



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

Desarrollo Front-end de un sistema de registro de visitas hospitalarias con enfoque en usabilidad y accesibilidad

Alexander Matias Alzamora Arce

alexander.alzamora@usm.cl

Pamela Gatica
Profesor Guía

David Larrondo
Profesor Correferente

Resumen: El proyecto se enfoca en el desarrollo de una plataforma web para la optimización del registro de visitas hospitalarias para un hospital de alta complejidad de la V región. La problemática inicial nace del actual sistema de registro de visitas, el cual se sigue realizando a mano, lo que ocasiona largas esperas y descontento por parte de los usuarios. Por esta razón se solicitó desarrollar una solución para mejorar el sistema de registro y optimizar el flujo dentro del hospital con las partes involucradas. Para solucionar estos problemas, se diseñó VisitaConnect, un sistema que busca la mejora de la experiencia tanto de los visitantes como del personal administrativo y clínico. Para evaluar la solución, se realizaron pruebas de usabilidad guiadas con usuarios reales y se realizaron presentaciones de los flujos que se identificaron con el equipo de desarrollo y el personal en contacto del hospital. Los resultados que se obtuvieron demostraron que el proyecto es viable mostrando una mejora de experiencia en los tiempos de espera y en la percepción positiva del usuario lo cual demuestra que se puede implementar en un entorno real.

Palabras clave: Usabilidad, Registro de visitas, Mejora de experiencia, Plataforma web, Percepción.



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: Desarrollo Front-end de un sistema de registro de visitas hospitalarias con enfoque en usabilidad y accesibilidad

Nombre del candidato(a): Alexander Matías Eduardo Alzamora Arce

Carrera / Grado: Ingeniería en Informática

Campus: Sede Viña del Mar **Departamento:** Electrotecnia e Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Pamela Alexandra Gatica Caballero, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 05/08/2026

Firma:

Estudiante o Candidato(a):

Fecha: 05-08-2026

Firma:

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

1.Introducción

1.1 Contexto y relevancia del desarrollo de componentes específicos en el proyecto

Actualmente, en el Hospital Gustavo Fricke se atiende a la gran mayoría de la población que hay dentro de la región. Al haber una gran cantidad de flujo de personas en el recinto, se generan problemas de gestión, especialmente con el tema del registro de visitas, esto se debe a que actualmente el proceso de agendamiento se realiza de manera manual lo que ocasiona que se generen largas colas de espera, además de provocar poca claridad respecto a la información al momento de ir a visitar a un paciente.

Esta ineficiencia que se encuentra actualmente en el recinto no solo afecta a los visitantes, sino que a la vez genera una sobrecarga al personal administrativo y afecta en la trazabilidad de la información, ya que deben contestar consultas de manera repetitiva, dar información constantemente y entregar la información cada vez que los formularios son rellenos en las distintas áreas que requieren la información del visitante. Además de contar con una deficiencia en términos de seguridad ya que no se cuenta con un control de identidad al momento de ingresar a la visita lo que genera un proceso poco seguro.

1.2 Definición del problema.

El problema central que enfrenta el Hospital Gustavo Fricke comienza en el proceso de registro de visita, el cual aún se realiza de manera manual. Este proceso al realizarlo de esa forma genera los siguientes subproblemas:

1. **Falta de información:** Este problema se genera debido a que no hay una claridad respecto a los horarios y la normativa que hay al momento de ir a visitar un paciente, lo que afecta directamente a la experiencia del usuario.
2. **Sobrecarga administrativa:** Se genera debido a que no se cuenta con una información clara respecto a las normativas y las ubicaciones dentro del recinto, lo que provoca que el personal administrativo tenga que responder consultas de manera reiterada lo cual limita su capacidad de enfocarse en labores más relevantes.
3. **Seguridad:** En términos de la seguridad se ve afectada ya que no hay una forma de validación de identidad del visitante, lo que permite que cualquier usuario pueda ingresar a ver al paciente, representando un riesgo para este último.
4. **Trazabilidad de la información:** Este problema se debe a que el flujo de la información hacia los distintos involucrados luego de haber sido completado el formulario es lento, esto provoca que el personal administrativo tenga que hacer seguimiento a la persona, si llegó a la sala de la visita, si sigue allí o si ya no se encuentra con el paciente.

1.3 Breve descripción sobre la propuesta de solución

Para solucionar la problemática que se encuentra en el Hospital Gustavo Fricke se creó VisitaConnect el cual está enfocado en optimizar el registro de visitas, que permita lograr una reserva sencilla y organizada,



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

enfoque en la disminución de esperas innecesarias, disminución de la carga administrativa y mejora en la seguridad del paciente.

Para lograr esto se incluirá en este proyecto:

1. Desarrollo del sistema web para el registro de las visitas
2. Accesibilidad desde cualquier dispositivo
3. Integración de notificaciones.
4. Módulo administrativo

1.4 Objetivo General y Específicos

Objetivo General:

Desarrollar una interfaz web responsiva y organizada para el registro de visitas de un recinto hospitalario, que permita disminuir la carga hacia el personal del hospital, además de mejorar la percepción de los usuarios hacia el recinto.

Para los objetivos específicos que se presentarán son:

- Desarrollar una interfaz gráfica organizada e intuitiva que facilite el registro de visitas.
- Implementar un diseño responsivo que permita a los usuarios ingresar desde cualquier dispositivo
- Implementar una interfaz para la validación de datos en el Front-end antes de su envío al Back-end para minimizar errores.
- Implementar criterios de accesibilidad web en la interfaz, asegurando la compatibilidad con lectores de pantalla, el contraste visual adecuado y el diseño responsivo para garantizar el acceso desde distintos perfiles de usuario.
- Optimizar la experiencia del usuario a través de pruebas de usabilidad para lograr una mejora en eficiencia y facilidad de uso de la aplicación.

1.5 Justificación de proyecto

Este proyecto es necesario debido a la falta de modernización dentro del hospital, hoy en día se necesita esta modernización para lograr una cercanía con los usuarios y ofrecerles una experiencia más agradable. La gestión de las visitas que se siga realizando de manera manual provoca que tanto personal del hospital como visitantes les provoque un malestar. Automatizar la gestión de las visitas no solo es una mejora para el personal del recinto y los visitantes, también ayuda en cómo es visto el hospital aumentando la sensación de calidad en el mismo.

Junto con esto al incluir funcionalidades como lo son la validación de identidad mediante códigos QR y los correos automáticos para eliminar la visita en caso de algún inconveniente, ayuda a fortalecer la seguridad del recinto y fortalece la comunicación con los visitantes.

1.6 Metodología

Para el desarrollo Front-end de VisitaConnect se utilizó la metodología ágil Scrum. Esta metodología permitió una retroalimentación continua y esto es esencial en el entorno en el que nos enfocamos ya que se busca mejorar la experiencia de los usuarios involucrados. El desarrollo se planificó en 3 sprints los



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

cuales contaban con objetivos específicos, los cuales buscaban tener una versión funcional realizando pruebas de usabilidad, para obtener retroalimentaciones para los siguientes sprints, esto permite reducir problemas y asegurar un diseño final intuitivo y accesible.

1. En el primer sprint el desarrollo del home se realizó mediante referencia visuales de páginas médicas y sitio original del hospital, de esta forma tener un prototipo inicial para las futuras pruebas, junto con esto se implementó un diseño inicial en donde se enfoca en mostrar los principales componentes para el correcto registro de visitas sin un enfoque atractivo para el usuario, junto con esto se realizaron las primeras pruebas de usabilidad y permitiendo que el encargado de UI pudiera diseñar un prototipo de media fidelidad para luego poder ser implementado en el siguiente sprint ya con un enfoque más para la satisfacción y facilidad de uso para el usuario.
2. En el segundo sprint se implementaron los primeros prototipos de baja fidelidad guiado por diseños para las vistas del personal clínico y del guardia, esto debido a que se habían implementado en este sprint sus modelos, junto con esto se aplicó una actualización de las vistas implementadas en el primer sprint con diseños más dirigidos al usuario debido a la retroalimentación recibida con las pruebas de usabilidad, además de agregar el formato responsivo para las vistas de los distintos usuarios, junto con esto se incluyó las últimas pruebas de usabilidad para lograr una actualización en el diseño para el personal clínico y del guardia.
3. En el tercer sprint se plantea implementar las últimas correcciones con las pruebas de usabilidad del sprint anterior y en posterior entregar el producto con su respectiva documentación logrando así una implementación exitosa del software.

1.7 Organización del informe

El informe cuenta con cuatro capítulos. El primero trata de la presentación del contexto y la relevancia del proyecto, la definición del problema con la solución presentada, los objetivos que hay y la metodología utilizada. Para el segundo capítulo se aborda el marco teórico, presentando los fundamentos del desarrollo Front-end, la comunicación entre capas y las tecnologías que fueron empleadas. Para el tercer capítulo se explica la elección de la arquitectura con la cual se desarrolló el proyecto junto con los beneficios que tiene, además de detallar el proceso de desarrollo y la selección de componentes. Finalmente, el cuarto capítulo presenta las conclusiones del proyecto, los conocimientos adquiridos, la presentación de problemas y los beneficios que se entregaron.



2. Marco Teórico

2.1 Fundamentos del desarrollo del Front-end

El desarrollo Front-end corresponde a la parte interactiva que los usuarios ven de una aplicación o una página web en donde se ingresa información y reciben respuestas visuales, por lo que su propósito es preocuparse de todo lo que los usuarios ven e interactúan desde diseño visual hasta las interacciones con los botones, formularios, carruseles, entre otros. Sus componentes claves que hacen todo esto posible son HTML, CSS y JavaScript (Freeman, 2014). Es esencial en la construcción de software ya que es responsable de la apariencia y la sensación que genera el software debido a que es lo primero que el usuario ve, además de favorecer la experiencia del usuario permitiendo formar una interfaz clara y fluida para que la navegación en el software sea intuitiva y eficiente (Nielsen, 1994).

En los inicios del internet el desarrollo Front-end ha evolucionado en muchos aspectos, partieron desde páginas HTML sencillas hasta los dinámicos softwares que hay hoy en día. El desarrollo Front-end comenzó a mediados de 1990-1995 con páginas HTML estáticas con un CSS limitado, ya que no se esperaba una interactividad con los usuarios, luego de estos años donde no se tenía en cuenta la interactividad comenzó a surgir JavaScript, estaba diseñado como un lenguaje ligero para lograr añadir interactividad y dinamismo en las páginas web, actualmente con el auge de los framework y bibliotecas de Front-end como podrían ser Angular JS, React y Vue.js se ha podido optimizar los flujos de trabajo y crear código fácil de mantener, esto debido a que las aplicaciones web cada vez se vuelven más complejas por lo que estos frameworks daban estructura para la creación de los sitios (Freeman, 2014)

La importancia de la calidad que debe tener el desarrollo Front-end radica directamente sobre la experiencia del usuario, esto debido a que determina cómo las personas pueden comprender, aprender y utilizar una aplicación. La usabilidad depende de cosas como la facilidad de aprendizaje, la prevención de errores, la satisfacción del usuario, estos aspectos se deben considerar durante todo el proceso de diseño (Nielsen, 1994).

Tiene un gran impacto en cuanto a la mejora de la experiencia del usuario, si un sitio es funcional y está diseñado para que sea atractivo para cualquier dispositivo tiende a atraer a más visitantes ya que los motivan a explorar el contenido que hay, si un sitio web no es atractivo tiende a que el usuario lo descarte inmediatamente. Por lo que el diseño de la interfaz es un proceso continuo que necesita de pruebas de usabilidad y mejoras constantes para garantizar que la experiencia del usuario sea óptima (Nielsen, 1994).

