

COLSUBSIDIO – GERENCIA DE TRANSFORMACIÓN DIGITAL

# PRUEBA TÉCNICA DESARROLLADOR FULLSTACK

**BACK – JAVA ( $\geq 1.8$ ) + SPRING BOOT**  
**FRONT - ANGULAR**

**AUTOR:** Jeffrey Cortés

**NEGOCIO:** Operaciones de Subsidio - Afiliaciones

**ACTIVO DIGITAL:** Portal Personas



**Revisión / Historial del documento**

Número Versión	Acción, C,M,D,A*	Fecha Acción**	Resumen Cambios	Responsables de la Acción	Aprobado por	Distribuido a
1	C	18/11/2022	Creación	Jeffrey Cortés		

\*C =Creación, M =Modificación, D =Distribución, A = Aprobación (Excluyentes).

\*\*El contenido de la columna Fecha Acción (dd-mm-aaaa) es requerido y se diligencia de la siguiente manera: Escribir para las acciones de Creación y Modificación del documento la fecha de iniciación de la acción, para la de Aprobación la fecha de su finalización y para Distribución la fecha de envío del documento.



Capa	Backend
Tecnología:	Java
Versión	>= 8
Framework:	Spring Boot + Maven
Plazo	<b>5 días</b> , si tienes algún inconveniente con el tiempo comunícate con nosotros

### Descripción de la prueba

Se requiere una aplicación que exponga una API REST FULL para la administración de usuarios, en el que todos los endpoints acepten y retornen solamente JSON, inclusive para los mensajes de error.

Todos los mensajes deben seguir el formato: {"mensaje": "mensaje de error"}. Puede complementarlo a su consideración.

#### 1. Endpoint para registrar usuario

Este endpoint deberá recibir un usuario con los campos "nombre", "correo", "contraseña", más un listado de objetos "teléfono", respetando el siguiente formato:

```
{
  "name": "Juan Rodriguez",
  "email": "juan@rodriguez.org",
  "password": "hunter2",
  "phones": [
    {
      "number": "1234567",
      "citycode": "1",
      "contrycode": "57"
    }
  ]
}
```

En caso de éxito, se debe retornar el usuario y los siguientes campos:

- id: id del usuario (puede ser lo que se genera por el banco de datos, pero sería más deseable un UUID).
- created: fecha de creación del usuario.
- modified: fecha de la última actualización de usuario.
- lastLogin: del último ingreso (en caso de nuevo usuario, va a coincidir con la fecha de creación).
- token: token de acceso de la API (JWT).
- isActive: Indica si el usuario sigue habilitado dentro del sistema.

Si el correo ya está registrado en la base de datos, se debe retornar un error con el mensaje "El correo ya está registrado".

**Reglas de negocio:**

- El correo debe seguir una expresión regular para validar que formato sea el correcto (jeff@colsubsidio.com).
- La clave debe seguir una expresión regular para validar que el formato sea el correcto. El valor de la expresión regular debe ser configurable.
- Se debe generar un token JWT y deberá ser persistido junto con el usuario.

**2. Endpoint para consultar usuarios**

Este endpoint deberá recibir los campos que usted considere necesarios para consultar un listado de usuarios o un usuario específico y la respuesta debe ser en formato JSON.

**Criterios de aceptación:**

1. Entrega en un repositorio público (github o bitbucket) con el código fuente y script de creación de BD.
2. Banco de datos en memoria o liviana. Ejemplo: HSQLDB o H2.
3. Proceso de build vía Maven.
4. Persistencia con JPA. Ejemplo: EclipseLink, Hibernate u OpenJPA.
5. Readme explicando cómo probarlo.
6. Diagrama de la solución.
7. Swagger (deseable).
8. Pruebas unitarias (opcional, por lo menos un 20% de cobertura).

<b>Capa</b>	Frontend
<b>Tecnología:</b>	Angular
<b>Versión</b>	>= 2 + typescript

**Descripción de la prueba**

Se requiere una web app que permita visualizar y buscar los usuarios registrados en el reto de back-end.

**1. Formulario de visualización de usuarios**

Se requiere un formulario que muestre por defecto todos los usuarios registrados en la base de datos por el API de gestión de usuarios cuya estructura incluya las siguientes secciones:

- Encabezado con logo y menú tipo caja (Facebook).
- Componente de filtrado por los campos que usted considere relevantes.
- Botón para buscar o filtrar usuarios.
- Tabla de resultados en la que se muestren los campos que usted considere deban visualizarse.
- Footer de 3 secciones que contenga:
  - Su información de contacto personal.
  - Links a sus redes sociales.
  - Año, derechos y licencia de la webapp.

**Reglas de negocio:**

- Mientras se realiza la carga de usuarios, se debe mostrar una animación o texto que le indique al usuario que se está procesando la consulta o búsqueda de usuario(s).
- Los campos deben recibir los caracteres que correspondan según el modelo de datos (campos de números solo deben permitir números, correo debe ser validado con expresión regular, etc).
- Incluir en el package.json los comandos para compilar en debug y compilar para distribuir (productivo).
- Mostrar los mensajes de error en un modal con la descripción del error clara.

