

## **Licenciatura em Informática**

- **Análise e Conceção de Sistemas de Informação -**
  - **Bases de Dados II -**
  - **Tecnologias de Internet -**

**SiGREGs - Sistema de Gestão de Reserva de Espaços**

## **Documento de Projeto - Trabalho de Grupo**

**Realizado por:**  
**Hugo Maçoas, N° 150118003**  
**João Madeira, N°**  
**150118036**  
**Miguel Tomás, N°**  
**150118037**

Santarém, 16 de janeiro de 2017

## Índice

Introdução .....	3
Descrição do sistema .....	4
Principais funções do sistema .....	9
Diagrama casos de utilização .....	10
Descrição dos casos de utilização .....	11
Diagrama de classes .....	18
Modelo/esquema logico de dados .....	19
Interfaces do Utilizador.....	20
Diagrama de sequência .....	27
Diagramas de comunicação .....	32
Diagrama de estados .....	38
Diagrama de atividades .....	39
Implementação de subprogramas armazenados .....	40
Conclusões .....	54
Bibliografia .....	55

## Introdução

Neste trabalho temos como objetivo desenvolver um sistema de reservas de espaços para a escola superior de gestão, que permita ao administrador adicionar e remover utilizadores, ao responsável de espaço adicionar uma reserva, eliminar uma reserva, listar as reservas ou adicionar/remover/alterar espaços, o professor poderá adicionar uma reserva, eliminar uma reserva e listar as reservas já o funcionário só poderá listar as reservas, caso os utilizadores estejam devidamente identificados. Para o administrador aceder à plataforma deve-se autenticar no sistema. Desta maneira deverá ser muito mais rápida e cómoda para o utilizador. Para efetuar este projeto irá ser necessário utilizar os conhecimentos aprendidos nas disciplinas de Base de Dados 2, Tecnologias da Internet e Análise e Conceção de Sistemas de Informação e combiná-las de maneira a funcionarem como uma só. O motor de Base de Dados a utilizar é o Oracle 11g Express Edition.

## **Descrição do Sistema**

A Escola Superior de Gestão e Tecnologia do Instituto Politécnico de Santarém necessita de um sistema de informação para controlar a disponibilidade de salas de aula e outros espaços. Até agora, o processo de reserva de sala, para as diversas atividades, tem sido feito manualmente: registada em agendas pessoais ou calendários de parede. Como podemos imaginar, há vários problemas decorrentes desses métodos. Essencialmente, torna-se muito complicado, para quem precisa de uma sala, saber se ela está disponível para uma atividade, numa data e hora específicas.

Inicialmente, tentando evitar esse problema, a direção da escola ordenou que o horário de ocupação da sala fosse afixado perto da porta da mesma. No entanto, outras questões foram despertadas: As atualizações regulares para os horários dos cursos, torna a informação rapidamente desatualizada. Então, quando alguém está à entrada da sala, a olhar para o calendário publicado lá, não sabe exatamente se a sala está livre ou não... Segundo o calendário publicado junto da sala ela poderia estar disponível naquele momento, mas alguém pode ter feito uma reserva dessa mesma sala, a iniciar dali por 5 minutos, junto de uma funcionária. Este tipo de informação é registado apenas numa agenda pessoal para os serviços administrativos da escola.

Se um professor quer saber se uma sala específica estará disponível daqui por 2 meses, é impossível sem falar com a pessoa responsável pela tarefa de gerir salas. Outro problema vem das horas de trabalho... Se alguém precisa dessas informações após as 18 horas, os serviços não estão disponíveis.

### **A NECESSIDADE**

---

Assim, a necessidade deste sistema é apoiada em três pontos principais:

- 1) Integração - o sistema deve centralizar todas as informações sobre as reservas;
- 2) Disponibilidade - o sistema deve estar sempre disponível;
- 3) Segurança - o sistema deve controlar quem é concedida para fazer as diferentes operações e registrar quem fez as operações.

#### **1) Integração**

Os dados fundamentais deste novo sistema serão os horários dos cursos regulares, e em geral todos os cursos, na escola. Cursos como Gestão de Empresas, Contabilidade e Finanças, Marketing e Publicidade, Informática, Negócios Internacionais e Erasmus têm os seus próprios horários que ocupam as salas na maioria das vezes. Os cursos de Mestrado e os TeSP também ocupam regularmente as salas.

As outras reservas são muito específicas e são feitas por professores para, eventualmente, ter uma reunião com os alunos, fazer um exame ou uma prova escrita, uma atividade de projeto, ou outro tipo de atividades que requeiram uma sala.

Os horários são produzidos pela Comissão de Horários e publicado antes do início de cada semestre. Após o início do semestre, pode haver atualizações para os horários dos cursos e haverá outra publicação, sempre que for necessário.

O sistema não permite a reserva no passado, ou seja, não podemos registar uma reserva com a data de ontem. Os Utilizadores podem inserir reservas para o ano/semestre letivo seguinte. O utilizador (professor ou funcionário) pode registar a reserva de uma sala para um período específico de tempo. Haverá uma pessoa responsável para cada sala e, se ocorrer a reserva de sala por outro utilizador, entretanto, a pessoa responsável será sempre informada para poder analisar e aprovar. O sistema vai garantir a comunicação necessária entre os utilizadores interessados nas reservas e as pessoas responsáveis pelas salas.

O endereço de e-mail dos utilizadores deve ser registado no sistema, pois toda a comunicação será enviada por aí. Na maioria das operações, o Sistema tem que enviar mensagens de e-mail pré-definidos para os usuários. O texto destas mensagens de e-mail é padrão e específico para o tipo de operação que o gerar. Algumas das mensagens pode incluir links para algumas das funcionalidades do sistema (confirmação da reserva, e assim por diante).

## **2) Disponibilidade**

Dado que o sistema deve estar disponível todos os dias, todas as horas, em qualquer lugar, a Internet foi o meio ambiente selecionado.

## **3) Segurança**

Por motivos de segurança, vários tipos de utilizadores serão considerados. Cada um dos quais tem um conjunto específico de funcionalidade para usar. O processo de autenticação será assegurado pelo próprio sistema. Esta funcionalidade é para ser considerada. As credenciais dos novos utilizadores devem ser carregadas no sistema por um administrador.

## **OS UTILIZADORES**

---

Os principais utilizadores do sistema são, em termos funcionais de negócio: o Professor, o Funcionário do piso e o Responsável de sala. Em termos funcionais técnicos, temos a considerar um Administrador do sistema que gere as credenciais dos restantes.

## O PROFESSOR

Em termos práticos, resumidamente, um Professor pode realizar 3 operações em torno das reservas de sala: Adicionar, Consultar e Remover. Pode ainda fazer pesquisas sobre as características e ocupação das salas.

**A) Adicionar nova reserva** - O registo de uma reserva pressupõe (A1) a escolha da sala e (A2) a definição do *timing* da reserva.

**A1)** Na escolha da sala, o Professor pode optar por seleccionar directamente a sala que pretende ou pode recorrer à pesquisa de sala disponível. Neste último caso, ao procurar por uma sala, o professor pode começar por indicar uma data e uma hora e, em seguida, o sistema mostra todas as salas livres (incluindo breve descrição das características). A descrição das aulas na lista será (SL1: 20 computadores, 30 lugares, e outro tipo de recursos - videoprojector, etc.).

Caso contrário, o professor pode escolher a data e hora e ainda pode indicar qual/ais o(s) tipo(s) de recursos deve(m) existir nessa sala (número de computadores, projetor de vídeo, número de lugares).

A lista de salas é apresentada com todas as salas disponíveis (e com as funcionalidades necessárias).

Em seguida, o professor selecciona apenas uma sala da lista e clica no botão de reserva.

Caso o professor já saiba a sala que pretenda, pode indicar directamente qual é a sala e escolher para reservar.

**A2)** Na definição dos tempos das reservas, o professor pode optar por usar uma de três vias:

- **para um intervalo** - escolhe o dia, a sala, e define a hora de início e de fim da reserva;
- **por período** - escolhe o dia, a sala e o período: Manhã - das 8:30 até 13:30; Tarde - de 13:30 até 18:30, ou; Noite - a partir de 18:30 até 23:50. O sistema deve considerar automaticamente a hora de início e de fim);
- **e baseado num Padrão** - baseado num padrão repetitivo - uma sala, vários dias, um intervalo de horas e a regra de repetição do evento:

Independentemente da forma como o professor seleccionou a sala podem ocorrer duas situações:

- 1) (L) se a sala estiver livre para o período definido, o sistema irá enviar automaticamente uma notificação por e-mail para o professor e para o responsável de sala, no momento em que o professor clica no botão da reserva. Esta mensagem "*A sua reserva foi gravada. Por favor, aguarde a mensagem de confirmação*" será enviada para o professor e uma outra mensagem, contendo um "Link Confirmação", será enviada para o responsável da sala. Em termos de sistema, a confirmação é feita pelo responsável quando clicar no link existente na mensagem (gerada aquando da reserva) que recebeu. Se isso acontecer, o processo fica concluído e o professor receberá a mensagem de confirmação: "*A sua reserva foi confirmada*". Se o responsável não concorda, outra mensagem será enviada (responsável pode propor uma sala diferente), com um texto diferente: "*A sua reserva não foi confirmada. Reservámos a sala xpto como uma alternativa*", ou "*Sua reserva não foi confirmada e nenhum espaço foi proposto. Entre em contato com os Serviços*". Neste último caso, uma sala nova será negociada fora do contexto do sistema e um dos utilizadores (professor ou responsável de sala) fará a reserva de novo.
  
- 2) (O) se a sala estiver ocupada, o sistema não permitirá a reserva de imediato e pode dar-se o caso de ter de ocorrer uma negociação (dentro e/ou fora do sistema). Por exemplo: o professor 1 regista no sistema uma reserva para a sala 1.3 para quarta-feira, 2 de novembro de 2016, 17:00-18:00. Posteriormente, o professor 2 entra no sistema e tenta reservar a mesma sala (1.3), no mesmo dia (quarta-feira, 2 de novembro de 2016) para as 17:30-18:30. Então, uma mensagem será mostrada ao professor 2: "*Esta sala, para esta data, está reservada pelo professor 1*" e há dois botões: "Continuar" e "Cancelar". Se o Professor 2 clicar em Cancelar, não há problema, pois assume-se que desistiu da reserva. Mas, se acionar o botão "Continuar", o sistema envia e-mail ao responsável da sala informando da reserva em sobreposição. O responsável da sala terá de obter uma autorização do Professor 1 para realizar o cancelamento da reserva inicial. Se o professor 1 concordar, o responsável de sala exclui a reserva e confirma reserva do Professor 2. O Sistema envia e-mail com informações tanto para os professores (a reserva é excluída para o professor 1 e a reserva é confirmada para o professor 2) como para o responsável da sala. O responsável de sala, ou o próprio professor 1, pode posteriormente fazer uma nova reserva.

**B) Listar Reservas** - O Professor pode ver as suas próprias reservas e as respetivas descrições.

**C) Excluir Reservas** - O Professor pode excluir as suas próprias reservas a partir de lista de reserva. Sistema envia e-mail a informar o professor e o responsável da sala.

#### O RESPONSÁVEL DE SALA

O Responsável de sala pode realizar várias operações em torno das reservas de sala: Adicionar, Consultar, Remover e Confirmar reservas de outros (desde que sejam das salas que monitoriza). Pode ainda fazer

pesquisas sobre as características e ocupação de todas as salas. Ao responsável de sala cabe ainda gerir as próprias salas, ou seja, criar novas salas (fica responsável automaticamente) ou inativar as salas (sob sua responsabilidade) que vierem a ficar inoperacionais.

**A) Adicionar nova reserva** - O processo é em tudo idêntico ao processo de reserva feito pelo professor, com a pequena diferença de que a reserva pode não ficar em nome do responsável de sala, mas sim no nome de outra pessoa (um docente, ou outra entidade - professor, funcionário, entidade externa, etc.). Se se aperceber de que a reserva apresenta conflitos com outras reservas, o responsável deverá “negociar”, fora do sistema, uma solução com as partes envolvidas. Só nesse momento é que registará no sistema a solução encontrada.

**B) Confirmar reservas** - Aquando de uma reserva, o Responsável tem a última palavra e deve confirmar a reserva através do link que consta do e-mail que recebeu a dar conta da reserva.

**C) Listar Reservas** - o Responsável pode ver uma lista de reserva para todas as salas. Responsável pode filtrar por professor e por sala.

**D) Excluir Reservas** - o Responsável pode excluir Reservas, sendo o professor (ou quem tenha realizado a reserva) notificado dessa ocorrência por e-mail.

**E) Gerir Salas** - o Responsável pode inserir novas salas para o sistema e configurar as instalações das salas novas ou já existentes.

