



## Requerimientos Técnicos:

El aspirante deberá continuar el desarrollo de un sistema web para la gestión de citas médicas, extendiendo la funcionalidad para incluir la agenda de los médicos. Las citas deben asociarse con la disponibilidad del médico, asegurando que las citas solo puedan ser programadas dentro de los horarios establecidos en su agenda.

### 1. Backend (PHP):

- Implementar la lógica para crear, modificar y eliminar citas, validando que solo se programen dentro de los horarios disponibles del médico.
- Relacionar las citas de los pacientes con la agenda del médico (id\_turno).
- Validar que no se puedan agendar citas en horarios ya ocupados o fuera de la disponibilidad del médico.
- Utilizar buenas prácticas de desarrollo, como separación de capas (modelo, vista, controlador) y manejo seguro de datos.

### 2. Base de Datos (SQL):

- La base de datos incluirá las siguientes tablas:
  - **Pacientes:** nombre, apellido, correo, teléfono.
  - **Médicos:** nombre, especialidad, correo, teléfono.
  - **Agenda Médica:** id\_medico, día de la semana, hora\_inicio, hora\_fin (indica la disponibilidad semanal de cada médico).
  - **Citas:** id\_paciente, id\_medico, id\_turno, fecha, hora, estado (confirmada, cancelada).
- Crear las consultas necesarias para:
  - Agregar una nueva cita solo dentro de los horarios disponibles del médico.
  - Modificar o cancelar una cita existente.
  - Obtener una lista de citas programadas para un paciente o un médico en una fecha específica.
  - Consultar la agenda semanal de un médico para mostrar su disponibilidad.

### 3. Frontend (JavaScript y Bootstrap):

- Crear una interfaz amigable y responsive para que el personal médico o administrativo pueda visualizar y gestionar las citas.
- Mostrar un calendario o selector de horas que permita elegir solo los horarios disponibles según la agenda del médico.
- Utilizar Bootstrap para la maquetación y diseño de la interfaz.
- Implementar interacciones con JavaScript para evitar recargar la página al hacer operaciones CRUD (Crear, Leer, Actualizar, Eliminar).
- Mostrar alertas o mensajes claros en caso de errores o confirmaciones (ej. cita creada, modificada o eliminada exitosamente).

## Instrucciones:

### 1. Base del proyecto:

- Te proporcionamos una base del proyecto con las tablas ya creadas en la base de datos y la estructura del backend configurada para que puedas continuar el desarrollo.
- La base de datos incluye:
  - **Tabla de Pacientes:** nombre, apellido, correo, teléfono.
  - **Tabla de Médicos:** nombre, especialidad, correo, teléfono.
  - **Tabla de Agenda Médica:** id\_medico, día de la semana, hora\_inicio, hora\_fin (disponibilidad semanal del médico).
  - **Tabla de Citas:** id\_paciente, id\_medico, fecha, hora, estado (confirmada, cancelada).

### 2. Tareas del aspirante:

- Completar las funcionalidades en PHP necesarias para gestionar las citas, relacionándolas con la agenda del médico.
- Asegurarse de que las citas solo se puedan programar dentro de los horarios disponibles del médico y que no haya conflictos de horarios.
- Desarrollar una interfaz donde se pueda visualizar la agenda del médico y seleccionar los horarios disponibles al agendar una cita.
- Implementar las interacciones necesarias en JavaScript para gestionar la agenda y las citas sin recargar la página.
- Realizar pruebas funcionales para garantizar que todo funcione correctamente.

### 3. Consideraciones:

- El código debe ser claro y bien documentado.
- Se valorará el uso de buenas prácticas en la arquitectura del código, eficiencia en las consultas SQL y el manejo adecuado de errores en el frontend.

### 4. Entrega:

- El proyecto debe entregarse en un repositorio de GitHub, con instrucciones claras para la instalación y prueba.
- Adjuntar video del aplicativo funcionando (Donde se vea la Interfaz web).

## Criterios de Evaluación:

1. Funcionamiento correcto del sistema, especialmente la validación de disponibilidad en la agenda del médico.
2. Calidad del código (estructura, claridad, organización).
3. Uso correcto de PHP y SQL para el manejo de la base de datos.
4. Interfaz bien diseñada y responsive usando Bootstrap.
5. Habilidad para manejar interacciones dinámicas con JavaScript.
6. Buenas prácticas en el control de versiones (commits significativos, organización del repositorio).