Junto con esto el crecimiento al acceso a internet durante los años desde dispositivos móviles promovió la adopción del diseño web responsivo, el cual permite que la interfaz desarrollada se ajuste automáticamente a la distribución de la pantalla a la cual se está mostrando sin perder funcionalidad (Marcotte, 2011).



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

2.2 Accesibilidad Web

La accesibilidad web habla del diseño y desarrollo de aplicaciones web que puedan ser utilizadas por todas las personas, independiente de sus capacidades tanto físicas, sensoriales o del entorno desde el que accedan (World Wide Web Consortium, 2018). El propósito principal de la accesibilidad web es eliminar las barreras que hay en el internet, para poder garantizar la igualdad de acceso a la información y su propia funcionalidad a los usuarios que puedan tener discapacidades temporales, permanentes o hasta asociadas a la vejez (Pickering, 2016).

Para estandarizar los criterios, el W3C definió pautas de accesibilidad al contenido web (WCAG 2.1), estas fueron creadas en base a cuatro principios conocidos como los principios POUR (Horton & Quesenbery, 2014) (World Wide Web Consortium, 2018):

- **Perceptible:** Esta habla que la información y los componentes que se presentan tienen que poder ser percibidos por los sentidos disponibles.
- **Operable:** Hace referencia a que los componentes de la interfaz y la navegación tengan que ser manejables mediante distintos tipos de entrada, además que se asegure que los usuarios dispongan del tiempo suficiente para interactuar con el contenido.
- **Comprensible:** Habla de que la información y los componentes de la interfaz deben ser claros y predecibles. Dice que el texto debe ser legible, los patrones de navegación tengan que ser coherentes y los formularios deben tener mecanismos de prevención y corrección de errores de envío de datos.
- **Robusto:** El contenido que se encuentre tenga que poder ser interpretado de manera confiable por una amplia variedad de usuarios.

De igual manera, el diseño visual debe cumplir con criterios específicos de contraste para garantizar la perceptibilidad. Las pautas WCAG 2.1 establecen una relación de contraste como mínimo de 4.5:1 para el texto normal y de 3:1 para el texto grande respecto de su fondo (World Wide Web Consortium, 2018). Cumplir con esto es importante no solo para los usuarios de baja visión o daltonismo, sino también para situaciones de la cotidianidad en la cual se puedan encontrar con baja visibilidad como lo puede ser bajo la luz del sol de manera directa.

2.3 Arquitectura de software y patrón de diseño

La arquitectura de software corresponde a la organización clave que tiene un sistema, describe componentes, relaciones y principios de diseño. Una arquitectura que encaje a lo que uno propone ayuda a la mantenibilidad y escalabilidad del software (Bass, Clements, & Kazman, 2022).

Para el patrón de diseño se seleccionó MVC, el patrón MVC propone la separación entre la lógica de negocios, la capa de presentación y las solicitudes, esto reduce la unión entre los componentes y facilita el mantenimiento del software (Fowler, 2002). Este patrón es implementado nativamente en ASP.NET Core, es comúnmente utilizado para incluir interfaces de usuario, datos y lógica de control, esta forma permite tener una mejor división del trabajo y mantenimiento (Microsoft).

Se divide en 3 componentes principales (Fowler, 2002) (Mozilla Contributors, 2023):

- **Modelo:** tiene la lógica del negocio, reglas de validación y la gestión de datos del sistema, en esta se incluye las clases que definen al negocio, como lo pueden ser los usuarios o productos, además

de manejar lo que permite leer, eliminar y actualizar los datos. En resumen, es el encargado de manejar el estado de la aplicación.

- Vista: La vista está encargada de mostrar la información al usuario de una manera accesible y atractiva, todo esto para lograr una satisfacción en la experiencia del usuario.
- Controlador: Es el encargado de procesar las solicitudes del usuario, esto lo realiza como intermediario entre la vista y el modelo, esto le permite procesar acciones según sea necesario y actualizar el modelo para que luego la vista muestra información actualizada.

Se escogió debido a que ASP.NET Core está diseñado para implementar este patrón, ya que ofrece herramientas integradas como controladores, vistas Razor y modelos (Microsoft). Además, al poder dividir la aplicación web en estas 3 capas, permite poder trabajar simultáneamente con el resto del equipo sin ocasionar problemas y de esta forma acelerar el proceso de desarrollo, utilizar este patrón nos permite poder crear una interfaz ordenada y dinámica.

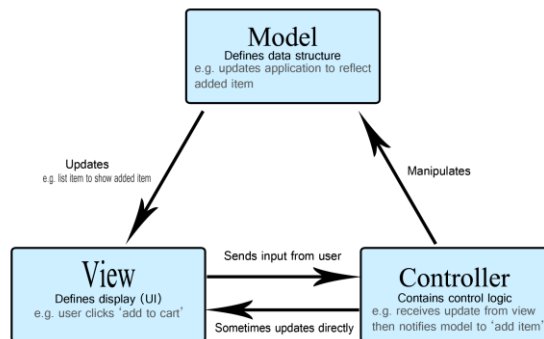


Figura 1: Patrón de diseño MVC
Fuente: <https://developer.mozilla.org/es/docs/Glossary/MVC>

Es importante saber que el patrón MVC y la arquitectura por capas operan en niveles de abstracción de manera distinta dentro del diseño de software. Si bien la arquitectura por capas organiza la estructura general del sistema de manera delimitada, el patrón MVC se basa específicamente en la organización interna de la capa de presentación (Bass, Clements, & Kazman, 2022). Por lo que implementar esta arquitectura permite probar cada capa de manera aislada sin preocuparse de generar errores en otra capa lo que mejora la calidad del software.

2.4 Tecnologías y frameworks utilizados.

Para el desarrollo de VisitaConnect se utilizó ASP.NET MVC el cual cuenta con Razor pages y se utilizó como framework principal, se consideró debido a las capacidades que ofrece Razor para mostrar HTML



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

de manera dinámica, permitiendo de esta manera tener una comunicación correcta entre las capas de MVC (Microsoft)

En cuanto a las tecnologías principales, se utilizaron los principales lenguajes que cubren la necesidad para la maquetación, presentación y experiencia del usuario:

- **HTML:** Se utilizó para darle buena estructura a las distintas vistas para la interfaz de los usuarios, lo que permitió tener una buena organización del contenido.
- **CSS:** Se utilizó para darle diseño a los diferentes elementos que se crearon en los distintos HTML. Creando archivos personalizados para cada una de las vistas evitando errores de diseños.
- **JavaScript:** Se utilizó para optimizar el proceso de creación de una cita en el formulario de reservas, de manera de reducir errores tipográficos, además de ocuparlo para validaciones con la base de datos.
- **Bootstrap:** Bootstrap es un framework de código abierto que va dirigido al diseño de sitios y aplicaciones web, su arquitectura se basa en un sistema de cuadrículado basado en Flexbox, contando con una amplia variedad de componentes pre-hechos (Bootstrap). Se utilizó en algunas vistas para la maquetación de componentes que al realizarlos de manera manual hubieran requerido de mayor esfuerzo y de esta forma se logró un diseño consistente.
- **Flatpickr:** Flatpickr es una biblioteca de JavaScript independiente, que está diseñada para la accesibilidad y selección de fechas y horas (Flatpickr), Flatpickr se incorporó ya que muestra un calendario interactivo y minimalista para la selección de fechas en los formularios, esta herramienta fue utilizada ya que entrega un calendario interactivo y facilita la entrada de los datos. lo que permite reducir errores en la entrada de información.

Junto con esto se incluyó en CSS las **@media queries**, es un módulo de CSS que permite la aplicación de reglas de estilos basándose en las características del dispositivo en el que se renderiza, para cada dispositivo el navegador evalúa el contexto del cliente y si las condiciones establecidas se cumplen aplica las reglas establecidas que se le asignaron (World Wide Web Consortium, 2012). De esta manera se pudo lograr una correcta visualización en distintos tipos de dispositivos. El proyecto utilizó CSS puro en combinación con Bootstrap, y el diseño de cada vista se organizó modularmente para que de esta forma no afectará a las distintas vistas que hay en el proyecto. No se utilizaron frameworks de JavaScript, debido a que toda interacción y validación de los datos se gestionaron mediante Razor y los controladores de ASP.NET Core.

2.5 Comunicación entre capas

Para el desarrollo del Front-end se basó en directrices de diseño y experiencias del usuario previamente definidas por el encargado de UI, los elementos visuales del sitio como lo son la paleta de colores, espaciado, estructura, todos estos fueron implementados mediante HTML, CSS y Bootstrap, de esta forma lograr mantener la coherencia con el diseño propuesto y la interfaz final.



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

El sistema al utilizar el patrón MVC que entrega ASP.NET Core la comunicación entre las capas se logra realizar en la cual el usuario interactúa con la interfaz mediante el formulario del registro de la visita, luego recibe la solicitud el controlador para procesar los datos y luego se dirige hacia el modelo en donde se realizan las validaciones y consultas con la base de datos.

3. Diseño e implementación

3.1 Diseño de Componentes

3.1.1 Arquitectura y estructura de los componentes.

3.1.1.1 Elección del Framework

Para la elección del framework que se utilizó en el desarrollo de la plataforma se basó en los requerimientos y los comentarios que se rescataron en las reuniones que se realizaron en conjunto del hospital, los requerimientos que se mencionaron fueron los siguientes:

1. **Seguridad:** Este requerimiento es importante para preservar los datos de una organización grande como lo es el Hospital Gustavo Frick, pero lo que se nos destacó en este apartado es la sensibilidad de los datos del paciente, esto debido a que al ingresar al paciente este podría tener problemas legales activos o problemas personales, los cuales sí son vulnerables pueden poner en riesgo la integridad del paciente.
 - a. **Mecanismos de seguridad integrados:** Para resguardar y proteger tanto los datos de los usuarios como el sistema, de manera nativa ASP.NET incorpora los siguientes controles nativos en el Front-end.
 - i. **Prevención XSS:** ASP.NET de manera nativa utiliza de manera nativa vistas Razor, por lo que la visualización de datos que envía el usuario al servidor es limpiada mediante codificación HTML de manera automática. Lo que previene los intentos de inyección de scripts maliciosos.
 - ii. **Mitigación de CSRF:** los formularios que se encuentran en el sistema tanto como para el registro de visita como para la actualización de datos del paciente, se le incorpora de manera obligatoria la siguiente línea “@Html.AntiForgeryToken()”, que es la validación de tokens anti forgery que tiene ASP.NET, esto asegura que las solicitudes de reserva provengan del usuario que esté en una sesión activa y no de una petición de manera externa.
2. **Usabilidad:** Este punto destaca ya que las visitas que se contemplan en usar la plataforma va desde los 18 años en adelante, además de que aunque la plataforma está enfocada para que todo el proceso de registro de la visita sea de realizado online, esta de igual manera va a ser utilizada por personal de



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

recepción del hospital, las cuales nos comentaban que se manejaban poco con plataformas nuevas y les costaba aprender, por lo que la plataforma tiene que ser de fácil entendimiento y sencilla de utilizar.

