



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

Unidad Académica de Formación Técnica y Tecnológica – PUCE TEC

AGENDAMIENTO D CITAS PARA CONSULTORIOODONTOLOGICOO

**Proyecto de titulación previo a la obtención del título de: Tecnólogo superior en
Desarrollo de Software**

Autor: Bryan Stev Pillajo Lamiña

Tutor: German Orlando Freire Villacis

Quito, Ecuador

2024

Tabla de contenidos

Lista de figuras	3
Capítulo I.....	6
Levantamiento de Requisitos y Diseño del Sistema	6
Capítulo II	9
Construcción del Sistema	9
Capítulo III.....	12
Pruebas y Puesta en Producción	12
5. Pruebas del sistema.....	12
Conclusiones.....	13
Recomendaciones	14
Referencias bibliográficas	14

Lista de figuras

Figura 1: Arquitectura	15
Figura 2:Base de Datos	16
Figura 3:Interfaz	17

DECLARACIÓN y AUTORIZACIÓN

Yo, Bryan Stev Pillajo Lamiña con C.I. 1752866051 autor(a) del trabajo de Integración Curricular intitulado: “Sistema de Agendamiento de Citas para consultorio odontológico”, previa a la obtención del título de Tecnólogo Superior en Desarrollo de Software en la Unidad Académica de Formación Técnica y Tecnológica PUCE TEC:

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE el referido trabajo de titulación, respetando las políticas de propiedad intelectual de Universidad.

Quito, 15/08/2024



Bryan Stev Pillajo Lamiña

C.I. 1752866051

Agradecimientos

Agradezco a mi tutor, German Freire por su orientación ya apoyo durante el desarrollo de este proyecto. A mis familiares por su constante aliento y comprensión la Pontificia Universidad Católica del Ecuador por brindarme la oportunidad de crecer académica y profesionalmente.

Introducción

El presente proyecto de titulación se centra en el desarrollo de un sistema de agendamiento de citas para un consultorio odontológico. Con el objetivo de optimalizar la gestión del tiempo y mejorar la experiencia del paciente, el sistema propuesto permitirá que el personal del consultorio pueda programar y gestionar las citas de manera eficiente. La implementación del sistema busca reducir el tiempo de gestión del consultorio además de la mejora en la organización.

Capítulo I

Levantamiento de Requisitos y Diseño del Sistema

1. Requisitos Funcionales:

1.1 Gestión de Pacientes:

- RF1: El sistema debe permitir el registro de nuevos pacientes, incluyendo información como nombres, apellidos, fecha de nacimiento, teléfono, cédula, número de historia clínica y detalles relevantes de salud.
- RF1: Debe ser posible modificar y eliminar la información de pacientes existentes.
- RF:3 Se debe permitir la búsqueda y visualización rápida de pacientes registrados.

1.2 Agendamiento de Citas:

- RF4: Debe ser posible programar nuevas citas para los pacientes, indicando fecha, hora y tipo de cita (por ejemplo, limpieza, examen, tratamiento).
- RF5: Se deben poder editar y cancelar citas existentes.
- RF6: El sistema debe mostrar una vista de calendario para una fácil visualización de las citas programadas.

1.3 Historial Clínico:

- RF7: Debe existir un formato estructurado para el registro de historiales clínicos de pacientes, incluyendo antecedentes patológicos, estomatológicos, detalles de tratamientos, evolución del tratamiento, bonos realizados y resultados de exámenes.

- RF8: Se debe permitir adjuntar archivos (por ejemplo, imágenes de radiografías) al historial clínico de cada paciente.
- RF9: El sistema debe garantizar la confidencialidad y seguridad de la información del historial clínico.

1.4 Seguimiento de Pagos:

- RF10: Debe ser posible registrar pagos de pacientes, especificando el monto, método de pago y detalles relevantes.
- RF11: El sistema debe permitir el seguimiento del saldo pendiente de cada paciente.

