



Society Agendor

Aplicativo para agendar campo de futebol society.

Aluno: Vinícius de Souza Avansini

RA: 0040481512038

Orientador: Prof. Dr. Kleber de Oliveira Andrade

Agenda

1. Problema
2. Objetivo
3. Metodologia de Desenvolvimento;
4. Requisitos Funcionais;
5. Requisitos Não Funcionais;
6. Planejamento;
7. Recursos e Ferramentas;
8. Modelagem:
 - 8.1. MER;
 - 8.2. Diagrama de Casos de Uso;
 - 8.3. Diagrama de Classes;
9. Testes;
10. Demonstração do Aplicativo;
11. Considerações Finais.

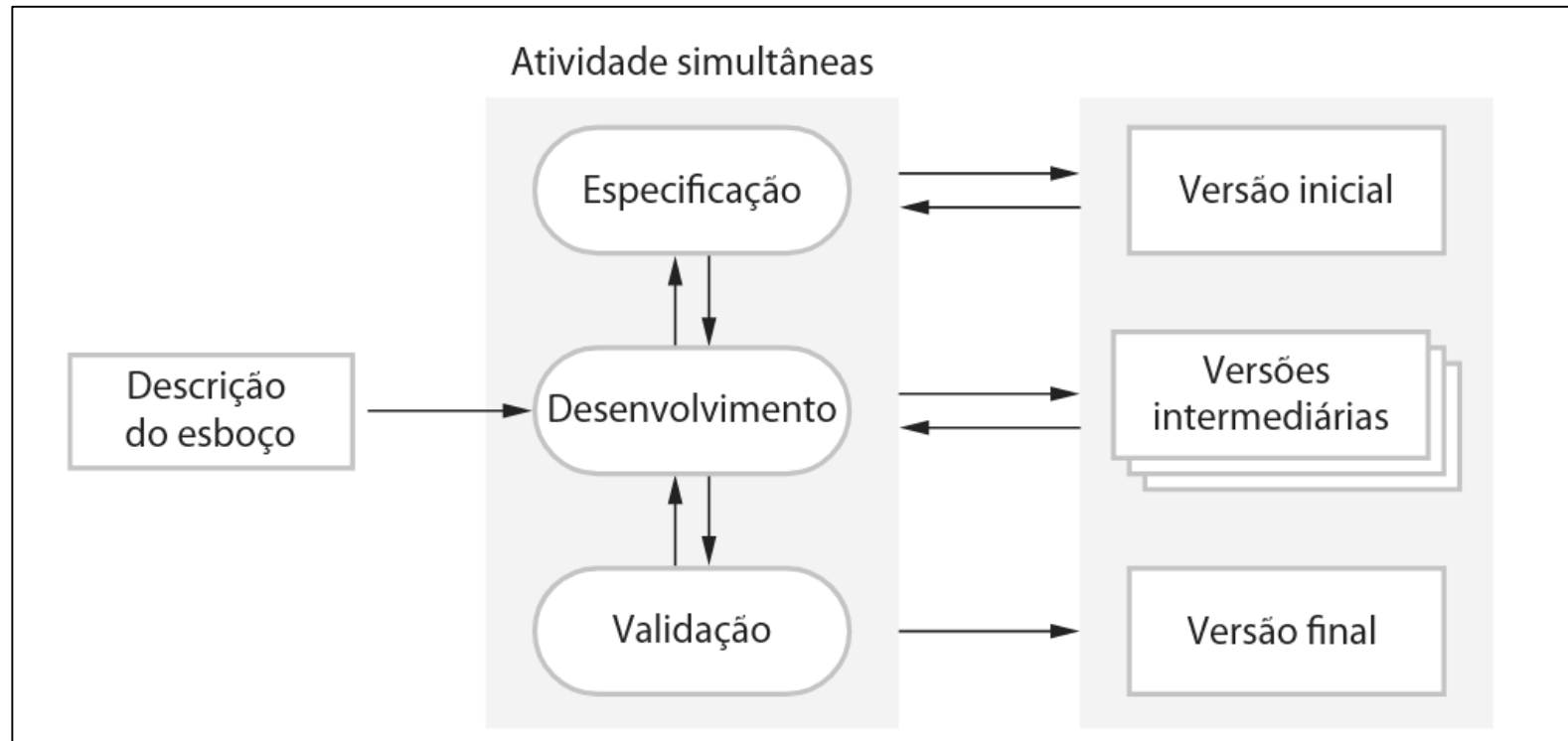
Problema

- Quais foram as razões para o desenvolvimento do aplicativo?

Objetivo

- Quais são os objetivos do aplicativo?

Metodologia de Desenvolvimento



Fonte: SOMMERVILLE, 2011, p. 22.

Engenharia de Requisitos - Requisitos Funcionais

Identificação	Requisito Funcional	Categoria	Prioridade
RF001	O sistema deve possuir um botão que possibilite o login com o Facebook, possibilitando o resgate de informações do usuário e facilite seu cadastro.	Design e Programação	Média
RF002	O sistema deve exibir apenas os agendamentos do usuário "logado".	Programação	Muito alta
RF003	Para se utilizar o sistema, o usuário deve entrar com sua conta do Facebook.	Programação	Alta
RF004	O sistema automaticamente deve excluir, do banco de dados, os agendamentos que já ocorreram.	Programação	Muito alta
RF005	O sistema deve apontar os estabelecimentos através do Google Maps.	Design e Programação	Alta
RF006	O sistema deve emitir uma mensagem de alerta caso o dispositivo não possua conexão com a internet.	Programação	Média
RF007	Apenas o usuário que agendou o local pode editar ou cancelar o agendamento.	Programação	Muito alta
RF008	O sistema deve emitir uma notificação uma hora antes do evento acontecer.	Programação	Média

Engenharia de Requisitos – Requisitos Não Funcionais

Identificação	Requisito Não Funcional	Categoria	Prioridade
RNF001	O aplicativo deve executar em celulares a partir da API 19 do Android.	Programação	Muito alta
RNF002	O aplicativo não irá funcionar sem conexão com a internet.	Programação	Muito alta
RNF003	Se o usuário deixar algum campo de digitação em branco, o sistema deve referenciá-lo com uma cor avermelhada.	Design e Programação	Média
RNF004	Minimizar a quantidade de campos de entrada de texto, tornando menos cansativo para o usuário.	Design	Alta
RNF005	Todos os métodos que possuem entrada de dados, devem possuir validação para saber se os campos estão nulos ou com espaço em branco.	Programação	Média
RNF006	Exibir mensagens de alerta caso algum campo de texto esteja vazio.	Programação	Baixa

Fonte: Elaborado pelo autor.

Planejamento

		Modo da Tarefa ▼	Nome da tarefa ▼	% concluída ▼	Início ▼	Término ▼	Predecessoras ▼
1	✓		▸ Início do Projeto	100%	Qua 05/07/17	Seg 06/11/17	
2	✓		▸ Planejamento e Desenvolvimento	100%	Qua 05/07/17	Sáb 04/11/17	
3	✓		Mock-up das telas do aplicativo (versão 1)	100%	Qua 05/07/17	Qui 06/07/17	
4	✓		Desenvolvimento Tela de Login (versão 1)	100%	Qui 06/07/17	Seg 10/07/17	
5	✓		Desenvolvimento Tela Inicial (versão 1)	100%	Qui 13/07/17	Qui 13/07/17	4
6	✓		Desenvolvimento Tela de Perfil (versão 1)	100%	Sex 14/07/17	Sex 14/07/17	5
7	✓		Desenvolvimento Tela de Splash Screen (versão 1)	100%	Qua 02/08/17	Sex 04/08/17	
8	✓		Mock-up das telas do aplicativo (versão 2)	100%	Sex 11/08/17	Sex 11/08/17	22;23
9	✓		Desenvolvimento Tela de Login (versão 2)	100%	Seg 04/09/17	Qua 06/09/17	
10	✓		Desenvolvimento Cadastrar usuário	100%	Qui 07/09/17	Sáb 16/09/17	9
11	✓		Desenvolvimento Login usuário	100%	Dom 17/09/17	Ter 19/09/17	10
12	✓		Desenvolvimento Visualizar perfil de usuário	100%	Qua 20/09/17	Qui 21/09/17	11
13	✓		Desenvolvimento Atualizar perfil de usuário	100%	Sex 22/09/17	Seg 25/09/17	12
14	✓		Desenvolvimento de API	100%	Ter 03/10/17	Qua 04/10/17	
15	✓		Desenvolvimento de Cadastrar agendamento	100%	Qui 05/10/17	Qua 18/10/17	14
16	✓		Desenvolvimento Listar Agendamentos	100%	Qui 19/10/17	Sex 20/10/17	15
17	✓		Desenvolvimento Tela de Edição de agendamento	100%	Sáb 21/10/17	Dom 22/10/17	16
18	✓		Desenvolvimento Atualizar agendamento	100%	Seg 23/10/17	Qua 25/10/17	17
19	✓		Desenvolvimento Apagar agendamento	100%	Qui 26/10/17	Seg 30/10/17	18
20	✓		Desenvolvimento Notificação de jogo	100%	Qua 01/11/17	Sáb 04/11/17	
21	✓		▸ Levantamento de Requisitos	100%	Seg 07/08/17	Qui 10/08/17	
22	✓		Requisito Funcional	100%	Seg 07/08/17	Ter 08/08/17	
23	✓		Requisito Não Funcional	100%	Qua 09/08/17	Qui 10/08/17	22
24	✓		▸ Documentação	100%	Qua 25/10/17	Seg 06/11/17	
25	✓		Diagrama de Casos de Uso	100%	Qua 25/10/17	Qui 26/10/17	
26	✓		Diagrama de Classes	100%	Sex 03/11/17	Sex 03/11/17	
27	✓		Cronograma do Projeto	100%	Seg 06/11/17	Seg 06/11/17	

Fonte: Elaborado pelo autor.

