales tal como R.U.P. y Espiral. Se logró un equilibrio entre las 3 metodologías establecidas obteniendo las ventajas de cada una y equilibrando las desventajas.

14.5.2 División de tareas

Siendo un grupo de 2 personas, la asignación de tareas no fue un proceso que precisó de un análisis detallado. Aun así existieron ciertos criterios de definición, no solo con los requerimientos a desarrollar, sino también definición de responsables por áreas realizado en los inicios del proyecto.

La decisión del área responsable de cada integrante se basó en el *expertise* de cada uno, la confianza y afinidad con el área.

La división por área se muestra en la tabla 13.

Responsable	Área del Proyecto
Agustín Alonso	Gerencia
Agustín Alonso	Ingeniería de Requerimientos
Agustín Alonso	Desarrollo
Máximo Halty	Arquitectura y Diseño
Máximo Halty	SCM
Máximo Halty	Pruebas
Ambos integrantes	SQA

Tabla 13 - Responsable de cada área

A su vez, la división de tareas por requerimiento se dividió principalmente por sistema de software a desarrollar. Como se mencionó en otros capítulos, la solución de Lookea.Me cuenta con los siguientes componentes de software:

- Sitio Web para los usuarios.
- Sitio Web para los administradores del sitio.
- Sitio Web para las tiendas.
- Aplicación de escritorio para la carga masiva de artículos por parte de las tiendas.
- Programas de soporte tales como envíos de mail y procesos batch.

De esta manera se identificó un responsable para cada sistema a ser desarrollado, de forma que existía un referente para cada uno de ellos y las principales decisiones y cambios estaban centralizados en una persona. La magnitud de los sistemas no se distribuyó uniformemente, siendo el sitio Web de tiendas y usuarios de mayor alcance que los demás.

La división de los sistemas en cuanto a la responsabilidad de su realización está representada en la tabla 14.

Responsable	Sistema
Agustín Alonso	Aplicación de escritorio para tiendas.
Agustín Alonso	Servicios de soporte.
Agustín Alonso	Sitio Web de administradores.
Máximo Halty	Sitio Web de usuarios.
Máximo Halty	Sitio Web de tiendas.

Tabla 14 - Divisiones para el desarrollo del sistema

Es importante resaltar la aproximación de la división de requerimientos establecidos en la primera planificación del sistema. Dicha división se estableció de forma de cumplir con lo establecido por las metodologías SCRUM, R.U.P. y Espiral, realizando una arquitectura base en la cual trabajar. A su vez, los requerimientos a realizar en cada *sprint* fueron planificados acorde a las precedencias existentes entre los requerimientos.

Como ejemplo: el requerimiento “Comprar un artículo” no era desarrollable en el primer *sprint*, pues para que se pudiera realizar el mismo, existían otros previos que deben estar desarrollados, tales como el “Registro” y “Login” del usuario.

Cada iteración fue planificada en base al objetivo de entregar un producto de SW funcional al terminar cada iteración, cumpliendo de esta manera con lo establecido en las metodologías anteriormente propuestas.

La primera aproximación de requerimientos por *sprints* está representada en la figura 41.

Funcionalidades x Responsable				
Responsable	1° Spt	2° Spt	3° Spt	Total
Agustín Alonso	6	7	9	22
Máximo Halty	4	7	7	18
Total Grupo	10	14	16	40

Funcionalidades x Aplicación				
Responsable	1° Spt	2° Spt	3° Spt	Total
Admin	0	4	0	4
Stores	3	2	8	13
Users	7	8	8	23
Total Grupo	10	14	16	40

Figura 41 - Primera planificación de requerimientos por sprint

14.5.3 Principales tareas

14.5.3.1 Reunión semanal

Si bien SCRUM establece una reunión diaria a la cual llama “Daily SCRUM”, no era aplicable de forma directa al proyecto. Esto se debe a que las reuniones diarias propuestas suponían que los integrantes del equipo trabajan todos los días en el proyecto de forma regular y básicamente, en forma presencial. No fue así en el presente proyecto, donde la concentración de horas se daba los fines de semana, distribuyendo algunas horas durante el transcurso de la misma.

En otro ejemplo de adaptación metodológica exitosa, se sustituyó la reunión diaria tal cual propuesta por SCRUM, por una reunión semanal un día del fin de semana. Esto en general ocurría los sábados por la mañana, donde el grupo proyectaba las funcionalidades restantes, posibles dudas y problemas que hubieran surgido en la semana.

Si bien esta reunión no tenía como propósito ser una reunión de avance, durante el transcurso del proyecto surgieron problemas con la planificación que fueron emergiendo en dichas reuniones. Es por esto último, que la reunión servía para planificar la semana y establecer un alcance de objetivos en base a los pendientes y a la planificación propuesta.

14.5.3.2 Avance de tareas

Por lo general, la reunión semanal ocurría los sábados por la mañana. Aun así, al terminar el fin de semana, los domingos por la tarde se realizaba una puesta a punto de la situación actual y los pendientes, estableciéndose un avance de las tareas y una proyección semanal para el sábado por la mañana. Esta pequeña puesta a punto se establecía con el soporte de documentos formalizados.

14.5.3.3 Registro de esfuerzo

De forma periódica cada integrante debía registrar las horas realizadas contándose con la información actualizada del esfuerzo realizado por cada integrante y un avance total en base al esfuerzo estimado (Figura 42).

En el registro debía especificar:

- Fecha
- Esfuerzo (en horas)

- Responsable
- Tarea
- Descripción

Fecha	Horas	Responsable	Tarea	Descripción
26-oct-12	3	Agustín Alonso	Codificación (Sprt 3)	Comencé a migrar el diseño al sistema de LME de us
28-oct-12	2	Agustín Alonso	Codificación (Sprt 3)	Arreglé algunos detalles del diseño
29-oct-12	5	Maximo Halty	Codificación (Sprt 3)	Solucioné un problema del update de imágenes de
13-oct-12	5	Agustín Alonso	Diseño (Sprt 2)	Diseño de aplicación de Desktop. No mas con excel
14-oct-12	4	Agustín Alonso	Diseño (Sprt 2)	Diseño de aplicación de Desktop. No mas con excel
21-oct-12	2	Agustín Alonso	Diseño (Sprt 2)	Diseño de datos en aplicación desktop para cambio:

Figura 42 – Registro de esfuerzo por Excel

14.5.3.4 PSP Semanal

Según lo establece PSP, soportado por la planificación en *sprints* de SCRUM, cada uno de los integrantes mantuvo un registro de las planificaciones semanales en base a la metodología PROBE (véase Anexo IX). Para la misma se utilizó como parámetro de planificación la cantidad de clases que los requerimientos requerían para ser implementadas, basado en información histórica de cada integrante.

Esta planificación era individual e independiente de la planificación general del proyecto, siendo la primera actividad del proceso de desarrollo propuesto por PSP. Como base para el aseguramiento de la calidad, la utilización de PSP implicó que cada integrante establecía un proceso de desarrollo individual e independiente entre ellos, medible y controlable, de forma de mejorar en cada iteración.

Como recomienda esta práctica, se estableció un seguimiento de los defectos inyectados en el producto, siendo estos corregidos y analizados para tomar acciones correctivas, minimizándose la probabilidad de cometer el mismo error dos veces, con el objetivo de mejorar la calidad del proceso, y por ende, del producto.

14.5.4 Control y seguimiento

El esfuerzo estimado por persona por semana era de 16 horas. Si bien aparece como de poca flexibilidad, cada integrante contaba con una lista de requerimientos asignados, por lo que teniendo los requerimientos y el esfuerzo estimado, podía compensar las horas de una semana con otra en el caso de no poder cumplir con el esfuerzo planificado.

Por este motivo el seguimiento de esfuerzo se hizo de forma mensual, correspondiendo con la duración promedio de los *sprints*. El detalle semanal fue propuesto para detectar las desviaciones de forma temprana y no recién a final de mes.

