



FACULTAD DE ECONOMÍA Y NEGOCIOS
ESCUELA DE INGENIERÍA INFORMÁTICA
EMPRESARIAL

Desarrollo e implementación de portal web para sistema de reservas. Caso: Lingo's Talca

Autores

César Soto Rojas
Bastián Cáceres Andrades

Profesor guía

Luis Eduardo Canales Carrasco

Proyecto de tesis para optar al título de INGENIERO EN INFORMÁTICO EMPRESARIAL

TALCA-CHILE
2021

CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su unidad de procesos técnicos certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.



Talca, 2023

Índice de contenido

| | |
|---|----|
| Índice de contenido | 2 |
| Resumen ejecutivo | 4 |
| Abstract..... | 5 |
| Agradecimientos | 6 |
| 1. Introducción | 7 |
| 2. Marco Teórico | 11 |
| 2.1 Sistemas de información | 11 |
| 2.1.1 Sistema de información personalizado | 11 |
| 2.1.1.1 Ventajas de un software personalizado | 11 |
| 2.1.1.2 Desventajas de un software personalizado | 12 |
| 2.1.2 Sistemas de información web | 12 |
| 2.1.2.1 Componentes característicos de los sistemas web | 12 |
| 2.1.2.2 Tipos de alojamiento web..... | 14 |
| 2.1.2.3 Dominios web | 16 |
| 2.1.2.4 Frontend v/s Backend | 17 |
| 2.1.2.5 Frameworks | 22 |
| 2.2 Arquitectura de Software | 23 |
| 2.2.1 Patrones Arquitectónicos | 23 |
| 2.2.2 Modelo - vista - controlador (MVC)..... | 24 |
| 2.3 Adopción de sistemas de información en pymes | 24 |
| 2.4 Sistemas de información para reserva de espacios de trabajo colaborativo | 25 |
| 2.5 Ingeniería de requerimientos | 27 |
| 2.5.1 Tipos de requerimientos..... | 27 |
| 2.6 Metodologías de desarrollo de sistemas de información..... | 28 |
| 2.6.1 Metodologías de desarrollo de software tradicionales | 28 |
| 2.6.2 Metodologías de desarrollo de software ágiles..... | 33 |
| 2.6.3 Metodologías Ágiles VS Metodologías Tradicionales | 44 |
| 3. Metodologías por ocupar..... | 46 |
| 3.1 Aplicación de las metodologías | 47 |
| 3.1.1 Adaptación de RUP como metodología en la estructura general del proyecto | 47 |
| 3.1.2 Kanban como metodología complementaria en la gestión de tareas..... | 49 |
| 4. Resultados..... | 50 |
| 4.1 Fase de Inception | 50 |
| 4.2 Fase de Elaboración..... | 53 |

| | |
|--|----|
| 4.3 Fase de construcción..... | 60 |
| 4.4.1 Herramientas por utilizar | 60 |
| 4.4.2 Sprint 1 | 62 |
| 4.4.3 Sprint 2 | 70 |
| 4.4.4 Sprint 3 | 73 |
| 4.4.5 Sprint 4 | 77 |
| 4.5 Fase de Transición | 81 |
| Eventos y observaciones de la iteración final..... | 82 |
| 5 Conclusiones | 83 |
| Bibliografía..... | 85 |
| Anexos..... | 88 |

Resumen ejecutivo

La utilización de sistemas de información en las pymes ha sido un pilar fundamental durante los últimos años, debido a la optimización y control que estos entregan al negocio, sin embargo, estos pequeños negocios no siempre tienen la posibilidad de contar con estos sistemas debido al costo que estos poseen. Así es actualmente en el caso de la empresa Lingo's, la cual se dedica al arriendo de espacios de trabajo personal. El método que posee actualmente Lingo's para realizar reservas es de modo manual, siendo solicitadas vía WhatsApp, llamada o acudiendo presencialmente al local, anotando la hora de la reserva y todos los detalles de esta en un cuaderno o en Google calendar, lo que en ocasiones generaba una descoordinación o que se realizara más de una reserva para diferentes personas en un mismo horario.

Se puede concluir a partir del desarrollo de este trabajo que la implementación de un sistema de información que permita realizar y pagar reservas ayuda a mejorar notablemente la experiencia del usuario nativo digital con la empresa además de reducir el margen de error humano en comparación con el método tradicional de reservas y mejorar el manejo de datos de clientes. Con este proyecto se busca facilitar el acceso a las tecnologías de información para las pequeñas y medianas empresas mostrando la relevancia que estas tienen para su mejor funcionamiento.

Abstract

The use of information systems in SMEs has been a fundamental pillar in recent years, due to the optimization and control that they deliver to the business, however, these small businesses do not always have the possibility of having these systems due to the cost that they have. This is currently the case with the Lingo's company, which is dedicated to the rental of personal workspaces. The method that Lingo's currently has to make reservations is manually, being requested via WhatsApp, calling or going in person to the premises, noting the time of the reservation and all the details of this in a notebook or in Google calendar, which sometimes It generated a lack of coordination or that more than one reservation was made for different people at the same time.

It can be concluded from the development of this work that the implementation of an information system that allows making and paying reservations can significantly improve the experience of the digital native user with the company in addition to reducing the margin of human error compared to the traditional method reservations and improve customer data management. This project seeks to facilitate access to information technologies for small and medium-sized companies, showing the relevance that these have for their better functioning.

Agradecimientos

“Mis eternos agradecimientos en cada logro siempre irán dirigidos a mis padres, los cuales han dado todos sus esfuerzos en apoyarme y entregar lo mejor de sí mismos para llegar permitirme llegar hasta acá. Por último, pero no menos importante, gracias a mi mismo por nunca rendirme, por siempre seguir adelante pese a las dificultades y levantándome cada vez que caía.”

César Esteban Soto Rojas

1. Introducción

En la actualidad, los sistemas de información forman parte fundamental para el éxito de las pymes (Pequeñas y Medianas Empresas), es por esto que se ven bajo la presión de adaptarse a los cambios digitales que el mercado demanda, impulsados por la revolución tecnológica que marca esta nueva era. La noción imperante trae consigo el concepto de que las TI (Tecnologías de Información) se han vuelto más accesibles, fáciles de usar y más baratas, dando acceso a las pymes a una amplia gama de tecnologías que apoyan las funciones comerciales, finanzas, contabilidad, administración de recursos humanos, entre otros, entregando información crucial a los emprendedores para un proceso de toma de decisiones más informado y, por ende, más asertivo (Pelletier-Cloutier, 2019).

Dentro de este grupo de enigmáticos cambios tecnológicos y económicos podemos observar cómo los servicios se han convertido en el sector más importante de la economía mundial. Incluso en nuestro país en donde la extracción de materias primas y la agricultura han predominado en las últimas décadas, la denominada economía de servicios se encuentra mucho más desarrollada que en otros países asiáticos o africanos y se asemeja a la europea, sin embargo, su crecimiento es lento y paulatino, por lo que la innovación se vuelve un factor particularmente importante (Arias-Ortiz, 2015).

Según Pavón, Puente, Infante y Blanco (2016), un gran número de empresas busca innovar mediante la informatización, utilizando sistemas de Planificación de Recursos Empresariales (ERP) debido a que estos abarcan una gran cantidad de aplicaciones software que soportan las operaciones de negocios diarias y la toma de decisiones basado en la integración de los dominios funcionales en las organizaciones, lo que se ve presente con mayor frecuencia en pymes del sector de servicios debido a la estructura de costes y cadena de valor que estas poseen.

La PYME Lingo's de la ciudad de Talca no es la excepción a la regla y también se encuentra en la búsqueda de factores innovadores que les permitan diferenciar sus servicios, visualizando como prioridad una completa transformación digital de sus procesos de negocio. Esta PYME está dedicada al servicio de arriendo de oficinas multipropósitos y espacios equipados para la realización de talleres, sesiones de psicología, kinesiología, yoga, clases de idiomas, entre otros, según los requerimientos de sus clientes. Esta particularidad de enfocarse en un tipo de cliente

que realiza reuniones recurrentes para sesiones de distinto índole les ha permitido posicionarse en un mercado altamente competitivo en la ciudad, debido mayormente a la ubicación central de estos, no siendo el caso de la empresa Lingo's.



Figura 1 – Mapa de los principales coworking en Talca – Fuente: elaboración propia (2021)

Actualmente el proceso de reservación en Lingo's se realiza vía telefónica, registrando los datos de reserva a través de Google Calendar o de forma manual en un bloc de notas lo que provoca muchas veces confusión a la hora de la confirmación de pagos o cuando se hacen cancelaciones imprevistas. Esto conlleva insatisfacción por parte de los clientes traducido en una baja tasa de retención de estos, sobrecarga laboral de los trabajadores producto del desorden y mal manejo de los datos.

A pesar de que anteriormente se han utilizado otras herramientas ERP de pago para gestionar las reservas de las salas, estas no contaban con la característica que permite realizar reservas online lo cual es la principal problemática en la empresa.

Adicionalmente, debido a la crisis sanitaria del SarsCov-19, el Ministerio de Salud de Chile (MINSAL) a través de Sernatur (2020) exige a las pymes del rubro de servicios protocolos sanitarios en el registro de datos de sus clientes, los cuales deberán hacer

una declaración jurada sobre el comportamiento dando cuenta de que no han tenido contacto estrecho con algún infectado y no han presentado síntomas en los últimos 14 días, promoviendo el uso de sistemas web para facilitar el seguimiento de posibles contagiados en caso de la aparición de un brote del virus y de esta forma tomar las acciones necesarias de forma rápida para prevenir su esparcimiento.

Es por esto por lo que se propone el desarrollo e implementación de un portal web que permita a los clientes realizar reservaciones de las salas disponibles a través de la página web de la empresa, poder ver la disponibilidad de las salas y sus características (m2, instalaciones, dependencias, implementos, etc.), cancelar una reservación, cambiar el horario o fecha de una, realizar pagos. Además, se podrán contratar planes mensuales para clientes que necesiten un lugar fijo para realizar sus actividades o talleres de 16, 32 y 64 horas. Por parte del administrador, el sistema permite ver las reservas realizadas, almacenar datos de los clientes, ver los pagos realizados, visualizar la disponibilidad de las salas, edición de las características de las salas en caso de remodelación o implementación de nuevas características.

Los objetivos de este proyecto son:

Objetivo General:

Desarrollar e implementar un sistema de gestión de reservas para la empresa Lingo's Talca

Objetivos Específicos:

- OE1: Realizar un diagnóstico de la situación actual, para de esta forma entender de buena manera la problemática de la empresa.
- OE2: Diseñar y desarrollar una solución que resuelva las problemáticas de la PYME.
- OE3: Desplegar una solución que resuelva la problemática diagnosticada.

2. Marco Teórico

2.1 Sistemas de información

Los sistemas de información juegan un rol importante en la era digital, ayudando a las empresas emergentes a posicionarse en el mercado en el que entran y a establecer ventajas competitivas por sobre las empresas competidoras. Es por ello por lo que es importante que estas empresas elijan qué tipo de sistema implementar según las características atribuibles a la organización y que estas se ajusten a las características de los sistemas disponibles.

2.1.1 Sistema de información personalizado

Adaptado para cada organización, ha sido la solución preferida para las empresas durante las últimas décadas, debido a que se realiza de manera adaptativa a la empresa, optimiza y entrega soluciones específicas a las problemáticas de la empresa para la que ha sido desarrollado, además de la gran susceptibilidad al cambio que estos poseen, debido que ante cualquier cambio que los clientes del sistema en uso, pueden ser agregadas nuevas características para dar solución a nuevas problemáticas dentro de la empresa (Wenchao-Jian-Ping-Patni, 2013)

2.1.1.1 Ventajas de un software personalizado

- **Alta personalización:** Un software on premise es diseñado a la medida y desarrollado para resolver las problemáticas específicas de la organización. A medida que el cliente requiera más características en el software, estas pueden ir siendo desarrolladas e integradas sin problema.
- **Seguridad:** Al ser un software instalado de forma local en la organización, el acceso queda limitado a los integrantes de esta, dificultando a las personas que de forma malintencionada quieran acceder a esta.

2.1.1.2 Desventajas de un software personalizado

- **Tiempo de implementación:** Al ser un tipo de software diseñado a la medida de la empresa y que debe ser instalado de forma local en la máquina y que este funcione en conjunto con todo el hardware (servidores, equipos informáticos, etc.), los tiempos de desarrollo e implementación del software son mucho mayores al de un software almacenado en la nube.

2.1.2 Sistemas de información web

Los sistemas de información son herramientas destinadas a potenciar a las organizaciones que logran hacer un uso óptimo de estas, ayudando a lograr que estas sean completadas más rápido a través de la automatización y además entregan información valiosa a la organización para ayudar en la toma de decisiones.

En un mundo cada vez más digitalizado es muy importante mantener a disposición datos que respalden las decisiones estratégicas de la organización, las cuales pueden definir el éxito de esta, permitiendo lograr las metas y objetivos establecidos por la empresa, todo esto debido a que la toma de decisiones a través de los datos permite mantener un bajo nivel de error humano, y simplemente se basa en datos estadísticos (Marcano-Talavera, 2007).

Además del factor del buen manejo de la información que un sistema de información le puede entregar a una organización, es importante destacar que permite mantener una relación más cercana con los clientes (CRM), conocerlos y saber qué decisiones tomar para mantener una buena fidelización de los clientes.

2.1.2.1 Componentes característicos de los sistemas web

Las principales componentes de los sistemas web son clientes, servidores y el protocolo mediante ambos se comunican. Estos se encuentran estandarizados y no son creados por el programador.

Protocolo HTTP: El protocolo http se encuentra dentro de los protocolos más utilizados en la web, que permite la conexión de sistemas, haciendo más fácil el intercambio de información entre distintos computadores.

Ciente web: programa mediante el cual el usuario interactúa solicitando al servidor el envío de recursos que desea obtener mediante HTTP (Luján Mora, 2002). Este se encuentra formado por el código HTML que da estructura a la página web además del código ejecutable realizado en lenguaje script (Javascript) o pequeños programas realizados en Java. Este también puede emplear plug-ins permitiendo mostrar otros contenidos multimedia (Macromedia Flash).

La función del cliente web es interpretar las páginas HTML y sus diferentes recursos (imágenes, sonidos, etc.)

Se suelen utilizar las siguientes tecnologías para su programación:

- ✓ HTML
- ✓ CSS
- ✓ DHTML
- ✓ Lenguajes de script como Javascript, VBScript
- ✓ ActiveX
- ✓ Applets programados en Java
- ✓ Tecnologías que requieren existencia de un plug-in dentro del navegador (Adobe Acrobat Reader, Autodesk MapGuide, VRML, etc.)

Servidor: El servidor web es quién espera constantemente solicitudes por parte del cliente a través del protocolo HTTP. La parte del servidor de la aplicación web se encuentra formada principalmente por:

- Páginas estáticas: Son documentos que contienen lenguaje de marcado Html, y muestran siempre la misma información.
- Recursos multimedia: Son documentos, contenido visual, etc. Que están disponibles para el cliente dentro del sitio para ser descargado o visualizado.
- Programas o Scripts: Estos son ejecutados por el servidor web, los cuales arrojan como salida una página HTML luego de ser solicitado en el navegador del cliente

Protocolo: El protocolo permite la comunicación entre el cliente y el servidor, permitiendo facilitar el intercambio de información entre distintos computadores según Luján Mora (2002).

2.1.2.2 Tipos de alojamiento web

El alojamiento web o hosting como es más conocido dentro de la comunidad de desarrollo, es un espacio en un servidor que permite almacenar los archivos que componen a un sitio web (los del frontend y backend), permitiendo a los usuarios visualizar un sitio desde la web.

A decisión del desarrollador o del equipo de trabajo, existen varios tipos de hostings, dependiendo del tamaño del sitio web, a cuántas personas se quiere llegar y cuánto rendimiento se necesita en este, es por ello por lo que los principales son:

Hosting compartido : Este tipo de alojamiento es uno de los más básicos y económicos, por lo que en la mayoría de los casos es utilizado por personas que recién están iniciando en el mundo del desarrollo web o también quienes desarrollan sitios web básicos y no necesitan tanta velocidad en el servidor, sin embargo, así mismo como es el más económico, es el más inestable, puesto que al ser un hosting con servidores compartidos, cuando uno de estos servidores tiene problemas de rendimiento afecta directamente a todos los demás usuarios del hosting.

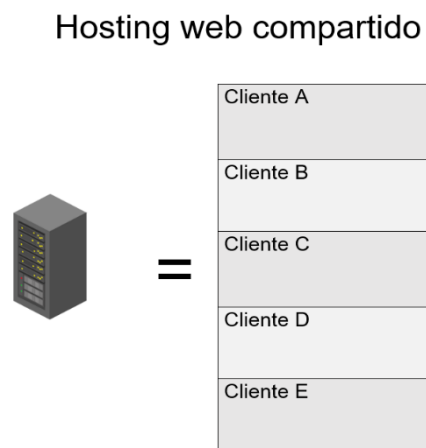


Figura 2 - Hosting web compartido – Fuente: elaboración propia (2021)

Hosting en servidor dedicado: Este tipo de alojamiento web es el más caro, puesto que requiere comprar un servidor propio, en el cual se aloja el sitio web, teniendo total control y administración del sistema, teniendo también un muy buen rendimiento al no tener que compartir con otros usuarios, además de que la o las máquinas conectadas al servidor, interactúan directamente con él.

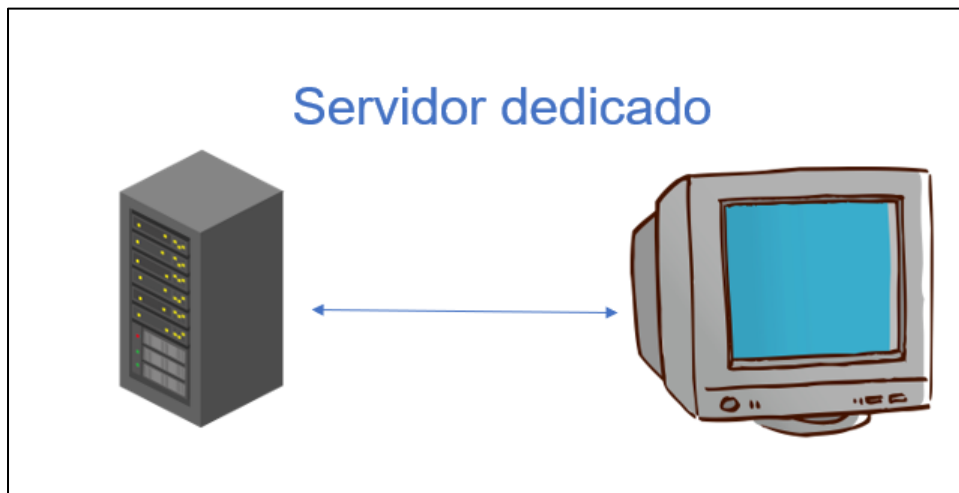


Figura 3 - Hosting servidor dedicado – Fuente: elaboración propia (2021)

Servidor privado virtual: Los servidores privados virtuales se encuentran justo en medio de los servidores compartidos y los dedicados, ya que es ideal para usuarios que necesitan mayor control en su sitio web, pero no necesariamente necesitan un servidor dedicado debido a su alto costo. A pesar de que de la misma forma que los servidores compartidos, se convive con otros usuarios en el servidor, el servidor privado virtual posee un mayor rendimiento que el servidor compartido, además de poseer la característica de ser virtualizados y ser privados, por ende, solamente alguien que tenga permitido acceder a este servidor podrá hacerlo.

Hosting en la nube: El hosting en la nube o cloud hosting, es un tipo de alojamiento web orientado a almacenar sitios o aplicaciones web en un servidor virtual que se encuentra en la nube, utilizando múltiples servidores a la vez para mejorar la

disponibilidad, el cual a su vez puede acceder a un cluster (grupo de servidores) y cuando uno de ellos en caso de fallar otro se activa para siempre mantener la estabilidad de los servidores. Existen principalmente dos opciones a la hora de utilizar un cloud hosting:

- **Disco duro:** Es la opción más barata y rústica, la cual le entrega al usuario una correcta velocidad de carga, pero no la mejor
- **Disco de estado sólido:** Es la opción más moderna y rápida, sin embargo, posee precios más elevados a cambio del rendimiento que entrega y los tiempos de carga del sitio o aplicación web.

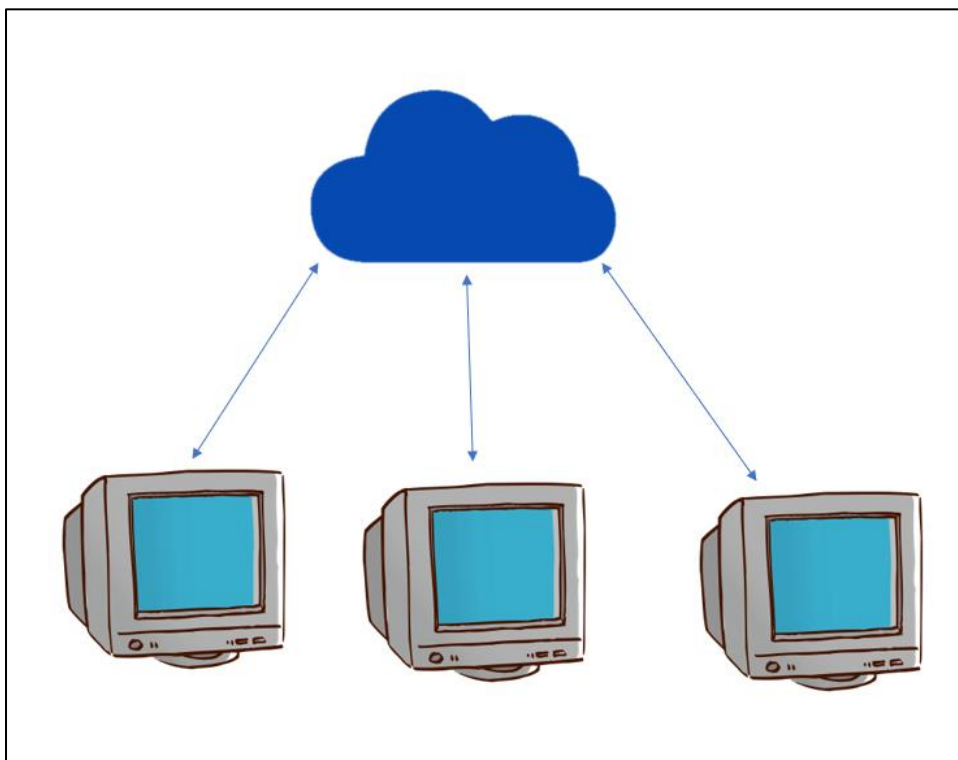


Figura 4 - Hosting en la nube – Fuente: elaboración propia (2021)

2.1.2.3 Dominios web

A la hora de subir un sitio web a la red, se debe tener en cuenta asignar un nombre único que le permita a los usuarios acceder al sitio a través de una dirección, puesto que, si no existiera este nombre de dominio, el usuario debería escribir exactamente la IP del sitio web al que quiere acceder, lo cual no es una experiencia intuitiva. El

dominio web se compone del nombre del sitio y la extensión web que dependerá de cada usuario cuál utilizar dependiendo de su ubicación geográfica y cuánto dinero desea desembolsar por este, existiendo principalmente los siguientes:

Genéricos: Son los mayormente utilizados en la web y los más populares, principalmente porque no se ajustan a una localización territorial determinada, sino más bien define el cómo el sitio web terminará en su extensión. (.com, .net, org, gob, etc)

Dominio territorial: Poseen una terminación en su extensión para cada país, como, por ejemplo .CL, .CO, .AR, .MX, etc.

Dominio de tercer nivel: Los dominios de tercer nivel poseen una extensión compuesta por los dos dominios anteriormente mencionados, entregando para cada usuario dependiendo de su localización territorial, la extensión .com más la extensión territorial correspondiente, por ejemplo: www.paginaweb.com.cl

2.1.2.4 Frontend v/s Backend

La web se compone de muchos documentos, los que se encuentran unidos entre sí por medio de enlaces o más conocido como su terminología en inglés como “Links”. Cuando un usuario desea entrar a una página web, escribe la URL comenzada en “WWW” en la barra del navegador, lo que envía una solicitud al servidor para que nos muestre la página web que ha requerido el usuario.

Una vez realizada la solicitud, el backend mantiene una comunicación constante con el frontend para que nos muestre la interfaz y los datos que solicita el usuario.

Sin embargo, luego de saber de manera breve cómo funciona el frontend y el backend, ¿qué es cada una de estas capas de un sistema web?

