

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA



ISEL | DEETC

**DEPARTAMENTO DE ENGENHARIA DE
ELECTRÓNICA E TELECOMUNICAÇÕES E DE COMPUTADORES**

Engenharia de Software

RELATÓRIO TRABALHO PRÁTICO Nº 2 SISTEMA DE GESTÃO DE PROJECTOS DE FINANCIAMENTO

SEMESTRE DE INVERNO 2013/2014

GRUPO 3

ALUNOS:

31656, Ricardo Nunes

32466, Diogo Cardoso

ÍNDICE

1	INTRODUÇÃO	1
2	ESTRUTURA DO PROJECTO	2
3	VIEWS	3
4	MODELO DE DOMÍNIO	4
5	VIEWMODELS	6
6	MODEL	7
6.1	ACESSO A DADOS	7
6.1.1	<i>Contracto</i>	<i>7</i>
6.1.2	<i>Implementação.....</i>	<i>7</i>
6.2	CONTROLLERS.....	7
7	INTERFACE GRÁFICA.....	9
7.1	LOGIN.....	9
7.2	TÉCNICO DE FINANCIAMENTO	9
7.3	MEMBRO DA COMISSÃO DE FINANCIAMENTO	10
7.4	GESTOR DE FINANCIAMENTO.....	11
8	CONCLUSÃO	12

1 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 (ilustrados na Ilustração 1).

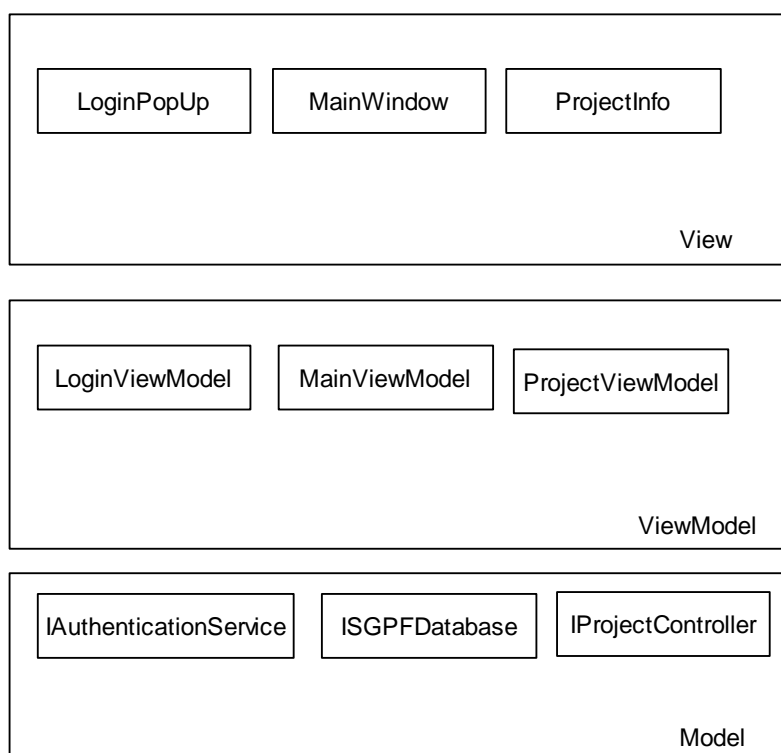


Ilustração 1 - Arquitectura da solução.

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*.

2 Estrutura do Projecto

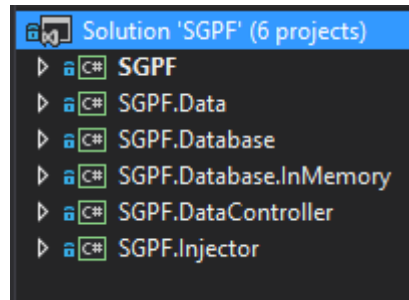


Ilustração 2 - Estrutura do projecto.

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.

