



AVISO DE CONFIDENCIALIDAD

“El presente documento es de carácter confidencial. Su lectura está autorizada al personal del CONSUL.Ti y restringida al público en general. La distribución o publicación de este documento sin previa autorización está completamente prohibida”.



Antecedentes

Se encuentra en la Fase Práctica de la Prueba Inicial del Proceso de Reclutamiento de CONSUL.Ti. En este punto, usted debe tener acceso a un repositorio de GIT con su nombre en donde deberá agregar los archivos de acuerdo a la siguiente estructura:

```
PRUEBA_INICIAL
a. PRÁCTICA
    i. Backend
    ii. Frontend
```

Durante la evaluación

El postulante debe tener en cuenta que su prueba práctica será evaluada utilizando los siguientes criterios:

1. Arquitectura del proyecto adecuada (uso adecuado de capas)
2. Uso del archivo Readme para explicar los pasos para levantar su prueba e información importante para la ejecución de la misma.
3. Código correctamente documentado (sólo funciones que involucren lógica de negocio)
4. Buen manejo del ORM
5. Separación correcta de responsabilidades
6. Solución escalable

Adicional debe considerar lo siguiente:

1. La aplicación debe estar **separada FRONTEND y BACKEND**
Proyecto Frontend
Proyecto Backend
2. El lenguaje Frontend es JAVA.
 - a. Puede utilizar JAVA Maven Project o Spring Boot
3. El lenguaje Backend es Angular / Javascript + HTML / JQuery
4. Para el estilo pueden usar sus propios estilos o usar frameworks de CSS como Bootstrap Material, etc.
5. Base de Datos MySQL, PostgreSQL, MongoDB
6. Comunicación entre Frontend y Backend mediante **api rest**
7. Scripts de creación de la base deben estar en el repositorio
8. Debe realizar comentarios en el código
9. Es deseable usar archivos README en el repositorio para instrucciones generales.



Descripción

Se solicita realizar un sistema de ingreso y administración de información de usuarios

Requerimientos Iniciales

1. Se debe consumir el siguiente endpoint <https://dummyjson.com/users> el cual la información devuelta corresponde a usuarios y debe guardarse en la base de datos (**esta acción debe de realizarse solo la primera vez**), posterior se debe trabajar con la información insertada en la BD, los campos a guardar son:
 - a. firstname
 - b. lastname
 - c. age
 - d. email
 - e. image
 - f. status (setearle por default 'Activo')

* El campo status no viene en el json, ese se debe setear por default en el valor indicado.

2. Realizar una pantalla que permita ejecutar el **listado y eliminación(eliminado lógico)** de usuarios registrados
3. Realizar una opción que permita agregar:
 - Número telefónico
 - Fecha de nacimiento
 - Estado civil (seleccionar: Soltero o Casado)
 - Observación.

*Se deben agregar los campos necesarios en la tabla de base de datos.

El flujo es el siguiente:

- a. Debe ser una opción por usuario
 - b. Se debe levantar un popup solicitando la información adicional
 - c. Si el estado civil es 'Casado' se debe pedir el # de hijos, caso contrario debe guardarse 0.
 - d. Ingresar una observación
4. Agregar filtros de búsqueda, como: Estado Civil y la edad (filtro que permita buscar por los usuarios mayores a la edad ingresada, por ejemplo: visualizar los usuarios que tengan mas de 30 años.

* Para los usuarios que no posean la información adicional se debe mostrar en esos campos un 'N/A'

Requerimientos Intermedios

1. Encapsular el sistema dentro de un mecanismo de logueo de usuario/contraseña
2. En el CRUD de usuarios, agregar la modificación de usuarios (considerar solo los campos: firstname, lastname, age, email)
3. En el CRUD de usuarios, al editar el correo validar que sea un correo válido.



Requerimientos Avanzados

1. En el CRUD de usuarios, agregar la modificación de imágenes.
2. Envío de notificación al email ingresado, cuando se realice la modificación de la data de un usuario.
3. Dockerizar los ambientes y el despliegue del mismo

