



# Ejercicio práctico BackEnd Developer

## Indicaciones Generales

- El ejercicio debe ser cargado a un repositorio Git público, la dirección del repositorio debe ser enviado por correo a las direcciones indicadas.
- El nombre del proyecto debe poseer como prefijo el nombre y apellido del candidato (EJ: `juanPerez_repos`)
- Lea detenidamente lo solicitado en el ejercicio, recuerde que la solución de cada requerimiento será tomada en cuenta para la evaluación.
- Se debe utilizar cualquier ORM para la persistencia y la base de datos se debe crear en la siguiente pagina <https://www.cockroachlabs.com/>.
- La solución se debe realizar en NestJS o Node con TypeScript

## Expectativas técnicas

### Ejercicio 1

- Diseñar un servicio simulado tipo GET

### Ejercicio 2

- Diseñar una API Rest con las operaciones básicas (CRUD).
- Arquitectura del proyecto.
- Buenas prácticas.
- Mapeado de entidad a un DTO (Data Transfer Object).
- Micro commits.

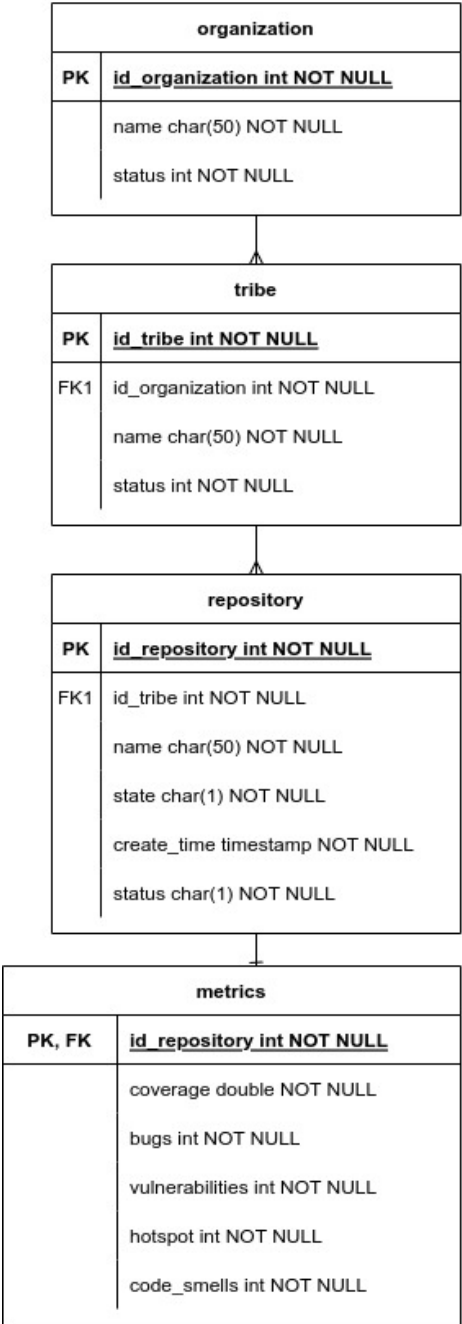
### Ejercicio 3

- Relaciones entre de entidades.
- Control de excepciones.
- Pruebas unitarias por comportamientos.
- Formato de respuesta requerida.
- Uso correcto de enumerados y constantes.

### Ejercicio 4

- Generar reporte en csv a través de un endpoint.

# Modelo



## Organización

Organización propietaria de los repositorios.

## Tribu

Equipo de trabajo de una organización enfocado en un objetivo específico

## Repositorios

Infomación de repositorios centralizado que maneja una tribu

Estados de repositorios: (state)

- E: Enable
- D: Disable
- A: Archived

Estado lógico del registro: (status)

- A: Active
- I: Inactive

## Metricas

Indicadores de calidad de código estático de los repositorios

Cobertura de código:

- Porcentaje sobre 100% expresada como Double

## Ejercicio 1: Servicio simulado (Mock)

**Con el objetivo** de mostrar el tipo de verificación que posee un repositorio  
**Como** administrador de la plataforma

**Quiero** tener un servicio GET simulado que retorne la siguiente respuesta:

```
{
  "repositories": [
    {
      "id": 1,
      "state": 604
    },
    {
      "id": 2,
      "state": 605
    },
    {
      "id": 3,
      "state": 606
    }
  ]
}
```

**Donde:**

- Id: Representa al identificador único del repositorio
- State: Representa el código de verificación actual del repositorio

**Código de verificación**

- 604: Verificado
- 605: En espera
- 606: Aprobado

**Criterios de aceptación**

- **Escenario 1: Consultar servicio simulado:**

**Dado** que quiero obtener los códigos de verificación de mis repositorios

**Cuando** hago una petición tipo GET al servicio simulado

**Entonces** se debe retornar los identificadores de los repositorios con su código de verificación.

**Tareas**

- Crear un servicio mock que permita obtener la respuesta solicitada para la solución del ejercicio número 3.

