Sapien Soft

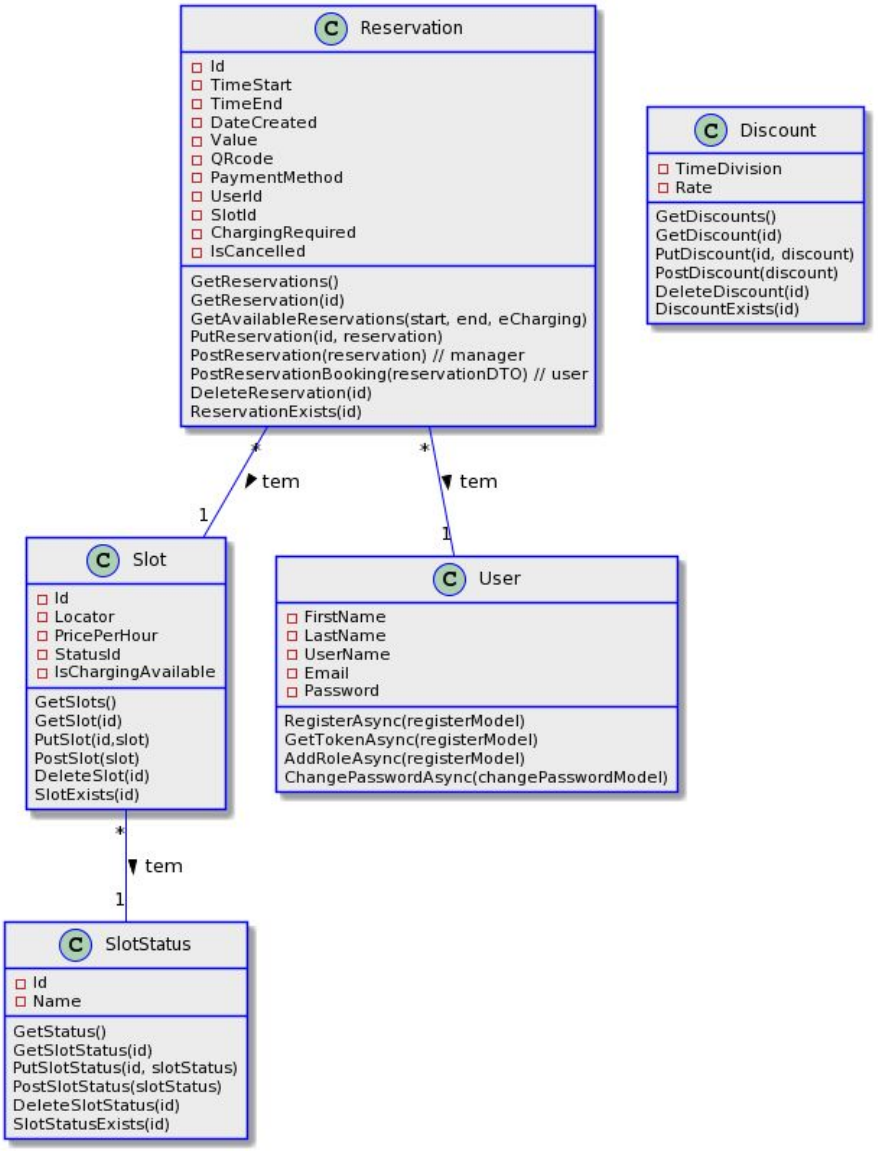
#SPRINT2

API Private Park - Requisitos, Análise e Design

Índice:

- Estrutura e Associação entre classes (pág 1-2);
- User Stories (pág 2);
- UC1 - Registo de User (pág 3-5);
- UC2 - Disponibilizar lugares disponíveis (pág 6-9);
- UC3 - Gestão do Parque - CRUD (pág 10-13);
- UC4 - User faz uma Reserva (pág 14-17);
- UC5 - User cancela Reserva (pág 18-20);
- UC6 - User altera a sua password (pág 21-24).

Estrutura da API:



Associação entre classes

Conceito A	Associação	Conceito B
ParkManager(UserRole)	define (CRUD)	- Discount - Status - Slots - Reservation - User
API Private Park	possui	- Reservation - Discount
Reservation	possui	- User - Slot
Slot	possui	- Status

User Stories

User Story	As a <type of user>	I want to <perform some task>	So that I can <achieve some goal>
1	ParkManager (Administrator)	Registo na Plataforma.	Utilizar os serviços da Plataforma..
2	User	Obter lugares disponíveis para determinada data/hora de início e fim.	Conseguir encontrar a melhor oferta (lugar ao melhor preço/localização).
3	ParkManager (Administrator)	Proceder à Gestão interna do Parque através de operações CRUD sobre os seus principais elementos.	Atualizar registos da Plataforma.
4	User	Fazer uma reserva.	Garantir um lugar de estacionamento.
5	ParkManager (Administrator), User	Cancelar uma reserva previamente registada.	Evitar custos com um serviço que não vou utilizar.
6	ParkManager (Administrator), Use	User altera password	Garantir segurança da minha conta.

UC1 - Registo de User

1. Requisitos

1.1 Breve Descrição

O ParkManager inicia o registo de um novo User. O sistema solicita os dados necessários sobre o User (i.e. UserName, Email, PhoneNumber). O ParkManager introduz os dados solicitados. O Sistema valida e apresenta os dados pedindo confirmação. O ParkManager confirma. O Sistema regista os dados do novo User e informa o ParkManager do sucesso da operação.

1.2 SSD

1.3 Descrição Completa

1.3.1 Ator Principal

ParkManager

1.3.2 Partes Interessadas

- **User não-registado** - Ficará registado e poderá usufruir dos serviços da API do Parque Privateo.
- **ParkManager** - Mais Users a usar os seus serviços.

1.3.3 Pré-condições

Condições estabelecidas pelas leis nacionais / locais destinadas a este serviço, sendo que o novo User teve que ser submetido a um processo de concurso pela gestão do parque.

1.3.4 Pós-condições

A informação do registo é armazenada no sistema.

1.3.5 Cenário Principal de Sucesso

1. O ParkManager faz HttpPost com os dados do novo User;
2. O sistema confirma e apresenta mensagem de sucesso;

1.3.6 Extensões (ou fluxos alternativos)

- O novo User solicita ao ParkManager o cancelamento do registo.
    - O caso de uso termina.

- O sistema informa que dados mínimos obrigatórios estão em falta.

        - O sistema informa que dados obrigatórios estão em falta.
        - O sistema permite a introdução dos dados em falta.
            - O ParkManager não altera os dados. O caso de uso termina.

- O sistema informa que detetou que os dados (ou algum subconjunto dos dados) introduzidos devem ser únicos e que já existem no sistema.
    - O sistema alerta o ParkManager para o facto.
    - O sistema permite a sua alteração.
        - O ParkManager não altera os dados. O caso de uso termina.

