



FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

PELUQUERÍA: CORTE & COLOR

Autor(es):

Barsallo Gomez, Victor Manuel

Curso:

Diseño y arquitectura de software

Docente:

Guevara Jimenez, Jorge Alfredo

INDICE

Contenido

RESUMEN EJECUTIVO	5
ABSTRACT	6
CAPÍTULO 1. Introducción	7
1.1. IDENTIFICACIÓN DEL PROBLEMA	7
1.2. PLANTEAMIENTO DE SOLUCIÓN	7
1.3. CÓMO IMPLEMENTAR LA SOLUCIÓN	7
1.4. JUSTIFICACIÓN Y LIMITACIONES DE LA INVESTIGACIÓN	7
1.4.1. Justificación	7
1.4.2. Limitaciones	8
1.5. VENTAJAS DE LA SOLUCIÓN	8
1.6. DESVENTAJAS DE LA SOLUCIÓN	9
1.7. OBJETIVO GENERAL	9
1.8. OBJETIVOS ESPECÍFICOS	10
CAPÍTULO 2. MARCO TEÓRICO	11
2.1. Marco Teórico	11
2.2. Mapa de proceso	12
2.3. Diagrama de casos de uso	12
2.4. Modelo entidad relación	12
2.5. Plan de pruebas	12
2.6. Requisitos funcionales	13
2.7. Requisitos no funcionales	13
2.8. Especificación de casos de uso	13
2.9. Diagrama de actividad	14
2.10. Modelo M-V-C	14
2.11. MySQL	15
2.12. HTML5	15
2.13. CSS	16
2.14. JavaScript	16
2.15. JQuery	16
2.16. PATRONES ARQUITECTÓNICOS:	17
2.16.1. TUBERÍA Y FILTRO:	17
2.16.2. ARQUITECTURA EN CAPAS	17
2.17. Patrón CRC	17
2.18. Patrón SOA	17
2.19. MARCO METODOLÓGICO	18
CAPÍTULO 3. IMPLEMENTACIÓN DE LA SOLUCIÓN PLANTEADA	19
3.1. MAPA DE PROCESOS	19
3.2. DIAGRAMA DE CASOS DE USO	22
3.3. ESPECIFICACION DE CASOS DE USO	23
3.4. UNIDADES DE PROGRAMACIÓN	25
3.5. TABLAS ACME	26

3.6.	Arquitectura por capas	27
3.7.	Arquitectura de HW/SW	29
3.8.	Modelo E-R	30
3.9.	Modelo de tubería y filtros	30
3.10.	Esquema de Plan de Pruebas de la Iteración	31
3.10.1.	<i>Ejecución de Pruebas de la Iteración:</i>	31
3.11.	Esquema de Plan de Pruebas de la Iteración 2	37
1.1.	Esquema de arquitectura de Reposito del Proyecto	38
3.12.	Cuenta Github y Documentación	39
3.13.	DIAGRAMA DE SECUENCIAS	39
3.13.1.	<i>Caso de uso: Registrar usuario</i>	39
3.13.2.	<i>Caso de uso: Buscar y Seleccionar servicio</i>	39
3.13.3.	<i>Caso de uso: Buscar y Seleccionar horario</i>	40
3.13.4.	<i>Caso de uso: Buscar y Seleccionar Peluquero</i>	40
3.13.5.	<i>Caso de uso: Registrar Cita</i>	40
3.14.	Diagrama de subsistemas e interfaces	41
3.15.	Patrón CRC	41
3.16.	Patrón SOA	43
3.16.1.	<i>Gmail API</i>	43
3.16.2.	<i>Excel API</i>	43
CAPÍTULO 4.	CONCLUSIONES	44
CAPÍTULO 5.	RECOMENDACIONES.....	44
CAPÍTULO 6.	Referencias:	45

Índice de tablas

<i>Tabla 1: Tabla de requerimientos</i>	20
<i>Tabla 2: Plan de iteraciones</i>	21
<i>Tabla 3: Especificación del caso de uso “Registrar usuario”</i>	23
<i>Tabla 4: Especificación del caso de uso “Registrar cita”</i>	23
<i>Tabla 5: Especificación del caso de uso “Pagar cita”</i>	24
<i>Tabla 6: Planificación de unidades de programación</i>	25
<i>Tabla 7: Modelo ACME</i>	26
<i>Tabla 8: Entorno de desarrollo</i>	27
<i>Tabla 9: Entorno de calidad</i>	28
<i>Tabla 10: Entorno de producción</i>	28
<i>Tabla 11: Plan de pruebas de la interacción 1</i>	31
<i>Tabla 12: Patrón CRC - Modelo Servicio</i>	41
<i>Tabla 13: Patrón CRC - Controlador Servicio</i>	42
<i>Tabla 14: Patrón CRC - Vista Servicio</i>	42

Índice de Ilustraciones

<i>Ilustración 1: Mapa de Procesos</i>	19
<i>Ilustración 2: Diagrama de Casos de Uso</i>	22
<i>Ilustración 3: Arquitectura de capa de Producción</i>	29
<i>Ilustración 4: Arquitectura de HW/SW</i>	29
<i>Ilustración 5: Modelo E-R</i>	30
<i>Ilustración 6: Modelo de tubería y filtro</i>	30
<i>Ilustración 7: Repositorio de proyecto</i>	38
<i>Ilustración 8: Github y documentación</i>	39
<i>Ilustración 9: Diagrama de Secuencia1</i>	39
<i>Ilustración 10: Diagrama de Secuencia</i>	39
<i>Ilustración 11: Diagrama de Secuencia3</i>	40
<i>Ilustración 12: Diagrama de Secuencia4</i>	40
<i>Ilustración 13: Diagrama de Secuencia5</i>	40
<i>Ilustración 14: Patrón SOA (Gmail-API)</i>	43
<i>Ilustración 15: Patrón SOA (Excel-API)</i>	43

RESUMEN EJECUTIVO

La Peluquería Corte y Color, es una empresa dedicada a cortes, peinados, entre otras cosas. Esta empresa tiene un prometedor futuro ya que ofrece varios diseños y cortes más populares del momento; además de que ofrece una atención al cliente de primera. Sin embargo, no cuenta con un sistema que les ayude a optimizar el proceso de ventas y el pedido de citas. Tampoco cuenta con un registro de datos ordenado (tienen un registro de datos guardados de todo tipo de información en cuadernos o folders).

El propósito general de esta peluquería es crear relaciones con los clientes a través de una excelente calidad de servicio. Además, espera complacer a los clientes para que esta relación se convierta en el medio de comunicación ante otros clientes mediante las recomendaciones y buenas críticas del local.

Y es por eso que el propósito del desarrollo de este software es ser utilizado como una herramienta para ayudar a alcanzar este objetivo; optimizando los procesos de ventas o llevar un registro de mercaderías usando una base de datos, es como el software podrá aportar al progreso de la peluquería

ABSTRACT

The Hairdresser Cut and Color, is a company dedicated to cuts, hairstyles, among other things. This company has a promising future as it offers several designs and most popular cuts of the moment; In addition to offering first-class customer service. However, it does not have a system that helps them optimize the sales process and the request for appointments. Nor does it have an orderly data record (they have a data record saved of all kinds of information in notebooks or folders).

The general purpose of this hairdresser is to create relationships with customers through excellent quality of service. In addition, he hopes to please customers so that this relationship becomes the means of communication with other customers through recommendations and good reviews of the place.

And that is why the purpose of the development of this software is to be used as a tool to help achieve this goal; optimizing sales processes or keeping track of merchandise using a database, this is how the software can contribute to the progress of the hairdresser.

CAPÍTULO 1. Introducción

1.1. IDENTIFICACIÓN DEL PROBLEMA

Los dispositivos ofrecidos se encuentran en un sector específico del mercado, por lo cual no está dirigido a todo el público, esto dificulta la expansión de la bolsa de clientes.

Al ser una PYME, dicha empresa desconoce parcialmente sus objetivos en un mediano-largo plazo, lo cual infiere una dificultad en el planteamiento “MISIÓN-VISIÓN”.

1.2. PLANTEAMIENTO DE SOLUCIÓN

La solución que proponemos es realizar una aplicación por medio de la web que contenga un sistema para poder gestionar mucho mejor las citas, además de que se optimicen los procesos relacionados a estas. Dicho sistema tendrá varias opciones tales como una pequeña reseña de la empresa, a que se dedica principalmente, su misión, visión, colaboradores, entre otros. Tendrá una opción de servicios, en la cual se podrá visualizar y elegir entre los variados que ofrece esta empresa. Estas opciones se irán modificando conforme se realice el proyecto. Para tener conocimiento de lo que se está comprando se tendrá una opción de carrito en la cual se tendrá más detalle de cuantos servicios querrá adquirir el cliente, así tanto para ellos como para la empresa se podrá realizar y gestionar las ventas/citas de manera mucho más rápida y eficiente.

1.3. CÓMO IMPLEMENTAR LA SOLUCIÓN

La ubicación de la empresa en el mercado digital se realizará mediante una página web alojada en un hosting con la identidad de la empresa. Implementaremos tanto el proyecto como la base de datos en los servidores del proveedor.

1.4. JUSTIFICACIÓN Y LIMITACIONES DE LA INVESTIGACIÓN

1.4.1. Justificación

- La digitalización de una empresa nos permite que la gestión de las citas sea mucho más eficiente.
- La modernización de la empresa para la satisfacción y beneficio tanto para el cliente como la empresa.
- La importancia de tener modelado de un sistema para poder copar las necesidades de la empresa para con el cliente.

- Se diseñará un sitio web conectado a una base de datos SQL, con la cual almacenaremos datos necesarios para esta organización
- Se adaptará a un computador con características básicas para un buen funcionamiento.
- La página web es la mejor forma de posicionar la empresa digitalmente, acercándola más a futuros clientes.
- La página web presenta una forma más cómoda y rápida de visualizar los productos y servicios ofrecidos por la empresa.
- La implementación de un servicio informático provee datos esenciales de los clientes y su comportamiento que puede aumentar los beneficios de la empresa.

1.4.2. Limitaciones

- Por la falta de tiempo y con solo tener algunos meses para realizar proyecto no se podrá realizar un software completo para todos los procesos de digitalización que necesita la empresa por lo tanto solo nos basaremos en la gestión de citas, ya que es el principal problema que se presencia en la empresa.
- El correcto diseño de una página web tanto en visibilidad, adaptabilidad y en seguridad requiere altos conocimientos en diferentes sectores de la programación.
- Implementación de seguridad aplicada para las compras de productos y reservaciones de citas en la página.
- Negatividad por parte de la empresa para invertir en novedades tecnológicas.
- No se cuenta con personal capacitado para el uso del sistema.

