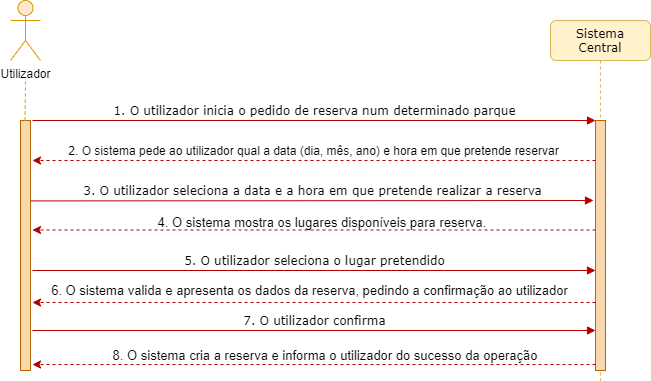
**UC 11 – Realizar pedido de reserva de determinado lugar (utilizador para API central)**

**Formato breve:**

O utilizador registado inicia o pedido de reserva num determinado parque. O sistema pede ao utilizador qual a data (dia, mês, ano) e hora em que pretende reservar. O utilizador seleciona a data e a hora em que pretende realizar a reserva. O sistema mostra os lugares disponíveis para reserva. O utilizador seleciona o lugar pretendido. O sistema valida e apresenta os dados da reserva, pedindo que o utilizador os confirme. O utilizador confirma. O sistema cria a reserva e informa o utilizador do sucesso da operação.



**Formato completo:**

**Ator primário:**

* Utilizador

**Partes interessadas e seus interesses:**

* Utilizador: pretende realizar uma reserva;
* Sistema Central: pretende registar uma reserva;

**Pré-condições:**

**-**

**Pós-condições:**

* A informação sobre uma reserva é atualizada no sistema.

**Cenário de sucesso principal (ou de fluxo básico):**

1. O utilizador registado inicia o pedido de reserva num determinado parque.
2. O sistema pede ao utilizador qual a data (dia, mês, ano) e hora em que pretende reservar.
3. O utilizador seleciona a data e a hora em que pretende realizar a reserva.
4. O sistema mostra os lugares disponíveis para reserva.
5. O utilizador seleciona o lugar pretendido.
6. O sistema valida e apresenta os dados da reserva, pedindo que o utilizador os confirme.
7. O utilizador confirma. O sistema cria a reserva e informa o utilizador do sucesso da operação.

**Extensões:**

\*a. O utilizador solicita o cancelamento da operação. O caso de uso termina.

3a. Dados mínimos obrigatórios em falta.

1. O sistema informa quais os dados em falta.

2. O sistema permite a introdução dos dados em falta (passo 3)

2a. O utilizador não altera os dados. O caso de uso termina.

3b. O sistema deteta que os dados (ou algum subconjunto dos dados) introduzidos devem ser únicos e que já existem no sistema.

1. O sistema alerta o utilizador para o facto.

2. O sistema permite a sua alteração.

2a. O utilizador não altera os dados. O caso de uso termina.

**Requisitos especiais:**

**-**

**Variações em tecnologias e dados:**

**-**

**Frequência de ocorrência:**

**-**

**Questões em aberto:**

**-**

**2. Análise OO**

**Excerto do Modelo de Domínio Relevante para o UC**

**3. Design – Realização do UC**

**Racional**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fluxo Principal** | **Questão:Que Classe…** | **Resposta** | **Justificação** |
| 1. |  |  |  |
| 2. |  |  |  |
| 3.. |  |  |  |
| 4.. |  |  |  |
| 5. |  |  |  |
| 6. |  |  |  |

**Sistematização**

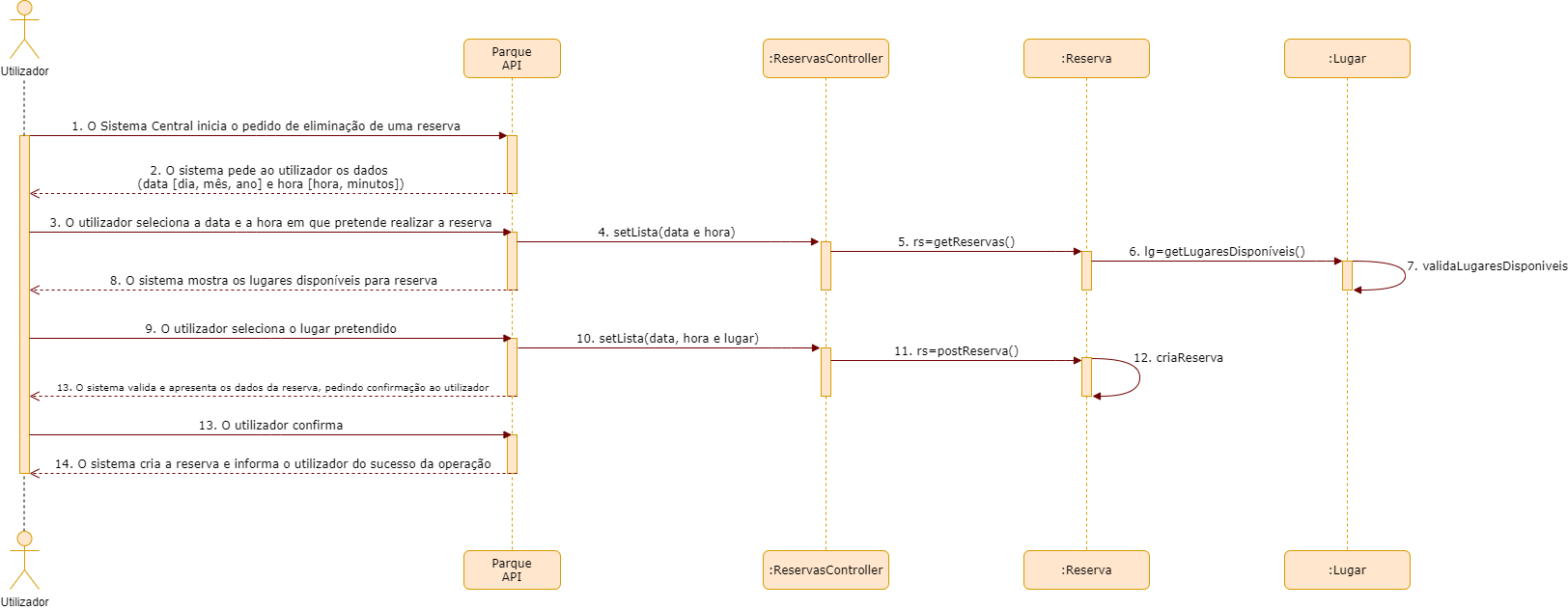
Do racional resulta que as classes conceptuais promovidas a classes de software são:

* Parque API
* Reservas
* Lugares
* Utilizador

Outras classes de software(i.e. Pure Fabrication) identificadas:

* ListaPessoasUI
* ListaPessoasController

**Diagrama de Sequência**



**Diagrama de Classes**