Mes	Agustín Alonso	Máximo Halty	Total Mes	Promedio Sem A.A	Promedio Sem M.H	Promedio Sem Eq.
Abril 2012	12	23,5	35,5	3	6	8
Mayo 2012	50	41	91	12	10	22
Junio 2012	36,25	28,5	64,75	9	7	15
Julio 2012	70	74,5	144,5	17	18	34
Agosto 2012	70	76	146	17	18	35
Setiembre 2012	75,5	92	167,5	18	22	40
Octubre 2012	69	70	139	16	17	33
Noviembre 2012	49	60	109	12	14	26
Diciembre 2012	68	81	149	16	19	35
Enero 2013	68,5	76	144,5	16	18	34
Febrero 2013	83	79	162	17	16	33
Total Tesis	651	702	1353	14	15	29
			Promedio Julio	16	18	34
			%Meta Julio	101%	111%	106%
			Objetivo	16	16	32
			%Meta Final	86%	93%	90%

Figura 43 - Seguimiento y control de esfuerzo

En la figura 43 se puede observar el avance mensual de cada integrante por separado y el avance del equipo. Asimismo se puede observar el esfuerzo promedio semanal, calculado en base al mensual, suponiendo que un mes tiene aproximadamente 4,2 semanas y como el seguimiento era mensual y no semanal, pudiendo re-distribuir el esfuerzo una semana con otra.

Se contaba con un seguimiento del cumplimiento del esfuerzo planificado representado en la tabla mostrada según la figura 44.

Mes	Agustín Alonso	Máximo Halty	Equipo
Julio 2012	104%	111%	108%
Agosto 2012	104%	113%	109%
Setiembre 2012	112%	137%	125%
Octubre 2012	103%	104%	103%
Noviembre 2012	73%	89%	81%
Diciembre 2012	101%	121%	111%
Enero 2013	102%	113%	108%
Febrero 2013	106%	101%	103%
Total Tesis	101%	111%	106%

OBJETIVO (Hrs)	16
-----------------------	----

Figura 44 - Esfuerzo real en base al planificado

En la figura 44 se puede evaluar el cumplimiento de esfuerzo por integrante y por equipo, que permitió realizar un seguimiento al esfuerzo semanal en base a que una semana equivale aproximadamente el 24% del mes.

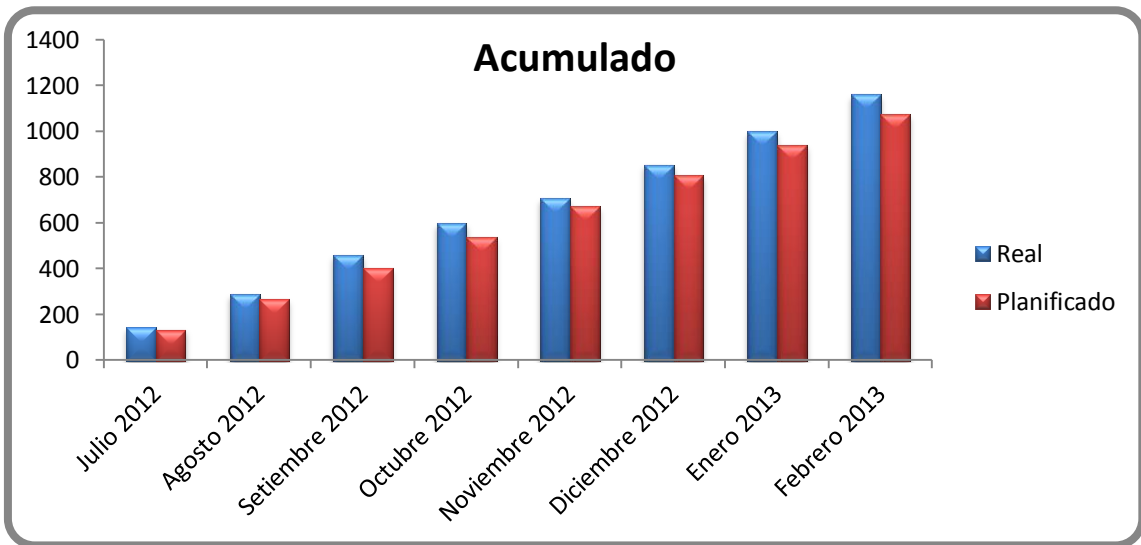


Figura 45 - Esfuerzo acumulado

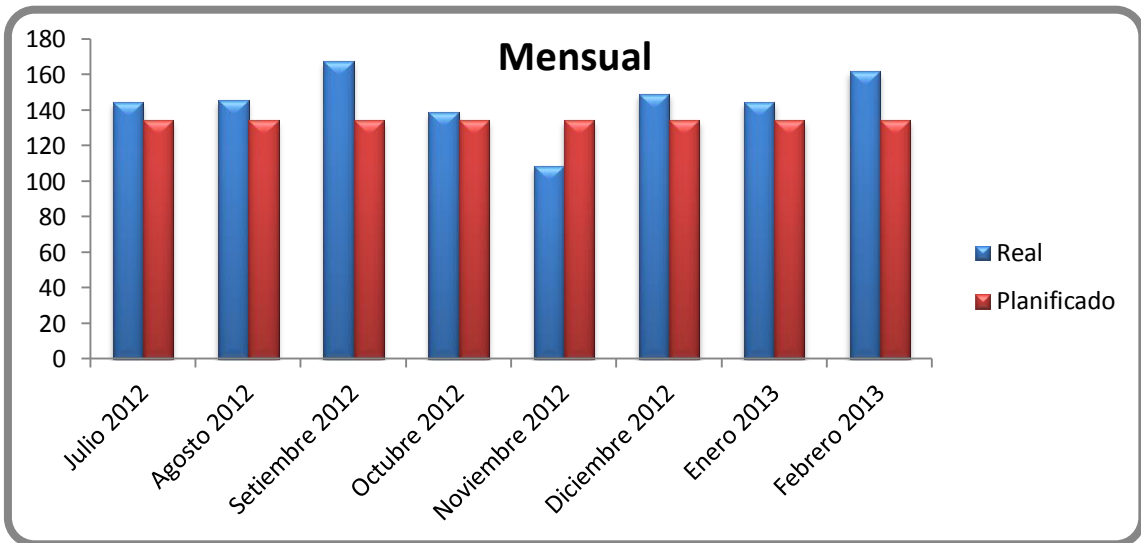


Figura 46 - Esfuerzo mensual

Finalmente, el seguimiento de esfuerzo también contaba con dos gráficas (Figuras 45 y 46) donde se podía observar el esfuerzo realizado de forma acumulada en el proyecto y el esfuerzo mensual.

El esfuerzo acumulado logró estar siempre alineado con el esfuerzo estimado al comienzo del proyecto, pero no así con el avance mensual. Se puede observar cómo Setiembre sobrepasó en gran cantidad el esfuerzo planificado, mientras que Noviembre no lo alcanzó.

En cuanto a Setiembre, esto se debió al receso de la Universidad donde ninguno de los integrantes del equipo tuvo que rendir exámenes o cursos alternativos. Por su lado, en Noviembre se sufrió el impacto de realizar materias curriculares de la carrera de forma simultánea al proyecto, donde en este período se hizo énfasis en culminar los trabajos obligatorios de dichas materias.

En cuanto al esfuerzo por área del proyecto, en el siguiente gráfico se puede observar como la distribución del esfuerzo estuvo principalmente distribuida en el desarrollo, la documentación y la investigación (Figura 47). Esta distribución no fue casualidad ya que dichas áreas fueron las más importantes del proyecto y en las que se invirtió el mayor esfuerzo, debiendo realizar una importante investigación para conocer el negocio y documentar dicho análisis, en conjunto con los documentos técnicos y de gestión.



Figura 47 - Esfuerzo por área

A su vez, siendo un proyecto de Ingeniería de Sistemas, donde además se esperaba un producto funcional como resultado del proyecto, se entiende justificada la diferencia de esfuerzo entre el desarrollo y las demás áreas.

Por su parte, se realizó también un seguimiento al avance por *sprint* de cada requerimiento realizado. La misma se realizó con planillas electrónicas (Excel), donde se establecía un avance estimado en “%” por requerimiento y sistema, el responsable asociado al mismo y en a qué *sprint* pertenece. A continuación se muestra un ejemplo del avance documentado para el primer *sprint* (Figura 48).