1.5. VENTAJAS DE LA SOLUCIÓN

- Expansión del flujo de citas que se tiene
- Oportunidad de digitalización para así que sea mucho más competitiva en el campo laboral actual.
- Buena referencia de la empresa por parte del cliente y se sentirá mucho más

confiado al optar por esta.

- Mejor organización para y con los clientes
- No se necesitará la instalación de ningún programa localmente, de esta forma, se ubicará todo el contenido en un servicio controlado.
- Controlar los datos, el comportamiento y preferencias de los clientes para así aumentar el beneficio de la empresa.
- Resulta menos costoso ya que se ahorran recursos, materiales y servicios humanos necesarios para que el sistema de trabajo pueda funcionar.
- Ofrece una mayor facilidad en la migración a otro proveedor web, en caso fuere necesario.
- Mayor publicidad para la peluquería

1.6. DESVENTAJAS DE LA SOLUCIÓN

- Pago de un alojamiento web (hosting) mensualmente.
- Al estar alojado en la web es más sucesible a ataques informáticos.
- Investigación de todos los requerimientos legales en una página web.
- Mediante el uso, la dependencia de un grupo informático será mayor ante posibles dudas y problemas.
- Costos para la implementación en la parte del hardware
- Posibles fallas con la página, ya que en algunos momentos podría suceder desconexiones/caídas.
- Asesoramiento constante en el manejo del nuevo software.

1.7. OBJETIVO GENERAL

- Realizar un sistema de gestión de citas para la empresa Peluquería Corte y Color, lo cual permitirá organizar la gestión de citas.

1.8. OBJETIVOS ESPECÍFICOS

- Diseñar el sistema utilizando una metodología de desarrollo de sistemas orientados a web.
- Diseñar un sistema que cumpla todas las especificaciones requeridas por la empresa, garantizando estabilidad y calidad en la presentación del servicio.
- Implantar el sistema para optimizar el control de la información.
- Proceso de supervisión más rápida.
- Identificar de forma eficaz y rápida los datos de la empresa.

CAPÍTULO 2. MARCO TEÓRICO

2.1. Marco Teórico

En la actualidad, toda empresa debería estar digitalizada para que esté al alcance de más personas y se faciliten los procesos generados por esta. La Empresa “Corte y Color” ha mantenido un crecimiento, pero teme por un estancamiento, dado que no cuenta con las herramientas necesarias para seguir impulsando el dicho progreso, corriendo peligro de ser afectado por las competencias que se han generado durante el tiempo, beneficiándose de ello.

Para solucionar esta problemática se desarrolló e implementó una página web para dicha empresa que permite al cliente buscar el servicio requerido, ver los horarios disponibles y escoger su peluquero, además de poder pagar la cita mediante o una tarjeta o reservarla y pagarla presencialmente en efectivo. De esta manera, tanto los clientes como el mismo dueño del local podrán gestionar mejor las citas solicitadas.

Para realizar este objetivo daremos uso de los modelados de software ya que es muy importante, tal como lo indica (Booch Grady, Rumbaugh James, Jacobson I., “un modelo de un sistema de software ayuda a los desarrolladores a explorar varias arquitecturas y soluciones de diseño fácilmente, antes de escribir el código. Un buen lenguaje de modelado permite que el diseñador consiga la arquitectura correcta antes de que comience el diseño detallado”).

Para ello, haremos uso de diferentes diagramas, como el mapa de proceso, plan de iteraciones, MVP, casos de uso, especificaciones de casos de uso, entre otros.

En este proyecto se implementa el diseño de software que permite representar los aspectos más importantes en el desarrollo de un sistema de trabajo, con este uso se implementará tipos de modelos de procesos en la empresa, en la cual se gestiona la información de los productos y servicios, así como, también, se especifica los procesos del modelo, solicitando los usos del producto y los actores que realizan las acciones en el sistema de trabajo mejorando el buen uso del software y tener una mayor finalidad de requerimiento del producto y ser más factible en la empresa.

Se utilizarán diferentes softwares para cada etapa del desarrollo del sistema. Tales como Balsamiq para un primer acercamiento a la vista general de la página, MYSQL Workbench como gestor de la base de datos, que luego se exportará al dominio requerido. La

implementación de la página se realizará con diferentes frameworks de desarrollo web: Spring, Angular, Ajax; así como HTML, CSS, PHP y Javascript.

2.2. Mapa de proceso

Es un diagrama de valor que representa, a manera de inventario gráfico, los procesos de una organización en forma interrelacionada. El mapa de procesos recoge la interrelación de todos los procesos que realiza una organización.

2.3. Diagrama de casos de uso

Es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. El Lenguaje de Modelado Unificado (UML), define una notación gráfica para representar casos de uso llamada modelo de casos de uso. (Vega M., 2010)

2.4. Modelo entidad relación

Un modelo entidad relación es un diagrama que es una herramienta que nos ayudara a realizar nuestro modelado de datos de nuestro software dicho modelo nos permite representar las entidades las cuales van a actuar en nuestro sistema. “Una clase de una entidad en el modelo ER es el nombre de una "cosa" u objeto de importancia para una organización o negocio, ya sea real o imaginaria, cuya información se necesita conocer o mantener (Barker, 1989). Puede ser una cosa concreta, como una persona o la ubicación geográfica; o puede ser una abstracción como “Consejo Comunal miembros” o” Rol en el Consejo Comunal. En UML, un subconjunto del concepto de "clase" puede utilizarse siempre y cuando se entienda dentro del modelo ER. El nombre de la clase entidad es una particularidad y refiere a una instancia de esa clase (Andrade, 2012)”

2.5. Plan de pruebas

El plan de pruebas del software es una herramienta que utilizaremos para comprobar la funcionalidad del software si en todo caso es factible o si podemos realizar mejorar o también si en todo caso tenga un error poder corregirlo y pulirlo antes de entregar el sistema con su completa funcionalidad “El plan de pruebas debe comprender los guiones para probar los componentes del software. Esto hace parte del sistema de calidad del proyecto.” (Parra, 2011).

2.6. Requisitos funcionales

Un requisito funcional es lo que nos permite definir una funcionalidad que forma parte del sistema que vamos a realizar. “Los requerimientos funcionales para un sistema refieren lo que el sistema debe hacer, tales requerimientos dependen del tipo de software que se esté desarrollando, de los usuarios esperados del software y del enfoque general que adopta la organización cuando se escriben los requerimientos. Al expresarse como requerimientos del usuario, los requerimientos funcionales se describen por lo general de forma abstracta que entiendan los usuarios del sistema. Sin embargo, requerimientos funcionales más específicos del sistema detallan las funciones del sistema, sus entradas y salidas, sus excepciones, etcétera. Los requerimientos funcionales del sistema varían desde requerimientos generales que cubren lo que tiene que hacer el sistema, hasta requerimientos muy específicos que reflejan maneras locales de trabajar o los sistemas existentes de una organización” (Sommerville ,2011).

2.7. Requisitos no funcionales

Requisitos que no describen información a guardar, ni funciones a realizar, sino características de funcionamiento, por eso suelen denominarse Atributos de calidad de un sistema. Queda entonces el requisito no funcional, que son las restricciones o condiciones que impone el cliente al programa que necesita, por ejemplo, el tiempo de entrega del programa, el lenguaje o la cantidad de usuarios.

“Requerimientos que no se refieren específicamente a la funcionalidad de un sistema. Se imponen restricciones sobre el producto que se está desarrollando y el proceso de desarrollo, y que especifican restricciones externas que el producto debe cumplir”. (Sommerville ,2011).

2.8. Especificación de casos de uso

La especificación del caso de uso es un cuadro que nos ayudará nuevamente a entender el proceso que estamos realizando dentro de un caso de uso. Está conformada por los siguientes puntos:

- Nombre: Nombre asignado que justifica la existencia.
- Actores: Entidad externa al sistema software que se especifica y que interactúa con él.

- Objetivo: La meta que se quiere llegar al dar acceso al caso.
- Pre-condición: Lista cualquier actividad que tiene que ocurrir antes de que el software comience.
- Post-condición: Describe el estado del sistema al finalizar la ejecución del software.
- Escenario Básico: Proporciona una descripción detallada de las elecciones del usuario y respuestas del sistema.
- Escenario Alternativo: Proporciona una descripción de posibles escenarios, diferencias en la secuencia de pasos del flujo Básico.
- Prioridad: Indica la importancia de su implementación.

2.9. Diagrama de actividad

Muestra el flujo de actividades, siendo una actividad una ejecución general entre los objetos que se está ejecutando en un momento dado dentro de una máquina de estados, el resultado de una actividad es una acción que producen un cambio en el estado del sistema o la devolución de un valor.

“Los diagramas de actividad intentan mostrar las actividades que incluyen un proceso de sistema, así como el flujo de control de una actividad a otra. El inicio de un proceso se indica con un círculo lleno; el fin, mediante un círculo lleno dentro de otro círculo. Los rectángulos con esquinas redondeadas representan actividades, esto es, los subprocesos específicos que hay que realizar” (Sommerville, 2015).

2.10. Modelo M-V-C

El patrón de diseño de modelo-vista-controlador (MVC) especifica que una aplicación consta de un modelo de datos, de información de presentación y de información de control. El patrón requiere que cada uno de estos elementos esté separado en distintos objetos.

El modelo (por ejemplo, la información de datos) contiene únicamente los datos puros de aplicación; no contiene lógica que describe cómo pueden presentarse los datos a un usuario.

La vista (por ejemplo, la información de presentación) presenta al usuario los datos del modelo. La vista sabe cómo acceder a los datos del modelo, pero no sabe el significado de estos datos ni lo que el usuario puede hacer para manipularlos.

Por último, el controlador (por ejemplo, la información de control) está entre la vista y el modelo. Escucha los sucesos desencadenados por la vista (u otro origen externo) y ejecuta la reacción apropiada a estos sucesos. En la mayoría de los casos, la reacción es llamar a un método del modelo. Puesto que la vista y el modelo están conectados a través de un mecanismo de notificación, el resultado de esta acción se reflejará automáticamente en la vista.

“El patrón Modelo-Vista-Controlador (MVC) surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones....

El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página; el modelo es el Sistema de Gestión de Bases de Datos y la Lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista”. (Fernández Y. y Díaz Y., 2012)

2.11. MySQL

MySQL es un sistema de administración de bases de datos relational (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. MySQL compite con sistemas RDBMS propietarios conocidos, como Oracle, SQL Server y DB2. “MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger y hacer volcados de datos. Puede desarrollar sus propias aplicaciones de base de datos en la mayor parte de los lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos, incluyendo algunos de los que probablemente no ha oído nunca hablar.” (Gilfillan 2003, La biblia de MySQL) MySQL utiliza el lenguaje de consulta estructurado (SQL). Se trata del lenguaje utilizado por todas las bases de relacionales, que presentaremos en una sección posterior. Este lenguaje permite crear bases de datos, así como agregar, manipular y recuperar datos en función de criterios específicos.