Recursos e Ferramentas



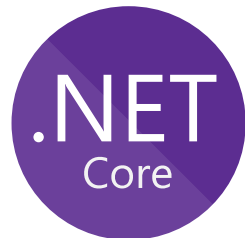
Xamarin



IDE do Visual Studio 2017



Visual Studio Team Services



ASP.NET Core Web API

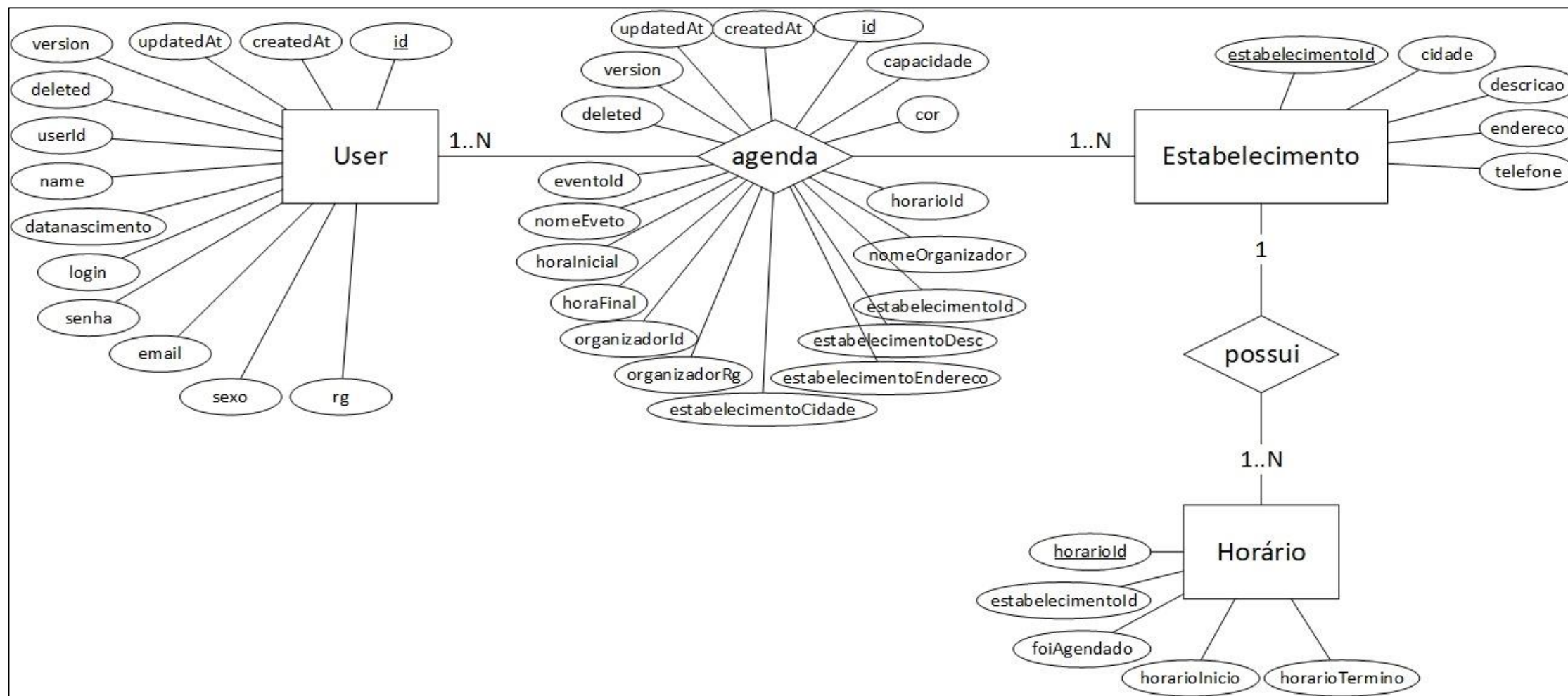


Google Maps API



Facebook Graph API

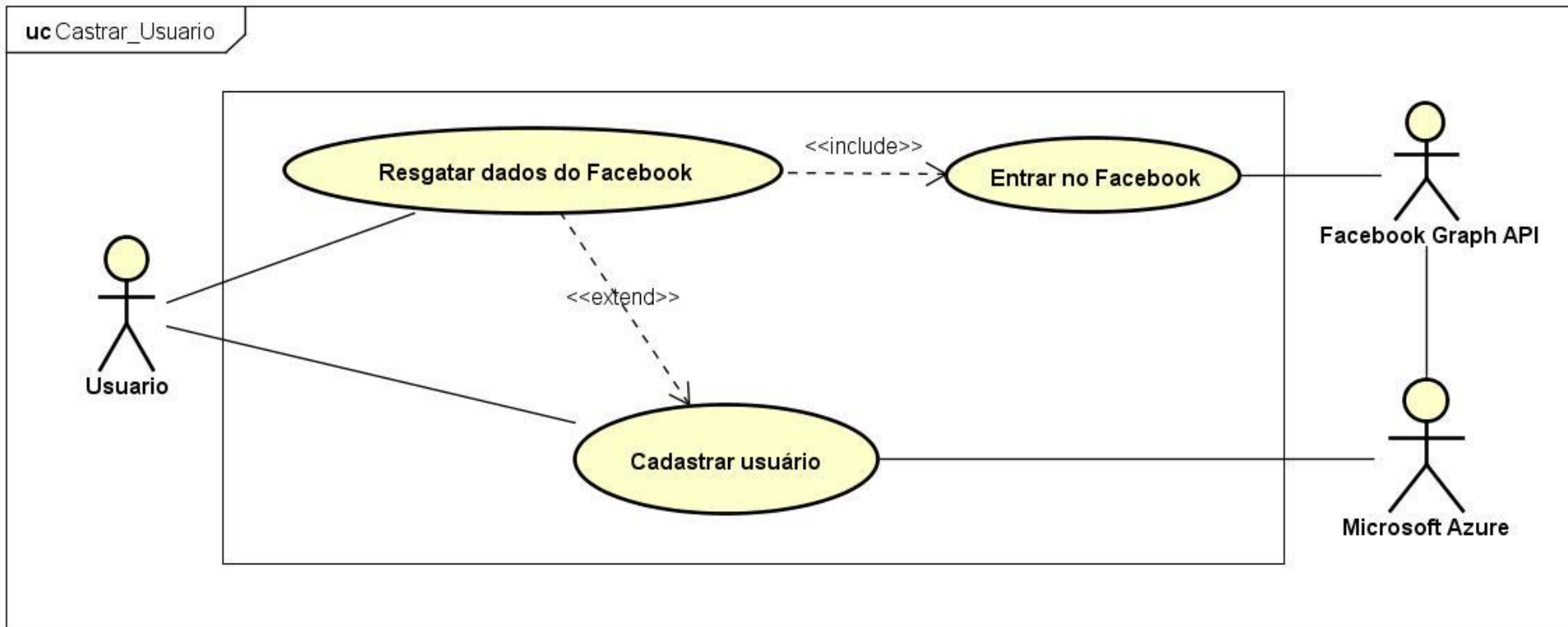
Modelagem - MER



Fonte: Elaborado pelo autor.

