**Universidade do Minho**

**Departamento de Informática**

**Desenvolvimento de Sistemas de Software**

**Relatório de Projeto – Fase 1**

**** Bruno Arieira – a70565

 Joel Morais – a70841

 Carlos Vieira – a73974

**** João Palmeira – a73864

 Rodrigo Ferreira – a74216 **MIEI - 2017/2018**

**Introdução**

Este projeto visa o desenvolvimento de um sistema de gestão dos turnos práticos de um curso (licenciatura ou mestrado integrado). O objetivo é desenvolver uma aplicação que suporte o registo das Unidades Curriculares (UCs) e dos alunos e a gestão da alocação dos alunos aos turnos das UCs. A utilização desta aplicação é extremamente útil, pois facilita bastante a organização e distribuição dos alunos pelas diferentes UCs do curso em questão, evitando problemas como a sobreposição de horários de diferentes unidades curriculares e o excesso de alunos em cada turno, que poderiam colocar em causa o bom funcionamento do curso e consequente aproveitamento dos alunos.

De seguida, iremos explicar resumidamente o funcionamento da aplicação referida, e quais os parâmetros mais importantes a que a mesma deve obedecer.

Numa primeira fase, a alocação dos alunos deverá ser da responsabilidade da direção de curso. Posteriormente deverá existir uma fase que permita a troca entre alunos. Aos alunos com estatuto especial é permitido mudar de turno sem ser necessário um elemento de permuta, desde que o limite de alunos desse turno não seja ultrapassado. A capacidade de cada turno depende da sala e do tipo de turno. A terceira fase, que ocorre após o início das aulas, consiste em fazer a gestão dos turnos. Nesta fase, apenas o docente responsável da UC pode alterar a composição dos turnos, com base no excesso de faltas do aluno.

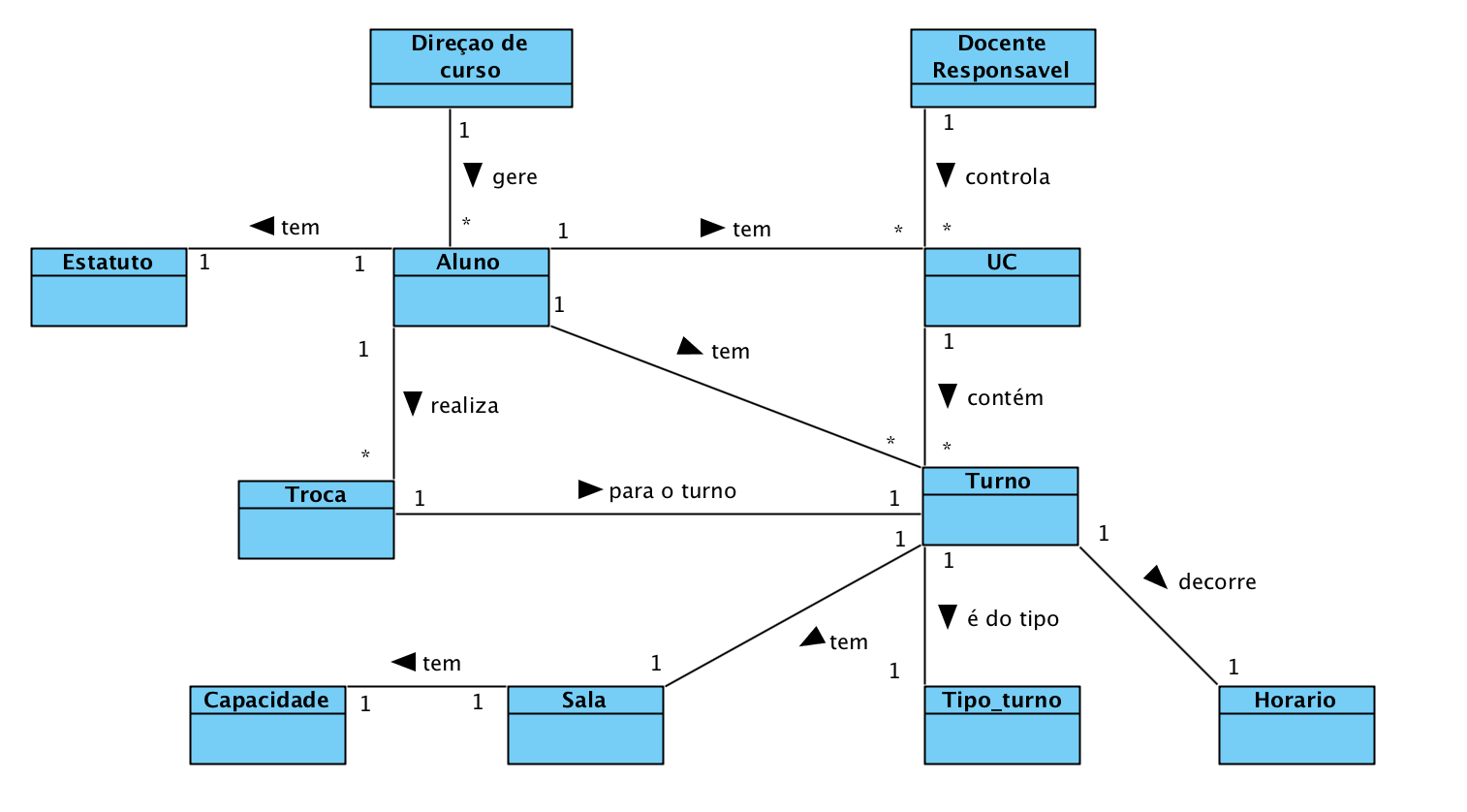
O sistema deverá ser robusto (evitar sobreposição de aulas no horário) e flexível (ser capaz de lidar com situações em que a sobreposição é inevitável). Para permitir uma avaliação do desempenho da gestão de turnos, deverá ser disponibilizada informação sobre a evolução dos turnos ao longo do semestre.