2.1.2.5 Frontend

En la capa del frontend nos encontramos todo lo que el usuario podrá interactuar directamente, ya sean las animaciones de los menús desplegables, el color de los botones, las funcionalidades que interactúan directamente con el cliente, etc.

Según (UNAM, 2008), los lenguajes de programación utilizan caracteres de alfabeto para poder proporcionar información a las computadoras y así poder entregar las instrucciones de lo que se quiere realizar. En un inicio, los lenguajes de programación eran basados en código "máquina", el cuál era directamente entendido por ella e interpretaba los bloques de código escritos por los desarrolladores, es por ello, que esta engorrosa tarea de escribir en lenguaje máquina, el cuál era difícil de comprender fue tomando un rumbo distinto, en donde los desarrolladores comenzaron a buscar formas de crear lenguajes de programación parecidos a los lenguajes humanos y que estos luego fueran interpretados por la máquina.

Para la capa del frontend se utilizan principalmente 3 de los lenguajes más conocidos en el desarrollo web.

HTML: Se ve por primera vez en el año 1991 en el lanzamiento de la web bajo su primera versión (HTML1) creada por Tim Berners-Lee, siendo utilizada para crear las primeras páginas web existentes en internet.

Como dicen sus siglas en inglés "HyperText Markup Language" o lenguaje de marcado de hipertexto, este no se consideraría como tal un lenguaje de programación, debido a que no posee lógica en su escritura, sino más bien define la estructura o lo que se le podría denominar el "esqueleto" del sitio web, en el que podemos escribir títulos, párrafos, insertar imágenes, mapas, etc.

A modo de síntesis, según menciona (Luján Mora, 2002, p.23), HTML está compuesto por una serie de etiquetas, lo cual permite definir el contenido y apariencia del sitio web.

CSS: Posterior a la creación de la web, en donde hasta el año 1991 solo se utilizaba el lenguaje HTML para poder dar formato a los sitios web. Según (Menéndez-

Barzanallana, s.f.) es que en el año 1996 se crea el CSS, con la necesidad de entregar estilos y formato a los sitios web sin tener extensas cantidades de etiquetas HTML para la construcción de una página web.

La sigla CSS proviene de su nombre en inglés "Cascade Style Sheets" u hoja de estilos en cascada. Al igual que el anteriormente mencionado HTML, este tampoco es denominado un lenguaje de programación, puesto que tampoco posee lógica en la sintaxis del código. La principal funcionalidad de CSS es definir los estilos del sitio web y pudiendo definir una paleta de colores para el desarrollo del sitio, además de definir tamaño de letras, separación de los bloques o las grillas.

JavaScript: Según indica (Menéndez-Barzanallana, s.f.) el lenguaje de programación JavaScript fue creado en el año 1995 por Brendan Eich el que en aquella época era trabajador de Netscape Communications Corporation, la cual poseía el navegador Netscape, principal competidor durante esa época de internet explorer creado por Microsoft.

JavaScript o su abreviación JS, es un lenguaje de programación interpretado, utilizado fuertemente en el desarrollo de sistemas de información web, el cuál por la misma característica de lenguaje interpretado mencionada previamente, permite que este lenguaje sea ejecutado directamente en el navegador sin necesidad de pasar por un compilador que interprete las líneas de código. Los principales usos a los que se destina el desarrollo de software utilizando el lenguaje de programación JavaScript son en sistemas web (Aporta funcionalidades e interacciones del usuario con el sistema), videojuegos, aplicaciones móviles, etc.

2.1.2.6 Backend

La capa del backend de un sistema de información, corresponde a la parte que no es visible por el usuario, donde se desarrolla la lógica de negocios del sistema, se almacenan datos, envían datos y además se alojan las bases de datos correspondientes al software. A diferencia del frontend, que es donde el usuario interactúa e incluso puede ver el código fuente de un sitio web a través del inspeccionador de elementos, en el backend el usuario no tiene acceso a este por

motivos de seguridad del sistema y que no se puedan realizar ataques maliciosos o cambiar datos del sistema.

Para el desarrollo de backend existen una gran variedad de lenguajes de programación y técnicas, dentro de las cuales encontramos las principales mencionadas a continuación:

PHP: Según (Achour Mehdi, Betz Friedhelm, Dovgal Antony, et al.2021) el lenguaje de programación PhP fue una evolución de un producto anteriormente creado llamado PHP/FI, el cuál posteriormente sirvió de inspiración para Rasmus Lerdorf en el año 1994 para finalmente crear como conocemos hoy en día a PHP, en un inicio creado para poder rastrear las visitas que obtenía su curriculum en la red.

Este lenguaje de programación es uno de los más antiguos para el desarrollo de sistemas en la parte del servidor, el cual permite entregarle características dinámicas a un sitio web, recopilando datos de formularios, contenidos multimedia dinámicos, recibir cookies, etc.

Python: Según indica (Venners Bill, 2003), el lenguaje de programación Python fue creado por Guido van Rossum a fines de los años 80, basándose fuertemente en el lenguaje de programación ABC, pero intentando que fuera mucho más simple en su comprensión y sintáxis.

Este lenguaje de programación es uno de los más versátiles y populares en la actualidad, siendo utilizado para diversos motivos en el desarrollo de software, en donde encontramos principalmente: Inteligencia artificial, desarrollo web, análisis de datos, etc.

JAVA: El lenguaje de programación Java es lanzado en 1995 en su versión beta, justo en una época en dónde aparece una nueva generación de lenguajes, los cuales apuntaban a distintas características y necesidades por parte de las empresas de tecnologías emergentes durante esos años. Inmediatamente después del lanzamiento oficial de Java, este se convirtió en uno de los más populares debido a

su extrema portabilidad y capacidad de ser ejecutado en múltiples plataformas, así pudiendo ser considerado una muy buena opción para ser utilizado en el desarrollo de aplicaciones de negocio. (Binstock Andrew, 2015.)

Java es uno de los lenguajes de programación más utilizados y populares entre las comunidades de desarrolladores y a diferencia de los lenguajes interpretados, necesita de un compilador o una máquina virtual para ser ejecutado. el cuál es destinado para usos como: Desarrollo de backend, aplicaciones móviles o aplicaciones de escritorio.

A modo de síntesis, se puede obtener una vista general sobre el uso de los distintos lenguajes de programación y cuáles de ellos son los más populares entre la comunidad de desarrolladores, como indica la figura a continuación:

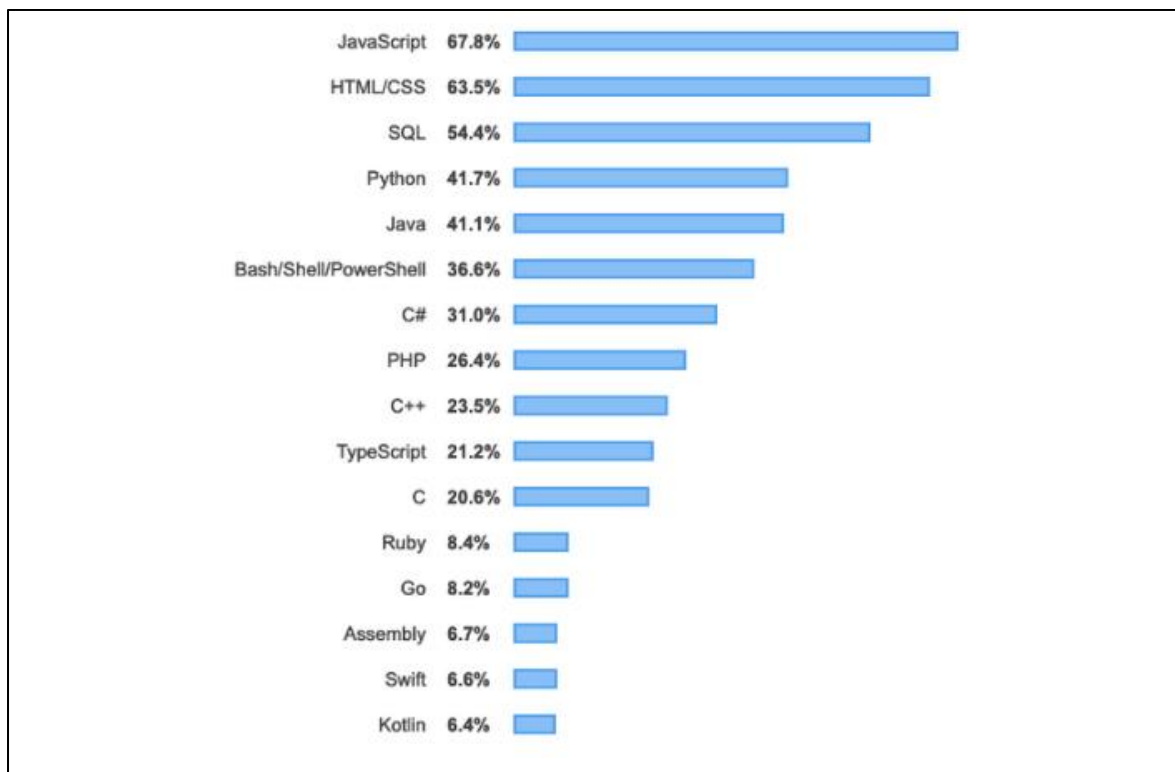


Figura 5 - The most popular programming languages based on the 2019 annual survey. Fuente: Aruleba Kehinde, Obaido George, et al. (2021)

2.1.2.5 Frameworks

Un framework o marco de trabajo, puede ser definido como un entorno de trabajo, en donde se entregan herramientas preestablecidas dependiendo de cada framework y de qué es lo que se querrá desarrollar, para así facilitar el desarrollo de un proyecto para los programadores y su equipo de trabajo.

Según Fayad, Schmidt (1997) un framework es una aplicación reutilizable que se encuentra semi completa con todas sus características y herramientas, la cuál es utilizada para el desarrollo de software personalizado.

Dado las grandes utilidades y disminución de tiempos que puede entregar la utilización de un framework, es que la mayoría de las aplicaciones y software que utilizamos diariamente están desarrollados sobre uno, es el caso de Uber que está construido con node.js, youtube con phoenix, Spotify con Chromium Embedded Framework, Instagram con Django, entre muchas más de las aplicaciones que utilizamos en nuestros smartphones o computadoras.

2.1.2.8 Tipos de Frameworks

Según Venkateswara (2011), los frameworks se componen de una amplia cantidad de componentes reutilizables, es por ello por lo que un framework es una gran inversión para las empresas de desarrollo de software por todas las ventajas que este trae. Dependiendo del objetivo del proyecto y para qué será utilizado el framework, existe una clasificación de estos:

Caja blanca: En este tipo de framework, la arquitectura de este es conocida por los desarrolladores y estos construyen sus aplicaciones sobre este framework.

Todo el diseño de la aplicación debe ser completamente documentado, ya que, a partir de esto, el framework puede ser adaptado a la necesidad específica para el desarrollo de una aplicación.

Caja negra: En los framework de caja negra la arquitectura de este se encuentra oculta para los desarrolladores, estos solamente conocen los puntos más importantes

de y los aspectos generales del mismo. Comúnmente, el mecanismo utilizado para entregar flexibilidad es la composición, ya que los desarrolladores solamente poseen un conocimiento general del framework y de su arquitectura, y simplemente se ve limitado a cuál será el conjunto de componentes que será implementado.

Caja gris: Los framework de caja gris son una fusión de los de caja negra y los de caja blanca, ofreciendo herencia como el mecanismo de composición, posee una caja blanca que es la encargada de entregar la interfaz y las clases abstractas que son las que entregan la arquitectura del framework.

El desarrollador tiene la posibilidad de crear clases personalizadas, heredando las clases abstractas o las clases concretas pertenecientes a cajas blancas o negras.

2.2 Arquitectura de Software

Según Bass, Clements, Kazman; Addison Wesley (2003) la arquitectura de software es “una vista del sistema que incluye los componentes principales del mismo”. La arquitectura de software se podría definir como un componente que integra el sistema, las propiedades propias a la misma, y cómo estos interactúan entre sí, para finalmente dar forma al software a desarrollar.

2.2.1 Patrones Arquitectónicos

Según Almeira & Pérez (2007), estos representan el nivel más alto dentro del sistema de patrones y expresan el esquema de la estructura fundamental de la organización para sistemas de software. Abastecen un conjunto de subsistemas predefinidos especificando sus responsabilidades e incluyendo reglas y guías para organizar las relaciones entre ellos. Cada patrón ayuda a lograr una propiedad específica del sistema global como es la adaptabilidad de la interfaz del usuario.

| Categoría de Patrones | | | |
|-----------------------|--------------------------------|------------------------|--|
| | Arquitectura | Diseño | |
| Según el Problema | Cimientos | Layers Pipes-Filter | |
| | Sistemas Distribuidos | Broker | |
| | Sistemas Interactivos | Model-View-Controller | |
| | Sistemas Adaptables | Microkernel Reflection | |
| | Creación | | Abstract Factory Prototype Builder |
| | Descomposición Estructural | | Whole Part Composite |
| | Organización del Trabajo | | Chain of Responsibility Command Mediator Master-Slave |
| | Control de Acceso | | Proxy, Facade, Iterator |
| | Variación de Servicios | | Bridge, Strategy, State |
| | Extensión de servicios | | Decorator Visitor |
| | Administración | | Memento |
| | Adaptación | | Adapter |
| | Comunicación | | Forwarder-Receiver Client-Dispatcher-Server Publisher-Subscriber |
| | Estructuración y Configuración | | Extension Interface |
| | Manejo de Recursos | | Flyweight |

Tabla 1 - Categoría de patrones según el tipo de problemas - Fuente (Arquitectura de Software: Estilos y Patrones, 2007)

2.2.2 Modelo - vista - controlador (MVC)

El patrón de diseño modelo - vista -controlador (MVC de ahora en adelante), divide una aplicación interactiva en 3 partes:

Modelo: Posee los datos básicos y las funcionalidades centrales del sistema

Vista: Despliega la información perteneciente al usuario, y se puede definir más de una vista.

Controlador: Los controladores son los encargados de las entradas realizadas por el usuario, y mantiene la conexión entre el modelo y la vista.

2.3 Adopción de sistemas de información en pymes

La información se ha convertido en un activo valioso para las organizaciones y ha representado gran importancia en el proceso de toma de decisiones asumiendo un papel predominante como insumo para los emprendedores. Es por esto que los sistemas de información han impulsado grandes cambios en la gestión de los datos haciendo a los procesos organizacionales más eficientes en entornos organizativos múltiples y en donde se perciben datos tan tremendamente heterogéneos. Existen

grandes desafíos para las organizaciones como lo son ajustar los mecanismos de captura, sistematización, almacenamiento y distribución para gestionar adecuadamente sus operaciones. (De Pablos Heredero, et al. 2012)

Muchas veces el mal manejo de estos grandes desafíos por parte de las Pymes se traduce en desajustes operativos que impiden el crecimiento óptimo de estas, además de las barreras que se presentan a continuación:

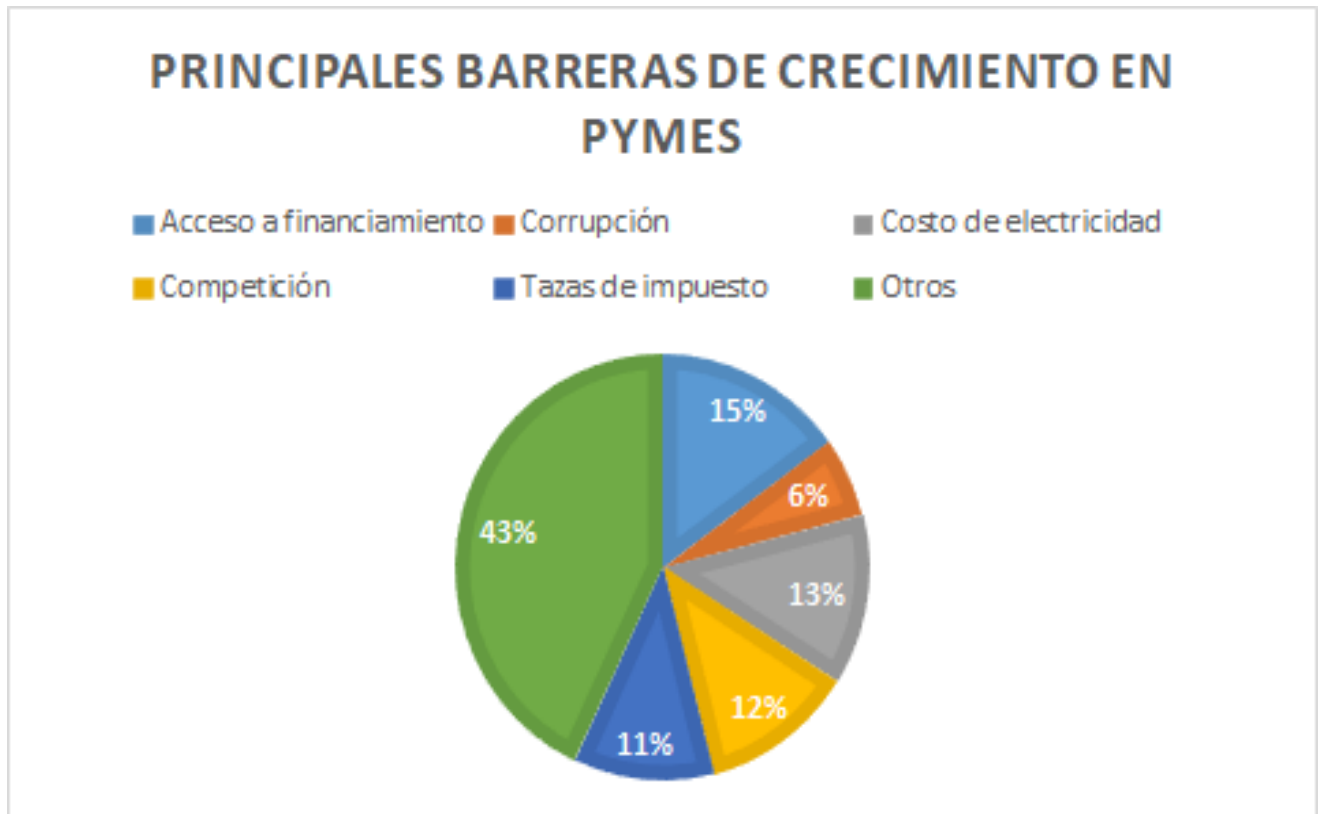


Figura 6 - The Main Barriers to Growth as Perceived by SMEs - Fuente: (Wang, 2018) – Adaptado por el autor

2.4 Sistemas de información para reserva de espacios de trabajo colaborativo

Las nuevas tendencias de trabajo como las jornadas flexibles o las posibilidades de trabajar desde lugares remotos permiten la irrupción en el mercado de nuevas formas de hacer negocios, promoviendo las interacciones entre los usuarios en espacios de trabajo adaptados a medida de los clientes que los ocupan. Los beneficios de este tipo de espacios son variados y generan impacto en el ecosistema emprendedor

debido a la comunidad que se crea en torno a ellos, en donde se reúnen usuarios pertenecientes a distintos tipos de disciplinas con el fin de prestar sus servicios a quien los necesite. Esto apoya al desarrollo económico y finalmente aporta a que muchas iniciativas que surgen como individuales se transforman en oportunidades para asociaciones, redes y apoyo mutuo dentro de la comunidad en general (Schuermann,2014)

Como lo describe M D Rahmatya et al 2020, el proceso de reservación de estos espacios generalmente requiere que el usuario visite el lugar, obtenga toda la información necesaria y pueda reservar de forma correcta. Estas tareas pueden hacerse de forma mucho más fácil a través de un sistema de reservas online que permita el agendamiento de estos espacios en cualquier momento y lugar, evitando las llamadas telefónicas innecesarias, pérdidas de tiempo por parte de los usuarios y administradores, y promoviendo un método rápido y eficiente de agendamiento sin la intervención de un intermediario para ejecutarlo.

En la figura 7 se muestra el caso de uso de la interacción que tienen los usuarios con el sistema comenzando con la visualización de la información sobre el servicio y su disponibilidad, luego, el registro de los datos de usuario para pasar a la reserva de la sala y finalmente la confirmación de la reserva con el pago de esta.

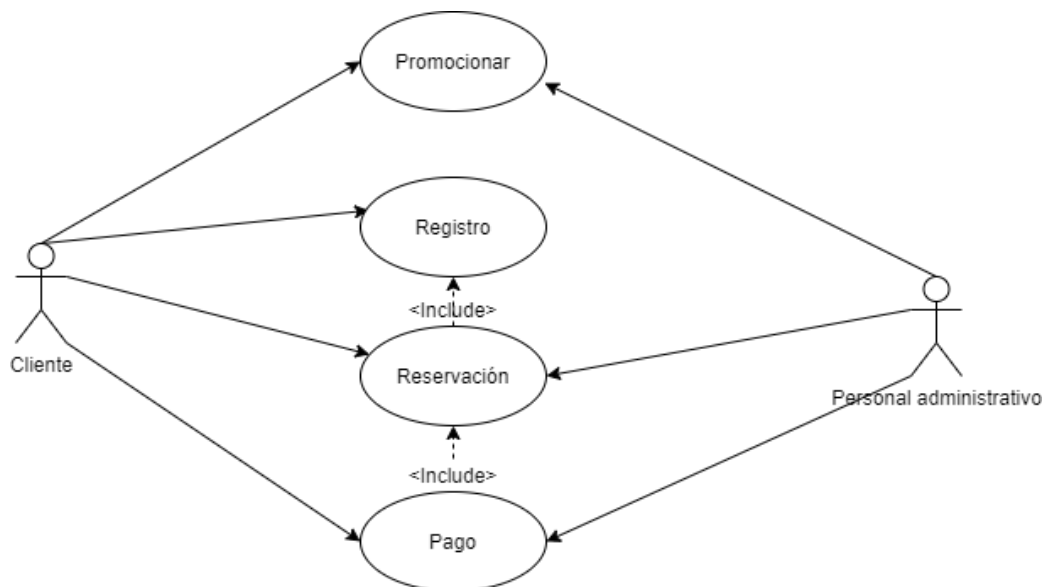


Figura 7 - Caso de uso - Fuente: (M D Rahmatya et al 2020) – Adaptado por el autor

2.5 Ingeniería de requerimientos

La técnica de la ingeniería de requerimientos es el procedimiento de recopilar, estudiar y corroborar las características que debe poseer un software para cumplir de manera precisa con las necesidades del cliente, y que estas logren satisfacer o resolver las problemáticas que presente la organización en cuestión. El proceso de ingeniería de requerimientos ha sido definido por una serie de autores a lo largo del tiempo, donde se define ingeniería de software según Leite (1978) “Es el proceso mediante el cual se intercambian diferentes puntos de vista para recopilar y modelar lo que el sistema va a realizar”.

2.5.1 Tipos de requerimientos

El proceso de ingeniería de requerimientos se realiza para entregar una descripción detallada de lo que el sistema puede o no hacer y qué necesidades va a satisfacer. Sommerville (1988), establece que existen los siguientes tipos de requerimientos:

- **Requerimientos NO funcionales:** Hace alusión a las características que posee el sistema desde el punto de vista ambiental o del software en sí, como, por ejemplo, rendimiento, tiempo de respuesta, disponibilidad, etc., mas no de las características o funcionalidades del software
- **Requerimientos Funcionales:** Al establecer los requerimientos funcionales de un software, se declara las funcionalidades que este debe poseer y cómo responderá ante ciertos eventos, y también es de suma importancia establecer lo que el software NO debe hacer.
- **Requerimientos de usuario:** descripción de los requerimientos funcionales y no funcionales en términos comprensibles para el usuario especificando el comportamiento externo del sistema y evitando el uso de lenguaje técnico y las características de diseño de sistema
- **Requerimientos del sistema:** versión extendida de los requerimientos de usuario utilizados por los ingenieros de software como punto de partida para el diseño del sistema

2.6 Metodologías de desarrollo de sistemas de información

Se define una metodología según Maida et al. (2015) como el conjunto de herramientas, métodos y técnicas que nos permite abordar de una misma manera cada una de las etapas de un proyecto. En el caso de las metodologías de desarrollo de software, estas nos brindan un modo sistemático de administrar nuestro proyecto, o como comúnmente se conoce, un marco de trabajo usado para estructurar, controlar y planificar el proceso de desarrollo de un sistema de información para así cumplir de la mejor forma con los objetivos establecidos, destacando su rol de optimizar tanto el proceso como el producto final, enfrentando el proyecto bajo la misma estrategia general.