Tarea	Sprint	Aplicación	Responsable	Avance (%)
Login (Store)	1ero	Stores	Máximo	80%
Modificar perfil	1ero	Stores	Máximo	100%
Publicar artículo	1ero	Stores	Máximo	40%
Mantenimiento de artículos	1ero	Stores	Máximo	0%
Registrarse	1ero	Users	Agustín	50%
Login (User)	1ero	Users	Agustín	100%
Personalización	1ero	Users	Agustín	80%
Búsqueda de artículo	1ero	Users	Máximo	100%
Envío por mails	1ero	Users	Agustín	0%
Recibir news letter	1ero	Users	Agustín	100%
Envío de Feedback	1ero	Users	Agustín	10%

Figura 48 – Avance por sprint

Se contaba con un resumen del avance donde se podía obtener una aproximación al estado del sistema de forma rápida y simple de interpretar. Dicho avance se veía plasmado en las siguientes tablas mostradas en la figura 49:

Avance por Sprint		Alcance por Fecha	
Sprint	1° Spt	Fecha Límite	
1ero	60%	10-sep	
2do	100%	25-oct	
3ero	100%	09-dic	
4to	80%	14-ene	
Total	81%	Atrasado	-21,4%

Comienzo de Programación	3-jul-12
Termino de Programación	14-ene-13
Total de días estimados	195
Total de días a la fecha	200
Estimado	103%
Avance Real	81%

Figura 49 - Resumen del avance

14.6 Re planificación

Como se mencionó anteriormente, el control y seguimiento mensual del avance por requerimiento e integrante lograba, si las hubiera, encontrar las desviaciones en etapas tempranas al proyecto. De esta forma, se comenzaron a detectar algunos problemas en la planificación de los *sprints* del proyecto. Si bien el primer *sprint* terminó de forma acorde con lo planificado, el mismo contenía funcionalidades menos complejas que otros *sprints*, pero aun así fue planificado de tal forma de contener el mismo esfuerzo en cantidad de horas.

Por este motivo, los *sprints* restantes se vieron forzados a ser re-programados de forma que el alcance del *sprint* y el esfuerzo del mismo estuvieran distribuidos

equilibradamente, basándose en la experiencia del primer *sprint*. Una vez realizado el ajuste de esfuerzo, la duración del *sprint* se vio fuertemente afectada, de forma tal que se creó un nuevo *sprint* y los requerimientos fueron distribuidos nuevamente.

Finalmente se establecieron 4 *sprints* de igual duración y esfuerzo, existiendo algunos problemas en cuanto a la adaptación de la metodología en *sprints* propuesta por SCRUM, debido a que los *sprints* planificados en primera instancia eran largos en comparación con el estándar de SCRUM.

Dentro de los problemas detectados en la planificación, se destacaron el diseño gráfico y la utilización de ASP.NET MVC como arquitectura del sistema de usuarios. Los mismos fueron subestimados en la planificación inicial, debiendo corregir dichos aspectos en la re-planificación, donde se obviaron aspectos que requerían esfuerzo significativo y que no fueron incluidos en la primera planificación. Dentro de estos se destacan las altas, bajas y modificaciones de datos, que si bien eran necesarios para que el sistema funcionara de forma centralizada, no formaban parte del objetivo principal del sistema.

Por estos motivos se realizó una instancia de negociación con los clientes, de modo de ajustar el alcance del producto y sus requerimientos asociados. Dentro de estos se encontraban los requerimientos que de alguna forma u otra afectaban la compra y/o reserva de ítems, siendo suplantados por diseño gráfico y aspectos tecnológicos, muchos de estos vinculados con ASP.NET MVC.

Habiendo realizado todos los ajustes necesarios, donde el mayor cambio produjo en la fecha de finalización de la programación y la inclusión de un cuarto *sprint*, la planificación final se muestra en la figura 50.

Tarea	Descripción	Principales Requerimientos	Fecha de Finalización
ESRE	Documento de Especificación de Requerimientos	-	30-jun-12
Diseño Inicial	Diseño de arquitectura, MER y diseño alto nivel	-	18-jul-12
Sprint 1	Principales de User y backoffice para tiendas	Registro, Login, Búsqueda de Artículos	10-sep-12
Sprint 2	Diseño web con diseñadora gráfica	Diseño Web, migración MVC	25-oct-12
Sprint 3	Se pasa todo el diseño web a .NET	Publicación masiva, Like/Want, mi colección	9-dic-12
Sprint 4	Se termina por completo la web	Wish lists, aprobación de contenidos, reviews	14-ene-13
Puesta en producción	Subir la aplicación pública desde el servidor	Todas	14-ene-13
Pruebas de usuarios	Validaciones por parte de usuario + Testing	Todas	14-ene-13
Documentación Final	-	-	7-mar-13

Figura 50 – Planificación final

La lista de requerimientos a ser desarrollados por *sprint* fue adaptada a la planificación determinada por la realización de cuatro *sprints*. De esta forma la distribución final de requerimientos por *sprint*, por integrante y por sistema es representada en la figura 51.

Funcionalidades x Responsable					
Responsable	1° Spt	2° Spt	3° Spt	4° Spt	Total
Agustín	6	4	2	3	15
Máximo	5	2	5	7	19
Total Grupo	11	6	7	10	34

Funcionalidades x Aplicación					
Responsable	1° Spt	2° Spt	3° Spt	4° Spt	Total
Admin	0	1	0	3	4
Stores	4	3	1	0	8
Users	7	2	6	7	22
Total Grupo	7	3	6	10	34

Funcionalidades x Sprint	
Sprint	Cantidad
1ero	11
2do	6
3ero	7
4to	10
Total	34

Figura 51- Distribución final de requerimientos por sprint

Finalmente se destaca la distribución final de requerimientos por sistema y por integrante (Figura 52). En este listado se agregan ítems de investigación, diseño gráfico y aspectos tecnológicos, eliminándose los requerimientos para disminuir el alcance propuesto inicialmente, donde antes eran 40 requerimientos y finalmente totalizan 34, agregándose los aspectos mencionados:

Tarea	Sprint	Aplicación	Responsable
Login (Store)	1ero	Stores	Máximo
Modificar perfil	1ero	Stores	Máximo
Publicar artículo	1ero	Stores	Máximo
Mantenimiento de artículos	1ero	Stores	Máximo
Registrarse	1ero	Users	Agustín
Login (User)	1ero	Users	Agustín
Personalización	1ero	Users	Agustín
Búsqueda de artículo	1ero	Users	Máximo
Envío por mails	1ero	Users	Agustín
Recibir news letter	1ero	Users	Agustín
Envío de Feedback	1ero	Users	Agustín
Dar de alta una tienda	2do	Admin	Máximo
Diseño web de página	2do	Users	Agustín
Publicación masiva de artículos (50%)	2do	Stores	Agustín
Inicio y cierre de sesión	2do	Stores	Agustín
Migracion	2do	Users	Máximo
Investigacion tecnológica para Desktop	2do	Stores	Agustín
Pasar a .NET Diseño Web	3ero	Users	Máximo
Publicación masiva de artículos (50%)	3ero	Stores	Agustín
Opción "Me Gusta"	3ero	Users	Máximo
Agregar artículos a mi colección	3ero	Users	Máximo
Publicación en redes sociales	3ero	Users	Agustín
Seguir tiendas y/o usuarios	3ero	Users	Máximo
Visualización de artículos destacados	3ero	Users	Máximo
Crear news letter	4to	Admin	Agustín
Subir Contenidos	4to	Admin	Agustín
Aprobar ingreso de contenidos	4to	Admin	Agustín
Venta cruzada	4to	Users	Máximo
Escribir review de artículo	4to	Users	Máximo
Escribir review de tienda	4to	Users	Máximo
Wish List	4to	Users	Máximo
Categorización por tipos de usuarios	4to	Users	Máximo
Cambio de categoría	4to	Users	Máximo
Visualizar mi colección	4to	Users	Máximo

Figura 52 - Lista de requerimientos final

Un detalle de la planificación final es relevante a la hora de observar los cuatro sprints anteriormente definidos, destacándose el ciclo de vida incremental instanciado estableciendo claramente una primera iteración de investigación e Ingeniería de requerimientos, seguida por cuatro iteraciones idénticas y finalizando con la documentación y entrega final del proyecto. El Gantt del proyecto se muestra en la figura 53.

