Aplicació de gestió de clients.

Biel Aymerich

Xavier sane

INDEX

[1. Introducció 2](#_Toc179993161)

[Objectiu del projecte: 2](#_Toc179993162)

[Tecnologies utilitzades: 2](#_Toc179993163)

[2.Arquitectura del Projecte 2](#_Toc179993164)

[Patró MVVM: 2](#_Toc179993165)

[Model: 2](#_Toc179993166)

[View: 2](#_Toc179993167)

[ViewModel: 3](#_Toc179993168)

[3.Detalls d'Implementació 3](#_Toc179993169)

[Interfície gràfica (XAML): 3](#_Toc179993170)

[4. Problemes i Solucions 6](#_Toc179993171)

[5. Conclusions 7](#_Toc179993172)

# 1. Introducció

## Objectiu del projecte:

L’objectiu es aprendre a menjar el patró MVVM en aquest cas ja teníem una base preexistent i nosaltres hem afegit lògica i visualitzacions per completar l’exercici.

## Tecnologies utilitzades:

Hem treballat amb el programa Visual Studio, fent servir llenguatges com: C#, Xaml

# 2.Arquitectura del Projecte

## Patró MVVM:

És una arquitectura de disseny de software que es fa servir principalment en UI, la seva funcionalitat ens permet fer una separació de responsabilitats fent que el codi sigui modular, fàcil de mantenir i testejable.

### Model:

En el model nosaltres tenim l’arxiu client.cs que conté les dades i la lògica per comunicar-se amb la resta del programa.

### View:

* **clientsview.xaml:** Visualització dels clients.
* **custommessagebox.xaml:** Ens crea un pop up personalitzat per mostrar errors o informació.
* **editarusuari.xaml:** Pantalla per editar usuaris.
* **graphicsview.xaml:** Pantalla que inclou les gràfiques.
* **iniciview.xaml:** Pantalla d'inici.

### ViewModel:

* + - **clientviewmodel.cs:**

S’encarrega de processar la lògica per gestionar les dades dels clients.

* + - **iniciviewmodel.cs:**

S’encarrega de processar la lògica que controla els temes clar i fosc.

* + - **mainviewmodel.cs:**

S’encarrega de processar la lògica que fa el canvi entre les vistes de l’aplicació de manera dinàmica.

* + - **relaycommand.cs:**

S’encarrega de processar la lògica que ens permet fer els enllaços amb la UI i així poder executar les accions.

# 3.Detalls d'Implementació

## Interfície gràfica (XAML):

En primer lloc definim el clientsView.xaml, dissenyada per mostrar una taula de clients utilitzant un UserControl, un Grid amb dos files, DataGrid, StackPanel i botons.

Tot això permet al usuari realitzar accions com afegir, editar, eliminar o veure el gràfics mitjançant botons, tot està connectat mitjançant Bindings amb el ViewModel

Captura de pantalla de un celular

Descripción generada automáticamenteInterfaz de usuario gráfica, Tabla

Descripción generada automáticamente

En segon lloc definim CustomMessageBox.xaml hem utilitzat un UserControl, Grid, StackPanel i Botons per crear un pop up per mostrar errors i informació als usuaris de l’aplicació on l’usuari pot donar un boto OK, tot està connectat mitjançant Bindings amb el ViewModel.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

En tercer lloc definim EditarUsuari.xaml, dissenyada per permetre al usuari de l’aplicació editar les dades del client mitjançant un formulari, hem utilitzar UserControl, Grid, StackPanel, Botons i events Text\_Changed i Checked, tot està connectat mitjançant Bindings amb el ViewModel

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente con confianza media

En quart lloc definim GraphicsView.xaml, dissenyada per mostrar gràfics interactius utilitzant la llibreria LiveChartWpf que ens ajuda a representar el rendiment anual de cada client, hem utilitzat UserControl, Grid de dos files, Grafics, botons i tot està connectat mitjançant Bindings amb el ViewModel.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

En cinquè lloc definim IniciView.xaml, es la pantalla d’inici que en el nostre projecte ens serveix per triar el tema de l’aplicació si el volem fosc o clar, hem utilitzat Grid, Label, StackPanel i Botons, tot està connectat mitjançant Bindings amb el ViewModel.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

* **Gràfiques:** Tenim dos tipus de gràfiques, línies i barres.  
  El procés que segueix l’aplicació és:
  + Seleccionem l’usuari que volem mostrar la gràfica
  + El ShowResultsCommand carga els resultats del client al ViewModel
  + Canviem la vista a GraphicsView y el gràfic mostra els resultats anuals del client

# 4. Problemes i Solucions

* **Deshabilitar el botó de save en funció de que tots els camps necessaris hagin set omplerts:** Al no dependre d’una sola propietat sinó d’un grup no es podia fer un “Binding” directe amb la qual cosa vàrem provar de implementar una classe converter però no funcionava i al final vàrem fer un “bool” el qual implementava un mètode que retornava fals en funció de si trobava null en un array de strings.
* **Implementació de DarkTheme:** Hem tingut problemes per entendre com funcionava tot però un cop hem entès com aplicar els style mitjançant Keys en cas de ser elements concrets i deixant-los oberts en cas de ser globals i fent servir el Dynamic Resource per a canviar elements en la pantalla i Static Resource per a les que iniciaran després del procés.
* **Aconseguir subratllar les rows d’un DataGrid amb l’implementació d’un trigger “IsSelected”:** Vàrem provar de vàries maneres diferents però no era possible des del objecte DataGridRow ho hem fet aplicant un style sobre un objecte DataGridCells.
* **Implementació de persistència de dades: vàrem implementar la persistència de dades a una DB SQL mitjançant Entity Framework Core:** No vàrem tenir prou temps per implementar totes les classes i finalitzar l’ implementació amb la qual cosa ho vam acabar descartant per a poder entregar a temps.
* **Implementació de cursor personalitzat**: Hem provat d’implementar el nostre propi cursor ja que en el mode fosc al tenir el color de les TextBoxes el ibean que vé per default no es veia visible però no hem tingut temps de finalitzar la implementació.

# 5. Conclusions

Com a primer contacte amb el MVVM, aquest projecte ens ha proporcionat un bon primer contacte, al treballar amb la separació clara entre Model, Vista ViewModel el principi sembla molt complex, ja que estem acostumats a ficar la lògica directament dins de la UI.  
Anar avançant en el projecte veiem que la separació que proporciona el MVVM facilita l’organització i la comprensió del codi.  
El DataBinding es una eina que hem anat veient al llarg del projecte i que ens mostra com les dades es reflecteixen automàticament a la UI, una eina que hem trobat molt resolutiva.

La reutilització del codi i la flexibilitat que ens proporciona el patró ens seran molt útils per futurs projectes.