Dentro de los elementos principales de una metodología de desarrollo de software podemos mencionar las tareas a realizar, las fases o conjunto de tareas previamente ordenadas según la naturaleza del proyecto y la prioridad de estas mismas, los productos entregables o documentos obtenidos asociados a cada fase, los procedimientos y tecnologías que brindan soporte a cada fase y, finalmente, los criterios de evaluación tanto de los procesos como de los productos para la medición del cumplimiento de los objetivos propuestos.

Dentro de este gran conjunto de metodologías de desarrollo podemos encontrar dos principales corrientes que se debaten entre lo prescriptivo, ordenado y estructurado, y lo ágil, dinámico y cambiante. Estas son las metodologías tradicionales y las metodologías ágiles, respectivamente, que se describen más detalladamente en los siguientes subcapítulos.

2.6.1 Metodologías de desarrollo de software tradicionales

Se consideran metodologías prescriptivas o metodologías tradicionales a aquellas que buscan establecer un orden y una estructura para el desarrollo de un sistema de información (Pressman, 2002). Estas metodologías generalmente se aplican a proyectos cuyas etapas están previamente definidas y son conocidas por quienes lo abordan, insertos en ambientes estáticos, con objetivos claramente definidos debido

a la poca flexibilidad que ofrecen a la hora de enfrentarse a cambios inesperados durante su construcción.

2.6.1.1 Rational unified process (RUP)

RUP se define según Kruchten (2001) como “un proceso de ingeniería de software que provee a la organización de desarrollo un acercamiento a la asignación de tareas y responsabilidades, cuyo objetivo es asegurar la más alta calidad en la producción de software”. Los equipos de desarrollo Rational se caracterizan por trabajar en conjunto con los usuarios finales, stakeholders e incluso consultores Rational para mantener el proceso de desarrollo continuamente actualizado. Este modelo cuenta con cuatro fases:

1. **Inception:** en esta fase se estudia a fondo el caso de negocio y se acota el alcance del proyecto. Se identifican los actores externos con los que el sistema va a interactuar y se definen los tipos de interacciones que estos tendrán. Teniendo en consideración lo anterior se verifica la factibilidad del proyecto para avanzar o no a la siguiente fase.
2. **Elaboración:** en esta fase se analiza a fondo el problema que busca solucionar el proyecto y se realiza el plan de trabajo para su desarrollo, tomando en cuenta los principales riesgos y buscando eliminarlos o reducirlos al máximo. Es la fase más crítica ya que determinará el desarrollo del proyecto.
3. **Construcción:** se desarrollan todas las características y aplicaciones del sistema integrándose al producto final a través de ciclos iterativos incrementales.
4. **Transición:** en esta fase se busca una transición desde el producto desarrollado a uno que use el cliente final en un entorno real de puesta en marcha. Se resuelven problemas finales que pudiesen surgir y se corrigen los errores adaptando la propuesta a las necesidades reales de los usuarios.

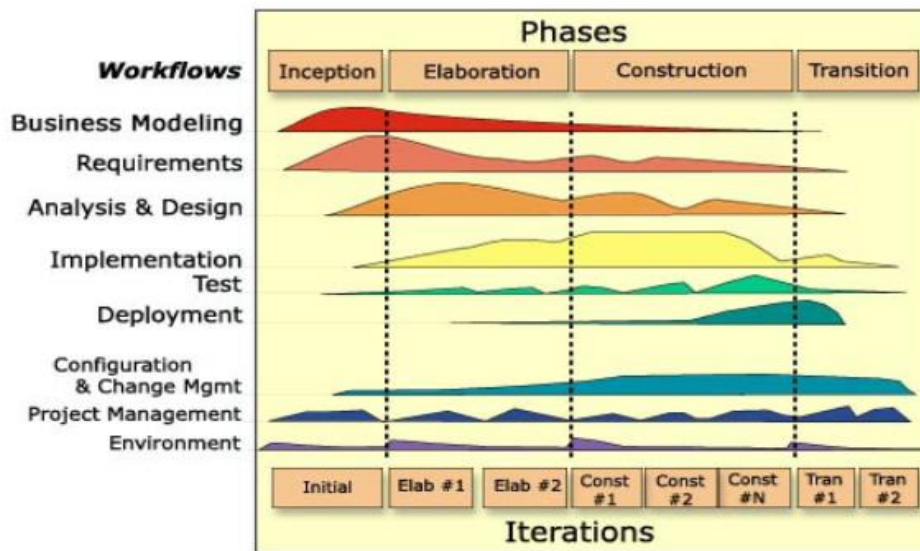


Figura 8 – Two dimensions of the RUP Fuente: (Kruchten, 2001)

2.6.1.2 Modelo de desarrollo en cascada

Denominada así por la forma en que se dispone y relaciona cada fase con el anterior ordenando de manera rigurosa sus etapas de forma que cada una de ellas deba esperar la finalización de la anterior para comenzar. Se realizan revisiones al finalizar cada fase para verificar que el proyecto cumple con lo necesario para avanzar a la siguiente. Según Sommerville (2002) las etapas de desarrollo de este modelo son las siguientes:

5. **Análisis y definición de requerimientos:** esta es una definición de los servicios, restricciones y metas del sistema mediante la estrecha colaboración de quienes lo usarán.
6. **Diseño del sistema y del software:** en esta etapa se dividen los requerimientos en sistemas hardware o software.
7. **Implementación y prueba de unidades:** se lleva a cabo el diseño del software como un conjunto o unidades de

programas. Se verifica que cada unidad cumpla su especificación.

8. **Integración y pruebas de sistema:** se integran las unidades de programa de la fase anterior y se prueban como un sistema completo. Después de las pruebas se entrega el software al cliente.
9. **Funcionamiento y mantenimiento:** el software se instala y pone en funcionamiento. Se corrigen errores no descubiertos en las etapas anteriores.

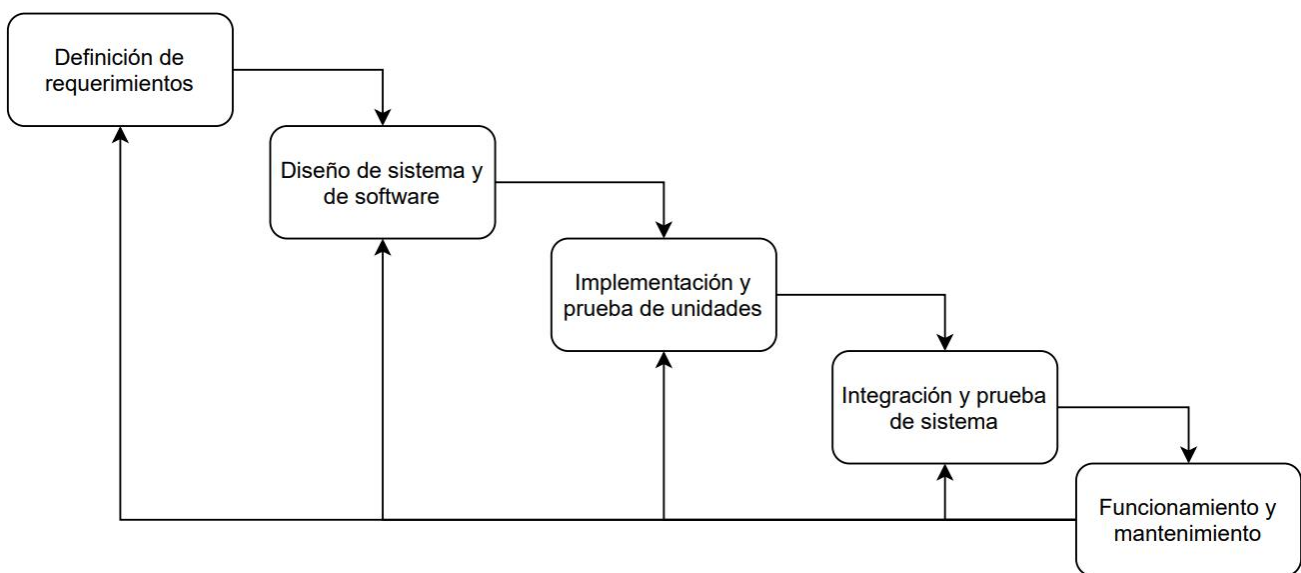


Figura 9 – Ciclo de vida del software. Fuente Sommerville 2002 – adaptado por el autor

2.6.1.3 Modelo de desarrollo evolutivo

Este modelo se basa en desarrollar una implementación inicial la cual va siendo refinada según los requerimientos del usuario mediante la revisión de versiones actualizadas mostradas por el equipo desarrollador. Existe una rápida retroalimentación entre las etapas de este tipo de desarrollo de sistemas de información, las cuales se unifican en lugar de separarse, tal como se aprecia en la figura 7.

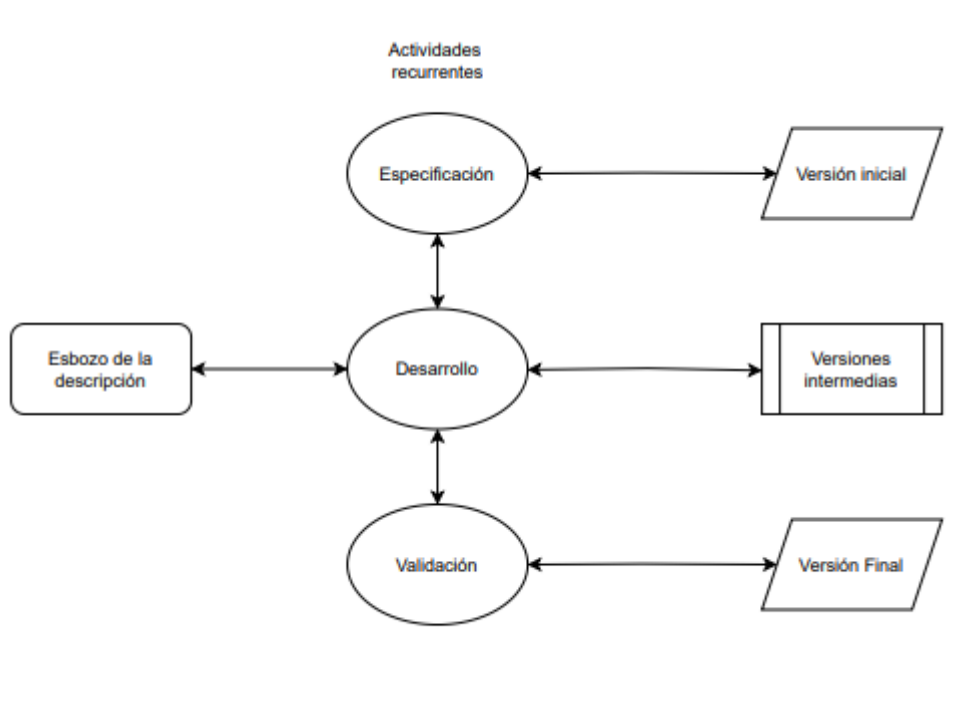


Figura 10 – Desarrollo evolutivo. Fuente: Sommerville 2002 – adaptado por el autor

2.6.1.4 Desarrollo basado en componentes

Para Sommerville (2002), este modelo de desarrollo denominado CBSE por sus siglas en inglés (component-based software engineering) se basa en la reutilización de código, modificando componentes preexistentes e incorporándose al sistema de forma que se adapten a los requerimientos de este. Este enfoque abarca una gran cantidad de componentes que aportan al sistema funcionalidades específicas, comúnmente estimaciones estadísticas que siguen el proceso genérico que se detalla en la figura 8 y sus etapas son las siguientes:

1. Análisis de componentes
2. Modificación de requerimientos
3. Diseño del sistema con reutilización
4. Desarrollo e integración

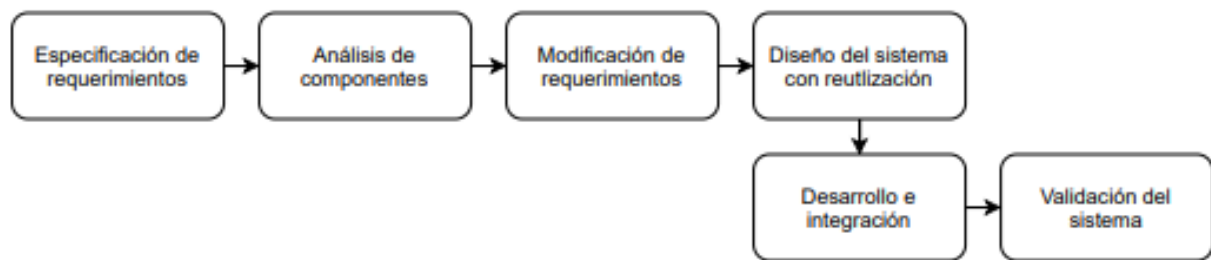


Figura 11 – Ingeniería del software basado en componentes – Fuente: Sommerville 2002 – adaptado por el autor

2.6.2 Metodologías de desarrollo de software ágiles

Durante los últimos años en el mundo del desarrollo de aplicaciones y sistemas de información, los equipos de desarrollo han mostrado un fuerte interés en la utilización de metodologías ágiles para el transcurso de sus proyectos, las cuales entregan herramientas que permiten mantener una buena comunicación entre el equipo y el cliente final del producto. Las metodologías ágiles surgen como una alternativa a las metodologías tradicionales, puesto que surge la necesidad para los desarrolladores de no siempre esperar hasta la finalización de un producto para que el cliente pudiera ver los resultados, puesto que muchas veces si el cliente no quedaba satisfecho con el producto final, hacer ajustes resultaba incluso más caro que haberlos realizado en etapas tempranas (Como si se puede realizar bajo la utilización de metodologías ágiles), además que permite el desarrollo de software en un menor tiempo sin perder la calidad del mismo. (Uribe, 2007)

El manifiesto ágil destaca sus 4 valores, los cuales deben prestar apoyo en el desarrollo de software (Herrera, 2007):

- **Individuos e interacciones sobre procesos y herramientas:** Para garantizar la buena calidad del producto final, los individuos que integran el equipo de trabajo, sus habilidades técnicas y de comunicación son fundamentales, incluso más que las herramientas o procesos que se puedan aplicar al desarrollo del proyecto.
- **Software funcionando sobre documentación extensiva:** Las metodologías ágiles valoran la intuitividad y comprensión de la documentación de un producto final, la cuál debe ser comprendida por el cliente.
- **Colaboración con el cliente sobre negociación contractual:** Durante el desarrollo de la metodología ágil es importante el diálogo con el cliente durante el desarrollo del producto, dado que refuerza el entendimiento del problema, la retroalimentación de lo desarrollado a través de cada etapa del proyecto y el cumplimiento de los requerimientos y características que necesita el cliente para solucionar la problemática de su negocio.
- **Respuesta ante el cambio sobre seguir un plan:** Una de las principales características que poseen las metodologías ágiles es la adaptación ante los cambios o imprevistos que puedan surgir a lo largo del proyecto.

Existen características que diferencian a las metodologías ágiles de las tradicionales, las cuales forman parte del eje central del desarrollo de software utilizando estas metodologías, las cuales se encuentran dadas por los 12 principios del manifiesto ágil. “Manifiesto for Agile Software Development” (Beedle et al.,2001):

- **“Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor”:** De las características más importantes atribuibles a las metodologías ágiles es la entrega de software funcional que permita al cliente entregar retroalimentación a los

desarrolladores, lo cual disminuye la incertidumbre y permite cumplir a cabalidad con los requerimientos del cliente.

- **“Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente”**: Es poco probable que un cliente entregue de manera concluyente todos los requerimientos del sistema a desarrollar, y una característica importante de las metodologías ágiles es adaptarse a las necesidades del cliente a medida que este requiera ciertas características y los requisitos cambien a medida que las etapas del proyecto vayan transcurriendo.
- **“Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible”**: La entrega de software funcional es de suma importancia para que el cliente pueda probar las características que solicitó en el software y así disminuir su incertidumbre ante las largas esperas que a veces supone para el cliente ver características funcionales del software cuando se utilizan metodologías tradicionales.
- **“Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto”**: El usuario del producto debe estar en todo momento en comunicación el equipo de desarrollo, puesto que, a pesar de no poseer conocimientos técnicos, es importante que entregue retroalimentación sobre las funcionalidades del software en funcionamiento y así estar seguro de que el equipo está por buen camino.
- **“Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo”**: El clima de trabajo y mantener un equipo de desarrollo bien motivado es una de las claves del éxito para alcanzar un producto de calidad y satisfacer al usuario final.

- **“El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara”**: Para un buen trabajo en equipo, uno de los aspectos más importantes es la comunicación y que no existan intermediarios entre estos, ya sea una llamada telefónica, correo electrónico, WhatsApp, etc. Al existir una comunicación cara a cara se aumentan las posibilidades de que el mensaje al receptor sea más efectivo y entendible.
- **“El software funcionando es la medida principal de progreso”**: Entregar pruebas funcionales del producto y recibir respuestas positivas por parte del usuario indica que el equipo de desarrollo va por buen camino.
- **“Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida”**: Las tareas designadas siempre deberán ser cumplidas por el actor al que se le han sido indicadas sin aplazarlas, puesto que entorpece el ritmo de trabajo y no ayuda al cumplimiento de los objetivos establecidos.
- **“La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad”**: La agilidad se ve mejorada cuando además del cumplimiento de los requerimientos del usuario, el software posee aspectos técnicos de excelencia en cuanto a rendimiento, tiempos de respuesta, mantenimiento, diseño, etc.
- **“La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial”**: Si una funcionalidad u optimización del sistema no ha sido solicitada por el usuario, solamente se deben centrar los esfuerzos en lo que realmente importa y si ha sido solicitado por el, sin entrar en costos innecesarios en “perfeccionar” el sistema.
- **“Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados”**: Siempre se debe velar porque se mantenga la participación de todo el equipo en las decisiones o cambios que se realicen durante el desarrollo del proyecto.

- **“A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia”:** A medida que se reciben comentarios por parte del cliente, esto permite reducir el nivel de error y que el equipo de desarrollo pueda ir mejorando a medida que avancen las etapas del proyecto, recibiendo críticas por parte de los líderes del proyecto o del mismo usuario para poder hacerlo mejor en futuras instancias de este.

2.6.2.1 SCRUM

Tal y como lo declaran Schwaber & Sutherland (2017), Scrum es un marco de trabajo por el cual las personas pueden abordar problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente.

Por otro lado, Trigas () destaca que este marco de trabajo está ideado para trabajar en entornos que se caracterizan por tener incertidumbre (planteándose objetivos sin un plan detallado del producto), autoorganización (equipos autogestionados que carecen de roles definidos), control moderado (promoviendo la creatividad y espontaneidad de los miembros del equipo) y transmisión de conocimientos.

El desarrollo se estructura en ciclos de trabajo iterativos llamados sprints, los que según Schwaber & Sutherland (2017) son “el corazón de Scrum”. Estos mismos autores lo definen como un bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento de producto “Terminado” utilizable y potencialmente desplegable. Los sprints contienen y consisten en eventos como la Planificación del Sprint (Sprint Planning), los Scrums Diarios (Daily Scrums), el trabajo de desarrollo, la Revisión del Sprint (Sprint Review), y la Retrospectiva del Sprint (Sprint Retrospective).

En cuanto a los equipos de Scrum Schwaber & Sutherland (2017) remarcan que son equipos cross-funcional de alrededor de siete personas, en los que podemos destacar los siguientes roles:

- **Dueño del producto o Product Owner:** es el responsable de maximizar el valor del producto resultante del trabajo del Equipo de Desarrollo. También es la única persona responsable de gestionar la Lista del Producto (Product Backlog)

- **Equipo de desarrollo:** consiste en los profesionales que realizan el trabajo de entregar un Incremento de producto “Terminado” que potencialmente se pueda poner en producción al final de cada Sprint.
- **Scrum Máster:** es responsable de promover y apoyar Scrum como se define en la Guía de Scrum. Los Scrum Masters hacen esto ayudando a todos a entender la teoría, prácticas, reglas y valores de Scrum.

Además, dentro de la misma guía se mencionan los artefactos de Scrum diseñados específicamente para maximizar la transparencia de la información y así asegurar de que todo el equipo tenga el mismo entendimiento de estos. Los artefactos mencionados son los siguientes:

- **Lista de producto o Product Backlog:** es una lista ordenada de todo lo que se conoce que es necesario en el producto. Es la única fuente de requisitos para cualquier cambio a realizarse en el producto. (p.15)
- **Lista de pendientes del Sprint o Sprint Backlog:** es el conjunto de elementos de la Lista de Producto seleccionados para el Sprint, más un plan para entregar el Incremento de producto y conseguir el Objetivo del Sprint (p.16)
- **Incremento:** es la suma de todos los elementos de la Lista de Producto completados durante un Sprint y el valor de los incrementos de todos los Sprints anteriores (p.17)

La figura que se muestra a continuación es una representación gráfica de la visión general de este marco de trabajo en donde se muestran resumidos los principales roles, artefactos y eventos.

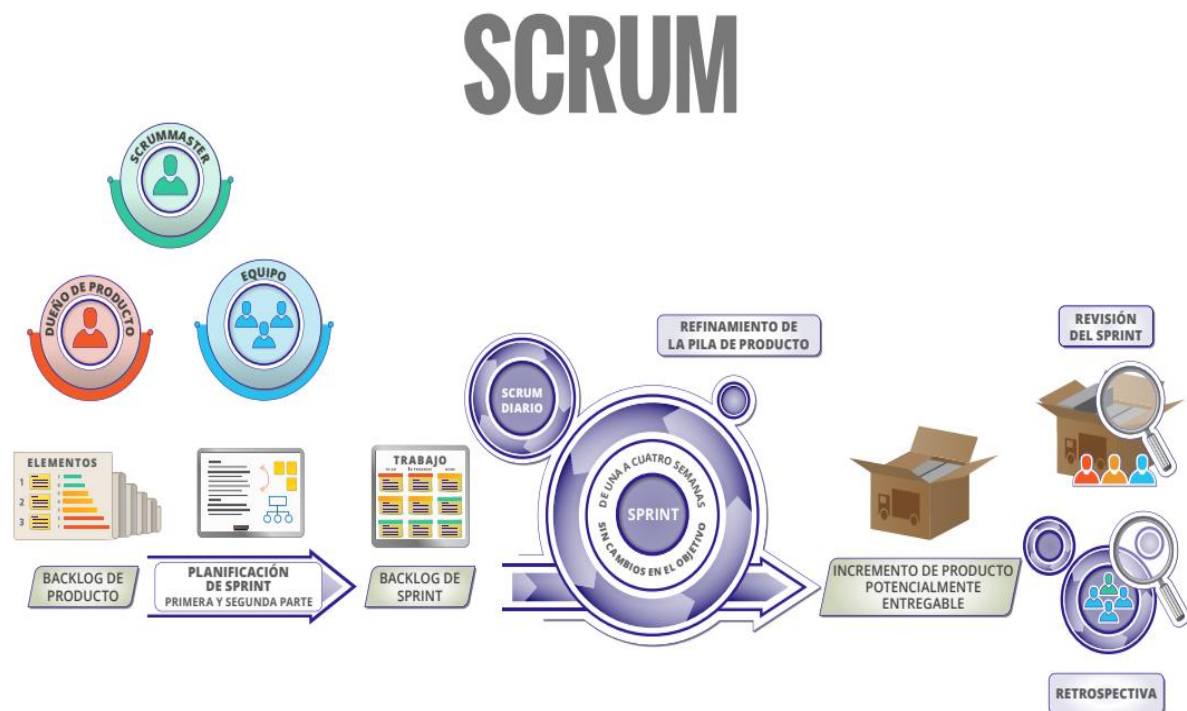


Figura 12 – Visión General de Scrum - Fuente: (Deemer et al., 2012)

2.6.2.2 Programación Extrema (XP)

Dentro de este modelo de desarrollo se utiliza un “enfoque orientado a objetos como paradigma preferido de desarrollo, englobando un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades estructurales” (Pressman 2002, p.62). Estas actividades son las siguientes:

1. Planeación: durante esta actividad, el equipo de desarrollo escucha al cliente para capturar los requerimientos de sistema necesarios. El cliente escribe las denominadas historias de usuario, que son características o funcionalidades del software que se requieren para el proyecto. Al mismo tiempo, el cliente les asigna un valor de prioridad a cada historia y el equipo de desarrollo les asigna un costo medido en semanas de trabajo, para así

administrar el tiempo de desarrollo y hacerlo más eficiente según las necesidades específicas del cliente.