3. **Accesibilidad:** En el inicio del proyecto se comunicó que el enfoque que tenía este apartado era garantizar la compatibilidad en múltiples dispositivos mediante el diseño responsivo, se asumió que con esto se resolvería por completo el acceso para el registro de visita en esta fase que se consideraba como un MVP, sin embargo durante el desarrollo se fueron agregando diversos componentes en enfoque a usuarios con diversidades funcionales o cognitivas, esto al tomar en cuenta nuevamente que la institución a la cual va dirigida esta plataforma cuenta con una variedad amplia de usuarios, no podía dejar simplemente en que el acceso a la plataforma estuviera solamente enfocada en la multiplataforma, ya que a medida que fueron realizando pruebas con usuarios un problema que se repite e identificó era el problema visual, además los usuarios revelaron problemas regulares al momento de ingresar los datos, por lo tanto para mitigar estos problemas se realizó lo siguiente.
 - a. **Navegación por teclado y lector de pantalla:** La que más destaca y a la vez más simple es la de navegación por teclado, se introdujo atributos como aria-label y required en los formularios, para permitir que las personas con discapacidad visual naveguen de mejor manera a través de los lectores de pantalla.
 - b. **Formateo automático de RUT:** Se integró un script que formatea en tiempo real los puntos y el guion verificador mientras el usuario escribe.
 - c. **Contraste y legibilidad:** Las paletas de colores utilizadas se garantizó que en los archivos mantuviera una guía acorde al hospital y que no entorpezca la lectura, incluso para personas con problemas visuales como lo pueden ser las personas con daltonismo, etc.

Esos fueron los puntos que destacaron para la elección del framework. Junto con el equipo de trabajo finalmente se terminó seleccionado ASP.NET, esta elección se realizó ya que ASP.NET por sí misma incorpora de manera nativa varios componentes que van alineados hacia los requerimientos y objetivos que teníamos con la plataforma. Este framework en términos de seguridad incorpora de manera nativa mecanismos de protección contra vulnerabilidades comunes que se mencionaron anteriormente, además está optimizado para rendimiento lo cual es esencial al esperar un flujo alto de usuarios, dando respuestas rápidas lo cual es un punto importante que tomar en cuenta ya que de los comentarios que pudimos recopilar, la mejora a la percepción del usuario es uno de los objetivos claves que quieren cubrir con esta solución.

3.1.2 Recopilación de información de usuarios.

Para la toma de requerimientos y el diseño general de la plataforma, con el equipo lo evaluamos durante varias reuniones con el hospital, ya que en cada reunión se daba a conocer una funcionalidad nueva de la cual estaba ligada el problema principal el cual era el poder que el visitante pudiera agendar una visita desde casa. Luego de muchos análisis se determinó que la plataforma contará con tres módulos distintos, y que la plataforma pudiese ser utilizada desde cualquier dispositivo de esta forma enfocarse en la usabilidad.



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

Por lo que estos tres módulos de los cuales se comentó recién serían para:

- **Visitante:** Quienes tienen que poder realizar la visita desde su casa.
- **Personal clínico:** Encargado del paciente, permite gestionar el estado si es que se encuentra disponible para recibir visitas o no, poder bloquear usuarios para que no puedan visitar al paciente o habilitar la lista blanca que tenga el paciente.
- **Guardia:** encargado de verificar que la persona que está yendo a visitar corresponde o no a la persona que agenda la hora.

3.1.3 Componentes

Para la organización de los distintos componentes nos guiamos mediante los casos de uso que se comentaron con el hospital para los módulos identificados, estos componentes favorecen tanto el flujo de la plataforma, como el flujo interno que hay en el hospital y la seguridad de los datos.

Los casos de uso que destacan en cada uno de los módulos son:

Visitante:

CU1: Agendar visita.

CU2: Cancelar reserva.

CU3: Recibir código QR para validación de datos.

Personal Clínico:

CU1: Gestionar estado del paciente.

CU2: Gestionar permisos para el paciente.

Guardia:

CU1: Validar códigos QR.

CU2: Visualizar datos para ayudar en la guía del recinto.

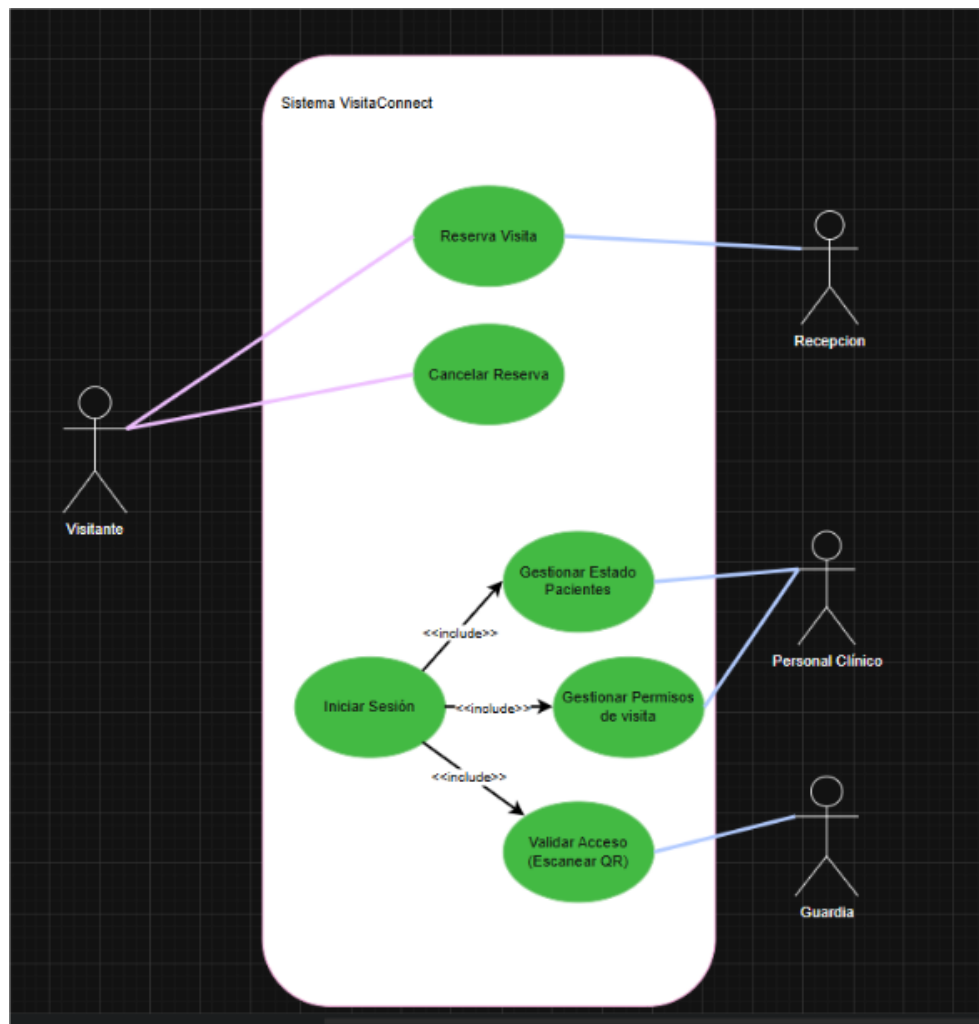


Figura 2: Diagrama de Casos de uso
Fuente: Visita Connect

Estos casos de uso de cada uno de los módulos permitieron apoyar en la selección de componentes de la plataforma, los cuales apoyarán en el problema central del proyecto que es lograr mejorar la percepción del visitante y disminuir la carga administrativa del hospital.

Para el desarrollo del diseño y selección de componentes hubo un enfoque en usabilidad y reducción de peso para el sistema, de esta manera se seleccionaron dos bibliotecas esenciales las cuales fueron Bootstrap y Flatpickr.

Flatpickr: Flatpickr es una biblioteca para la selección de fechas y horas, se seleccionó debido a ser ligera enfocada en mejorar la interacción con el usuario, además se adapta bien con ASP.NET, esta biblioteca además permite implementar reglas para la selección de fechas y horas, lo cual es esencial para uno de los procesos que están dentro de los casos de uso, además como se mencionó que favorece la interacción

con el usuario esto debido a que la interfaz que ofrece es minimalista por lo cual mejora el diseño principal de la vista en la cual está.

Bootstrap: Se seleccionó debido a la amplia cantidad de componentes que ofrece, de esta manera ayuda a reducir el tiempo de desarrollo de cada uno de los componentes, además el sistema en cuadrícula que cuenta Bootstrap permite estructurar el contenido de una manera más flexible, adaptándose a diferentes tipos de pantalla. Estos 3 puntos claves que entrega Bootstrap con sus componentes la convirtió en la opción más clara para apoyar el desarrollo de la plataforma.

Estas dos bibliotecas junto con desarrollo propio de algunos componentes, se logró integrar cada funcionalidad esperada y creada por parte del encargado del Back-end.

3.2 Uso de buenas prácticas y patrones de diseño.

Para el desarrollo de la plataforma se adoptó el paquete de arquitectura MVC (Modelo-Vista-Controlador) que viene nativa en ASP.NET, el uso de las vistas Razor, los archivos creados y el orden de los recursos que se realizaron van directamente alineados para dar un mejor desarrollo del sistema debido a que al contar 3 usuarios esenciales que usan la aplicación fomenta el orden.

Vista: Este es el apartado central del cual se realizó enfoque, en este punto la organización se realizó por secciones funcionales dentro de la carpeta views y la carpeta áreas, de esta manera facilita la trazabilidad y la organización del desarrollo del proyecto, permitió que cada vista creada no se mezclara entre las vistas de los otros usuarios. Favorece la coherencia visual al momento de desarrollo y evitar confusiones, para ejemplificar de mejor manera, se puede mencionar las vistas que dejan en claro cuál es su funcionalidad.

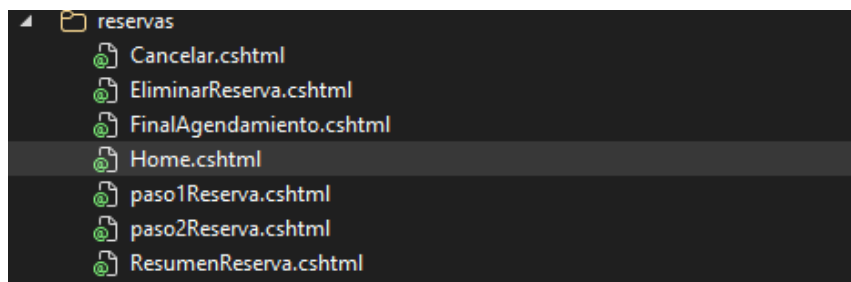


Figura 3: Archivos generados para proceso de toma de reserva
Elaboración propia

Archivo	Funcionalidad
Cancelar.cshtml	Hace referencia a la pantalla que muestra el momento de cancelar una hora que fue agendada, para así dar feedback que fue eliminada la cita con éxito.



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

FinalAgendamiento.cshtml	Es la interfaz que muestra al momento de finalizar el proceso de agendamiento, el cual muestra si la cita fue agendada y el código QR que servirá para ingresar a ver al paciente.
EliminarReserva.cshtml	Hace referencia a la pantalla que muestra el momento de presionar el botón de cancelar la hora que hay al recibir el correo. Mostrando las horas agendadas y cuál desea eliminar.
Home.cshtml	Pantalla inicial de la plataforma donde muestra datos importantes para la reserva de hora.
Paso1Reserva.cshtml	Pantalla que muestra el formulario que tiene que ingresar el usuario para el agendamiento de hora.
Paso2Reserva.cshtml	Pantalla donde el usuario debe seleccionar la fecha y hora que desea visitar al paciente que ingresó en el paso anterior.
ResumenReserva.cshtml	Interfaz que muestra datos importantes que ingresó el usuario para reducir los errores que pudo haber ingresado para el agendamiento de hora.

En el caso para los otros usuarios (clínico y guardia) al ser vistas que no derivan a una extra, solamente se dividieron en la carpeta Áreas, donde cuenta con el login y una carpeta para cada uno, en donde de igual manera se deja en claro cuál es la función de cada archivo creado para en caso de necesitar modificación futura se encuentre rápidamente.