1.3.7 Requisitos Especiais

1.3.8 Variações de Tecnologias de dados

1.3.9 Frequência de ocorrência

Uma única vez.

1.3.10 Questões em Aberto

- Os utilizadores finais (clientes da Sapiensoft) podem estacionar no parque privado sem reserva?

2. Análise OO

2.1. Modelo de Domínio relevante para o Caso de Uso

3. Design

3.1. Racional

Fluxo principal	Questão: Que classe...	Resposta	Justificação
1. O ParkManager envia um pedido HttpPost para registar no User	... interage com o ParkManager?	UserController	Controller
	... coordena o UC?	UserController	Controller
2. A classe UserController solicita à classe RegisterController a criação de um objeto.			

3. O RegisterModel cria o objeto 'model'.	… guarda os dados introduzidos?	RegisterModel	creator
4. A classe UserController valida e regista o 'model' na classe IUserService.	… valida os dados?	IUserService	IE: possui os seus próprios dados
6. A classe IUserService devolve ao ParkManager uma mensagem de sucesso da operação.	… guarda o User criado?	ApplicationUser	IE: No MD o API Private Park tem Users
	… altera o Role do User?	IUserService	IE: A gestão das Roles de Users é responsabilidade do UserService

3.2. Sistematização

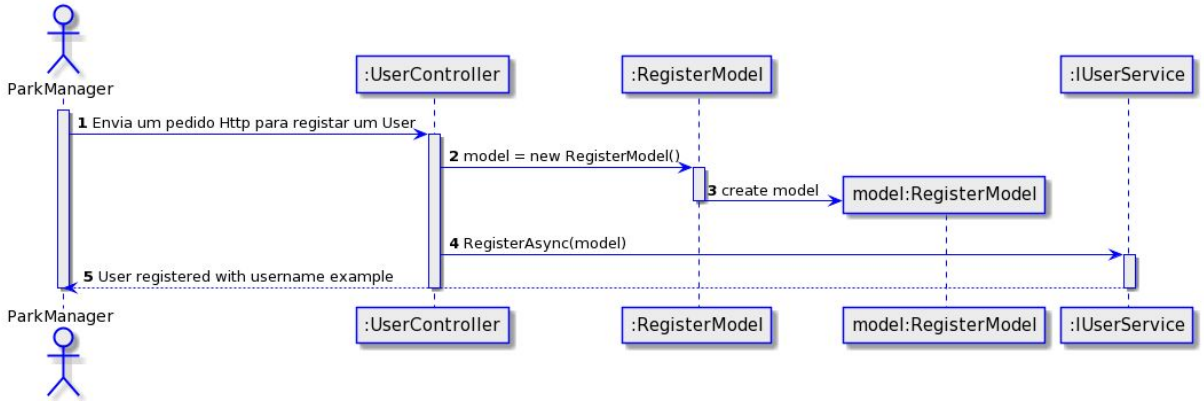
Do racional resulta que as classes conceituais promovidas a classes de software são:

- ApplicationUser
- RegisterModel

Outras classes de software identificadas:

- UserController
- IUserService

3.3. Diagrama de Sequência



3.4. Diagrama de Classes

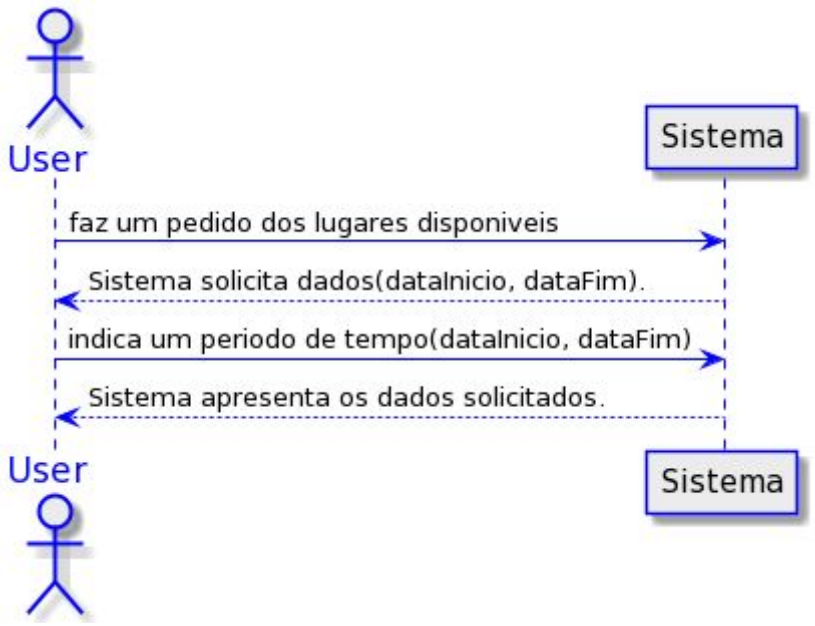
UC2 - Disponibilizar lugares disponíveis

1. Requisitos

1.1 Breve Descrição

O User(Plataforma Externa), registada pelo ParkManager, faz um pedido so Sistema (API Private Park) de todos os lugares disponíveis no parque, dentro de um período de tempo. A API Private Park deverá retornar a informação pretendida bem como o custo total da eventual reserva naquele período de tempo solicitado.

1.2 SSD



1.3 Descrição Completa

1.3.1 Ator Principal

User(Plataforma Externa, registada pelo ParkManager)

1.3.2 Partes Interessadas

- **ParkManager** - Utilização dos seus serviços.
- **User(Plataforma Externa)** - Disponibilização de informação detalhada para os seus users.

1.3.3 Pré-condições

User tem de ser registado e validada pelo ParkManager.

1.3.4 Pós-condições

O User(Plataforma Externa) consegue aceder à informação solicitada.

1.3.5 Cenário Principal de Sucesso

1. O User(Plataforma Externa) acede ao sistema e solicita os lugares vagos, dentro de uma data de inicio e uma data de fim
2. O Sistema apresenta os dados solicitados.

1.3.6 Extensões (ou fluxos alternativos)

- O User(Plataforma Externa) não insere data de inicio e/ou data de fim.
  - O sistema informa que dados mínimos obrigatórios estão em falta.
  - O sistema permite a inserção dos dados.

- Durante o tempo de escolha, um dos lugares apresentados ficou reservado naquele horário escolhido pelo User(Plataforma Externa).
    - Ao seleccionar a possível reserva, o Sistema terá de informar o User que a reserva seleccionada foi reservada durante o processo.
    - O caso de uso termina.

1.3.7 Requisitos Especiais

1.3.8 Variações de Tecnologias de dados

1.3.9 Frequência de ocorrência

Diária.