2. Diseño y rediseño: XP busca mantener un diseño sencillo siguiendo el principio MS (mantenlo sencillo) implementando el uso de tarjetas CRC (clase-responsabilidad-colaborador) para registrar y administrar de mejor manera las clases orientadas a los objetos que corresponden a cada incremento del software. En lo que respecta al rediseño, se busca reducir errores haciendo más eficiente la estructura interna del código sin alterar su funcionamiento externo. De esta forma se limpia y mejora el diseño del código después de haber sido escrito.
3. Codificación: una vez definidas las historias de usuario se comienzan a practicar pruebas unitarias a cada una de ellas antes de iniciar la codificación, de esta forma los desarrolladores se enfocan en mejorar lo que impide pasar dichas pruebas. Una de las claves de la XP es la programación en parejas de forma que los problemas se puedan resolver en tiempo real. Se asumen roles distintos dentro de una misma estación de trabajo para asegurar la calidad del código y minimizar los errores.
4. Pruebas: se busca automatizar las pruebas creadas anteriormente para que estas puedan ser ejecutadas frecuentemente, de manera fácil y sencilla de forma que las pruebas de integración y validación puedan ser aplicadas diariamente con el objetivo de corregir pequeños errores en pequeños espacios de tiempo y no grandes errores cerca de un plazo de entrega límite.

2.6.2.3 Método de desarrollo de sistemas dinámicos (MDSD)

El MDSD busca entregar una estructura de desarrollo y mantenimiento a proyectos insertos en ambientes dinámicos y cuyo plazo de ejecución se encuentra acotado. Este proceso iterativo sigue una regla inspirada en la de Pareto en la que el 80 por ciento del trabajo se puede realizar en el 20 por ciento del tiempo que requiere entregar el trabajo completo. De esta forma se promueve un ambiente de desarrollo dinámico, cuyas iteraciones se completan parcialmente con lo justo y necesario para dar paso a la siguiente iteración dejando los detalles pendientes para cuando se conozcan por completo los requerimientos del negocio y el cliente haya pedido cambios en el sistema. (Pressman, 2002).

Este proceso consta de tres ciclos iterativos, seguidos de dos actividades que conforman el ciclo de vida del MDSD:

1. **Estudio de factibilidad:** se establecen los requerimientos del negocio y se evalúa si es posible aplicar el proceso MDSD.
2. **Estudio de negocio:** se definen requerimientos y la arquitectura de la aplicación para maximizar el valor que esta pueda entregar al negocio.
3. **Iteración de modelo funcional:** se entregan al cliente prototipos funcionales para que este los teste y realice feedback al equipo de desarrollo recabando requerimientos adicionales en cada iteración
4. **Diseño e iteración de la construcción:** se revisan los prototipos construidos para garantizar que estos den valor operativo del negocio a los usuarios finales
5. **Implementación:** se pone en operación el prototipo más reciente

del software. En caso de que se requieran cambios, se vuelve a la actividad de iteración de modelo funcional

2.6.2.4 Desarrollo adaptativo de software (DAS)

Según Pressman (2002), este método de desarrollo se centra en la colaboración humana y en la organización propia del equipo de trabajo, en el que se define un ciclo de vida que incluye tres fases:

1. **Especulación:** en esta fase se realiza la planeación adaptativa del proyecto en donde se definen principalmente las restricciones de este y sus requerimientos básicos, teniendo en cuenta siempre que por más previsor que sea el plan, será inevitable que este cambie durante el transcurso del proyecto.
2. **Colaboración:** la colaboración humana es el foco principal debido a que esta motiva la creatividad por sobre los números absolutos multiplicando el talento del equipo. La comunicación y la confianza entre los miembros de equipos colaborativos será de vital importancia para su óptimo funcionamiento.
3. **Aprendizaje:** el aprendizaje ayuda a mejorar el nivel de comprensión por parte del equipo sobre el proyecto, el proceso y las tecnologías empleadas para su desarrollo, aprendiendo a través de grupos de enfoque, revisiones técnicas y análisis post mortem del proyecto.

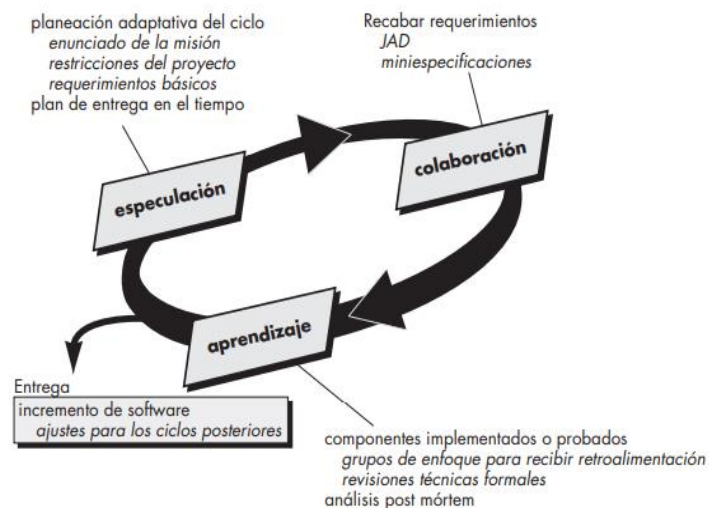


Figura 13 – Desarrollo adaptativo de software – Fuente: Pressman 2002 (p.68)

2.6.2.5 Kanban

Según Castellano (2019), la metodología de origen japonés Kanban, cuyo significado es “tarjeta” busca un control de materiales para así conseguir que el inventario del producto semiterminado recorra toda la cadena de suministros, desde los proveedores hasta los clientes mediante la creación de tarjetas organizadas por estado para ser visualizadas dentro de un tablero (ver figura 11).

Los seis principios en los que se basa el sistema Kanban son los siguientes:

1. Visualización: a través de la creación de un tablero en donde se organizan las tarjetas, se puede ver el estado de las tareas facilitando el trabajo al equipo
2. Calidad: se promueve un flujo de trabajo y mejora continua
3. Disminución de desperdicios: se detallan tareas, responsables, fechas límite, etc, para evitar contratiempos
4. Priorización/flexibilidad: se agregan tareas priorizadas al feed para facilitar el trabajo
5. En proceso: se promueve la modificación de actividades según las necesidades del proyecto
6. Mejora continua: se busca una mejora infinita de los procesos

Si bien esta metodología nace para ser implementada en operaciones de la industria manufacturera, sus bases se aplican hoy en día para el desarrollo de servicios tecnológicos, desarrollo de softwares, entre muchos otros a través de una gran variedad de herramientas ampliamente conocidas y utilizadas en entornos TI como Asana, Trello o Jira.

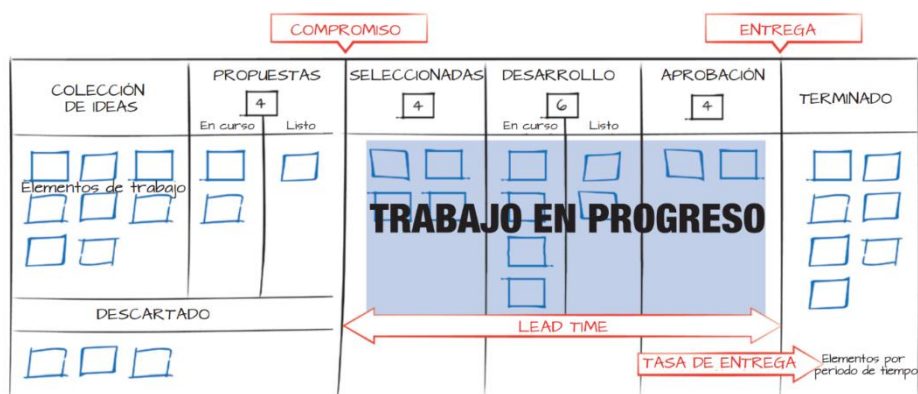


Figura 14 – Tablero Kanban – Fuente: Anderson et al., 2016

2.6.3 Metodologías Ágiles VS Metodologías Tradicionales

Existe una gran disyuntiva al intentar comparar la agilidad con las metodologías tradicionales de desarrollo de software principalmente porque no existe un enfoque que definitivamente sea mejor que otro. Si bien las metodologías ágiles han crecido en adeptos en distintas organizaciones actualmente y se logra preciar un auge de herramientas que gestionan equipos de desarrollo ágiles (Trello, Jira, Asana, etc), estos enfoques que buscan flexibilizar tareas y adaptarlas a entornos dinámicos, están estrechamente relacionados con la naturaleza de los proyectos que abordan. Es por esto por lo que la elección de la metodología de desarrollo a ocupar estará delimitada a la naturaleza intrínseca de cada proyecto TI, buscando siempre adaptar una metodología al proyecto y no adaptar el proyecto a una metodología en específico.

Tal como plantea Pressman (2013), las metodologías tradicionales buscaban poner orden en el caos de desarrollo de software en la década de los 60, dando estructura al trabajo de ingeniería de ese entonces. Por otra parte, las metodologías ágiles para Montero et al. (2018) surgen con el objetivo de “esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto”.

| Metodologías Ágiles | Metodologías Tradicionales |
|--|--|
| Basadas en heurísticas provenientes de prácticas de producción de código | Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo |
| Especialmente preparados para cambios durante el proyecto | Cierta resistencia a cambios |
| Impuestas internamente (por el equipo) | Impuestas externamente |
| Proceso menos controlado, con pocos principios | Proceso mucho más controlado, con numerosas políticas/normas |
| No existe contrato tradicional o al menos es bastante flexible | Existe un contrato prefijado |
| El cliente es parte del equipo de desarrollo | El cliente interactúa con el equipo de desarrollo mediante reuniones |

| | |
|---|--|
| Grupos pequeños (<10 personas) y trabajando en el mismo sitio | Grupos grandes y posiblemente distribuidos |
| Pocos artefactos | Más artefactos |
| Pocos roles | Más roles |
| Menos énfasis en la arquitectura de software | La arquitectura de software es esencial y se presenta mediante modelos |

Tabla 2 -Diferencias entre metodologías ágiles y tradicionales – Fuente: Letelier et al. (2003) adaptado por el autor

3. Metodologías por ocupar

Para la definición de la estructura general y el desarrollo de este proyecto se utilizó la metodología RUP (Rational unified process) principalmente debido a que esta se adapta mejor a las necesidades del proyecto y el equipo de desarrollo, proporcionando un ambiente de trabajo debidamente estructurado y una cercanía para el trabajo colaborativo con la contraparte del proyecto. De manera paralela se utilizó la metodología Kanban ya que esta proporciona un flujo de trabajo que da un orden priorizado a las actividades que componen cada fase del desarrollo de este proyecto.

Se descartaron las siguientes metodologías:

1. **Metodología cascada:** define una estructura demasiado rígida para el desarrollo de software obstaculizando el flujo continuo de las tareas y actividades en el paso de una fase a otra.
2. **Desarrollo evolutivo:** debido a la naturaleza estructurada y a plazo fijo del proyecto se descarta esta metodología, que se utiliza generalmente en proyectos sin fechas límite de entrega o deadlines.
3. **Desarrollo basado en componentes:** debido a que se trata de un proyecto completamente nuevo se descarta esta metodología que se basa en la reutilización de código o componentes.
4. **Desarrollo adaptativo (DAS):** el proyecto no cumple los requisitos para ser abordado por esta metodología que se centra en el desarrollo de sistemas complejos.
5. **Desarrollo de sistemas dinámico (MDSD):** se descarta debido al plazo del proyecto ya que en esta metodología los plazos de ejecución son demasiado acotados, además de dejar correcciones importantes para la última fase, acumulándose tareas en el flujo de trabajo.
6. **Programación extrema (XP):** se descarta debido su enfoque orientado a objetos y la implementación de tarjetas CRC en su fase de diseño, lo que no se adapta a las necesidades del equipo de desarrollo debido a que complejiza dicha fase.

7. **Scrum:** a pesar de que era una de las metodologías preferidas para este proyecto en un comienzo, se descarta debido a que el equipo no contaba con los miembros mínimos necesarios para su correcta implementación.

3.1 Aplicación de las metodologías

3.1.1 Adaptación de RUP como metodología en la estructura general del proyecto

Para aplicar este marco de trabajo al desarrollo de este proyecto se toman en cuenta las fases que se definen a continuación:

1. Inception: en esta fase se busca comprender el modelo de negocios presentado por la contraparte, sus principales dificultades a la hora de interactuar con sus clientes y cómo poder resolver dichos problemas. Se realizan reuniones vía Zoom entre la contraparte y el equipo de desarrollo con el fin de comprender a totalidad el funcionamiento actual del negocio, sus limitantes y comenzar a definir el problema que se abordará en el desarrollo del proyecto.

| Objetivo de la reunión | Fecha | Participantes |
|---|-------|-------------------------------------|
| Descripción detallada del negocio | 20/04 | Contraparte, Equipo de desarrollo |
| Análisis del proyecto y alcance | 04/05 | Equipo de desarrollo, profesor guía |
| Definición de actores y casos de uso | 11/05 | Equipo de desarrollo |
| Definición de arquitectura de sistema | 18/05 | Equipo de desarrollo, profesor guía |
| Propuesta inicial y feedback de la contraparte | 25/05 | Equipo de desarrollo, contraparte |

Tabla 3 – Tabla de reuniones fase Inception – elaborada por el autor (ver detalle en anexo 1)

2. Elaboración: en esta fase el foco se encuentra en el análisis completo del problema, establecer una arquitectura sólida para el proyecto, eliminar la mayor cantidad de riesgos (con especial atención en las historias de usuario más importantes) y definir un plan de trabajo estimando plazos de entrega y actividades. Para ello se llevarán a cabo reuniones de trabajo, las cuales se detallan en la siguiente tabla:

| Objetivo de la reunión | Fecha | Participantes |
|--|-------|----------------------|
| Revisión de historias de usuario críticas | 08/06 | Equipo de desarrollo |
| Identificación de requerimientos adicionales | 15/06 | Equipo de desarrollo |
| Elaboración del diseño del modelo | 22/06 | Equipo de desarrollo |

Tabla 4 – Tabla de reuniones fase Elaboración – elaborada por el autor (ver detalle en anexo 1)

3. Construcción: el objetivo principal de esta fase es alcanzar la capacidad operacional del portal de manera incremental mediante iteraciones. Para ello se deben implementar todas las características y componentes, integrarlas y probarlas en su totalidad. Para este proyecto se utilizarán conceptos de agilidad como lo son los “sprints” o ciclos de desarrollo iterativos e incrementales, los cuales se definen en la siguiente tabla:

| # Sprint | Objetivo |
|----------|---|
| #1 | Diseño de la arquitectura frontend del proyecto |
| #2 | Desarrollo de la página principal del proyecto |
| #3 | Creación de perfil de usuario y sistema de reserva de salas |
| #4 | Integración de planes y métodos de pago online |

Tabla 5 – Tabla de reuniones fase Construcción – elaborada por el autor

4. Fase de transición: una vez terminada la fase de construcción procede la transición del producto a manos del cliente. Para facilitar este proceso, esta etapa se plantea los siguientes objetivos:
1. Entregar un producto final que cumpla con los requisitos propuestos en su concepción
 2. Conseguir que el usuario interactúe con el portal de manera autónoma

Para conseguir estos objetivos se utilizarán las siguientes estrategias:

| #Objetivo | Estrategia |
|-----------|---|
| #1 | Pruebas de la versión Beta para validar el sistema |
| #1 | Reuniones de retroalimentación con la contraparte |
| #1 | Encuesta de satisfacción del producto por parte de clientes y usuarios |
| #2 | Capacitación del personal en el uso del portal y su administración |
| #2 | Entrega del producto documentado y actualizado al personal de marketing y ventas para la difusión de su uso |

Tabla 6 – Estrategias según objetivo – elaborada por el autor

3.1.2 Tablero Kanban

Para la gestión de tareas y actividades principales durante el desarrollo de este proyecto se analizó y decidió hacer uso de una de las principales herramientas de la metodología Kanban, la cual consiste en un tablero en donde se listan las actividades según su estado, ya sean pendientes, en proceso, revisión QA o terminado (ver detalle en capítulo 2.6.2.5) debido a que entrega un flujo de trabajo continuo en donde se agregan tareas o tarjetas nuevas de manera fácil y detallada, además de entregar información clara al equipo de desarrollo que puede ver el estado del proyecto con solo mirar el tablero.

Se utilizará la herramienta Trello para la gestión del proyecto debido a que se encuentra familiarizada a los integrantes del equipo de desarrollo, además de ser una herramienta de libre acceso y fácil uso.

4. Resultados

En el proceso de desarrollo de este proyecto y para definir su estructura general, se utilizó la metodología tradicional Rational Unified Process en conjunto con metodologías, herramientas y conceptos atribuidos al desarrollo ágil, adaptados a la naturaleza del proyecto y sus necesidades.

A continuación, se presentan los resultados de las distintas fases descritas en el capítulo anterior:

4.1 Fase de Inception

Durante esta fase se buscó interiorizar al equipo de desarrollo de las actividades de negocio y sus principales problemáticas a través de reuniones con la contraparte. Se elabora un User Story Map e Impact Mapping para plasmar de forma gráfica la temática a abordar. Para ello se realiza una sesión en conjunto con la contraparte en donde se discuten las bases del proyecto.

Productos de la etapa:

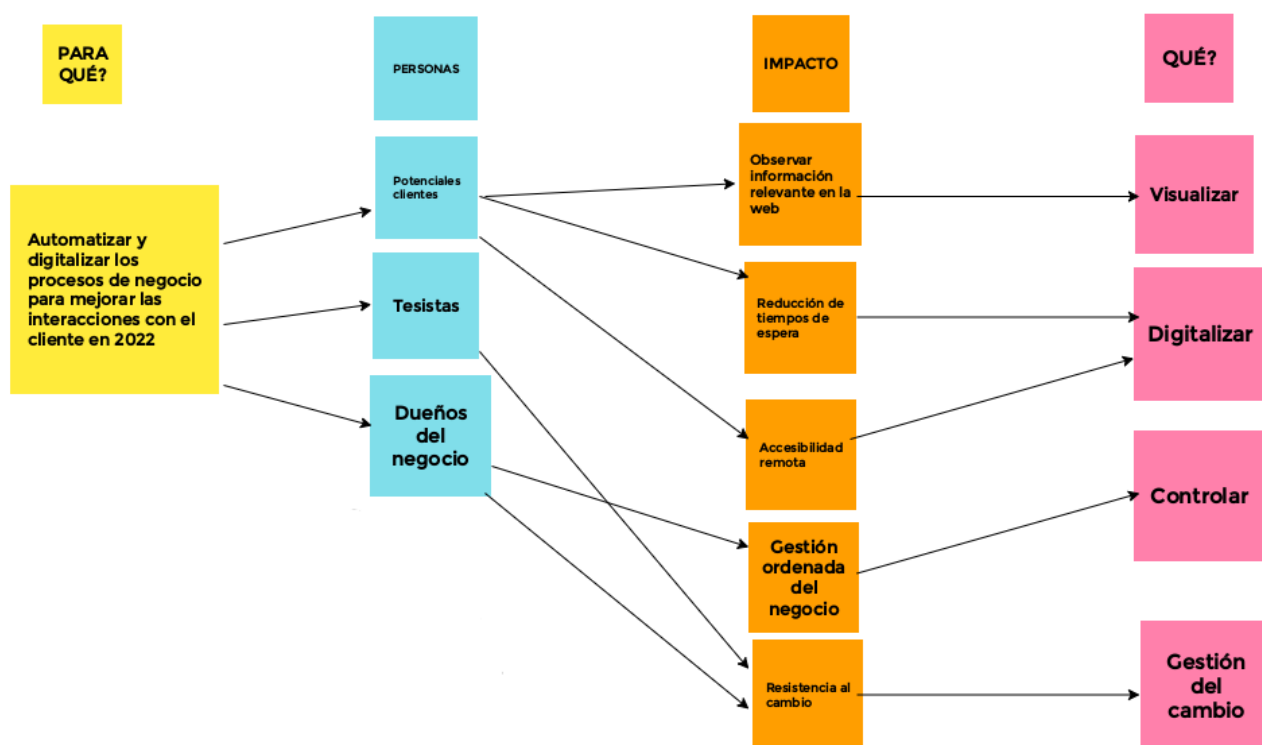


Figura 15 – Impact Mapping- Fuente: elaboración propia (2021)

- 1) Se define un “¿PARA QUÉ?” del proyecto en conjunto con la contraparte, este debe ser del tipo S.M.A.R.T., es decir, específico, medible, alcanzable, relevante y temporal (por sus siglas en inglés). Para ello se realiza una lluvia de ideas destacando las mas precisas en cuanto a la definición del propósito del proyecto y de los objetivos de la empresa, llegando finalmente a “Automatizar y digitalizar los procesos de negocio para mejorar las interacciones con los clientes en 2022” debido a que es la definición mas acertada de lo que se busca con el desarrollo de este proyecto.
- 2) Se define en la sección “PERSONAS” a quienes el proyecto impactará y, además, a quienes impactan el proyecto, tal y como se aprecia en la figura.
- 3) Se definen los principales impactos que tendrá el proyecto en las personas anteriormente definidas, y los principales impactos que estas personas tendrán en el proyecto, ya sean positivos o negativos.
- 4) Se definen los “¿QUÉ?” para dar respuesta a los impactos definidos anteriormente. Estas acciones garantizan que el proyecto impactará a las personas de la forma en la que se espera para el cumplimiento del objetivo principal o “¿PARA QUÉ?”.

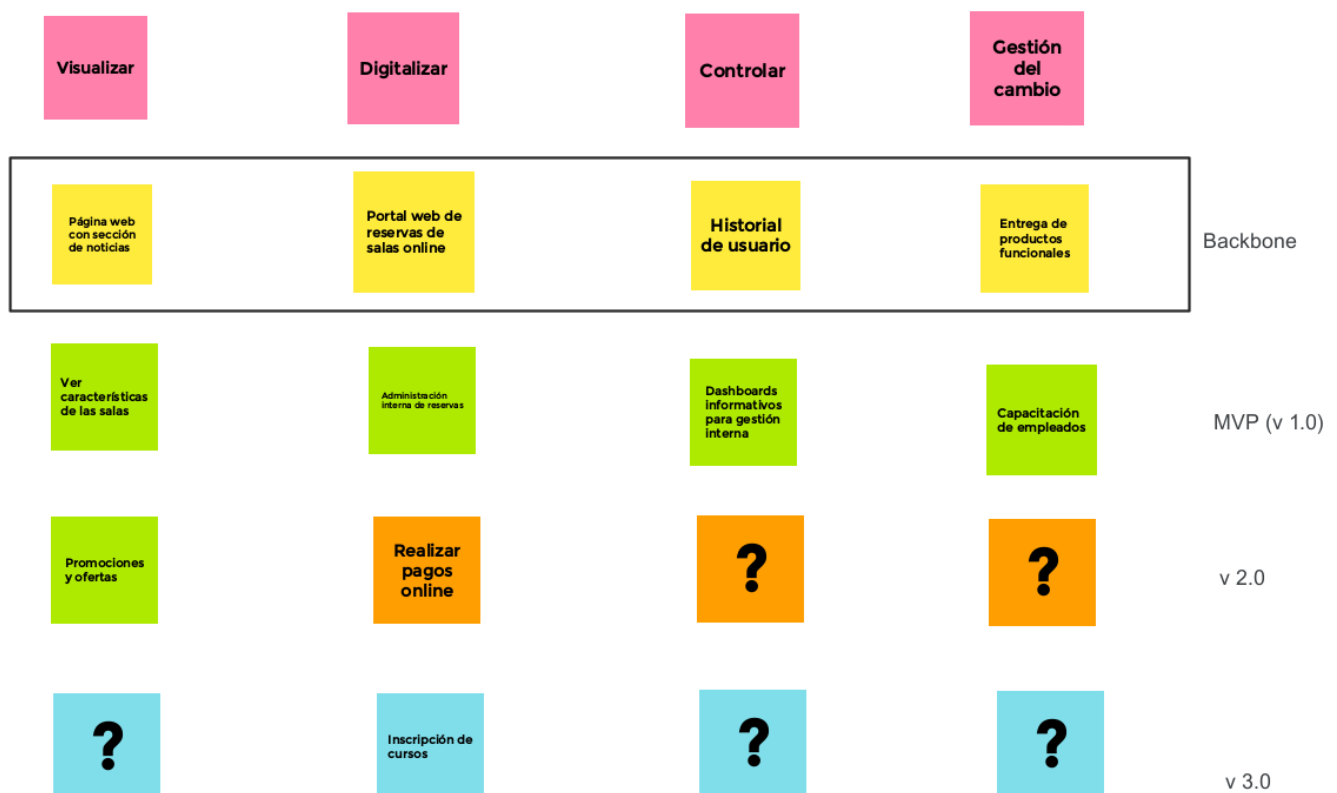


Figura 16 – User history map- Fuente: elaboración propia (2021)

Para este segundo mapa se toman los “¿QUÉ?’s” definidos en el mapa de impacto anterior y se desglosan las cosas que se deben hacer, luego se prioriza lo más relevante para el proyecto y finalmente, se arman las versiones del producto a desarrollar como tal.