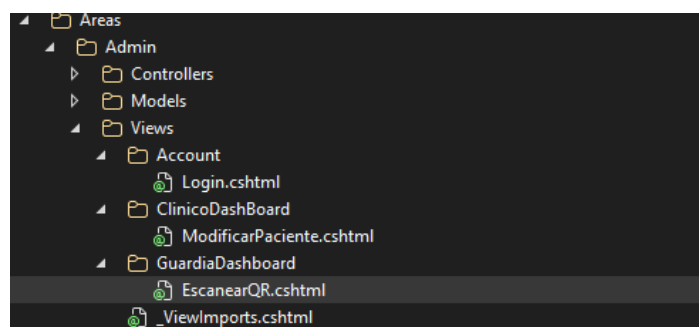


Figura 4: Archivos creados para las áreas de guardia y clínico
Elaboración propia



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

Archivo	Funcionalidad
Account/Login.cshtml	Pantalla donde se encuentra el ingreso para los usuarios permitidos del hospital, sirve para el ingreso tanto como para los usuarios clínicos, como para los guardias
ClinicoDashboard/ModificarPaciente.cshtml	Interfaz donde el personal clínico puede modificar el estado del paciente, tanto como para su estado clínico, como para bloquear y habilitar que el paciente pueda recibir visitas.
GuardiaDashboard/EscanearQR.cshtml	Pantalla donde el guardia pueda escanear los códigos QR de los visitantes, para verificar si pueden ingresar o no a ver a un paciente

Para el diseño de cada uno de los componentes tampoco se queda atrás, esto debido a que, al contar con una variedad amplia de componentes y vistas, se organizó de manera que cada vista tenga un archivo CSS propio, promoviendo de esta forma la facilidad en la implementación del diseño y reducir errores en las demás vistas.

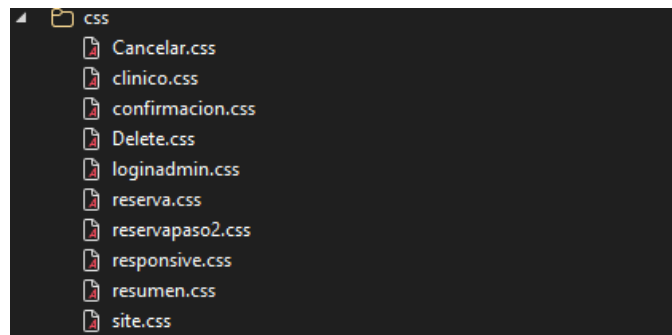


Figura 5: Archivos CSS generados para las vistas
Elaboración propia

Programación modular: Una vez explicado el desarrollo de las vistas empezamos con una de las prácticas de desarrollo que sirvió de guía, esta práctica de desarrollo permitió la facilidad de la reutilización de código, como se puede apreciar en la siguiente imagen, la carpeta centro /wwwroot evidencia claramente esta práctica, ya que los recursos se encuentran organizados de manera independiente, cada uno encapsula una responsabilidad específica, como lo puede ser los estilos CSS, los scripts, los códigos QR generados entre otros.

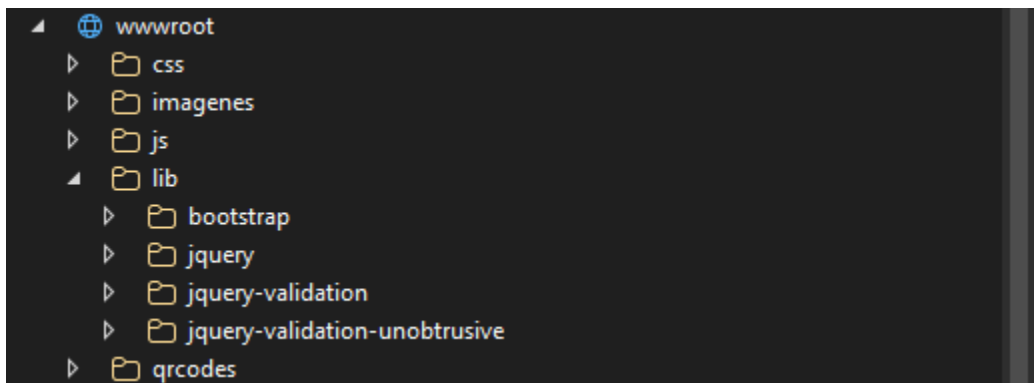


Figura 6: carpetas generadas para reducción de errores
Elaboración propia

Además, se puede apreciar la carpeta `lib` la cual contiene alguna de las bibliotecas externas y cada una separada en su propio módulo, lo cual permite actualizar esas dependencias sin afectar al resto del sistema.

3.3 Detalles de la codificación y desarrollo.

3.3.1 Gestión de Estado y flujo de reserva.

El proceso para la reserva de visitas se diseñó como un formulario en pasos secuenciales para evitar abrumar al usuario con exceso de campos y se realizó de la siguiente manera.

1. Validación en el usuario (Paso 1): El usuario ingresa los datos de identificación del paciente y del visitante, por el lado del cliente a través de scripts en JavaScript se realizan validaciones para la consistencia y los campos obligatorios. Una vez se realizan dichas validaciones los datos se empaquetan en un objeto JSON.
2. Transición (Paso 1 a Paso 2): Para evitar la pérdida de información en caso de cualquier problema que tenga el usuario, se implementó *SessionStorage* en el navegador para la retención de los datos, complementada con *TempData* del Back-end de ASP.NET. Una vez llegamos al paso 2, el Back-end recibe los datos mediante una solicitud HTTP, válida en el servidor y almacena de manera temporal el estado de la reserva en *TempData*.



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

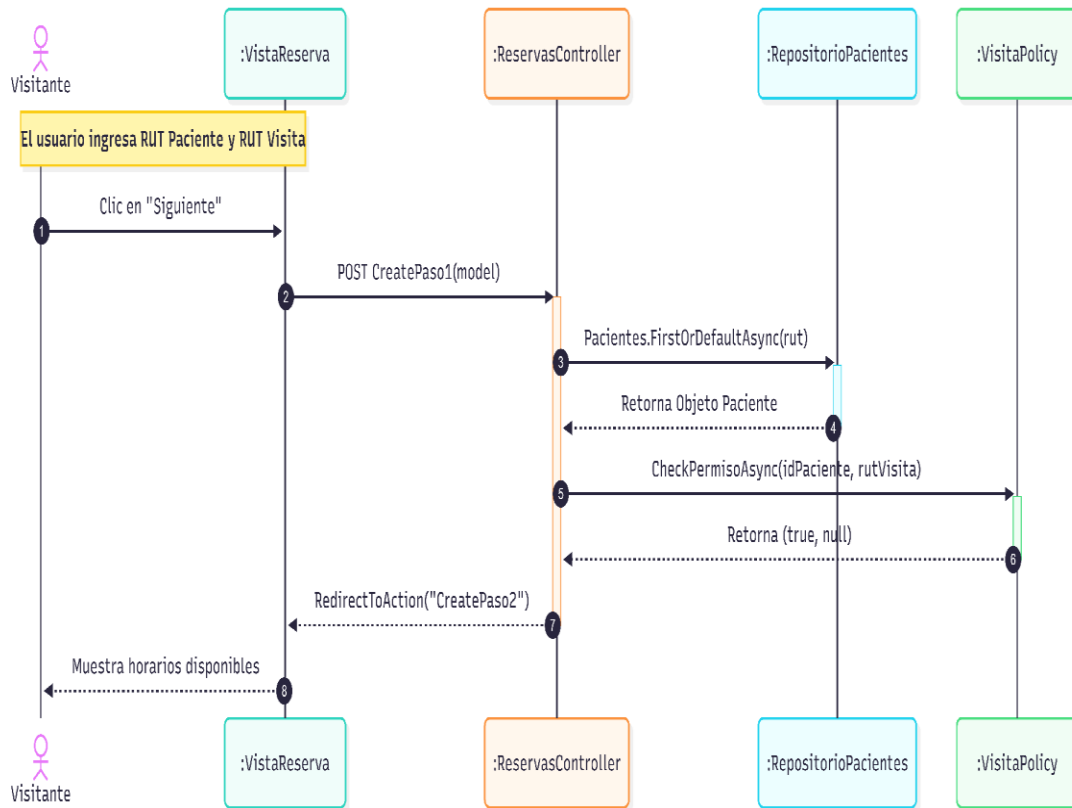


Figura 7: Diagrama de Secuencia validación y Transición

Fuente: Visita Connect

3. Confirmación (Paso 2 a Resumen): En el paso 2 se carga la disponibilidad de horarios del hospital que tiene el paciente. Cuando el usuario selecciona el horario y presiona el botón de 'Confirmar', el controlador del Back-end toma la información del paso1 que estaba almacenada en el *TempData* y la junta con la selección horaria del paso 2 en un solo *ViewModel* y redirige al usuario a la vista de resumen para la validación visual final antes de enviar todo a la base de datos mediante *Entity Framework*, una vez validado y confirmado por parte del usuario, se genera dinámicamente un código QR que es enviado al correo electrónico .



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

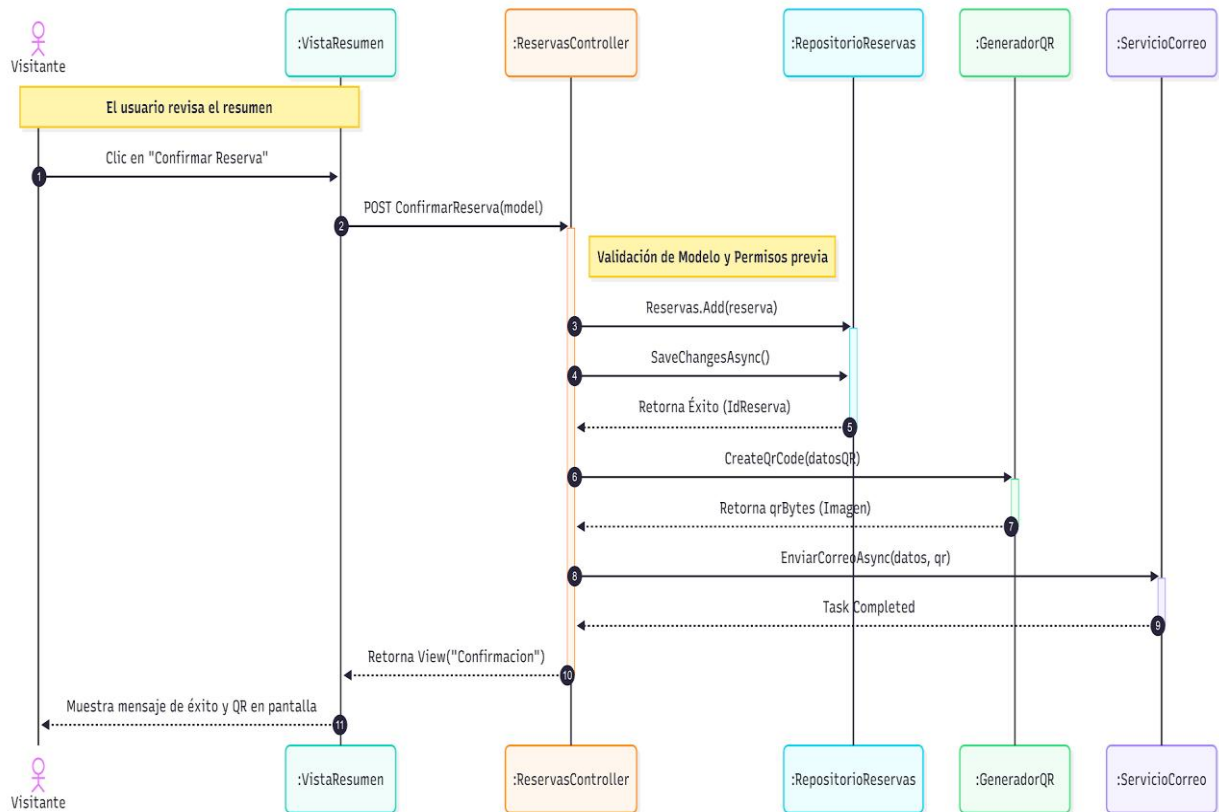


Figura 8: Diagrama de secuencia confirmación
Fuente: Visita Connect

3.3.2 Desarrollo de los sprints

Para el desarrollo del proyecto se utilizó la metodología Scrum, ya que permite guiar de una manera más ordenada el desarrollo del proyecto, el backlog del proyecto se dividió en tres fases, de las cuales las más destacadas son las dos primeras:

1. Sprint 1

- **Desarrollo del home de la página:** Para comenzar todo este proceso al no contar con ninguna referencia debido a no tener aún las primeras pruebas de usabilidad, se desarrolló un home con inicio más enfocado en el diseño que en la usabilidad principal.