2.12. HTML5

En los primeros días de la Red Mundial (entre 1990 y 1996), los sitios web consistían en poco más que un conjunto de archivos de hipertexto vinculados que presentaban la información con el empleo de texto y gráficas limitadas. Al pasar el tiempo, el aumento de HTML por medio de herramientas de desarrollo permitió a los ingenieros de la web brindar

capacidad de cómputo junto con contenido de información. Con esto habían nacido los sistemas y aplicaciones basados en web. (Pressman, 2010).

2.13. CSS

Tecnología utilizada para dotar la presentación de un documento estructurado escrito en un lenguaje de marcado.² Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etcétera. También permite aplicar estilos no visuales, como las hojas de estilo auditivas.

2.14. JavaScript

Javascript es un lenguaje interpretado usado para múltiples propósitos, pero solo considerado como un complemento hasta ahora. Una de las innovaciones que ayudó a cambiar el modo en que vemos Javascript fue el desarrollo de nuevos motores de interpretación, creados para acelerar el procesamiento de código. La clave de los motores más exitosos fue transformar el código Javascript en código máquina para lograr velocidades de ejecución similares a aquellas encontradas en aplicaciones de escritorio. Esta mejorada capacidad permitió superar viejas limitaciones de rendimiento y confirmar el lenguaje Javascript como la mejor opción para la web. Para aprovechar esta prometedora plataforma de trabajo ofrecida por los nuevos navegadores, Javascript fue expandido en relación con portabilidad e integración. A la vez, interfaces de programación de aplicaciones (APIs) fueron incorporadas por defecto en cada navegador para asistir al lenguaje en funciones elementales. Estas nuevas APIs (como Web Storage, Canvas, y otras) son interfaces para librerías incluidas en navegadores. “La idea es hacer disponible poderosas funciones a través de técnicas de programación sencillas y estándares, expandiendo el alcance del lenguaje y facilitando la creación de programas útiles para la web.”(Gauchat, 2012, El gran libro de HTML5, CSS3 y Javascript,).

2.15. JQuery

JQuery es una librería de JavaScript (JavaScript es un lenguaje de programación muy usado en desarrollo web). Esta librería de código abierto simplifica la tarea de programar en JavaScript y permite agregar interactividad a un sitio web sin tener conocimientos del lenguaje. Basados en esta librería, existe una infinita cantidad de plugins (gratis y pagos) creados por desarrolladores de todo el mundo. Estos plugins resuelven situaciones concretas dentro del maquetado de un sitio, por ejemplo: un menú responsive, una galería de fotos, un carrousel de imágenes, un slide, un header que cambia de tamaño, el

deslizamiento del scroll al hacer clic en un botón, la transición entre páginas y miles de efectos más. (Chuburu Laura, 2018).

2.16. PATRONES ARQUITECTÓNICOS:

2.16.1. TUBERÍA Y FILTRO:

El procesamiento de datos en un sistema se organiza de forma que cada componente de procesamiento (filtro) sea discreto y realice un tipo de transformación de datos. Los datos fluyen (como en una tubería) de un componente a otro para su procesamiento. (Somerville,2011).

2.16.2. ARQUITECTURA EN CAPAS

El patrón de arquitectura en capas es otra forma de lograr separación e independencia, cada una se apoya sólo en las facilidades y los servicios ofrecidos por la capa inmediatamente debajo de ella. Este enfoque en capas soporta el desarrollo incremental de sistemas. La arquitectura también es cambiable y portátil. En tanto, su interfaz no varíe, una capa puede sustituirse por otra equivalente. Más aún, cuando las interfaces de capa cambian o se agregan nuevas facilidades a una capa, sólo resulta afectada la capa adyacente. (Somerville, 2011).

2.17. Patrón CRC

El modelado clase-responsabilidad-colaborador (CRC) proporciona una manera sencilla de identificación y organización de las clases que son relevantes para los requerimientos de un sistema o producto. Ambler [Amb95] describe el modelado CRC en la siguiente forma: “Un modelo CRC en realidad es un conjunto de tarjetas índice estándar que representan clases. Las tarjetas se dividen en tres secciones. En la parte superior de la tarjeta se escribe el nombre de la clase, en la parte izquierda del cuerpo se enlistan las responsabilidades de la clase y en la derecha, los colaboradores” (Pressman, 2010)

2.18. Patrón SOA

SOA es un estilo de Arquitectura de Software basado en la definición de servicios reutilizables, con interfaces públicas bien definidas, donde los proveedores y consumidores de servicios interactúan en forma desacoplada para realizar los procesos de negocio. Como una arquitectura de software, SOA, puede consistir en una combinación de tecnologías, productos, API's, extensiones de apoyo a la infraestructura y muchas otras partes.

2.19. MARCO METODOLÓGICO

La metodología empleada para el proceso de implementación y desarrollo de la página web para la peluquería “Corte y Color” se basó en el transcurso del curso “Diseño y Arquitectura de Software”.

Concentrando la metodología seguida por el curso, la cual está conformada por las siguientes etapas

Etapas:

- Búsqueda de empresa/entidad con la problemática
- Consulta y entrevista con los encargados sobre las actividades que se realiza diariamente y sus problemáticas (recopilación de información).
- Identificación de los requisitos funcionales.
- Realización del mapa de proceso
- Realización del plan de iteraciones
- Identificación los actores
- Identificación los casos de uso
- Realización del diagrama de casos de uso
- Especificación de cada caso de uso
- Planteamiento de una solución en base a una página web
- Diseñar el sistema a partir de las necesidades y el sistema de trabajo de la empresa.
- Contratar el servicio de un proveedor de hosting
- Comprar un certificado SSL (para trabajar bajo el protocolo HTTPS).
- Mediante el diseño del sistema informático implementar una base de datos.
- Importar todos los datos necesarios de la empresa tanto en la base de datos, como en la página (usuarios, barcos, dispositivos, etc.)
- Implementar la estructura y funcionamiento de la página web.
- Subir el proyecto al hosting.

CAPÍTULO 3. IMPLEMENTACIÓN DE LA SOLUCIÓN PLANTEADA

3.1. MAPA DE PROCESOS

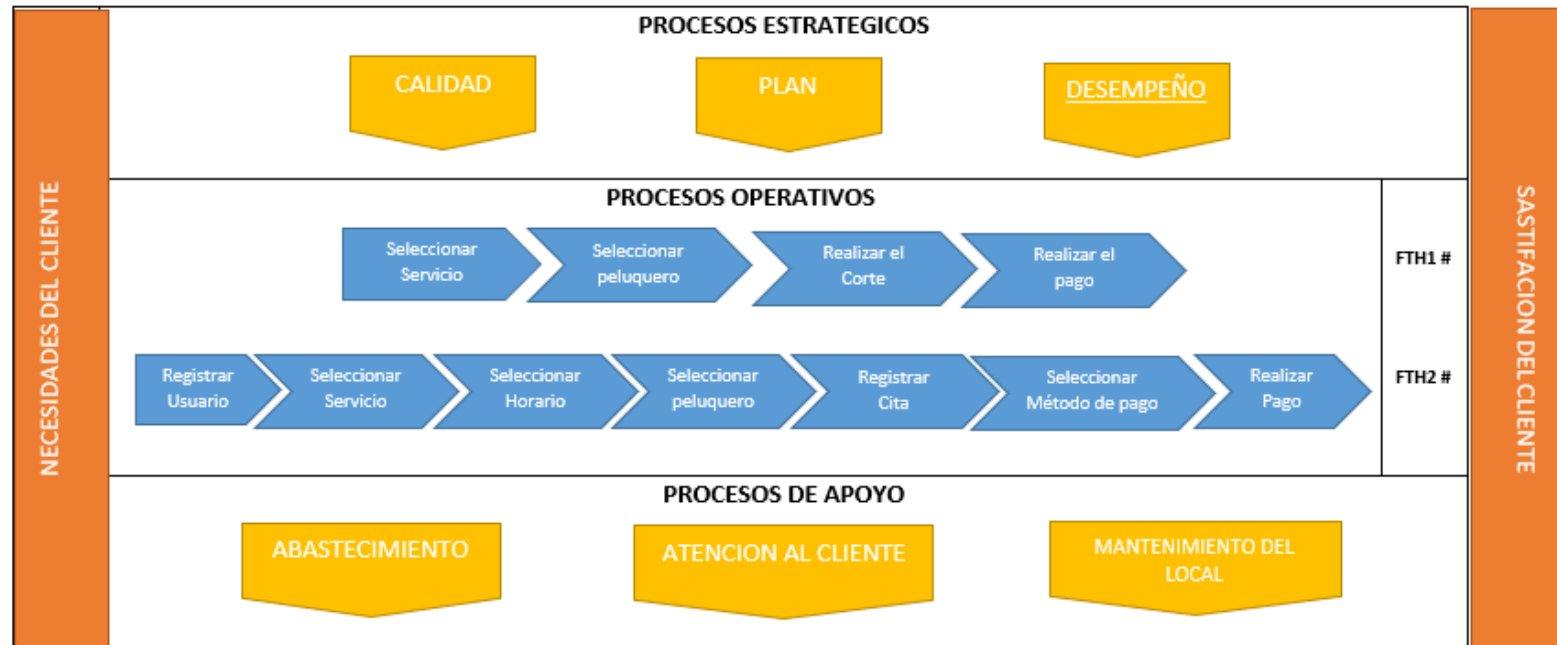


Ilustración 1: Mapa de Procesos

PLAN DE ITERACIONES

Tabla 1: Tabla de requerimientos

N°	Requerimientos	Casos de uso
1	El usuario se registra a la página web.	Registrar Usuario
2	El usuario busca el servicio que requiere	Buscar Servicio
3	El usuario busca el horario que se le acomode mejor	Buscar Horarios
4	El usuario selecciona el servicio que busco	Seleccionar Servicio
5	El usuario selecciona el horario que busco	Seleccionar Horarios
6		Seleccionar Peluquero
7	El usuario registra su cita	Registrar Cita
8	El usuario paga su cita	Pagar cita
9	El usuario selecciona el método de pago	Seleccionar método de pago
10	Se emite el comprobante de pago al usuario y se le envía por correo	Emitir comprobante de pago

