Ejercicio

El cliente necesita que se le desarrolle una interfaz de tipo ATM (cajero automático) de acuerdo a las siguientes especificaciones:

Operaciones

- La home es una página donde se puede ingresar un número de tarjeta. Luego de que el número de tarjeta es ingresado por el usuario y haga click en el botón "Aceptar", se envía una petición a la base de datos. Si la tarjeta es encontrada y no se encuentra bloqueada, el usuario es enviado a una página donde debe ingresar el PIN de la tarjeta que ingreso previamente, de otro modo, si la tarjeta está bloqueada o no existe, el usuario es enviado a una página con un mensaje de error.
- Una vez que el usuario ingresa el PIN y hace click en el botón "Aceptar", los datos ingresados son validados por el sistema utilizando la base de datos. Si los datos son correctos, el usuario es enviado a la página de Operaciones, de otro modo, se muestra un mensaje de error. El usuario puede ingresar un PIN invalido hasta una cantidad máxima de 4 veces. En la cuarta vez, se muestra una página que informa que la tarjeta se bloqueó, y ese cambio debe impactarse en la base de datos.
- Dependiendo de la operación que el usuario elija, la página de "Balance" o "Retiro" será cargada.
- Si el usuario elije "Balance", entonces se creará un registro de operación en la tabla con el ID de la tarjeta, el momento en el tiempo y el código de la operación.
- Si el usuario elije "Retiro", entonces luego de que ingrese la cantidad de dinero y haga click en el botón "Aceptar", el sistema valida que la cantidad no exceda el balance en la tarjeta. En caso de exceder el balance, se muestra una página de error. De otro modo, se creará un registro de operación con el ID de la tarjeta, el momento en el tiempo, el código de la operación y la cantidad de dinero que se retiró. Además, se modifica el balance de la tarjeta en la tabla de Tarjetas. Luego de se hayan impactado todos los cambios en la base de datos, se mostrará una página con el resultado de la operación.

Inteface

1. Home: contiene un campo para que se cargue el número de la tarjeta, debe tener un teclado numérico con dígitos del 0 al 9, un botón "Aceptar" y otro botón "Limpiar". Debe sugerirle al usuario que ingrese un número de tarjeta de 16 dígitos. La única manera de ingresar dígitos es a través del teclado numérico previamente descripto, clickeando los botones correspondientes. Además, en el campo de la tarjeta, los números deben separarse en grupos de 4 dígitos. Por ejemplo, el número "1111111111111111" debe visualizarse como "1111-1111-1111-1111". El botón Limpiar debe descartar los dígitos ingresados y dejar el campo en blanco.
2. Página de ingreso de PIN: contiene un campo para ingresar el PIN, un teclado numérico, un botón "Aceptar", un botón "Limpiar" y un botón "Salir".
3. Página de operaciones: contiene tres botones: "Balance", "Retiro" y "Salir".
4. Página de balance: contiene información acerca de la tarjeta, es decir, número, fecha de vencimiento, cantidad en la cuenta y dos botones "Atrás" y "Salir".
5. Página de retiro: contiene un campo para indicar la cantidad que se retira, un teclado y los botones "Limpiar", "Aceptar" y "Salir".
6. Página de reporte de operación: contiene la información sobre el número de tarjeta, fecha y hora, cantidad retirada, balance de la cuenta y los botones "Atrás" y "Salir".
7. Página de errores: contiene un mensaje y el botón "Atrás".

Tarea

1. Crear la base de datos y todos los objetos que creas necesarios para que la aplicación funciones.
2. Insertar una cantidad mínima de datos para poder probar la aplicación.
3. Desarrollar una aplicación web, en un repositorio público, utilizando una herramienta de versionado como github/bitbucket/etc, que responda a los requisitos descriptos por el cliente.
4. Entregar un diagrama de entidad relación de las tablas planteadas (DER)

Notas Técnicas

La arquitectura de la aplicación puede ser:

- Web Application
- Base de datos:
    - Relacional (Sql Server o Mysql)

Deseable: Utilizar el patron Repository

El stack tecnológico deberá ser

- Para el frontend (elegir una opción):
  - Asp Net MVC >= 2.2
  - Angular >=8
  - React >=16.13

- Para acceso de base de datos:
  - Entity Framework o Dapper

- Para el backend de ser necesario:
  - Asp Net Core >= 2.2 (deseable swagger)

> No se evaluará el diseño ni el conocimiento sobre UI que posea el developer, sino la funcionalidad y la estabilidad de la solución, el modelo de datos, la atención a los requerimientos, seguridad, estructura del código y la actitud frente a los NFR (non-functional requirements), que son: mantenibilidad, extensibilidad y escalabilidad