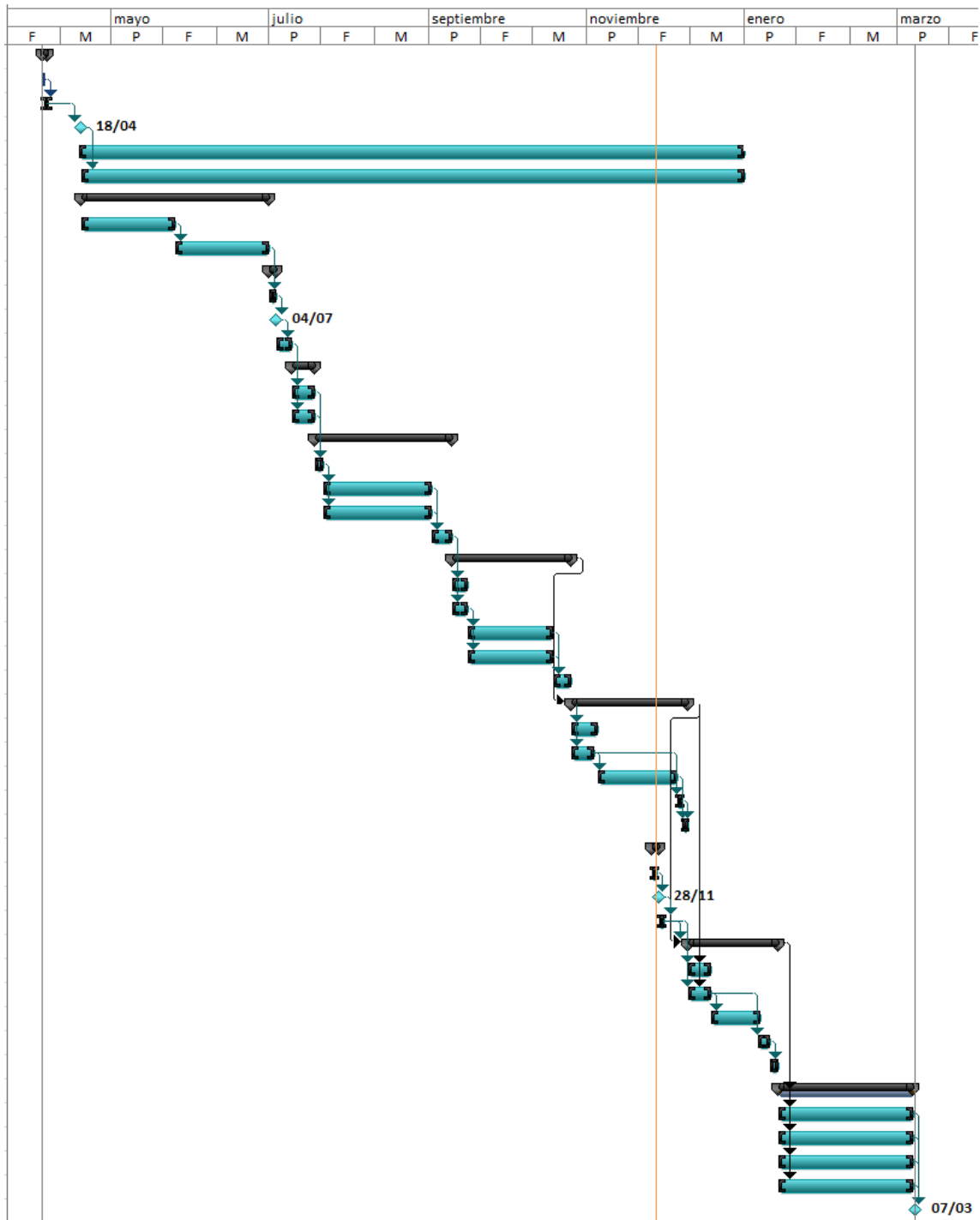


Figura 53 - Gantt del proyecto

14.7 Riesgos

14.7.1 Identificación

Al comienzo del proyecto se realizó una identificación primaria de riesgos dado que el mismo presentaba varios riesgos que necesitaban ser analizados de forma detallada. Se realizó una lista general de riesgos que fueron analizados de forma cualitativa y cuantitativa, evaluando su probabilidad de ocurrencia, impacto y magnitud.

Posteriormente, se tomaron todos los riesgos analizados, estableciendo una lista acotada de los riesgos más importantes según su magnitud, estableciendo planes de respuesta y contingencia junto con su respectivo seguimiento.

Para llevar a cabo la identificación de riesgo se utilizaron las siguientes técnicas:

- Tormenta de ideas.
- Reuniones con el Cliente.
- Juicio experto (Recomendaciones del tutor del proyecto y experiencias por parte del grupo).
- Revisión de documentación de proyectos anteriores.

Luego, fueron analizados en base a la tabla 15 de probabilidad de ocurrencia.

Impacto	Riesgos Negativos	Riesgos Positivos
0	Ninguno	Ninguno
1	Marginal	
2	Poco Importante	
3	Puede Atrasar	
4	Puede Detener	
5	Fracaso	Éxito

Tabla 15 - Tabla de impacto

Finalmente, se evalúa su magnitud en base a la siguiente fórmula:

$$\text{Magnitud} = \text{Impacto} \times \text{Probabilidad de Ocurrencia}$$

La probabilidad de ocurrencia se dividió según la tabla 16:

Nula	Escasa	Probable	Bastante	Muy	Hecho
0	0,2	0,4	0,6	0,8	1

Tabla 16 - Tabla de probabilidad de ocurrencia

14.7.2 Principales riesgos

En la tabla 17 se presentan los principales riesgos del proyecto junto con su descripción y su respectivo análisis cualitativo y cuantitativo.

Nombre	Descripción	Impacto	Probabilidad	Magnitud
Suplencias dentro del grupo	El grupo está formado por 2 integrantes. En consecuencia, si uno se atrasa o tiene un problema, el restante integrante queda como suplente obligatoriamente.	4	0,6	2,4
Experiencia en las distintas áreas	Como consecuencia de ser un grupo pequeño, es difícil que entre los dos integrantes se logre abarcar de forma completa todas las áreas de la gestión de proyecto.	3	0,4	1,2
Obligaciones con otras materias	Durante el transcurso del proyecto, ambos integrantes a su vez realizaron materias curriculares de la carrera. En el primer semestre fueron 3 materias por integrante y en el segundo semestre fueron 2.	3	0,6	1,8
Inestabilidad en las prioridades de los requerimientos	Si bien los requerimientos son bastante estables, la prioridad y secuencia entre ellos puede no estar bien definida. Esto se debe al dinamismo del negocio, impactando en las necesidades dentro del sistema.	3	0,4	1,2
Ambiente Local	Tener un ambiente local tiene sus ventajas prácticas pero en desventaja, el mantenimiento del ambiente debe ser realizado por el grupo. De esta manera, pueden ocurrir roturas de disco, pérdidas de información, entre otras.	4	0,6	2,4

Diseño Gráfico	Ninguno de los integrantes del grupo tiene conocimientos y capacidades para realizar un diseño gráfico acorde a las necesidades del negocio.	5	0,8	4
Tecnologías emergentes	La utilización de tecnologías emergentes como por ejemplo ASP.NET MVC, tiene sus ventajas en cuanto lo tecnológico. Sin embargo, ninguno de los integrantes tiene la experiencia suficiente para realizar un sistema basado en dicha tecnología.	4	0,8	3,2

Tabla 17 - Principales riesgos

Finalmente, para los riesgos propuestos anteriormente, se establecieron sus respectivos planes de respuesta y de contingencia (Tabla 18)

Nombre	Plan de Respuesta	Plan de Contingencia
Suplencias dentro del grupo	Realizar la planificación de forma tal que los problemas ocurridos que afecten el cronograma del proyecto puedan ser solucionados y las horas perdidas recuperadas en el transcurso de cada iteración. A su vez, importante resaltar que la estimación de esfuerzo semanal de 16 horas por persona incluye dichos aspectos, ya que en caso de no realizar las 16 horas semanales, las mismas pueden ser recuperadas en la próxima semana. Por otro lado, en caso que las 16 horas semanales no alcancen para desarrollar las funcionalidades propuestas en dichas horas, las mismas se pueden alargar siendo accesible realizar más horas semanales para llegar al objetivo.	En caso de ocurrir, el integrante restante toma la responsabilidad de realizar lo que el otro no pudo hacer. En este caso se acepta la suplencia, se documenta y queda pendiente de análisis una re-planificación para la siguiente iteración del desarrollo, si esta fuera necesaria.