Tabla 2: Plan de iteraciones

R. F	N° ITERACION	META	VALOR
Registrar Usuario	1	Reservar por Internet	Que el usuario pueda tener un perfil en la página web, desde ahí realizar sus citas, seleccionando su servicio, peluquero y horario.
Buscar Servicio			
Buscar Horarios			
Buscar Peluquero			
Seleccionar Servicio			
Seleccionar Horarios			
Seleccionar Peluquero			
Registrar Cita			
Pagar cita			
Seleccionar Peluquero de Preferencia	2	Comodidad	Que el usuario pueda solicitar al peluquero de preferencia
Seleccionar Promociones	3	Beneficios	Que el usuario pueda acceder a las promociones que presentemos y medios de pago.
Seleccionar método de pago			
Emitir comprobante de pago			

3.2. DIAGRAMA DE CASOS DE USO

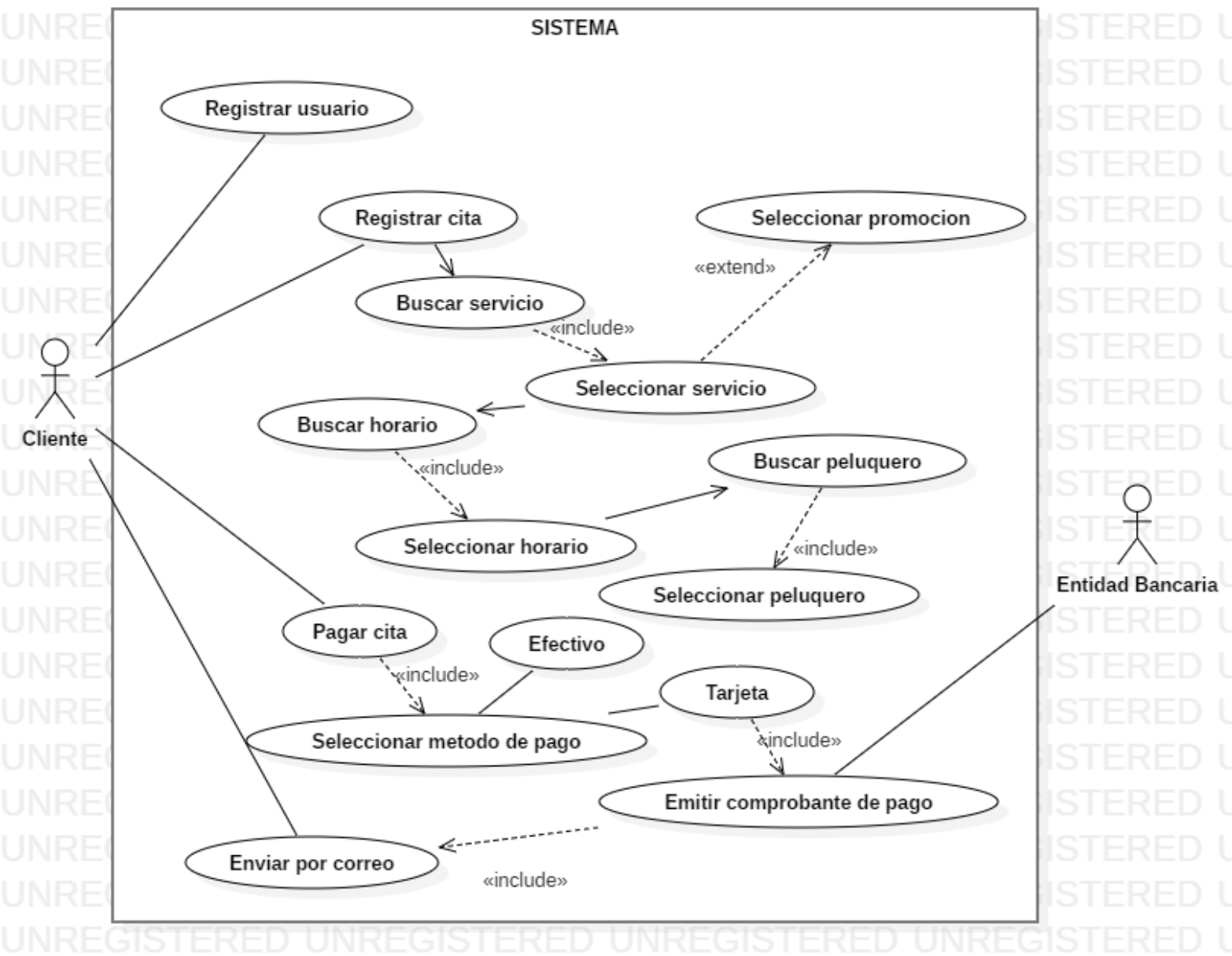


Ilustración 2: Diagrama de Casos de Uso

3.3. ESPECIFICACION DE CASOS DE USO

Tabla 3: Especificación del caso de uso “Registrar usuario”

ID	ITEM	DESCRIPCION
1	Nombre	REGISTRAR USUARIO
2	Actores	Cliente
3	Objetivo	Registrar un usuario en el sistema correctamente
4	Pre-condición	Ingresar a la página web
5	Post-condición	Usuario registrado
6	Escenario básico	Ingresar a la página web
		Seleccionar la opción "Registrarme"
		Rellenar los datos requeridos para el registro
		El sistema valida los datos
		El sistema guarda la información en la base de datos
		Registro de usuario satisfactorio
7	Escenario alternativo	Los datos ingresados en los campos son incorrectos
		El usuario ya existe
		El correo ya está registrado en el sistema
		Página web sufre una caída o se expira el tiempo para registro
8	Prioridad	1 / Reservar por internet

Tabla 4: Especificación del caso de uso “Registrar cita”

ID	ITEM	DESCRIPCION
1	Nombre	REGISTRAR CITA
2	Actores	Cliente
3	Objetivo	Registrar una cita en el sistema correctamente
4	Pre-condición	Usuario logueado
5	Post-condición	Cita registrada
6	Escenario básico	Loguearse en la página web
		Seleccionar servicio
		Seleccionar horario
		Seleccionar peluquero
		Ingresar una promoción (si existe en ese momento)
		El sistema valida los datos
		El sistema guarda la información en la base de datos
		Registro de cita satisfactorio
7	Escenario alternativo	Sesión no pudo iniciarse
		El servicio no está disponible
		El horario no está disponible
		El peluquero no está disponible
		Página web sufre una caída
8	Prioridad	1 / Reservar por internet

Tabla 5: Especificación del caso de uso “Pagar cita”

ID	ITEM	DESCRIPCION
1	Nombre	Pagar cita
2	Actores	Cliente
3	Objetivo	Registrar un usuario en el sistema correctamente
4	Pre-condición	Usuario selecciona todas las opciones requeridas para una cita
5	Post-condición	Cita pagada correctamente
6	Escenario básico	Usuario selecciona el botón pagar
		Se muestra los metodos de pago
		Si se selecciona vía efectivo:
		La cita se reserva y se pagará de manera presencial
		Si se selecciona vía tarjeta:
		La página redirecciona a la página de pago seguro
		Usuario proporciona datos de la tarjeta
		Los datos de la tarjeta son validados por el sistema
		La cita se reserva y fue pagada satisfactoriamente
7	Escenario alternativo	Sesión no pudo iniciarse
		La página de pago seguro no está operativa del todo
		Los datos de la tarjeta son incorrectos
		La tarjeta no cuenta con saldo suficiente
		Página web sufre una caída
		El tiempo que otorga el sistema pagos expira
8	Prioridad	1 / Reservar por internet

3.4. UNIDADES DE PROGRAMACIÓN

Tabla 6: Planificación de unidades de programación

N° de Interacción	R.F	Tipo de clase	Nombre de Clase	F. Inicio	F. Fin
1	Registrar Usuario	Modelo	M - Registrar Usuario	30/04/2020	05/05/2020
		Vista	V - Registrar Usuario	30/04/2020	05/05/2020
		Controlador	C - Registrar Usuario	30/04/2020	05/05/2020
	Buscar y Seleccionar Servicio	Modelo	M- Buscar y Seleccionar Servicio	30/04/2020	05/05/2020
		Vista	V - Buscar y Seleccionar Servicio	30/04/2020	05/05/2020
		Controlador	C - Buscar y Seleccionar Servicio	30/04/2020	05/05/2020
	Buscar y Seleccionar Horario	Modelo	M-Buscar y Seleccionar Horario	30/04/2020	05/05/2020
		Vista	V-Buscar y Seleccionar Horario	30/04/2020	05/05/2020
		Controlador	C-Buscar y Seleccionar Horario	30/04/2020	05/05/2020
	Buscar y Seleccionar Peluquero	Modelo	M- Buscar y Seleccionar Peluquero	30/04/2020	05/05/2020
		Vista	V - Buscar y Seleccionar Peluquero	30/04/2020	05/05/2020
		Controlador	C - Buscar y Seleccionar Peluquero	30/04/2020	05/05/2020
	Registrar Cita	Modelo	M - Registrar Cita	30/04/2020	05/05/2020
		Vista	V - Registrar Cita	30/04/2020	05/05/2020
		Controlador	C - Registrar Cita	30/04/2020	05/05/2020
	Pagar Cita	Modelo	M - Pagar Cita	30/04/2020	05/05/2020
		Vista	V - Pagar Cita	30/04/2020	05/05/2020
		Controlador	C - Pagar Cita	30/04/2020	05/05/2020
2	Seleccionar Peluquero de Preferencia	Modelo	M- Seleccionar Peluquero de Preferencia	30/04/2020	05/05/2020
		Vista	V- Seleccionar Peluquero de Preferencia	30/04/2020	05/05/2020
		Controlador	C- Seleccionar Peluquero de Preferencia	30/04/2020	05/05/2020
3	Seleccionar Promociones	Modelo	M- Seleccionar Promociones	30/04/2020	05/05/2020
		Vista	V- Seleccionar Promociones	30/04/2020	05/05/2020

	Seleccionar Método de Pago	Controlador	C- Seleccionar Promociones	30/04/2020	05/05/2020
		Modelo	M- Seleccionar Método de Pago	30/04/2020	05/05/2020
		Vista	V- Seleccionar Método de Pago	30/04/2020	05/05/2020
		Controlador	M- Seleccionar Método de Pago	30/04/2020	05/05/2020
	Emitir Comprobante de Pago	Modelo	M- Emitir Comprobante de Pago	30/04/2020	05/05/2020
		Vista	V- Emitir Comprobante de Pago	30/04/2020	05/05/2020
		Controlador	C-Emitir Comprobante de Pago	30/04/2020	05/05/2020