En amarillo se pueden ver los aspectos más importantes del sistema, estos pertenecen al Backbone del proyecto o columna vertebral. Estos elementos representan las funcionalidades más importantes para la contraparte y es la base sobre la cual se comienza a trabajar el desarrollo del sistema.

El resto de las tarjetas representan algunas otras funcionalidades que podrían agregar valor al sistema, también priorizadas por color, desde el verde (v 1.0), pasando por el naranja (v 2.0), hasta el cian (v 3.0). Se desarrollarán las funcionalidades por orden de prioridad, modificando lo necesario en caso de que la contraparte lo requiera.

4.2 Fase de Elaboración

Una vez terminada la fase de Inception, se comienzan a identificar los requerimientos del sistema que son validados por la contraparte mediante reuniones de trabajo colaborativas (ver detalle en anexo). Además, se planifica el desarrollo del sistema para dar resolución a cada uno de los requerimientos anteriormente listados. A continuación, se presenta la carta Gantt elaborada con los plazos de desarrollo estimados para el correcto cumplimiento de los objetivos planteados.

PROYECTO DE TESIS

AVANCE ACTUALIZADO AL 14-10**

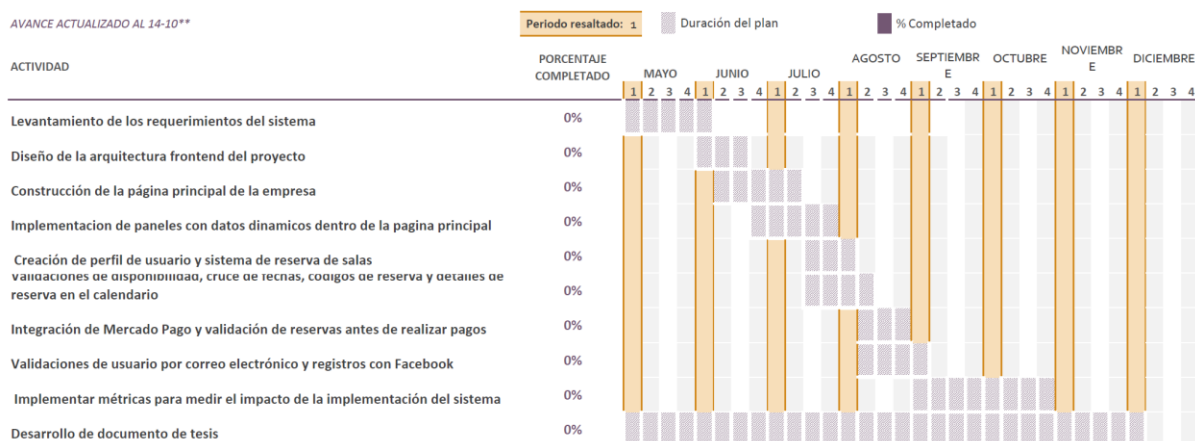


Figura 17 - Planificación del proyecto - Fuente: elaboración propia (2021)

En donde las fases del proyecto son:

1. Levantamiento de los requerimientos del sistema
2. Diseño de la arquitectura frontend del proyecto
3. Construcción de la página principal de la empresa
4. Implementación de paneles con datos dinámicos dentro de la página principal
5. Creación de perfil de usuario y sistema de reserva de salas
6. Validaciones de disponibilidad, cruce de fechas, códigos de reserva y detalles de reserva en el calendario
7. Integración de Mercado Pago y validaciones de reserva antes de realizar pagos
8. Validaciones de usuario por correo electrónico y registro con Facebook
9. Implementar métricas para la medición del impacto del sistema
10. Elaboración de libro de tesis

Tipos de usuario identificados:

Luego del análisis de requerimientos realizado anteriormente y después de las reuniones realizadas con la contraparte, logramos identificar los usuarios que podrán interactuar con el sistema una vez implementado, estos son los siguientes:

1. **Cliente estándar:** es el tipo de cliente que busca un espacio colaborativo para trabajar ya sea de forma individual o con un equipo de trabajo. Por lo general prefieren horas específicas para sus reuniones.
2. **Cliente académico:** es el tipo de cliente que realiza alguna actividad recurrente como académico o para enseñar a un grupo reducido alguna práctica o taller. Por lo general estos clientes prefieren contratar planes ya que sus reuniones requieren de un espacio en horarios determinados por el tiempo que dure el taller o curso que imparten.
3. **Administradores:** este usuario es quien administra el portal, modifica la información de la página web y visualiza los datos de las reservas hechas, entre otras tareas administrativas. Los administradores de Lingo's actualmente son dos.

A continuación, se presentan los requerimientos funcionales y no funcionales obtenidos en esta fase, validados por la contraparte

Requerimientos Funcionales de sistema

| N° | Requerimiento funcional | Descripción |
|----|-------------------------|---|
| 1 | Perfil de usuario | Permitir a los clientes el acceso a su perfil mediante validación por correo electrónico o Facebook |
| 2 | Disponibilidad | Permitir a los clientes registrados ver disponibilidad de salas en calendario |
| 3 | Reservas y pago | Permitir a los clientes realizar reserva de salas mediante el pago con MercadoPago |
| 4 | Planes | Permitir a los clientes comprar planes mensuales |
| 5 | Cancelar reserva | Permitir a los administradores cancelar una reserva ya realizada |
| 6 | Modificar reserva* | Permitir a los administradores modificar los detalles de una reserva |
| 7 | Analítica | Mostrar datos históricos de las reservas para la |

| | | |
|----|------------------|---|
| | | toma de decisiones administrativas |
| 8* | Gestión de datos | Gestionar datos multimedia de la página principal |

Requerimientos no funcionales

| N° | Requerimiento funcional | No | Descripción |
|----|-------------------------|----|---|
| 1 | Interfaz amigable | | El sistema debe ser fácil de usar tanto para los clientes como para los administradores |
| 2 | Seguridad | | El acceso al sistema se validará mediante usuario y contraseña |
| 3 | Accesibilidad | | Se podrá acceder desde cualquier navegador |
| 4* | Responsivo | | El sistema deberá adaptarse y funcionar correctamente en cualquier tipo de dispositivo |

Historias de usuario

| | |
|---|--------------------|
| #1 | Autenticar usuario |
| Como usuario Quiero ingresar al sitio web Para poder pagar reservas | |
| Prioridad: Alta | Valor: 8 |

| | |
|--|--------------------|
| #2 | Autenticar usuario |
| Como usuario Quiero ingresar al sitio web Para ver historial de reservas | |
| Prioridad: Media | Valor: 5 |

| | |
|--|--------------------|
| #3 | Autenticar usuario |
| Como usuario Quiero ingresar al sitio web Para poder ver reservas vencidas | |
| Prioridad: Media | Valor: 6 |

| | |
|--|--------------------|
| #4 | Autenticar usuario |
| Como usuario Quiero ingresar al sitio web | |

| | |
|---|----------|
| Para poder ver reservas próximas a vencer | |
| Prioridad: Media | Valor: 3 |

| | |
|---|----------|
| #5 | Contacto |
| Como usuario Quiero ingresar mis datos en el formulario Para contactar a la empresa | |
| Prioridad: Alta | Valor: 3 |

| | |
|--|----------------|
| #6 | Disponibilidad |
| Como usuario Quiero ingresar al sitio web Para poder ver disponibilidad de salas | |
| Prioridad: Alta | Valor: 8 |

| | |
|--|------------|
| #7 | Ver planes |
| Como usuario Quiero ingresar al sitio web Para poder ver los tipos de planes disponibles | |
| Prioridad: Alta | Valor: 7 |

| | |
|--|----------------|
| #8 | Contratar plan |
| Como usuario Quiero ingresar al sitio web Para poder contratar un plan mensual | |
| Prioridad: Alta | Valor: 6 |

| | |
|--|----------|
| #9 | Carta |
| Como usuario Quiero ingresar al sitio web Para poder ver los productos de la carta que se ofrecen en el café KAFFE | |
| Prioridad: Media | Valor: 4 |

4.3 Fase de construcción

4.4.1 Herramientas por utilizar

Para comenzar con el Desarrollo del sistema web, es importante establecer desde un inicio las herramientas con las cuales este sería construido, las que son presentadas a continuación:

Lenguajes de programación:

- **JavaScript:** Según Guha, Saftoiu, Krishnamurthi (2010) javascript es un lenguaje de programación rico en funcionalidades y con muchas peculiaridades, además de poseer una sintaxis fácil de comprender para los más novatos.
- **HTML:** Según sus siglas en inglés “HyperText Markup Language”, es un lenguaje de marcado compuesto de etiquetas, el cuál define la estructura o esqueleto de un sitio web, en donde se establecen los textos de un sitio, el título, el pie de página, etc. En el caso particular de nuestro proyecto, fue utilizada la versión HTML5.

Técnicas de desarrollo:

- **Ajax:** Según su sigla en inglés “Asynchronous JavaScript and XML”, según Bustos (2021) Ajax no es una tecnología como tal, más bien es una serie de técnicas o formas de escribir datos que son enviados al servidor, los cuales funcionan de manera asíncrona, así permitiendo que los sitios web puedan actualizarse constantemente sin tener que recargar el sitio completamente desde 0, lo cual entrega mayor optimización en la velocidad de respuesta de estos.

Gestor de bases de datos: En el caso particular de este proyecto, se ha utilizado MySQL, el cual es un gestor de bases de datos relacional que se encuentra en la herramienta PhpMyAdmin.

Servidor local: Para este proyecto se ha elegido el paquete de software XAMPP, el cual permite desplegar un servidor que entrega la posibilidad de trabajar de manera local con los archivos correspondientes al sitio web, además de entregar la herramienta PhpMyAdmin que nos permite gestionar las bases de datos MySQL.

Editor de texto: Microsoft Visual Studio Code es uno de los editores de texto más populares del mercado, contando con una extensa variedad de plug-ins que facilitan y ayudan el desarrollo de aplicaciones a los programadores. Este cuenta con soporte para prácticamente todos los lenguajes de programación existentes como JavaScript, TypeScript, Python y PHP, además de también contar con soporte para algunos de los frameworks más usados como Angular, node.js, etc.

Además, y como se mencionó en el capítulo de metodologías se utilizan elementos de la metodología Kanban mediante la herramienta web Trello para la gestión de tareas y roles dentro del proyecto.

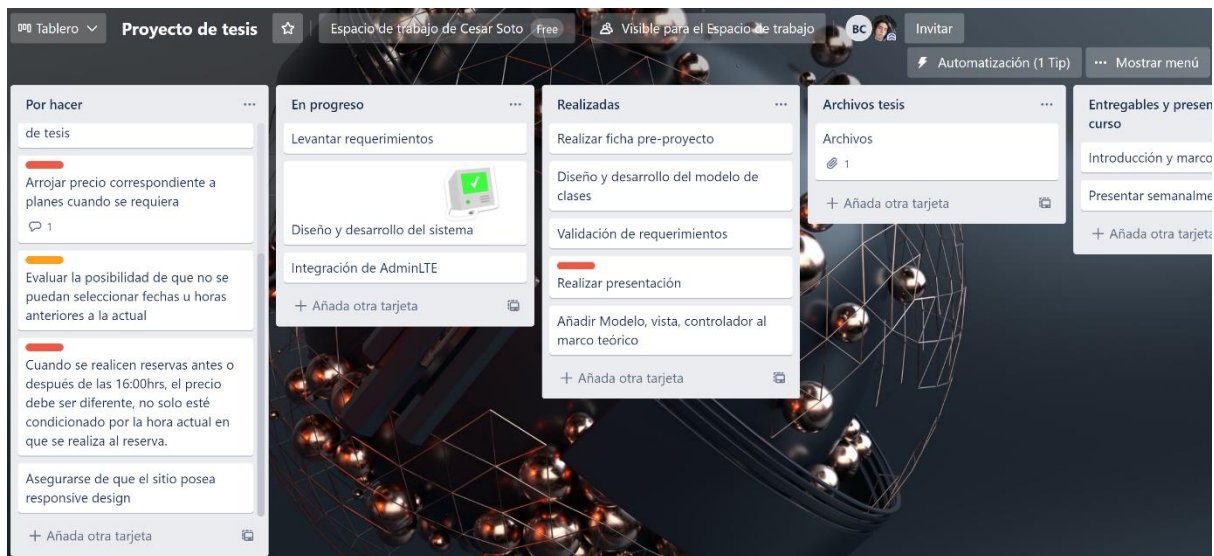


Figura 18 – Tablero de Trello del equipo de desarrollo - Fuente: elaboración propia (2021)

4.4.2 Sprint 1

Tal como se menciona en el capítulo anterior, se decidió utilizar la metodología RUP para la estructura base del proyecto, la cual se adaptó a las necesidades del equipo de desarrollo y del proyecto en sí mismo. Para ello se dividió el trabajo en sprints de desarrollo incrementales e iterativos, partiendo con el primero, cuyo objetivo es el diseño de la arquitectura frontend del proyecto. En este sprint se abordaron las siguientes historias de usuario y requerimientos:

| Requerimientos No funcionales | Historias de usuario |
|-------------------------------|----------------------|
| 1-3-4 | 5-6-7-9 |

Tabla 7 – Historias de usuario y requerimientos no funcionales a abordar en el sprint 1 - Fuente (elaboración propia, 2021)

Primeramente, se diseñó la estructura de la base de datos mediante la construcción del modelo relacional del proyecto, en donde se definieron las tablas principales y sus interacciones como se muestra en la figura 19.

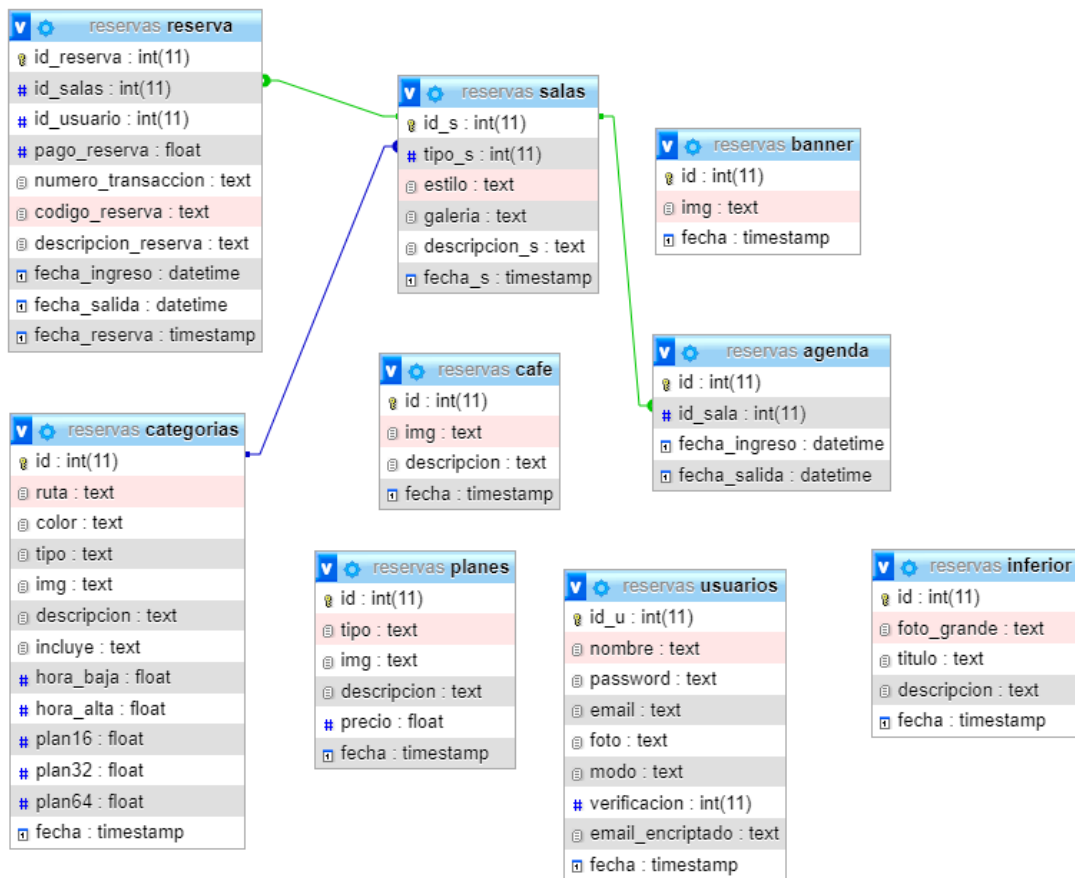


Figura 19 – Modelo Relacional - Fuente: elaboración propia (2021)

En este modelo relacional se estructuran las tablas del proyecto y de qué forma interactúan y se almacenan los datos ingresados en el portal de reservas. Podemos observar que las funciones de reservas de salas están determinadas por cuatro tablas principales que gestionan los datos que los usuarios registran según sus requerimientos, tal como se ve en el siguiente ejemplo:

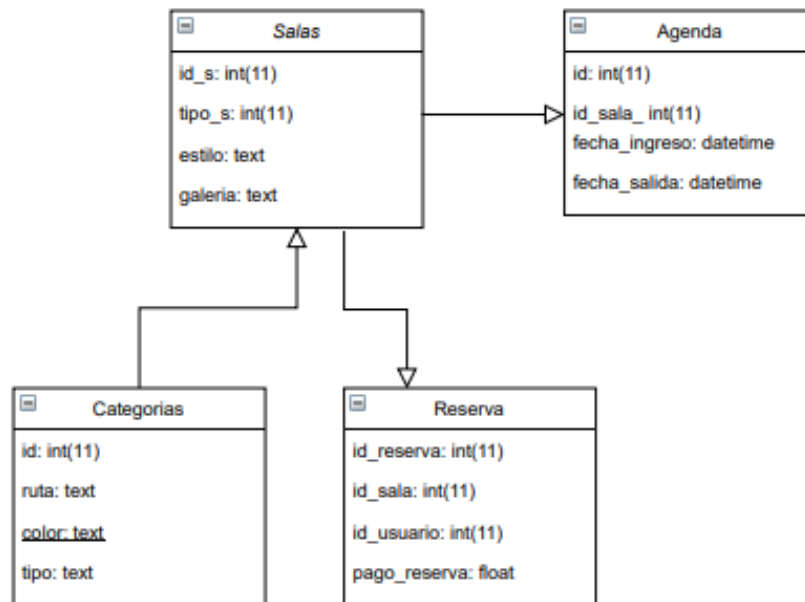


Figura 20 – Tablas modelo relacional - Fuente: elaboración propia (2021)

La tabla “SALAS” almacena todos los datos referentes a las salas y sus características, definidas con un ID único en la variable “id_s” del tipo INT, un tipo definido en “tipo_s” para verificar el tipo de sala (normal o grande), entre otras características como el estilo de la ambientación, su descripción, entre otros. Dentro de la tabla “CATEGORIAS” se agrega un ID único de seguimiento para cada categoría entre otras características como los planes asociados, su descripción. Por otra parte, la tabla de “RESERVA” almacena los datos referentes a las reservas realizadas, definiendo al igual que en las otras tablas un ID único para cada reserva. Este ID se genera automáticamente y de manera aleatoria una vez realizada la reserva. Dentro de esta tabla se almacenan además otros datos foráneos de otras tablas asociadas como lo son el id de la sala reservada, el id del usuario que hace la reserva, la confirmación del pago de la reserva junto a su correspondiente número de transacción, descripción, fecha de entrada y salida, entre otros.

Se realiza además el prototipo del sistema mediante el diseño de una maqueta o mockup como se muestra a continuación:

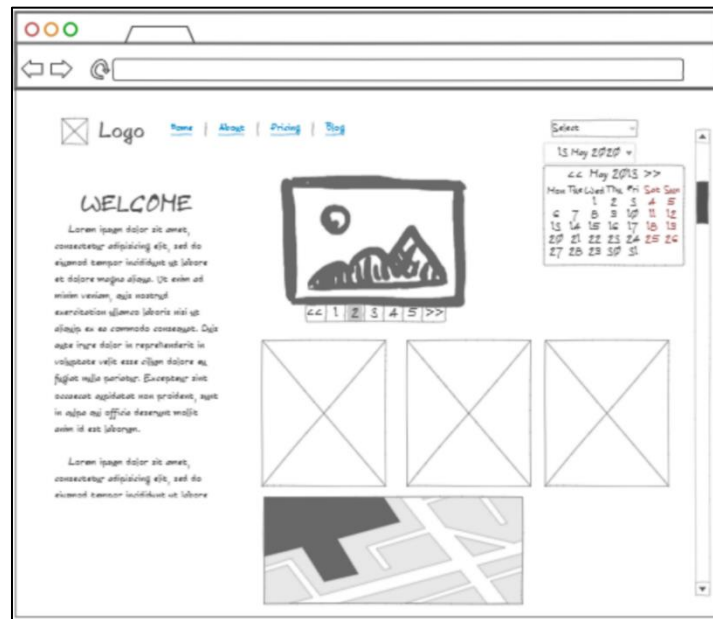


Figura 21 – Mockup / Landing page - Fuente: elaboración propia (2021)

La anterior es la página de aterrizaje o Landing page, posee ese nombre debido a que es el Home de la página y es donde llegarán los usuarios que accedan a ella. Es por eso por lo que su propósito es mostrar detalladamente al potencial cliente los servicios que se ofrecen, fotos atractivas del lugar, mapa de ubicación de la empresa, una breve descripción de las actividades que se realizan y una casilla de contacto en el footer (pie de página).

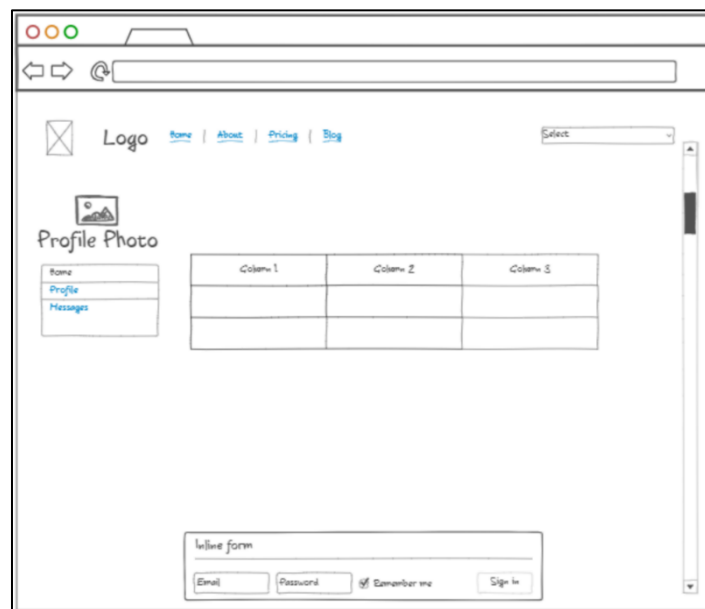


Figura 22 – Mockup / Perfil de usuario - Fuente: elaboración propia (2021)

Dentro del prototipo del perfil de usuario se espera tener un detalle del historial de reservas realizadas además de los datos personales de cada uno de los clientes. Estos deberán acceder a su perfil mediante validación de nombre de usuario y contraseña y el administrador podrá ver los detalles de cada reserva para utilizar dicha información en caso de que sea necesario realizar un seguimiento sanitario.

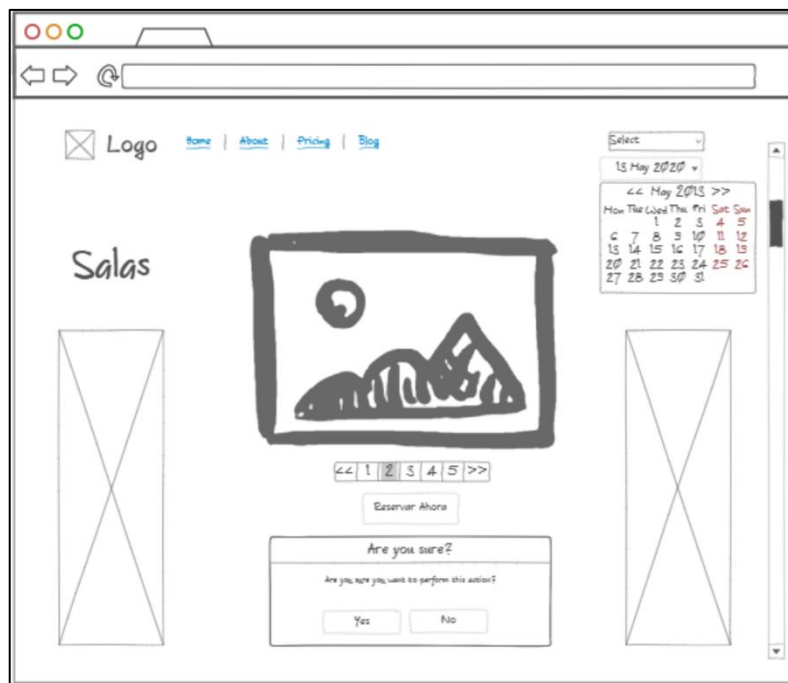


Figura 23 – Mockup / Salas - Fuente: elaboración propia (2021)

Dentro de la página de salas se podrá acceder a toda la información relevante al equipamiento, dimensiones y galería de imágenes de cada sala. Cabe destacar que para la contraparte este aspecto era de gran relevancia debido a que las sesiones realizadas en estos espacios suelen ser recurrentes y están dirigidos a un grupo de alumnos, por lo que para los tutores era muy importante contar con un detalle completo de todas las características de cada espacio de trabajo.