Experiencia en las distintas áreas	Capacitación a ambos integrantes del grupo sobre el área que ninguno de ellos tiene el conocimiento suficiente como para tomar la responsabilidad. La capacitación se hace con la ayuda de material de materias anteriores, material en la biblioteca y buenas prácticas propuestas por el tutor.	En caso que la capacitación anteriormente mencionada no sea suficiente, la universidad provee referentes de cada área del proyecto. De esta manera, el grupo podrá consultar y pedir asesoría, no solo con el tutor, sino también con los referentes de cada área. Cabe resaltar que estos referentes forman parte de ORTs ^f y no de L.I.S.I.
Obligaciones con otras materias	Las obligaciones con otras materias y el esfuerzo que estas insumen deben ser plasmadas dentro de la planificación y alcance del proyecto. Por otro lado, es conveniente que las materias a ser realizadas por cada integrante sean las mismas, de forma de distribuir las responsabilidades con los obligatorios, equilibrando y optimizando los tiempos.	En caso de requerir un esfuerzo mayor por parte de las materias, la planificación fue tal que el esfuerzo perdido en determinado momento del proyecto pueda ser contrarrestado por el esfuerzo en otro momento. Es importante tener en cuenta que el equilibrio de esfuerzo se debe hacer dentro del sprint/iteración donde ocurrió y desvío. De esta forma, si el esfuerzo perdido es contrarrestado dentro del mismo sprint, el mismo no es traslado a futuras etapas.
Inestabilidad en las prioridades de los requerimientos	Como los requerimientos en sí mismos son estables, se negocia con el cliente una lista de requerimientos que serán incluidos en el desarrollo. Luego, se establece un circuito de control de cambios al producto propuesto por SCM, donde se detalla qué características debe tener el requerimiento y cómo se debe realizar el pedido para que sea incluido o excluido del desarrollo.	En caso de aceptar el pedido por parte de los clientes, SCM establece qué se debe hacer con dicho pedido. Asumiendo que el requerimiento tiene prioridad alta o media, si el pedido tiene un esfuerzo bajo, se acepta y se hace. Si en cambio tiene un esfuerzo medio, se debe negociar cómo este impacta al resto de los requerimientos, pudiendo quitar de la lista de requerimientos algún requerimiento anteriormente aprobado. Finalmente, si el cambio tiene un esfuerzo alto, pueden ocurrir dos casos. Por un lado, si la prioridad del requerimiento en evaluación es alta, el mismo pasa por el filtro de la negociación anteriormente mencionado. Sin embargo, si la prioridad es media, el mismo ya es considerado como un cambio en futuras versiones. Esto último también ocurre en el caso que la prioridad del

		requerimiento sea baja, independientemente del esfuerzo que consuma.
Ambiente Local	Estadísticamente, un disco duro tiene una probabilidad de 7% de fallar en 5 años. Siendo que la PC tomada como servidor tiene aproximadamente este tiempo, es adecuado realizar las precauciones correspondientes. De esta manera, se realiza un backup semanal de los documentos más importantes. En este caso, se realiza un backup semanal del código fuente de la aplicación y de la base de datos de prueba.	Si bien la ocurrencia del fallo de disco no tendrá pérdida de datos (no solo por los backups, sino porque cada integrante tiene una copia local del código fuente y la base de datos), podría dejar inutilizable la PC. De esta manera, el grupo cuenta con una 2da PC de respaldo, la cual en caso que la principal quede inhabilitada, esta 2da PC cuenta con todo el SW y HW disponible para ser el servidor de respaldo.
Diseño Gráfico	Al no tener las capacidades técnicas para realizar el diseño gráfico correspondiente a la Web de usuario, dicho diseño fue tercerizado. De esta forma, se mitiga el riesgo de realizar un diseño erróneo. Sin embargo, la terciarización debe ser gestionada adecuadamente para que se cumplan las pautas establecidas.	Si bien el diseño fue tercerizado, quien realizó el diseño fue una diseñadora cercana al grupo. De esta forma se trabajó en conjunto con el equipo de desarrollo ya que el diseño no fue implementado dinámicamente, sino que fue un boceto estático, debiendo adaptarlo a la tecnología utilizada. En caso de realizar un diseño erróneo, se decide terciarizar el diseño a una empresa de diseño Web.

Tecnologías emergentes	Capacitación a ambos integrantes del grupo sobre ASP.NET MVC en base a material de la biblioteca y tutoriales de la Web.	En caso de no adaptarse a la tecnología o que la misma insuma más tiempo de lo planificado, se migrará lo realizado hasta el momento a la tecnología ASP.NET, la cual ambos integrantes tienen la experiencia suficiente para continuar desarrollando el producto con dicha tecnología.
-------------------------------	--	---

Tabla 18 - Planes de respuesta y contingencia

14.7.3 Seguimiento y control

Se realizó el seguimiento y control cada quince días, analizando los riesgos ya identificados, y relevando nuevos riesgos que pudieran aparecer durante el desarrollo del proyecto e identificando aquellos que hayan disminuido su magnitud.

Para realizar el seguimiento de los riesgos más importantes del proyecto fue adecuado realizar una tabla de comparación de la magnitud de los riesgos contra el tiempo, debido a que no todos los riesgos tenían igual impacto y probabilidad de ocurrencia en el tiempo. De esta forma, la evolución de la magnitud de los riesgos se muestra en la figura 54.

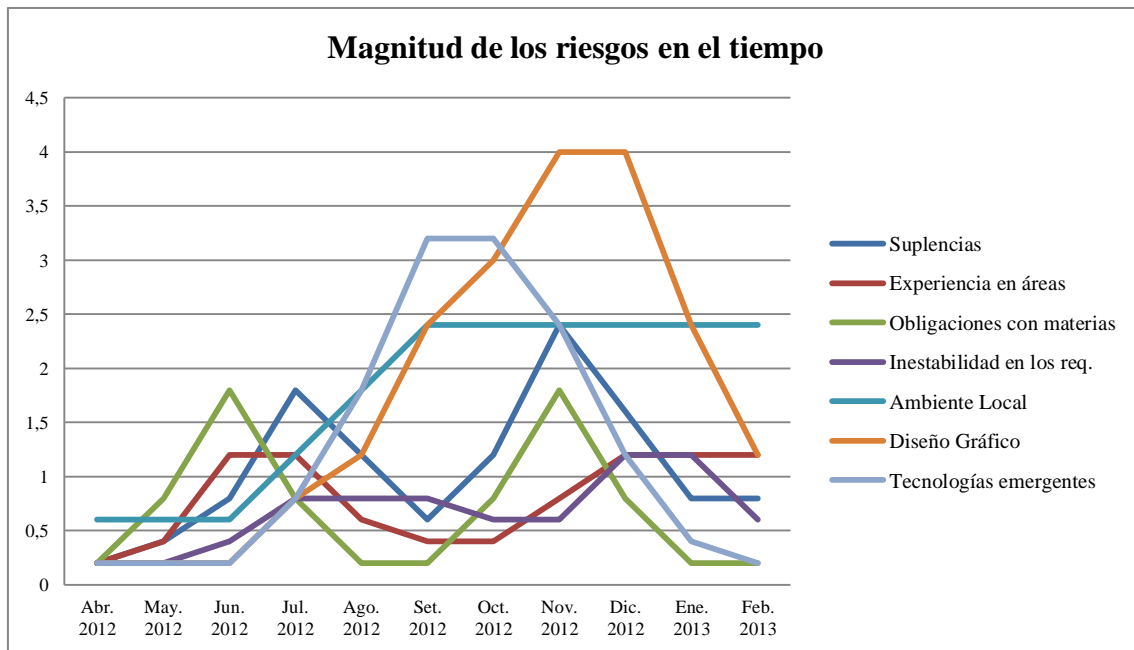


Figura 54 - Magnitud vs tiempo

Como se puede observar, los picos más importantes se dieron en Setiembre, Octubre y Noviembre, donde el desarrollo en ese período tomó un rol protagónico en el proyecto. De esta forma, la probabilidad ocurrencia de un riesgo, no solo era el más alto de todo el transcurso del proyecto, sino que además tenía un fuerte impacto sobre el mismo. Es por este motivo que en dichos meses se hizo hincapié, en el uso de tecnologías emergentes, diseño gráfico y las suplencias, siendo estos tres los riesgos con más impacto en todo el proyecto.