3.5. TABLAS ACME

Tabla 7: Modelo ACME

ID	TABLA	Incremento	Lista de Procesos Almacenados
1	Registrar Usuario (RU)	1	SP_A_TABLA_RU
2		1	SP_C_TABLA_RU
3		1	SP_M_TABLA_RU
4		1	SP_E_TABLA_RU
5	Buscar y Seleccionar Servicio (BSS)	1	SP_A_TABLA_BSS
6		1	SP_C_TABLA_BSS
7		1	SP_M_TABLA_BSS
8		1	SP_E_TABLA_BSS
9	Buscar y Seleccionar Peluquero (BSP)	1	SP_A_TABLA_BSP
10		1	SP_C_TABLA_BSP
11		1	SP_M_TABLA_BSP
12		1	SP_E_TABLA_BSP
13	Buscar y Seleccionar Horario (BSH)	1	SP_A_TABLA_BSH
14		1	SP_C_TABLA_BSH
15		1	SP_M_TABLA_BSH
16		1	SP_E_TABLA_BSH
17	Registrar Cita (RC)	1	SP_A_TABLA_RC
18		1	SP_C_TABLA_RC
19		1	SP_M_TABLA_RC
20		1	SP_E_TABLA_RC
21	Pagar Cita (PC)	1	SP_A_TABLA_PC
22		1	SP_C_TABLA_PC
23		1	SP_M_TABLA_PC
24		1	SP_E_TABLA_PC

3.6. Arquitectura por capas

Tabla 8: Entorno de desarrollo

Ventas por internet	Comodidad	Beneficios
Corte y Color		
Spring tools suite 4 y GITHUB <p>Se usará el programa Spring tools suite 4 para la programación del software.</p> 		
Lenguaje de programación: <ul style="list-style-type: none"> - Java - Spring framework 4 - Thymeleaf - Spring boot devtools - Spring data JPA 		
BASE DE DATOS: SQL Server <p>Se usará el programa MySQL para el óptimo funcionamiento de la base de datos en la página web desarrollada.</p> 		
Sistema Operativo Windows <p>Se usará el sistema operativo Windows 10 al ser ligero y rápido lo cual beneficia el correcto funcionamiento del software.</p> 		

Tabla 9: Entorno de calidad

Aplicación web
MySQL workbench 8.0
Soporte del sistema (windows 10)

Tabla 10: Entorno de producción

Aplicación web Hosting y Dominio
MySQL workbench 8.0
Soporte del sistema (windows 10)

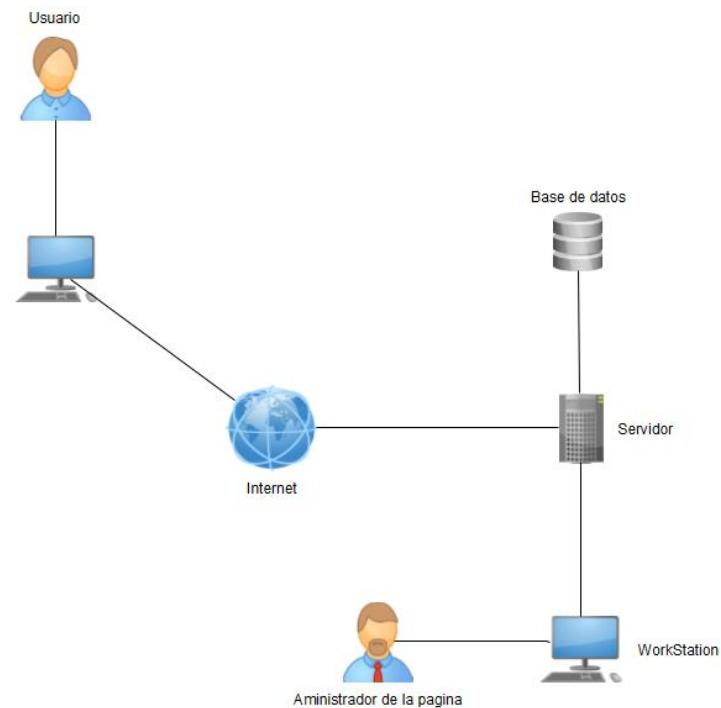


Ilustración 3: Arquitectura de capa de Producción

3.7. Arquitectura de HW/SW

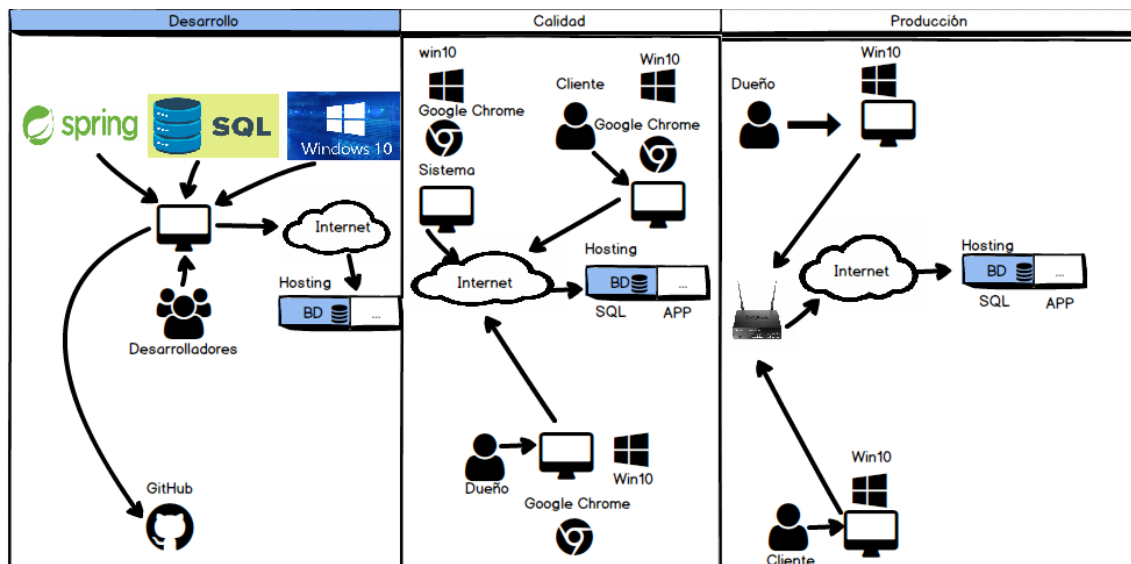
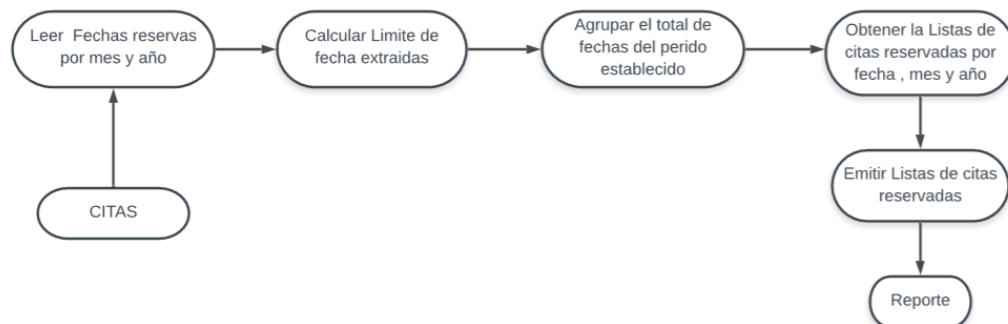
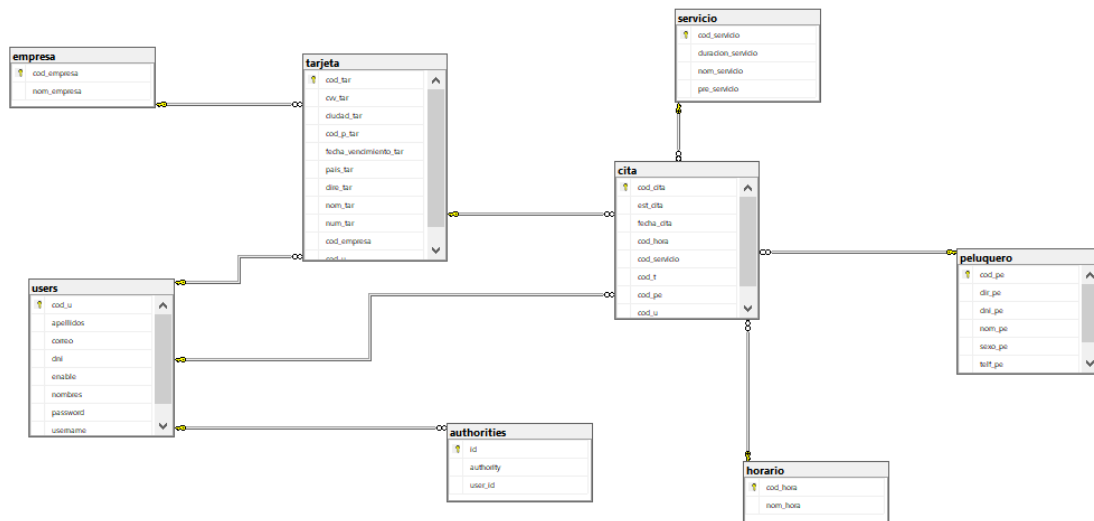


Ilustración 4: Arquitectura de HW/SW



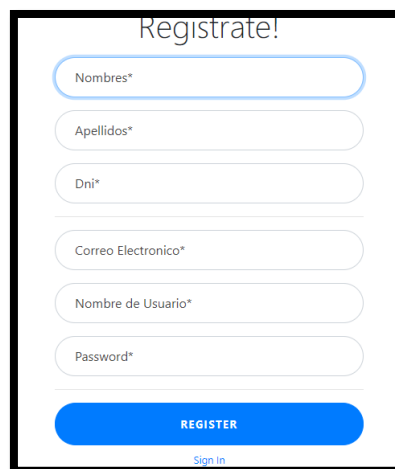
3.10. Esquema de Plan de Pruebas de la Iteración

Tabla 11: Plan de pruebas de la interacción 1

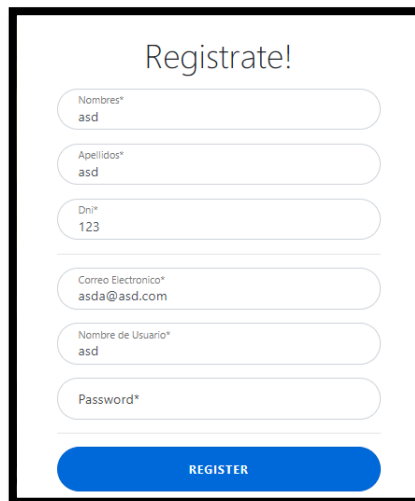
Caso de uso	Datos de entrada normal	Resultado esperado	Dato de entrada anómalo	Resultado esperado
Registrar usuario	Ingrese datos completos del usuario	Usuario Registrado	Ingresa de forma errónea los datos	Se muestra mensaje indicando que falta registrar dato.
Buscar y seleccionar servicio	Busca y selecciona el servicio deseado.	Servicio seleccionado y abre vista de seleccionar horario	No selecciono ningún servicio	No carga vista de selección horario
Buscar y seleccionar horario	Busca y selecciona el horario deseado.	Horario seleccionado y guardado	No selecciono ningún horario (Fecha)	El sistema alerta indicando que falta seleccionar horario.
Buscar y seleccionar peluquero	Busca y selecciona el peluquero deseado.	Selecciona peluquero	No selecciono peluquero	Se le asigna uno por defecto
Registrar Cita	Registrar cita.	Cita registrada	No se pudo registrar la cita	Se muestra mensaje de "La cita no pudo ser registrada"