Modelagem – Diagrama de Casos de Uso

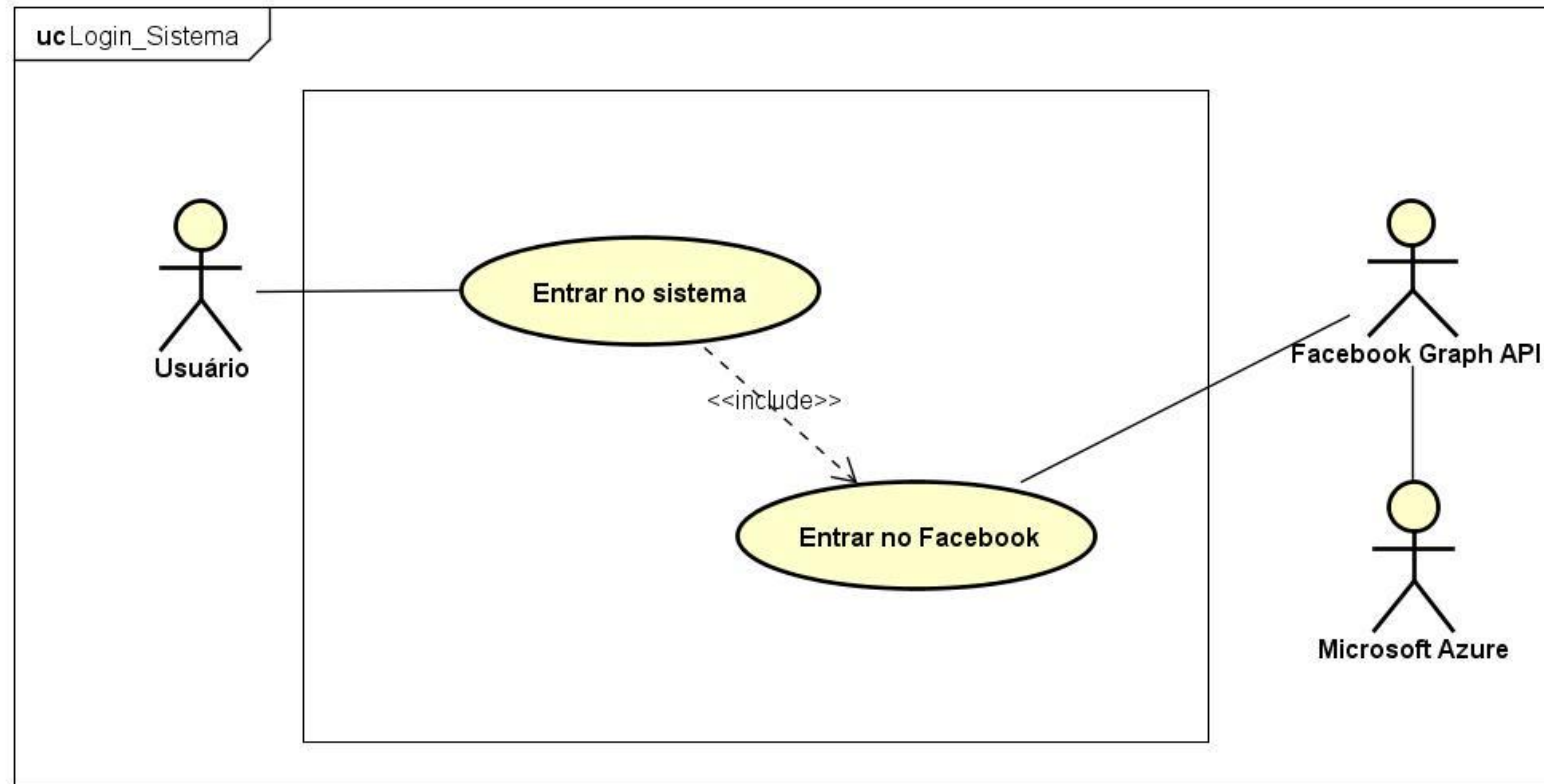
- Diagrama de Caso de Uso Cadastro de Usuário:



Fonte: Elaborado pelo autor.

Modelagem – Diagrama de Casos de Uso

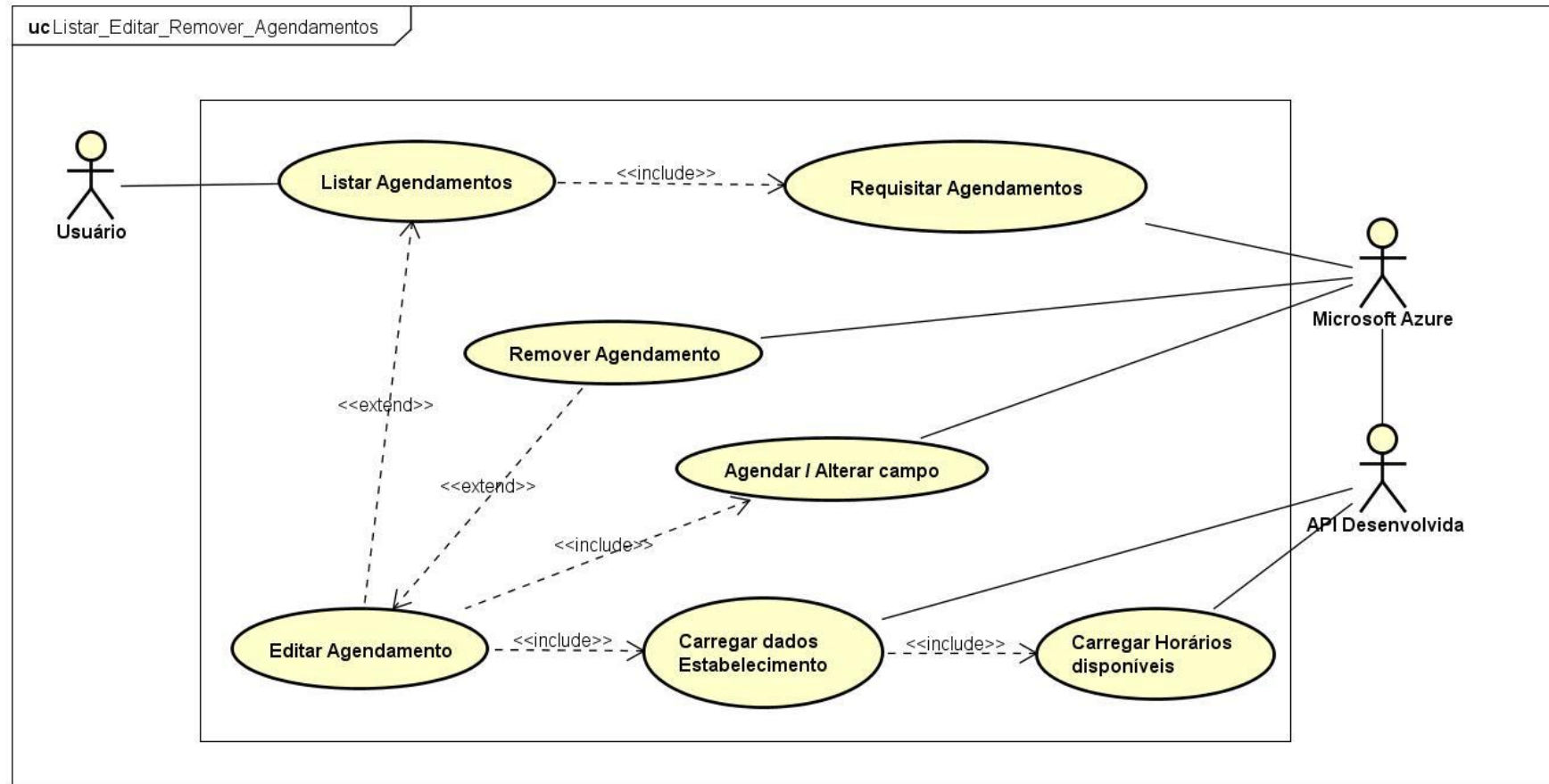
- Diagrama de Caso de Uso Login de Usuário:



Fonte: Elaborado pelo autor.

Modelagem – Diagrama de Casos de Uso

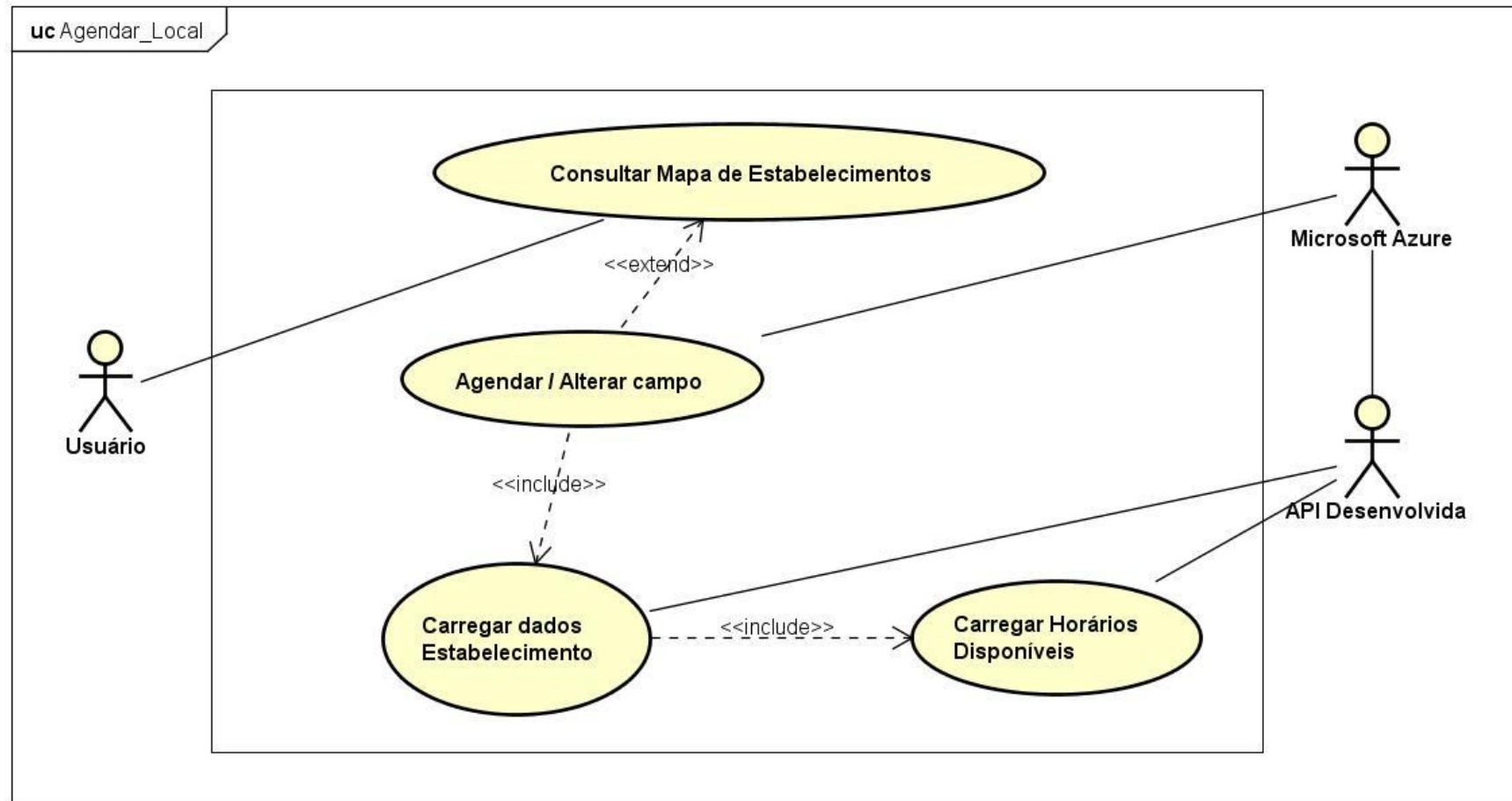
- Diagrama de Caso de Uso de Listar / Editar / Remover agendamento:



