# Introdução

Este relatório tem com objectivo a clarificação dos detalhes de implementação do *Sistema de Gestão de Projectos de Financiamento* (SGPF).

O sistema foi construído através de uma aplicação Windows sobre a tecnologia *Windows Presentation Foundation* (WPF).

Foi utilizado o padrão arquitectural *Model View ViewModel* (MVVM) suportado pela tecnologia de forma a segmentar os diferentes componentes da aplicação, estes componentes estão ilustrados na Ilustração 1.



Ilustração 1 - Arquitectura da solução.

Os mecanismos existentes em *WPF* foram a principal razão para o uso do padrão *MVVM*, nomeadamente:

* *Data binding* – este mecanismo permite que a *framework* esteja sempre à espera de alterações por parte dos dados actualizando assim a *User Interface* (UI) sem a necessidade de nenhum procedimento “manual”.
* *Commands* – ao contrário de outras tecnologias não é necessário o registo num evento de acção para realizar uma operação, os componentes *WPF* permitem registar *Commands* (operações) a eventos de forma a diminuir a dependência entre *ViewModels* e *Views*.

# Estrutura do Projecto

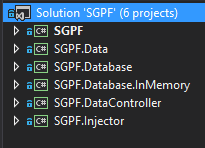


Ilustração 2 - Estrutura do projecto.

A Ilustração 2 apresenta os 6 módulos da aplicação:

1. *SGPF* – A aplicação gráfica, contém as *Views* e *ViewModels*.
2. *SGPF.Data* – O modelo da aplicação.
3. *SGPF.Database* – Os contractos para os serviços de base de dados.
4. *SGPF.Database.InMemory* – A implementação do acesso a dados em memória
5. *SGPF.DataController* – Os contractos e implementação dos controladores.
6. *SGPF.Injector* – O projecto que configura o injector de dependências da aplicação.

# Views

A aplicação tem apenas uma única View, todos os outros elementos gráficos da aplicação funcionam à custa de popups que são:

* LoginPopup – um simples controlo para fazer autenticação.
* ProjectInfo – Contém todas as informações e operações de um projecto.

A view única da app a MainWindow contém tabs para cada tipo de funcionário e as opções de pesquisa e criação de projecto.

# Modelo de domínio

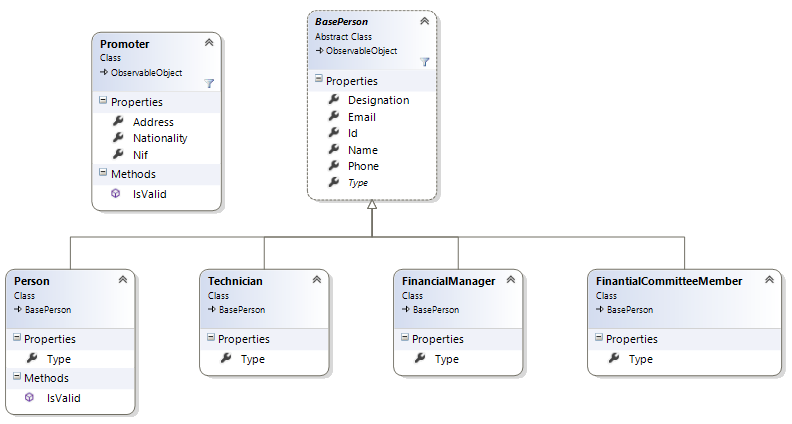


Ilustração 3 - Diagrama de classes de actores e participantes no sistema.

Na Ilustração 3 está presente os actores do sistema, decidiu-se que os funcionários e os responsáveis iriam partilhar a mesma classe base de forma a simplificar a sua utilização na aplicação. De salientar que a classe *BasePerson* implementa indirectamente a interface *INotifyPropertyChanged* através da classe *ObservableObject*. A *framework* WPF utiliza esta interface para realizar os data bindings.