3.10.1. Ejecución de Pruebas de la Iteración:

- **Registrar usuario:**
 - Datos entrada normal

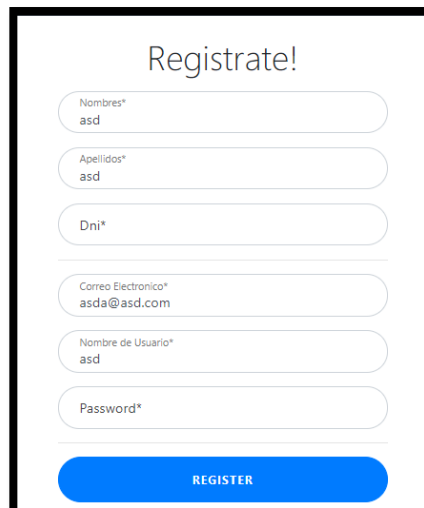


○ Resultado esperado



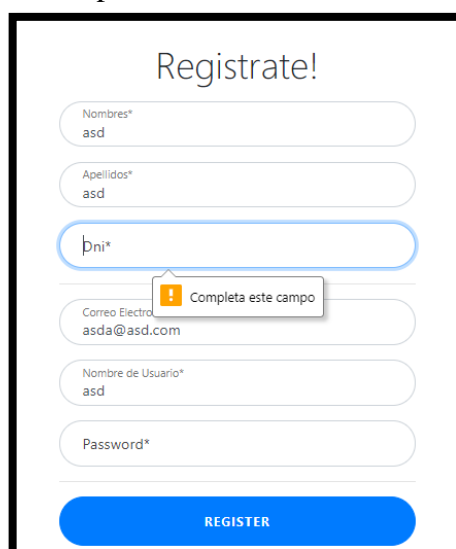
A screenshot of a web form titled "Registrate!". The form contains several input fields, each with a label and an asterisk indicating it is required. The fields are: "Nombres*" with the value "asd", "Apellidos*" with the value "asd", "Dni*" with the value "123", "Correo Electronico*" with the value "asda@asd.com", "Nombre de Usuario*" with the value "asd", and "Password*". Below the fields is a blue button labeled "REGISTER". The form is enclosed in a black border.

○ Datos entrada anómalo



A screenshot of the same "Registrate!" form. The input fields are: "Nombres*" with "asd", "Apellidos*" with "asd", "Dni*" with an empty field, "Correo Electronico*" with "asda@asd.com", "Nombre de Usuario*" with "asd", and "Password*" with an empty field. The blue "REGISTER" button is at the bottom. The form is enclosed in a black border.

○ Resultado esperado



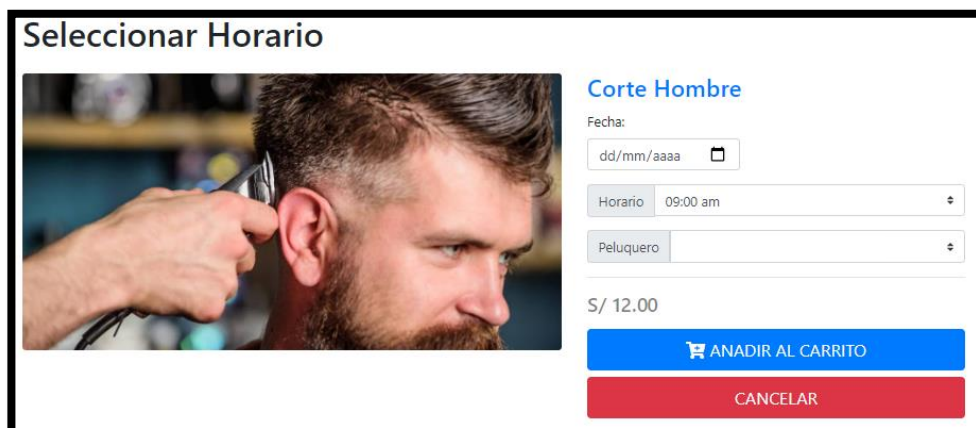
A screenshot of the "Registrate!" form. The input fields are: "Nombres*" with "asd", "Apellidos*" with "asd", "Dni*" with "dni", "Correo Electronico*" with "asda@asd.com", "Nombre de Usuario*" with "asd", and "Password*" with an empty field. A blue "REGISTER" button is at the bottom. A validation error message is displayed above the "Dni*" field, consisting of an orange exclamation mark icon and the text "Completa este campo". The form is enclosed in a black border.

- **Buscar y seleccionar servicio:**

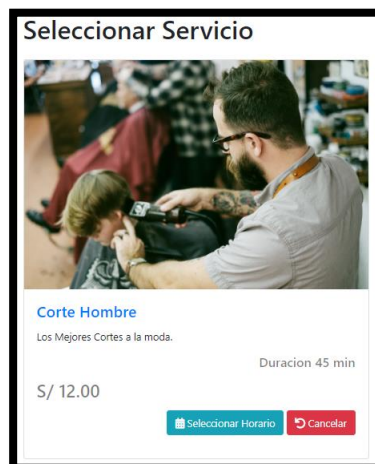
- Datos entrada normal



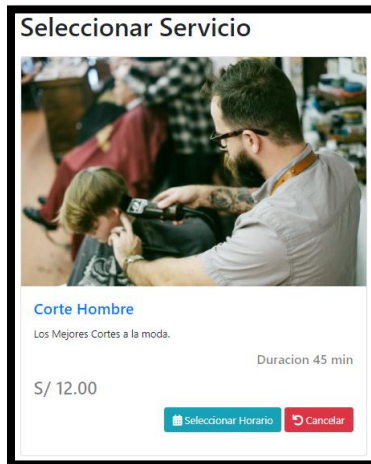
- Resultado esperado



- Datos entrada anómalo




- Resultado esperado



- **Buscar y seleccionar horario:**

- Datos entrada normal

Seleccionar Horario



Corte Hombre

Fecha:

Horario:

Peluquero:


S/ 12.00

ANADIR AL CARRITO

CANCELAR

- Resultado esperado

Seleccionar Horario



Corte Hombre

Fecha:

Horario:

Peluquero:

S/ 12.00

ANADIR AL CARRITO


CANCELAR

Producto añadido

Aceptar

- Datos entrada anómalo

Seleccionar Horario



Corte Hombre

Fecha:

Horario:

Peluquero:


S/ 12.00

[ANADIR AL CARRITO](#)

[CANCELAR](#)

- Resultado esperado

Seleccionar Horario



Corte Hombre

Fecha:

Peluquero:

S/ 12.00


[ANADIR AL CARRITO](#)

[CANCELAR](#)

- **Buscar y seleccionar peluquero:**

- Datos entrada normal

Seleccionar Horario



Corte Hombre

Fecha:

Horario:

Peluquero:

S/ 12.00

[ANADIR AL CARRITO](#)

[CANCELAR](#)

- Resultado esperado

- Datos entrada anómalo

- Resultado esperado

- **Registrar cita:**

- Datos entrada normal



- Resultado esperado



- Datos entrada anómalo



- Resultado esperado



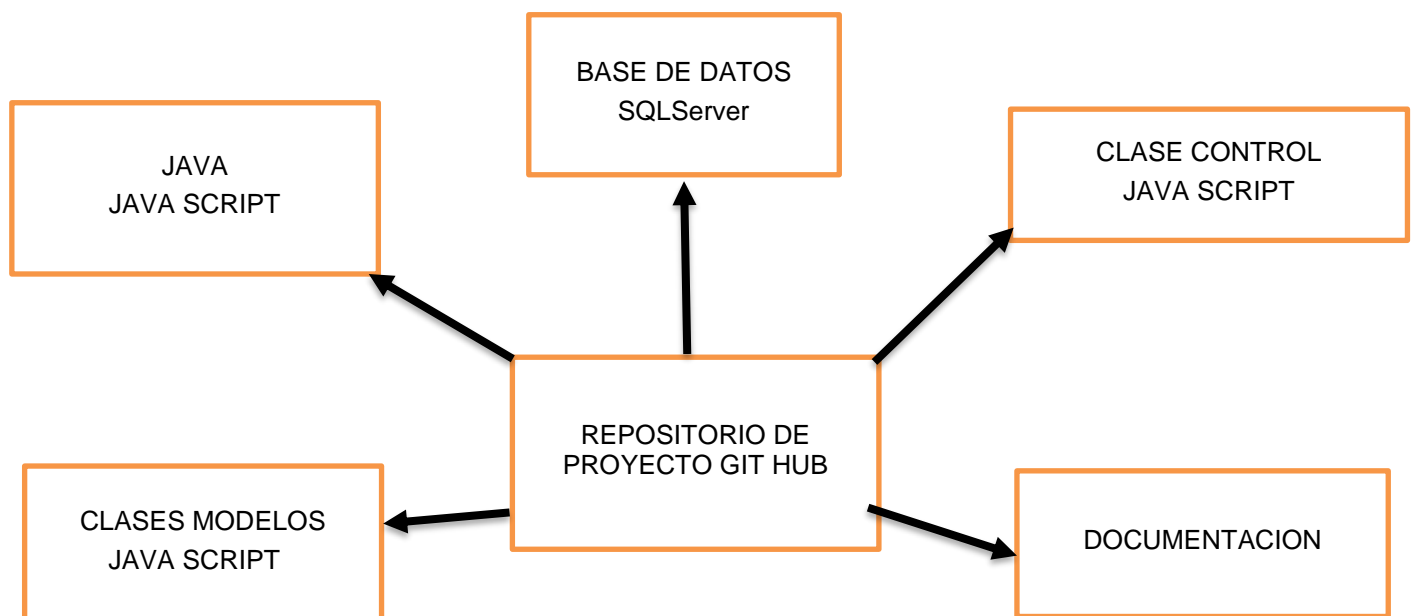
3.11. Esquema de Plan de Pruebas de la Iteración 2

