

# EXAMEN DE INGRESO

26 DE JULIO 2023

ANDRÉS ENCINA TAPIA

Ingeniero en Informática

## Contenido

<b>REQUERIMIENTOS DE EXAMEN - INICIO</b>	3
<b>PLANIFICACIÓN Y ESTIMACIÓN:</b>	4
Visión:	4
Tareas:	4
Tecnologías utilizadas:	4
Lanzamiento requerido:	4
<b>IMPLEMENTACIÓN</b>	5
Implementación Backend:	5
Creación de proyecto modular msvc-ciudadano:	5
MySQL y models.entity:	5
Creación de CiudadanoRepository:	5
Creación de CiudadanoService y CiudadanoServiceImpl:	5
Creación de CiudadanoController:	6
Prueba de mantenedores en Postman:	6
Creación de proyecto modular msvc-tareas:	6
Implementación Frontend:	7
Creación tareas y ciudadanos:	7

## REQUERIMIENTOS DE EXAMEN - INICIO

Según lo conversado copio a continuación Prueba Técnica

Para la siguiente prueba eres libre de dedicar el tiempo que consideres conveniente, lo ideal es que no dediques más de dos días a realizarla y en la respuesta a este correo puedes explicar con tus palabras como afrontaste el desafío y si queda algún punto que por el tiempo no pudiste resolver.

Miranda y El Ojo son dos villanos que tienen controladas las mentes de todo el pueblo de Eastview. Miranda, la bruja, usando sus poderes le impone tareas diariamente a cada ciudadano para lograr que el pueblo se vea lo más normal posible, pero esto consume mucho tiempo y energía de la bruja, por lo que El Ojo le recomendó que buscara y controlara a un desarrollador, para que cree una aplicación web. Ese desarrollador eres tú.

Esta aplicación web debe permitirle a El Ojo, registrar a cada ciudadano de Eastview y programar una lista de tareas para cada día de la semana. Luego Miranda, nuestra Bruja Esmeralda, podrá ver la lista de tareas programadas del día y ordenar la ejecución a cada ciudadano del pueblo.

El Ojo, conocedor de lo último en tecnología, ha dejado la lista de requerimientos:

- Una vista con CRUD (Create Read Update Delete) para administrar la lista de ciudadanos de Eastview.
- Una vista con CRUD para crear las tareas de los ciudadanos para cada día de la semana. Solo considerar días de la semana (de Lunes a Domingo) y no fechas calendario.
- Una vista para la bruja, donde pueda seleccionar un día de semana, y muestre la lista de tareas del día.

Cómo adicional:

Eres libre de seleccionar el Stack tecnológico de tu preferencia, se valorará el uso de lenguajes de alto nivel (Java, PHP), programación Orientada a Objetos y/o el uso de algún framework backend y/o frontend (Spring, Spring Boot, Angular, Reactjs)

Base de datos relacional (preferiblemente PostgreSQL, MySQL)

Todo el código debe estar versionado usando Git, en un repositorio público (preferiblemente Gitlab) y docketizado para su despliegue rápido en ambiente de pruebas

Debes presentar la prueba desplegada en un servidor gratuito. Vercel, Netlify u otro, o en su defecto incluir un archivo LEEME con las instrucciones para su despliegue.

>> La idea de esta prueba es evaluar tus habilidades, si alguna parte del reto se te dificulta y no puedes entregarlo, indícalo en tu respuesta <<

>> El diseño de la GUI (interfaz gráfica del usuario) es libre <<

Al momento de evaluar lo entregado se considerará la legibilidad del código, patrones de diseño, código limpio

Tienes hasta el día miércoles 26 Jul a las 20:00 para enviar tu respuesta, Exitó!

## PLANIFICACIÓN Y ESTIMACIÓN:

Visión:

Visión del proyecto
Aplicación web que permita a “El Ojo”: registrar a cada ciudadano de Eastview y programar una lista de tareas para cada día de la semana. Luego Miranda, nuestra Bruja Esmeralda, podrá ver la lista de tareas programadas del día y ordenar la ejecución a cada ciudadano del pueblo.