2. Requisitos No Funcionales:

2.1 Seguridad:

- RN1: Se debe establecer un sistema de autenticación seguro para el acceso al sistema.

2.2 Usabilidad:

- FN2: La interfaz de usuario del sistema debe ser intuitiva y fácil de usar, incluso para usuarios con poca experiencia en tecnología.
- FN3: El sistema debe ser compatible con múltiples dispositivos y navegadores web para facilitar su acceso desde diferentes ubicaciones.

2.3 Rendimiento:

- FN4: El sistema debe ser capaz de manejar un gran volumen de datos de pacientes y citas sin experimentar retrasos significativos en el rendimiento.

- FN5: Se deben realizar pruebas de carga para garantizar que el sistema funcione de manera eficiente incluso en momentos de alta demanda.

2.4 Escalabilidad:

- FN6: El sistema debe ser escalable para adaptarse al crecimiento futuro del consultorio y la incorporación de nuevos usuarios y funcionalidades.
- FN7: Se debe permitir la expansión del sistema para integrar nuevas características y servicios según sea necesario.

Esta especificación detallada de requisitos servirá como guía para el diseño, desarrollo e implementación del sistema de agendamiento de citas para el consultorio odontológico Yedent.

Capítulo II

Construcción del Sistema

En el capítulo 2 se describe el proceso del sistema, incluyendo la selección de herramientas y tecnologías, así como la implementación de las funcionalidades definida en el capítulo anterior.

1. Diseño del Sistema

La arquitectura se basa en un método modular y escalable lo que facilita el mantenimiento y la adición de nuevas funcionalidades, los componentes a utilizar:

1.1. Frontend: Para la construcción de la interfaz de usuario utilizando React con JavaScript, para el diseño responsivo Bootstrap y para la estilización css.

1.2. Backend: Implementando Spring Boot que proporciona un entorno robusto y flexible para el desarrollo de aplicaciones java. Este componente maneja la lógica del negocio, las operaciones CRUD y la comunicación con la base de datos.

1.3. Base de Datos: Utiliza PostgreSQL, un sistema de gestión de base de datos relacional, para el almacenamiento de datos del paciente, citas y otros registros relevantes.

2. Desarrollo del Backend

El Desarrollo del backend del sistema comenzó con la configuración del entorno de trabajo usando spring Boot. “Spring Boot proporciona un punto de partida rápido para la creación de aplicación con una configuración mínima”. Se inicio un proyecto base, configurando aspectos esenciales como la conexión con la base de datos, creación de modelos, repositorio, servicios, controladores y las dependencias necesarias para la

funcionalidad del sistema (Spring, 2023).

2.1. Implementación de Funcionalidades

Una vez que el entorno esta configurado, se abordo el desarrollo de las principales funcionalidades.

- **Gestión de Pacientes:** Se implemento la capacidad para registrar, modificar y estado del paciente, Esto incluye almacenar información detallada como nombres, apellidos, y datos médicos necesarios. La interfaz permite la búsqueda rápida, por nombre o por número de cedula, la visualización de los registros de pacientes, facilitando el acceso a la información necesaria para el personal consultorio.
- **Agendamiento de Citas:** El sistema permite programar, editar y cancelar citas. Se incorporo una vista de calendario que muestra todas las citas programadas, facilitando la gestión de horarios y la coordinación de la atención de los pacientes.
- **Historial Clínico:** Se desarrollo un formato estructurado para registrar el historial medico de cada paciente. Esto incluye detalles sobre tratamientos previos, resultados de exámenes, patologías del paciente entre otros datos relevantes. Además, se incorporó la opción de adjuntar archivos, como imágenes o resultados de exámenes al historial clínico para una documentación completa.
- **Seguimiento de Pagos:** La funcionalidad de pagos permite registrar los pagos realizados por los pacientes, incluyendo detalles como el monto y el método de pago. También se incluye una herramienta para seguir el saldo pendiente de cada paciente, ayudando en la gestión financiera el consultorio.