Se realiza además el sitemap del sitio web en donde se pueden observar las distintas secciones de la página de manera gráfica. Se muestra en el recuadro principal el contenido de la página de aterrizaje dividido en secciones (ver figura 20)

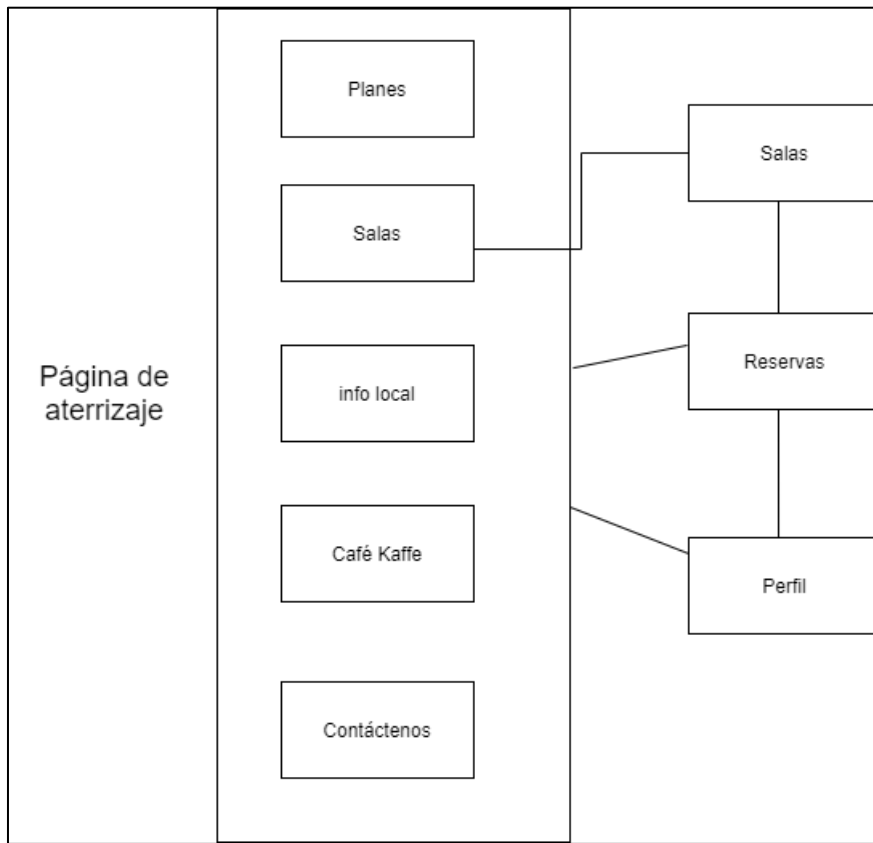


Figura 24 – Site Map - Fuente: elaboración propia (2021)

Diagramas UML

Los siguientes diagramas UML propios de la metodología seleccionada muestran primeramente (figura) el proceso de realizar una reserva, en donde el usuario inicia su sesión validando sus datos de nombre de usuario y contraseña previamente registrados. Una vez se encuentra dentro de su sesión ingresa las fechas de check in y check out de la reserva deseada para verificar la disponibilidad horaria de esta. Se realiza la consulta con la base de datos y se verifica la disponibilidad solicitada, en caso de haber disponibilidad se confirma la reserva, de lo contrario se pide cambiar los horarios de reserva.

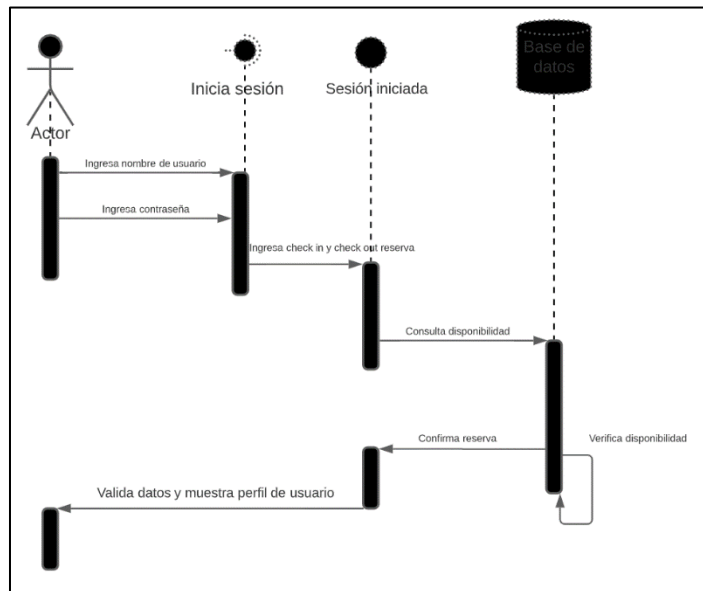


Figura 25 – Diagrama de secuencia de una reserva - Fuente: elaboración propia

(2021)

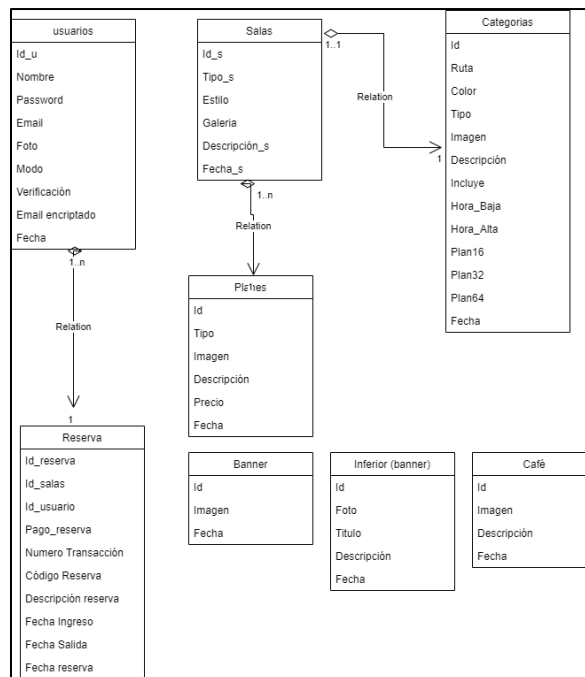


Figura 26 – Site Map - Fuente: elaboración propia (2021)

Luego, en la figura se muestran las tablas relacionales y como estas interactúan entre sí. Una sala pertenece a una única categoría, pero una categoría posee varias salas. Un usuario puede tener varias reservas en su historial, pero una reserva pertenece a un único usuario. Una sala puede tener distintos planes y un plan puede estar dirigido a más de una sala.

Dificultades de la iteración

Durante el transcurso de la tercera semana del sprint 1 se confirma el cambio de quien era hasta entonces el profesor guía del proyecto, lo que trajo consigo múltiples desajustes, principalmente del orden teórico y metodológico afectando así el desarrollo del sistema tal y como se tenía propuesto y propició un reordenamiento del plan de trabajo. Las principales modificaciones se presentan a continuación:

- 1) Herramientas por utilizar: se modifica la estructura base del proyecto pasando de ser desarrollado en un ERP integrado (Odoo 12) a una estructura Modelo Vista Controlador
- 2) Modificaciones en el plan de trabajo: para el correcto cumplimiento de las fechas de entregables y deadlines de algunos de los capítulos desarrollados en este proyecto se realizan modificaciones al plan de trabajo quedando de la siguiente manera:

PROYECTO DE TESIS

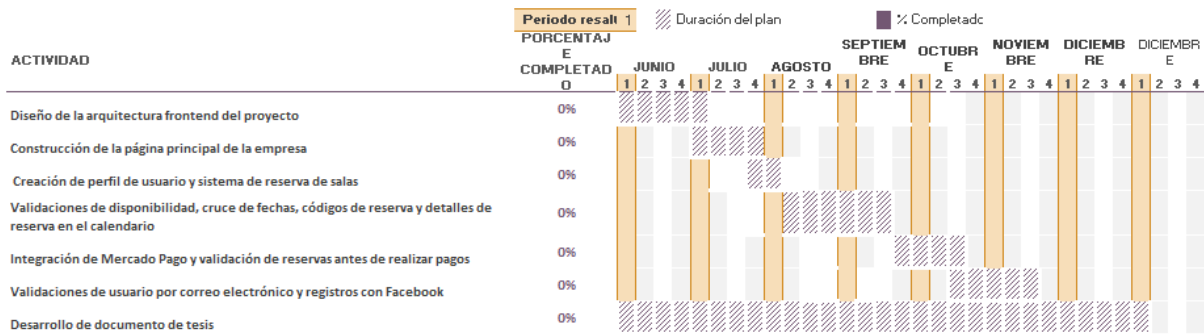


Figura 27 – Plan de trabajo actualizado - Fuente (elaboración propia, 2021)

- 3) Modificaciones en el backlog: se realizan además modificaciones en la lista de requerimientos, quedando descartados los requerimientos funcionales 6 y 8 además del requerimiento no funcional 4 (para detalles ver tabla xxx). Esto debido a que son los requerimientos que menor importancia tenían para la contraparte y de los que más tiempo tomaba desarrollar.

4.4.3 Sprint 2

El segundo sprint tiene como objetivo el desarrollo de la página principal del proyecto según la estructura definida anteriormente en el sprint 1. Para ello se abordarán los siguientes requerimientos funcionales e historias de usuario:

| Requerimientos No funcionales | Historias de usuario |
|-------------------------------|----------------------|
| 1-2 | 5-7-9 |

Tabla 8 – Historias de usuario y requerimientos no funcionales a abordar en el sprint 2 - Fuente (elaboración propia, 2021)

Se desarrollan los menús principales del portal web y sus principales secciones según la navegación establecida y priorizada por la contraparte.

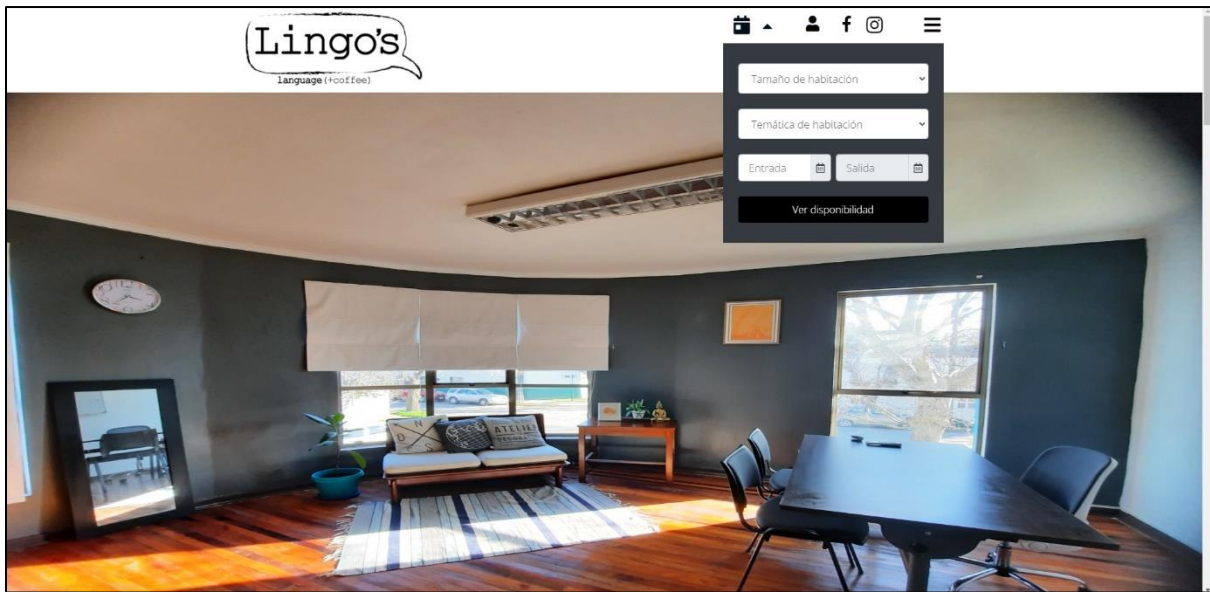


Figura 28 – Pantalla página principal – Elaboración propia (2021)

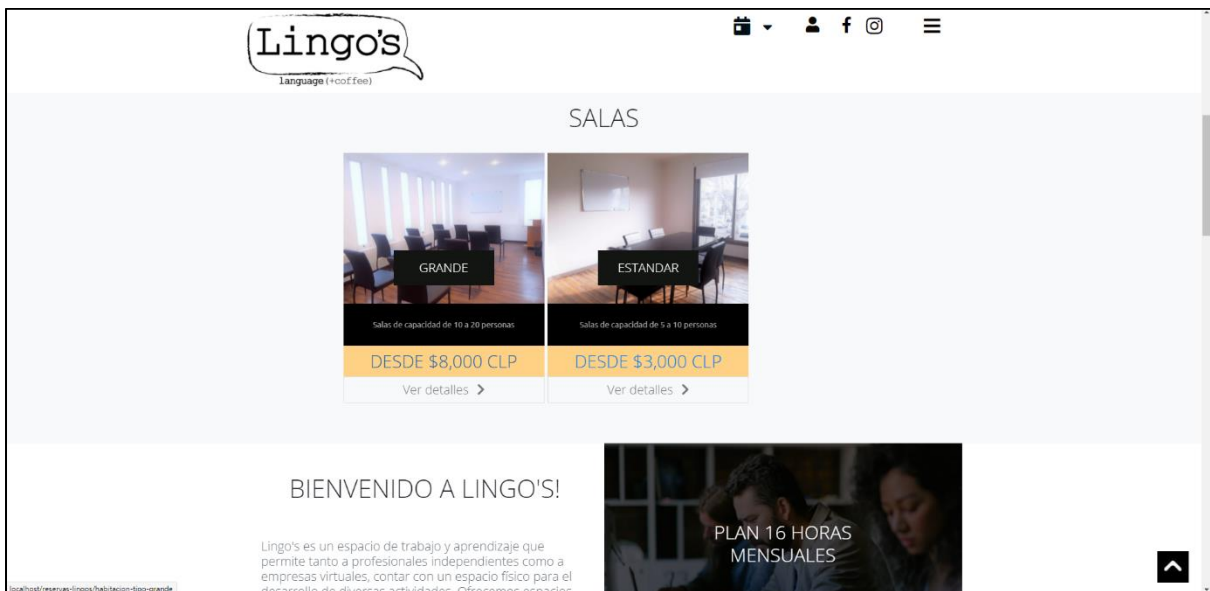


Figura 29 – Pantalla sección planes mensuales – Elaboración propia (2021)



Figura 25 – Pantalla sección Kaffe – Elaboración propia (2021)

Se establece el flujo de navegación dando prioridad a la sección de planes, esto debido a que actualmente los planes mensuales de reservas de salas son el servicio más ampliamente cotizado por los clientes y, además, el tipo de servicio que mayor beneficio otorga a la empresa ya que se utiliza a modo de fidelización de clientes. Estos tipos de clientes, anteriormente clasificados como “Cliente Académico” (ver subcapítulo 4.1) debido a que generalmente son tutores de alguna disciplina o área de conocimiento, buscan espacios para realizar reuniones recurrentes, por lo que tienden a realizar reservas con mayor frecuencia y representan para Lingo's a su tipo de cliente prioritario.

Posteriormente, se realiza sprint review para mostrar el incremento de sistema a la contraparte, validando el trabajo con modificaciones menores y dando paso al siguiente sprint.

Modificaciones realizadas:

- ✓ Se actualiza galería de imágenes de la navegación principal
- ✓ Se modifica tamaño y tipografía de texto en el menú desplegable

4.4.4 Sprint 3

El objetivo principal del tercer sprint es construir las funciones de registro de usuarios para realizar reservas, ver disponibilidad de salas en calendario, además de ver datos históricos de reservas e ingreso con cuentas de Facebook. Para ello se abordaron los siguientes requerimientos de sistema e historias de usuario:

Tabla 9 – Historias de usuario y requerimientos funcionales a abordar en el sprint 3 - Fuente (elaboración propia, 2021)

| Requerimientos funcionales | Historias de usuario |
|----------------------------|----------------------|
| 1-2-4 | 1-2-3-4-6-8 |

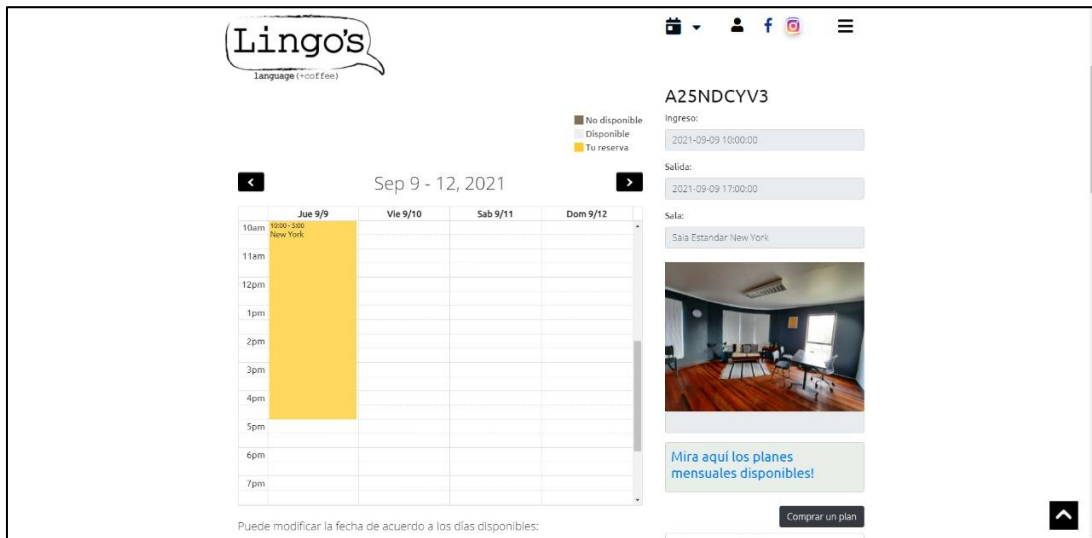


Figura 26 – Calendario de disponibilidad - Fuente: elaboración propia (2021)

Para realizar una reserva exitosamente, el usuario deberá verificar si existe disponibilidad de la sala seleccionada en el horario solicitado. Para ello el portal consulta con la base de datos de registros y muestra al usuario un calendario con la disponibilidad horaria para la sala y genera automáticamente un código único de reserva que se muestra en la parte superior derecha de la pantalla que se validará en caso de confirmar la reserva.

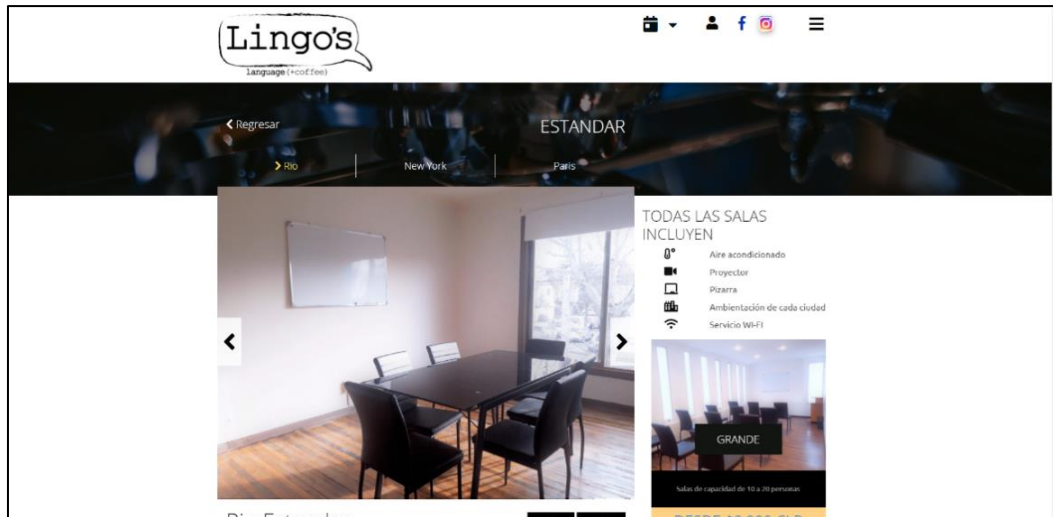


Figura 27 – Perfil de usuario 1 - Fuente: elaboración propia (2021)

Una vez que el cliente inicia su sesión, ya sea de manera tradicional mediante validación de usuario y contraseña o a través de Facebook, ingresa a los datos históricos de las reservas realizadas con su respectivo código de reserva y fechas de

ingreso y salida. Además, podrá ver sus datos personales y modificarlos de ser necesario.

Se añaden en este sprint las imágenes actualizadas de las salas con sus características y equipamiento disponible, además de una breve descripción de esta y una lista con los distintos planes mensuales que se ofrecen.



[Figura 28 – Detalle de salas - Fuente: elaboración propia \(2021\)](#)

Durante el sprint 3 fue muy importante que a medida que se iban construyendo algunas de las principales características del sistema, estas fueran siendo consultadas y mostradas a la contraparte a través de productos funcionales, lo que nos permitía como equipo conocer si se iba por un buen camino o si se debían realizar modificaciones de acuerdo con la retroalimentación obtenida.

El motivo por el cual se decide crear una característica de perfiles de usuarios es por el diagnóstico realizado en la etapa de Inception, en donde la contraparte solicita una característica que le permita mantener un control de los datos de los clientes que realizan las reservas de los espacios de trabajo disponibles, esto de cara a la pandemia permitiendo tener los datos de los clientes en caso de posibles contagios dentro del local, y así poder informar a ellos o a las autoridades correspondientes en caso de necesitar trazar los contagios.

Puesto que la característica de perfiles de usuario nos entrega bastante información con respecto a los clientes, ya que tanto de la forma tradicional y sobre todo vía

ingreso Facebook, se pueden almacenar los datos de los clientes con los fines previamente mencionados.

Durante la etapa de desarrollo de la pantalla de reservas fue dónde se presentaron mayores dificultades, puesto que se desconocía el uso de plug-ins y como hacer que estos interactuaran directamente con la base de datos, en un inicio se podían consultar solamente la disponibilidad según el día, pero no por horas como realmente era necesario para el negocio. Finalmente se logra resolver esta problemática y se logra hacer funcionar correctamente el plug-in de calendario por horas con la lógica ubicada en la capa de back-end.