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

4 Modelo de domínio

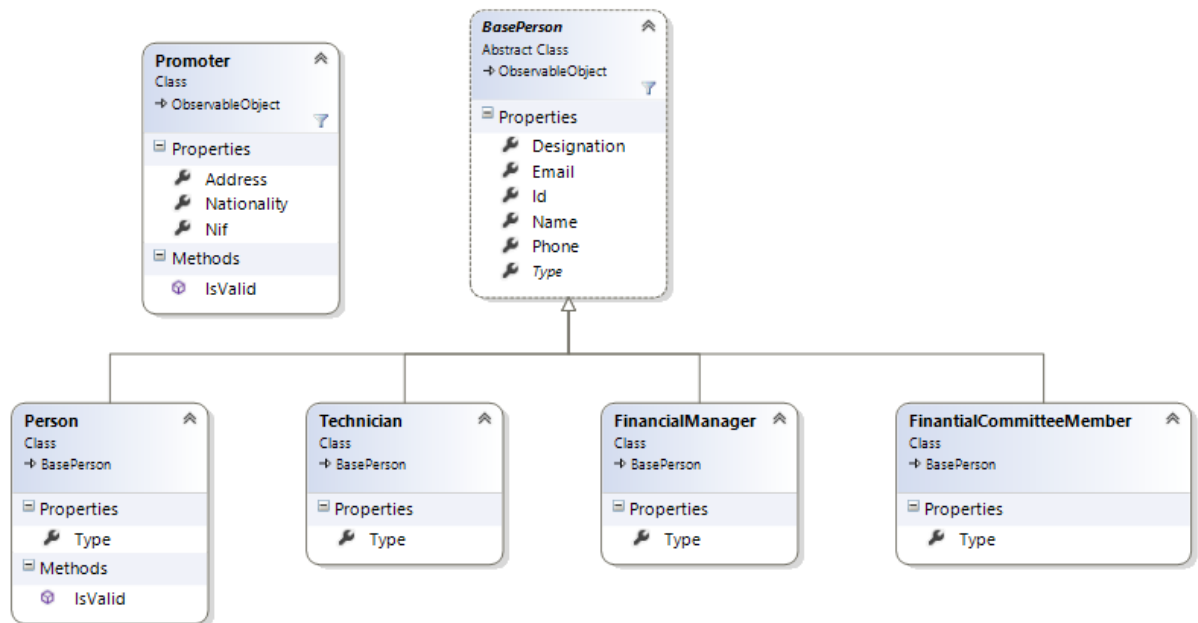


Ilustração 3 - Diagrama de classes de actores e participantes no sistema.

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*.

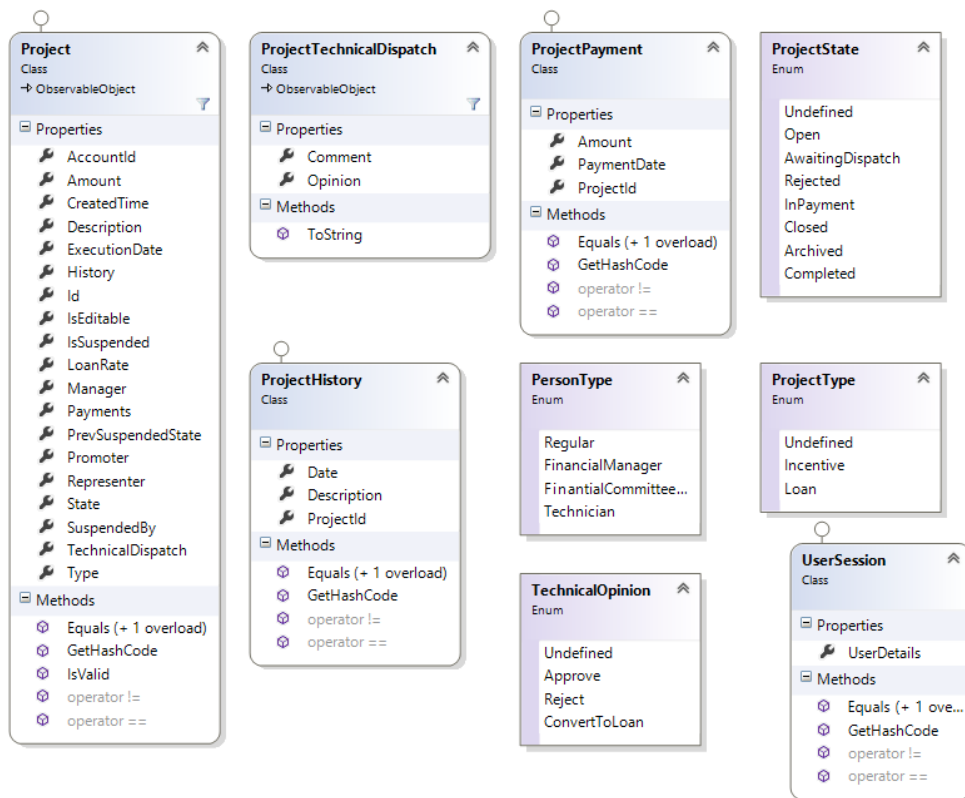


Ilustração 4 – Restantes classes do modelo.