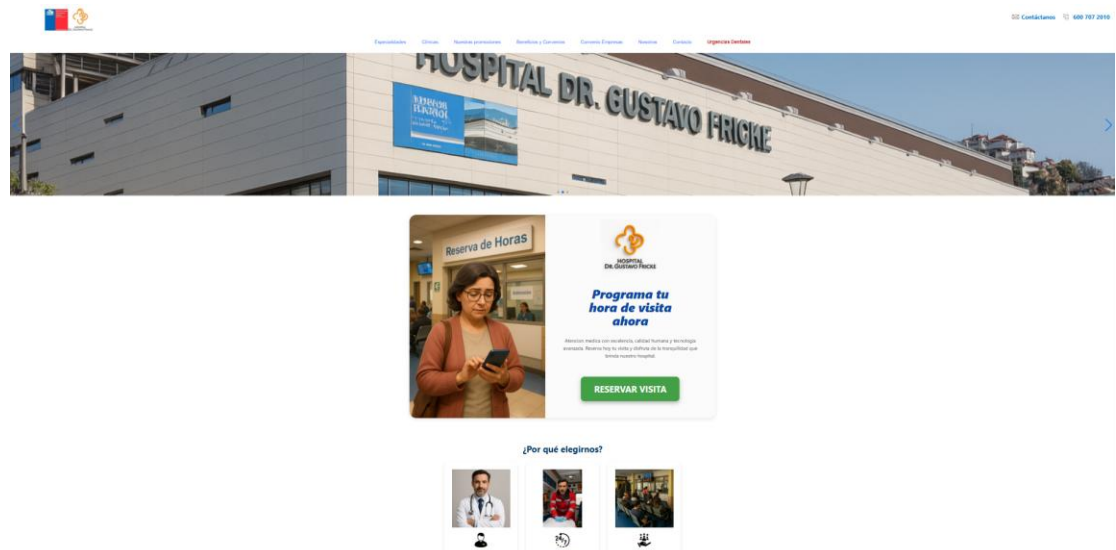


Figura 9: Prototipo home del proyecto
Elaboración propia

- Desarrollo e implementación de diseño proceso de Agendamiento:** Luego de haber creado el home en donde van a poder los visitantes ingresar al flujo principal de la plataforma, se desarrollaron las primeras vistas en base al diseño en baja fidelidad creados por parte del integrante de UI/UX, las demás imágenes que continúan con el flujo de agendamiento están incluidas en el **Anexo 1, Anexo 2 y Anexo 3**.

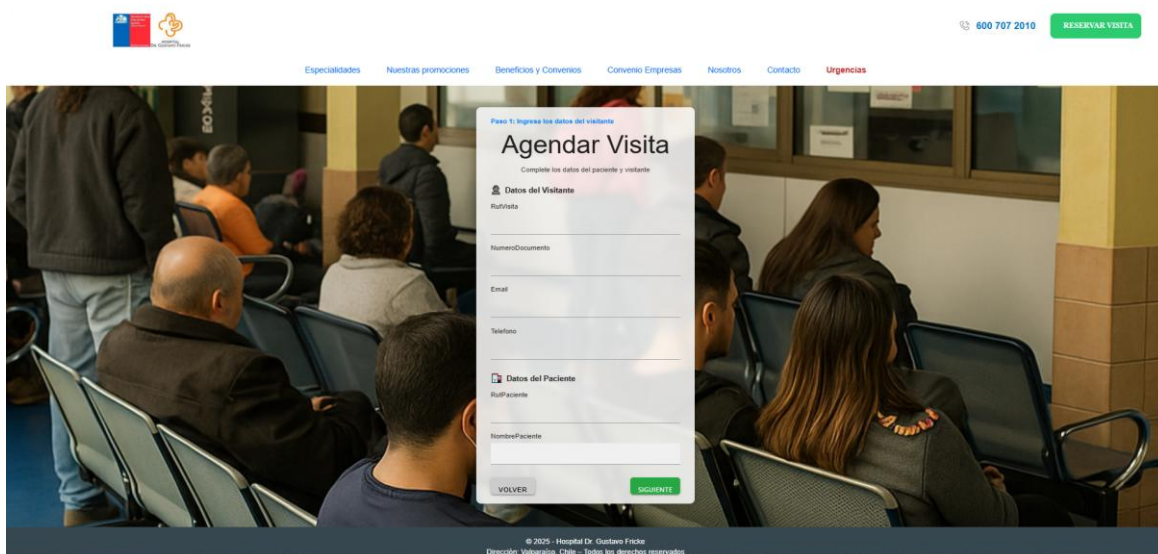


Figura 10: Prototipo inicial Reserva
Elaboración propia

- Comunicación de envío de datos:** Antes de las pruebas principales con usuarios, se verificó el correcto envío de datos ingresados una vez terminado el proceso de reserva, que se guardará correctamente en la base de datos. Este proceso se realizó de manera local viendo a la vez que se



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

terminaba el proceso de agendamiento se verifica los cambios registrados en similitud. Para reducir los errores de ingreso de datos por parte del usuario al momento del ingreso de los RUT, se ocupa un script que ejecuta de manera matemática la consistencia del RUT ingresado por el usuario, esto gracias a que el script limpia, aplica puntos y guion de manera automática. Para el caso de los pacientes este proceso lo realiza JavaScript de igual manera, pero en este caso intercepta los datos y envía una petición HTTP con el método GET al Back-end, mientras este proceso se envía la interfaz muestra un mensaje donde se menciona que se está buscando al paciente, al recibir respuesta por parte del Back-end, se muestra un mensaje donde pueden ocurrir dos casos, el primero es el caso en que encuentra el RUT asociado, entonces actualiza el campo del formulario de NombrePaciente y lo cambia con el nombre al que está asociado el RUT, el segundo caso en donde no encuentra registro por lo cual manda un mensaje de error mencionando que ingrese un RUT válido. De esta manera reduciendo la carga de procesamiento y errores de entrada en una etapa temprana.

- **Pruebas de usabilidad:** Luego de haber desarrollado el prototipo inicial se realizaron las primeras pruebas de usabilidad verificando errores comunes y recopilación de comentarios para mejora del siguiente sprint.

2. Sprint 2

- **Actualización de vistas del agendamiento de visitas:** Para comenzar se realizó el ajuste de las vistas con los nuevos diseños que realizó el encargado de UI/UX, de esta forma se pudo mejorar la calidad de la plataforma y enfocada en el usuario, se mostrará una actualización que recibió un cambio más notorio debido a los comentarios dejados en las pruebas, el cual es la vista para el registro de horario y las demás irán incluidas en el **Anexo 4, Anexo 5 y Anexo 6**.

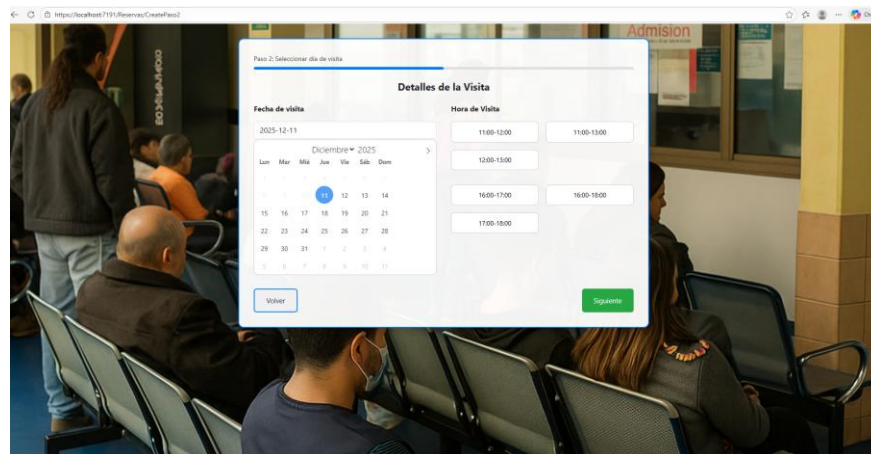


Figura 11: Prototipo actualizado selección de hora
Elaboración propia

- **Desarrollo e implementación del diseño de las vistas para guardia y personal clínico:** Se implementaron los primeros diseños para las cuentas que deben tener el personal del hospital, se generó un diseño básico en donde la vista principal es la de un login en donde ingresan con las cuentas creadas internamente y dependiendo del usuario se le muestra su vista en específica, ahora se muestra la vista

inicial que deriva a estos dos usuarios la cual es el login y las demás se presentan en el **Anexo 7 y Anexo 8**.

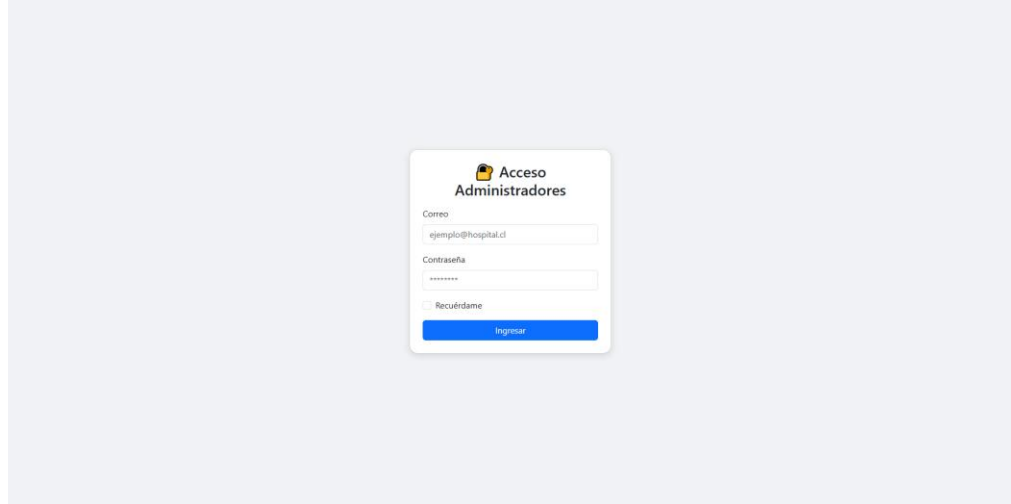


Figura 12: Prototipo inicial login para guardia y personal clínico
Elaboración propia

- **Implementación de formato responsivo a las vistas:** Junto con la integración del nuevo diseño a las vistas para el usuario, se actualizó para que el formato no solamente se pudiese acceder desde computadoras, se incluyó el modo responsivo en todas las vistas, esto se realizó mediante el formato responsivo que otorga las media queries, las cuales se añadieron en el CSS permitiendo adaptar la disposición y el tamaño de los elementos. De esta manera en esta interacción con la plataforma hubo un enfoque dirigido al usuario con las pruebas realizadas anteriormente, se mostrará la vista principal en formato teléfono y el resto irá adjuntado en el **Anexo 9, Anexo 10 y Anexo 11**.