Fonte: Elaborado pelo autor.

Modelagem – Diagrama de Casos de Uso

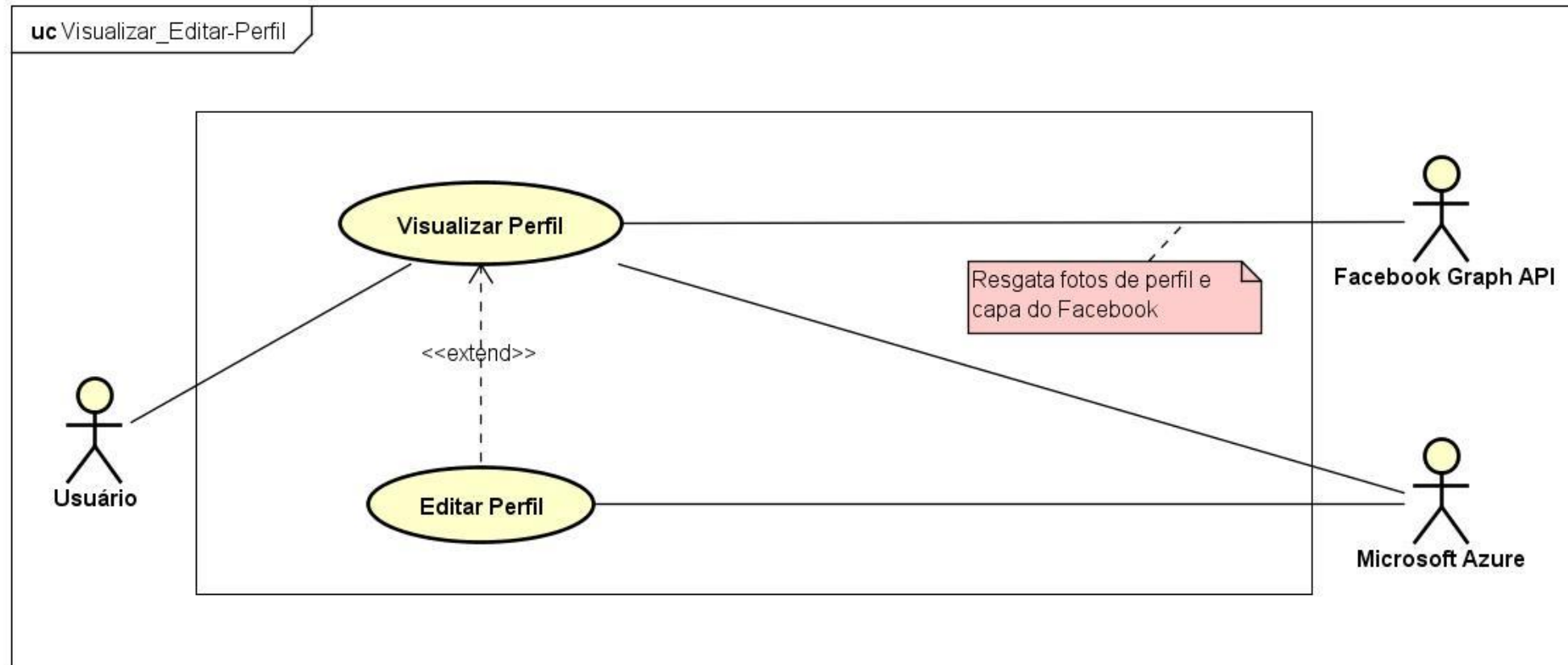
- Diagrama de Caso de Uso Agendar Local:



Fonte: Elaborado pelo autor.

Modelagem – Diagrama de Casos de Uso

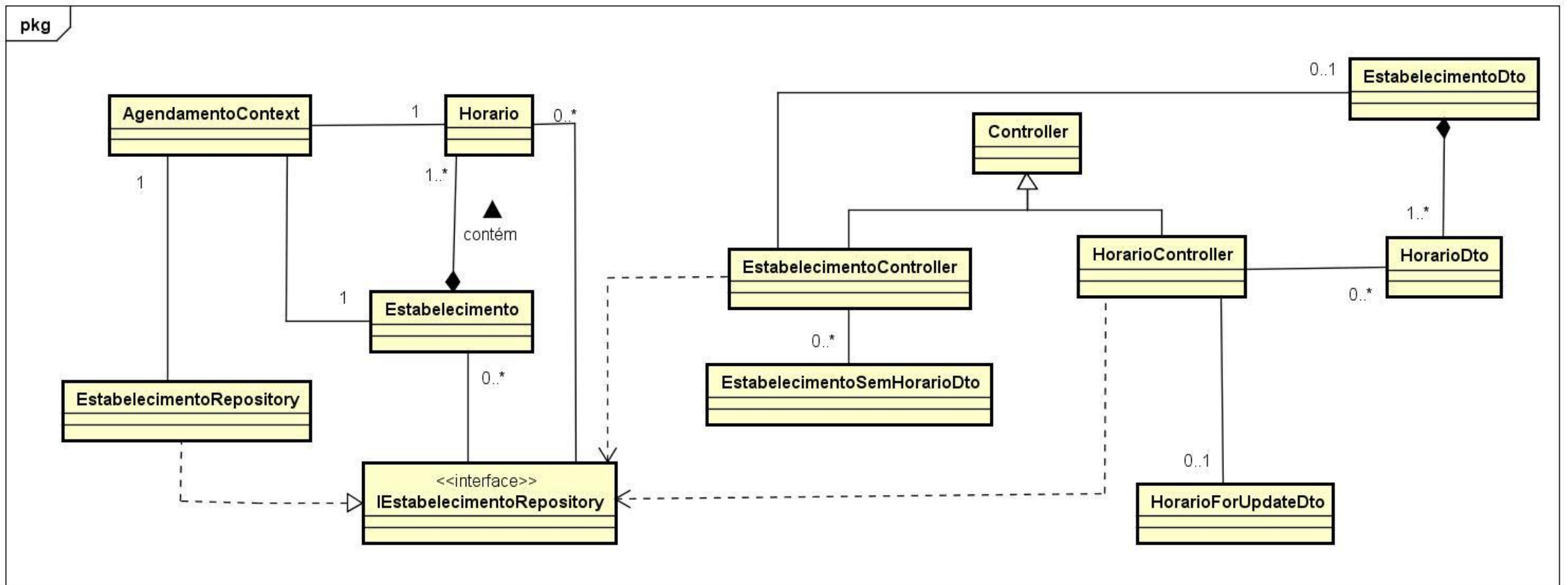
- Diagrama de Caso de Uso Visualizar / Editar Perfil:



Fonte: Elaborado pelo autor.