1.3.10 Questões em Aberto

---

2. Análise OO

2.1. Modelo de Domínio relevante para o Caso de Uso

3. Design

3.1. Racional

Fluxo principal	Questão: Que classe...	Resposta	Justificação
O User(Plataforma Externa) acede ao sistema e solicita os lugares vagos, dentro de uma data de inicio e uma data de fim	... interage com o User?	ReservationsController	Controller
	... coordena o UC?	ReservationsController	Controller

3.2. Sistematização

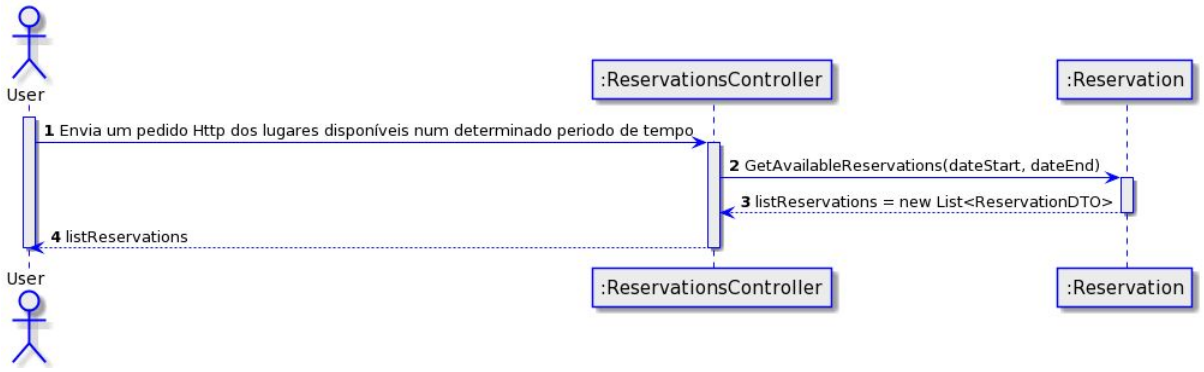
Do racional resulta que as classes conceptuais promovidas a classes de software são:

- Reservation

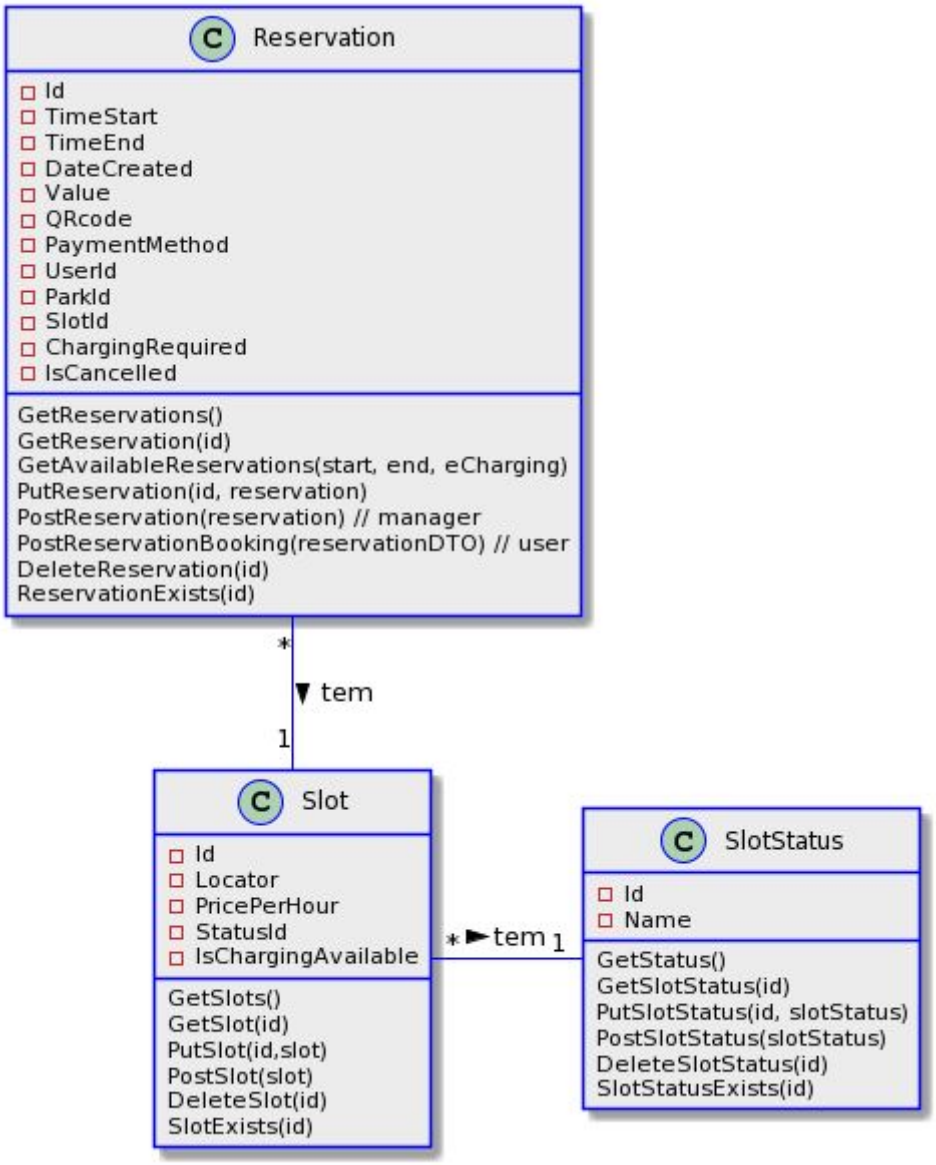
Outras classes de software identificadas:

- ReservationsController

3.3. Diagrama de Sequência



3.4. Diagrama de Classes



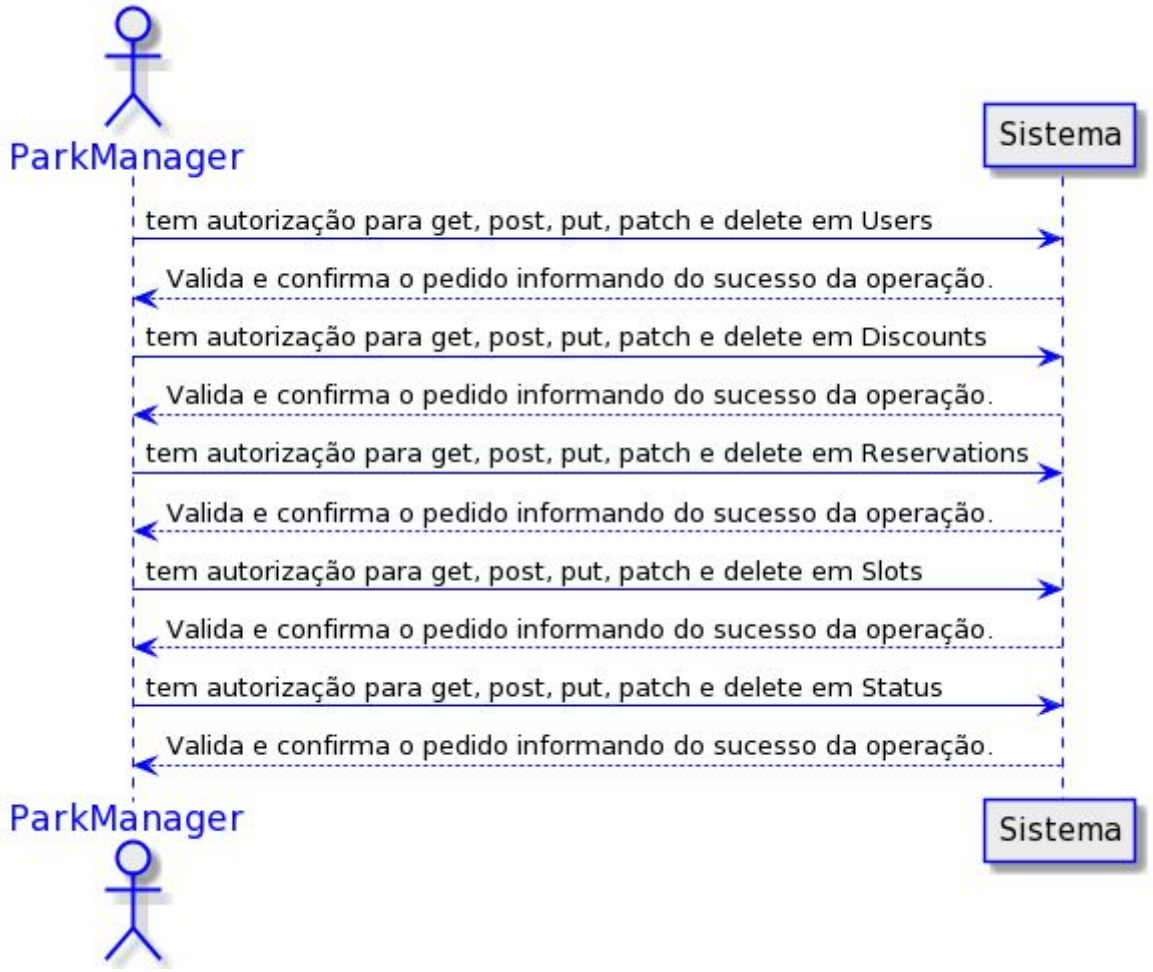
UC3 - Gestão do Parque (CRUD)

1. Requisitos

1.1 Breve Descrição

O ParkManager tem a autorização para criar, ler, alterar e apagar lugares de estacionamento, users(perfis de plataformas e/ou serviços externos) e reservas(Reservation). Bem como a tabela de preços (Discount) e os estado do lugar de estacionamento (Status)

1.2 SSD



1.3 Descrição Completa

1.3.1 Ator Principal

ParkManager

1.3.2 Partes Interessadas

- **ParkManager** - Gestão dos seus serviços.

1.3.3 Pré-condições

- ParkManager tem de ser um UserRole com permissões de administrador.

1.3.4 Pós-condições

1.3.5 Cenário Principal de Sucesso

1. O ParkManager tem acesso aos métodos CRUD para as entidades.

1.3.6 Extensões (ou fluxos alternativos)

- O User não tem a autorização necessária para determinada ação (por exemplo, o user é uma plataforma externa e não o ParkManager quer adicionar um lugar de estacionamento.)
    - O sistema informa que não tem autorização para criar lugares de estacionamento
    - O caso de uso termina.

1.3.7 Requisitos Especiais

- Plataformas como o POSTMAN ou SAP para permitir a gestão do parque.

1.3.8 Variações de Tecnologias de dados

1.3.9 Frequência de ocorrência

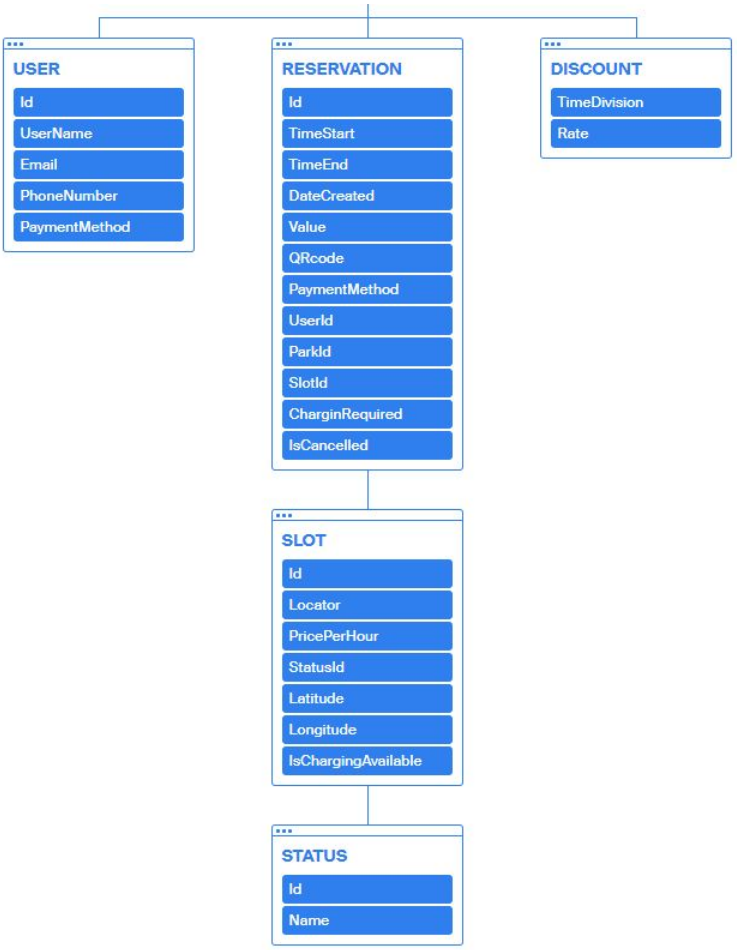
Diária.

