Documentação da Arquitetura de Software

Arquitetura do Sistema de Caronas

A Figura 1 ilustra a arquitetura de software a nível de módulos adotada para o sistema de caronas. Na imagem é possível ver a estruturação em camadas e subsistemas adotada, além das interações entre módulos a partir das interfaces, representadas por notações baseadas na UML.

A partir das nomenclaturas do subsistema e da camada relativos a um módulo, fica evidente o significado das abreviações (Ex: M. Aut. Apres. Representa o Módulo de Autenticação de Apresentação). De maneira análoga, é possível compreender o significado das siglas das interfaces (Ex: IUS é a Interface de Usuários de Serviço).

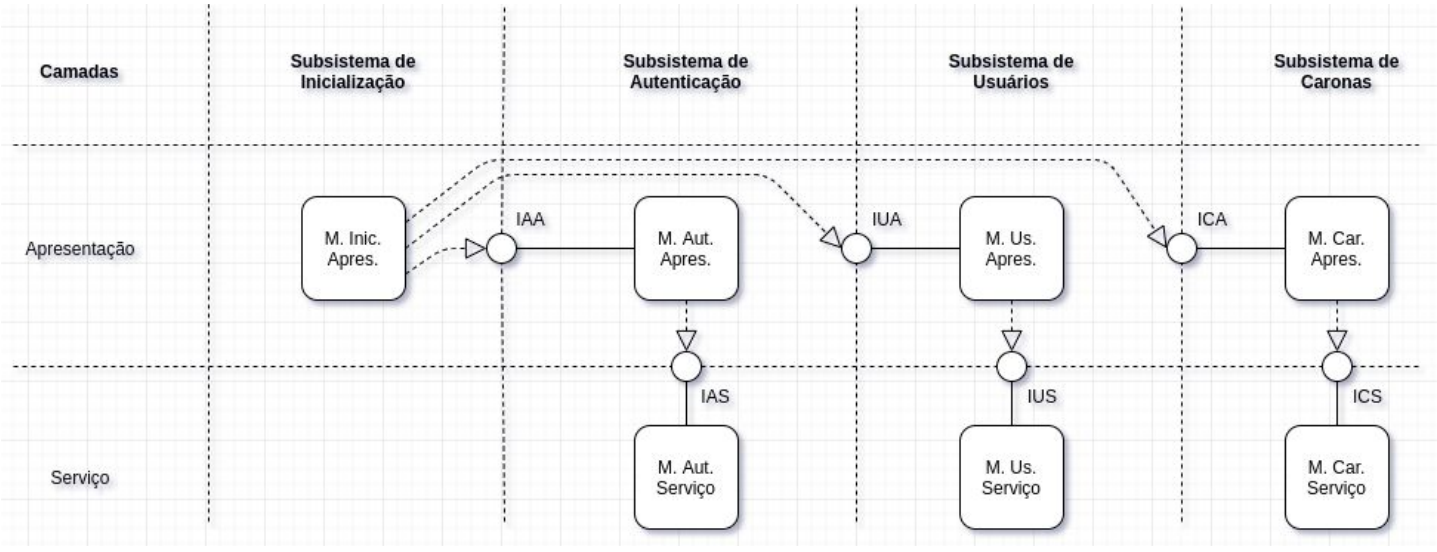

Figura 1 - Arquitetura do Software

Módulos do Sistema

- MIA

O Módulo de Inicialização de Apresentação (MIA) é responsável pela apresentação da Tela Inicial com as opções de serviço disponíveis ao cliente da aplicação. Nesse contexto,

fornece ao usuário a possibilidade de selecionar a ação necessária, interpreta a solicitação e a destina adequadamente.

- MAA

O Módulo de Autenticação de Apresentação é responsável por apresentar ao cliente do sistema os dados necessários para efetuar o “log-in” e captar o conteúdo preenchido. Além disso, verifica se os dados estão de acordo com o que foi definido para os domínios do sistema. Caso os dados estejam inválidos, notifica o cliente.

