## **Caso de uso aplicado**

**🎯 Caso de Uso: Registro de Abonados a Frentes de Seguridad**

**Objetivo:**

**Crear un sistema que permita registrar ciudadanos como abonados a un frente de seguridad, guardando su información y verificando su email mediante un código de validación.**

**Datos requeridos al registrar:**

* **Nombres**
* **Apellidos**
* **Número de identificación**
* **Correo electrónico**
* **Número de celular**
* **Barrio**
* **Dirección exacta**
* **Frente de Seguridad asociado (lista desplegable con frentes registrados)**
* **Tipo de documento (CC, CE, TI)**
* **Fecha de nacimiento**
* **Sexo (M/F/No binario/Prefiero no decirlo)**
* **¿Tiene antecedentes? (Sí/No - con justificación si aplica)**
* **Foto opcional (perfil o cédula escaneada)**

**Flujo del sistema:**

1. **Formulario Web** 
   * **Se presenta un formulario con los campos anteriores.**

* **Validaciones básicas (campos obligatorios, email válido, número de documento único, Número de celular, Nombres, Apellidos, Frente de Seguridad asociado, tipo de documento).**

1. **Backend (Node.js o Python):**
   * **Recibe los datos del formulario.**
   * **Verifica que el número de identificación y correo no estén ya registrados.**
   * **Genera un código aleatorio de 6 dígitos.**
   * **Guarda todos los datos en una base de datos (ej. PostgreSQL/MySQL o MongoDB) con el campo verificado: false.**
2. **Email de verificación:**
   * **El sistema envía el código de 6 dígitos al correo del usuario con un mensaje tipo:**

**"Tu código de verificación para el Frente de Seguridad es: 841205"**

1. **Pantalla de verificación:**
   * **El usuario ingresa el código recibido.**
   * **El backend compara con el código almacenado en la base de datos.**
   * **Si coincide, actualiza el estado del abonado a verificado: true.**

**🛠️ Tecnologías sugeridas:**

* **Backend: Node.js (Express) o Python (FastAPI – Django)**
* **Base de datos: MongoDB / MySQL – PostgreSQL - Otro**
* **Email: Nodemailer (Node.js) o smtplib/sendgrid (Python) u otro conocido**
* **Frontend: HTML/React/Vue.js u otro**

**✅ Validaciones clave**

* **No permitir correos o cédulas repetidas.**
* **Código de verificación debe expirar en 10-15 minutos (opcional).**
* **Registro de logs o trazabilidad del proceso. (opcional)**

**🔄 Extensiones posibles:**

* **Registro masivo desde archivo CSV.**
* **Formulario de administración con lista de abonados verificados/no verificados.**

**Criterios de Evaluación:**

✅ Código modular y limpio.  
✅ Uso eficiente de base de datos.  
✅ UI responsiva y clara.  
✅ Código bien documentado.