#### **O FUNCIONÁRIO**

**A) Lista Reservas** - Os funcionários da escola podem ver lista de reservas. O filtro pode ser apenas por sala e data.



## **Principais Funções do Sistema**

O sistema em análise tem

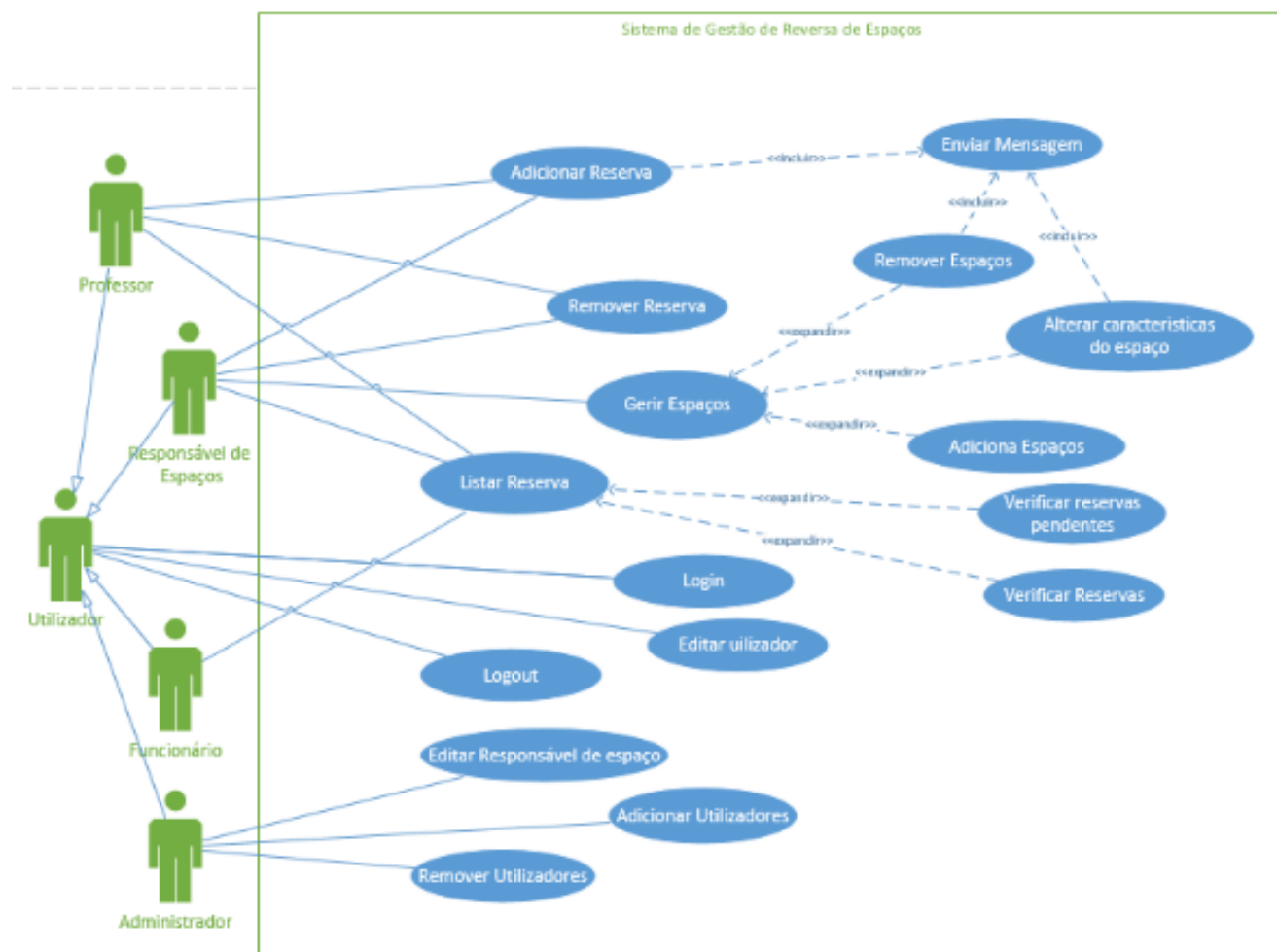
... os seguintes atores:

- Administradores (inserir novos utilizadores, remove utilizadores, editar responsável de sala);
- Professor (reservar salas, eliminar reservas, listar reservas);
- Responsável de sala (reservar salas, eliminar reservas, listar reservas, gerir espaços);
- Funcionário (Listar reservas);

... as seguintes funções:

- 1º Adicionar reservas de salas de aula;
- 2º Listar reservas de salas de aula;
- 3º Eliminar reservas de salas de aula;
- 4º Adicionar utilizadores;
- 5º Remover utilizadores;

## Diagrama de Casos de Utilização



## Descrição dos Casos de Utilização

1 - Nome	Login
2 - Atores	Utilizador
3 - Objetivos	Validar Utilizador
4 - Ecrãs	SGRE_L_01, SGRE_L_02, SGRE_L_03, SGRE_L_04, SGRE_L_05, SGRE_L_06
5- Pré-condições	O programa tem de estar iniciado
<b>6- Cenário principal</b> 1 – O utilizador (professor, responsável de espaço, funcionário, administrador de sistema) tem de aceder ao sistema; 2 – O sistema mostra um ecrã de validação onde tem de colocar a suas credenciais (utilizador, password) [SGRE_L_01]; 3 – O utilizador insere as credenciais e faz OK; 4 – O sistema verifica as credenciais e apresenta o ecrã [SGRE_L_02, SGRE_L_03, SGRE_L_04, SGRE_L_05] adequado ao tipo de utilizador (professor, responsável de espaço, funcionário, administrador de sistema);	
<b>6.1- Cenário alternativo - Erro pessoal</b> 4 – Caso as credenciais não existam, o utilizador é informado [SGRE_L_06] e é dada nova possibilidade de entrar no sistema;	
7- Observações	À terceira tentativa de entrada sem que a password esteja correta, o utilizador será bloqueado e permanecerá assim durante 1 hora;
8- Pós condições	Sessão iniciada e no respetivo ecrã.

1 - Nome	Logout
2 - Atores	Utilizador
3 - Objetivos	Desativar utilizadores
4- Ecrãs	SGRE LO 01
5- Pré-condições	O utilizador tem que ter sessão iniciada previamente
<b>6- Cenário principal</b> 1 - O utilizador aciona a função Logout 2 - O sistema remove toda a informação da sessão; 3 - O sistema mostra ecrã de fecho de sessão [SGRE_LO_01];	
7- Observações	
8- Pós condições	O utilizador fica com a sessão terminada;

1 - Nome	Listar reserva
2 - Atores	Professor, Responsável de espaços, funcionário
3 - Objetivos	Mostrar reservas
4- Ecrãs	SGRE_LR_01
5- Pré-condições	O ator tem que ter sessão iniciada previamente
<b>6- Cenário principal</b> 1 - O ator aciona a opção de listar reservas; 2 - O sistema mostra a lista de reservas [SGRE_LR_01], indicando as que estão pendentes e as que estão reservadas (professor, responsável de sala);	
7- Observações	
8- Pós condições	O ator fica com as reservas listadas;

1 - Nome	Adicionar espaços
2 - Atores	Responsável de espaço
3 - Objetivos	Adicionar novo espaço à base de dados do sistema
4- Ecrãs	SGRE_AE_01, SGRE_AE_02, SGRE_AE_03
5- Pré-condições	O Responsável de espaço tem que ter sessão iniciada previamente
6- Cenário principal	<p>1 – O responsável de espaço escolhe a opção adicionar novo espaço;</p> <p>2 - O sistema mostra o ecrã de adicionar[ SGRE_AE_01]</p> <p>3 - O responsável de espaço insere os dados do novo espaço e faz OK;</p> <p>4 - O sistema verifica se este espaço já existe;</p> <p>4.1 – Caso o espaço não exista o novo espaço é registado no sistema [SGRE_AE_02];</p> <p>4.2 – Caso o espaço já se encontre registado, o responsável de espaço é informado [SGRE_AE_03] e é dada nova possibilidade de introduzir os dados;</p>
7- Observações	
8- Pós condições	O responsável de espaço adicionou um espaço;

1 - Nome	Remover espaços
2 - Atores	Responsável de Espaço
3 - Objetivos	Remover espaço à base de dados do sistema
4- Ecrãs	SGRE_RE_01, SGRE_RE_02
5- Pré-condições	O Responsável de Espaço tem que ter sessão iniciada previamente
6- Cenário principal	<p>1 - O responsável de espaço escolhe a opção remover espaço;</p> <p>2 - O sistema mostra o ecrã de remover espaços [SGRE_RE_01]</p> <p>3 - O responsável de espaço seleciona o espaço que quer remover e faz OK [SGRE_RE_02];</p>
7- Observações	
8- Pós condições	O responsável de espaço removeu um espaço;

1 - Nome	Adicionar utilizadores
2 - Atores	Administrador
3 - Objetivos	Adicionar novo Utilizador à base de dados do sistema
4- Ecrãs	SGRE_AU_01, SGRE_AU_02, SGRE_AU_03
5- Pré-condições	O Administrador tem que ter sessão iniciada previamente
6- Cenário principal	<p>1 - O administrador escolhe a opção adicionar novo utilizador;</p> <p>2 - O sistema mostra o ecrã de adicionar utilizador [SGRE_AU_01]</p> <p>3 - O Administrador insere os dados do novo utilizador e faz OK;</p> <p>4 - O sistema verifica se este utilizador já existe;</p> <p>4.1 – Caso o utilizador não exista um novo utilizador é criado no sistema [SGRE_AU_02];</p> <p>4.2 – Caso o utilizador já se encontre registado, o administrador é informado [SGRE_AU_03] e é dada nova possibilidade de introduzir os dados;</p>
7- Observações	
8- Pós condições	O Administrador adicionou um novo utilizador;

1 - Nome	Remover utilizador
2 - Atores	Administrador
3 - Objetivos	Remover utilizador da base de dados do sistema
4- Ecrãs	SGRE_RU_01, SGRE_RU_02
5- Pré-condições	O Administrador tem que ter sessão iniciada previamente
6- Cenário principal	<p>1 - O administrador escolhe a opção remover utilizador;</p> <p>2 - O sistema mostra o ecrã de remover utilizador [SGRE_RU_01]</p> <p>3 - O Administrador seleciona o utilizador que quer remover e faz OK [SGRE_RU_02];</p>
7- Observações	
8- Pós condições	O Administrador removeu um utilizador;

1 - Nome	Editar utilizador
2 - Atores	Utilizador
3 - Objetivos	Editar utilizador da base de dados do sistema
4- Ecrãs	SGRE_EU_01, SGRE_EU_02
5- Pré-condições	O utilizador tem que ter sessão iniciada previamente
6- Cenário principal	1 - O utilizador escolhe a opção editar utilizador; 2 - O sistema mostra o ecrã de editar utilizador [SGRE_EU_01] 3 - O utilizador seleciona o utilizador que quer editar e faz OK [SGRE_EU_02];
7- Observações	
8- Pós condições	O utilizador editou a sua conta;

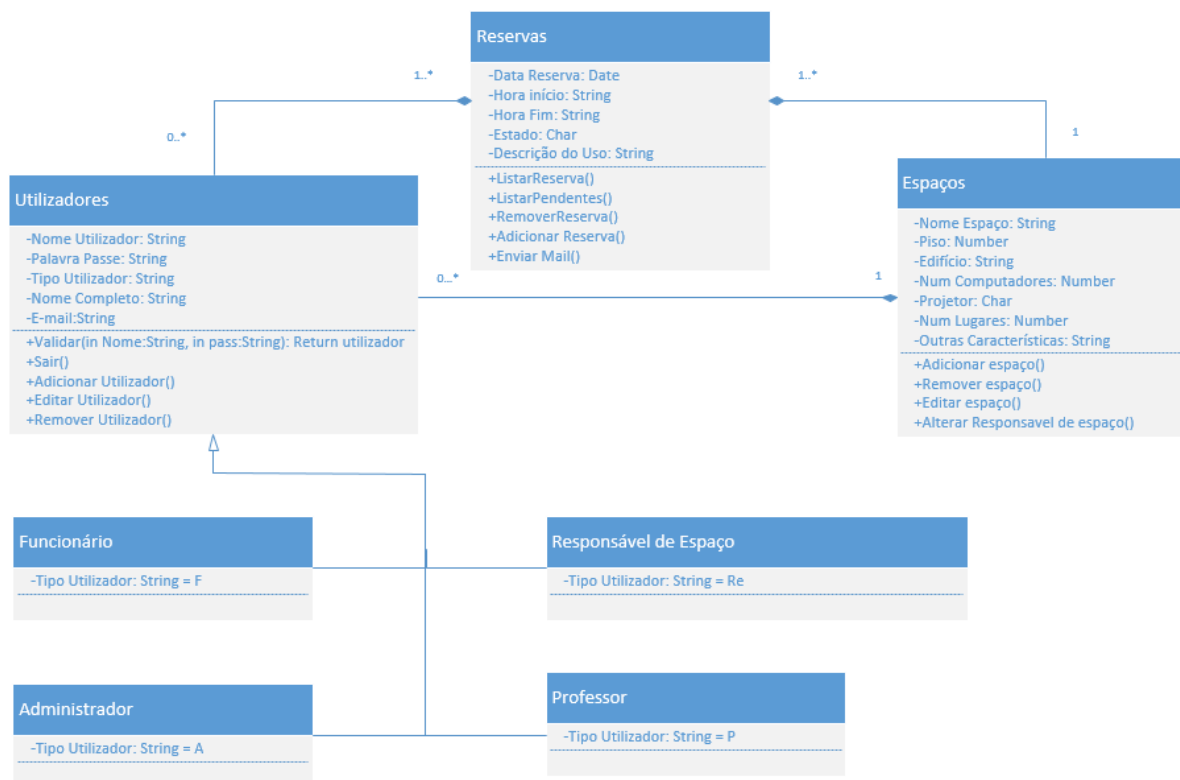
1 - Nome	Remover reserva
2 - Atores	Responsável de espaço, Professor
3 - Objetivos	Remover reserva de espaço
4- Ecrãs	SGRE_RR_01
5- Pré-condições	Os atores referidos no segundo ponto têm que ter sessão previamente iniciada.
6- Cenário principal	1 - O ator escolhe a opção remover reserva; 2 - O sistema mostra o ecrã de remover reserva [SGRE_RR_01] 3 - O ator seleciona a reserva que quer remover/cancelar e faz OK; 3.1 – Se acionar o botão OK, o sistema remove a reserva e envia uma notificação de e-mail para o
7- Observações	
8- Pós condições	Um dos atores removeu uma reserva;

1 - <b>Nome</b>	Adicionar reserva
2 - <b>Atores</b>	Responsável de espaço, Professor
3 - <b>Objetivos</b>	Adicionar reserva de espaço
4- <b>Ecrãs</b>	SGRE_AR_01
5- <b>Pré-condições</b>	Os atores referidos no segundo ponto têm que ter sessão previamente iniciada.
<b>6- Cenário principal</b> 1 - O ator escolhe a opção adicionar reserva; 2 - O sistema mostra o ecrã de adicionar reserva [SGRE_AR_01] 3 - O ator seleciona a reserva que quer adicionar e faz OK; 3.1 – Se acionar o botão OK, o sistema remove a reserva e envia uma notificação de e-mail para o	
7- <b>Observações</b>	
8- <b>Pós condições</b>	Um dos atores adicionou uma reserva;



1 - Nome	Gerir Espaço
2 - Atores	Responsável de espaço
3 - Objetivos	Gerir Espaço
4- Ecrãs	SGRE_GE_01, SGRE_GE_02, SGRE_GE_03
5- Pré-condições	O ator referido no segundo ponto tem que ter sessão previamente iniciada.
<b>6- Cenário principal</b> 1 - O ator escolhe a opção gerir espaços; 2 - O ator deve indicar uma das opções (adicionar espaço, remover espaço, alterar características): <ul style="list-style-type: none"> <li>• Caso escolha a opção adicionar espaço o ator preenche o formulário que lhe é apresentado e faz OK;</li> <li>• Caso escolha a opção remover espaço o ator indica qual o espaço que quer remover e faz OK;</li> <li>• Caso escolha a opção alterar características o ator seleciona o que quer alterar e faz OK.</li> </ul> 3 - O sistema mostra o ecrã correspondente à opção escolhida [SGRE_GE_01, SGRE_GE_02, SGRE_GE_03]; 4 - O ator seleciona a reserva que quer remover/cancelar e faz OK; 3.1 – Se acionar o botão OK, o sistema remove a reserva e envia uma notificação de e-mail para o	
7- Observações	
8- Pós condições	O ator adiciona, remove ou altera características do espaço;