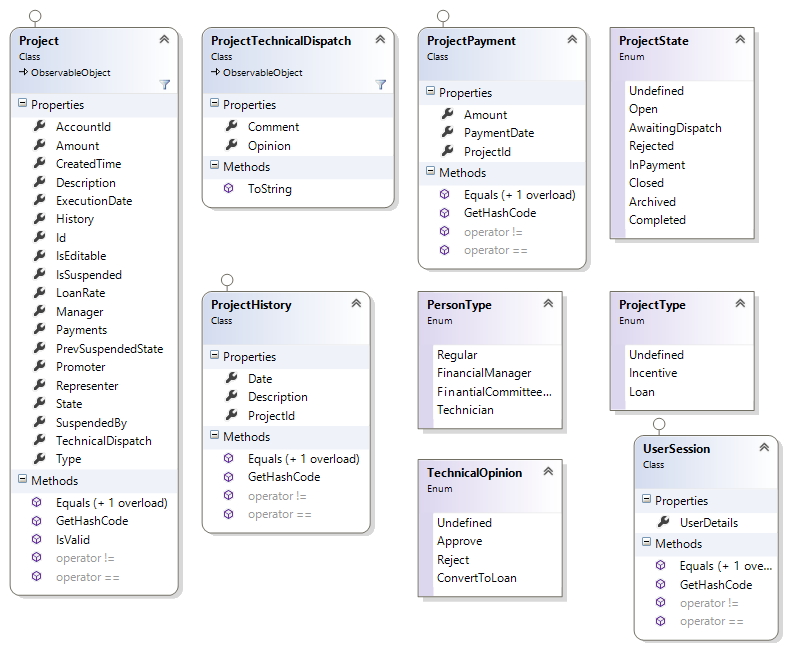


Ilustração 4 – Restantes classes do modelo.

Na Ilustração 4 é possível visualizar todos os tipos criados de forma a suportar os diferentes estados e operações dos projectos.

# ViewModels

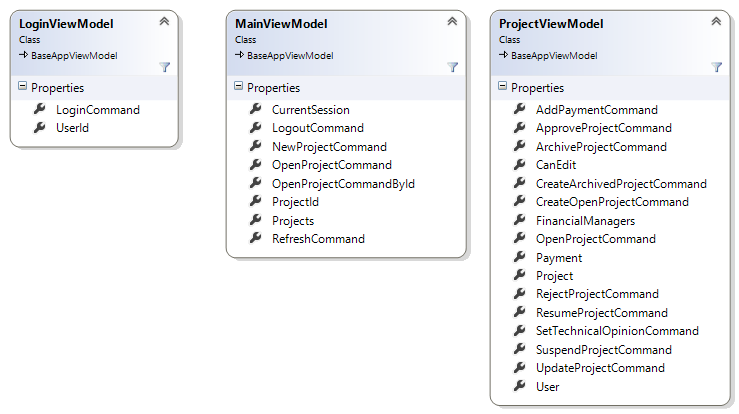
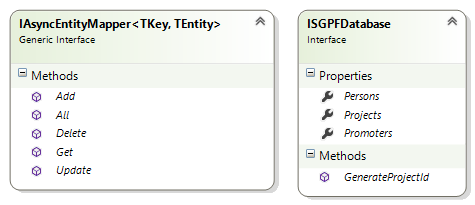


Ilustração 5 - Diagrama de classes dos ViewModels

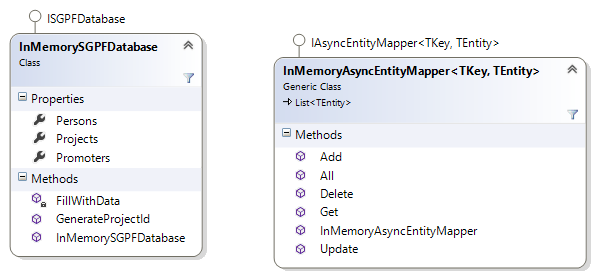
# Model

## Acesso a dados

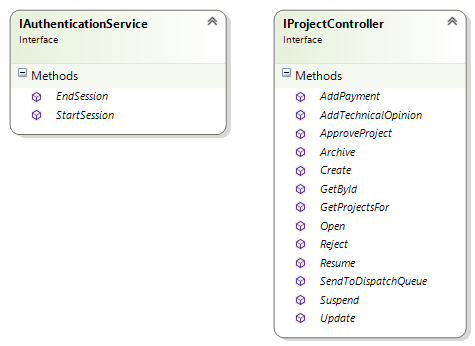
### Contracto



### Implementação



## Controllers



# Conclusão