Na Ilustração 4 é possível visualizar todos os tipos criados de forma a suportar os diferentes estados e operações dos projectos. Um projecto (*Project*) é descrito pelas várias pessoas envolvidas (e.g. *Promoter* para promotor, *Manager* para gestor de financiamento), o seu histórico (*History*) – uma lista de alterações feitas ao projecto, com autor e data –, o histórico de pagamentos (*Payments*) o despacho técnico emitido pelo gestor de financiamento e por dados de pagamento (e.g. total a pagar, tipo de financiamento).

5 ViewModels

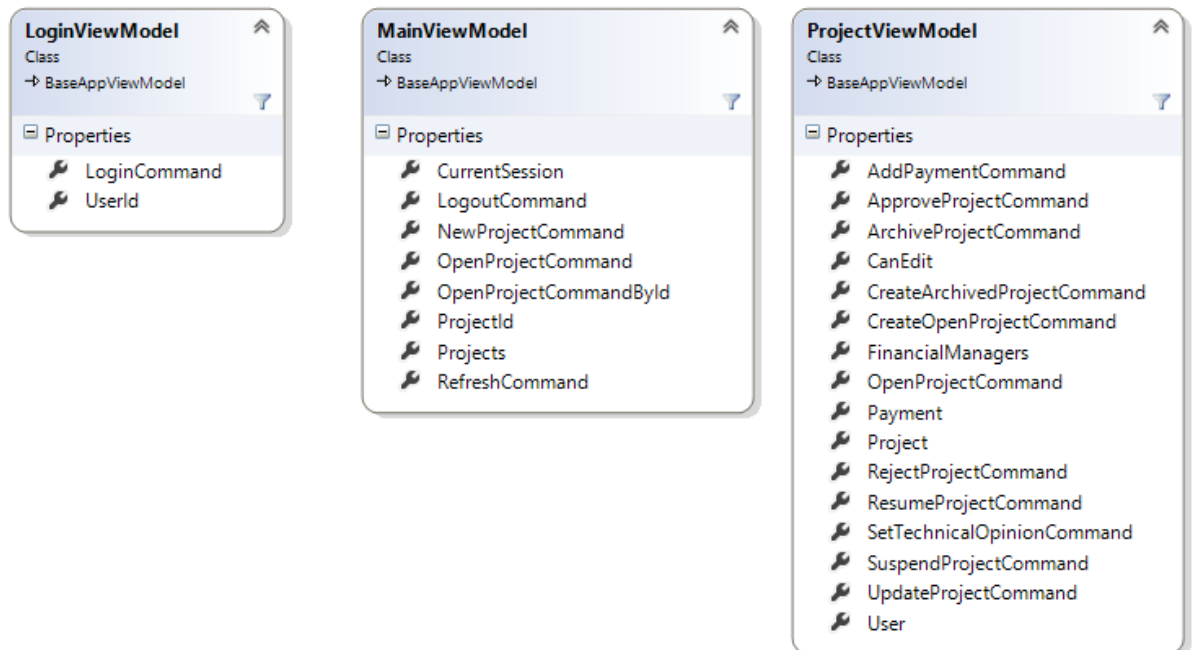


Ilustração 5 - Diagrama de classes dos ViewModels

A Ilustração 1 contém os campos dos ViewModels, estes são utilizados pela interface gráfica de forma a efectuar operações conforme as acções do utilizador. Os ViewModels contêm dois tipos de propriedades, Commands e dados. Os dados contêm a informação que a interface gráfica irá mostrar enquanto que os comandos contêm as acções que o utilizador poderá realizar.

6 Model

6.1 Acesso a dados

6.1.1 Contracto

Como se pode ver na Ilustração 6, o contracto da camada de acesso a dados contém 3 *IAsyncEntityMapper* (entidade com métodos CRUD para gestão dos vários tuplos), um para cada entidade a guardar (*Persons*, *Projects* e *Promoters*) e um método para geração de um ID único para um novo projecto.

