Trabajo Práctico 3
Diseño de Sistemas

Cóctel

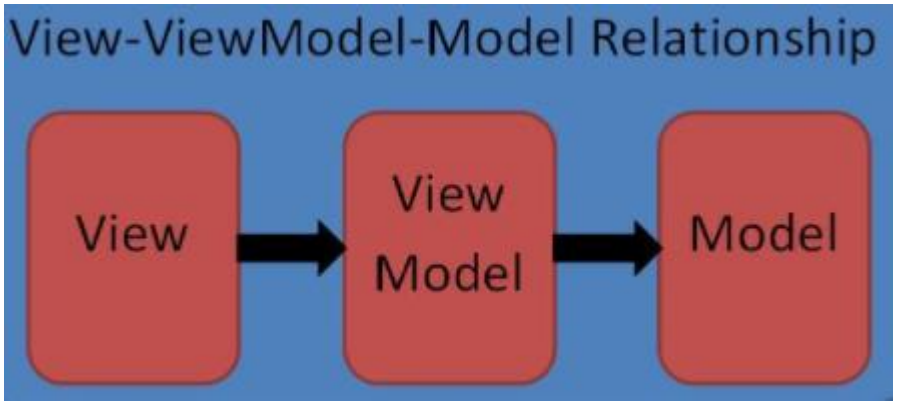
Implementación de Cóctel

Elección de Software

Para implementar el proyecto decidimos utilizar el IDE Microsoft Visual Studio y como base de datos decidimos utilizar Microsoft SQLServer. Porque los software nombrados son de nuestro conocimiento y nos pareció propio el utilizarlos.

Definición de Arquitectura

Decidimos utilizar la arquitectura Model-View-ViewModel, ya que es una arquitectura popular, que deriva de MVC, y nos pareció apropiada.



Las clases contenidas en los "Model", representan el modelo de dominio de la aplicación, dentro del cual se encuentran: El modelo de datos y la lógica empresarial. A la hora de iniciar una aplicación es el primer componente tomado en cuenta, ya que aquí definimos los atributos que estaremos utilizando en nuestra App.
El modelo de vista contiene toda la lógica de presentación. Implementa propiedades y comandos que definen las funcionalidades que tomará nuestra App.
Por último la Vista define cómo la información y las funcionalidades de nuestra App se mostrarán gráficamente.

Documentación Original de Cóctel

Dejamos el Link a la documentación original, para poder comparar la idea inicial con el resultado final del proyecto y poder observar los lineamientos y la definición de componentes principales.
Cóctel

Commit Inicial

Definición de base de datos MSSQL CoctelDB, todos los campos fueron definidos acorde al diccionario de datos establecido en el documento correspondiente.
Creación de la primera **vista**.

Cambios y Modificaciones

A medida que se realizó la implementación de código, surgieron cambios y modificaciones para mejorar el proyecto, cuestiones que a la hora del diseño se obviaron o se pensó que serían necesarias. A continuación se detallan los cambios, modificaciones y arreglos que se realizaron.

Se realizaron modificaciones a la vista realizada en el commit inicial, se revisaron las definiciones de la base de datos y se realizaron las primeras definiciones de métodos y clases en el **modelo**.

Modificación de la base de datos

Se encontraron varios errores en la definición de la base de datos. Se decidió solucionar los mismos en esta etapa temprana. Los siguientes cambios fueron realizados:

- Se **renombraron todas las tablas**, conservaron su nombre aunque fue reemplazado el plural por singular.
    - La tabla **Tipo** fue añadida para conservar la normalización.
    - La tabla **Inventario** fue añadida para relacionar a un usuario con una lista de ingredientes. Ésta cumple la función de la FK Ingredientes_ID de la tabla Usuario.
    - La tabla **Favorito** fue añadida para relacionar a un usuario con una lista de cócteles. Ésta cumple la función de la FK Coctel_ID de la tabla Usuario.
    - La tabla **Contenido** fue añadida para relacionar a un cóctel con una lista de ingredientes. Ésta cumple la función de la FK Ingredientes_ID de la tabla Cóctel.
- Se r**enombraron todos los campos**, conservaron su nombre sin capitalizar la primera letra.
    - El campo **Descripción** de la tabla Ingredientes fue renombrado a **Nombre**. Su tipo no fue modificado.
    - El campo **Tipo** de la tabla cotel ahora se renombró como **Tipo_ID**
- Se **modificó el tipo de dato** de todas las claves al tipo **integer** para utilizar las funciones de autoincremento.