## Ejercicio 2: Administración de organizaciones

**Con el objetivo** de poder administrar las organizaciones dentro de mi plataforma

**Como** administrador de la plataforma

**Quiero** tener una API que me permita crear, editar, obtener y eliminar

### Criterios de aceptación

- **Escenario 1: Crear organización:**

**Dado** que envío los datos de una organización

**Cuando** consumo el servicio de creación

**Entonces** se debe crear la organización

- **Escenario 2: Editar organización**

**Dado** que modifico los datos de una organización

**Cuando** consumo el servicio de actualización

**Entonces** se debe actualizar la organización

- **Escenario 3: Obtener organizaciones**

**Dado** que quiero obtener una lista de organizaciones

**Cuando** consumo servicio para obtener organizaciones

**Entonces** se debe obtener un listado de organizaciones

- **Escenario 4: Eliminar organizaciones**

**Dado** que quiero eliminar una organización

**Cuando** consumo servicio para eliminar una organización

**Entonces** se debe obtener un listado de organizaciones

### Tareas

- Crear un endpoint para crear una organización.
- Crear un endpoint para actualizar una organización.
- Crear un endpoint para obtener una lista de organizaciones.
- Crear un endpoint para eliminar una organización.

## Ejercicio 3: Servicio para obtener las métricas de un repositorio

Con el **objetivo** de obtener las métricas de un repositorio

**Como** administrador de la plataforma

**Quiero** tener un servicio que me permita obtener los repositorios de una tribu

### Criterios de aceptación

- **Escenario 1: Obtener métricas de repositorios por tribu:**

**Dado** que envío el identificador de una tribu

**Cuando** consumo el servicio para obtener los repositorios

**Entonces** me retornará el detalle de las métricas de los repositorios creados este año

Y que se encuentren habilitados (state: ENABLE)

Y que su cobertura sea superior a 75%

- **Escenario 2: Tribu inexistente.**

**Dado** que envío el identificador de una tribu

**Cuando** consumo el servicio para obtener los repositorios y la tribu no existe

**Entonces** me retornará el siguiente error: 'La Tribu no se encuentra registrada'

- **Escenario 3: Información de verificación.**

**Dado** que envío el identificador de una tribu

**Cuando** consumo el servicio para obtener los repositorios

Y obtengo el estado de verificación de los repositorios desde API Simulada (mock)

**Entonces** me retornará una etiqueta en la respuesta indicando un texto en lenguaje natural del estado de verificación actual de cada repositorio

- **Escenario 4: Tribu no tiene repositorios que cumplan con la cobertura.**

**Dado** que envío el identificador de una tribu

**Cuando** consumo el servicio para obtener los repositorios

Y la tribu no tiene repositorios que cumplan con el 75% de cobertura

**Entonces** me retornará el siguiente error: 'La Tribu no tiene repositorios que cumplan con la cobertura necesaria'

### Tareas

- Crear un solo endpoint que cumpla con los escenarios planteados.
- Consulta para filtrar los repositorios por fecha, estado y porcentaje.
- Consumo API Externa para validar códigos de verificación.
- Mapear estructura de respuesta.
- Pruebas unitarias de cada criterio de aceptación.

## Estructura de la respuesta esperada

```
{
  "repositories": [
    {
      "id": "1",
      "name": "cd-common-utils",
      "tribe": "Centro Digital",
      "organization": "Banco Pichincha",
      "coverage": "35%",
      "codeSmells": 0,
      "bugs": 0,
      "vulnerabilities": 0,
      "hotspots": 0,
      "verificationState": "Verificado",
      "state": "Habilitado"
    },
    {
      "id": "2",
      "name": "cd-common-text",
      "tribe": "Centro Digital",
      "organization": "Banco Pichincha",
      "coverage": "75%",
      "codeSmells": 1,
      "bugs": 0,
      "vulnerabilities": 2,
      "hotspots": 0,
      "verificationState": "En espera",
      "state": "Archivado"
    }
  ]
}
```

// identificador del repositorio  
// nombre del repositorio  
// nombre de la tribu  
// nombre de la organización  
// cobertura de pruebas unitarias  
  
// Estado de verificación (Mock)  
// Estado del repositorio (state)

## Ejercicio 4: Generar reporte CSV métricas repositorio

**Con el objetivo** de obtener con las métricas de un repositorio

**Como** administrador de la plataforma

**Quiero** tener un servicio que me permita generar un archivo .csv con los repositorios de una tribu

### Criterios de aceptación

- **Escenario 1: Generar reporte**

**Dado** que envío el identificador de una tribu

**Cuando** consumo el servicio para generar el reporte de métricas de repositorios

**Entonces** me retornará un archivo .csv con el detalle de la consulta elaborada en el ejercicio número 3, donde cada atributo se representa como una columna.

### Tarea

- Crear un endpoint que genere el reporte csv