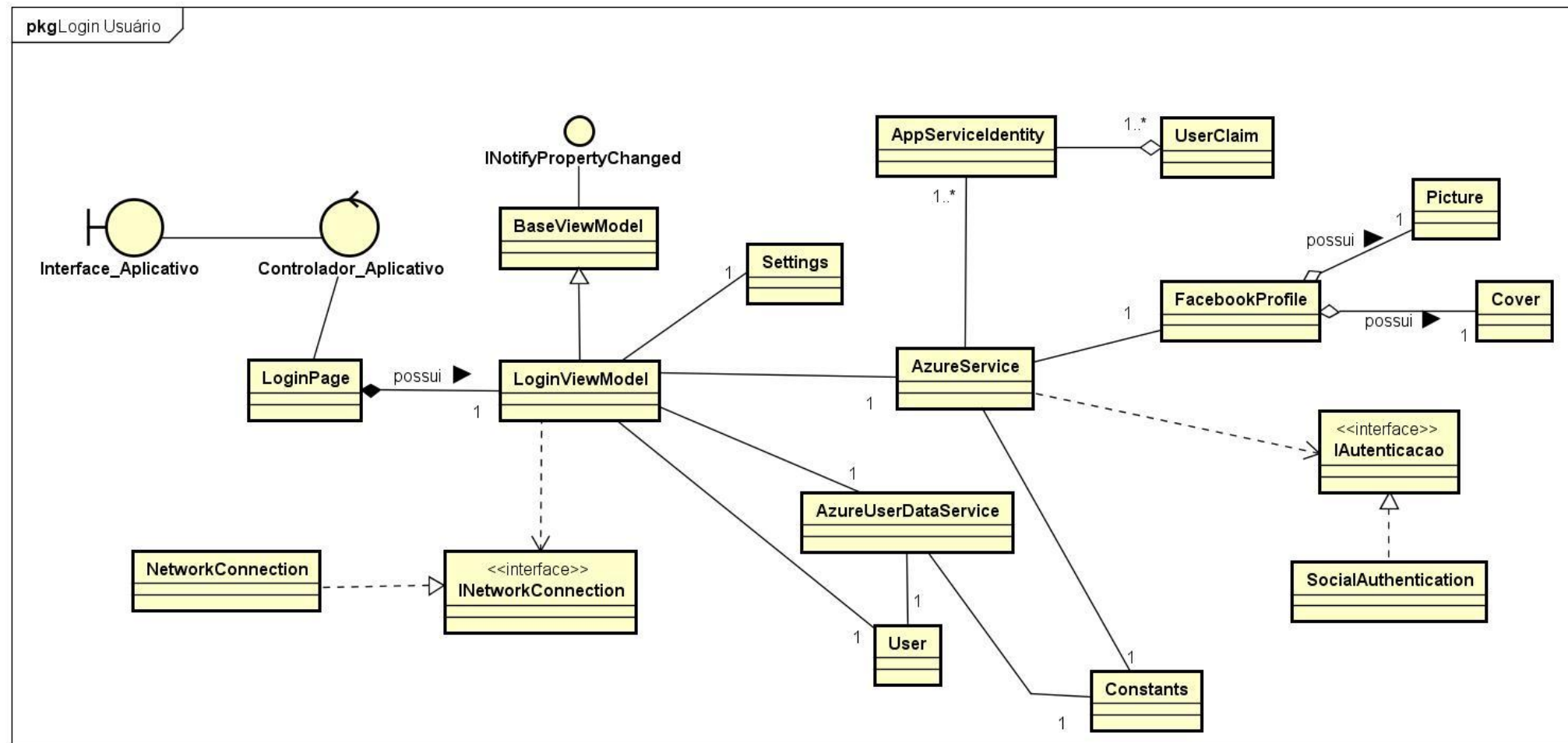
Modelagem – Diagrama de Classes

- Diagrama de Classe da API do projeto:



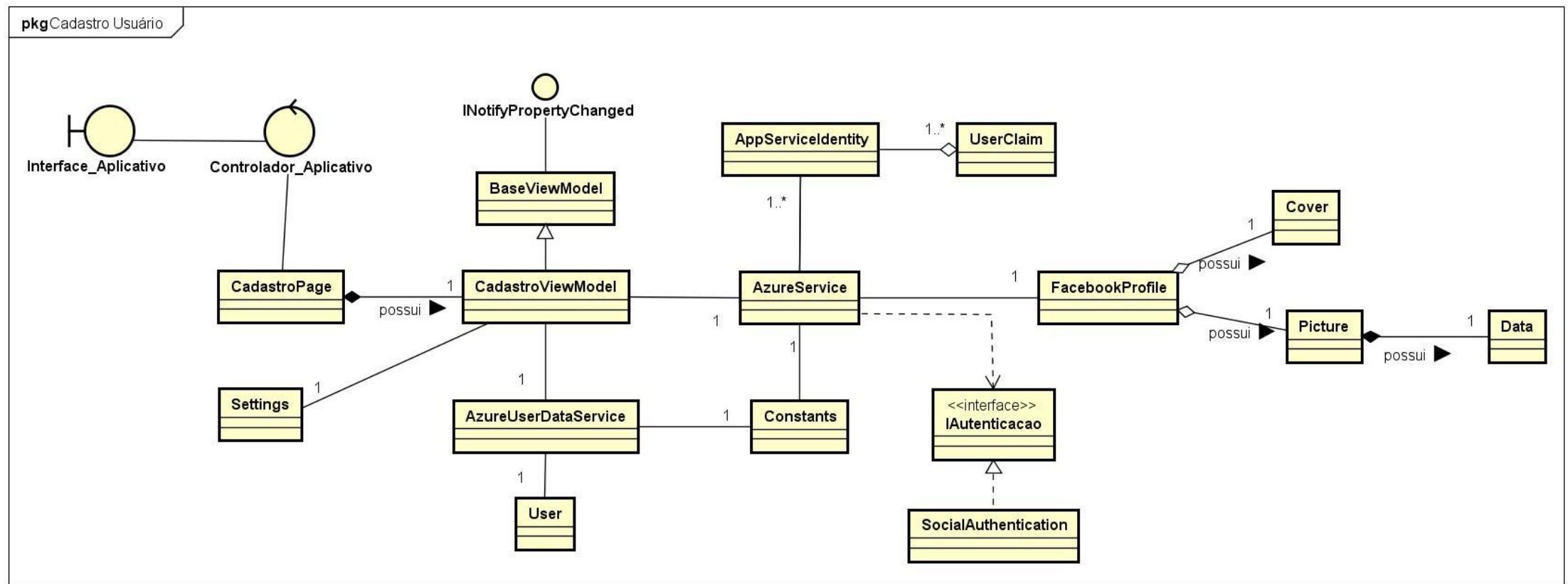
Modelagem – Diagrama de Classes

- Diagrama de Classe da Tela de Login:



Modelagem – Diagrama de Classes

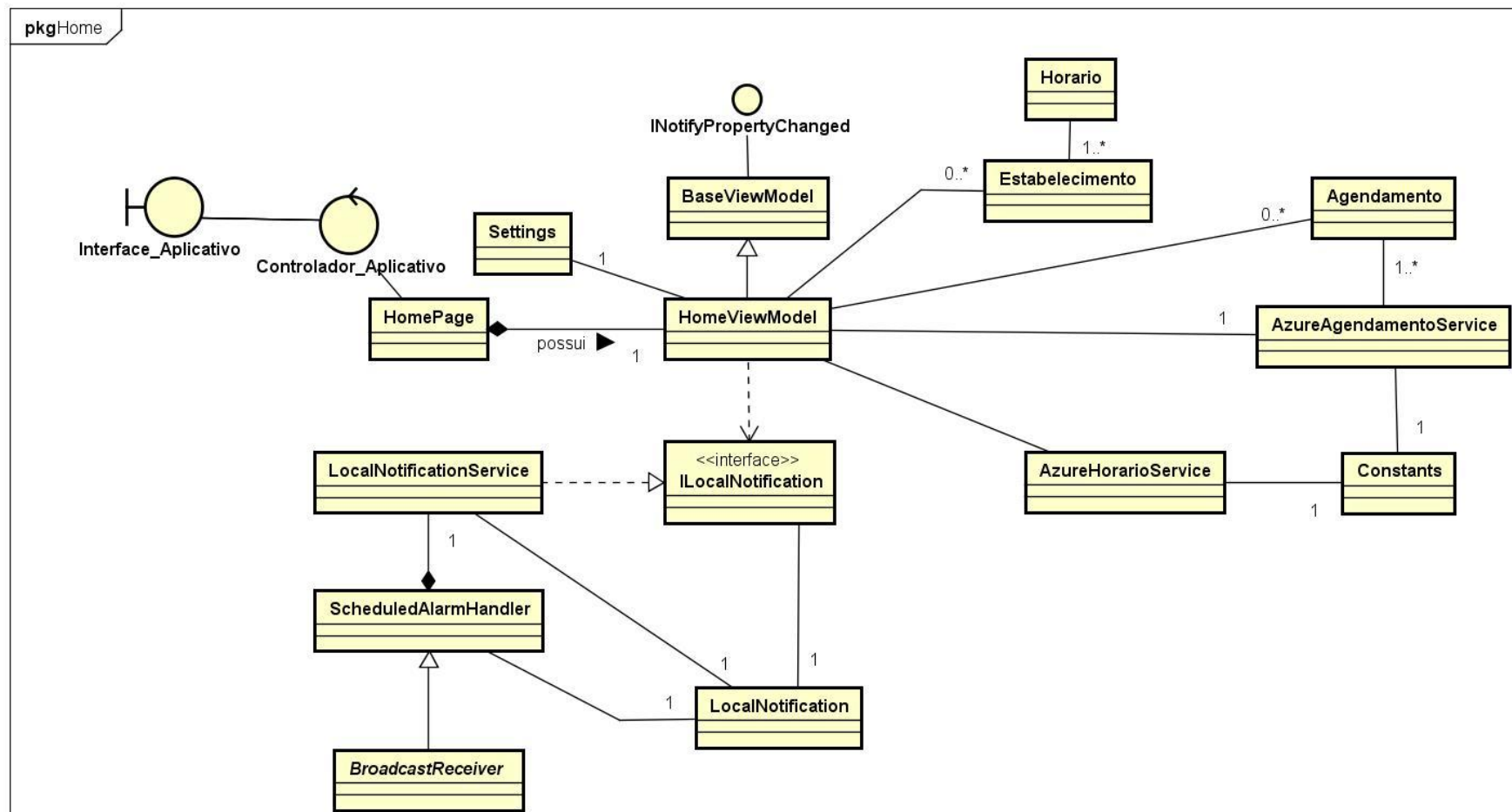
- Diagrama de Classe da Tela de Cadastro de usuário:



Fonte: Elaborado pelo autor.