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

Figura 13: Vista móvil actualizada formulario de reserva

Elaboración propia

- **Verificación de cambios:** Se revisó los cambios hechos en base a los comentarios anteriores, siendo el más destacado el manejo de estado de carga para encontrar a alguien en la base de datos, esto se puede visualizar al momento de ingresar el RUT del paciente que se desea visitar, el sistema notifica el estado de búsqueda de manera explícita, en caso de que el paciente no se encuentre en la base de datos el sistema notifica que no se encuentra, y en caso de que si rellena el espacio con el nombre del paciente.
- **Pruebas de usabilidad:** Se realizaron nuevas pruebas de usabilidad, para verificar los nuevos cambios realizados pero esta vez enfocadas en el personal del hospital, de esta manera permitirá ver la usabilidad y permitirá organizar los ajustes para el último sprint.
- **Comunicación de envío de datos:** En este sprint al igual que el primero se realizó una prueba de comunicación con el Back-end, debido a que en el sprint se actualizó la vista de toma de hora, por lo que había que adaptar al nuevo componente que se agregó para mejorar la experiencia visual del usuario. En este caso el Front-end interactúa en tiempo real con el calendario exportado desde Flatpickr, de igual manera la comunicación con el Back-end se ejecuta en tiempo real debido a que al tener que seleccionar hora para un paciente que se encuentra con otros en la misma habitación, a esa persona le puedan estar agendando en un horario que otra quiere ir y por capacidad no se puede. Todo este flujo se realiza de la siguiente manera. Cuando el usuario selecciona un día en el calendario se activa la función onChange de JavaScript que toma el fecha seleccionada y la normaliza, luego se toma la funcional local **cargarRangos**,

la cual solicita al endpoint los rangos de horarios, una vez recibidos los rangos el Front-end muestra de manera estructurada dos secciones, las cuales serían los bloques de mañana y tarde a lo que con JavaScript se toman estos bloques y se crean elementos <button>, para evitar que la habitación de los pacientes no esté fuera de los límites establecidos de los bloques horarios, si el Back-end indica que un bloque específico cuenta con la propiedad ocupado: true, el Front-end inhabilita de manera automática el botón correspondiente, cambiando su color para una claridad visual e impidiendo que pueda ser seleccionado.

3. Sprint 3:

- **Ajustes finales de diseño:** En esta iteración se realizó los ajustes finales para el diseño del personal del hospital, en este caso las vistas que se vieron afectadas fueron la del personal clínico y el login, no se actualizó la vista del guardia ya que dieron a conocer que el diseño les parecía el correcto, se mostrará el cambio de la vista del personal clínico y la actualización del login que se su actualización de diseño fue meramente para dar identidad que es parte del hospital irá incluida en el **Anexo 12**.

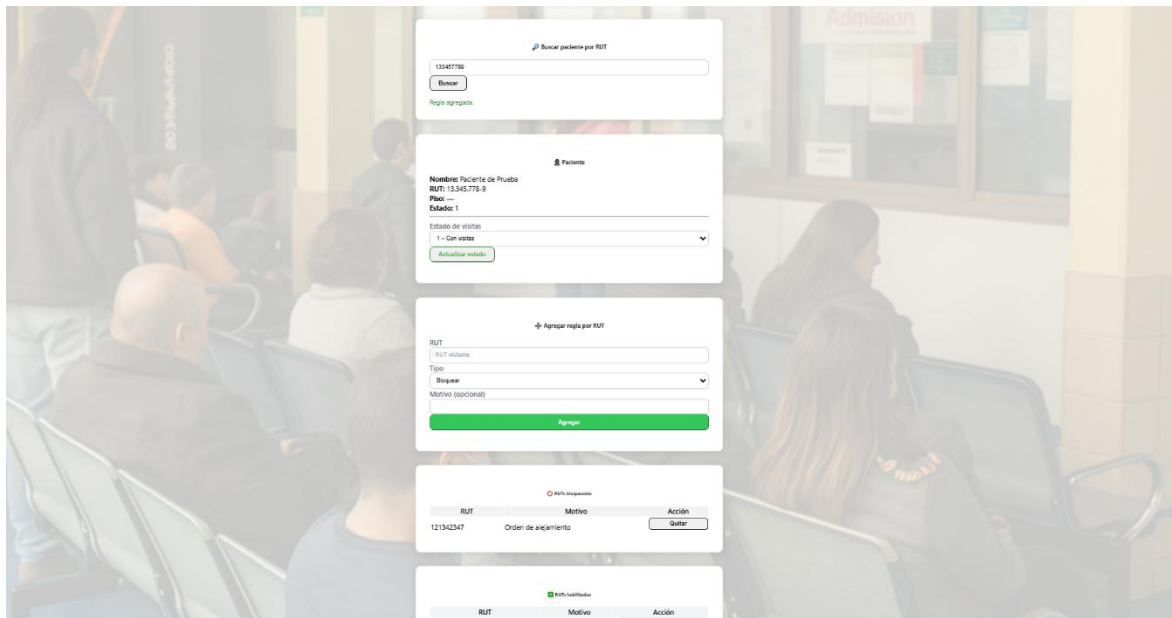


Figura 14: Prototipo actualizado vista clínico
Elaboración propia

- **Desarrollo e implementación de diseño para cancelar visita:** Para finalizar se realizó la vista para la cancelación de hora el diseño implementado se basó en el prototipo generado por parte del encargado de UI/UX. Las vistas creadas y diseñadas cuentan con formato responsivo igualmente, para darle la libertad al usuario, al igual que como puede realizar una reserva desde cualquier lado con el celular puede cancelarla sin tener que estar contactando directamente con el hospital. Las imágenes relacionadas están incluidas en el **Anexo 13 y Anexo 14**.



- **Finalización del proyecto:** Con la implementación de los últimos ajustes tanto como diseño y funcionalidades, damos finalizados el proyecto habiendo desarrollado todos los requerimientos que solicitaba el hospital.

4. Pruebas y validación.

Para las pruebas se realizaron de manera local en conjunto del equipo de trabajo, esto para solucionar errores de funcionamiento, comunicación entre el Back-end y el Front-end de la plataforma, además de realizar pruebas básicas respecto a el flujo principal de la aplicación. Las principales pruebas funcionales de manera local que se realizaron fueron:

- **Botones y acciones:** Se evaluó que cada botón responda correctamente a cada interacción que tuviera, además de enfocarnos en que cumpla con las normas de accesibilidad que se determinaron cuál era el tamaño óptimo para los dispositivos móviles y que se entendiera claramente su funcionalidad.
- **Navegación entre vistas:** Se evaluó que la navegación fuera fluida y comprensiva, que la ruta que se está planteando para el registro de visitas tenga coherencia.
- **Mensajes de error:** Esta prueba fue de las más importantes ya que se logró reducir errores al momento de agendar e identificar qué algunas funcionalidades creadas para la protección de datos del paciente fueron integradas de manera correcta, para dar ejemplos, se pudo reducir el error de ingresar un RUT no válido o escrito de mala manera, además de mostrar mensajes en caso de que algún RUT esté bloqueado y no pueda agendar a cierto paciente.

Junto con las pruebas de manera local se realizaron pruebas con usuarios potenciales del sistema (funcionarios, familiares, personas x), estas pruebas tenían objetivo reunir información respecto a que si la plataforma era comprensible y reunir comentarios que puedan mejorar el diseño del sistema. Para realizar las pruebas externas se utilizó la herramienta ngrok, la cual permitió generar un servidor localmente de manera segura mediante un túnel HTTPS, al momento de generar el servidor se tuvo que compartir la url pública que se genera, una vez teniendo la url con dispositivos propios se ingresa a la url la cual se conecta directamente con la plataforma. Con estas pruebas se pudo observar la responsividad debido a que se realizaron pruebas tanto como en computadores y dispositivos móviles, hubo verificación de diseños con personal del hospital y además en enfoque al visitante se observó fue lo siguiente:

- **Claridad de la interfaz:** Se observó si los usuarios entienden que acción realiza cada pantalla, los datos que deben ingresar son claros, además de que el diseño que se fuera implementando no les genera molestia y se pudieran centrar en el ingreso de datos.
- **Flujo de uso:** En este se observó al igual que el punto anterior la fluidez con la usaban la plataforma si era intuitivo, si daba a entender que debían realizar luego de haber agendado la hora de visita.



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

- **Errores Comunes:** Durante la observación del agendamiento de visita por parte de los usuarios se buscaba apreciar que errores tenían, para ejemplificar hubo usuarios que ingresaban su RUT sin puntos y guión, por lo cual se implementó que se añadiera automáticamente al ingresar los números.

Si bien pruebas que se realizaron de manera local luego nuevamente se realizaron de manera externa, de igual manera entrega una retroalimentación que favorece el desarrollo, en ocasiones cosas que al momento de desarrollar se cree que son claras para todo usuario, finalmente no lo es, especialmente en este caso ya que la plataforma al contar con un amplio rango de edad puede tender a mal flujo.

Para las validaciones que ocurrieron para la comunicación con el Back-end se mostrará el flujo que debe recorrer al momento de la validación del RUT y búsqueda del paciente, y el envío de los datos al finalizar el proceso de reserva.

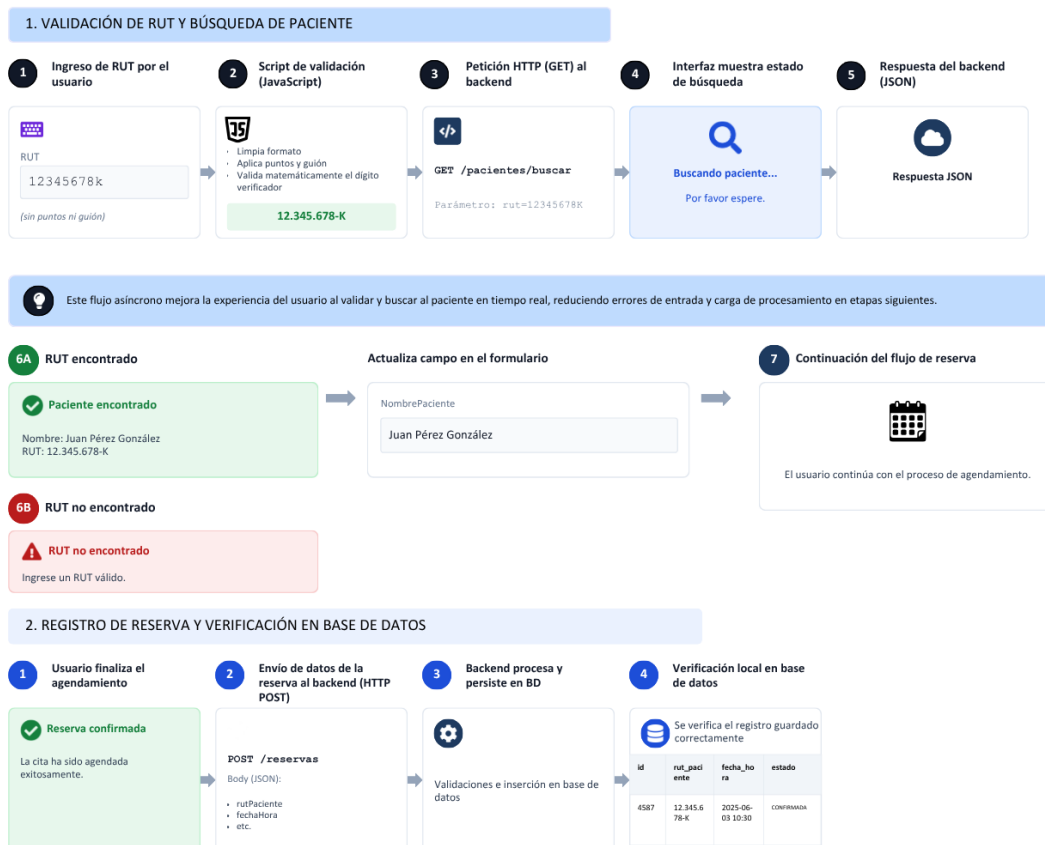


Figura 15: Flujo de Validación de RUT y Registro de Reserva
Elaboración propia



Esta imagen representa la verificación que se hicieron durante todos los sprint, en donde se buscaba mejorar el flujo y la comunicación con el Back-end.