Caso de uso	Datos de entrada normal	Resultado esperado	Dato de entrada anómalo	Resultado esperado
Seleccionar Peluquero de Preferencia	Que el usuario pueda solicitar al peluquero de preferencia	Que elija el peluquero que prefiera	No se pudo seleccionar peluquero	No se seleccionó el peluquero

1.1. Esquema de arquitectura de Reposito del Proyecto



Ilustración 7: Repositorio de proyecto



3.12. Cuenta Github y Documentación

VictorBars / peluqueria-corte-y-color

Unwatch 1 Unstar 1 Fork 0

<> Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights Settings

No description, website, or topics provided. Edit

Manage topics

2 commits 1 branch 0 packages 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

VictorBars Documentation		Latest commit e757eaa 19 hours ago
Proyecto-Peluqueria-master	Proyecto peluqueria	20 hours ago
Avanze proyecto_t1.pdf	Documentacion	19 hours ago
Avanze proyecto_t2.pdf	Documentacion	19 hours ago

Ilustración 8: Github y documentación

3.13. DIAGRAMA DE SECUENCIAS

3.13.1. Caso de uso: Registrar usuario

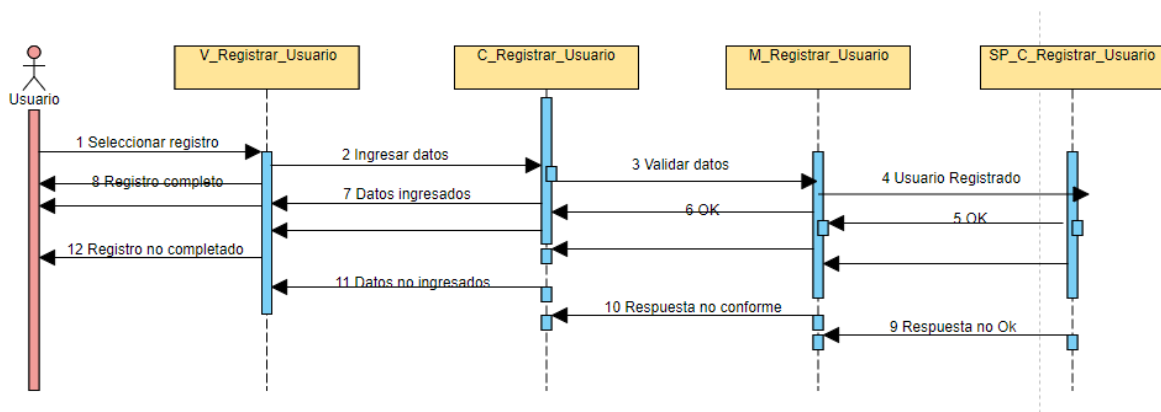


Ilustración 9:Diagrama de Secuencia1

3.13.2. Caso de uso: Buscar y Seleccionar servicio

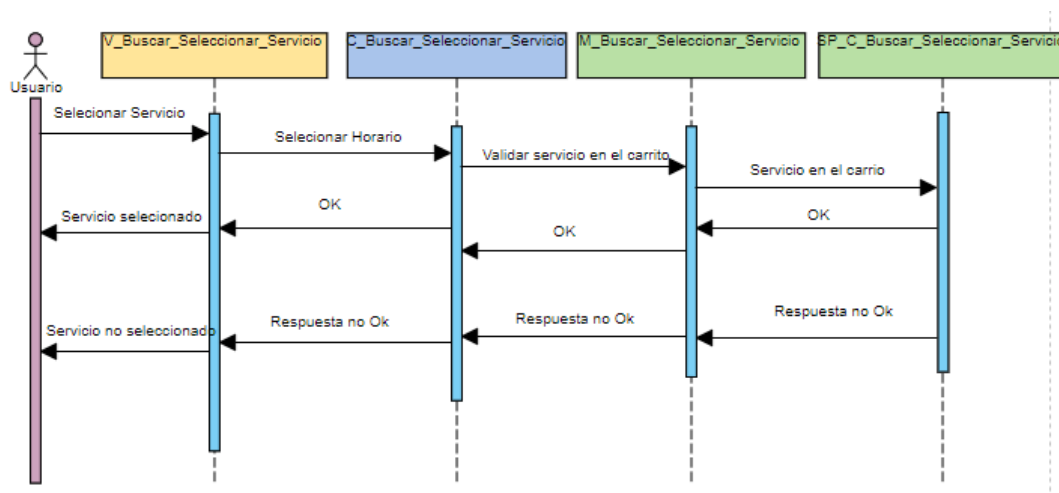


Ilustración 10:Diagrama de Secuencia

3.13.3. Caso de uso: Buscar y Seleccionar horario

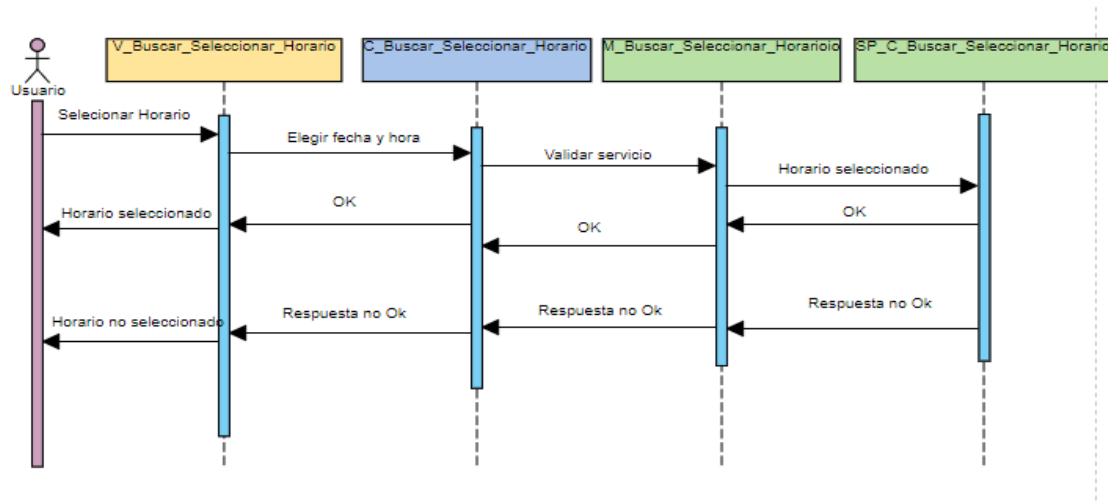


Ilustración 11:Diagrama de Secuencia3

3.13.4. Caso de uso: Buscar y Seleccionar Peluquero

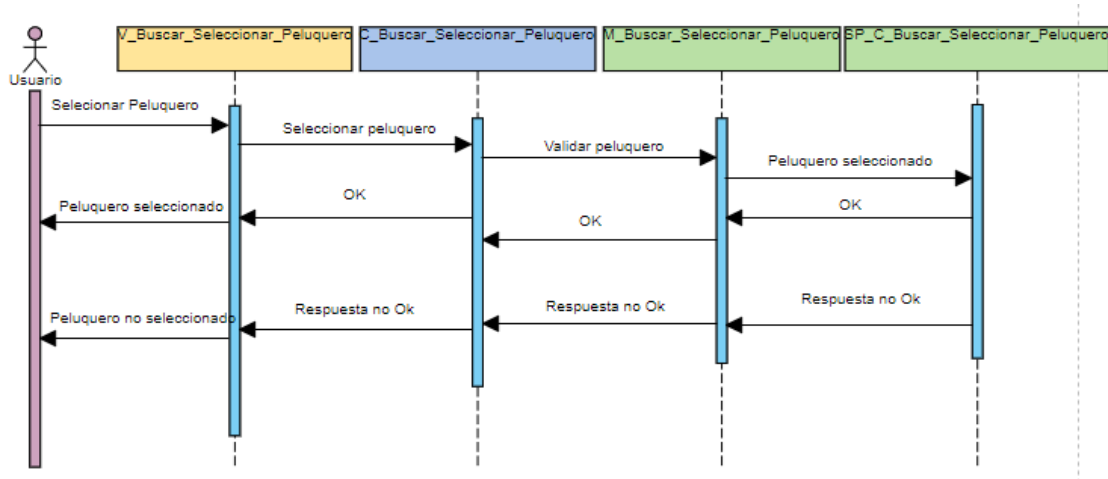


Ilustración 12:Diagrama de Secuencia4

3.13.5. Caso de uso: Registrar Cita

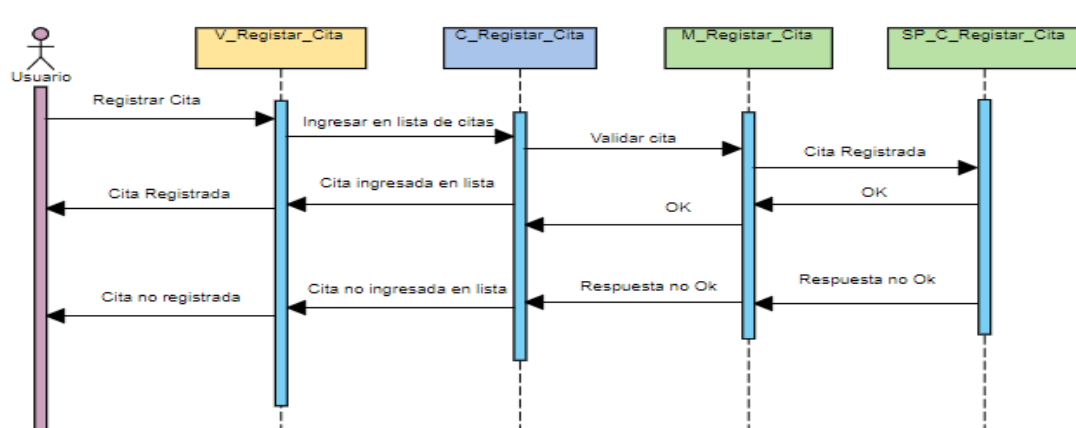
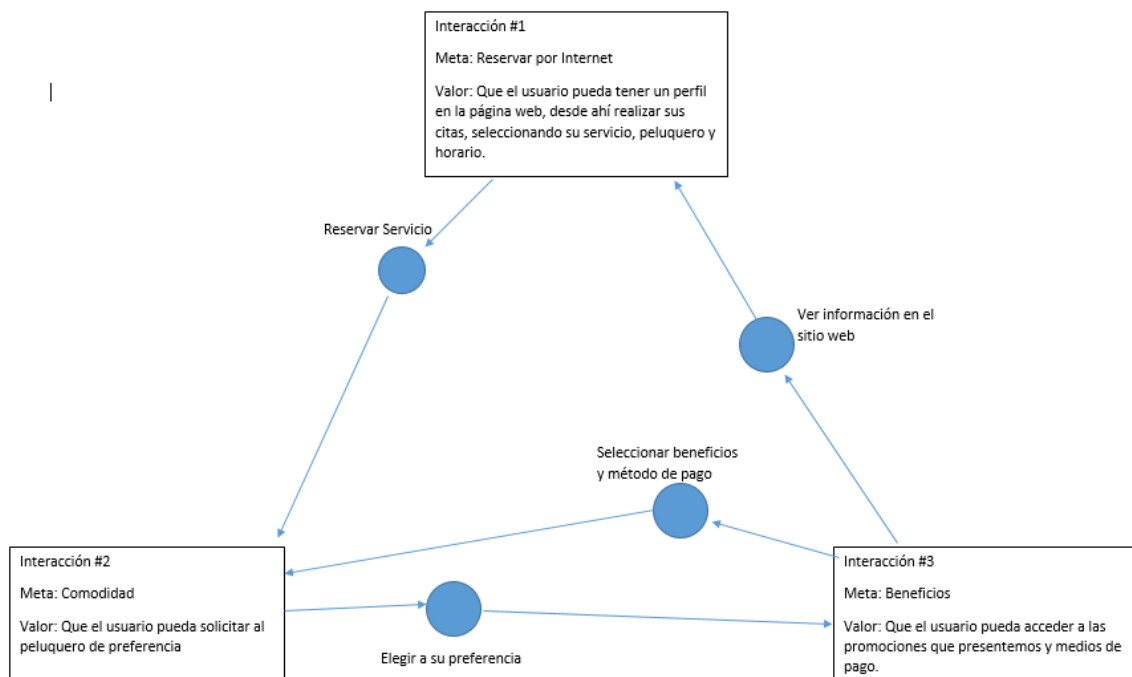


