

Universidad Autónoma de Zacatecas
Unidad Académica de Ingeniería Eléctrica

Ingeniería de Software

Análisis y Diseño Orientado a Objetos

PROYECTO FINAL – Documento de inicio de proyecto

4° D

Alumnos:

Juana Guadalupe Rodríguez Olivo

Matricula: 39208394

Jessica Natalia Arredondo Rebollo

Matricula: 21204017

Profesor: Cristian Boyain

Fecha: 11 de noviembre 2024

Documento de inicio del proyecto

Propósito del Proyecto:

El objetivo de este proyecto es desarrollar un sistema de control de citas digital para consultorios de ciencias de la salud, con el fin de optimizar el proceso de gestión de citas, mejorar la eficiencia en la organización de la agenda, y minimizar los errores derivados del uso de sistemas manuales. Este sistema permitirá a los consultorios médicos, odontológicos, veterinarios y ópticos registrar, modificar, consultar y cancelar citas de manera sencilla y centralizada.

Alcance:

El sistema tendrá funcionalidades básicas como:

- Registro de citas para pacientes con detalles como nombre, motivo de la consulta y contacto.
- Modificación o cancelación de citas previamente registradas.
- Consulta de citas programadas para un día específico.
- Acceso a los detalles de las citas.

En fases posteriores, se agregarán funciones como recordatorios automáticos para los pacientes y análisis de la actividad de citas.

Lista de Principios y Patrones Clave

Principios de Diseño Orientado a Objetos:

Abstracción: Modelar las entidades importantes como Cita, Paciente y Personal Administrativo.

Encapsulamiento: Proteger los datos sensibles, como los de los pacientes, y proporcionar acceso controlado a través de métodos.

Modularidad: Dividir el sistema en módulos independientes (por ejemplo, registro de citas, consulta de agenda, y visualización de detalles).

Polimorfismo: Utilizar clases base que puedan ser especializadas según el tipo de cita o usuario.

Patrones de Diseño Relevantes:

El modelo MVVC es una variación del patrón MVC que introduce el concepto de ViewModel para mejorar la separación de preocupaciones y la flexibilidad en el desarrollo de aplicaciones, especialmente en aplicaciones de interfaz de usuario complejas. Aquí está la descripción modificada según lo solicitado:

Singleton: Asegurar que componentes clave, como la conexión a la base de datos, sean instanciados una sola vez.

Factory Method: Crear objetos de manera centralizada para facilitar futuras modificaciones en la creación de instancias.

Habilidades Necesarias

Análisis de Requisitos:

Capacidad para identificar las necesidades del usuario final y documentar los requisitos funcionales del sistema.

Diseño de Software:

Crear diagramas UML (como de casos de uso, clases y secuencias) para representar la estructura del sistema y sus interacciones.

Desarrollo y Programación:

Conocimientos en lenguajes orientados a objetos en C# y desarrollo en WPF para la implementación del sistema.

Gestión de Bases de Datos:

Diseño de bases de datos para almacenar la información de citas y pacientes, asegurando la protección de datos.

Pruebas y Validación:

Desarrollo de pruebas para asegurar el correcto funcionamiento del sistema y que los requisitos se cumplan correctamente.

UML's Planeados

UML (Lenguaje Unificado de Modelado):

Se utilizará para crear diagramas visuales que representen el sistema y sus interacciones, como:

Diagrama de Casos de Uso: Para mostrar las interacciones entre los usuarios y el sistema.

Diagrama de Clases: Para detallar las entidades y sus relaciones en el sistema.

Diagrama de Secuencia: Para mostrar el flujo de interacciones entre objetos.