5. Integración con Otros Componentes

La plataforma contempla la integración de dos servicios externos para el funcionamiento operativo del sistema, los cuales son la API para validación de RUT y la API de disponibilidad de camas. Sin embargo, estas APIs no pudieron ser implementadas debido a que al ser un proyecto universitario no contábamos con acceso directo a estas, se solicitaron al hospital, pero no fue posible que nos la entregaran debido a que al ser datos oficiales que contienen datos sensibles, restricciones de seguridad, tampoco nos la pudieron entregar. Para buscar un reemplazo se investigó para encontrar una API pública para la validación de RUT, nos encontramos con boostr.cl la cual es un proveedor de APIs chileno, pero de igual forma no se pudo implementar debido a que no se obtuvo respuesta.

Para mitigar la vulnerabilidad y garantizar el flujo del proyecto y sistema, se implementó una base de datos de manera local en el entorno de desarrollo replicando los esquemas y tipos de datos que se encuentran en el hospital, todo esto ya que, si bien no nos podían entregar datos personales de pacientes, nos podían mencionar que campos eran los que ellos necesitaban para la integración de estos.

6. Conclusión

Con el desarrollo de esta plataforma se logró implementar una solución práctica para abordar la problemática que nos planteó el hospital. Este proyecto representó un desafío, debido a que al no conocer todas las reglas legales que puedan restringir el desarrollo y la necesidad de ver la herramienta que mejor se adapte a la solución que se pidió, fue necesario aprender una nueva tecnología en un corto periodo de tiempo, lo que derivó a mantener una organización y planificación más estricta para cumplir con lo solicitado.

Respecto a los objetivos que fueron planteados:

El primer objetivo era desarrollar una interfaz gráfica organizada e intuitiva que facilite el registro de visitas, ese objetivo se cumplió. Ya que se diseñó e implementó el flujo de agendamiento que fue dividido en pasos secuenciales, lo que pudo reducir la carga visual del usuario y fue validado positivamente en las pruebas de usabilidad.

El segundo objetivo fue implementar un diseño responsivo que permita a los usuarios ingresar desde cualquier dispositivo, este objetivo se cumplió. Ya que gracias a el uso de la media queries en CSS y el sistema de cuadrícula de Bootstrap, las vistas de la plataforma se adaptaron a los distintos tipos de pantalla, siendo verificado en las pruebas externas con ngrok.

El tercer objetivo, implementar una interfaz de validación de datos en el Front-end antes de su envío al Back-end para minimizar errores, ese objetivo se cumplió. Se integraron las validaciones mediante JavaScript para los campos obligatorios, corrección y consistencia de los datos, además de los mensajes



de error que se le muestran al usuario ante ingresos de información erróneos, al igual de crear un paso específico para que el usuario pudiera verificar de manera visual la información ingresada antes de su envío de reserva. Con esto se pudo reducir errores tanto como en el inicio como en el paso previo al envío de datos al servidor.

Para el cuarto objetivo, implementar criterios de accesibilidad web en la interfaz asegurando la compatibilidad con lectores de pantalla, el contraste visual adecuado y el acceso desde distintos perfiles de usuario, se cumplió de manera parcial. Se incorporó atributos como lo es aria-label y required para la compatibilidad con los lectores de pantalla, además se aplicó una paleta de colores con contraste adecuado considerando usuarios con daltonismo. Aun con esto se cumplió de manera parcial ya que no se realizó una evaluación formal contra los criterios WCAG 2.1, por lo que no se puede decir el cumplimiento completo de este punto del proyecto.

Para el quinto objetivo, optimizar la experiencia del usuario a través de pruebas de usabilidad para lograr mejora en eficiencia y facilidad de uso, este objetivo se cumplió. Se realizaron pruebas en cada uno de los sprints para poder ir mejorando la plataforma, estos usuarios incluyen personal clínico y visitantes, de esta manera se pudo identificar errores comunes y aplicar mejoras en el diseño del proyecto. Los resultados de estas pruebas mostraron una mejora en la percepción al sistema y fluidez en el proceso de agendamiento.

Como futuro objetivos quedaron pendientes ciertos aspectos, ya sea por limitaciones de tiempo o por limitaciones directas con los patrocinadores, el que más destaca es la integración de las dos APIs que no fueron entregadas por parte del hospital, otro objetivo que quedo pendiente es la auditoría respecto a los criterios WCAG 2.1, este es relevante de igual manera considerando la variedad de usuarios que utilizaran la plataforma, otro objetivo que quedó pendiente incorporar pruebas automatizadas más robustas y no solamente de manera manual al flujo de desarrollo para poder fortalecer la mantenibilidad del sistema.

Este proyecto no solamente permitió entregar una solución a una problemática crítica que hay en el hospital, sino que también entregó habilidades profesionales claves, como lo son el trabajo en equipo, la adaptación ante nuevos entornos. Además, se pudo apreciar de manera clara el beneficio que entrega esta solución al hospital gracias a la presentación que se realizó en la Feria de Software en donde usuarios los cuales les comentamos el proceso y que soluciona, reconocieron el potencial que tiene la plataforma y el valor que le entrega a la institución.

7. Agradecimientos

Quisiera agradecer a los colaboradores, a pesar de los desafíos que se plantearon en el proyecto, el haber trabajado juntos de esta manera permitió adquirir una nueva experiencia y mejorar el manejo del tiempo. También destacar que este desarrollo se hizo más llevadero gracias a la buena comunicación que había en el equipo, la disposición que había para resolver dudas y ayudar en caso de tener problemas agradezco a cada uno por grata colaboración no solo en este proyecto si no durante todos los años de carrera en los cuales colaboramos. Todo lo desarrollado, los problemas y las cosas buenas que ocurrieron en el transcurso del proyecto se convirtieron en un aporte valioso para mí.



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

8. Referencias

- Bass, L., Clements, P., & Kazman, R. (2022). *Software Architecture in Practice* (4th ed. ed.). Addison-Wesley.
- Bootstrap. (s.f.). Obtenido de Bootstrap: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- Flatpickr. (s.f.). Obtenido de flatpickr – lightweight, powerful datetime picker: [<https://flatpickr.js.org>]
- Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. Addison-Wesley.
- Freeman, E. (2014). *Head First HTML and CSS* (2nd ed. ed.). O'Reilly Media.
- Horton, S., & Quesenbery, W. (2014). *A Web for Everyone: Designing Accessible User Experiences*. Rosenfeld Media.
- Marcotte, E. (2011). *Responsive Web Design*. A List Apart.
- Microsoft. (s.f.). Obtenido de ASP.NET: [<https://dotnet.microsoft.com/en-us/apps/aspnet>]
- Mozilla Contributors. (13 de Noviembre de 2023). Obtenido de MVC – Glosario de MDN Web Docs: [<https://developer.mozilla.org/es/docs/Glossary/MVC>]
- Nielsen, J. (1994). *Usability Engineering*. Morgan Kaufmann.
- Pickering, H. (2016). *Inclusive Design Patterns: Coding Accessibility Into Web Design*. Smashing Magazine.
- World Wide Web Consortium. (2012). Obtenido de Media Queries. W3C Recommendation: [<https://www.w3.org/TR/css3-mediaqueries/>]
- World Wide Web Consortium. (2018). Obtenido de Web Content Accessibility Guidelines (WCAG) 2.1: [<https://www.w3.org/TR/WCAG21/>]

9. Anexos

Anexo 1: Prototipo inicial selección de hora
Elaboración propia



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

 600 707 2010 [RESERVAR VISITA](#)

[Especialidades](#) [Nuestras promociones](#) [Beneficios y Convenios](#) [Convenio Empresas](#) [Nosotros](#) [Contacto](#) [Urgencias](#)

Resumen de la Reserva

Rut del Visitante: 19.123.123-7

Nombre del Paciente: María González

Fecha de la Reserva: 25-12-2025

Hora: 21:33 - 22:33

[CONFIRMAR RESERVA](#)
[VOLVER](#)

© 2025 - Hospital Dr. Gustavo Fricke
Dirección: Valparaíso, Chile – Todos los derechos reservados

Anexo 2: Prototipo inicial resumen de reserva
Elaboración propia

 600 707 2010 [RESERVAR VISITA](#)

[Especialidades](#) [Nuestras promociones](#) [Beneficios y Convenios](#) [Convenio Empresas](#) [Nosotros](#) [Contacto](#) [Urgencias](#)

¡Reserva creada con éxito!

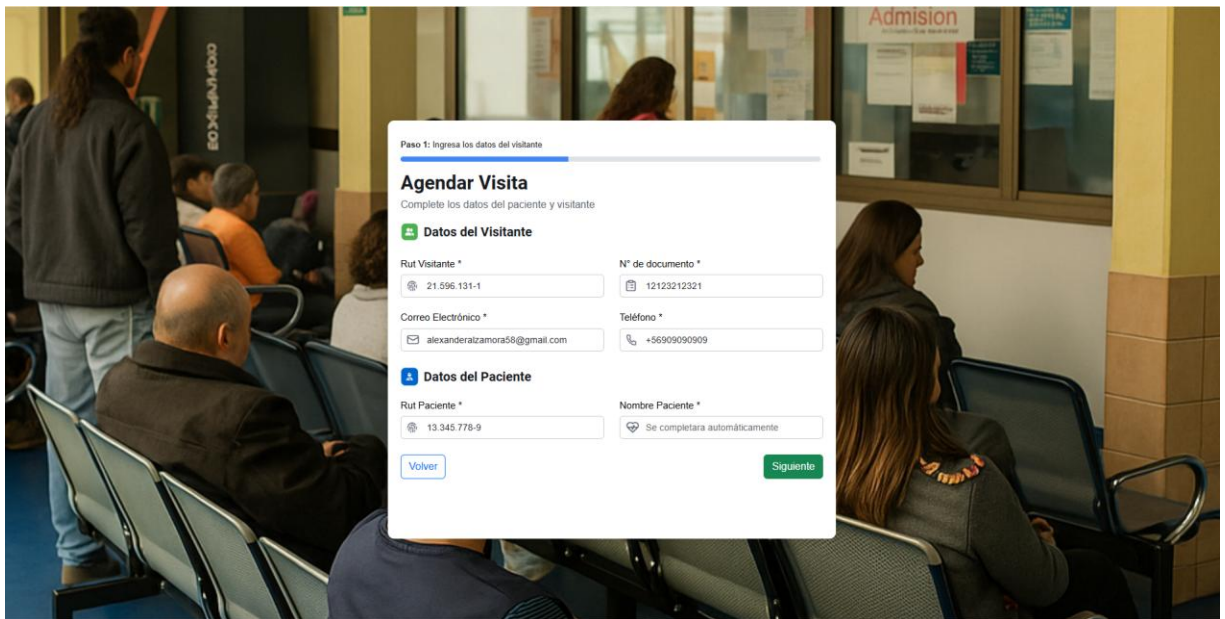
Gracias por registrar tu visita.

