

**Prueba técnica:** Ingeniero Microservicios.

**Descripción de la necesidad:**

El área de negocio de Colmedica requiere de un mecanismo para registrar la información de sus clientes, ya que desean conocer cuales son sus preferencias a la hora de seleccionar el plan de medicina prepagada.

Dentro de esta necesidad requieren registrar los siguientes datos del cliente:

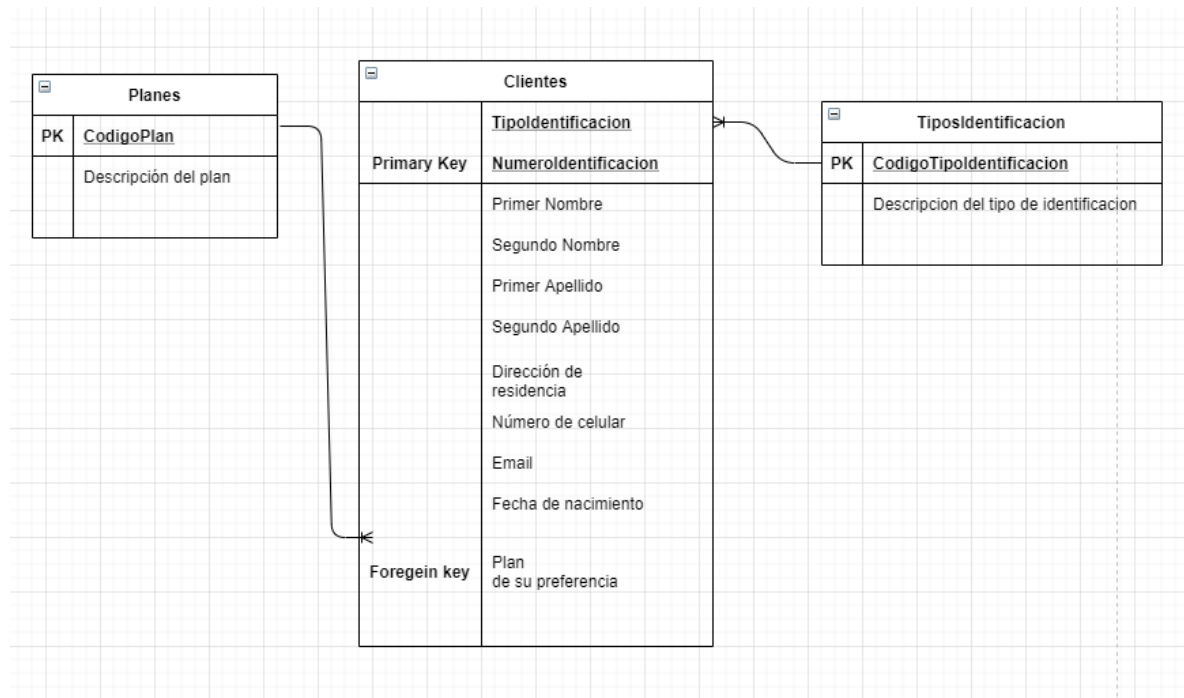
- Tipo de documento de identificación. (Se debe seleccionar de una lista de tipos de identificación)
- Numero de documento de identificación
- Fecha de nacimiento
- Primer Nombre
- Segundo Nombre
- Primer Apellido
- Segundo Apellido
- Dirección de residencia
- Número de celular
- Email
- Plan de su preferencia. (Se debe seleccionar de una lista de planes)

Para ello y como propuesta a la necesidad se requieren los siguientes 3 artefactos:

- Construir una base de datos que permita realizar la respectiva persistencia de los datos de los clientes.
- Construir un microservicio desarrollado en Java, usando el framework spring boot que realice las respectivas operaciones (CRUD) para administrar la información de los clientes.
- Construir una interfaz gráfica usando Angular, que realice las respectivas operaciones (CRUD) para administrar la información de los clientes. Esta a su vez debe realizar las operaciones anteriormente mencionadas haciendo uso del microservicio previamente desarrollado.