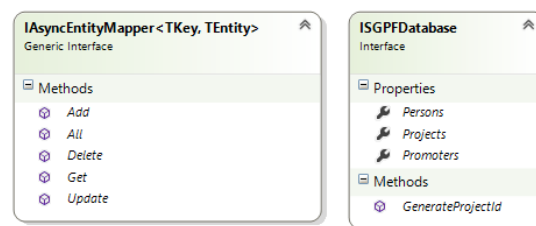


Ilustração 6 – Contracto da camada de acesso a dados

6.1.2 Implementação

Por motivos de simplificação decidimos implementar um repositório em RAM preenchido na inicialização com alguns dados para acelerar os testes.

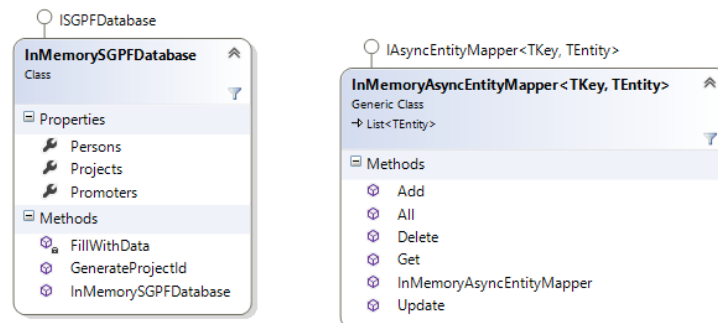


Ilustração 7 – Esquema da implementação da camada de acesso a dados

6.2 Controllers

O projecto tem um controlador de autenticação para guardar o utilizador actual (ver contracto na Ilustração 8, à esquerda) e outro para gestão de projectos (ver contracto na Ilustração 8, à direita).

Os métodos do controlador de projectos permitem-nos criar, procurar projectos usandoo seu identificador, alterar o estado do projecto (e.g. *Archive*, *Reject*, *Open*, *Suspend*, *Resume*, *ApproveProject*), enviar o projecto para a comissão de financiamento, adicionar uma opinião técnica e obter uma lista de projectos associada a cada pessoa.

O primeiro parâmetro de todos métodos é a pessoa que faz o pedido e esse parâmetro é usado para efeitos de *logging* e/ou para validar se as acções pedidas podem ser feitas naquele momento, por aquela pessoa.

Todos os métodos têm uma fase de validação, onde, entre outras validações, verifica-se se o estado actual do projecto suporta a acção pedida (e.g. um projecto suspenso não pode ser arquivado) e se a pessoa em questão têm autorização para a fazer (e.g. um gestor que não esteja associado ao projecto não pode dar o seu aval). Caso a operação não seja válida o controlador retorna uma excepção, que resulta numa *message box* com a descrição do erro (e.g. “*Project 2 it’s suspended*”).

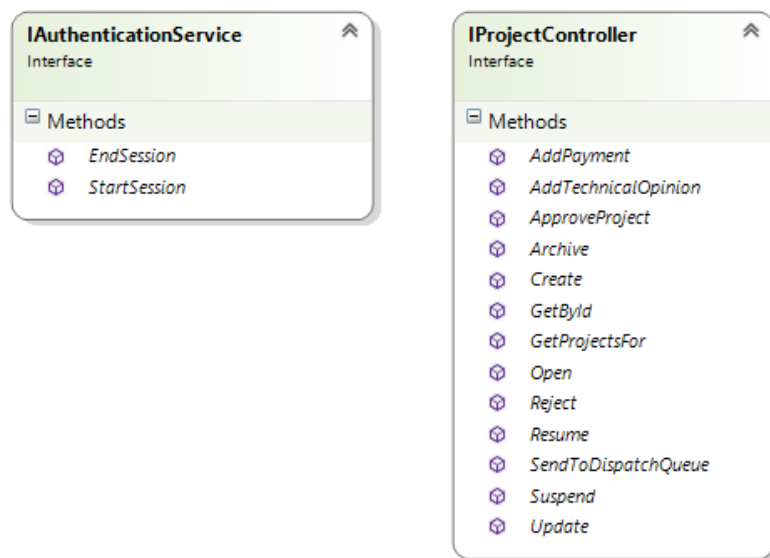


Ilustração 8 – Contracto do serviço de autenticação (à esquerda) e do controlador de projectos (à direita)

7 Interface Gráfica

7.1 Login

No arranque da aplicação temos a janela de *login* (Ilustração 9), onde apenas é necessário introduzir um identificador de utilizador válido para iniciar sessão. Para cada tipo de utilizador (técnico, gestor ou membro da comissão) existe uma interface diferente caso o utilizador autentique-se com sucesso.