[VOLVER AL INICIO](#)

Correo enviado correctamente 

© 2025 - Hospital Dr. Gustavo Fricke
Dirección: Valparaíso, Chile – Todos los derechos reservados

Anexo 3: Prototipo inicial mensaje de confirmación de reserva
Elaboración propia



Paso 1: Ingrese los datos del visitante

Agendar Visita

Complete los datos del paciente y visitante

Datos del Visitante

Rut Visitante *
21.596.131-1

N° de documento *
12123212321

Correo Electrónico *
alexanderzamoraa5@gmail.com

Teléfono *
+56909090909

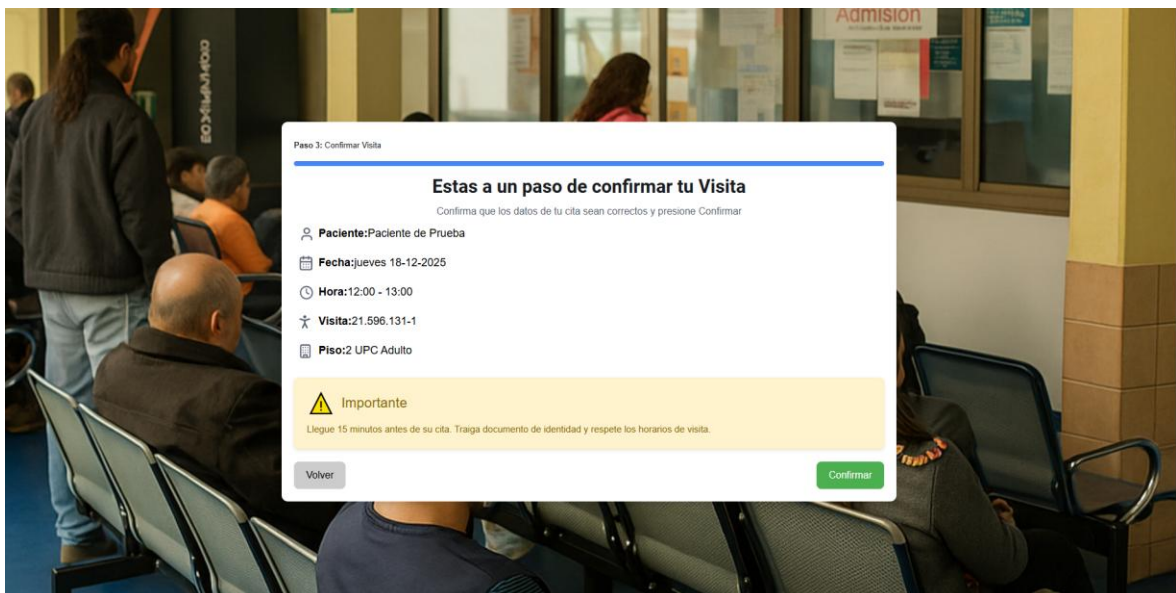
Datos del Paciente

Rut Paciente *
13.345.778-9

Nombre Paciente *
Se completará automáticamente

[Volver](#) [Siguiente](#)

Anexo 4: Prototipo modificado para agendar visita
Elaboración propia



Paso 3: Confirmar Visita

Estas a un paso de confirmar tu Visita

Confirma que los datos de tu cita sean correctos y presione Confirmar

Paciente: Paciente de Prueba

Fecha: jueves 18-12-2025

Hora: 12:00 - 13:00

Visita: 21.596.131-1

Piso: 2 UPC Adulto

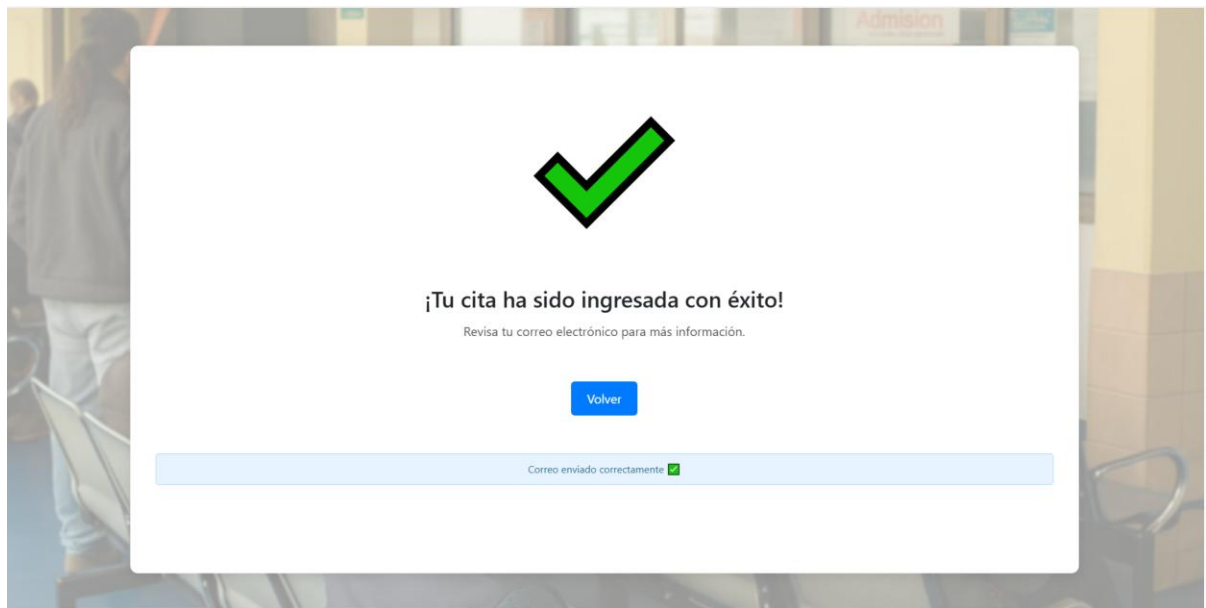
Importante
Llegue 15 minutos antes de su cita. Traiga documento de identidad y respete los horarios de visita.

[Volver](#) [Confirmar](#)

Anexo 5: Segundo prototipo resumen reserva
Elaboración propia



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA



Anexo 6: Segundo prototipo confirmación de reserva
Elaboración propia

Buscar paciente por RUT

Paciente

Nombre: Paciente de Prueba
RUT: 13.345.778-9
Piso: —
Estado: 1

Estado de visitas | 1 - Con visitas ▼

Agregar regla por RUT

RUT
Tipo
Motivo (opcional)

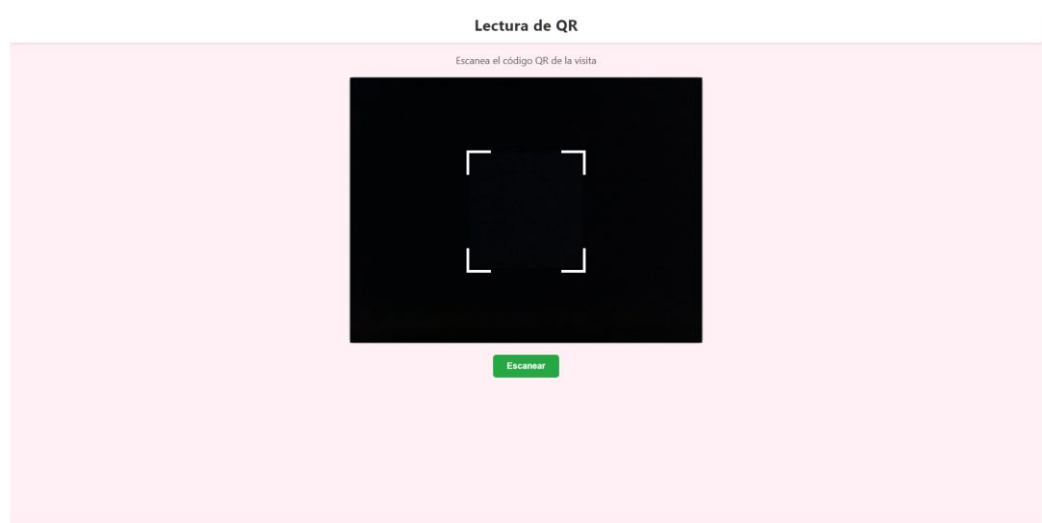
RUTs bloqueados

RUT	Motivo	Acción
21961311	hermoso	Quitar

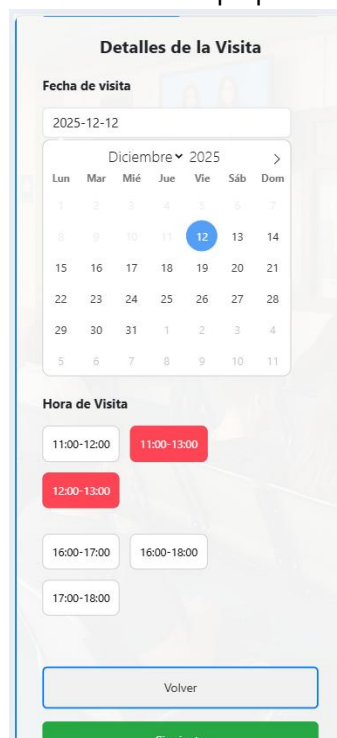
RUTs habilitados

RUT	Motivo	Acción
3214589	precioso	Quitar

Anexo 7: Prototipo inicial vista clínico
Elaboración propia



Anexo 8: Prototipo vista guardia
Elaboración propia



Anexo 9: Versión móvil de plataforma reserva hora
Elaboración propia

Paso 3: Confirmar Visita

Estas a un paso de confirmar tu Visita

Confirma que los datos de tu cita sean correctos y presione Confirmar

 **Paciente:** Paciente de Prueba

 **Fecha:** viernes 12-12-2025

 **Hora:** 16:00 - 17:00

 **Visita:** 21.596.131-1

 **Sala:** Médica Adulto - 4 Oriente

 **Importante**

Llegue 15 minutos antes de su cita. Traiga documento de identidad y respete los horarios de visita.

Volver

Confirmar

Anexo 10: Versión móvil vista resumen reserva
Elaboración propia



¡Tu cita ha sido ingresada con éxito!

Revisa tu correo electrónico para más información.

Volver

Correo enviado correctamente 

Utiliza este QR para poder ingresar a tu visita



Descargar QR

Anexo 11: Versión móvil vista confirmación de agendamiento



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

Elaboración propia

Logo of Universidad Técnica Federico Santa María and Hospital Dr. Gustavo Tuckes

Iniciar Sesión

Correo Electrónico *

correo@ejemplo.com

Contraseña *

[¿Olvidó su contraseña?](#)

Iniciar Sesión

Anexo 12: Versión tablet vista login para personal de hospital
Elaboración propia

Logo of Universidad Técnica Federico Santa María and Hospital Dr. Gustavo Tuckes

¿Seguro que quieres cancelar tu visita?

Reserva

Fecha de la visita:
12-12-2025

Hora de inicio:
16:00

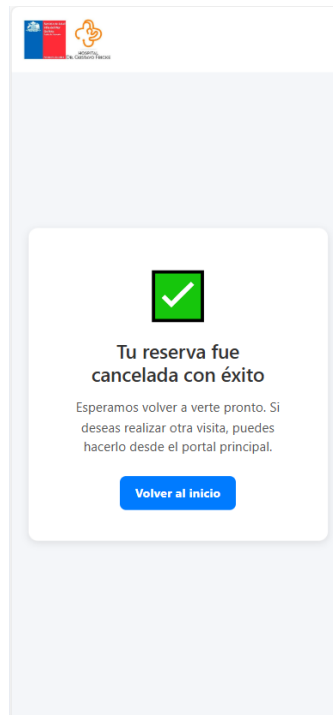
Hora de término:
17:00

Cancelar visita [Volver al inicio](#)

Anexo 13: Prototipo
Elaboración propia



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA



Anexo 14: Prototipo
Elaboración propia