### Insumos para el desarrollo:

Para la persistencia de datos se propone el siguiente modelo relacional:



Como mockup para el desarrollo de la parte de frontend se propone el siguiente mockup:

1. Pantalla principal: Se deberá construir una pantalla donde se puedan listar todos los clientes y a su vez las respectivas opciones de adicionar, modificar y eliminar. En esta pantalla se hará uso de listar los datos usando la lectura de clientes desde el microservicio.

Administración de clientes Colmedica

Agregar Cliente

Tipo Identificación	Numero identificación	Primer Nombre	Segundo Nombre	Primer Apellido	Segundo Apellido	Fecha Nacimiento	Email	Dirección	Celular	Acciones
CC - Cédula de Ciudadanía	123456	Pepito	Andres	Perez	Jimenez	01-01-1900	pepito@com	Calle 123 #56-78	3201234567	<button>Editar</button> <button>Eliminar</button>

2. Al momento de adicionar y/o editar la información del cliente, el sistema me deberá dirigir a una pantalla de detalle para que el usuario registre la información del cliente.

Agregar información del cliente

**Tipo de identificación**

**Numero de identificación**

**Primer Nombre**

**Segundo Nombre**

**Primer Apellido**

**Segundo Apellido**

**Fecha de nacimiento**

**Email**

**Dirección**

**Celular**

**Plan de preferencia**

Guardar

Cancelar

### **Criterios de aceptación a nivel funcional:**

Los criterios de aceptación a nivel funcional, que debe tener la solución que se entrega son los siguientes:

- No se debe permitir registrar un cliente que ya se encuentre registrado, esto se valida con el respectivo tipo y numero de identificación del cliente.
- El email del cliente se debe validar para que solo permita el ingreso de emails con estructura valida. (Regex).
- El formato en se visualice la fecha debe ser dd-MM-yyyy
- La selección de las fechas debe ser a través de un calendario
- Cuando se adicione o se modifique la información del cliente, debe emitir un mensaje de informativo, con el resultado de dicha ejecución.
- Cuando se elimine un cliente, se deberá pedir confirmación al usuario para su respectiva eliminación.

### **Criterios de aceptación a nivel técnico:**

Los criterios de aceptación a nivel técnico que debe tener la solución que se entrega son los siguientes:

#### **Base de datos:**

- Deberá generar la base de datos, con sus respectivas tablas y que estas contengan sus respectivas llaves primarias y foráneas. Se deberá generar un script para la creación de esta estructura.

#### **Backend:**

- Se deberá hacer uso de los respectivos código HTTP de respuesta según corresponda. (200) para una respuesta satisfactoria, (500) para errores no controlados, (404) para cuando no se encuentren resultados, y los demás que crea convenientes.
- Cada método que se exponga debe ser del tipo al que corresponda, GET, PUT, POST, DELETE, PATCH e implementarlos según la necesidad.
- Pruebas unitarias del microservicio

#### **Frontend:**

- Se deberá hacer uso de componentes
- Se deberá hacer uso de pipes
- Se deberá hacer uso de servicios
- Es un plus (Opcional), el uso de bootstrap como librería css para el diseño de las pantallas

### **Artefactos de entrega para evaluar la prueba:**

Una vez se finalice la implementación de la solución que comprende base de datos, backend y frontend, se deberá subir a un repositorio en git que tenga acceso público, desde el cual se pueda realizar la respectiva descarga para realizar la respectiva evaluación.

Adicionalmente de incluir un archivo detallando el proceso de instalación para poder ejecutar la solución anteriormente planteada. Lo puede realizar usando README.md de git o en un archivo PDF a su elección.

*Es un plus, pero NO obligatorio, crear la imagen en Docker tanto para el microservicio como para el frontend, y subirla a un repositorio en Dockerhub con acceso público para su respectiva descarga.*