## Diagrama de Classes



## Modelo/Esquema Lógico de Dados

- **Utilizadores**

(id\_utilizador, nome\_utilizador, palavra\_passe, tipo\_utilizador, nome\_completo, email)

- **Espaços**

(id\_espaco, nome, piso, edificio, n\_Computadores, projetor, n\_lugares, outras\_caracteristicas, id\_utilizador)

- **Reservas**

(id\_Reservas, data\_reserva, hora\_inicio, hora\_fim, estado, descrição\_uso, id\_espaco, id\_utilizador)

## Interfaces do Utilizador

SGRE\_L\_01: Página inicial



SGRE\_L\_02: Ecrã principal Professor



### SGRE\_L\_03: Ecrã principal Responsável de Espaço



### SGRE\_L\_04: Ecrã principal Funcionário

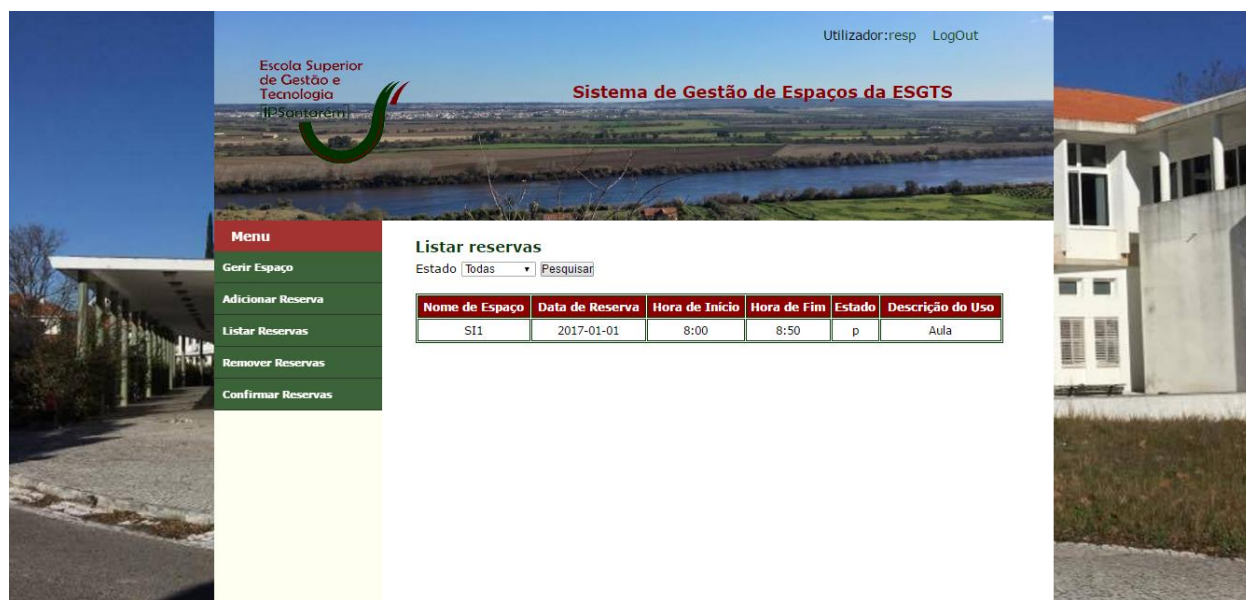


### SGRE\_L\_05: Ecrã principal Administrador





## SGRE\_LR\_01: Ecrã Listar Reservas



## SGRE\_AE\_01: Ecrã Adicionar Espaços

Utilizador: resp Logout

Escola Superior de Gestão e Tecnologia [IP Santarém]

**Sistema de Gestão de Espaços da ESGTS**

**Menu**

- Gerir Espaço
- Adicionar Reserva
- Listar Reservas
- Remover Reservas
- Confirmar Reservas

**Adicionar Espaço**

Nome de Espaço:

Piso:

Edifício:

Número de Computadores:

Projetor:

Número de lugares:

Outras Características:

## SGRE\_RE\_01: Ecrã Remover Espaços

Utilizador: resp Logout

Escola Superior de Gestão e Tecnologia [IP Santarém]

**Sistema de Gestão de Espaços da ESGTS**

**Menu**

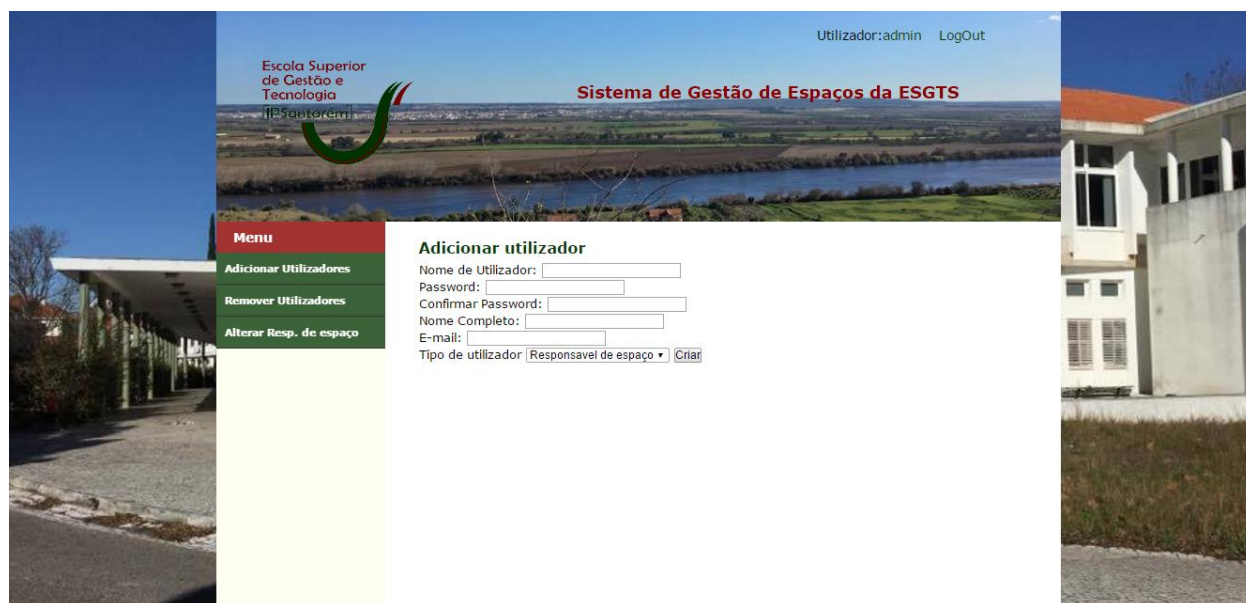
- Gerir Espaço
- Adicionar Reserva
- Listar Reservas
- Remover Reservas
- Confirmar Reservas

**Remover Espaço**

Edifício:  Piso:

Nome do Espaço	Nº Computadores	Projetor	Nº Lugares	Outras Características	Editar Espaço
1.2	0	s	58	Não tem	<a href="#">Remover</a>
1.3	0	s	54	Não tem	<a href="#">Remover</a>
1.4	0	s	94	Não tem	<a href="#">Remover</a>
1.5	0	s	38	Não tem	<a href="#">Remover</a>
1.6	0	s	37	Não tem	<a href="#">Remover</a>
1.1	0	s	58	Não tem	<a href="#">Remover</a>

## SGRE\_AU\_01: Ecrã Adicionar Utilizadores



Utilizador: admin LogOut

Escola Superior de Gestão e Tecnologia [IP Santarém]

**Sistema de Gestão de Espaços da ESGTS**

**Menu**

- Adicionar Utilizadores
- Remover Utilizadores
- Alterar Resp. de espaço

**Adicionar utilizador**

Nome de Utilizador:

Password:

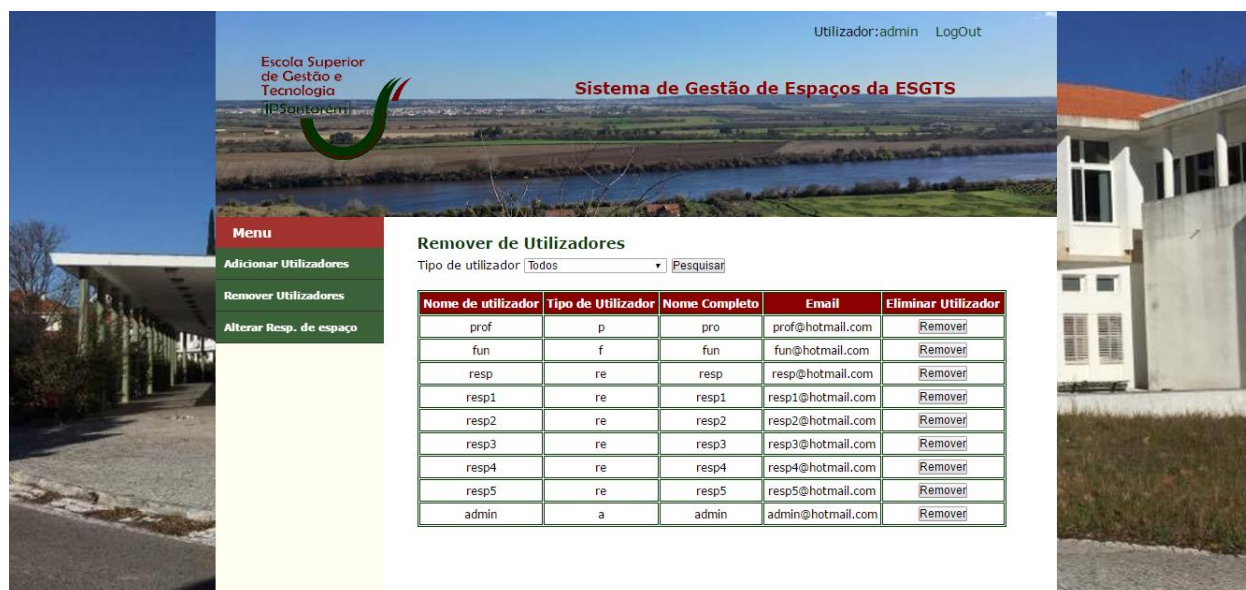
Confirmar Password:

Nome Completo:

E-mail:

Tipo de utilizador: Responsável de espaço

## SGRE\_RU\_01: Ecrã Remover Utilizadores



Utilizador: admin LogOut

Escola Superior de Gestão e Tecnologia [IP Santarém]

**Sistema de Gestão de Espaços da ESGTS**

**Menu**

- Adicionar Utilizadores
- Remover Utilizadores
- Alterar Resp. de espaço

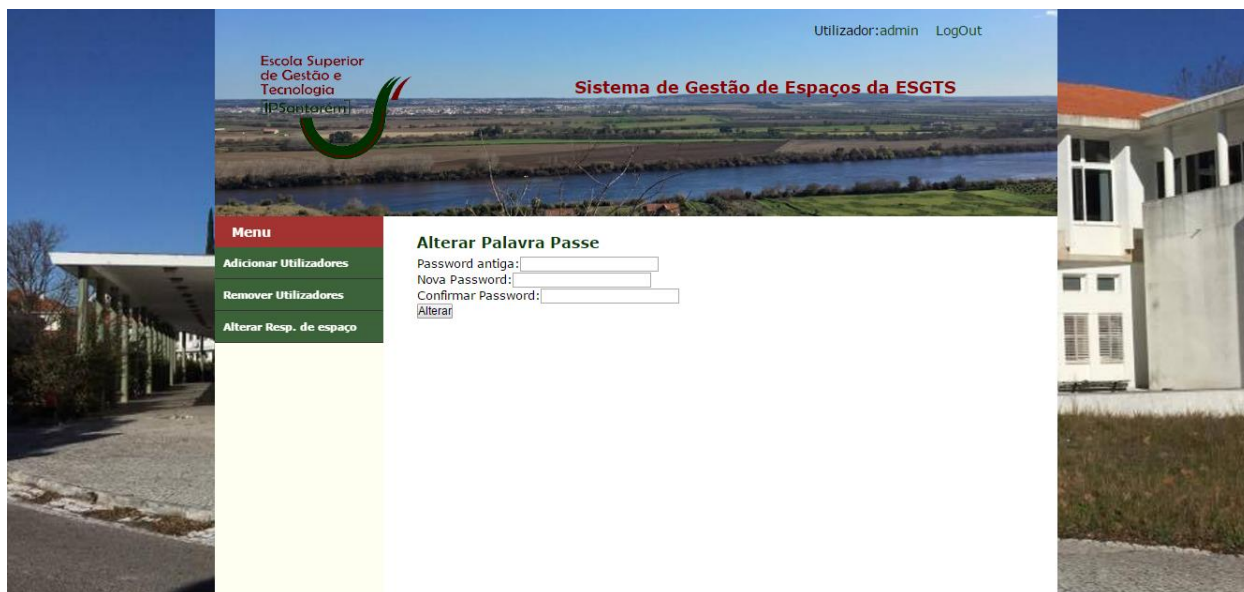
**Remover de Utilizadores**

Tipo de utilizador: Todos

Nome de utilizador	Tipo de Utilizador	Nome Completo	Email	Eliminar Utilizador
prof	p	pro	prof@hotmail.com	<a href="#">Remover</a>
fun	f	fun	fun@hotmail.com	<a href="#">Remover</a>
resp	re	resp	resp@hotmail.com	<a href="#">Remover</a>
resp1	re	resp1	resp1@hotmail.com	<a href="#">Remover</a>
resp2	re	resp2	resp2@hotmail.com	<a href="#">Remover</a>
resp3	re	resp3	resp3@hotmail.com	<a href="#">Remover</a>
resp4	re	resp4	resp4@hotmail.com	<a href="#">Remover</a>
resp5	re	resp5	resp5@hotmail.com	<a href="#">Remover</a>
admin	a	admin	admin@hotmail.com	<a href="#">Remover</a>



## SGRE\_EU\_01: Ecrã Editar Utilizadores



Utilizador:admin LogOut

Escola Superior de Gestão e Tecnologia [IP Santarém]