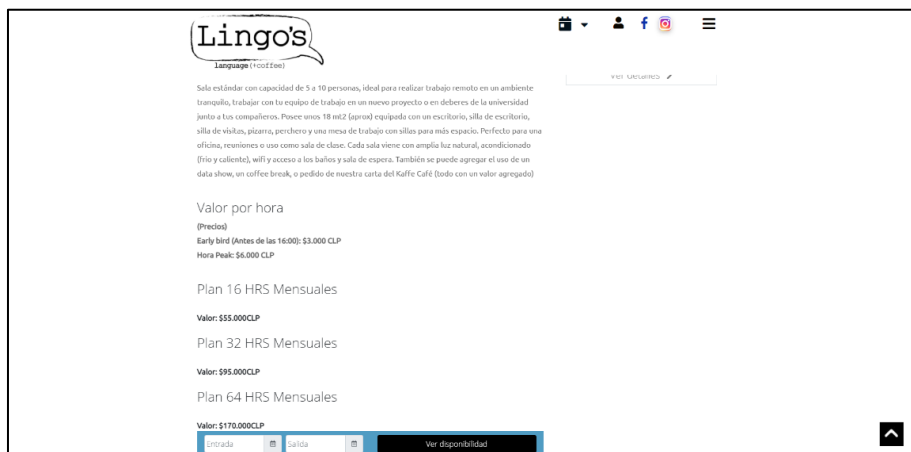


Figura 29 – Detalle de planes - Fuente: elaboración propia (2021)

4.4.5 Sprint 4

En el cuarto sprint los principales objetivos son la integración de método de pago con Mercado Pago, pruebas de validación de sistema. Para ello se abordaron los siguientes requerimientos e historias de usuario:

| Requerimientos funcionales | Historias de usuario |
|----------------------------|----------------------|
| 3-4 | 1-8 |

Tabla 10 – Historias de usuario y requerimientos funcionales a abordar en el sprint 4
- Fuente (elaboración propia, 2021)

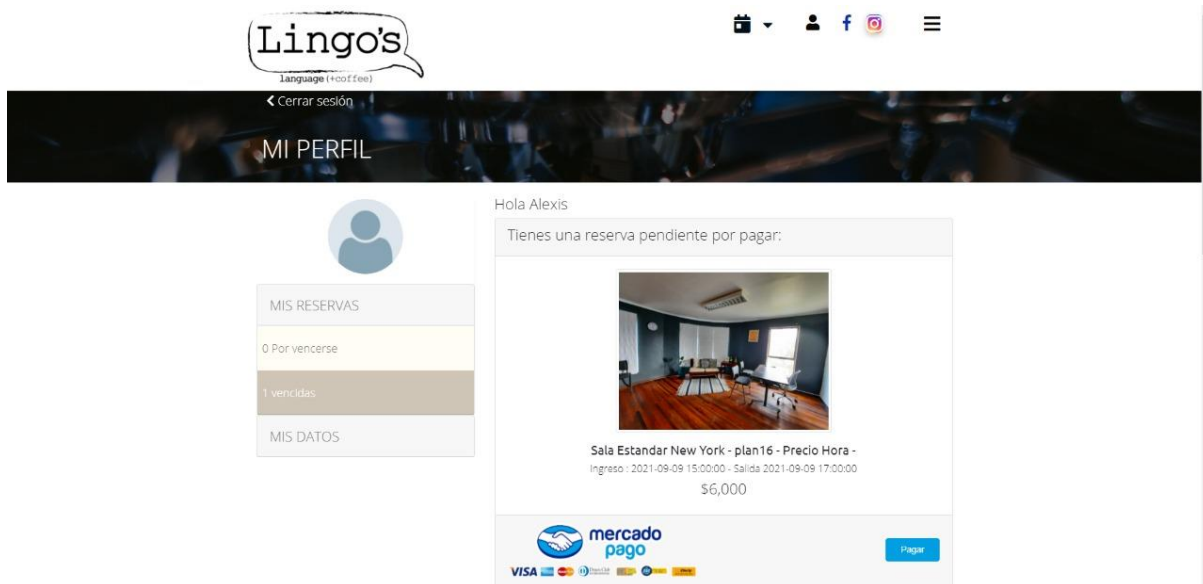


Figura 30 – Perfil de usuario - Fuente: elaboración propia (2021)

Una vez que el usuario confirma el horario de la reserva y esta hora se encuentra disponible, comienza el proceso de pago mediante la integración de la plataforma de Mercado Pago. Este método permite el pago de reservas con tarjetas de crédito o débito de cualquier banco otorgando la máxima seguridad en transacciones ACID para garantizar que los datos se procesan de manera fiable.

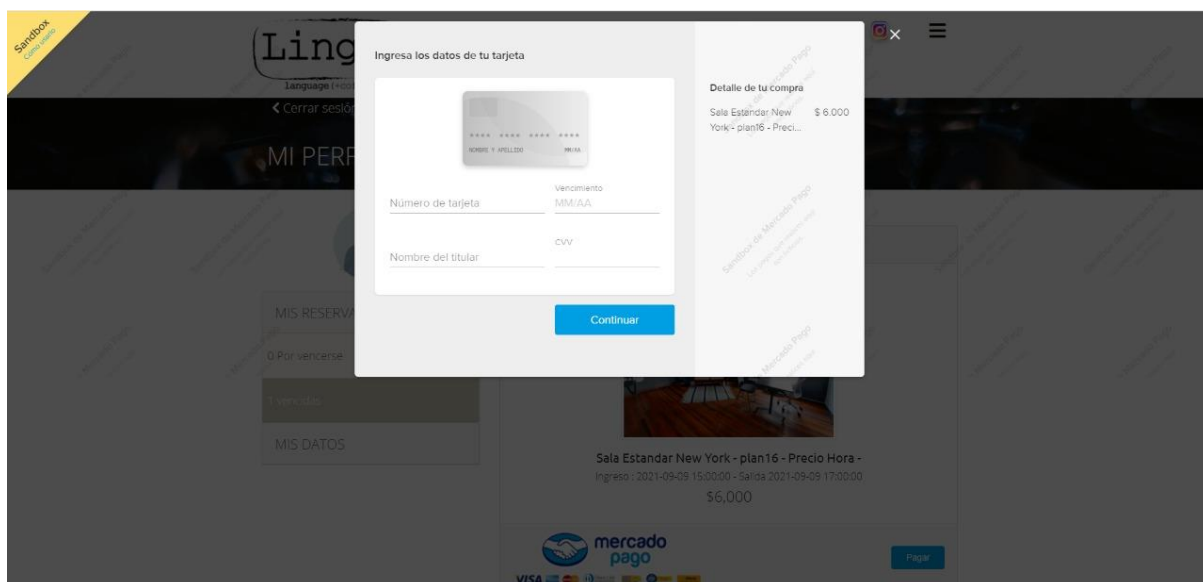
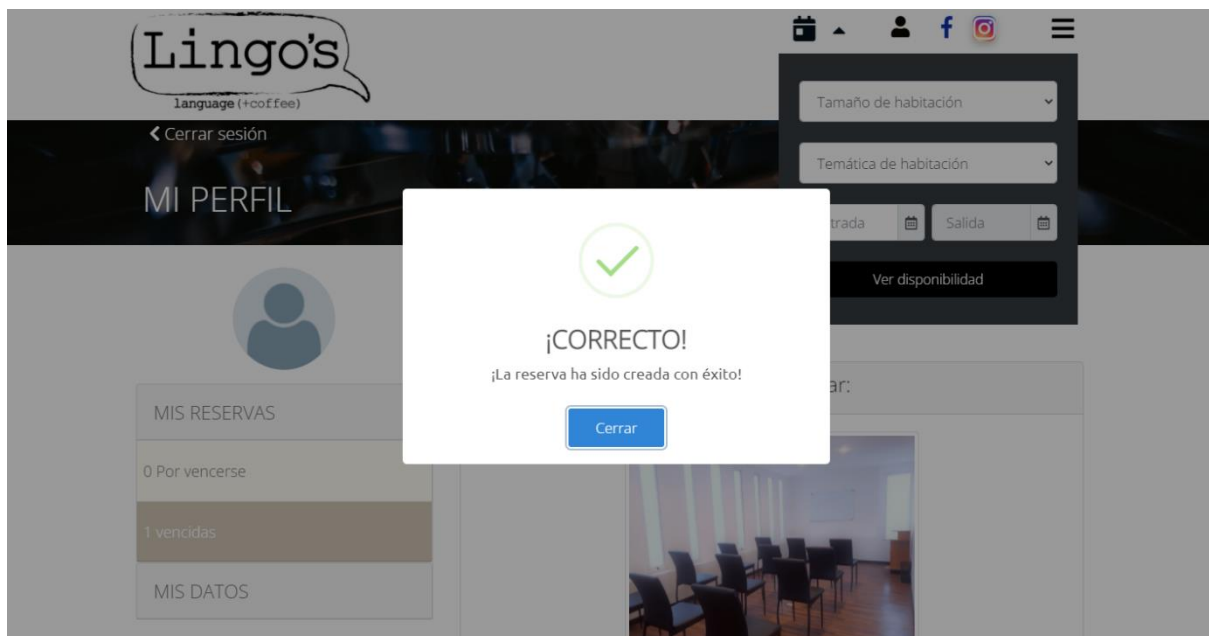


Figura 31 – Pago de reservas - Fuente: elaboración propia (2021)



[Figura 32 – Pago de reservas - Fuente: elaboración propia \(2021\)](#)

Una vez confirmado el pago de la reserva aparecerá un mensaje de confirmación que confirma que la reserva se realizó de manera exitosa. Además, se mostrará la reserva en la lista de reservas pendientes dentro del perfil del usuario en sus datos de reserva históricos.

Otra de las características desarrolladas durante este sprint son los mensajes de confirmación por correo electrónico a las interacciones realizadas por el usuario respecto a su cuenta y a las consultas realizadas en el formulario de contacto.



[Figura 33 – Pago de reservas - Fuente: elaboración propia \(2021\)](#)



HA RECIBIDO UNA CONSULTA

CESAR SOTO

Móvil: 84848375378

Email: cesar@gmail.com

Tengo una pregunta.

Figura 34 – Pago de reservas - Fuente: elaboración propia (2021)



SOLICITUD DE NUEVA CONTRASEÑA

Su nueva contraseña: 32423sgxdsg34

[Haz click aquí](#)

Ingrese nuevamente al sitio con esta contraseña y recuerde cambiarla en el panel de perfil de usuario

Si no se inscribió en esta cuenta, puede ignorar este correo electrónico y la cuenta se eliminará.

Figura 35 – Pago de reservas - Fuente: elaboración propia (2021)

4.5 Fase de Transición

Posterior a la finalización del sprint 4 del proyecto, este procedió a trasladarse desde un entorno de desarrollo local hacia el entorno operativo en el hosting que poseía previamente el cliente (Inchosting). Este hosting entrega la posibilidad de subir el sitio web a la red, entregando un servicio con SSD integrado, aumentando la velocidad de carga. Adicionalmente, el servicio de hosting provee a los clientes la posibilidad de manejar correos electrónicos entre los trabajadores o administradores del negocio, un cPanel para la gestión de los archivos del sitio web, etc.

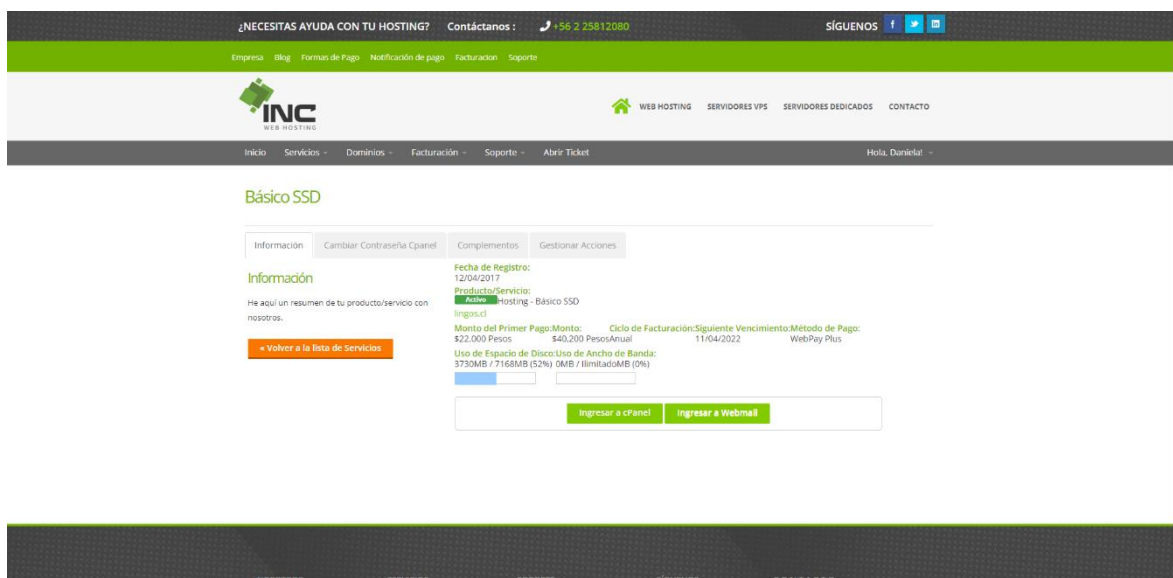


Figura 36 - Vista de inchosting y detalles del servicio que posee Lingo's (Creación propia, 2021)

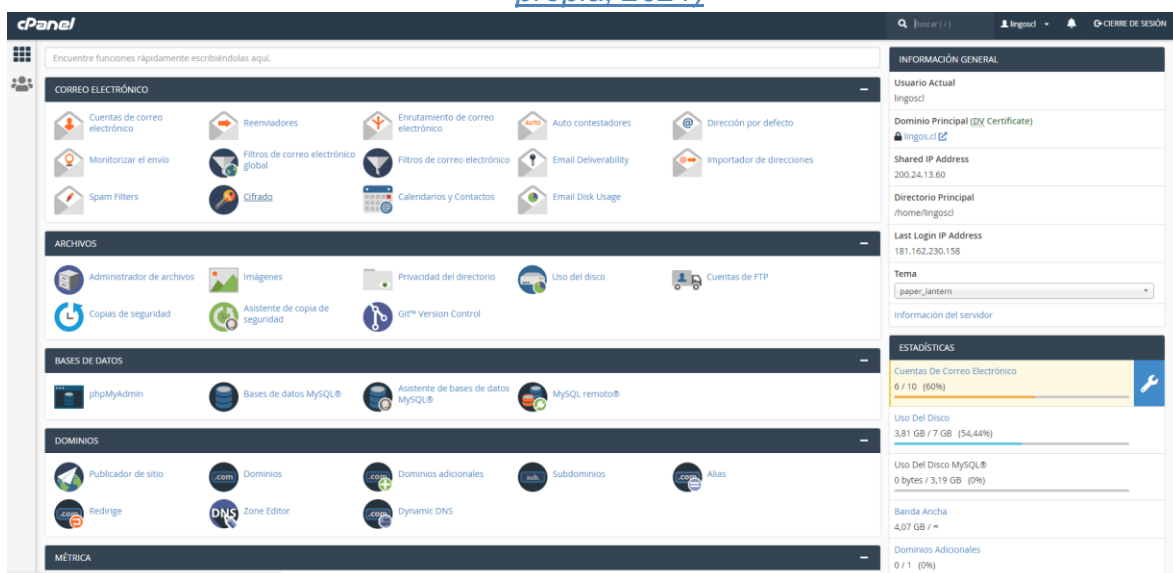


Figura 37 - Vista del cPanel y de las funcionalidades que entrega (Creación propia, 2021)

Eventos y observaciones de la iteración final

Una vez finalizado el sprint 4, se procede a subir al hosting previamente mencionado, los archivos correspondientes al sitio web desarrollado al gestor de archivos del este, ubicado dentro del cPanel reemplazando los archivos pertenecientes al sitio web antiguo que poseía la empresa.

Al inicio del sprint 4 se comienza con la integración de la pasarela de pagos “Mercado pago”, se elige esta plataforma de pagos de reservas o planes debido a la facilidad en su integración, además de la extensa cantidad de tarjetas de débito y crédito aceptadas, entregando a los usuarios una gran variedad de métodos de pago.

Posterior a la integración de la pasarela de pagos, se comienza a aplicar la multimedia final a utilizar en el sitio web, cargando las imágenes e información sobre los precios de reserva, planes y las características de cada sala, además de comenzar con los ajustes finales para comenzar con el despliegue del portal web.

Durante la etapa final del proyecto se presentan dificultades al momento de desplegar el sitio web en el hosting, debido al desconocimiento por parte de equipo con respecto a esta tarea, sin embargo, finalmente se logra el despliegue correcto del sitio web tras una fase de investigación sobre cómo utilizar las características del cPanel, varios intentos, reajustes y actualizaciones de versiones de PHP en el hosting que posee el negocio.

Luego del correcto despliegue del sitio web construido a lo largo del proyecto, se procede a mostrar el producto final a la contraparte, recibiendo comentarios muy positivos de su parte y la aprobación de este, principalmente destacando el diseño que este posee, la funcionalidad de la pasarela de pagos y permitir una correcta gestión de los datos de los clientes y de las reservas.

5 Conclusiones

En este proyecto se tomó como materia de estudio el caso de Lingo's SPA, un Coworking de la ciudad de Talca que buscaba principalmente una alternativa digital para sistematizar su actual proceso de reserva manual. Para ello se analizó el caso recuperando la mayor cantidad de datos para convertirlos en información relevante con la ayuda de herramientas propias de la metodología RUP, como mapas de impacto, mapas de historial de usuario, entre otros, lo que permitió al equipo comprender de mejor manera la real problemática de la empresa para posteriormente trabajar en el diseño y desarrollo de una solución propicia. Este diseño y desarrollo de sistema se realizó mediante ciclos de desarrollo iterativos con estrecha colaboración de la contraparte la cual validó cada una de las entregas, haciéndose partícipe de este. Esto permitió que el despliegue del sistema fuera mucho más cómodo para ambas partes ya que disminuye la resistencia al cambio que generan este tipo de proyectos.

Los principales beneficios de tener un portal web de reservas en Lingo's tienen que ver con facilitar el proceso de reserva para sus clientes, sobre todo en el contexto sanitario y social en que nos encontramos, además de llevar registro de la utilización de las salas para realizar seguimientos en caso de que sea necesario.

Otro beneficio por considerar es la digitalización de los procesos de negocio que otorga el portal, principalmente, mediante la automatización del proceso de reserva que permite a los clientes realizar reservas de manera autónoma y sencilla, ofreciendo distintos medios de pago online confiables. Esto permite ampliarse a un segmento de clientes nativos digitales o mayormente instruidos en las tecnologías de información, segmento que crece día a día y a los cuales no estaban llegando de manera efectiva con su método de reservas tradicional/manual.

El acceso a la información es otro aspecto que mejoró notablemente con la implementación del portal web, ya que en este se detallan todas las características de las salas, se pueden ver imágenes reales del local y su entorno en la galería de imágenes, el cliente es capaz de realizar consultas directamente a los administradores vía correo electrónico, y por sobre todo, el acceso a la información de cada cliente dentro de su perfil de usuario para llevar un registro de las salas utilizadas y los horarios de uso para seguimiento sanitario y en caso de que surja un

brote del virus Covid-19 se puedan tomar las acciones necesarias a tiempo para salvaguardar la integridad física tanto de clientes como de trabajadores de la empresa.

Sin duda que la importancia de que este tipo de sistemas se vuelvan más accesibles para las PyME's se hace cada vez mayor para un desarrollo económico sostenido dentro de un sector tan particularmente afectado, tanto por la crisis sanitaria como por la posterior recesión económica mundial.

Dentro de los principales aprendizajes que adquirimos durante el desarrollo de este proyecto podemos destacar la importancia de los aspectos metodológicos en proyectos de este tipo. El apoyo que proporciona la utilización de una metodología (o varias) que se adapte a las necesidades del proyecto resulta crucial para llevar a cabo el trabajo de manera óptima, ya que proporciona un marco de trabajo sistemático y ordena el desarrollo de este, disminuyendo los contratiempos que pudiesen surgir.

Además, pudimos aprender a gestionar proyectos TI de manera remota eficientemente mediante la utilización de herramientas como Trello y Microsoft Teams, que facilitaron nuestro trabajo y se volvieron indispensables para adaptar las diferentes metodologías utilizadas a un contexto de desarrollo real y que fuera propicio para el equipo.

Bibliografía

Pelletier Claudia, & Cloutier Martin. (2019). Conceptualising digital transformation in SMEs: an ecosystemic perspective: Recuperado en 8 de mayo de 2021, desde: [\(PDF\) Conceptualising digital transformation in SMEs: an ecosystemic perspective \(researchgate.net\)](#)

Arias-Ortíz, et al. (2015). La innovación y la nueva economía de servicios en América Latina y el Caribe Retos e implicaciones de política: Recuperado en 8 de mayo de 2021, desde: [La innovación y la nueva economía de servicios en América Latina y el Caribe: Retos e implicaciones de política | Publications \(iadb.org\)](#)

Pavón Yanelis, et al (2016). Experiencia de trabajo para la configuración del ERP Odoos en pequeños negocios. Caso de éxito en TostoneT: Recuperado en 8 de mayo de 2021, desde: [0718-3305-ingeniare-26-03-00514.pdf \(conicyt.cl\)](#)

Sernatur (2020). PROTOCOLO DE MANEJO Y PREVENCIÓN ANTE COVID-19 EN Restaurantes, Cafés y Otros Establecimientos Análogos: Recuperado en 8 de mayo de 2021, desde: [restaurant-v2.pdf \(sernatur.cl\)](#)

Marcano Aular, Yelitz Josefina, Talavera Pereira, Rosalba. (2007). Minería de Datos como soporte a la toma de decisiones empresariales. Recuperado en 8 de mayo de 2021, desde: [Minería de Datos como soporte a la toma de decisiones empresariales \(scielo.org\)](#)

Hernández Nelly & Smith Anderson. (2014). COMPUTACIÓN EN LA NUBE CLOUD COMPUTING: Recuperado en 8 de mayo de 2021, desde: <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjz07MssXwAhXyr5UCHRjMDLsQFjAAegQIBhAD&url=https%3A%2F%2Fdialnet.unirioja.es%2Fdescarga%2Farticulo%2F5109245.pdf&usq=AOvVaw0Sx6g39uu28gycJGH7FJ8r>

Dubey Abhijit & Wagle Dilip. (2007). Delivering software as a service. Recuperado en 9 de mayo de 2021, desde: [delievering software as a service.pdf \(weebly.com\)](#)

Primorac Carlos. (2014). Computación en Nube: Recuperado en 9 de mayo de 2021, desde: [primorac_monografia_computacion_en_nube.pdf \(unne.edu.ar\)](#)

Sun Wenchao, Shi Jian-Ping & Patni Chandra. (2013) Packaging system for customizing software. Recuperado en 9 de mayo de 2021, desde: [\(PDF\) Packaging system for customizing software \(researchgate.net\)](#)

Bibi Stamatia, Katsaros Dimitrios & Bozani Panayiotis. (2012). Business Application Acquisition: On-Premise or SaaS-Based Solutions? .Recuperado en 9 de mayo de 2021, desde: [\(PDF\) Business Application Acquisition: On-Premise or SaaS-Based Solutions? \(researchgate.net\)](#)

Diario financiero. (2020). TRABAJO REMOTO: CÓMO SE MUEVEN LAS EMPRESAS EN EL NUEVO CONTEXTO. Recuperado en 9 de mayo de 2021, desde: [A9R6AB9.tmp.pdf](#)

Ramkhelawan Suman, Cadarsaib Zarine, & Gobin Baby. (2015). Cloud Computing as an Alternative for on-Premise Software for Mauritian Hotels. Recuperado en 9 de mayo de 2021, desde: [\(2\) \(PDF\) Cloud Computing as an Alternative for on-Premise Software for Mauritian Hotels | Zarine Cadarsaib - Academia.edu](#)

Herederó Carmen, et al. (2012). Organización y transformación de los sistemas de información en la empresa. Recuperado en 9 de mayo de 2021, desde: [Organización y transformación de los sistemas de información en la empresa - Carmen de Pablos Herederó, José Joaquín López Hermoso Agius, Santiago Martín-Romo Romero, Sonia Medina Salgado - Google Libros](#)

Wang Yao. (2016). What are the biggest obstacles to growth of SMEs in developing countries? - An empirical evidence from an enterprise survey. Recuperado en 9 de mayo de 2021, desde: [\(PDF\) What are the biggest obstacles to growth of SMEs in](#)

[developing countries? - An empirical evidence from an enterprise survey \(researchgate.net\)](#)

Fayad Mohammed, Schmidt Douglas (1997) Object-Oriented Application Frameworks. Recuperado el 31 de Agosto de 2021, desde: [\(PDF\) Application Frameworks \(researchgate.net\)](#)

David J. Anderson & Andy Carmichael (2016) Kanban Esencial Condensado. Recuperado el 04 de Octubre de 2021, desde: [Essential-Kanban-Condensed-Spanish.pdf \(tigalia.com\)](#)

Arjun Guha et al. (2019) The essence of JavaScript. Recuperado el 18 de Octubre de 2021, desde: [jssem.dvi \(brown.edu\)](#)

Luján Mora, S. (2002). Programación de aplicaciones web: historia, principios básicos y clientes web. November. <http://rua.ua.es/dspace/handle/10045/16995>

Universidad Autónoma de México (2021). Recuperado el de 8 diciembre de 2021, desde: http://fcasua.contad.unam.mx/apuntes/interiores/docs/98/4/informatica_4.pdf

Rafael Menéndez-Barzanallana (s.f.) Recuperado el 5 de diciembre de 2021 desde: [Lenguaje-de-programacion-HTML-2.pdf \(um.es\)](#)

Rafael Menéndez-Barzanallana (s.f.) Recuperado el 8 de diciembre de 2021 desde: <https://www.um.es/docencia/barzana/DAWEB/Lenguaje-de-programacion-JavaScript-1.pdf>

Anexos

Anexo 1: Minutas de reunión



Lingo's Talca – Minuta

de reunión

Correo:

bacaceres16@alumnos.otalca.cl
csoto16@alumnos.otalca.cl

Fono: +569 41198764
+569 77601880

Minuta miércoles 22 de septiembre de 2021 – 17:00 hrs (presencial)

| | |
|----------------------|---|
| Convocada por | Bastián Cáceres |
| Tema | Presentación DEMO del portal a la contraparte |
| Organizador | César Soto |
| Duración | 1 hora 12 minutos |
| Asistentes | Taylor Burdett, César Soto, Bastián Cáceres |

Temas en tabla

Discusión

- Se presenta DEMO final del portal de reservas a la contraparte para su aprobación y comentarios. Se realiza capacitación mediante explicación expositiva del funcionamiento del portal a la administradora.