1.3.10 Questões em Aberto

---

2. Análise OO

2.1. Modelo de Domínio relevante para o Caso de Uso



3. Design

3.1. Racional

Fluxo principal	Questão: Que classe...	Resposta	Justificação
ParkManager CRUD Slot	… coordena o UC?	SlotsController	Controller
ParkManager CRUD Status	… coordena o UC?	StatusController	Controller
ParkManager CRUD Reservation	… coordena o UC?	ReservationsController	Controller
ParkManager CRUD Discount	… coordena o UC?	DiscountsController	Controller
ParkManager CRUD User	… coordena o UC?	UserController	Controller

3.2. Sistematização

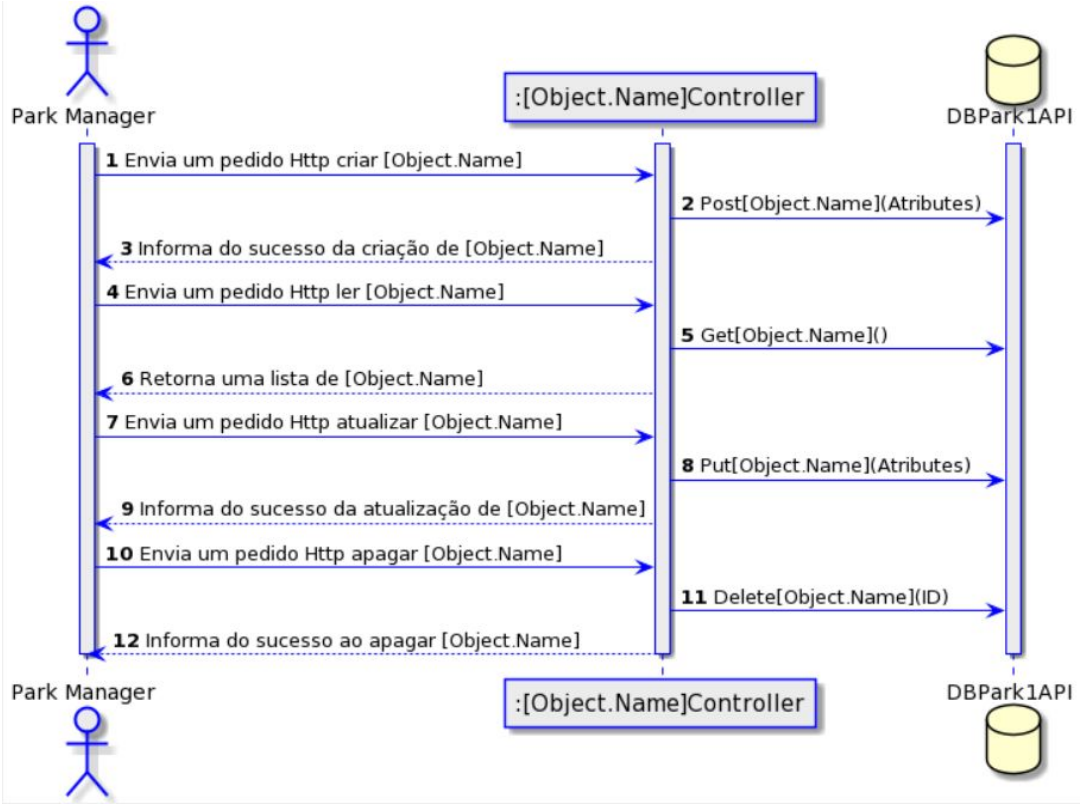
Do racional resulta que as classes conceptuais promovidas a classes de software são:

- Reservation
- Slot
- Status
- Reservation
- Discount
- User

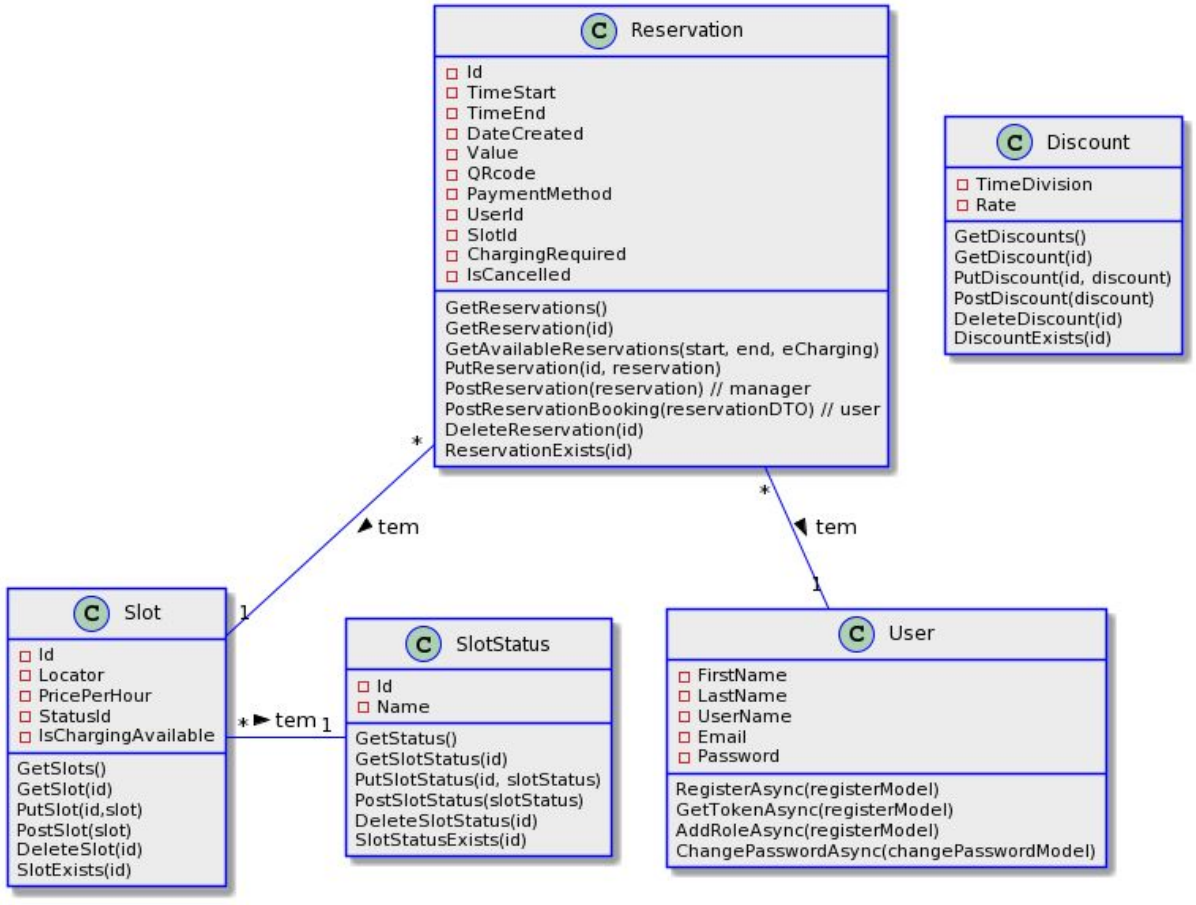
Outras classes de software identificadas:

- ReservationsController
- SlotsController
- StatusController
- ReservationsController
- DiscountsController
- UserController

