

Plan de Estudio del Sistema de Reservas: Barber-Soft

Guía paso a paso para principiantes

Resumen

Este plan es para construir una página donde la gente reserve citas en una barbería. Primero aprenderemos a guardar nuestros avances, luego a organizar la información y finalmente a hacer que la página se vea bien en el celular.

Introducción: La Ruta Ajustada (Git → C# → SQL → Web)

El camino se divide en tres grandes bloques que van desde la configuración del entorno hasta el despliegue de una herramienta funcional y escalable.

Paso	Herramienta	Objetivo Clave
1.1	Git y GitHub	Implementar control de versiones profesional con manejo de ramas.
1.2	SQL Server	Diseñar la base de datos relacional (Barberos, Clientes, Citas).
1.3	C# (Lenguaje)	Dominar la lógica de negocio y la conexión de datos vía ORM.
2.0	WebAPI	Crear el servidor que procesa las reglas de reserva y validación.
3.0	HTML/CSS/Java	Desarrollar la interfaz responsiva y conectarla con el backend.

FASE 1: Preparación de entorno y diseño de proyecto

Objetivo: Preparar las herramientas y diseñar dónde se guardará la información.

1.1. Configuración del Ecosistema:

- Instalación de Visual Studio Code y el SDK de .NET 8 (el motor para que C# funcione).
- Instalación de SQL Server Express y SSMS (tu archivador de datos).

1.2. Control de Versiones:

- Creación de cuenta en GitHub.
- Inicialización de Git en tu carpeta: aprenderás a tomar "fotos" de tu progreso para no perder nada.
- Configuración de Git: Comandos de `init`, `commit` y `push`
- Uso de ramas: una para tu código final (`main`) y otra para hacer pruebas (`dev`).

1.3. Diseño de la Base de Datos:

- Creación de la base de datos `BarberiaDB` En SQL Server.
 - Creación de tablas que se conectan entre sí:
 - Barberos: (Nombre, disponibilidad).
 - Servicios: (Corte de cabello, barba, precios).
 - Citas: (Quién viene, con quién se corta y a qué hora).
-

FASE 2: Desarrollo del servidor:

Objetivo: Escribir el código que da las órdenes y conecta la página con la base de datos.

2.1. Persistencia (Entity Framework Core):

- Configuración del traductor (ORM) para conectar C# con SQL.
- Uso de migraciones para actualizar la base de datos desde el código.

2.2. Construcción de WebAPI:

- Creación de controladores para recibir peticiones de la página web.
- Implementación de funciones CRUD (Crear, Ver, Editar y Borrar reservas).

2.3. Validaciones y lógica:

- Programación de reglas: evitar que un barbero tenga dos citas a la misma hora.
 - Manejo de fechas y horas complejas en el servidor..
-

FASE 3: Interfaz de Usuario (Frontend):

objetivo: Crear la página web que el cliente verá en su celular.

3.1. Frontend Dinámico:

- Maquetación con HTML5: formularios de reserva y listas de servicios.
- Estilizado con CSS3: colores, fuentes y diseño visual de la barbería.

3.2. Adaptabilidad y Conectividad:

- Diseño Responsivo: uso de Flexbox para que la web funcione en celulares.
- Conexión con JavaScript: envío de datos desde el navegador hacia la API de C#.