**PRACTICA OBLIGATORIA AZURE SERVICE BUS**

Fecha de entrega (antes del examen de certificación)

**Escenario**: Una empresa de pagos electrónicos necesita un sistema que procese **pagos con tarjeta y transferencias bancarias**. El sistema debe:

* **Recibir transacciones** desde una interfaz de usuario.
* **Enviar transacciones a una cola** para su procesamiento.
* **Procesar transacciones en segundo plano** y almacenarlas en una base de datos.
* **Utilizar temas y suscripciones** para notificar eventos importantes a otros sistemas (por ejemplo, auditoría y notificaciones a clientes).

**Especificaciones del Proyecto**. Tecnologías a utilizar:

* **ASP.NET Core MVC (.NET 8)**
* Azure Service Bus (Colas y Temas)
* Entity Framework Core (para almacenamiento en base de datos)
* Bootstrap

**Funcionalidades**:

* Módulo de Envío de Transacciones
  + Formulario para ingresar monto, tipo de transacción (pago con tarjeta o transferencia), cuenta de destino, y detalles adicionales.
  + Botón para enviar transacción a la cola.
* Procesamiento de Transacciones (Cola)
  + Un servicio en segundo plano consumirá las transacciones de la cola "TransaccionesPendientes".
  + Simulación de validación de la transacción (saldo suficiente, autenticación, etc.).
  + Si la transacción es exitosa, se almacena en la base de datos.
  + Si la transacción falla, se envía a una cola de mensajes no entregados (DLQ).
* Notificación de Transacciones (Temas y Suscripciones)
  + Un tema "EventosTransacciones" notificará eventos importantes:
    - "Transacción aprobada"
    - "Transacción fallida"
  + Sistemas suscriptores:
    - Auditoría → Recibe todas las transacciones para análisis.
    - Notificaciones al Cliente → Recibe solo transacciones exitosas y envía confirmación por email

**Interfaz de Usuario:**

* Lista de transacciones pendientes, exitosas y fallidas.
* Botón para recibir transacciones desde la cola en tiempo real.
* Panel de notificaciones con mensajes de eventos recibidos desde el tema.

**Modelo de Datos:**

**1. Entidad: Transacción**.

Esta entidad representa una transacción que se procesa, ya sea un pago con tarjeta o una transferencia bancaria. Incluye la información básica de la transacción, el estado, y otros detalles necesarios para el procesamiento.

Propiedades:

* TransaccionId (int) - Identificador único de la transacción (clave primaria).
* Monto (decimal) - Monto de la transacción.
* TipoTransaccion (string) - Tipo de transacción (puede ser 'PagoConTarjeta' o 'TransferenciaBancaria').
* CuentaDestino (string) - Cuenta de destino de la transacción (por ejemplo, número de cuenta bancaria o número de tarjeta).
* DetallesAdicionales (string) - Información adicional sobre la transacción (por ejemplo, concepto de pago).
* Estado (string) - Estado de la transacción (puede ser 'Pendiente', 'Exitosa', 'Fallida').
* FechaCreacion (DateTime) - Fecha y hora en que se creó la transacción.
* FechaProcesamiento (DateTime?) - Fecha y hora en que se procesó la transacción (puede ser nula si no se ha procesado).
* FechaNotificacion (DateTime?) - Fecha y hora en que se envió la notificación (puede ser nula si no se envió).

**2. Entidad: EventoTransaccion**

Esta entidad representa los eventos importantes que ocurren durante el procesamiento de la transacción, como la aprobación o el fallo de la misma.

**Propiedades:**

* EventoId (int) - Identificador único del evento (clave primaria).
* TransaccionId (int) - Identificador de la transacción relacionada (clave foránea).
* TipoEvento (string) - Tipo de evento (por ejemplo, 'Transacción aprobada', 'Transacción fallida').
* Descripcion (string) - Descripción del evento.
* FechaEvento (DateTime) - Fecha y hora en que ocurrió el evento.

**3. Entidad: Notificacion**

Esta entidad representa las notificaciones enviadas a los sistemas suscriptores como parte de los eventos de transacciones exitosas.

**Propiedades:**

* NotificacionId (int) - Identificador único de la notificación (clave primaria).
* TransaccionId (int) - Identificador de la transacción relacionada (clave foránea).
* EmailCliente (string) - Dirección de correo electrónico del cliente que recibirá la notificación (solo para transacciones exitosas).
* EstadoNotificacion (string) - Estado de la notificación (puede ser 'Enviada', 'Fallida').
* FechaEnvio (DateTime) - Fecha y hora en que se envió la notificación.

**4. Entidad: ColaTransaccionesPendientes**

Esta entidad no se almacenaría en la base de datos, sino que representaría la cola de transacciones en Azure Service Bus. El sistema consume estas transacciones desde la cola y las procesa en segundo plano.

**Propiedades:**

* TransaccionId (int) - Identificador de la transacción.
* Monto (decimal) - Monto de la transacción.
* TipoTransaccion (string) - Tipo de transacción.
* Estado (string) - Estado de la transacción.

**5. Entidad: ColaTransaccionesNoEntregadas (DLQ)**

Esta entidad tampoco se almacena en la base de datos, sino que representa la cola de transacciones no entregadas o fallidas, las cuales deben ser revisadas o procesadas nuevamente.

**Propiedades:**

* TransaccionId (int) - Identificador de la transacción.
* Error (string) - Descripción del error que causó el fallo de la transacción (por ejemplo, "Saldo insuficiente").
* FechaError (DateTime) - Fecha y hora en que ocurrió el error.

**6. Relaciones entre las entidades:**

* Una Transacción puede generar varios EventosTransaccion
* Una Transacción puede generar una Notificacion si es exitosa.
* Las transacciones son enviadas y procesadas desde la ColaTransaccionesPendientes.
* Las transacciones fallidas se almacenan en la ColaTransaccionesNoEntregadas (DLQ).

7. Publicación:

* Publicar la aplicación en Azure App Services
* Base de datos Azure Sql

**Evaluación**:

* Funcionalidad Completa (40%) → La aplicación envía, procesa y recibe mensajes correctamente.
* Uso correcto de Azure Service Bus (30%) → Implementación de colas, temas y suscripciones.
* Interfaz de Usuario (20%) → Diseño claro y funcional para interactuar con el sistema.
* Documentación y Código Limpio (10%) → Explicación del flujo y buenas prácticas de desarrollo.
* Entregar url publicada en Azure