Nombre Completo: \_\_ Nehomar Miguel Herrera Rojas \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Fecha: \_\_ 15/01/2023 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Aspiración Salarial: \_\_ 5.500.000 (neto) \_\_\_\_\_

## Contexto

La empresa XYZ necesita realizar gestión de los usuarios del sistema de información ABC, y a usted como ingeniero de desarrollo le solicitan desarrollar parte del sistema para cumpla con ese objetivo. Para esto, le entregan lo siguiente:

Tabla

Descripción generada automáticamente con confianza media

* La pantalla debe ser como lo muestra la imagen.
* Debe tener la opción de crear usuarios con los siguientes datos adicionales:
  + Rol (Administrador, Soporte, Asistente)
  + Cargo (Auditor, Administrador, Desarrollador, Gerente)
  + Contraseña
* La contraseña no debe ser visible en ninguna capa.
* Dentro de la pantalla deben realizarse las operaciones básicas de una entidad (CRUD) Crear, Listar, Editar y Eliminar.
* Un usuario puede tener varios roles, pero un solo cargo.

## Prueba

Los entregables que se requieren para esta prueba son los siguientes:

1. Modelo entidad relación

Diagrama

Descripción generada automáticamente

1. Diagrama de flujo

Diagrama

Descripción generada automáticamente

1. Clases en JAVA (lógica de negocio, entidades)

Clases: *Usuario*, *Cargo*, *Rol*

1. Persistencias JPA y consultas en JPQL

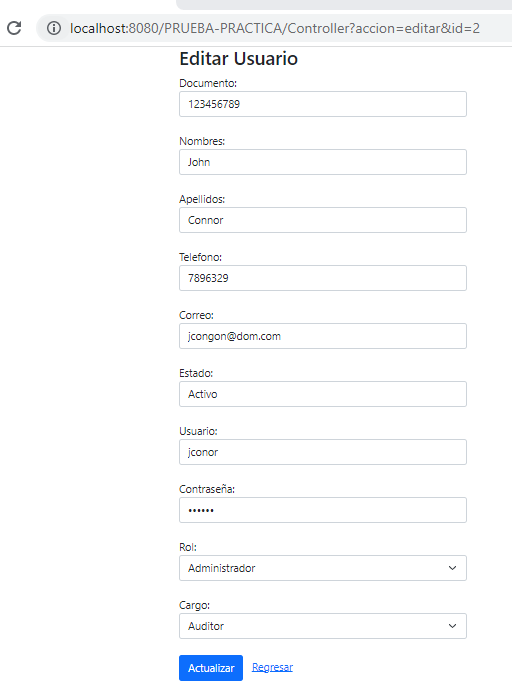
Se maneja persistencia mediante el DAO (*UsuarioDAO*), que implementa métodos un CRUD, *List* (*get* de todos los usuarios), list(*id\_usuario*), add (*post* del objeto Usuario), edit (*put* actualización de datos del usuario mediante el *id\_usuario*) y eliminar (*delete* mediante el *id\_usuario*).

1. Diseño para la(s) pantalla(s) que vea necesario

***Listado de Usuarios***



***Editar Usuario***



***Crear Usuario***



1. Mejoras que puedas aportar al requerimiento

El requerimiento identifica de forma clara la funcionalidad del entregable, la cual es realizar un CRUD del usuario, sin embargo, referenciando al aspecto técnico, de arquitectura y de procesos, se pudiesen disgregar. Por ejemplo, tipo de arquitectura a utilizar (monolito, MVC, orientado a servicios, etc.), tipo de encriptación para la *contraseña*, sugerencia de los tipos de datos para el campo de *documento*, validar si el campo *estado* estaría separado de la entidad usuario proyectando que los estados del mismo se puedan modificar, establecer como regla de negocio las combinatorias y dependencias entre *roles* y *cargos*. Para el front, también se pudiese sugerir ciertas especificaciones, como son: identificar si es reactivo, validaciones de los campos de los distintos formularios, como tipos de datos, longitud, obligatoriedad, etc. En particular para el estilizado del desarrollo se utilizó la librería *Bootstrap*.

Por el detalle de lo anterior, quedaría abierto y sugestivo al desarrollador qué tecnologías y cómo utilizarlas para llegar requerimiento final, el CRUD.

1. Explicación de como funcionaría el módulo que se va a desarrollar

Se tendría una pantalla principal que listaría en una tabla los usuarios registrados en el sistema, donde se muestran opciones rápidas para su administración (*editar*, *eliminar*) por cada registro. Así mismo, se dispondría de un botón para la creación rápida de un usuario, que redirecciona a un formulario para diligenciar los datos correspondientes. El editar un usuario, redirecciona a una nueva vista con todos sus datos precargados, donde se modificarían los mismo según las reglas del negocio. Para eliminar un usuario, se dispone de un botón donde con único se elimina al momento el registro.

## Condiciones

El tiempo destinado para realizar esta prueba es de 4 horas contadas a partir del envío del correo. Todo deberá quedar comprimido en un archivo llamado “PRUEBA\_PRACTICA\_*NOMBRE\_COMPLETO*.zip”.

**Cualquier duda o inquietud por favor comunicarla inmediatamente.**