Modelagem – Diagrama de Classes

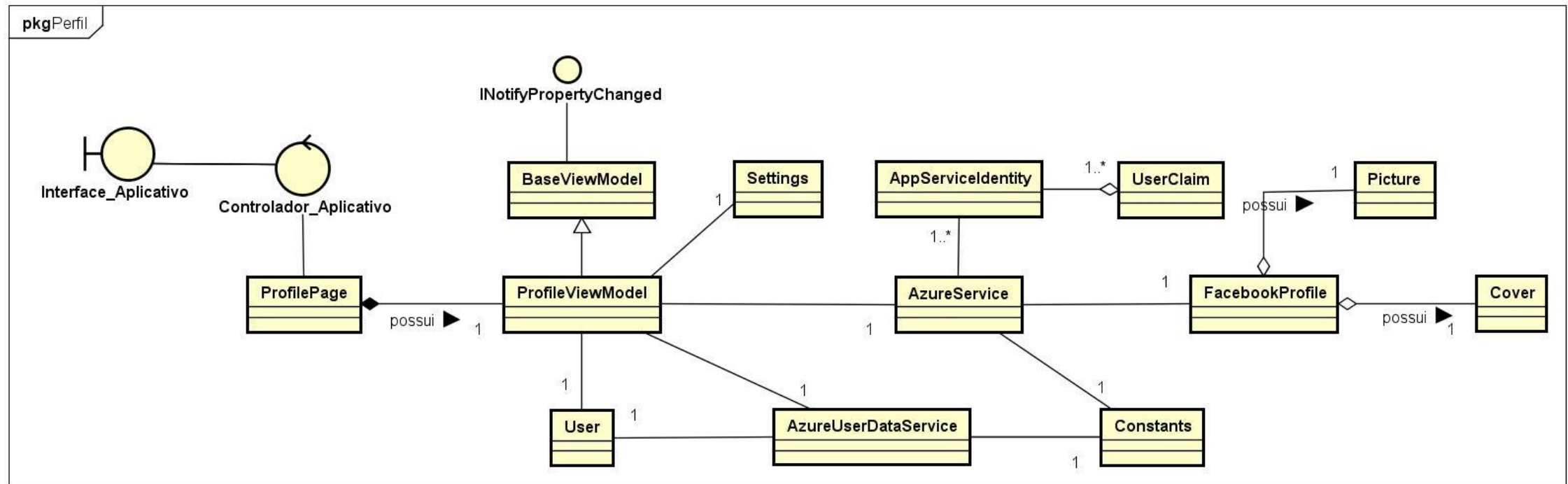
- Diagrama de Classe da Tela Inicial:



Fonte: Elaborado pelo autor.

Modelagem – Diagrama de Classes

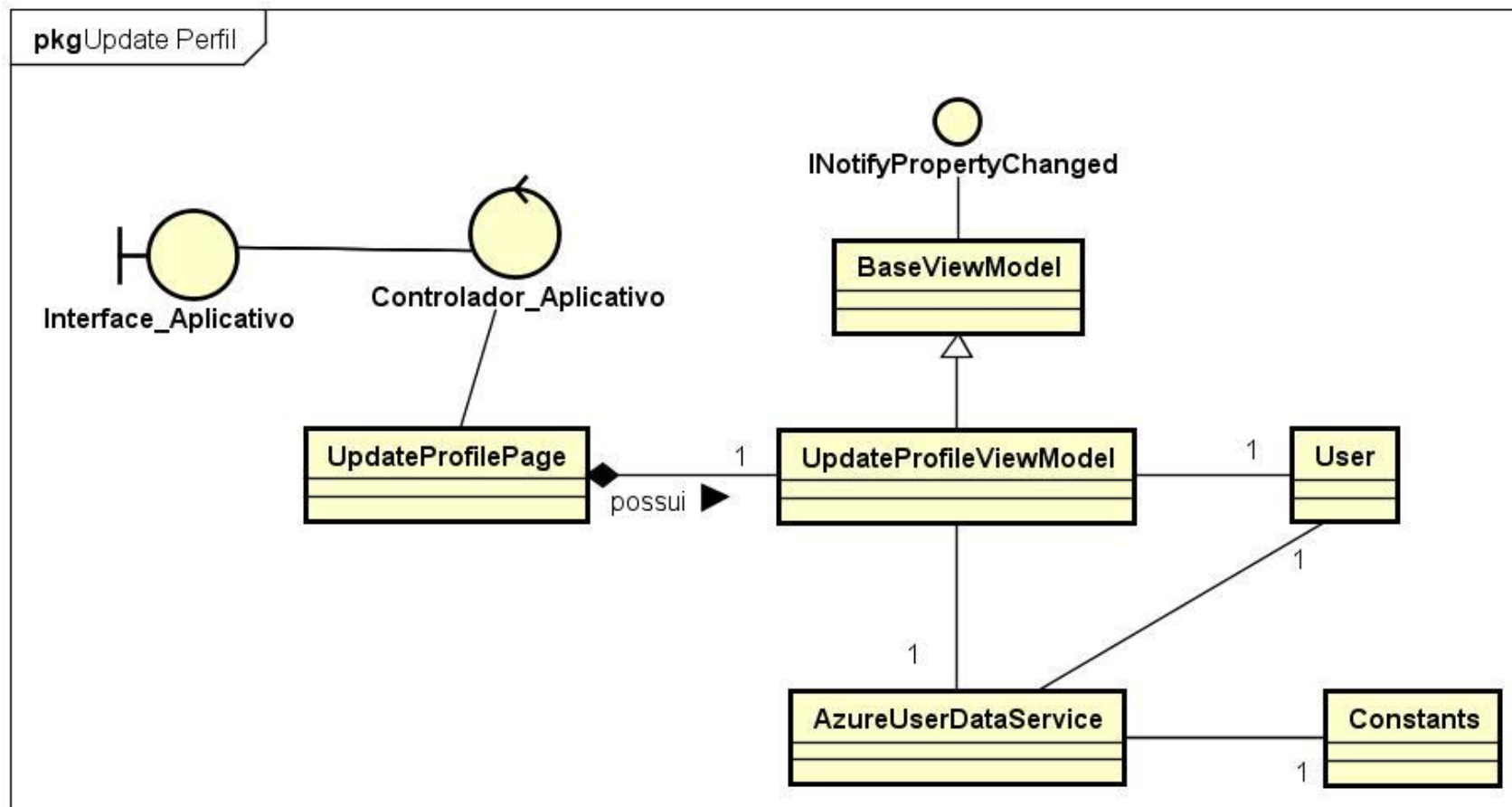
- Diagrama de Classe da Tela de Perfil do usuário:



Fonte: Elaborado pelo autor.

