**LABORATORIO OBLIGATORIO: *Integración de AWS SNS en una Aplicación MVC Core para Enviar Notificaciones en Tiempo Real***

**Contexto:** Imagina que trabajas en una empresa de comercio electrónico y estás desarrollando una plataforma web para gestionar los registros de usuarios y la creación de pedidos. Un aspecto clave de la plataforma es que los usuarios deben recibir notificaciones sobre eventos importantes como su registro exitoso o la creación de un nuevo pedido. Estas notificaciones pueden ser enviadas por **correo electrónico** o **SMS**, y es necesario que el sistema sea escalable y fácil de gestionar.

**Objetivo:** El objetivo de esta práctica es desarrollar una aplicación **ASP.NET Core MVC** que interactúe con **Amazon SNS (Simple Notification Service)** para enviar notificaciones a los usuarios cuando ocurran ciertos eventos importantes en la plataforma. Estos eventos pueden incluir:

1. **Registro de un nuevo usuario.**
2. **Creación de un nuevo pedido.**

**Especificaciones del Proyecto:**

**1. Requisitos Funcionales:**

**Interfaz de Usuario (UI):**

* Crear una vista sencilla para permitir el registro de un nuevo usuario.
* Mostrar una vista de éxito cuando un usuario se registre correctamente.
* Enviar una notificación por **correo electrónico** o **SMS** (dependiendo de la configuración) a los usuarios cuando se registre correctamente.
* Crear un modelo básico de pedido que permita a los usuarios hacer compras, y enviar una notificación a través de SNS cuando un pedido se crea exitosamente.

**SNS Integration:**

* Integrar **AWS SNS** en el proyecto MVC Core.
* Realizar las siguientes acciones:
  + **Configurar un tema SNS** (para el registro de usuarios y la creación de pedidos).
  + **Suscribir direcciones de correo electrónico y/o números de teléfono** al tema SNS.
  + **Enviar notificaciones a un tema SNS** cada vez que se registre un nuevo usuario o se cree un nuevo pedido.
  + Configurar **credenciales de AWS** (mediante el uso de variables de entorno o configuración en el archivo appsettings.json).

**Manejo de Excepciones:**

* Implementar manejo de excepciones adecuado para cuando no se pueda enviar la notificación (por ejemplo, si AWS SNS no está configurado correctamente, si la suscripción no es válida, etc.).
* Mostrar un mensaje de error en la interfaz si ocurre un problema al intentar enviar la notificación.

#### ****Requisitos Técnicos:****

1. **Uso de MVC Core:**
   * La aplicación debe seguir el patrón **MVC** (Modelo-Vista-Controlador).
   * Se debe crear al menos un **modelo de usuario** con los campos básicos (nombre, correo electrónico, contraseña) y un **modelo de pedido** con los campos correspondientes (nombre del producto, cantidad, precio, etc.).
2. **Integración de AWS SNS:**
   * Usar el SDK de **AWS para .NET** para integrar **SNS** en su aplicación.
   * Se deben configurar las credenciales de AWS para acceder a SNS desde la aplicación.
   * Se debe crear un **tema SNS** en la consola de AWS y obtener el ARN del tema para poder usarlo en tu aplicación.
   * La notificación debe enviarse a los **suscriptores** del tema (usuarios que se han registrado para recibir correos electrónicos o SMS).
3. **Almacenamiento y Gestión de Usuarios:**

* Los usuarios deben ser almacenados en una base de datos **local** (SQL Server, SQLite o cualquier otra base de datos compatible).
* Cuando un usuario se registre correctamente, la aplicación debe generar un mensaje de bienvenida y enviarlo a través de SNS a los suscriptores.

1. Notificación de Pedidos:

 Después de realizar un pedido, se enviará una notificación a los usuarios registrados indicando que el pedido ha sido creado exitosamente.

 El mensaje de SNS debe contener detalles relevantes del pedido (por ejemplo, nombre del producto, cantidad, precio total).

**Requisitos de Seguridad:**

1. **Gestión de credenciales AWS:**
   * Debes gestionar las credenciales de AWS de forma segura. Es recomendable que utilices variables de entorno o un archivo de configuración (por ejemplo, appsettings.json) para almacenarlas, pero no deben ser hardcodeadas directamente en el código.
2. **Control de acceso:**
   * La aplicación debe permitir que solo los usuarios registrados y autenticados puedan acceder a las funcionalidades de envío de notificaciones (por ejemplo, solo un usuario autenticado puede hacer un pedido).