14.7.4 Riesgos ocurridos

La mayor parte de los riesgos fueron correctamente gestionados, habiendo establecido un correcto plan de respuesta para cada uno de ellos. Si bien los planes de respuesta lograron mitigar el efecto negativo que pudieron haber ocasionado en caso de ocurrir, el proyecto sufrió la ocurrencia de dos de los riesgos más importantes anteriormente mencionados: Tecnologías Emergentes y Diseño Gráfico.

La incorporación de ASP.NET MVC en el desarrollo del sitio Web de los usuarios fue el principal problema relacionado al uso de tecnologías emergentes. Si bien los usuarios fueron capacitados en dicho aspecto, la curva de aprendizaje de dicha tecnología fue más lenta de lo planificado, atrasando ciertas funcionalidades vinculadas únicamente al sitio Web de los usuarios.

En cuanto al diseño gráfico, fue error del equipo de proyecto no incluir como requerimiento del sistema la adaptación del diseño propuesto por la diseñadora a la tecnología utilizada, ya que el diseño propuesto tenía formato HTML estático.

Vinculado con el riesgo anterior, el pasaje del diseño gráfico a ASP.NET MVC también influyó en el uso de la herramienta, consumiendo más tiempo de lo planificado.

14.7.5 Solución

El atraso ocasionado por la inexperiencia en ASP.NET MVC tuvo lugar en las últimas semanas del primer *sprint*. Pues la estimación inicial de esfuerzo en relación a las funcionalidades propuestas para el proyecto no era acorde con el alcance del mismo.

Al culminar el primer *sprint* se realizó una reunión con los clientes para re-negociar el alcance del proyecto. Entre los problemas ocurridos en el primer *sprint*, se discutió el atraso por el uso de la tecnología ASP.NET MVC, resaltando frente a los clientes las ventajas que tenía el uso de esta en comparación con ASP.NET y los problemas vinculados al pasaje del diseño gráfico de forma estática y en formato HTML a formato compatible con ASP.NET MVC.

Se estableció un nuevo alcance del proyecto donde el pasaje del diseño gráfico fue incluido como un requerimiento nuevo y el uso de la tecnología ASP.NET MVC fue aceptado por parte de los clientes entendiendo su importancia y los problemas ocurridos.

La decisión de continuar el desarrollo con la tecnología anteriormente mencionada fue tomada por parte del equipo del proyecto, suponiendo que el conocimiento necesario fue adquirido en el primer *sprint*. De esta manera, los clientes accedieron a aceptar los

desvíos ocasionados en el primer sprint pero no futuros desvíos bajo el mismo concepto. De esta forma, el equipo asumió el riesgo de utilizar ASP.NET MVC nuevamente, pero desde otro nivel de *expertise* que le permitió asegurarse que dicho desvío no volviera a ocurrir, o en su defecto, fuera adecuadamente gestionable.

14.8 Comunicaciones

Las comunicaciones, tanto entre los integrantes del equipo como las comunicaciones con los agentes externos al mismo, eran muy importantes para el éxito del proyecto, pues cada involucrado al proyecto. Una buena gestión de las comunicaciones permitió asegurar la generación y distribución de la información del proyecto en tiempo y forma a cada involucrado.

No sólo se debía contar con la información a distribuir, sino que además se debían elegir correctamente los canales y el momento en el cual distribuir dicha información. Por este motivo, a continuación se detallan las distintas formas de comunicación utilizadas durante el transcurso del proyecto.

14.8.1 Interna

Sábados y domingos eran los días en los cuales más tiempo se dedicaba al proyecto, debido a que cada integrante mantenía un trabajo de entre 7 y 9 horas cada uno. Si bien existía una división de tareas bien definidas, el equipo optó por juntarse los fines de semana para trabajar.

En la decisión no solamente influyó la ayuda que cada integrante pudiese dar en caso de problemas, sino también la forma de comunicación entre los integrantes. Si bien en la actualidad existen tecnologías que permiten hablar con otras personas por Internet, además de poder compartir las pantallas en tiempo real, fue más eficiente trabajar en el mismo ambiente, tal cual lo sugiere la metodología SCRUM. Durante la semana, la comunicación se realizaba por “Google Talk” (véase Anexo III) y/o telefónica.

Es importante incluir al tutor dentro de la comunicación interna pues si bien no formaba parte del equipo de desarrollo, fue una base de influencia y ayuda durante todo el transcurso del proyecto. En este caso la comunicación periódica era vía mail, habiéndose creado un “Google Group” con 3 integrantes que contaban con casillas de correo de Google.

Se planificaba una reunión presencial mensual con el tutor, donde se realizaba una puesta a punto del proyecto, análisis de problemas ocurridos y discusión de dudas puntuales propuestas por el equipo y el tutor, realizándose en el campus del centro de Montevideo de la Universidad ORT Uruguay (véase Anexo VI).

14.8.2 Externa

Debido a la cantidad de potenciales tiendas de moda y usuarios, los clientes José Miguel Majó y Juan Manuel Dellepiane centralizaron todas las decisiones, siendo los principales clientes para el equipo.

La comunicación con ellos era fluida debido a la cercanía que tenían los integrantes del grupo con cada uno de ellos, recibiendo llamadas telefónicas para discutir algún tema puntual no demasiado importante o simplemente para obtener cierta información de forma rápida.

En las primeras etapas del proyecto donde se realizó la investigación y la Ingeniería de Requerimientos, el grupo se juntó todos los sábados durante un mes y medio para discutir los datos relevados del mercado. De esta forma se logró llegar al documento de especificación de requerimientos de forma conjunta, discutiendo las diferencias y similitudes y llegando a un acuerdo por ambas partes. Este es un aspecto muy importante a tener en cuenta, ya que realizando estas primeras etapas en forma conjunta y presencial, se lograron establecer vínculos de confianza y respeto entre las cuatro personas.

Luego de culminada la primer etapa, la comunicación entre el grupo y los clientes fue disminuyendo en frecuencia, donde además también se cambió el canal de comunicación. Si bien muchas veces se optaba por la vía telefónica para realizar consultas rápidas, se comenzó a utilizar la herramienta “Skype” o “Google HangOut” para realizar video-conferencias (véase Anexo III).

En ciertos momentos del proyecto se establecieron reuniones presenciales para lograr llegar a un acuerdo de forma más rápida y eficiente. En el momento que se tercerizó el diseño gráfico del Portal Web en la diseñadora gráfica, la comunicación con los clientes pasó nuevamente a ser presencial, debiendo evaluar de forma conjunta los diferentes diseñadores/as que podrían llegar a realizar el diseño Web de la página.

En cuanto a la comunicación con la diseñadora gráfica, se planificaron reuniones de seguimiento, las cuales fueron presenciales en el hogar de la diseñadora. En ellas se evaluaba el status del diseño, se discutían ideas y se proponían cambios. Una vez culminada la reunión de seguimiento, la comunicación pasaba a ser vía mail, donde se discutían ciertos plazos de entrega de cada diseño a realizar. A su vez, se utilizó la herramienta “Dropbox” para sincronizar los diseños que dicha diseñadora realizaba, ya que la misma permitía, no solo una herramienta fácil y eficiente para compartir archivos, sino también la posibilidad de realizar cambios por parte del equipo, estando 100% del tiempo sincronizados con el diseño por parte de la diseñadora.