**Sistema de Gestão de Espaços da ESGTS**

**Menu**

- Adicionar Utilizadores
- Remover Utilizadores
- Alterar Resp. de espaço

**Alterar Palavra Passe**

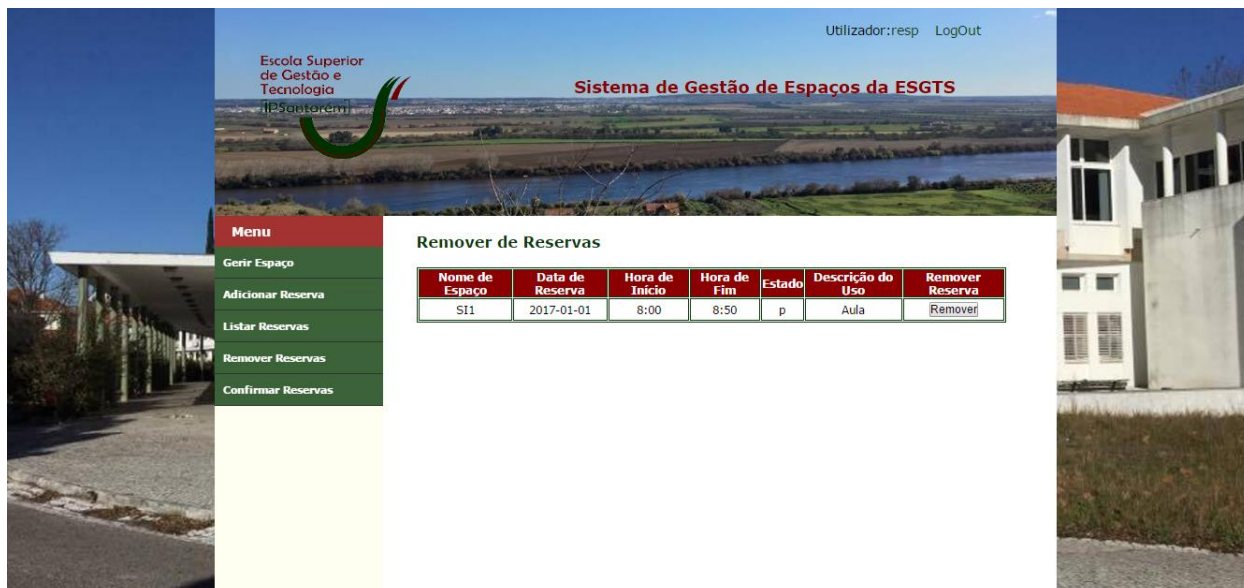
Password antiga:

Nova Password:

Confirmar Password:

[Alterar](#)

## SGRE\_RR\_01: Ecrã Remover Reserva



Utilizador:resp LogOut

Escola Superior de Gestão e Tecnologia [IP Santarém]

**Sistema de Gestão de Espaços da ESGTS**

**Menu**

- Gerir Espaço
- Adicionar Reserva
- Listar Reservas
- Remover Reservas
- Confirmar Reservas

**Remover de Reservas**

Nome de Espaço	Data de Reserva	Hora de Início	Hora de Fim	Estado	Descrição do Uso	Remover Reserva
SI1	2017-01-01	8:00	8:50	p	Aula	<a href="#">Remover</a>

## SGRE\_AR\_01: Ecrã Adicionar Reserva

Utilizador: prof LogOut

Escola Superior de Gestão e Tecnologia IP Santarém

**Sistema de Gestão de Espaços da ESGTS**

**Menu**

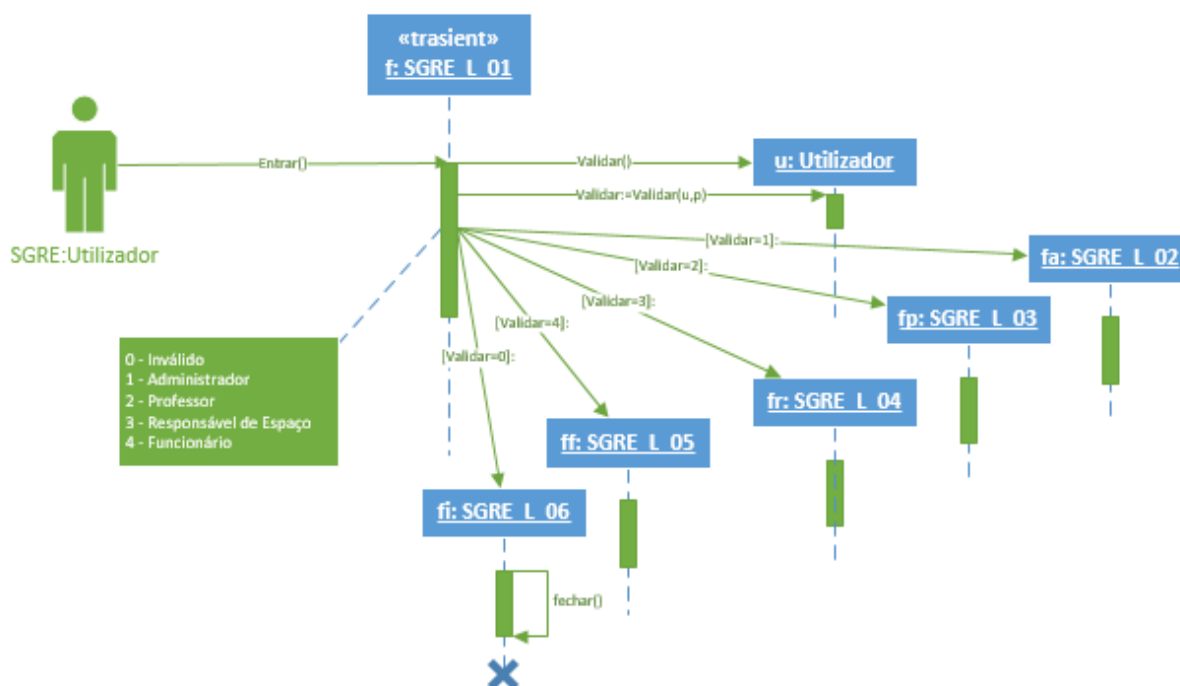
- Adicionar Reservas
- Remover Reservas
- Listar Reservas

**Pesquisar Espaço a Reservar**

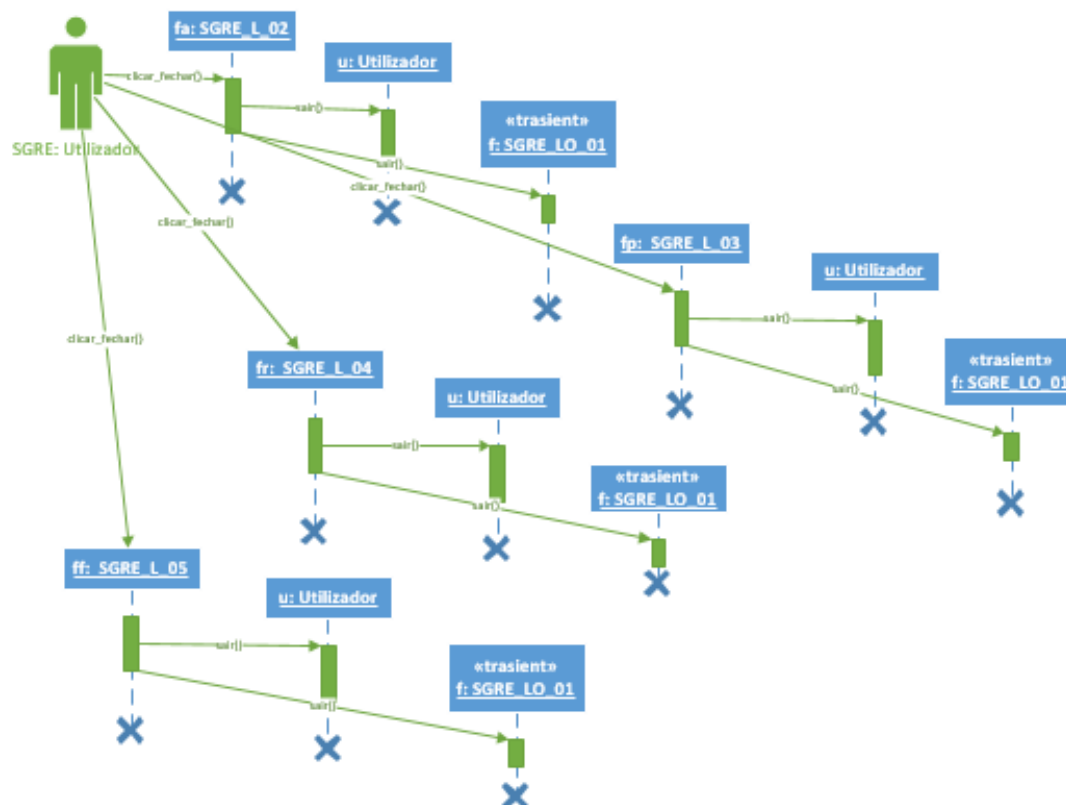
Edifício: [Edifício Principal ▼]  
Piso: [Piso 0 ▼]  
Número de Computadores: 0  
Projetor: [Sim ▼]  
Número de lugares: 10  
Pesquisar

## Diagrama de Sequência

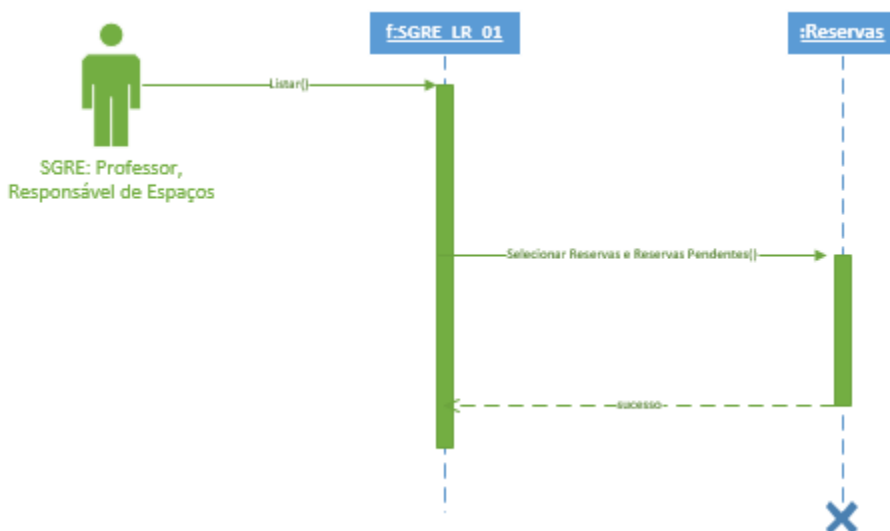
### Login



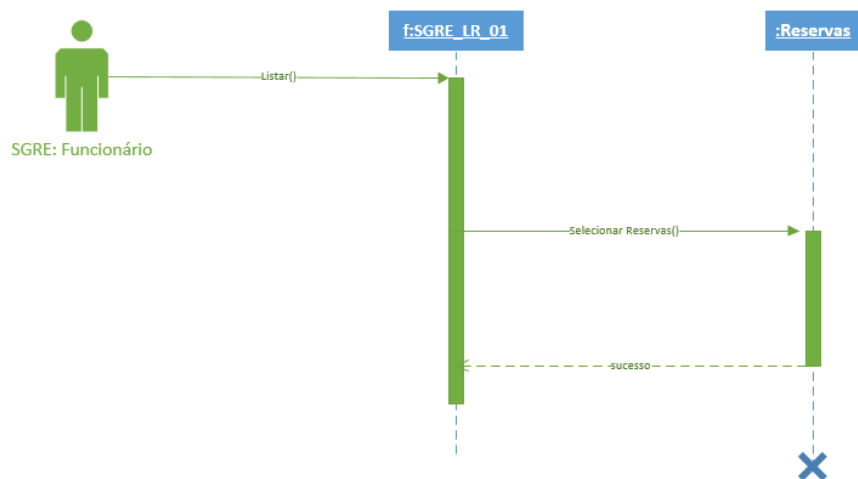
### Logout



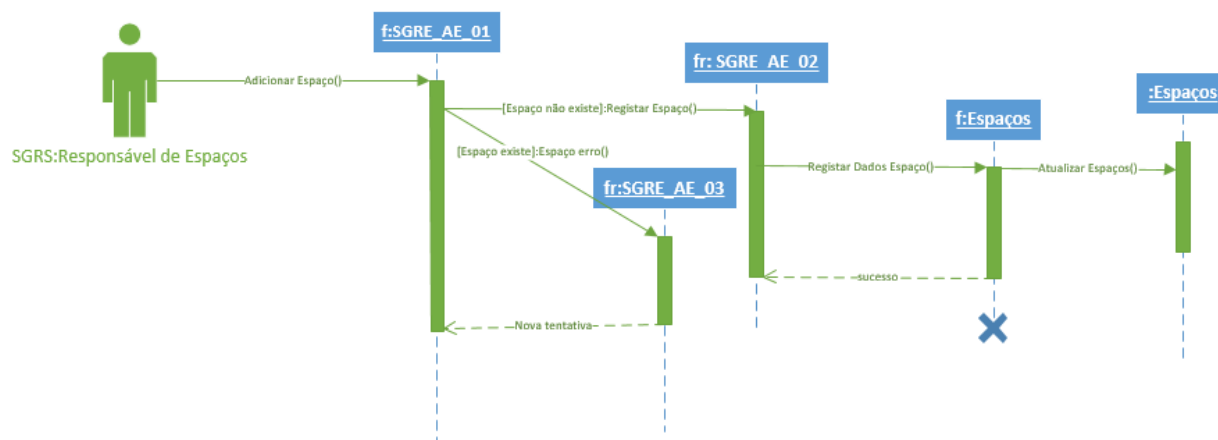
## Listar reserva (Professor, Responsável de Espaço)



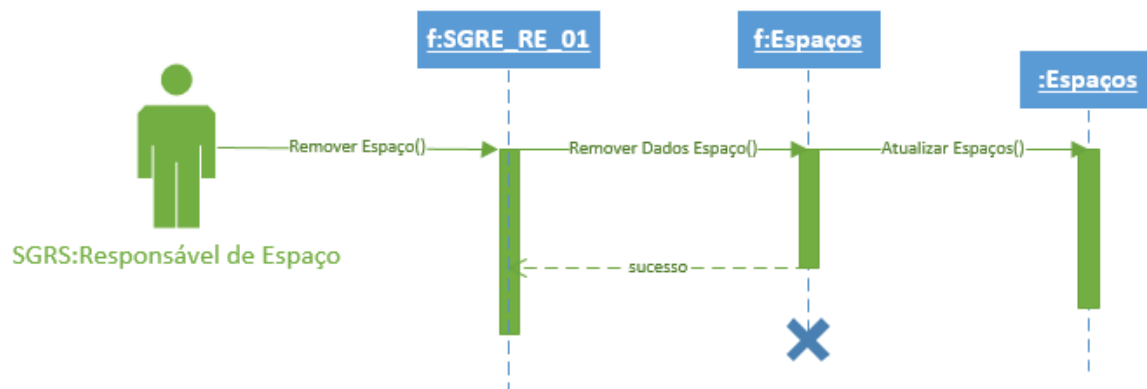
## Listar reserva (Funcionário)



