

---

**Test Software Developer**

**GetechnologiesMx**

---

---

<b>Introducción</b>	<b>1</b>
<b>Herramientas</b>	<b>1</b>
<b>Ejercicios</b>	<b>1</b>
<b>Contactos</b>	<b>3</b>
Larry Ruiz Delgadillo	3
Rony Ricardo Díaz Ortega	4

## Introducción

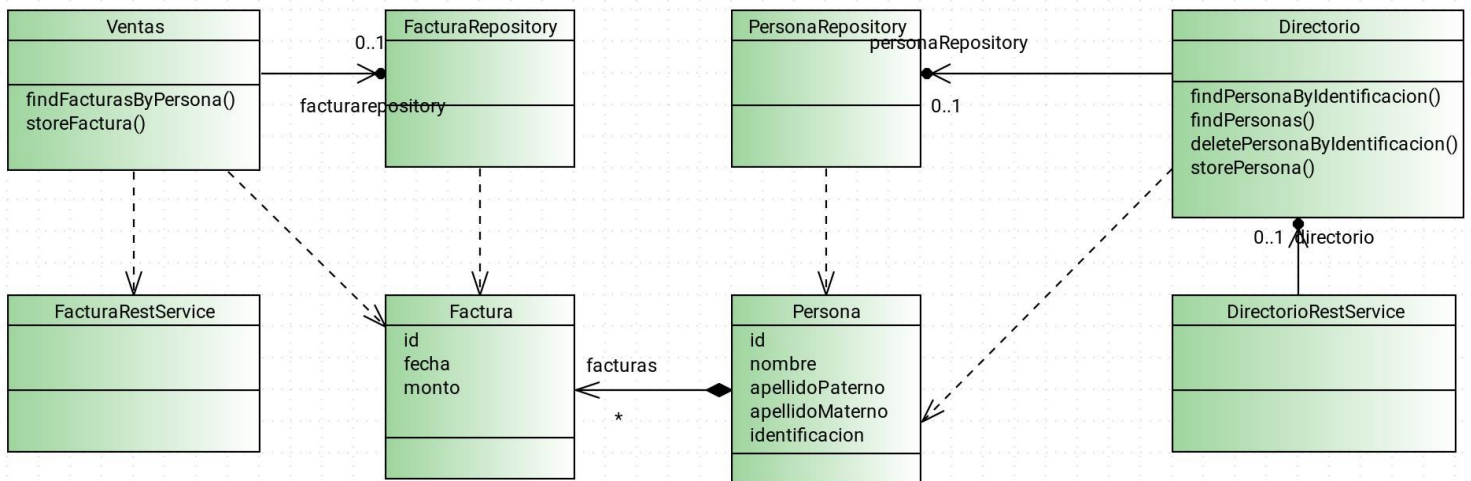
El ejercicio trata de dar evidencia de las capacidades de análisis, diseño, y conocimientos en las tecnologías de interés.

## Herramientas

- Backend - Java, Maven o Gradle, frameworks - spring-boot o quarkus.
- Frontend - NodeJS, NPM, JavaScript puedes utilizar el framework de tu preferencia o sencillamente vanilla, recomendaciones de framework - (Angular, React, Vue, Svelte). - Mobile - Tecnologías recomendadas Ionic (angular), React Native, Java - Kotlin, Swift - Git (cuenta Gitlab o Github con acceso público para verificar la solución).
- Postman

## Ejercicios

1. Crear un proyecto Java Spring Boot o Quarkus.



## 2. Implementar el siguiente modelo de clases en el proyecto.

**DirectorioRestService:** Componente que expone el servicio REST.

**Directorio:** Servicio de negocio con el objetivo de gestionar la información de personas (almacenar, buscar, borrar), el borrado de una persona elimina todos los datos de una persona incluso su facturación.

**Ventas:** Servicio de negocio con el objetivo de gestionar la información de facturación (almacenar, buscar),

**Persona:** Clase del modelo de dominio, que representa a una persona.

**Factura:** Clase del modelo de dominio, que representa las facturas de una persona.

**PersonaRepository:** Componente encargado de la interacción del datasource para obtener, reconstruir los datos de las personas.

**FacturaRepository:** Componente encargado de la interacción del datasource para obtener, reconstruir los datos de las facturas.

Todos los atributos de la clase **Persona** son obligatorios para persistir el objeto excepto `apellidoMaterno`.

Utilizar una base de datos embebida como h2 para el ejercicio.

Como plus puedes agregar logger para mostrar en consola la información (elige uno de los métodos) recomendación de librería `slf4j`, pero puedes ocupar la que desees.

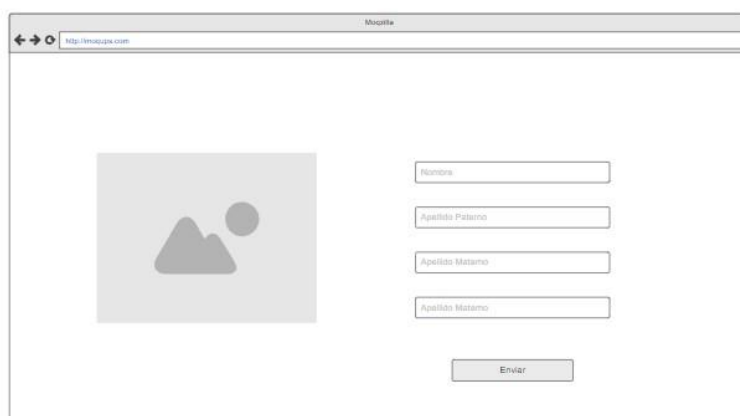
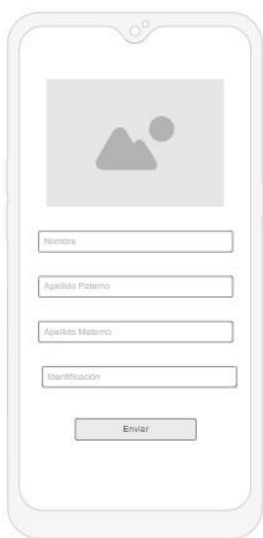
## 3. Realiza un cliente para la API anteriormente creada en el paso 1 y 2 dejamos carta abierta puede ser web o móvil tomando en cuenta las tecnologías mencionadas en la sección de herramientas

utiliza la que prefieras. Puedes utilizar el framework de estilos que prefieras o bien, crear tus propios estilos.

El objetivo es registrar usuarios en el sistema.

Valida los datos que son obligatorios (nombre,ApellidoPaterno,identificación), mostrar mensaje al usuario si no los ha ingresado, de forma reactiva sería excelente.

Puedes darle el diseño que desees a tu aplicación, te dejamos un mockup de ejemplo de diseño.



No olvides subir tu código a GitLab o Github y pasarnos el link.

## Contactos

Larry Ruiz Delgadillo

[lruiz@getechnologiesmx.com](mailto:lruiz@getechnologiesmx.com)

---

Rony Ricardo Díaz Ortega

[ricardo.diaz@gettechnologiesmx.com](mailto:ricardo.diaz@gettechnologiesmx.com)