~~Usuarios~~ → Usuario			
Campo	Tipo de dato antiguo		Tipo de dato nuevo
usuario_ID	varchar(20)	→	int

~~Ingredientes~~ → Ingrediente

Campo	Tipo de dato antiguo		Tipo de dato nuevo
ingredientes_ID	varchar(50)	→	int

~~Cócteles~~ → Cóctel			
Campo	Tipo de dato antiguo		Tipo de dato nuevo
coctel_ID	varchar(20)	→	int

Cambios en el DER

//añadir nuevo DER

Adición al Diccionario de Datos

Inventario			
Campo	Tipo de Dato	Clave Tipo	Descripción
Usuario_ID	varchar(20)	PFK	Id alfanumérico único que identifica al usuario

Ingredientes_ID	varchar(50)	PFK	Id único que identifica al ingrediente (Nombre del mismo)

Contenido			
Campo	Tipo de Dato	Clave Tipo	Descripción
Ingredientes_ID	varchar(50)	PFK	Id único que identifica al ingrediente (Nombre del mismo)
Coctel_ID	varchar(20)	PFK	Id alfanumérico único que identifica al cóctel

Favorito			
Campo	Tipo de Dato	Clave Tipo	Descripción
Usuario_ID	varchar(20)	PFK	Id alfanumérico único que identifica al usuario
Coctel_ID	varchar(20)	PFK	Id alfanumérico único que identifica al cóctel

Tipo			
Campo	Tipo de Dato	Clave Tipo	Descripción
Tipo_ID	varchar(100)	-	Si contiene o no alcohol y grupo al que pertenece (Batido, directo, etc)
Nombre	varchar(50)	-	Nombre del Cóctel

Vista

Se rediseñó la grilla para contener 4 secciones:

- Sección dedicada al mensaje de bienvenida
- Sección dedicada al login
- Sección dedicada a la búsqueda
- Sección dedicada a la lista de cócteles
- Sección dedicada a la documentación

Se añadieron los campos de búsqueda y los botones para realizar la misma.
No se implementó ninguna funcionalidad

Definición de clases

Se definieron en el modelo las siguientes clases:

- Cóctel
- Usuario
- Ingrediente

Los campos se corresponden a los definidos en la base de datos.

Se definieron los siguientes métodos para la clase Usuario:

- NuevoFavorito: recibe un Cóctel como parámetro y lo añade a la lista de Cócteles favoritos del Usuario.
- NuevoIngrediente: recibe un Ingrediente y lo añade al inventario del Usuario. El sistema no hace distinción para la cantidad de ingredientes en el inventario.
- EliminarFavorito: recibe un Cóctel como parámetro y lo elimina de la lista de Cócteles favoritos del Usuario.
- EliminarIngrediente: recibe un Ingrediente y lo elimina del inventario del Usuario.

Implementación de funcionalidad T-SQL

Dentro del ViewModel se definió la clase **DatabaseVM**, ésta posee su propio namespace y está contenida dentro del directorio Helpers.
El objetivo de esta clase es contener todos los métodos necesarios para realizar la manipulación de datos de la base de datos, así como definir los parámetros de conexión con la misma.
En este commit se añade el package Microsoft.Data.SqlClient, con el cual se establece la conexión y se realizan las operaciones antes mencionadas.

Métodos contenidos en DatabaseVM

Read(), Read(query) y Read(cóctel)

Esta familia de métodos se encarga de realizar las búsquedas y devuelve **listas** como resultado.

- **Read()**: Al llamarse al método sin parámetros se recibe como resultado una lista del tipo cóctel (List<Cocktail>). Esta lista estará conformada por todos los cócteles de la base de datos, ordenados por el campo porcentaje_Recomendacion.
  Esta búsqueda utiliza las tablas Cóctel y Tipo.
- **Read(cóctel)**: Al llamarse con un método de tipo cóctel (Cocktail) se recibe como resultado una lista del tipo ingrediente (List<Ingredient>). Esta lista estará conformada por todos los ingredientes del cóctel utilizado como parámetro.
  Esta búsqueda utiliza las tablas Contenido e Ingrediente.
- **Read(query)**: Al llamarse con un parámetro del tipo string se recibe como resultado una lista del tipo cóctel (List<Cocktail>). Esta lista estará conformada por todos los cócteles que contengan en su campo *nombre* el parámetro ingresado, ordenados por el campo nombre.
  Esta búsqueda utiliza las tablas Cóctel y Tipo.

Insert(cóctel) e Insert(ingrediente)

Esta familia de métodos se encarga de