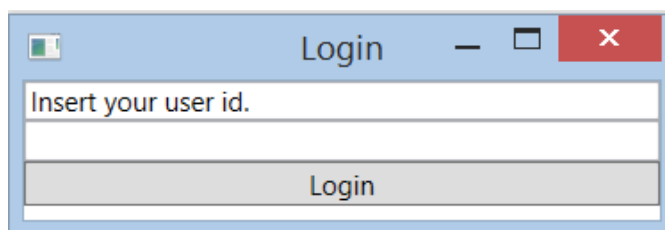


Ilustração 9 – Janela de *login*

7.2 Técnico de financiamento

A interface do técnico de financiamento (Ilustração 10) permite-lhe criar novos projectos (Ilustração 11) e abrir projectos já criados.

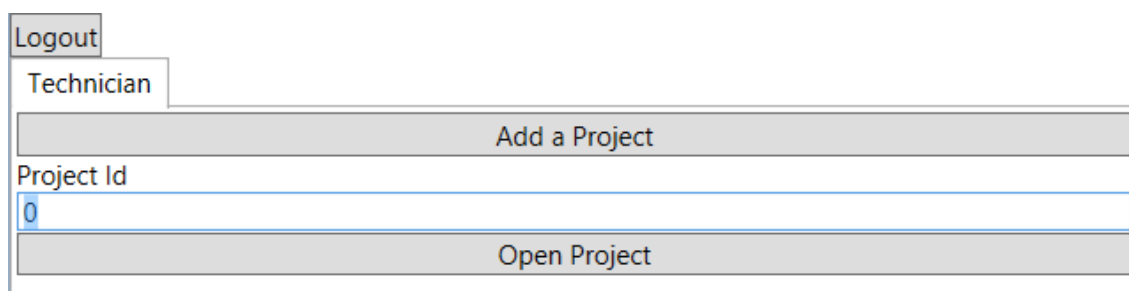


Ilustração 10 – Interface para técnicos de financiamento

Representer

Designation

Email

Id

Name

Phone

Create opened project

Create archived project

Ilustração 11 – Interface de criação de novos projectos

7.3 Membro da comissão de financiamento

O ecrã inicial para o membro da comissão de financiamento (Ilustração 12) mostra a lista de projectos que aguardam alguma acção por parte da comissão (e.g. emitir despacho, reactivação).

Ao escolher um dos projectos é possível realizar as várias acções necessárias numa nova janela (Ilustração 13), onde, entre outras acções, é possível alterar dados de projecto, atribuir-lhe um gestor, ver o histórico do projecto e dos pagamentos do mesmo e até suspender, arquivar ou aprovar o projecto.

Logout

Finantial Committee Member

0 AwaitingDispatch

1 Archived

2 Open

Ilustração 12 – Ecrã inicial do membro da comissão de financiamento

Payments
History
Technical Report
Opinion
Comments
Change Project Type
Undefined
Incentive
Loan
Approve Reject

Ilustração 13 – Ecrã (parcial) de projecto para membro da comissão de financiamento

7.4 Gestor de financiamento

O ecrã inicial do gestor de financiamento é igual ao do membro da comissão de financiamento (Ilustração 12), sendo que a única diferença está no ecrã de acções por projecto (Ilustração 14), onde é possível emitir despachos técnicos, alterar dados do projecto, arquivar ou suspender (e o processo inverso) e efetuar pagamentos.

Loan Rate	
0.6	
Technical Dispatch	
Opinion	Comments
Undefined	
Approve	
Reject	
ConvertToLoan	
Add Technical Opinion	

Ilustração 14 - Interface (parcial) para gestores de financiamento

8 Conclusão

Com este trabalho percebemos a importância do trabalho realizado na primeira fase, passar de toda a análise de requisitos para código foi muito simples, comparado com uma implementação semelhante com pobre documentação (ou análise de requisitos muito vaga).