Modelagem – Diagrama de Classes

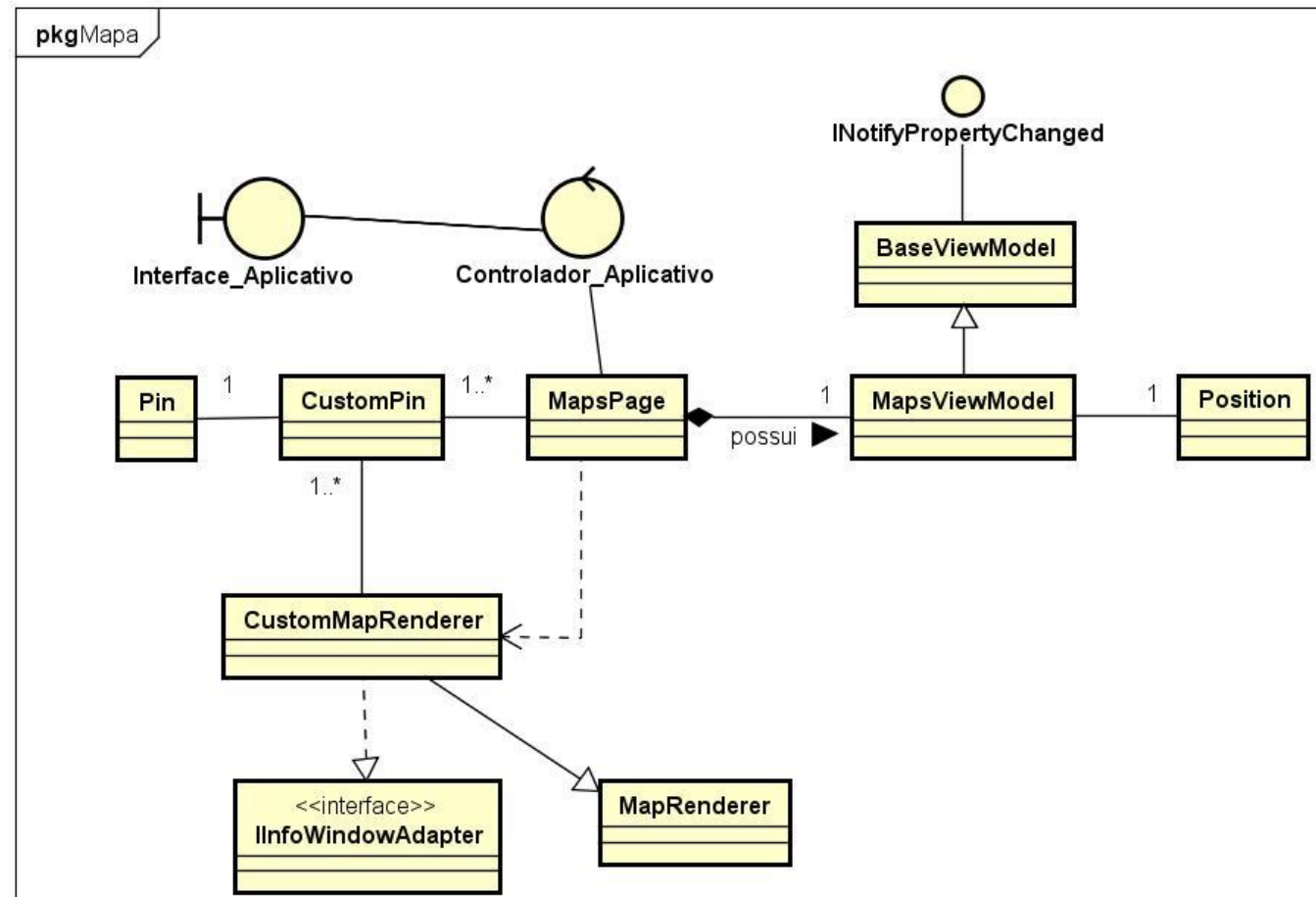
- Diagrama de Classe da Tela de Edição de Perfil:



Fonte: Elaborado pelo autor.

Modelagem – Diagrama de Classes

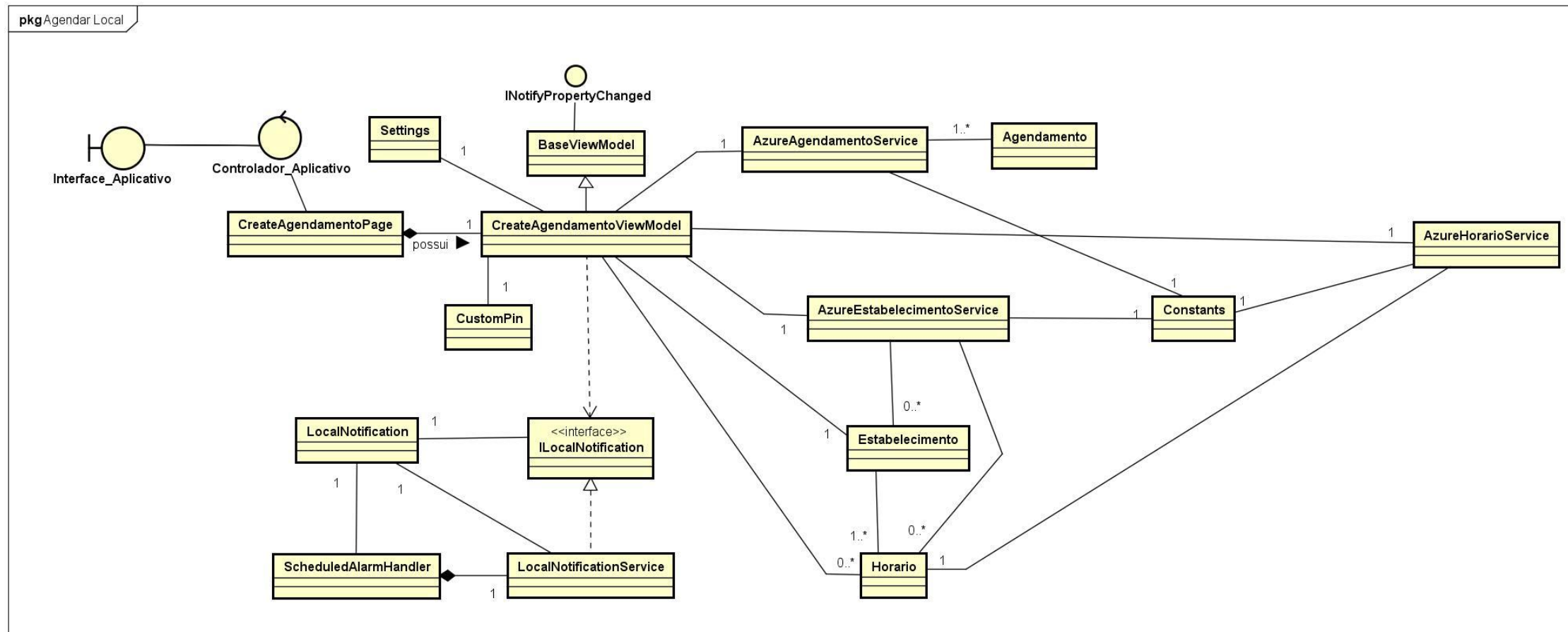
- Diagrama de Classe da Tela de exibição dos Estabelecimentos:



Fonte: Elaborado pelo autor.

Modelagem – Diagrama de Classes

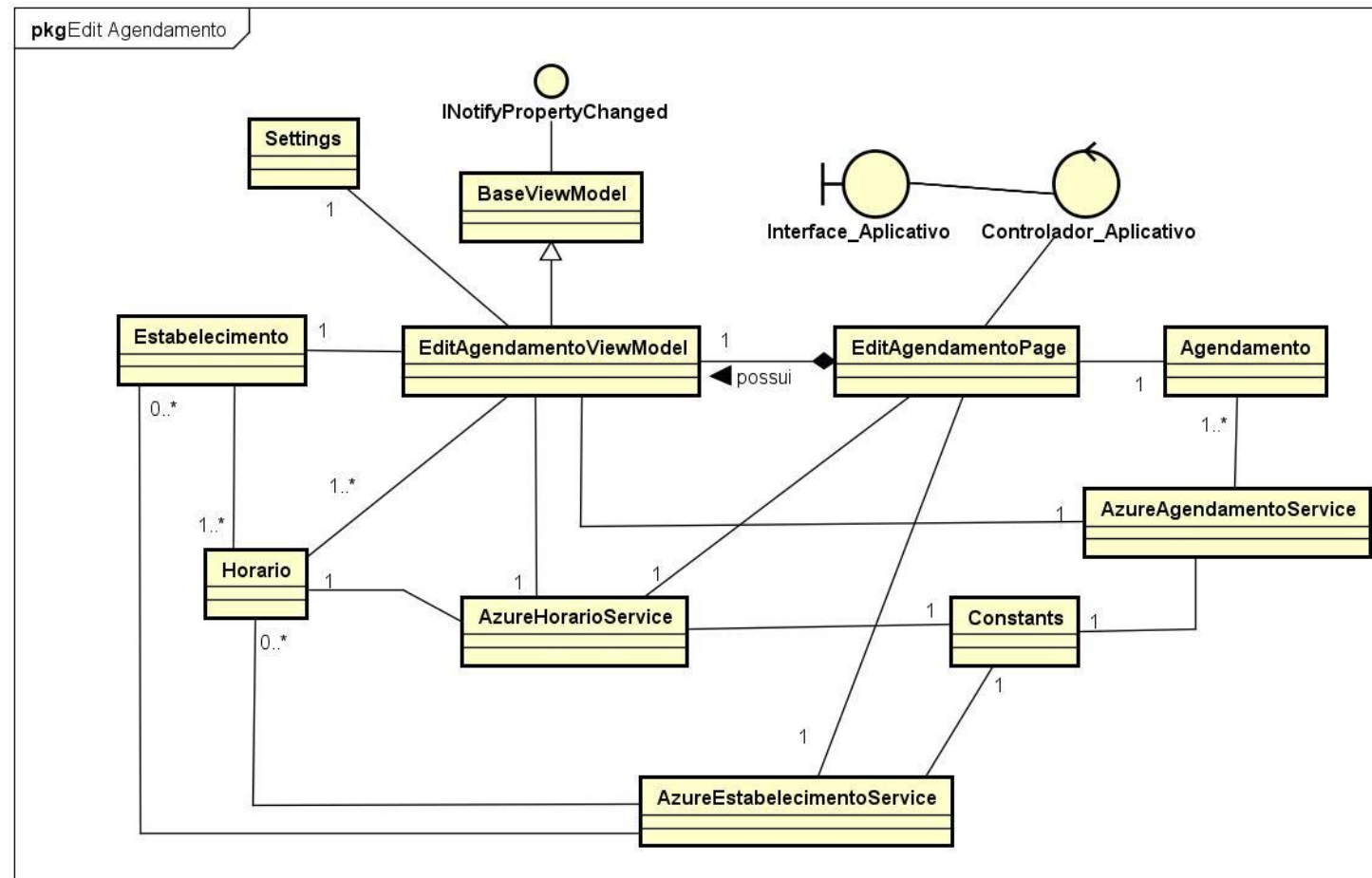
- Diagrama de Classe da Tela de Criação do Agendamento:



Fonte: Elaborado pelo autor.