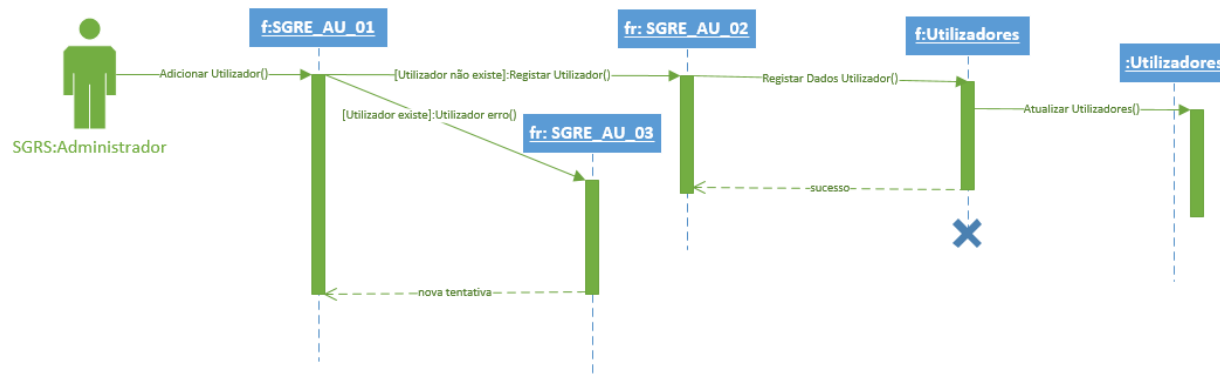
## Adicionar espaço



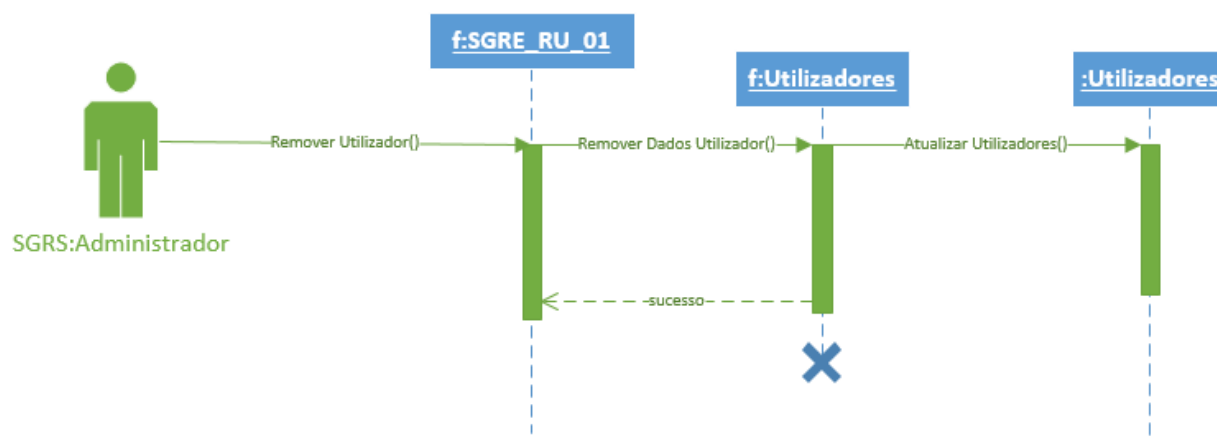
## Remover espaço



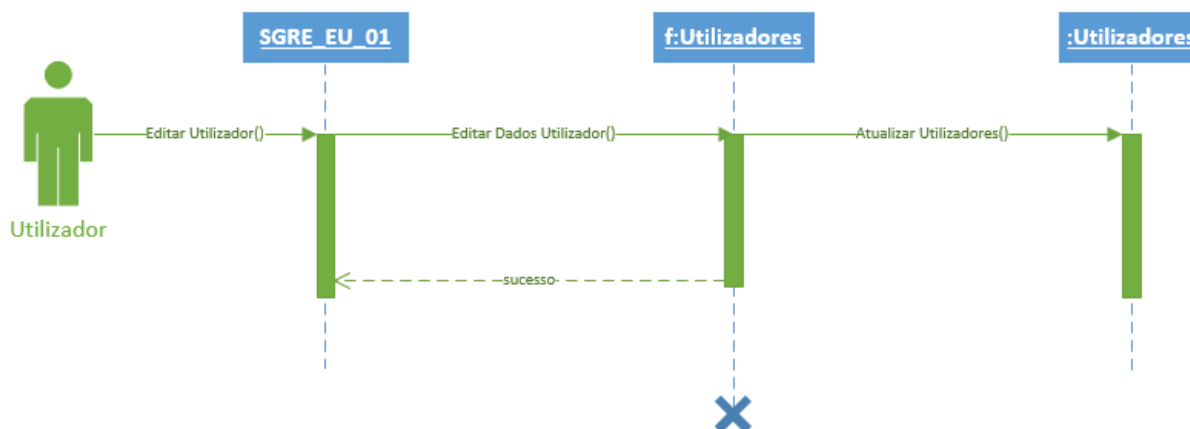
## Adicionar utilizador



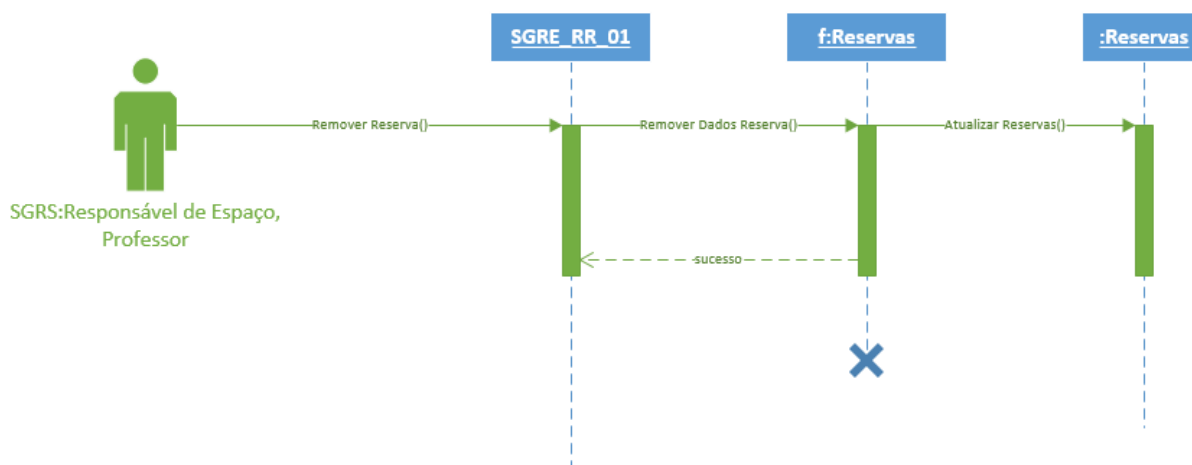
## Remover utilizador



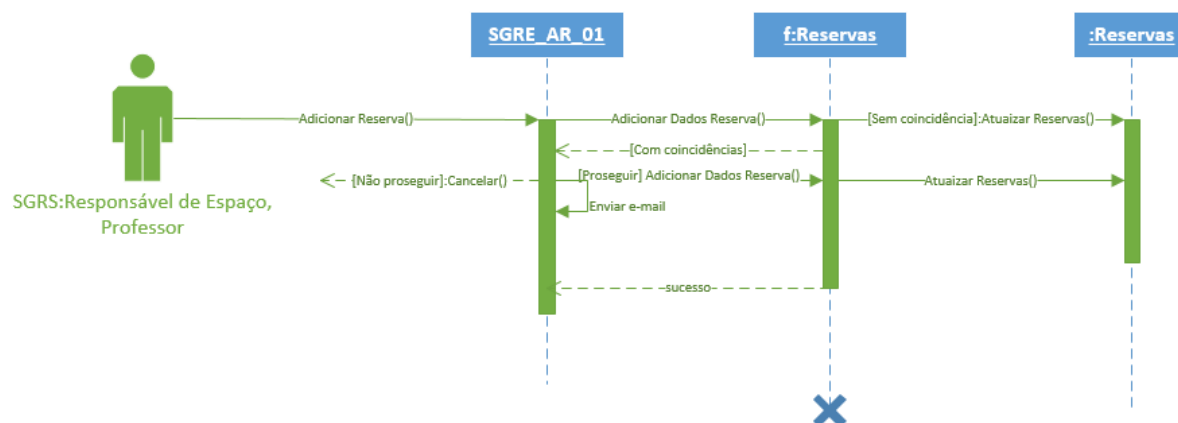
## Editar utilizador



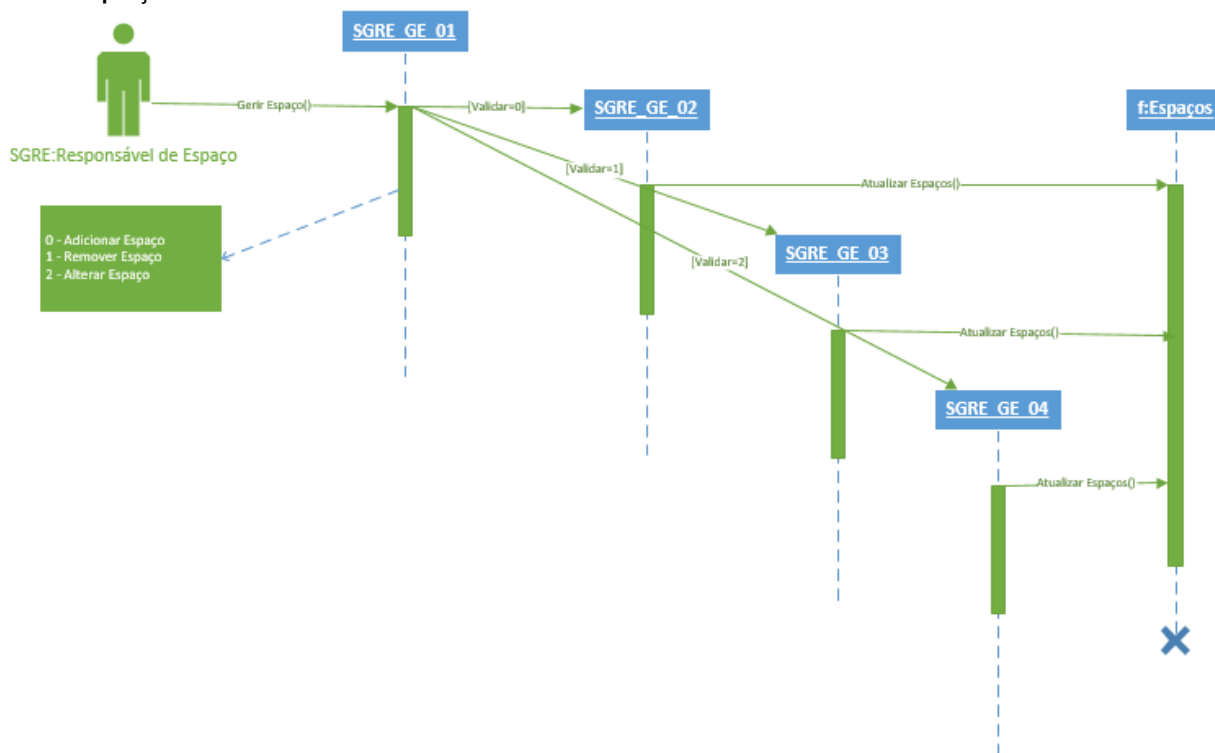
## Remover reserva



## Adicionar reserva



## Gerir espaços

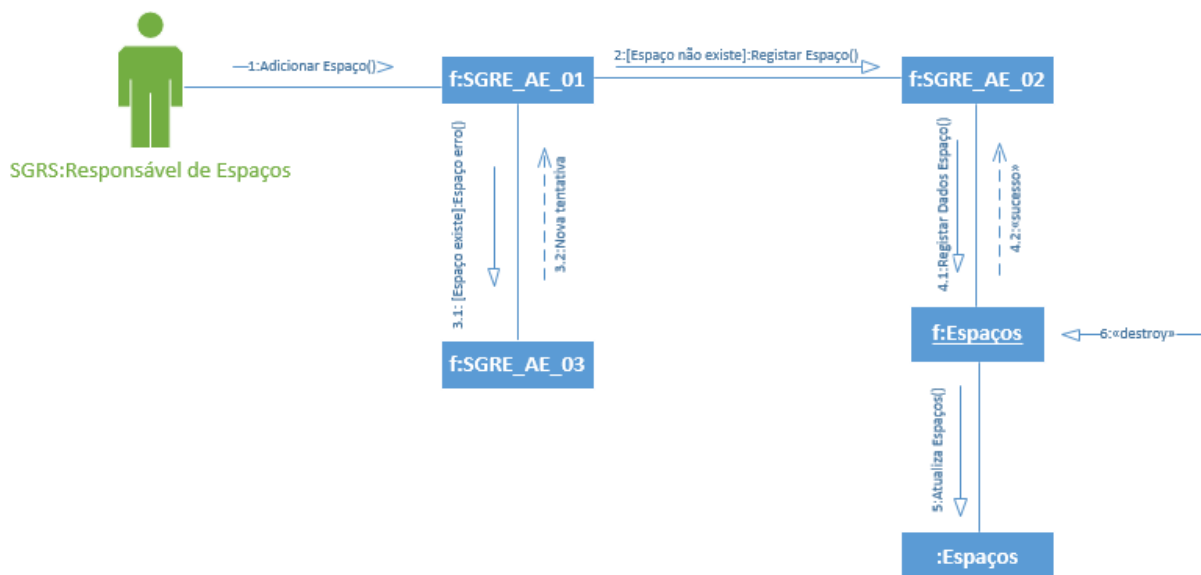


## Diagrama de Comunicação

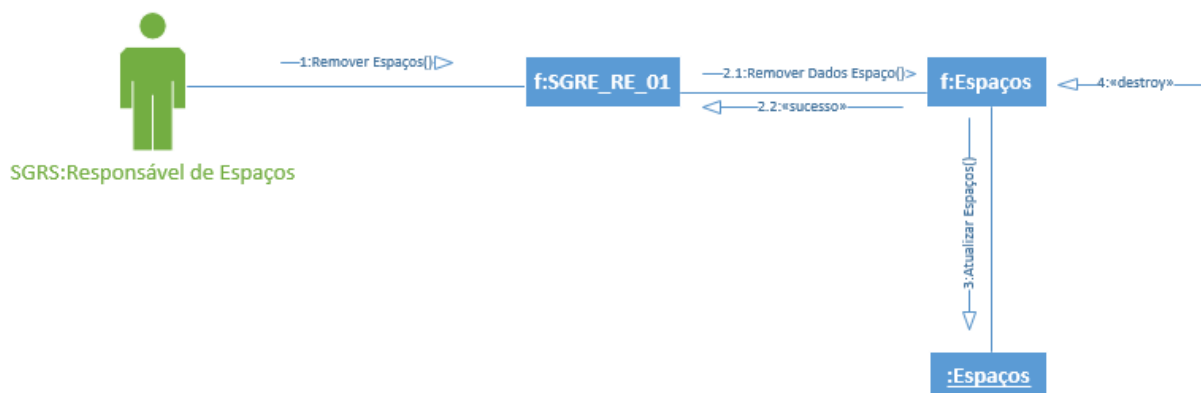
### Listar reserva (Funcionário)