3.3. Diagrama de Sequência



3.4. Diagrama de Classes



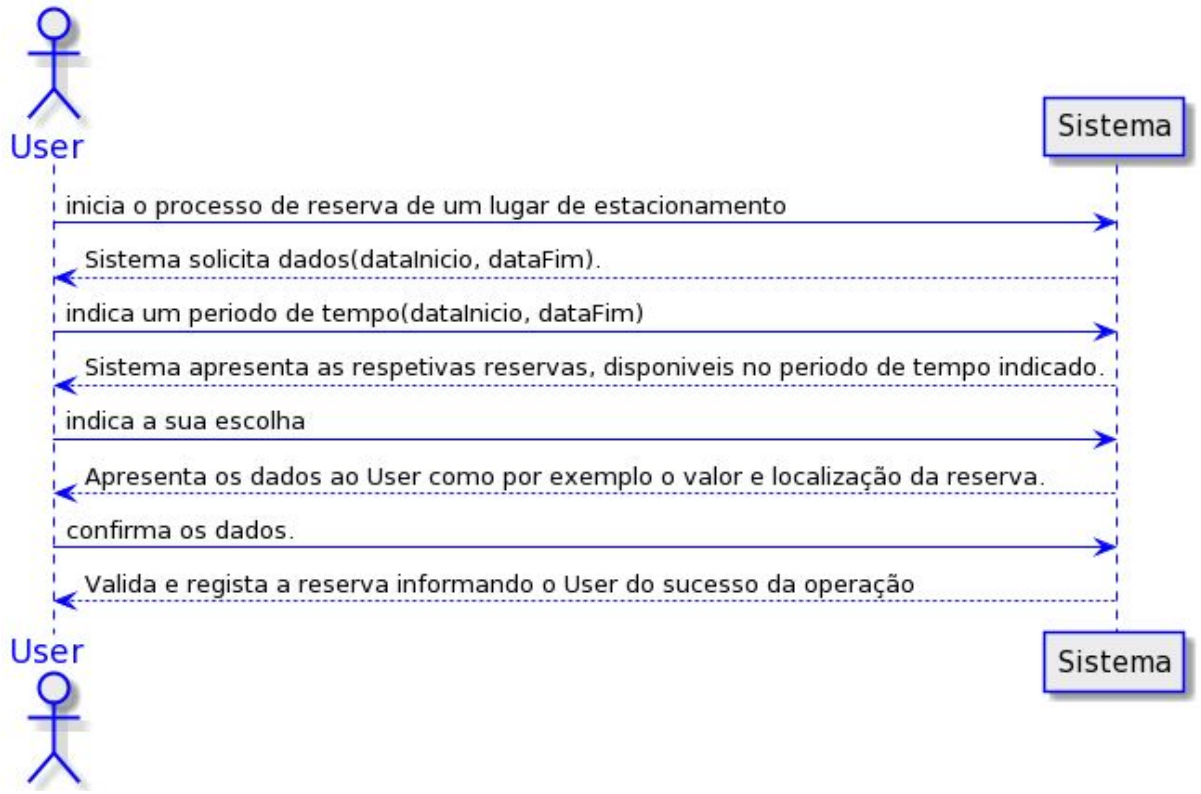
UC4 - User faz uma Reserva

1. Requisitos

1.1 Breve Descrição

O User(plataforma externa) inicia o processo de reserva de um lugar de estacionamento, por um determinado periodo de tempo. O Sistema apresenta os dados da reserva como por exemplo o valor a pagar. O User confirma que os dados estão corretos. O Sistema valida e regista a reserva informando o User do sucesso da operação.

1.2 SSD



1.3 Descrição Completa

1.3.1 Ator Principal

User(plataforma externa)

1.3.2 Partes Interessadas

- User: realiza uma reserva de um lugar
- Plataforma: fornece os seu serviço

1.3.3 Pré-condições

- O user já se encontra registrado

1.3.4 Pós-condições

1.3.5 Cenário Principal de Sucesso

O User realiza a reserva do seu lugar, paga a mesma e por fim usufruir do lugar

1.3.6 Extensões (ou fluxos alternativos)

- User desiste da reserva a meio de realizar a mesma
- O caso de uso termina.

1.3.7 Requisitos Especiais

1.3.8 Variações de Tecnologias de dados

1.3.9 Frequência de ocorrência

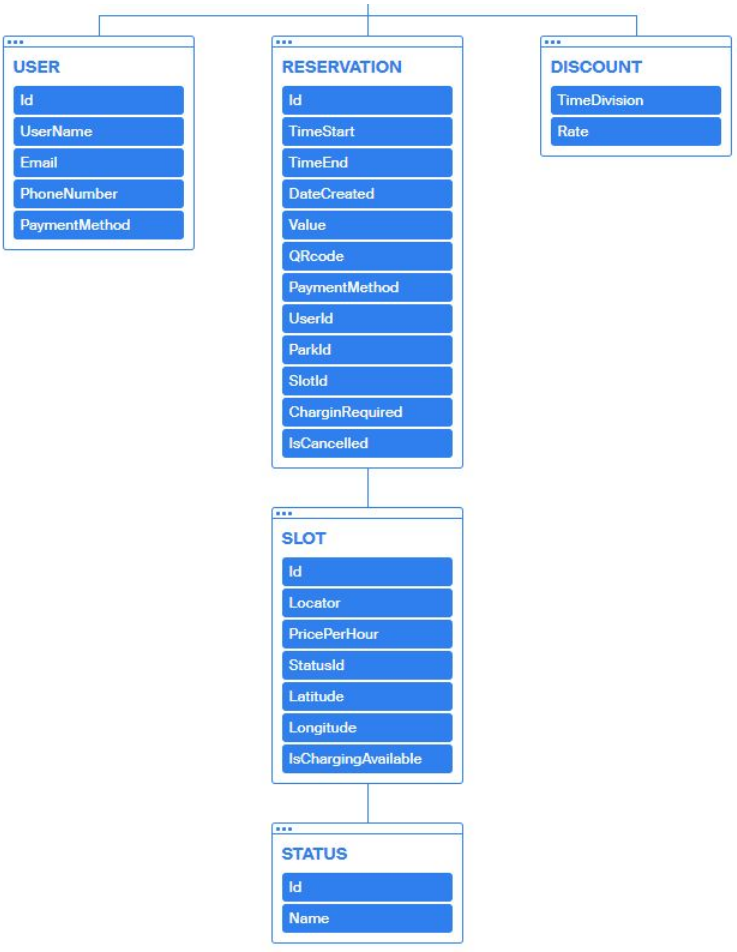
Constante

1.3.10 Questões em Aberto

---

2. Análise OO

2.1. Modelo de Domínio relevante para o Caso de Uso



3. Design

3.1. Racional

Fluxo principal	Questão: Que classe...	Resposta	Justificação
O User inicia o processo de reserva	... apresenta as reservas disponíveis	ReservationsController	Controller
O User escolhe o lugar	... cria a nova reserva	ReservationsController	Controller