15 Conclusiones

15.1 Ingeniería de requerimientos

Luego de obtener la lista de requerimientos, la misma fue priorizada en conjunto con los clientes, negociando qué requerimientos serán incluidos dentro del desarrollo del proyecto académico, y cuáles quedarán pendientes para futuras versiones de Looke.Me.

Luego de la primera iteración, en la cual hubo ciertos desvíos de planificación, el alcance debió ser acotado en funcionalidades. Habiendo negociado nuevamente con los clientes en nuevo alcance del sistema, se puede decir que el equipo gestionó de forma correcta el alcance definido en base a los requerimientos obtenidos y sus respectivas prioridades.

Por otro lado, cabe resaltar la correcta elección de los métodos utilizados para la extracción de requerimientos, ya que estos últimos lograron estabilidad en cuanto a su definición. No así ocurrió con la prioridad asociada a cada uno de ellos, debiendo ocasionalmente analizar el efecto de cambio de prioridad de los requerimientos.

15.2 Arquitectura y diseño

El grupo logró establecer una arquitectura estable desde un inicio del proyecto, planificando un diseño macro de la arquitectura con los diferentes componentes necesarios para desarrollar el sistema completo, dejando para las diferentes iteraciones un diseño más a bajo nivel. Esto permitió mitigar el riesgo de arquitectura inestable proveniente del uso de las metodologías R.U.P. y Espiral, logrando con gran éxito establecer la base arquitectónica, que a su vez incluye el M.E.R. de la base de datos.

Sin embargo, el equipo no supo gestionar correctamente la utilización de ASP.NET MVC como tecnología a utilizar en el sistema de usuarios. Esto se debe a que en un inicio se comenzó con ASP.NET, debiendo migrar las funcionalidades realizadas hacia ASP.NET MVC. Si bien esto comenzó a realizarse poco antes de culminar el primer *sprint*, teniendo pocas funcionalidades que migrar, el trabajo no fue fácil. Esto se debió a que la capacitación de los usuarios no fue suficiente, realizando pruebas de concepto poco autocontenidas.

De esta forma se concluye que es de suma importancia contar con un diseño arquitectónico completo a comienzos del proyecto, ya que este permite luego diseñar de forma más eficiente la capa lógica y de más bajo nivel. A su vez, es conveniente realizar pruebas de concepto en etapas tempranas del proyecto, y en lo posible, antes de comenzar el desarrollo.

15.3 Gestión de la configuración

Si bien algunos problemas encontrados fueron mencionados anteriormente en el capítulo 10 – Gestión de la configuración, es importante destacar en este momento la necesidad de planificar adecuadamente las herramientas necesarias para gestionar de forma los documentos, código fuente y los posibles cambios propuestos.

El equipo logró una correcta planificación en cuanto a las herramientas utilizadas, estableciendo un propósito para cada una de ellas de acorde a las facilidades que la misma puede ofrecer. Por ejemplo, la herramienta Dropbox permitió un excelente método para compartir y modificar archivos de documentación de forma remota entre integrantes del equipo. A su vez, la herramienta permitió compartir también dichos documentos con las personas adecuadas, siendo útil para el equipo compartir documentos tanto con el tutor como con los clientes.

Si bien Dropbox centraliza los archivos para que dos o más personas puedan acceder a ellos, no sería adecuado utilizar Dropbox como repositorio para el código fuente, ya que Dropbox no cuenta con conceptos tales como *Check-Out* o *Check-In*.

Por su lado, resulta de extrema importancia comunicar correctamente las herramientas a ser utilizadas. Este problema se vio principalmente en el cliente, ya que muchas veces este no solía entrar en las carpetas para ver ciertos cambios realizados. Incluso, enviaba material por correo electrónico, descentralizando la información incluida en Dropbox, debiendo realizar trabajo extra para mantener un correcto orden de los archivos de Dropbox, teniendo en cuenta además las diferentes versiones de cada archivo.

Finalmente, es importante resaltar la importancia de la implementación de respaldos periódicos de la información del servidor. Si bien dichos respaldos no fueron utilizados, esto brindó seguridad al equipo, trabajando con la tranquilidad de que su trabajo no será perdido.

15.4 Gestión de la calidad

Dentro de los aspectos más importantes de la calidad, se destaca la decisión de utilizar la metodología PSP para el proceso de desarrollo personal. De esta forma, ser un grupo integrado por 2 personas toma ventaja ya que mediante el uso de PSP se logra asegurar que el desarrollo del producto final fue realizado en base a procesos establecidos con el fin de asegurar la calidad del mismo, estableciendo métodos de control y mejora continua. Con contar con el rol de “Responsable de Calidad”, cada integrante es responsable y se compromete a desarrollar basado en procesos que aseguran la calidad del producto. A su vez, se utilizan métodos basados por TSP para la integración del trabajo de ambos integrantes.

Cabe resaltar el rol protagónico que asumieron las pruebas, no en las etapas finales del proyecto, sino también en el transcurso del mismo. Las pruebas permitieron al equipo validar con los clientes y verificar técnicamente que realmente se construían los sistemas de la forma que se esperaba, no solo superando las expectativas de los clientes, sino también generando cierta ansiedad por parte de los usuarios. De esta forma las pruebas lograron, no solo validar y verificar los requerimientos desarrollados, sino que, habiendo realizado pruebas en etapas intermedias del proyecto, se lograron detectar desvíos y oportunidades de mejora en etapas tempranas del mismo.

15.5 Gerencia

Siendo un proyecto desarrollado por únicamente 2 personas y con un alcance mucho mayor al alcance de una tesis de grado, puede afirmarse que la tarea más compleja realizada fue la planificación del proyecto. Si bien se emplearon los métodos y herramientas correctas, esto no fue suficiente para realizar una correcta planificación desde un inicio del proyecto.

Los problemas que ocasionaron los desvíos no fueron completamente de planificación. Como se mencionó anteriormente, uno de los principales problemas que ocasionó una re-planificación fue la incorporación de la tecnología ASP.NET MVC en el desarrollo de la Web de los usuarios. Si bien esto no fue planificado, el problema se agravó por mala capacitación y escasas pruebas de concepto realizadas.

Por otro lado, otro factor que influyó en la re-planificación fue la tercerización del diseño gráfico. El error cometido en ese momento no fue la tercerización en sí misma, sino la escasa planificación de la misma. Esta se realizó una vez comenzado el desarrollo, habiendo negociado y definido previamente el alcance con los clientes. De esta forma, no hubo manera de prever que el diseño gráfico sería entregado en formato HTML y estático, debiendo adaptar el mismo a ASP.NET MVC y darle comportamiento dinámico. Debió incorporarse una nueva tarea dentro del *Sprint Backlog*, no programada desde sus inicios, y en consecuencia, atrasó el calendario debiendo redefinir el alcance.

Es importante destacar la correcta utilización de planillas de seguimiento de esfuerzo y planificación, ya que la misma permitió al equipo a tomar medidas correctivas de forma temprana. Si bien la re-planificación fue uno de los problemas más grandes encontrados en la gestión del proyecto, dichos problemas fueron encontrados rápidamente gracias a las herramientas de control establecidas. De esta forma, si el equipo no hubiera contado con estas herramientas, el problema de la re-planificación pudo causar mayores desviaciones a las encontradas en el proyecto.

15.6 Conclusiones del proyecto

El equipo de desarrollo logró obtener la experiencia suficiente de la aplicación sistemática de procesos para lograr un producto final de calidad, habiendo establecido correctamente las metodologías a ser utilizadas.

Se realizó con éxito la puesta en práctica de los conocimientos de las distintas áreas del proyecto adquiridos durante el transcurso de la carrera, donde además se incluyó la experiencia del diseño gráfico como uno de los objetivos principales del proyecto. Si bien este diseño fue tercerizado, el equipo debió realizar la adaptación de dicho diseño sobre la tecnología utilizada en la tesis, adquiriendo nuevos conocimientos de diseño.

El trabajo en equipo superó las expectativas, no solo en cuanto a resultados sino también en la forma de trabajo, logrando un ambiente de respeto y apoyo mutuo, tanto entre los integrantes del equipo como también con los clientes y tutor del proyecto.