3. Desarrollo del frontend

Para el desarrollo del frontend se utilizó React para crear una interfaz de usuario moderna e interactiva. La herramienta 'create-react-app' facilitó la creación del proyecto, configurando automáticamente un entorno de desarrollo adecuado. Bootstrap se incorporó para asegurar que la aplicación se vea bien, CSS asegura la visualización de colores y contornos de tamaños de pantalla. (Facebook,2022; Bootstrap,2023).

3.1. Implementación de Funcionalidades

El desarrollo del frontend se centró en crear una experiencia de usuario intuitiva:

- **Interfaz de Usuario:** Se diseñaron varios componentes, como formularios para la gestión del paciente y citas, vistas para el historial clínico y pantallas para el seguimiento de pagos. Estos componentes permiten al administrador interactuar fácilmente con el sistema y acceder a la información que necesitan.
- **Integración con Backend:** Se estableció la comunicación entre el frontend y el backend utilizando servicio web. Esto permite que la interfaz de usuario obtenga y datos de manera eficiente, garantizando que la información esté actualizada y sincronizada.
- **Validación y manejo de errores:** se implementaron mecanismos de validación en los formularios para asegurarse de que los datos ingresados sean correctos antes de enviar a la base de datos.

4. Gestión de Base de Datos

El diseño de la base de datos se realizó para soportar todas las funcionalidades del sistema. Durante el desarrollo de la base de datos se crearon prototipos en la plataforma dbdiagram.io donde se fueron modificando en base a las necesidades del usuario. Se crearon tablas para almacenar datos de paciente, citas, historial clínico (Tratamiento, evolución tratamiento, patologías) y pagos. Estas tablas están relacionadas de manera que se mantiene la integridad de

Capítulo III

Pruebas y Puesta en Producción

5. Pruebas del sistema

Las pruebas del sistema se realizaron para garantizar que todas las funcionalidades fueran operativas y efectivas. Debido a que el sistema se encuentra en una fase de desarrollo local, Las pruebas se llevarán a cabo principalmente en un entorno controlado.

5.1. Pruebas unitarias: Estas pruebas se realizaron para verificar funcionalidades principales del sistema como el agendamiento, el registro de pacientes y el seguimiento del paciente.

5.2. Pruebas de integración: Se comprobó la interacción entre el frontend y el backend. Estas pruebas aseguraron que los datos se manejaran correctamente a través de las interfaces del sistema.

5.2.1.1.1. Pruebas de Carga: Las pruebas de carga se adaptaron para el entorno local, enfocándose en el rendimiento bajo condiciones simuladas de uso intenso.

6. Puesta en Producción

Debido que el sistema esta destinado a ser usado en una sola maquina local, la puesta en producción se centra en asegurar que el software estuviera completamente funcional en este entorno.

- 6.1. Despliegue del Software: instalar y configurar el sistema en la maquina local, asegurando que todas las dependencias estuvieran adecuadamente configuradas.
- 6.2. Verificación Post-Despliegue: Se realizarán pruebas en el entorno local para confirmar que el sistema funcione en la máquina de destino.
- 6.3. Capacitación: Se capacitará al usuario para asegurar que el usuario pueda operar el sistema de manera eficiente

Conclusiones

El sistema de agendamiento de citas ha sido desarrollado con éxito en un entorno local. Las pruebas realizadas confirman que todas las funcionalidades operan según lo esperado y que el sistema sea estable en el entorno de destino.

Se a logrado establecer una solución funcional y efectiva para la gestión de citas y datos del paciente.

1. Cumplimiento de Requisitos: El sistema desarrollado ha cumplido con los requisitos funcionales y no funcionales definidos en la fase de levantamiento de requisitos.
2. Desafíos y soluciones: Durante el desarrollo hubo desafíos relacionados con la integración de tecnologías y la adaptación de funcionalidades a las necesidades específicas del usuario.
3. Preparación para el futuro: Aunque el sistema está actualmente diseñado para funcionar en una sola máquina, se ha construido con una arquitectura que permite su escalabilidad futura.