Modelagem – Diagrama de Classes

- Diagrama de Classe da Tela de Edição de Agendamento:

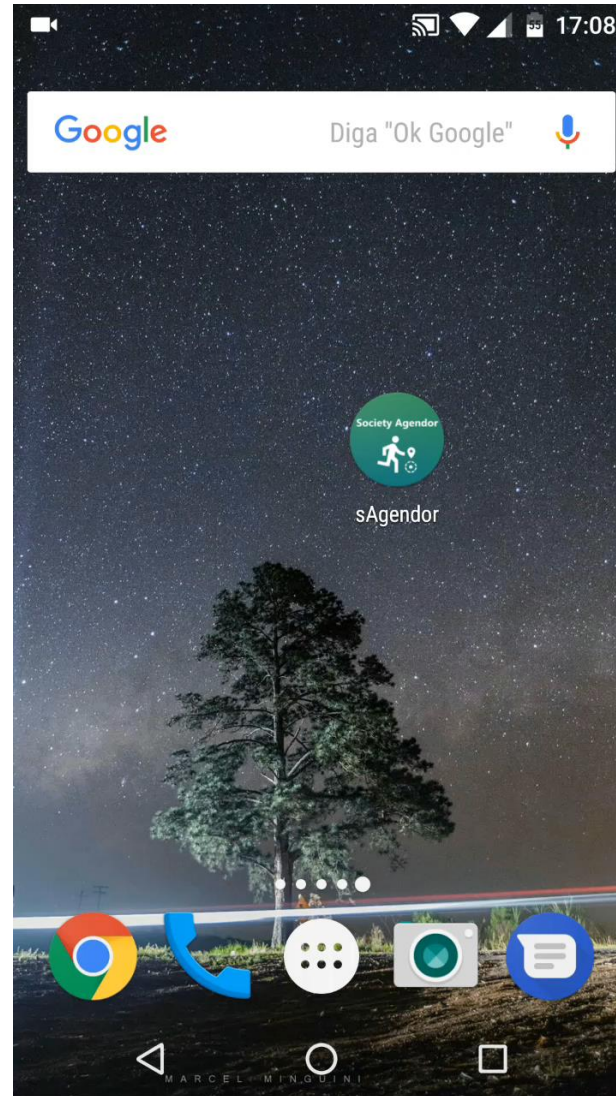


Fonte: Elaborado pelo autor.

Testes

Identificador	Objeto de Teste	Localização	Resultado Esperado	Resultado Obtido
TA001	RF001	Tela de Cadastro de Usuário	<ol style="list-style-type: none"> 1. Sistema recebe o Token de Acesso do usuário. 2. Sistema preenche os campos de texto com as informações recuperadas da Graph API do Facebook a partir do Token de Acesso. 	Realizado com sucesso.
TA002	RF002	Tela Principal	<ol style="list-style-type: none"> 1. O sistema pode receber nenhum ou muitos itens da API. 2. O sistema deverá organizar a exibição dos agendamentos em ordem crescente, de acordo com a data agendada. 3. Se algum erro acontecer, o sistema irá exibir um alerta ao usuário, acusando qual foi o erro. 	Resultados 1 e 2 realizados com sucesso. Resultado 3 só exibirá a exceção gerada pelo sistema e não apontando o erro ocorrido.
TA003	RF003	Tela de Login	<ol style="list-style-type: none"> 1. O sistema abrirá uma tela de login com o Facebook, onde o usuário irá entrar com seu login e senha do provedor. 2. Facebook valida as informações do usuário. 3. O usuário irá aceitar as permissões do aplicativo. 4. Token de Acesso do usuário é armazenado. 	Resultados 1, 3 e 4 realizados com sucesso. Resultado 2 foi realizado com sucesso, mas o sistema não realiza o tratamento de algum erro caso o usuário rejeite as permissões do aplicativo.
TA004	RF004	Tela Principal	<ol style="list-style-type: none"> 1. A API deve retornar no mínimo um agendamento para que possa checar se a data do agendamento é ultrapassada. 2. Se for ultrapassada, o sistema exclui. 	Realizado com sucesso.
TA005	RF005	Tela de Estabelecimentos	<ol style="list-style-type: none"> 1. A API do Google Maps irá demarcar os estabelecimentos (já estabelecidos pelo método) através de pins. 2. O sistema irá identificar a posição atual do usuário, através do GPS. 3. Se o GPS estiver desligado, o sistema irá alertar o usuário. 	Resultado 1 e 2 realizados com sucesso. O resultado 3 não foi desenvolvido.
TA006	RF006	Tela de Login	Mesmo se o usuário preencheu usuário e senha incorretos, o sistema exibirá um alerta para usuário, informando que o aparelho não está conectado com a Internet.	Realizado com sucesso.
TA007	RF007	Tela de Edição de Agendamento	Serão exibidos apenas os agendamentos relacionados ao identificador do usuário atual.	Realizado com sucesso.
TA008	RF008	Tela de Novo Agendamento	O sistema cria uma notificação local uma hora antes do evento da locação acontecer.	Realizado com sucesso.

Demonstração do Aplicativo



Considerações Finais

- Processo de desenvolvimento do aplicativo.
- Utilização de recursos da nuvem.
- Dificuldades encontradas.
- Melhorias futuras:
 - Convidar amigos;
 - Ponto de intersecção entre os amigos;
 - Portal para o cadastro de estabelecimentos;
 - Segurança na API;
 - Centralizar as transações de banco dados na API.

Referências

FACEBOOK FOR DEVELOPERS. **Visão geral da Graph API**, 2017. Disponível em: <<https://developers.facebook.com/docs/graph-api/overview/>>. Acesso em 07 de set. 2017.

GOOGLE DEVELOPERS. **Introdução à Google Maps Android API**, 2017. Disponível em: <<https://developers.google.com/maps/documentation/android-api/intro?hl=pt-br>>. Acesso em 09 de out. 2017.

GUEDES, Gilleanes T. A. **UML 2: uma abordagem prática**. São Paulo: Novatec Editora, 2009.

MICROSOFT. O que é o SaaS? *In.*: **Microsoft Azure: Visão geral**, 2017. Disponível em: <<https://azure.microsoft.com/pt-br/overview/what-is-saas/>>. Acesso em 13 de jun. 2017, às 14h38min.

MICROSOFT. O que é PaaS? *In.*: **Microsoft Azure: Visão geral**, 2017. Disponível em: <<https://azure.microsoft.com/pt-br/overview/what-is-paas/>>. Acesso em 13 de jun. 2017, às 14h15min.

MICROSOFT. Por que o Azure? *In.*: **Microsoft Azure: O que é o Azure?**, 2017. Disponível em: <<https://azure.microsoft.com/pt-br/overview/what-is-azure/>>. Acesso em 10 de jun. 2017, às 18h36min.

MICROSOFT. The MVVM Pattern. *In.*: **Microsoft: developer network**, 2012. Disponível em: <[https://msdn.microsoft.com/pt-br/library/hh848246\(d=printer\).aspx](https://msdn.microsoft.com/pt-br/library/hh848246(d=printer).aspx)>. Acesso em: 15 jun 2017.

MICROSOFT. Um tour pela linguagem C#. *In.*: **Microsoft Docs**, 2016. Disponível em: <<https://docs.microsoft.com/pt-br/dotnet/csharp/tour-of-csharp/index>>. Acesso em 03 de maio 2017, às 17h35min.

MONO. About Mono. *In.*: **Docs**, 2016. Disponível em: <<http://www.mono-project.com/docs/about-mono/>>. Acesso em 14 de jun. 2017, às 15h.

SOMMERVILLE, Ian. **Engenharia de Software**. 9ª. ed. São Paulo: Pearson Prentice Hall, 2011. p. 2-159.



Society Agendor

Aplicativo para agendar campo de futebol society.

Aluno: Vinícius de Souza Avansini

RA: 0040481512038

Orientador: Prof. Dr. Kleber de Oliveira Andrade