Tareas:

COD	TAREA	ESTADO
01	Una vista con CRUD (Create Read Update Delete) para administrar la <b>lista de ciudadanos</b> de Eastview.	Realizado
02	Una vista con CRUD para crear las <b>tareas de los ciudadanos</b> para cada día de la semana. Solo considerar días de la semana (de Lunes a Domingo) y no fechas calendario.	Realizado
03	Una vista para la bruja, donde pueda <b>seleccionar un día de semana, y muestre la lista de tareas del día.</b>	Solo backend vista incompleta

Tecnologías utilizadas:

COD	TECNOLOGÍAS	OBS
	Backend: <ul style="list-style-type: none"><li>- Project: Maven</li><li>- Lenguaje: Java 17</li><li>- Spring Boot: 2.7.14</li><li>- MySQL PostgreSQL 15.2 (pg Admin)</li></ul>	
	Frontend: <ul style="list-style-type: none"><li>- Angular</li><li>- Bootstrap</li><li>- Visual Studio Code</li></ul>	
	Otros: <ul style="list-style-type: none"><li>- IntelliJ IDEA</li><li>- GITLab</li><li>- <b>Netlify</b></li><li>- Docker</li><li>- Postman</li><li>- Packaging: Jar</li></ul>	

Lanzamiento requerido:

FECHA	OBS
-------	-----

**INICIO:** 24 de julio 17:00 hrs

**TÉRMINO:** 26 de Julio 2023 20:00 hrs

## IMPLEMENTACIÓN

### Implementación Backend:

Creación de proyecto modular msvc-ciudadano:

COD	ACTIVIDAD	OBS
01	Descarga inicial en <b>Spring Initializr</b> = <a href="https://start.spring.io/">https://start.spring.io/</a> <b>Dependencias:</b> <ul style="list-style-type: none"><li>- <b>Spring Boot DevTools.</b></li><li>- <b>Spring Web.</b></li><li>- <b>Spring Data JPA</b></li><li>- <b>OpenFeign</b></li><li>- <b>PostgreSQL Drive</b></li><li>- <b>Validation</b></li></ul>	Realizado
02	Creación de microservicio con IntelliJ IDEA	Realizado
03	Pegado de archivo descargado de Spring Initializr en back msvc-ciudadanos	Realizado
04	Configuración de POM de proyecto (padre)	Realizado
05	Configuración de POM de msvc-ciudadanos	Realizado
06	Push con GIT de Modulo	Realizado

MySQL y models.entity:

COD	ACTIVIDAD	OBS
07	Creación de base de datos en PostgreSQL	Realizado
08	Creación de contexto de persistencia JPA/Hibernate	Realizado
09	Se agrega dependencia de validación	Realizado
10	Crea models.entity/ y clase Ciudadano	Realizado
11	Validaciones con starter validation de spring boot	Realizado

Creación de CiudadanoRepository:

COD	ACTIVIDAD	OBS
12	Creación de Package repositories	Realizado
13	Creación de Interface CiudadanoRepositoy	Realizado
14	Optional validador de email	Realizado

Creación de CiudadanoService y CiudadanoServiceImpl:

COD	ACTIVIDAD	OBS
15	Creación de Package services	Realizado
16	Creación de interface CiudadanoRepositoy y clase CiudadanoServiceImpl	Realizado

Creación de CiudadanoController:

COD	ACTIVIDAD	OBS
17	Creación de Package controller	Realizado
18	Creación de clase CiudadanoController	Realizado
19	Documentación	Realizado

Prueba de mantenedores en Postman:

COD	ACTIVIDAD	OBS
20	Insertar ciudadano	Realizado
21	Verificación de validaciones	Realizado

COD	ACTIVIDAD	OBS
22	Listar ciudadanos	Realizado
23	Listar por id	Realizado

COD	ACTIVIDAD	OBS
24	Actualizar	Realizado
25	Verificación de validaciones	Realizado

COD	ACTIVIDAD	OBS
26	Eliminar ciudadanos	Realizado

Creación de proyecto modular msvc-tareas:

COD	ACTIVIDAD	OBS
02	Creación de microservicio	Realizado
03	Descarga de archivo descargado de Spring Initializr en back msvc-tareas	Realizado
04	Configuración de POM de proyecto (padre)	Realizado
05	Configuración de POM de msvc-ciudadanos	Realizado
06	Push con GIT de Modulo	Realizado

## Implementación Frontend:

Creación tareas y ciudadanos:

COD	ACTIVIDAD	OBS
01	Vistas tareas	Realizado
02	Vistas ciudadanos	Realizado
03	CRUD tareas	Realizado
04	CRUD vistas	Realizado
05	Vista detalle por semana	Solo vista principal, falta rescate de información

NOTA: Falto dockerizar bases de datos.

Sistema basado en microservicios, con bases de datos diferentes (MySQL Workbench y PostgreSQL), frontend con angular.

Lamento no haber podido terminar el proyecto como hubiera querido, por tiempo y el proyecto de mi trabajo. Hubiera terminado un excelente proyecto.