Recomendaciones

1. Monitoreo de Sistema: Aunque el sistema se ejecute localmente, se debe implementar un monitorio básico para detectar cualquier problema que presente en el futuro.
2. Actualizar de manera regular: Asegurarse de mantener el sistema actualizado con las últimas versiones o estables de software, además de dependencias para garantizar la seguridad y estabilidad.
3. Documentación completa: Realizar una documentación actualizada para facilitar el mantenimiento y la resolución de problemas que puedan surgir durante el uso del sistema.

Referencias bibliográficas

Bootstrap. (2023). *Bootstrap Documentation*. <https://getbootstrap.com/docs/>

Facebook. (2022). *React Documentation*. <https://reactjs.org/docs/getting-started.html>

PostgreSQL Global Development Group. (2023). *PostgreSQL Documentation*.

<https://www.postgresql.org/docs/>

Spring. (2023). *Spring Boot Documentation*. <https://docs.spring.io/spring->

[boot/docs/current/reference/htmlsingle/](https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/)

W3Schools. (2023). *CSS Reference*. <https://www.w3schools.com/cssref/>

Anexos

Representación de la Arquitectura del Sistema

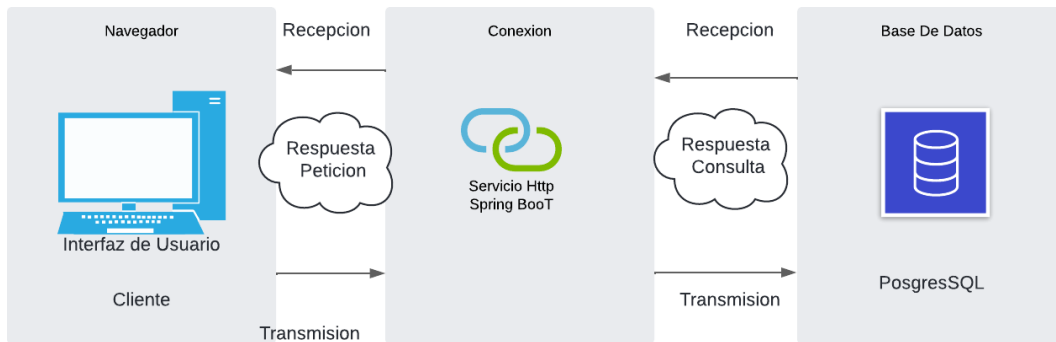


Figura 1: Arquitectura

El diagrama de arquitectura del sistema, creado en lucidchart, presenta la estructura general del sistema de agendamiento de citas para el consultorio odontológico. Este diagrama visualiza las tres principales capas del sistema:

1. Frontend: Representado por una interfaz de usuario construida con React y Bootstrap. Muestra cómo los usuarios interactúan con el sistema a través de formularios y vistas.
2. Backend: Implementado con Spring Boot, que maneja la lógica de negocio y procesa las solicitudes del usuario. Esta capa se encarga de la validación de datos y la coordinación con la base de datos.
3. Base de Datos: Utiliza PostgreSQL para almacenar toda la información relacionada con pacientes, citas y pagos. La base de datos interactúa con el backend para guardar y recuperar datos.