### Adicionar espaço

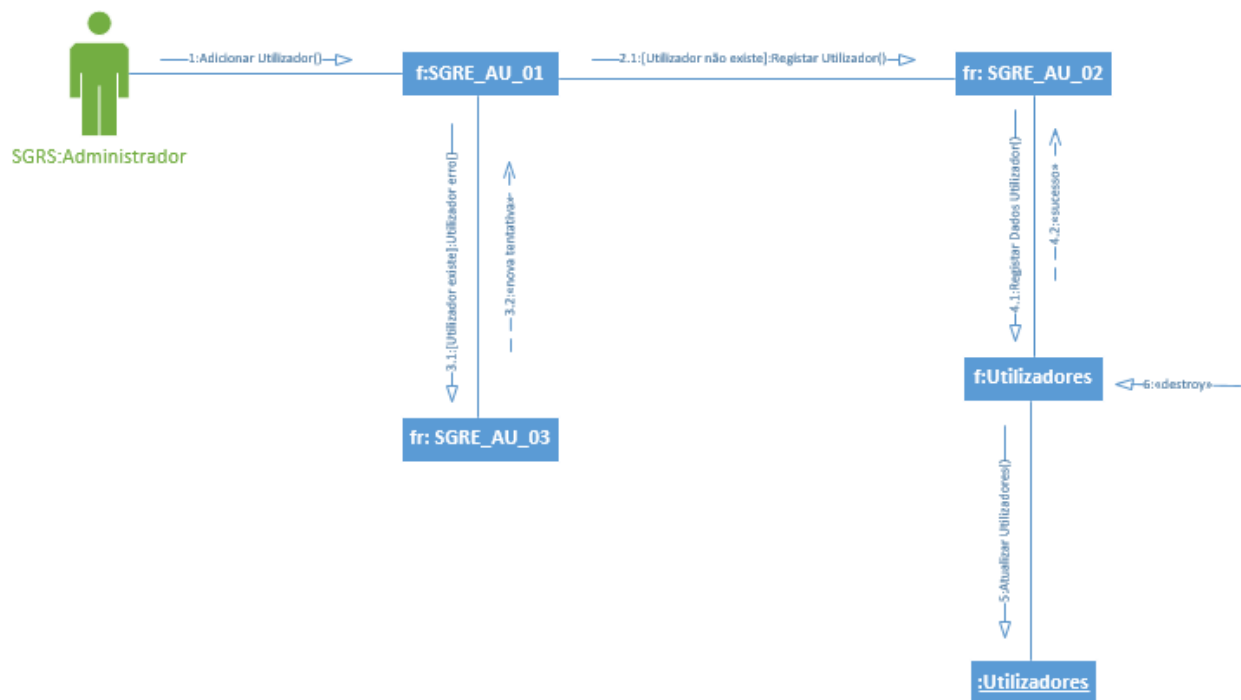


### Remover espaço

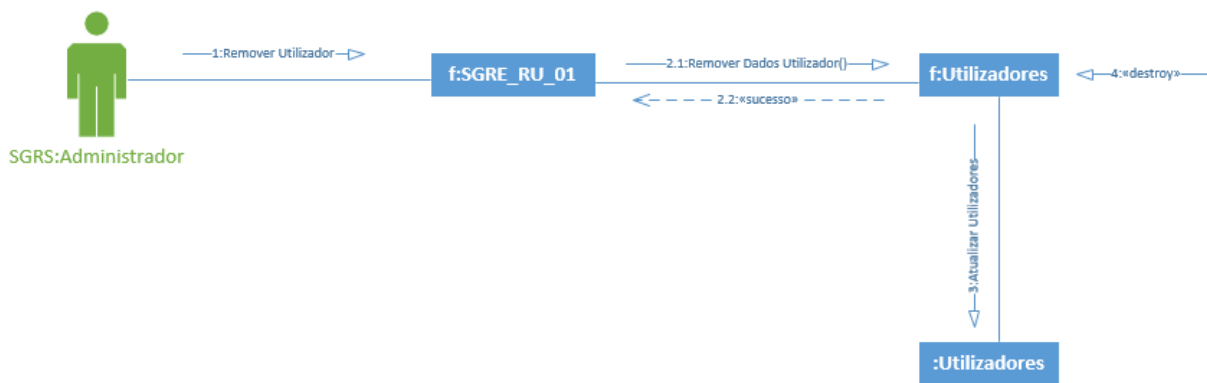




## Adicionar utilizador



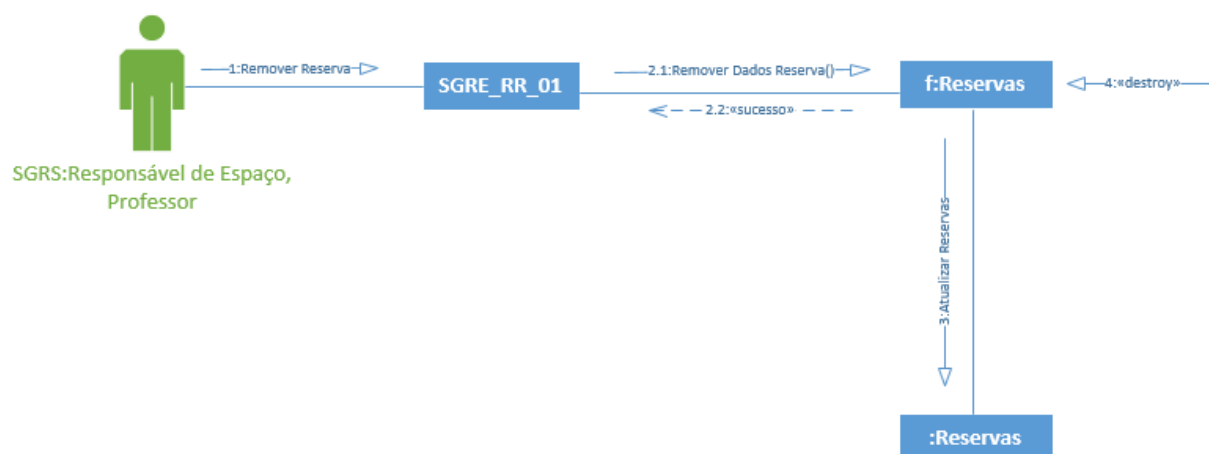
## Remover utilizador



## Editar utilizador



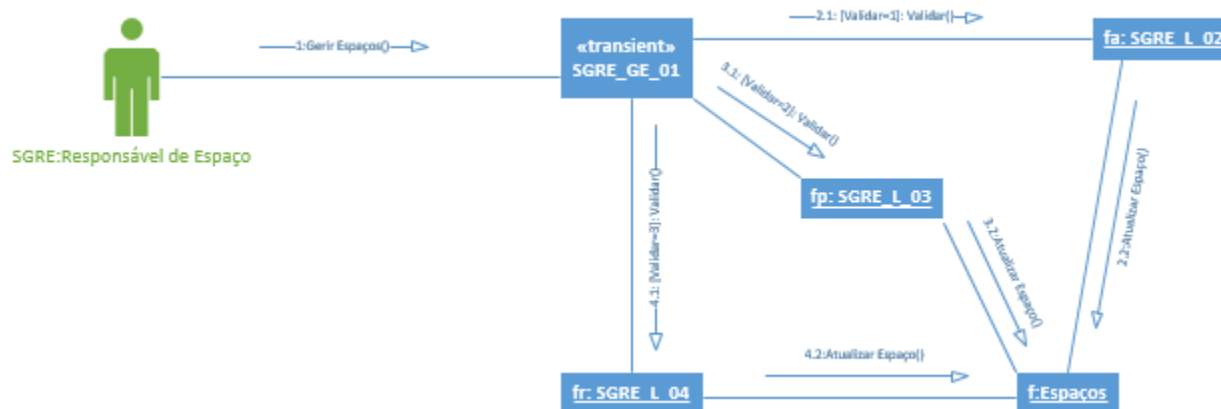
## Remover reserva



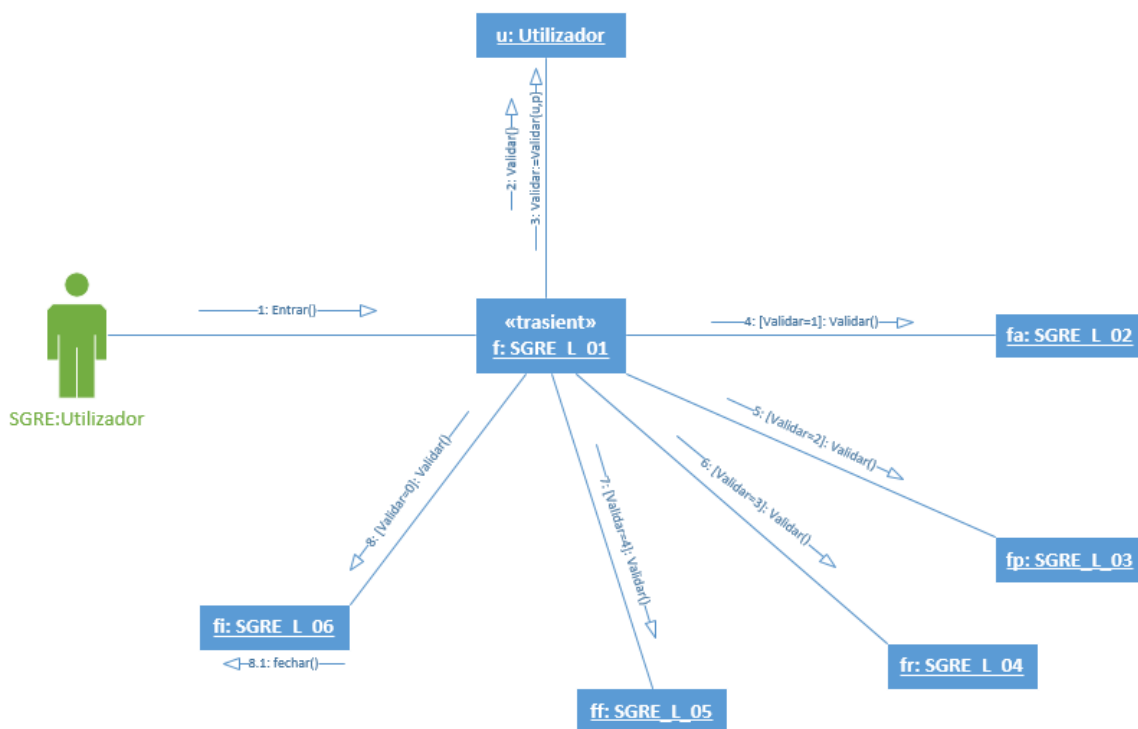
## Adicionar reserva



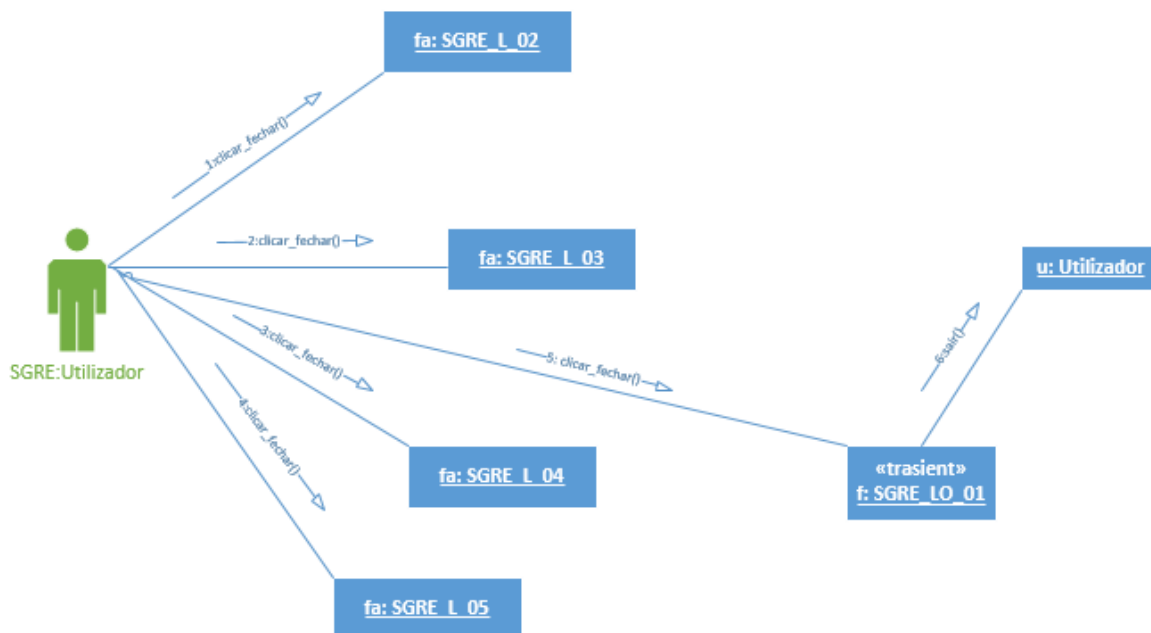
## Gerir espaços



## Login



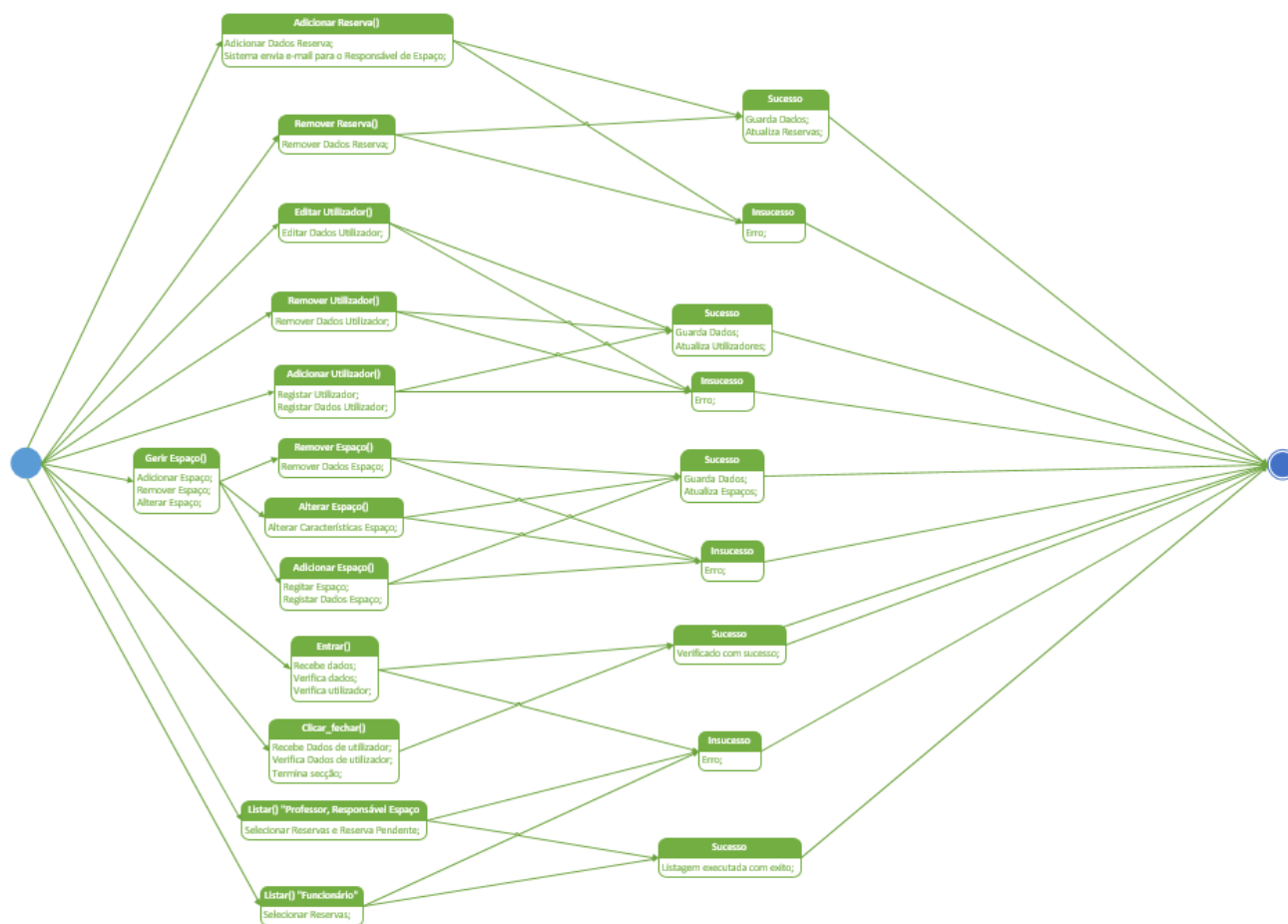
## Logout



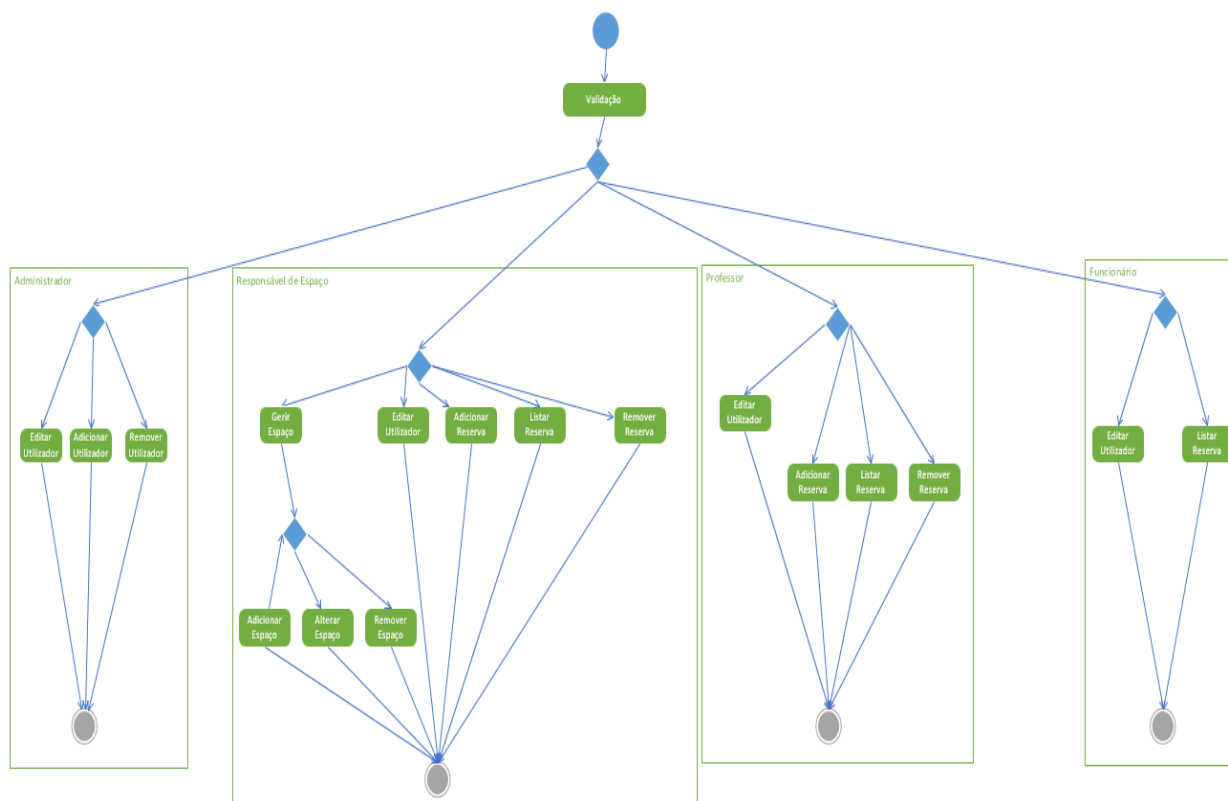
## Listar Reserva (Professor, Responsável de Espaço)



## Diagrama de Estados



## Diagrama de Atividades



## Implementação dos Subprogramas Armazenados

### Funcionalidade 1

Esta funcionalidade tem como objetivo criar um procedimento que altere o responsável de um espaço. O código de criação é o seguinte:

```
CREATE PROCEDURE alterar_responsavel_de_espaco(p_utilizador_atual IN
VARCHAR2, p_utilizador_novo IN VARCHAR2, res OUT NUMBER) IS
    v_id_antigo Utilizadores.id_utilizador%TYPE;
    v_id_novo Utilizadores.id_utilizador%TYPE;
BEGIN
    SELECT id_utilizador INTO v_id_antigo FROM Utilizadores WHERE nome_utilizador =
p_utilizador_atual;
    SELECT id_utilizador INTO v_id_novo FROM Utilizadores WHERE nome_utilizador =
p_utilizador_novo;
    UPDATE Espacos SET id_utilizador=v_id_novo WHERE id_utilizador=v_id_antigo;
    res := 0;
END;
```

### Funcionalidade 2

Esta funcionalidade tem como objetivo criar um procedimento com o intuito de se poder alterar a palavra-passe do utilizador. O código de criação é o seguinte:

```
CREATE PROCEDURE edi_user_pass (p_nome_Utilizador IN VARCHAR2, p_pass_antigo IN
VARCHAR2, p_pass_novo IN VARCHAR2) IS
    v_pass Utilizadores.palavra_passe%TYPE,
    v_id_user Utilizadores.id_Utilizador%TYPE;
BEGIN
    SELECT id_utilizador, palavra_passe INTO v_id_user, v_pass FROM Utilizadores WHERE
nome_utilizador=p_nome_Utilizador;
    IF v_pass = p_pass_antigo THEN
        UPDATE Utilizadores SET palavra_passe=p_pass_novo WHERE id_utilizador=v_id_user;
    END IF;
END;
```

### Funcionalidade 3

Esta funcionalidade tem como objetivo criar um procedimento que altere o estado de uma reserva. O código de criação é o seguinte:

```
CREATE PROCEDURE alterarEstadoDaReservaAceite(p_id_reserva IN NUMBER, res OUT NUMBER)
IS
BEGIN
    UPDATE Reservas SET estado='a' WHERE id_Reserva=p_id_reserva;
    res := 0;
END;
```



## Funcionalidade 4

Esta funcionalidade tem como objetivo criar um procedimento com o intuito de se poder editar espaços. O código de criação é o seguinte:

```
CREATE PROCEDURE editarEspaco (p_nome_atual IN VARCHAR2, p_nome IN VARCHAR2, p_piso IN
NUMBER, p_edificio IN VARCHAR2,
    p_n_computadores IN NUMBER, p_projedor IN CHAR, p_n_lugares IN NUMBER,
p_outras_caracteristicas IN VARCHAR2,
    p_nome_utilizador IN VARCHAR2, res OUT NUMBER) IS
    v_id_user Utilizadores.id_utilizador%TYPE;
    v_id_espaco Espacos.id_espaco%TYPE;
BEGIN
    SELECT id_utilizador INTO v_id_user FROM Utilizadores WHERE NOME_UTILIZADOR =
p_nome_utilizador;
    SELECT id_espaco INTO v_id_espaco FROM Espacos WHERE nome=p_nome_atual;
    UPDATE Espacos SET nome=p_nome, piso=p_piso, edificio=p_edificio,
n_computadores=p_n_computadores,
    projetor=p_projedor, n_lugares=p_n_lugares,
outras_caracteristicas=p_outras_caracteristicas,
    id_utilizador=v_id_user WHERE id_espaco=v_id_espaco;
    res := 0;
END;
```

## Funcionalidade 5

Esta funcionalidade tem como objetivo criar um procedimento com o intuito de se poder adicionar espaços. O código de criação é o seguinte:

```
CREATE PROCEDURE adicionarEspaco (p_nome IN VARCHAR2, p_piso IN NUMBER, p_edificio IN
VARCHAR2,
    p_n_computadores IN NUMBER, p_projedor IN CHAR, p_n_lugares IN NUMBER,
p_outras_caracteristicas IN VARCHAR2,