- MUA

O Módulo de Usuário de Apresentação é responsável por informar e captar do cliente os dados necessários para se cadastrar no sistema. Todo campo preenchido passa por uma validação do formato da entrada de acordo com o que foi definido para o sistema.

Também permite que um usuário cadastrado solicite a exclusão da sua conta.

- MCA

O Módulo de Carona de Apresentação também apresenta ao cliente os dados exigidos e valida os dados de entrada a depender do serviço. Porém, por conta do domínio de aplicação do sistema, possui uma série de telas (serviços) relacionadas às diversas solicitações referentes a caronas e reservas, como cadastro, cancelamento, listagem, etc.

- MAS

O Módulo de Autenticação de Serviço é responsável por, a partir dos dados fornecidos para autenticação, verificar se já existe um usuário com aquele registro e se aquela autenticação pode ser validada ou não. Para isso interage com o repositório destinado aos Usuários.

- MUS

O Módulo de Usuário de Serviço realiza os serviços de cadastrar ou excluir um usuário. Após obter os dados de um usuário, avalia as regras de negócio para realização das ações demandadas, como por exemplo avaliar se um usuário que deseja se cadastrar já é cadastrado ou se uma exclusão de um usuário de fato pode ser feita (definida pelas regras de negócio).

Nesse contexto, o módulo em questão tem de interagir com os repositórios de todas as classes de entidades do sistema.

- MCS

O Módulo de Carona de Serviço realiza os serviços centrais do software, como gerenciar as caronas e suas reservas. Avalia as regras de negócio para cadastrar e descadastrar caronas e reservas, bem como obter dos repositórios adequados os dados relacionados com alguma característica demandada.

Gerencia a persistência das entidades fundamentais para o funcionamento do software que determinam o domínio de aplicação.

Interfaces

A Figura 1 evidencia as relações de dependência e realização de interfaces pelos módulos. De modo a evidenciar a natureza dos serviços e deixar a descrição dos módulos mais palpável, os métodos virtuais puros declarados nas interfaces são exibidos na sequência.

- IAA - Interface de Autenticação de Apresentação

```
virtual bool autenticar(Email * email) = 0;
```

- IAS - Interface de Autenticação de Serviço

```
virtual bool autenticar(Email * email, Senha senha) = 0;
```

- IUA - Interface de Usuários de Apresentação

```
virtual void cadastrar() = 0;
virtual bool excluir(Email * email) = 0;
```

- IUS - Interface de Usuários de Serviço

```
virtual bool cadastrarUsuario(Usuario * usuario) = 0;
virtual void cadastrarConta(Conta * conta) = 0;
virtual bool excluir(Email * email) = 0;
```

- ICA - Interface de Caronas de Apresentação

```
virtual void cadastrarCarona(Email * email) = 0;
virtual void descadastrarCarona(Email * email) = 0;
virtual void reservarCarona(Email * email) = 0;
virtual void obterDadosCarona() = 0;
virtual void listarReservas(Email * email) = 0;
virtual void cancelarReserva(Email * email) = 0;
```

- ICS - Interface de Caronas de Serviço

```
virtual bool cadastrarCarona (Carona * carona, Email * email) = 0;
virtual int descadastrarCarona (Email * email, CodigoDeCarona * codigo) = 0;
```

```
        virtual bool efetuarReserva (CodigoDeCarona * rideCode, Assento * seat,
Bagagem * bag, CodigoDeReserva * reservaCode, vector<Conta> * vetorDeContas, Email *
email) = 0;
        virtual vector<Carona> pesquisarCaronas(Carona * dominiosSolicitados) = 0;
        virtual bool listarReservas(Email * email, CodigoDeCarona * rideCode,
vector<Reserva> * vetorDeReservas) = 0;
        virtual bool cancelarReserva(CodigoDeReserva * reservaCode) = 0;
```