Si bien el proyecto tuvo sus ciertas desviaciones en cuanto al alcance definido en una primera etapa, el equipo logró gestionar dichas desviaciones de forma tal que las propuestas de cambios fueron aceptadas por parte de los clientes. Estas desviaciones fueron ocasionadas por un error en la gestión de los riesgos, habiendo mitigado el diseño gráfico propiamente dicho pero no su integración con el sistema.

El equipo de tesis se involucró en el desarrollo del negocio por parte del grupo comercial de Looke.Me, adquiriendo conocimientos relevantes del negocio de la moda que ayudaron a desarrollar un producto de mayor calidad desde el punto de vista del usuario final.

El resultado obtenido de las pruebas de usabilidad no fueron los esperados, aun cuando estos errores fueron específicamente de interfaz y no necesariamente de implementación. La corrección de dichos errores no insumió esfuerzo considerable en comparación a un cambio en la implementación del sistema.

Teniendo en cuenta los comentarios realizados por parte de los usuarios que realizaron las pruebas de usabilidad, la motivación para que Looke.Me lograra entrar en el negocio de la moda es aún mayor, estableciendo un vínculo comercial con los clientes José Miguel Majó y Juan Manuel Dellepiane.

16 Bibliografía

[1] Pressman, Roger. *Ingeniería de Software - Un enfoque práctico 5ta ed.* Nueva York: Mc Graw Hill, 2002.

[2] Kruchten, Philippe. *Architectural Blueprints - The "4+1" View Model of Software Architecture.* IEEE Computer Society, 1995.

[3] Sommerville, Ian. *Ingeniería del software. 7ta ed.* Madrid: Pearson, 2005.

[4] Buschmann F., Meunier R., Rohnert H., Sommerlad P. y Stal M. *Pattern-Oriented Software Architecture – A system of Patterns.* Chichester: John Wiley & Sons, 1996.

[5] Matalonga, Santiago. *Transparencias del curso Prácticas de Mejora de la Ingeniería de Software.* Montevideo, 2012.

[6] Delgado, Bruno. *Transparencias del curso Gestión de Proyectos.* Montevideo, 2011.

[7] Delgado, Bruno. *Transparencias del curso Ingeniería de Software 2.* Montevideo, 2011.

[8] Mousques, Gastón. *Transparencias del curso Arquitectura de Software.* Montevideo, 2011.

[9] Mousques, Gastón. *Transparencias del curso Diseño de Aplicaciones.* Montevideo, 2010.

[10] Schwaber, Ken. *Agile Project Management with Scrum.* Washington: Microsoft Press, 2004.

[11] Wikipedia (2013, Enero). [On-line]. Disponible en:
<http://es.wikipedia.org/wiki/Scrum>

[12] Osterwalder A. y Pigneur Y. *Business Model Generation.* Nueva Jersey: John Wiley & Sons, 2010.

[13] Highsmith J y Cockburn A. (2013, Enero 15). *Agile Software Development: The Business of Innovation*. [On-line] Disponible en: http://sunset.usc.edu/events/2002/arr/Agile%20Software%20Development_The%20Business%20of%20Innovation.pdf

[14] Manifesto for Agile Software Development. (2013, Enero). [On-line]. Disponible en: <http://agilemanifesto.org/>

[15] Bass L., Clements P. y Kazman R. *Architecture in Practice 2da ed.* Boston: Addison-Wesley Professional, 1998.

[16] Booch G., Rumbaugh J. y Jacobson I. *The Unified Modeling Language User Guide*. Massachusetts: Addison-Wesley Professional, 1999.

[17] PMI Publications. *A guide to the Project Management Body of Knowledge*. Pennsylvania: Project Management Institute Inc., 2004.

[18] Thayer R. y Dorfman M. *Software Requirements Engineering*. Washington: IEEE Computer Society, 2000.

[19] Pinto, Jeffrey K. *Project Management Handbook*. San Francisco: Jossey-Bass Inc., 1998.

[20] Leffingwell D. y Widrig D. *Managing Software Requirements*. Indianapolis: Addison-Wesley Professional, 2000.

[21] Martin, Robert C. *Agile Software Development – Principles, Patterns and Practices*. New Jersey: Prentice Hall, 2003.

[22] Bosch, Jan. *Design & Use of Software Architectures*. Nueva York: Addison-Wesley Professional, 2000.

[23] Design Patterns and Architectural Guidance. (2012, Julio). [On-line]. Disponible en: <http://www.dofactory.com/Default.aspx>

[24] Humphrey, Watts S. *The Personal Software Process*. Massachusetts: Addison-Wesley Professional, 1997.

[25] Esty (2012, Mayo). [On-line]. Disponible en: <http://www.etsy.com>

[26] Dress For Less (2012, Mayo). [On-line]. Disponible en: <http://www.dress-for-less.es/>

[27] Tienda Nube (2012, Mayo). [On-line]. Disponible en: <http://www.tiendanube.com/>

[28] Vitryn (2012, Mayo). [On-line]. Disponible en: <http://vitryn.com/>

[29] Dafiti (2012, Mayo). [On-line]. Disponible en: <http://www.dafiti.com.ar/>

[30] Forever21 (2012, Mayo). [On-line]. Disponible en: <http://www.forever21.com/>

[31] Urban-Outfitters (2012, Mayo). [On-line]. Disponible en:
<http://www.urbanoutfitters.com>

[32] Icónica (2012, Mayo). [On-line]. Disponible en: <http://www.iconica.com.uy/>

[33] Lookea.Me en el CIE (2012, Julio). [On-line]. Disponible en:
<http://cie.ort.edu.uy/innovaportal/v/4157/1/cie.ort.front/lookeame.html>

[34] Wikipedia (2013, Febrero). [On-line] Disponible en:
http://es.wikipedia.org/wiki/Centro_comercial

[35] Facebook (2013, Febrero). [On-line] Disponible en: <https://www.facebook.com/>

[36] Twitter (2013, Febrero). [On-line] Disponible en: <https://twitter.com/>

[37] Wikipedia (2013, Febrero). [On-line] Disponible en:
<http://es.wikipedia.org/wiki/Franquicia>

[38] El diario de info-negocios (2013, Febrero). [On-line] Disponible en:
<http://www.infonegocios.biz/nota.asp?nrc=25864&nprt=1>

[39] Comind Work (2012, Noviembre). [On-line] Disponible en:
<http://grupo3.comindwork.com/Login.aspx>

- [40] Dropbox (2013, Febrero). [On-line] Disponible en: <https://www.dropbox.com/home>
- [41] GeneXus (2012, Agosto). [On-line] Disponible en: <http://www.genexus.com/>
- [42] Entity Framework (2012, Julio). [On-line] Disponible en: <http://msdn.microsoft.com/es-es/library/bb399572.aspx>
- [43] Wikipedia (2013, Febrero). [On-line] Disponible en: http://en.wikipedia.org/wiki/Domain-driven_design
- [44] Unit of Work (2013, Febrero). [On-line] Disponible en: <http://geeks.ms/blogs/jtorrecilla/archive/2011/08/16/patr-243-n-unit-of-work-iow-o-unidad-de-trabajo.aspx>
- [45] Unit of Work (2013, Febrero). [On-line] Disponible en: <http://martinfowler.com/eaCatalog/unitOfWork.html>
- [46] DB4O (2012, Octubre). [On-line] Disponible en: <http://www.db4o.com/>
- [47] Ciclo de Deming (2012, Diciembre). [On-line] Disponible en: http://es.wikipedia.org/wiki/C%C3%ADrculo_de_Deming
- [48] Zoom-Us (2013, Enero). [On-line] Disponible en: <http://zoom.us/>
- [49] Mantis (2013, Febrero). [On-line] Disponible en: <http://www.mantisbt.org/>
- [50] CIE (2013, Febrero). [On-line] Disponible en: <http://cie.ort.edu.uy/>
- [51] Amazon (2012, Junio). [On-line] Disponible en: <http://www.amazon.com/>
- [52] Razor (2012, Setiembre) [On-line] Disponible en: http://msdn.microsoft.com/en-us/vs2010trainingcourse_aspnetmvc3razor.aspx
- [53] Equipos Mori (2012, Agosto) [On-line] <http://www.equipos.com.uy/>