    p_nome_utilizador IN VARCHAR2, res OUT NUMBER) IS
    v_id_espaco Espacos.id_espaco%TYPE;
    v_id_user Utilizadores.id_utilizador%TYPE;
BEGIN
    v_id_espaco := id_proximo_utilizador;

    SELECT id_utilizador INTO v_id_user FROM Utilizadores WHERE NOME_UTILIZADOR =
p_nome_utilizador;

    INSERT INTO Espacos(id_espaco, nome, piso, edificio, n_computadores, projetor,
```

```
        n_lugares, outras_caracteristicas,id_utilizador) VALUES (v_id_espaco,p_nome,  
p_piso, p_edificio,  
  
        p_n_computadores, p_projeto, p_n_lugares,  
p_outras_caracteristicas,v_id_user);  
  
        res := 0;  
  
END;
```

## Funcionalidade 6

Esta funcionalidade tem como objetivo criar um procedimento com o intuito de se poder inserir utilizadores. O código de criação é o seguinte:

```
CREATE PROCEDURE inserir_utilizadores(p_nome_utilizador IN VARCHAR2, p_palavra_passe  
IN VARCHAR2,  
  
        p_tipo_utilizador IN VARCHAR2, p_nome_completo IN VARCHAR2, p_email IN  
VARCHAR2, res OUT NUMBER) IS  
  
        v_id_utilizador Utilizadores.id_utilizador%TYPE;  
  
BEGIN  
  
        v_id_utilizador := id_proximo_utilizador;  
  
        INSERT INTO Utilizadores(id_utilizador,  
nome_utilizador,palavra_passe,tipo_utilizador,  
  
        nome_completo,email) VALUES (v_id_utilizador, p_nome_utilizador,  
p_palavra_passe, p_tipo_utilizador,  
  
        p_nome_completo, p_email);  
  
        res := 0;  
  
END;
```

## Funcionalidade 7

Esta funcionalidade tem como objetivo criar um procedimento com o intuito de se poder inserir reservas. O código de criação é o seguinte:

```
CREATE PROCEDURE inserir_reserva(p_data_reserva IN DATE, p_hora_inicio IN CHAR,
p_hora_fim IN CHAR, p_estado IN CHAR,

    p_descricao_uso IN VARCHAR2, p_nome_utilizador IN VARCHAR2, p_nome_espaco IN
VARCHAR2, res OUT NUMBER) IS

    v_id_reservas Reservas.id_reserva%TYPE;

    v_id_utilizador Utilizadores.id_utilizador%TYPE;

    v_id_espaco Espacos.id_espaco%TYPE;

BEGIN

    SELECT id_utilizador INTO v_id_utilizador FROM Utilizadores WHERE
nome_utilizador=p_nome_utilizador;

    SELECT id_espaco INTO v_id_espaco FROM Espacos WHERE nome=p_nome_espaco;

    v_id_reservas := id_proximo_reservas;

    INSERT INTO Reservas(id_reserva, data_reserva, hora_inicio, hora_fim, estado,
descricao_uso,

        id_utilizador, id_espaco) VALUES (v_id_reservas, p_data_reserva,
p_hora_inicio, p_hora_fim, p_estado,

        p_descricao_uso, v_id_utilizador, v_id_espaco);

    res := 0;

END;
```

## Funcionalidade 8

Esta funcionalidade tem como objetivo criar um procedimento com o intuito de se poder listar espaços. O código de criação é o seguinte:

```
CREATE PROCEDURE listarEspaco (p_edificio IN VARCHAR2, p_piso IN NUMBER, p_lista OUT  
SYS_REFCURSOR) IS  
BEGIN  
    OPEN p_lista FOR  
        SELECT * FROM Espacos WHERE edificio = p_edificio AND piso = p_piso;  
END;
```

## Funcionalidade 9

Esta funcionalidade tem como objetivo criar um procedimento com o intuito de se poder listar espaços que estejam disponíveis. O código de criação é o seguinte:

```
CREATE PROCEDURE listarEspacosDisponiveis(p_piso IN NUMBER, p_edificio IN VARCHAR2,  
p_nPC IN NUMBER, projetor IN CHAR, p_nLugares IN NUMBER,  
  
p_dia_reserva IN DATE, p_hora_inicio IN VARCHAR2, p_hora_fim IN VARCHAR2, p_lista  
OUT SYS_REFCURSOR) IS  
  
BEGIN  
  
    OPEN p_lista FOR  
  
        SELECT E.nome, E.outras_caracteristicas FROM Espacos E INNER JOIN Reservas R  
ON E.id_espaco = R.id_espaco WHERE E.piso = p_piso AND  
  
        E.edificio = p_edificio AND E.n_computadores = p_nPC AND E.projetor =  
projetor AND E.n_lugares = p_nLugares AND  
  
        R.data_reserva = p_dia_reserva AND R.hora_inicio <> p_hora_inicio AND  
R.hora_fim <> p_hora_fim;  
  
END;
```