Ilustración 13:Diagrama de Secuencia5

3.14. Diagrama de subsistemas e interfaces



3.15. Patrón CRC

Modelo Servicio

Tabla 12: Patrón CRC - Modelo Servicio

IDENTIFICADOR	DESCRIPCION
1. Nombre de Clase	Servicio
2. Descripción de la Clase	Seleccionar Servicio
3. Tipo de Clase MVC	Modelo
4. Atributos de la Clase	
Lista de atributos	Código, Nombre, Precio, ListaCitas
5. Responsabilidades	
Servicios que ofrece la Clase	Ingresar Servicios
	Seleccionar Servicio
	Envía el evento a Seleccionar Horario
	Envía los datos servicios a Registrar Cita
	Envía el evento a cancelar
6. Colaboraciones	
Nombre de Clase Servidor	Método getListacita
	Método setListacita

Controlador Servicio

Tabla 13: Patrón CRC - Controlador Servicio

IDENTIFICADOR	DESCRIPCION
1. Nombre de Clase	Servicio
2. Descripción de la Clase	Seleccionar Servicio
3. Tipo de Clase MVC	Controlador
4. Atributos de la Clase	
Lista de atributos	servicioService
	horarioService
	citaService
	peluqueroService
5. Responsabilidades	
Servicios que ofrece la Clase	Envía al evento Mostrar Servicios
	Envía al evento Seleccionar Servicios
	Envía al evento Ingresar Servicios
	Envía al evento Mostrar Horarios
	Envía al evento Seleccionar Horarios
	Envía al evento Ingresar Horarios
	Envía al evento Mostrar Peluqueros
	Envía al evento Seleccionar Peluqueros
	Envía al evento Ingresar Peluqueros
	Envía al evento Detalle Cita
	Envía al evento Registrar Pago Cita
6. Colaboraciones	
Nombre de Clase Servidor	Método Inicio
	Método info
	Método Cita
	Método save
	Método Pago Cita

Vista Servicio

Tabla 14: Patrón CRC - Vista Servicio

IDENTIFICADOR	DESCRIPCION
1. Nombre de Clase	Servicio
2. Descripción de la Clase	Seleccionar Servicio
3. Tipo de Clase MVC	Vista
4. Atributos de la Clase	
Lista de Atributos	-----
5. Responsabilidades	
Servicios que ofrece la Clase	-----
6. Colaboraciones	
Nombre de Clase Servidor	Dependencia de la Clase Controlador Servicio

3.16. Patrón SOA

3.16.1. Gmail API

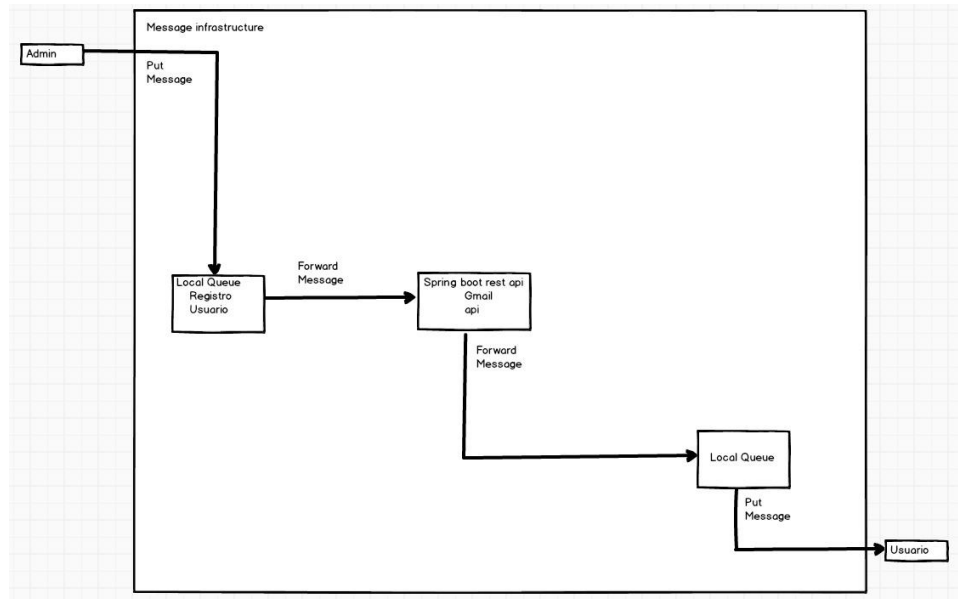


Ilustración 14: Patrón SOA (Gmail-API)

3.16.2. Excel API

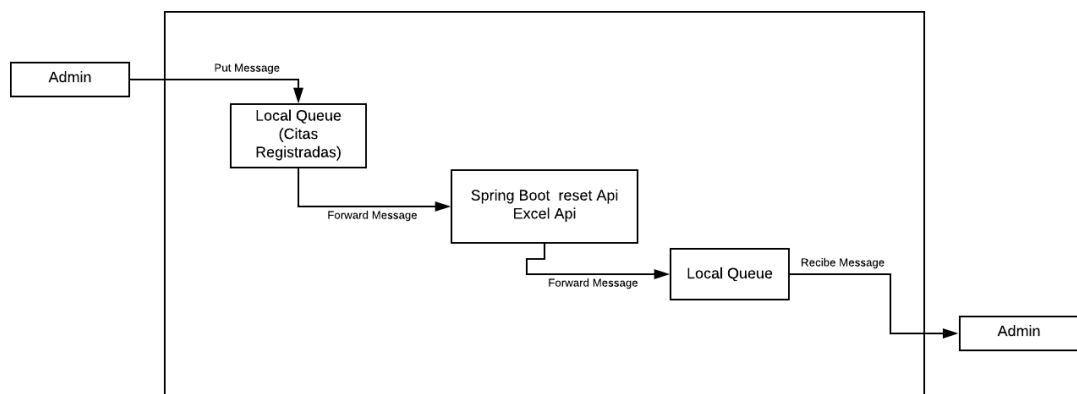


Ilustración 15: Patrón SOA (Excel-API)

CAPÍTULO 4. CONCLUSIONES

- Hoy en día es indispensable la accesibilidad a un sistema para cualquier tipo de trabajo, para aumentar productividad y reducir documentación.
- La interacción sistema – usuario genera mayor interés de las personas hacia la tecnología.
- Las constantes visitas de personas a la empresa, depende tanto de los beneficios que ofrezcan como la innovación que tengan para presentar dichos servicios.
- El sistema ayuda a la interacción con los usuarios.

CAPÍTULO 5. RECOMENDACIONES

- Contratar a un diseñador web para reestructurar la apariencia visual de la página web.
- Mantener siempre capacitados a los trabajadores de la empresa para que tengan un buen entendimiento y manejo de las distintas características que contiene el sitio web.
- Otorgar tutoriales tanto para los trabajadores como usuarios para que puedan emplear correctamente las características de la página.

CAPÍTULO 6. Referencias:

- Booch, G., Rumbaugh, J., & Jacobson, I. (2006). EL LENGUAJE UNIFICADO DE MODELADO GUIA DEL USUARIO (2ª ed.). Madrid, España: Pearson Educación.
- Bracho, M. (s.f.). Plan de Pruebas. Recuperado 13 octubre, 2019, de <https://es.slideshare.net/choselin/plan-de-pruebas-15563690>
- EcuRed. (s.f.). Diagrama de despliegue. Recuperado 13 octubre, 2019, de https://www.ecured.cu/Diagrama_de_despliegue
- González, A. (2015). Algunas Herramientas de Apoyo al Análisis y Diseño de Software. Recuperado 3 noviembre, 2019, de http://profesores.elo.utfsm.cl/~agv/elo329/1s09/lectures/UseCase_UML.pdf
- Granollers, T. (2014, 30 julio). Prototipos de Software. Recuperado 13 octubre, 2019, de <http://mpiua.invid.udl.cat/prototipos-software/>
- Pardo, J. (2012). Configuración y usos de un mapa de procesos. Madrid, España: AENOR.
- Rodriguez, A. (s.f.). Estructuración y Especificación de Casos de Uso. Recuperado 19 septiembre, 2019, de <https://sites.google.com/site/alfonsoperezr/investigacion/estructuracin-y-especificacin-de-casos-de-uos>
- Romero, Y., & Diaz, Y. (2012). PATRON MODELO-VISTA-CONTROLADOR. Telemática, 11.
- Sommerville I (2011). Ingeniería de software , Pearson Educación, S.A., Madrid
- Pressman , R.(2001) Ingeniería de software: Un enfoque Práctico, Madrid
- Sommerville, I. (2005). INGENIERIA DE SOFTWARE (7ª ed.). Madrid, España: Pearson Educación.
- Techgspyunis. (2014, 3 diciembre). Pipe and Filter [Publicación en un blog]. Recuperado 16 octubre, 2019, de <http://arquitecturasomos4.blogspot.com/2014/12/pipe-and-filter.html>
- Vega, M. (2010, octubre). Casos de uso - UML. Recuperado 18 septiembre, 2019, de <https://lsi.ugr.es/~mvega/docis/casos%20de%20uso.pdf>