3.2. Sistematização

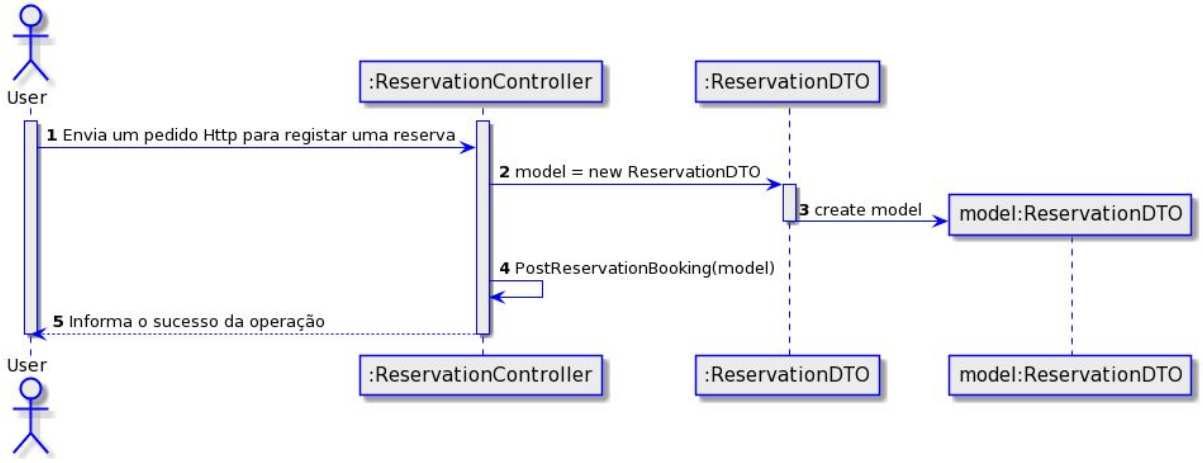
Do racional resulta que as classes conceituais promovidas a classes de software são:

- Reservation
- Slot
- Status
- Reservation
- Discount
- User

Outras classes de software identificadas:

- ReservationsController
- SlotsController
- StatusController
- ReservationsController
- DiscountsController
- UserController

3.3. Diagrama de Sequência



3.4. Diagrama de Classes

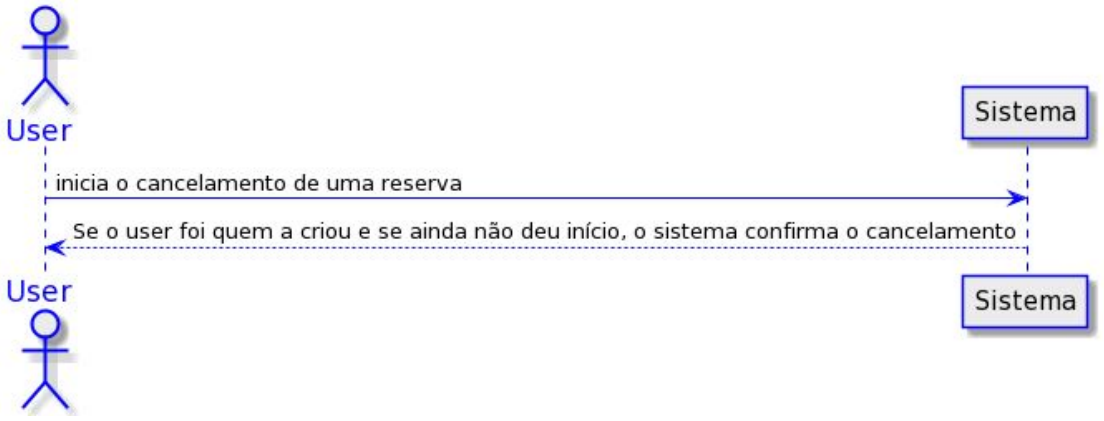
UC5 - User cancela Reserva

1. Requisitos

1.1 Breve Descrição

O User(Plataforma Externa, registada pelo ParkManager) faz um pedido so Sistema(API Private Park) para cancelar uma reserva. A API Private Park deverá validar se a reserva foi gerada pelo mesmo User que está a fazer o pedido de cancelamento e ainda, se o período de tempo da reserva ainda não iniciou.

1.2 SSD



1.3 Descrição Completa

1.3.1 Ator Principal

User(Plataforma Externa, registada pelo ParkManager)

1.3.2 Partes Interessadas

- **ParkManager** - Utilização dos seus serviços e atualização de disponibilidade para outros possíveis utilizadores do API do parque privado.
- **User(Plataforma Externa)** - Possibilidade de atender a pedidos de cancelamento.

1.3.3 Pré-condições

User tem de ser registado e validado pelo ParkManager.

1.3.4 Pós-condições

A Reserva tem de ser removida do sistema de forma a manter atualizada a disponibilidade dos lugares de estacionamento.

1.3.5 Cenário Principal de Sucesso

1. O User(Plataforma Externa) acede ao sistema e inicia um processo de cancelamento de reserva
2. O Sistema confirma se o User está autorizado (Se foi o criador da reserva e se esta ainda não iniciou) e confirma o sucesso do processo..

1.3.6 Extensões (ou fluxos alternativos)

- O User(Plataforma Externa) não é o criador e/ou o período permitido para p cancelamento foi ultrapassado.
    - (Não é o criador:) O sistema informa que o User não tem autorização para fazer aquele pedido.
    - O caso de uso termina.
    - (O período de cancelamento foi ultrapassado:) O sistema informa o User do razão da impossibilidade de proceder ao cancelamento da reserva.

        - O caso de uso termina.

- Durante o tempo de escolha, um dos lugares apresentados ficou reservado naquele horário escolhido pelo User(Plataforma Externa).
    - Ao selecionar a possível reserva, o Sistema terá de informar o User que a reserva escolhida foi reservada durante o processo.
    - O caso de uso termina.

1.3.7 Requisitos Especiais

- Devolução Monetária

1.3.8 Variações de Tecnologias de dados

- Método de pagamento; Devolução monetária.

1.3.9 Frequência de ocorrência

Possivelmente Diária.