## Funcionalidade 10

Esta funcionalidade tem como objetivo criar um procedimento com o intuito de se poder listar as reservas de cada utilizador. O código de criação é o seguinte:

```
CREATE PROCEDURE listarReservasUtilizador(p_estado IN VARCHAR2, p_nome_utilizador IN
VARCHAR2, p_lista OUT SYS_REFCURSOR) IS
    v_id_utilizador Utilizadores.id_utilizador%TYPE;
BEGIN
    SELECT id_utilizador INTO v_id_utilizador FROM Utilizadores WHERE nome_utilizador =
p_nome_utilizador;
    IF p_estado = 't' THEN
        OPEN p_lista FOR
            SELECT * FROM Reservas WHERE id_utilizador = v_id_utilizador;
    ELSE
        OPEN p_lista FOR
            SELECT * FROM Reservas WHERE id_utilizador = v_id_utilizador AND
estado=p_estado;
    END IF;
END;
```

## Funcionalidade 11

Esta funcionalidade tem como objetivo criar um procedimento com o intuito de se poder listar todos os espaços. O código de criação é o seguinte:

```
CREATE PROCEDURE listarTodosEspaco(p_lista OUT SYS_REFCURSOR) IS
BEGIN
    OPEN p_lista FOR
        SELECT nome FROM Espacos;
END;
```

## Funcionalidade 12

Esta funcionalidade tem como objetivo criar um procedimento com o intuito de se poder listar todos os responsáveis de espaços. O código de criação é o seguinte:

```
create or replace PROCEDURE listarTodosResponsaveis (p_lista OUT SYS_REFCURSOR) IS
BEGIN

    OPEN p_lista FOR

        SELECT U.nome_utilizador, U.nome_completo, U.email, COUNT(E.id_espaco) AS
n_espacos

        FROM Utilizadores U INNER JOIN Espacos E ON U.id_utilizador=E.id_utilizador
WHERE U.TIPO_UTILIZADOR = 're'
```

```
GROUP BY U.nome_utilizador, U.nome_completo, U.email;
```

```
END;
```

## Funcionalidade 13

Esta funcionalidade tem como objetivo criar um procedimento com o intuito de se fazer a creditação do utilizador. O código de criação é o seguinte:

```
CREATE PROCEDURE creditar(p_nome_utilizador IN VARCHAR2, p_pass IN VARCHAR2, p_user  
OUT SYS_REFCURSOR  
  
    ) IS  
  
    v_nome_utilizador NUMBER(1);  
  
    v_pass NUMBER(1);  
  
BEGIN  
  
    SELECT COUNT(nome_utilizador), COUNT(palavra_passe) INTO v_nome_utilizador, v_pass  
FROM Utilizadores  
  
    WHERE nome_utilizador=p_nome_utilizador AND palavra_passe=p_pass;  
  
IF v_nome_utilizador>0 AND v_pass>0 THEN  
  
    OPEN p_user FOR  
  
        SELECT * FROM Utilizadores WHERE nome_utilizador=p_nome_utilizador AND  
palavra_passe=p_pass;  
  
    END IF;  
  
END;
```

## Funcionalidade 14

Esta funcionalidade tem como objetivo criar um procedimento com o intuito de se pesquisar espaços. O código de criação é o seguinte:

```
CREATE PROCEDURE pesquisarEspaco(p_edificio IN VARCHAR2, p_piso IN NUMBER, p_lista OUT
SYS_REFCURSOR) IS
BEGIN
    OPEN p_lista FOR
        SELECT NOME FROM Espacos WHERE edificio = p_edificio AND piso = p_piso;
END;
```

## Funcionalidade 15

Esta funcionalidade tem como objetivo criar um procedimento com o intuito de se pesquisar um determinado espaço. O código de criação é o seguinte:

```
CREATE PROCEDURE pesquisarEspacoX(p_nome IN VARCHAR2, p_lista OUT SYS_REFCURSOR) IS

BEGIN

    OPEN p_lista FOR

        SELECT * FROM Espacos WHERE nome=p_nome;

END;
```

## Funcionalidade 16

Esta funcionalidade tem como objetivo criar um procedimento com o intuito de se pesquisar utilizadores. O código de criação é o seguinte:

```
CREATE PROCEDURE pesquisarUtilizadores(p_filtro IN VARCHAR2, p_lista OUT
SYS_REFCURSOR) IS

BEGIN

    IF p_filtro = 't' THEN

        OPEN p_lista FOR

            SELECT * FROM Utilizadores;

    ELSE
```

```
OPEN p_lista FOR

SELECT * FROM Utilizadores WHERE tipo_utilizador = p_filtro;

END IF;

END;
```

## Funcionalidade 17

Esta funcionalidade tem como objetivo criar um procedimento com o intuito de se remover utilizadores. O código de criação é o seguinte:

```
CREATE PROCEDURE remover_utilizadores(p_nome_utilizador IN VARCHAR2, res OUT NUMBER)
IS
BEGIN
    DELETE FROM Utilizadores WHERE nome_utilizador=p_nome_utilizador;

    res := 0;

END;
```

## Funcionalidade 18

Esta funcionalidade tem como objetivo criar um procedimento com o intuito de se remover espaços. O código de criação é o seguinte:

```
CREATE PROCEDURE eliminarEspaco(p_nome_espaco IN VARCHAR2, res OUT NUMBER) IS
BEGIN
    DELETE FROM Espacos WHERE nome=p_nome_espaco;
    res :=0;
END;
```

## Funcionalidade 19

Esta funcionalidade tem como objetivo criar um procedimento com o intuito de se remover reservas. O código de criação é o seguinte:

```
CREATE PROCEDURE removerReserva(p_id_reserva IN NUMBER, res OUT NUMBER) IS
BEGIN
    DELETE FROM Reservas WHERE id_reserva=p_id_reserva;
    res :=0;
END;
```



## Funcionalidade 20

Esta funcionalidade tem como objetivo criar uma função que retorna o próximo id disponível da tabela Espacos. O código de criação é o seguinte:

```
CREATE FUNCTION id_proximo_espaco
RETURN NUMBER IS
    v_ultimo Espacos.id_espaco%TYPE;
    v_max Espacos.id_espaco%TYPE;
BEGIN
    SELECT COUNT(*) INTO v_ultimo FROM Espacos;
    SELECT MAX(id_espaco) INTO v_max FROM Espacos;
    IF v_ultimo = v_max THEN
        RETURN v_max+1;
    ELSIF v_ultimo = 0 THEN
        RETURN 1;
    ELSE
        FOR i IN 1..v_max LOOP
            SELECT COUNT(*) INTO v_ultimo FROM Espacos WHERE id_espaco = i;
            IF v_ultimo=0 THEN
                RETURN i;
            END IF;
        END LOOP;
    END IF;
END id_proximo_espaco;
```

## Funcionalidade 21

Esta funcionalidade tem como objetivo criar uma função que retorna o próximo id disponível da tabela Reservas. O código de criação é o seguinte:

```
CREATE FUNCTION id_proximo_reservas
RETURN NUMBER IS
    v_ultimo Reservas.id_reserva%TYPE;
    v_max Reservas.id_reserva%TYPE;
BEGIN
    SELECT COUNT(*) INTO v_ultimo FROM Reservas;
    SELECT MAX(id_reserva) INTO v_max FROM Reservas;
    IF v_ultimo = v_max THEN
        RETURN v_max+1;
    ELSIF v_ultimo = 0 THEN
        RETURN 1;
    ELSE
        FOR i IN 1..v_max LOOP
            SELECT COUNT(*) INTO v_ultimo FROM Reservas WHERE id_reserva = i;
            IF v_ultimo=0 THEN
                RETURN i;
            END IF;
        END LOOP;
    END IF;
END id_proximo_reservas;
```

## Funcionalidade 22

Esta funcionalidade tem como objetivo criar uma função que retorna o próximo id disponível da tabela Utilizadores. O código de criação é o seguinte:

```
CREATE FUNCTION id_proximo_utilizador
RETURN NUMBER IS
    v_ultimo Utilizadores.id_utilizador%TYPE;
    v_max Utilizadores.id_utilizador%TYPE;
BEGIN
    SELECT COUNT(*) INTO v_ultimo FROM Utilizadores;
    SELECT MAX(id_utilizador) INTO v_max FROM Utilizadores;
    IF v_ultimo = v_max THEN
        RETURN v_max+1;
    ELSIF v_ultimo = 0 THEN
        RETURN 1;
    ELSE
        FOR i IN 1..v_max LOOP
            SELECT COUNT(*) INTO v_ultimo FROM Utilizadores WHERE id_utilizador = i;
            IF v_ultimo=0 THEN
                RETURN i;
            END IF;
        END LOOP;
    END IF;
END id_proximo_utilizador;
```

## Funcionalidade 23

Esta funcionalidade tem como objetivo criar um trigger que ao ser removido um espaço apaga automaticamente as reservas desse espaço. O código de criação é o seguinte:

```
CREATE TRIGGER t_RemoverEspacos
BEFORE DELETE ON Reservas
FOR EACH ROW
BEGIN
    DELETE FROM Reservas WHERE id_Espaco=:OLD.id_Espaco;
END;
```

## Funcionalidade 24

Esta funcionalidade tem como objetivo criar um trigger que ao ser removido um utilizador apaga automaticamente as reservas desse utilizador. O código de criação é o seguinte:

```
CREATE TRIGGER t_RemoverUtilizador  
  BEFORE DELETE ON Utilizadores  
  FOR EACH ROW  
  BEGIN  
    DELETE FROM Reservas WHERE id_utilizador=:OLD.id_utilizador;  
  END;
```

## Funcionalidade 25

Esta funcionalidade tem como objetivo criar um procedimento com o intuito de se confirmar reservas. O código de criação é o seguinte:

```
CREATE PROCEDURE confirmar_reserva(p_id_reserva IN NUMBER, p_funcao IN VARCHAR2, res OUT  
NUMBER) IS  
  BEGIN  
    IF p_funcao = 'a' THEN  
      UPDATE Reservas SET estado = 'a' WHERE id_reserva=p_id_reserva;  
    ELSE  
      DELETE FROM Reservas WHERE id_reserva=p_id_reserva;  
    END IF;  
    res :=0;  
  END;
```

## Funcionalidade 26

Esta funcionalidade tem como objetivo criar um procedimento com o intuito de se listar reservas pendentes. O código de criação é o seguinte:

```
CREATE PROCEDURE listarReservasPendentes(p_id_responsavel IN NUMBER, p_lista OUT
SYS_REFCURSOR) IS

BEGIN

    OPEN p_lista FOR

        SELECT E.nome, R.data_reserva, R.hora_inicio, R.hora_fim, R.descricao_uso,
        U.nome_Utilizador, R.id_reserva

        FROM (Reservas R INNER JOIN Espacos E ON E.id_espaco = R.id_espaco) INNER JOIN
        Utilizadores U ON R.id_utilizador = U.id_utilizador

        WHERE E.id_utilizador = p_id_responsavel ORDER BY R.data_reserva,
        R.hora_inicio;

END;
```

## Funcionalidade 27

Esta funcionalidade tem como objetivo criar um procedimento com o intuito de se pesquisar reservas. O código de criação é o seguinte:

```
CREATE PROCEDURE pesquisarReserva(p_data IN VARCHAR2, p_id_espaco IN
NUMBER, p_lista OUT SYS_REFCURSOR) IS

    v_data Reservas.data_reserva%TYPE;

BEGIN

    v_data := TO_DATE(p_data,'dd/mm/yyyy');

    OPEN p_lista FOR

        SELECT R.hora_inicio, R.hora_fim, R.descricao_uso, U.nome_Utilizador
```

```
FROM (Reservas R INNER JOIN Espacos E ON E.id_espaco = R.id_espaco) INNER  
JOIN Utilizadores U ON R.id_utilizador = U.id_utilizador
```

```
WHERE E.id_espaco = p_id_espaco AND R.data_Reserva = v_data ORDER BY  
R.hora_inicio;
```

END;. O código de criação é o seguinte:

## **Conclusões**

Com a realização deste trabalho, e todos os diagramas a ele associados, desenvolver esta plataforma será mais fácil uma vez que já se encontram no documento todas as informações referentes ao projeto, bastando apenas programa-las. Foi possível ainda reestruturar a forma como a plataforma estava pensada para melhorar o seu funcionamento. Sem todo o processo de criação deste documento não seria possível observar os erros e corrigi-los, a fim de conseguirmos melhorar todo o processo de implementação e criação. Este trabalho dotou-nos ainda de técnicas e métodos para a análise e conceção de sistema informação, para no futuro poderemos produzir de forma clara e estruturada, documentos base para a criação de novas aplicações.

O nosso sistema é simples e eficaz, permitindo qualquer utilizador devidamente autenticado fazer a reserva de salas.

## **Bibliografia**

Mota, Ana - Desenvolvimento do Sistema de informação. 2004. Gestão comercial Playground Utilizando UML. Trabalho de fim de curso da Licenciatura em Informática de Gestão.

SAMARTINHO, João; NASCIMENTO, João - Introdução ao UML. Instituto Politécnico de Santarém - Escola Superior de Tecnologias e Gestão de Santarém;

SAMARTINHO, João; NASCIMENTO, João - Diagramas de Casos de Utilização. Instituto Politécnico de Santarém - Escola Superior de Tecnologias e Gestão de Santarém;

SAMARTINHO, João; NASCIMENTO, João - Diagramas de Classes. Instituto Politécnico de Santarém - Escola Superior de Tecnologias e Gestão de Santarém;

SAMARTINHO, João, NASCIMENTO, João - Diagramas de Sequências. Instituto Politécnico de Santarém - Escola Superior de Tecnologias e Gestão de Santarém;

SAMARTINHO, João, NASCIMENTO, João - Diagramas de Comunicação e de Estado. Instituto Politécnico de Santarém - Escola Superior de Tecnologias e Gestão de Santarém;

SAMARTINHO, João; NASCIMENTO, João - Diagramas de Atividade. Instituto Politécnico de Santarém - Escola Superior de Tecnologias e Gestão de Santarém;

NASCIMENTO, João - Tipos de dados compostos. Instituto Politécnico de Santarém - Escola Superior de Tecnologias e Gestão de Santarém;

NASCIMENTO, João - Utilização de Cursores; Estruturas de Controlo. Instituto Politécnico de Santarém - Escola Superior de Tecnologias e Gestão de Santarém;

NASCIMENTO, João - Cursores explícitos. Instituto Politécnico de Santarém - Escola Superior de Tecnologias e Gestão de Santarém;

NASCIMENTO, João - Conceitos avançados sobre cursores explícitos. Instituto Politécnico de Santarém - Escola Superior de Tecnologias e Gestão de Santarém;

NASCIMENTO, João - Exceções. Instituto Politécnico de Santarém - Escola Superior de Tecnologias e Gestão de Santarém;

NASCIMENTO, João - Subprogramas (Apresentação dada em aula). Instituto Politécnico de Santarém - Escola Superior de Tecnologias e Gestão de Santarém;

NASCIMENTO, João - Funções: Aplicacionais e Armazenadas. Instituto Politécnico de Santarém - Escola Superior de Tecnologias e Gestão de Santarém;