	Prueba Desarrollador Fullstack Gobernación	Fecha creación: 31-08-2023
		Versión: 1.0
		Pág: 1y 2

PRUEBA DE DESARROLLADOR FULLSTACK GOBERNACIÓN

Nombre:

Cargo:

Fecha:

La clínica **I4-Especialistas** requiere poder clasificar las citas agendadas por sus pacientes, según una clasificación basada en especialidades, es por esto que han solicitado se desarrolle una solución que permita:


Administrar un maestro de "**Tipos de Cita**", el cual será utilizado en su sistema de agendamiento.

Un "**Tipo de cita**" está definido por las siguientes propiedades:

Propiedad	Tipo	Obligatoria	Descripción
id	numérico	Si	Identificador único de un tipo de cita
name	cadena	Si	Nombre de un tipo de cita
description	cadena	No	Descripción detallada de un tipo de cita
duration_minutes	numérico	Si	duración máxima de un determinado tipo de cita expresada en minutos
color_hex_code	cadena	Si	código hexadecimal de color que identifica al un tipo de cita: Ej: FFEE00

El cliente espera contar con una interfaz administrativa que le permita:

1. Tener una vista paginada de los tipos de cita que existen en el sistema
2. Crear nuevos tipos de cita
3. Eliminar tipos de cita

	Prueba Desarrollador Fullstack Gobernación	Fecha creación: 31-08-2023
		Versión: 1.0
		Pág: 1y 2

Algunos ejemplos de tipos de cita son:

- Cita de revisión odontológica primera vez
- Cita de revisión odontológica control
- Cita medicina general primera vez
- Cita medicina general control
- Cita con especialista de pediatría primera vez
- Cita con especialista de pediatría control

I4 te ha seleccionado para implementar la solución a esta necesidad, por lo tanto requerimos la construcción de una solución cliente-servidor aplicando buenas practicas de diseño, buenas practicas de codificación y utilizando programación reactiva con Java, Spring boot, Webflux (NO spring-mvc) y Persistencia mediante R2DBC (NO JPA, Ni JDBC). Para la interfaz grafica puedes utilizar el framework y librerías de tu preferencia.

Es importante que esta solución tenga los siguientes atributos

- Desarrollo con Paradigma Reactivo (Project Reactor, Webflux)
- Buena organización del código fuente
- Aplicación de principios y patrones de diseño
- Interfaz amigable e intuitiva para el usuario final

Son deseables y se valoran mucho si se cumplen:

- Incluir test unitarios

Cuando la solución se encuentre implementada puedes compartirnos vía mensaje de correo electrónico, el repositorio GIT donde ha quedado subida y las instrucciones para probarla.

¡Muchos éxitos...!