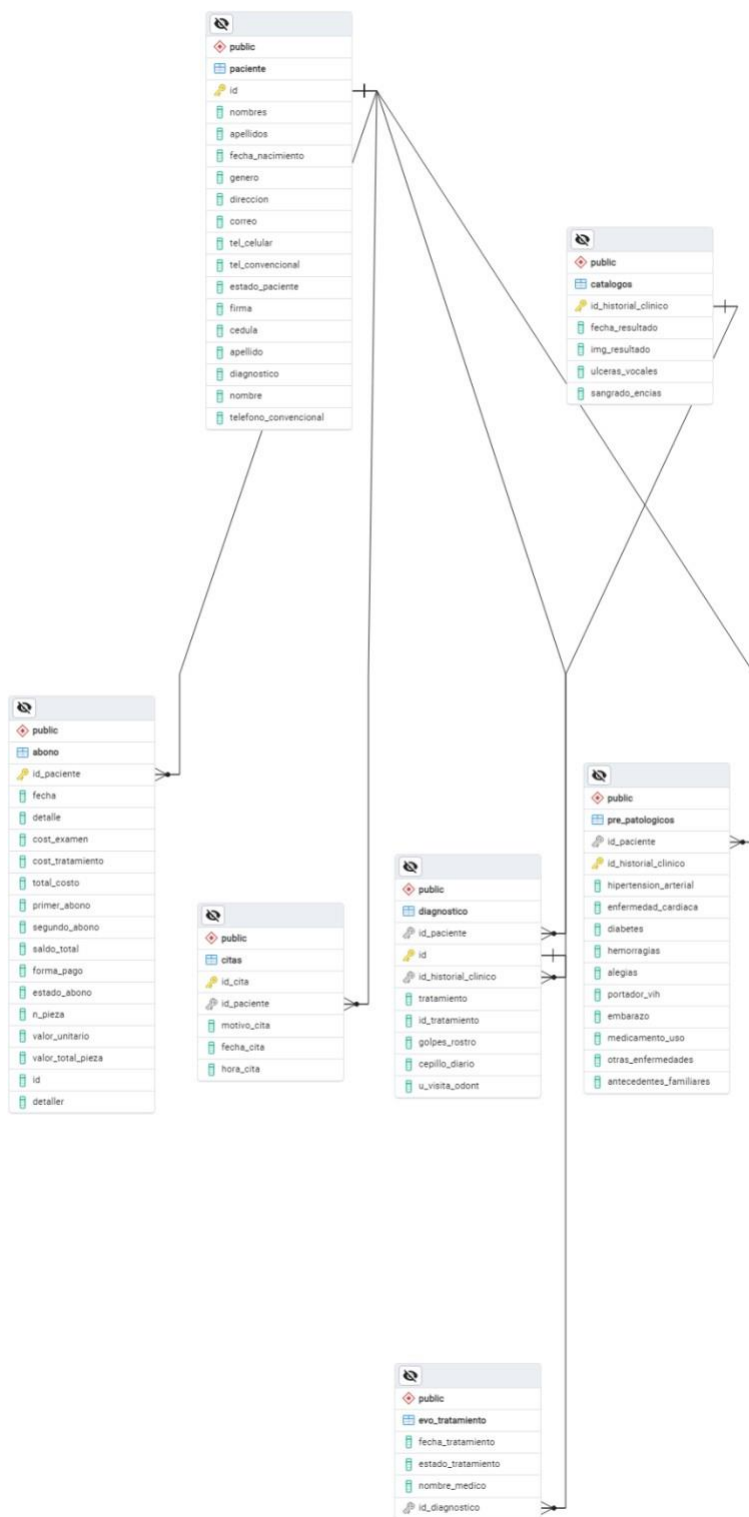


Figura 2: Base de Datos

Se puede desplegar una pequeña representación de las tablas a utilizar para el manejo de la información de los pacientes.

Pacientes: Información personal y de contacto de los pacientes, como nombres, apellidos, fecha de nacimiento y número de historia clínica.

Citas: Datos sobre las citas agendadas, incluyendo fecha, hora, tipo de cita y paciente asociado.

Historial Clínico: Registros detallados de tratamientos, antecedentes médicos y resultados de exámenes.

Pagos: Información sobre los pagos realizados por los pacientes, incluyendo montos, métodos de pago y saldos pendientes.



Figura 3: Interfaz

El diseño del frontend del sistema de agendamiento de citas se centra en proporcionar una interfaz intuitiva y fácil de usar.