1.3.10 Questões em Aberto

---

2. Análise OO

2.1. Modelo de Domínio relevante para o Caso de Uso

3. Design

3.1. Racional

Fluxo principal	Questão: Que classe...	Resposta	Justificação

O User(Plataforma Externa) acede ao sistema e inicia um processo de cancelamento de reserva	… interage com o User?	ReservationsController	Controller
	… coordena o UC?	ReservationsController	Controller(Delete)

3.2. Sistematização

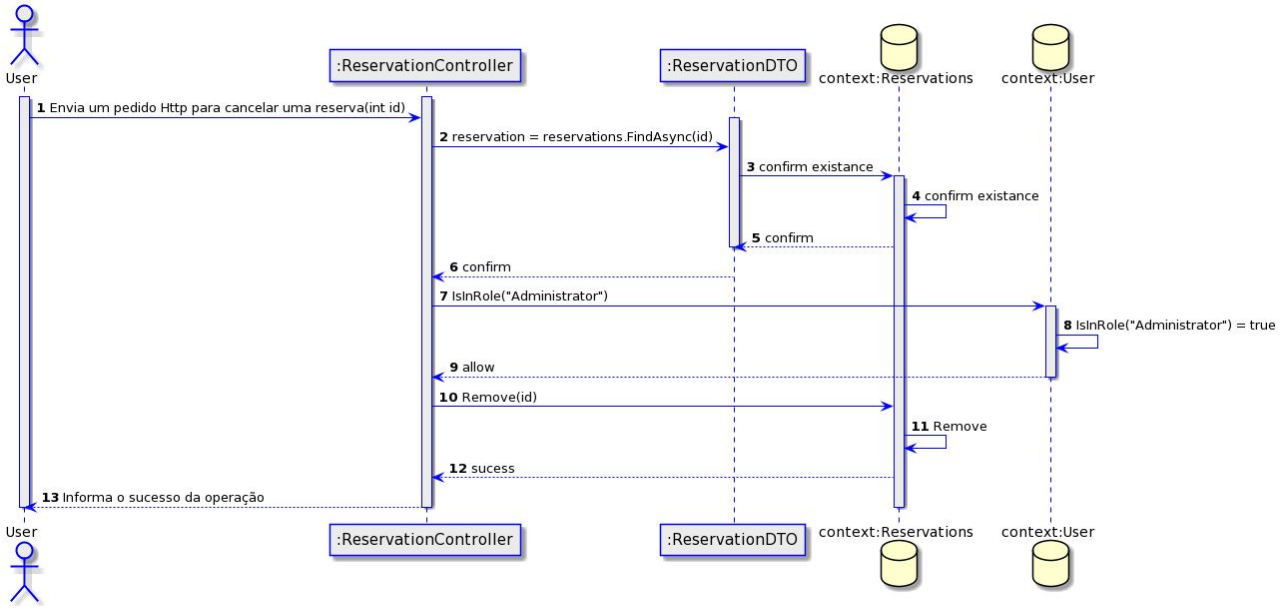
Do racional resulta que as classes conceptuais promovidas a classes de software são:

- Reservation

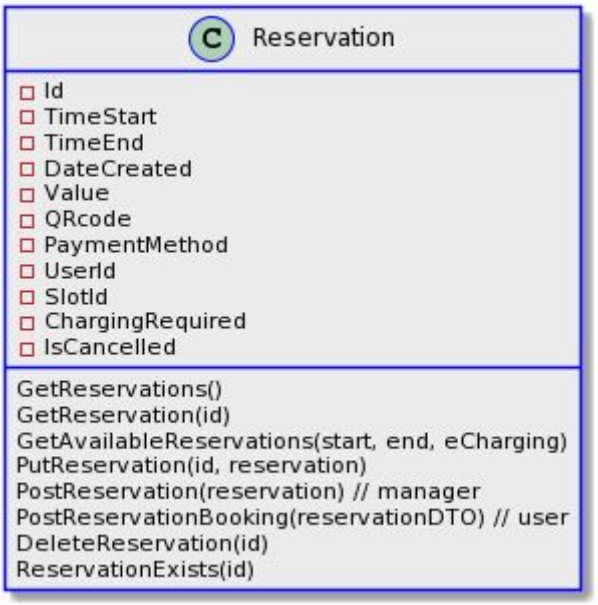
Outras classes de software identificadas:

- ReservationsController

3.3. Diagrama de Sequência



3.4. Diagrama de Classes



UC6 - User altera a sua password

1. Requisitos

1.1 Breve Descrição

O User(plataforma externa) inicia o processo para alterar a sua password. O sistema solicita o E-mail do User assim como a password atual e a nova. O Sistema valida os dados e depois informa o User do sucesso da operação.

1.2 SSD

1.3 Descrição Completa

1.3.1 Ator Principal

User(plataforma externa)

1.3.2 Partes Interessadas

- User: Alterar a sua password.

1.3.3 Pré-condições

- O user já se encontra registrado

1.3.4 Pós-condições

1.3.5 Cenário Principal de Sucesso

O User solicita uma alteração de password, fornecendo os dados necessários. Conclui com a password alterada.

1.3.6 Extensões (ou fluxos alternativos)

- User insere um E-mail não registado
    - O Sistema informa o User, e reinicia o processo.

- User não insere a password atual correctamente
    - O Sistema informa o User, e reinicia o processo.
- Password nova não preenche os pré-requisitos
    - O Sistema informa o User, e reinicia o processo.

1.3.7 Requisitos Especiais

1.3.8 Variações de Tecnologias de dados

1.3.9 Frequência de ocorrência

- Ocasional

1.3.10 Questões em Aberto

---

2. Análise OO

2.1. Modelo de Domínio relevante para o Caso de Uso

3. Design

3.1. Racional

Fluxo principal	Questão: Que classe...	Resposta	Justificação
O User inicia o processo de Alteração de password	... solicita os dados novos	UsersController	Controller
O User insere o e-mail, a password antiga e a nova.	... valida e procede à alteração	UsersController	Controller

3.2. Sistematização

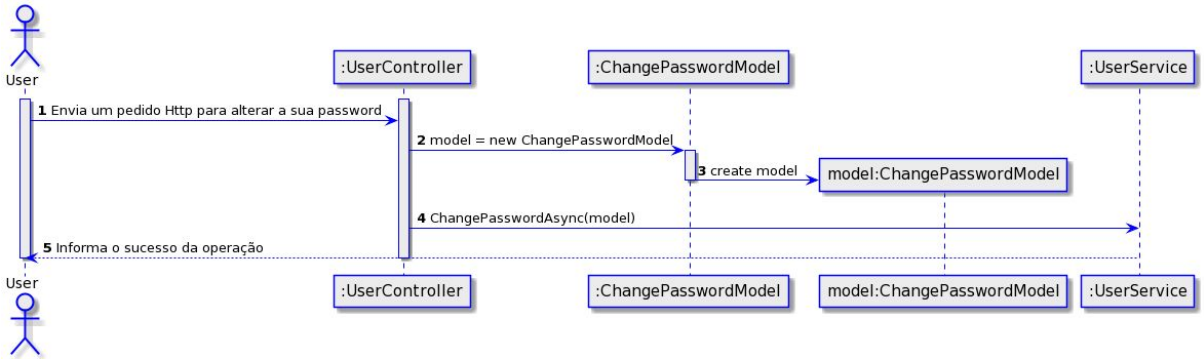
Do racional resulta que as classes conceituais promovidas a classes de software são:

- User

Outras classes de software identificadas:

- UserController

3.3. Diagrama de Sequência



3.4. Diagrama de Classes