Conclusiones

- La contraparte da su visto bueno al sistema y su funcionamiento. Detalles estéticos menores por corregir para mejorar la interacción del usuario con el portal.

| Plan de acción | Responsable | Plazo |
|--|------------------------------|------------------|
| <ul style="list-style-type: none">- Actualización de imágenes- Reubicación menú desplegable principal- Mejorar responsive design | César Soto – Bastián Cáceres | 30 de septiembre |



Lingo's Talca – Minuta
de reunión

Correo:
bacaceres16@alumnos.otalca.cl
csoto16@alumnos.otalca.cl

Fono: +569 41198764
 +569 77601880

Minuta jueves 24 de junio de 2021 – 15:30 hrs (vía Zoom)

| | |
|----------------------|---|
| Convocada por | César Soto |
| Tema | Sprint review 1 |
| Organizador | César Soto |
| Duración | 47 minutos |
| Asistentes | Taylor Burdett, César Soto, Bastián Cáceres |

Temas en tabla

| |
|--|
| Discusión |
| - Se presenta el incremento del sistema que corresponde a la arquitectura frontend del sistema |
| Conclusiones/acuerdos |
| - La contraparte se muestra conforme con el avance y se acuerda continuar con el siguiente sprint |

| Plan de acción | Responsable | Plazo |
|----------------------------|------------------------------|--------------|
| - <i>Comenzar Sprint 2</i> | César Soto – Bastián Cáceres | 01 de julio |



Lingo's Talca – Minuta

de reunión

Correo:
bacaceres16@alumnos.otalca.cl
csoto16@alumnos.otalca.cl

Fono: +569 41198764
+569 77601880

Minuta jueves 29 de julio de 2021 – 15:30 hrs (vía Zoom)

| | |
|----------------------|---|
| Convocada por | Bastían Cáceres |
| Tema | Sprint review 2 |
| Organizador | César Soto |
| Duración | 35 minutos |
| Asistentes | Taylor Burdett, César Soto, Bastían Cáceres |

Temas en tabla

| |
|--|
| Discusión |
| - Se presenta incremento del sistema correspondiente a la versión inicial de la página web |
| Conclusiones |
| - La contraparte da su visto bueno y se acuerda iniciar el siguiente sprint |

| Plan de acción | Responsable | Plazo |
|-----------------------|------------------------------|--------------|
| - Comenzar sprint 3 | César Soto – Bastían Cáceres | 05 de agosto |



Lingo's Talca – Minuta

de reunión

Correo:
bacaceres16@alumnos.otalca.cl
csoto16@alumnos.otalca.cl

Fono: +569 41198764
 +569 77601880

Minuta jueves 02 de septiembre de 2021 – 17:00 hrs (vía Zoom)

| | |
|---------------|---|
| Convocada por | Bastían Cáceres |
| Tema | Sprint review 3 |
| Organizador | César Soto |
| Duración | 42 minutos |
| Asistentes | Taylor Burdett, César Soto, Bastían Cáceres |

Temas en tabla

| |
|--|
| Discusión |
| - Se presenta incremento de sistema correspondiente a característica de agendamiento de salas en su versión inicial |
| Conclusiones |
| - La contraparte da su visto bueno al sistema y su funcionamiento. Se acuerda comenzar el siguiente sprint |

| Plan de acción | Responsable | Plazo |
|----------------------------|------------------------------|------------------|
| - Comenzar sprint 4 | César Soto – Bastían Cáceres | 30 de septiembre |



Correo:
bacaceres16@alumnos.otalca.cl
csoto16@alumnos.otalca.cl

Fono: +569 41198764
+569 77601880

Lingo's Talca – Minuta

de reunión

Minuta jueves 07 de octubre de 2021 – 15:30 hrs (vía Zoom)

| | |
|----------------------|---|
| Convocada por | Bastián Cáceres |
| Tema | Sprint review 4 |
| Organizador | Bastián Cáceres |
| Duración | 56 minutos |
| Asistentes | Taylor Burdett, César Soto, Bastián Cáceres |

Temas en tabla

Discusión

- **Se presenta DEMO final del portal de reservas a la contraparte con los cambios sugeridos.**

Conclusiones

- **La contraparte da su visto bueno al sistema y su funcionamiento. Se acuerda el despliegue del sistema e implementación**

| Plan de acción | Responsable | Plazo |
|---------------------------------|------------------------------|---------------|
| - Despliegue del sistema | César Soto – Bastián Cáceres | 21 de octubre |



Lingo's Talca – Minuta

de reunión

Correo:

bacaceres16@alumnos.otalca.cl
csoto16@alumnos.otalca.cl

Fono: +569 41198764
+569 77601880

Minuta lunes 20 de abril de septiembre de 2021 – 15:30 hrs (vía Zoom)

| | |
|----------------------|---|
| Convocada por | Bastián Cáceres |
| Tema | Presentación del equipo con la contraparte |
| Organizador | César Soto |
| Duración | 28 minutos |
| Asistentes | Taylor Burdett, César Soto, Bastián Cáceres |

Temas en tabla

Discusión

- Se presenta el equipo de desarrollo con la contraparte. La dueña explica detalladamente los procesos de negocio y las principales actividades relacionadas, además de identificar las principales problemáticas que se desean abordar.

Conclusiones/acuerdos

- Se acuerda abordar la problemática de reserva de salas y planes a través de Internet

| Plan de acción | Responsable | Plazo |
|--|------------------------------|--------------|
| - Elaboración Ficha Pre-proyecto | César Soto – Bastián Cáceres | 29 de abril |
| - Retroalimentación por parte del profesor guía | | |
| - Agregar sugerencias a ficha y presentación | | |



Lingo's Talca – Minuta
de reunión

Correo:
bacaceres16@alumnos.otalca.cl
csoto16@alumnos.otalca.cl

Fono: +569 41198764
+569 77601880

Minuta martes 04 de mayo de 2021 – 19:00 hrs (vía Microsoft Teams)

| | |
|----------------------|--|
| Convocada por | Fernando San Martín |
| Tema | Análisis de proyecto y alcance |
| Organizador | César Soto |
| Duración | 83 minutos |
| Asistentes | Fernando San Martín, César Soto, Bastián Cáceres |

Temas en tabla

| |
|---|
| Discusión |
| <ul style="list-style-type: none">- Niveles de implementación del sistema- Áreas de negocio que se abordarán con el sistema |
| Conclusiones/acuerdos |
| <ul style="list-style-type: none">- Se acuerda abordar área de reservas de salas y planes, implementando en fase de prueba. Se acuerda dar prioridad a la contratación de planes por sobre horas individuales |

| Plan de acción | Responsable | Plazo |
|---|------------------------------|--------------|
| <ul style="list-style-type: none">- Aplicar correcciones a la ficha | César Soto – Bastián Cáceres | 10 de mayo |



Correo:
bacaceres16@alumnos.utaalca.cl
csoto16@alumnos.utaalca.cl

Fono: +569 41198764
+569 77601880

Lingo's Talca – Minuta

de reunión

Minuta martes 11 de mayo de 2021 – 21:00 hrs (vía Discord)

| | |
|----------------------|--|
| Convocada por | Bastián Cáceres |
| Tema | Definición de actores e historias de usuario |
| Organizador | César Soto |
| Duración | 55 minutos |
| Asistentes | César Soto, Bastián Cáceres |

Temas en tabla

Discusión

- **Definir los procesos de interacción del cliente con el sistema**

Conclusiones/acuerdos

- **A partir de los requerimientos de sistema se definen los principales actores y las historias de usuario**

| Plan de acción | Responsable | Plazo |
|--|------------------------------|--------------|
| - Documentar roles e historias de usuario | César Soto – Bastián Cáceres | 15 de mayo |



Lingo's Talca – Minuta

de reunión

Correo:
bacaceres16@alumnos.utalca.cl
csoto16@alumnos.utalca.cl

Fono: +569 41198764
 +569 77601880

Minuta martes 18 de mayo de 2021 – 19:00 hrs (vía Microsoft Teams)

| | |
|----------------------|--|
| Convocada por | César Soto |
| Tema | Definición de arquitectura de sistema |
| Organizador | Bastían Cáceres |
| Duración | 41 minutos |
| Asistentes | César Soto, Bastían Cáceres, Fernando San Martín |

Temas en tabla

Discusión

- Revisión de arquitecturas de sistema

Conclusiones/acuerdos

- Se acuerda trabajar bajo la arquitectura modelo, vista y controlador

| Plan de acción | Responsable | Plazo |
|---|------------------------------|--------------|
| - Aplicar sugerencias por parte del profesor guía | César Soto – Bastían Cáceres | 25 de mayo |



Lingo's Talca – Minuta

de reunión

Correo:
bacaceres16@alumnos.otalca.cl
csoto16@alumnos.otalca.cl

Fono: +569 41198764
+569 77601880

Minuta miércoles 25 de mayo de 2021 – 17:00 hrs (presencial)

| | |
|----------------------|---|
| Convocada por | César Soto |
| Tema | Presentación de la propuesta inicial y feedback de la contraparte |
| Organizador | César Soto, Bastián Cáceres |
| Duración | 38 minutos |
| Asistentes | Taylor Burdett, César Soto, Bastián Cáceres |

Temas en tabla

| |
|---|
| Discusión |
| - Se presenta propuesta inicial a la contraparte |
| Conclusiones |
| - La contraparte da su visto bueno a la propuesta y se acuerda proceder con el desarrollo |

| Plan de acción | Responsable | Plazo |
|-----------------------|------------------------------|--------------|
| - Comenzar sprint 1 | César Soto – Bastián Cáceres | 27 de mayo |



Lingo's Talca – Minuta

de reunión

Correo:
bacaceres16@alumnos.otalca.cl
csoto16@alumnos.otalca.cl

Fono: +569 41198764
+569 77601880

Minuta martes 08 de junio de 2021 – 21:00 hrs (vía Discord)

| | |
|----------------------|---|
| Convocada por | Bastián Cáceres |
| Tema | Revisión de historias de usuario críticas |
| Organizador | Bastián Cáceres |
| Duración | 46 minutos |
| Asistentes | César Soto, Bastián Cáceres |

Temas en tabla

Discusión

- Se analizan historias de usuario poco definidas y su rol en el sistema

Conclusiones/acuerdos

- Se actualizan historias de usuario.

| Plan de acción | Responsable | Plazo |
|---|------------------------------|--------------|
| - Actualización de historias de usuario | César Soto – Bastián Cáceres | 17 de junio |



Lingo's Talca – Minuta

de reunión

Correo:
bacaceres16@alumnos.otalca.cl
csoto16@alumnos.otalca.cl

Fono: +569 41198764
 +569 77601880

Minuta martes 15 de junio de 2021 – 21:00 hrs (vía Discord)

| | |
|----------------------|--|
| Convocada por | César Soto |
| Tema | Identificación de requerimientos adicionales |
| Organizador | César Soto |
| Duración | 22 minutos |
| Asistentes | César Soto, Bastián Cáceres |

Temas en tabla

Discusión

- Análisis de sistema en base a objetivos planteados para la identificación de requerimientos adicionales o poco relevantes

Conclusiones

- Se actualizan requerimientos funcionales y no funcionales

| Plan de acción | Responsable | Plazo |
|---|------------------------------|--------------|
| - Actualización de lista de requerimientos de sistema | César Soto – Bastián Cáceres | 17 de junio |



Correo:
bacaceres16@alumnos.utalca.cl
csoto16@alumnos.utalca.cl

Fono: +569 41198764
+569 77601880

Lingo's Talca – Minuta

de reunión

Minuta martes 22 de junio de 2021 – 21:00 hrs (vía Discord)

| | |
|----------------------|-----------------------------------|
| Convocada por | Bastián Cáceres |
| Tema | Elaboración del diseño del modelo |
| Organizador | César Soto, César Soto |
| Duración | 3 horas 47 minutos |
| Asistentes | César Soto, Bastián Cáceres |

Temas en tabla



| |
|--|
| Discusión |
| - Se busca un diseño de sistema que cumpla con las características solicitadas por la contraparte |
| Conclusiones |
| - Se realiza el diseño de la base de datos al igual que la estructura general del portal con las principales interacciones con el cliente. |

| PLAN DE ACCIÓN | RESPONSABLE | PLAZO |
|---|------------------------------|--------------|
| - ELABORAR VERSIÓN INICIAL DEL PORTAL WEB | César Soto – Bastián Cáceres | 24 de junio |

Anexo 2: Pantallas de sistema

Lingo's
Language (+coffee)

🏠 👤 📱 📷 ☰



KAFFE

DISFRUTA DE UN COFFEE BREAK EN EL PRIMER PISO DEL LOCAL EN CAFÉ KAFFE, DONDE TE OFRECEMOS CAFÉ DE GRANO FRESCO Y LOS MEJORES PRODUCTOS PARA ACOMPAÑAR TU EXQUISITO CAFÉ Y TU TARDE DE TRABAJO

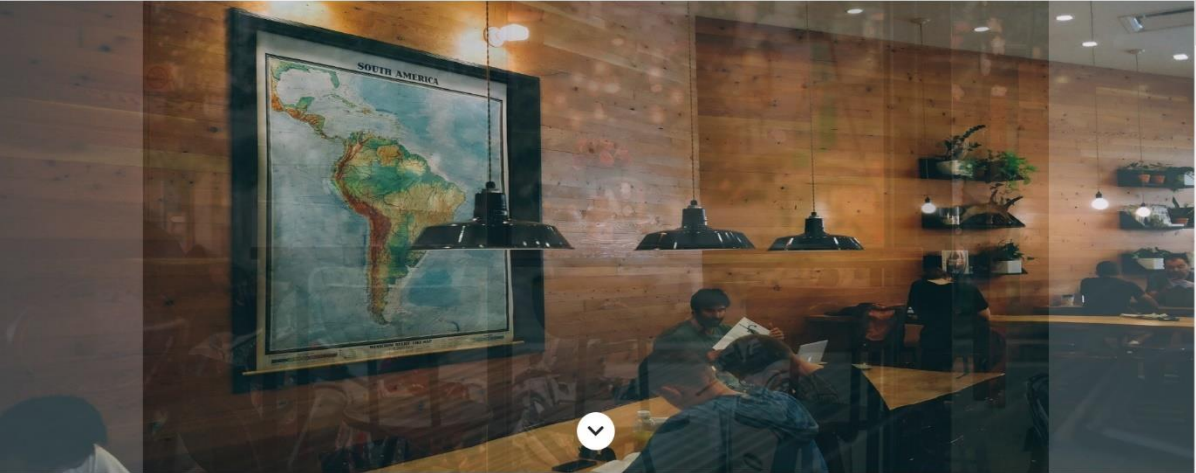
[VER LA CARTA](#)

CONTÁCTENOS

Nombre Apellido

Lingo's
Language (+coffee)

🏠 👤 📱 📷 ☰





SALAS

Lingo's
Language (+coffee)

🏠 👤 📱 📷 ☰


SALAS

| GRANDE | ESTANDAR |
|---|---|
|  |  |
| Salas de capacidad de 10 a 20 personas | Salas de capacidad de 5 a 10 personas |
| DESDE \$8,000 CLP | DESDE \$3,000 CLP |
| Ver detalles > | Ver detalles > |

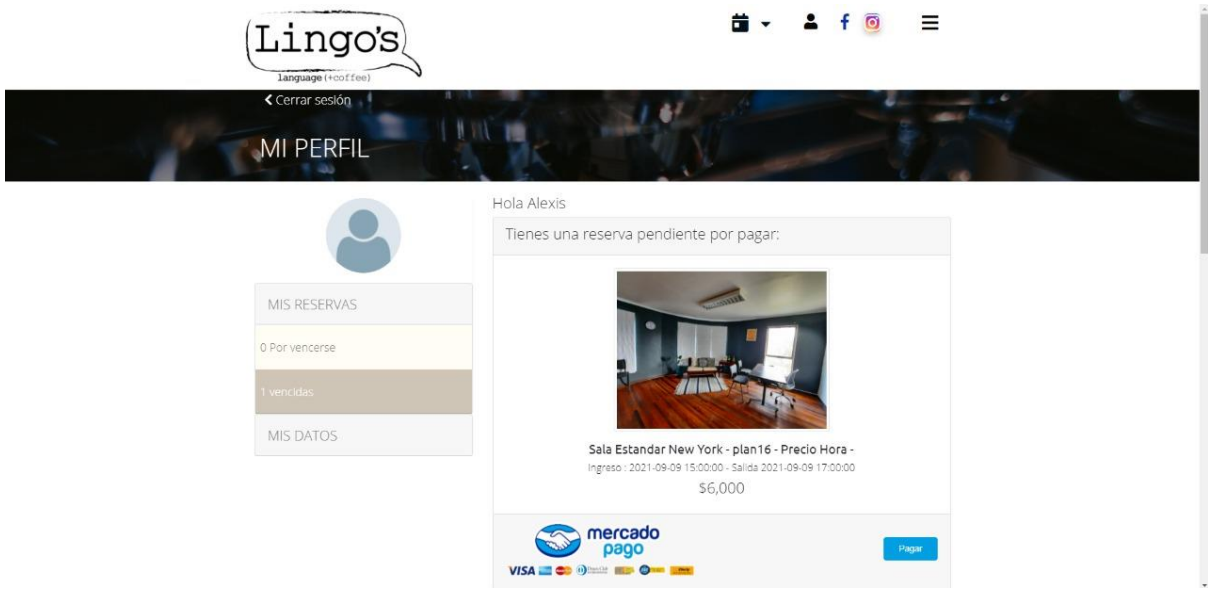
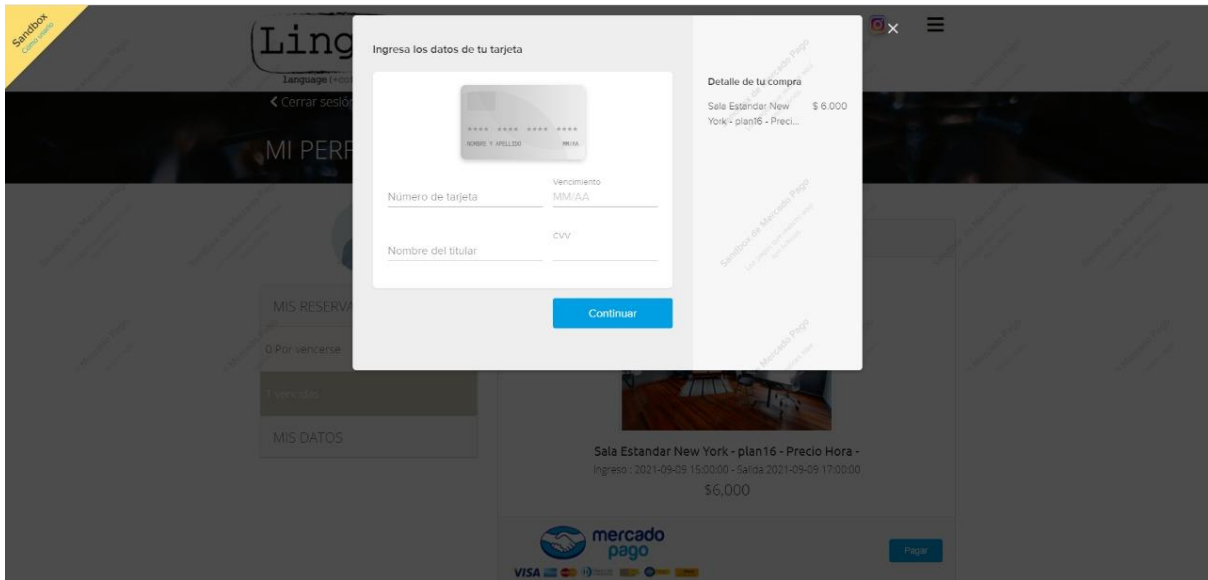
BIENVENIDO A LINGO'S!

Lingo's es un espacio de trabajo y aprendizaje que permite tanto a profesionales independientes como a empresas virtuales, contar con un espacio físico para el desarrollo de diversas actividades. Ofrecemos espacios

[local/host/reservas-lingos/habitacion-tipo-grande](#)



PLAN 16 HORAS MENSUALES



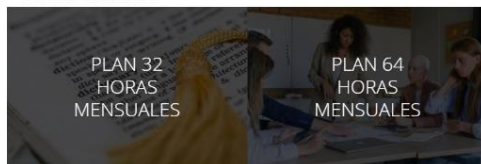
PLAN 64 HORAS MENSUALES

Plan que incluye 64 horas mensuales que podrán ser usadas al gusto del cliente como estime conveniente a lo largo de este periodo de tiempo.



BIENVENIDO A LINGO'S!

Lingo's es un espacio de trabajo y aprendizaje que permite tanto a profesionales independientes como a empresas virtuales, contar con un espacio físico para el desarrollo de diversas actividades. Ofrecemos espacios equipados: data, escritorios, mesas, pizarras para talleres, salas de reunión u oficinas según los requerimientos de nuestros clientes. Además ofrecemos el servicio de coffe break que entrega Kaffe Café ubicado en el primer nivel. ¿Quieres saber más sobre nuestros servicios? Contactanos!



[← Cerrar sesión](#)

MI PERFIL



Hola Alexis

Si necesita modificar o cancelar una reserva, favor escribirnos al WhatsApp -57 300 111 22 33

| # | Código | Nombre de Sala | Fecha de Ingreso | Fecha de Salida |
|---|--------|----------------|---------------------|---------------------|
| 1 | | LONDRES | 2021-08-09 17:00:00 | 2021-08-09 20:00:00 |

MIS RESERVAS

0 Por vencerse

1 vencidas

MIS DATOS

CONTÁCTENOS

Nombre

Apellido

Anexo 3: Historias de usuario

| | |
|---|--------------------|
| #1 | Autenticar usuario |
| Como usuario Quiero ingresar al sitio web Para poder pagar reservas | |
| Prioridad: Alta | Valor: 8 |

| | |
|--|--------------------|
| #2 | Autenticar usuario |
| Como usuario Quiero ingresar al sitio web Para ver historial de reservas | |
| Prioridad: Media | Valor: 5 |

| | |
|--|--------------------|
| #3 | Autenticar usuario |
| Como usuario Quiero ingresar al sitio web Para poder ver reservas vencidas | |
| Prioridad: Media | Valor: 6 |

| | |
|---|--------------------|
| #4 | Autenticar usuario |
| Como usuario Quiero ingresar al sitio web Para poder ver reservas próximas a vencer | |

| | |
|------------------|----------|
| Prioridad: Media | Valor: 3 |
|------------------|----------|

| | |
|---|----------|
| #5 | Contacto |
| Como usuario Quiero ingresar mis datos en el formulario Para contactar a la empresa | |
| Prioridad: Alta | Valor: 3 |

| | |
|--|----------------|
| #6 | Disponibilidad |
| Como usuario Quiero ingresar al sitio web Para poder ver disponibilidad de salas | |
| Prioridad: Alta | Valor: 8 |

| | |
|--|------------|
| #7 | Ver planes |
| Como usuario Quiero ingresar al sitio web Para poder ver los tipos de planes disponibles | |
| Prioridad: Alta | Valor: 7 |

| | |
|----|----------------|
| #8 | Contratar plan |
|----|----------------|

| | |
|---|----------|
| <p>Como usuario</p> <p>Quiero ingresar al sitio web</p> <p>Para poder contratar un plan mensual</p> | |
| Prioridad: Alta | Valor: 6 |

| | |
|---|----------|
| #9 | Carta |
| <p>Como usuario</p> <p>Quiero ingresar al sitio web</p> <p>Para poder ver los productos de la carta que se ofrecen en el café KAFFE</p> | |
| Prioridad: Media | Valor: 4 |