Neste relatório vamos apresentar o diagrama de Modelo de Domínio, o diagrama de Use Cases e a identificação/descrição dos Use Cases mais relevantes, tentando explicar as decisões tomadas ao realizar estes diagramas.

**Modelo de Domínio**

O Modelo de Domínio inclui todas as entidades que pensamos serem necessárias para o nosso projeto: *Aluno, Estatuto, Direção de Curso, Troca, UC, Docente Responsável, Turno, Tipo\_turno, Sala, Capacidade* e *Horário*.

Para um melhor planeamento, optamos por dividir o nosso modelo em três grupos: *Aluno, UC e Turno*. De seguida, iremos explicar a razão desta divisão.



**Figura 1** - Modelo de Domínio

* **Aluno**

As entidades que fazem parte desta categoria são: *Estatuto, Troca e Aluno*.

As entidades *Troca* e *Aluno* não iremos pronunciarmo-nos muito sobre isto já que os nomes dizem tudo. O aluno ao receber um horário ira fazer uma troca se assim o desejar, troca esta que poderá se fazer ou não consoante o estatuto que ele adquire. E é aqui que chegamos a outra entidade que inserimos neste grupo, *Estatuto*.

Criamos esta entidade, visto que consoante o estatuto que o aluno adquire (mais 1 outra “variável” que iremos falar no ultimo grupo) a troca de turno irá ser efetuada ou não.

* **UC**

As entidades que fazem parte desta categoria são: *Direção de Curso, Docente Responsável* e *UC*.

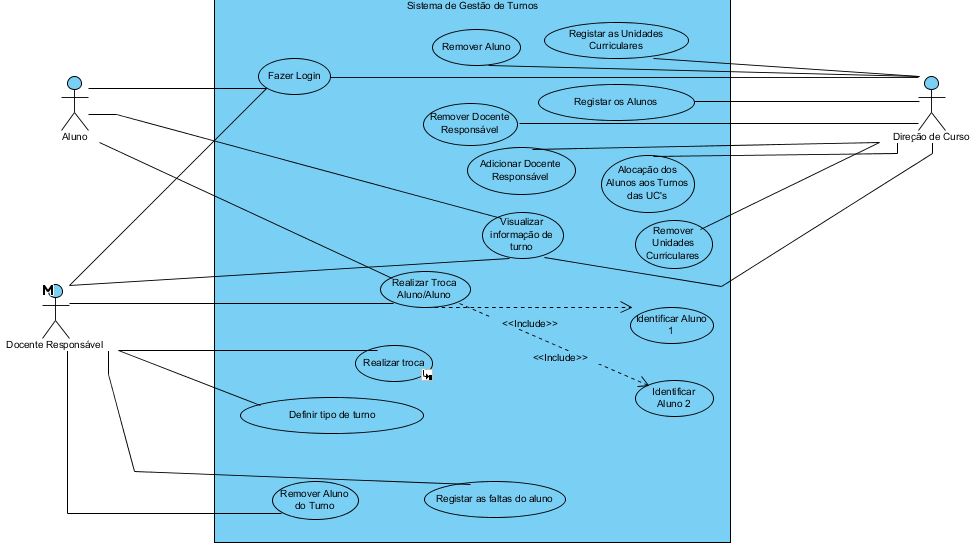
Há varias UC ‘s para fazer um horário logo criamos esta entidade que irá ter um docente responsável, docente este que foi escolhido pelo Diretor de Curso. E é assim que relacionamos estas 3 entidades neste nosso grupo. Sendo que tanto o Diretor de curso como o docente responsável terão poder para fazer as trocas de alunos.

* **Turno**

As entidades que fazem parte desta categoria são: *Turno, Tipo\_turno, Horário, Sala, Capacidade*.

Como o nome indica iremos ter 2 tipos de turno, sendo que consoante esse mesmo tipo de turno, a sala irá ser diferente. Se for um turno em que precise de mais gente iremos ter que escolher uma sala consoante a sua Capacidade par que caibam esses alunos todos do turno.

Apesar do nosso grupo ter dividido a explicação deste modelo em 3 grupos não quer dizer que não haja relação entre todas as Entidades. Existe e são várias as outras relações entre as entidades, mas para uma melhor explicação do porquê de termos escolhido este modelo, utilizamos estas entidades como exemplo.

**Descrição dos Use Cases**

**Figura 2** - Modelo de Use Case

1. **Fazer Login**

Este Use Case permite ao utilizador autentificar-se na plataforma. Para que tal aconteça, é necessário que o utilizador esteja registado.

Os dados do utilizador têm que ser inseridos para que estes possam ser validados pelo sistema. Se forem válidos, o utilizador fica autenticado e o sistema informa do sucesso desta operação. Caso contrário, o sistema informa da não validação dos mesmos e o registo do utilizador não é efetuado.

1. **Registar os alunos**

Permite ao diretor de curso registar um aluno. Para que tal aconteça, é necessário que o diretor de curso se encontre autenticado.

Em primeiro lugar, é necessário inserir uma unidade curricular, sendo a sua existência verificada pelo sistema. De seguida, deve-se inserir o número de um aluno, sendo também a sua existência verificada pelo sistema. O sistema irá verificar também se o aluno já está registado e, caso não esteja, irá prosseguir ao registo do mesmo. Neste caso, o sistema informa que o aluno foi registado com sucesso,

Caso a unidade curricular não exista, o número de aluno inserido não seja válido ou o aluno já esteja registado na UC inserida, o sistema informa o utilizador do sucedido.

1. **Registar as Unidades Curriculares**

Permite ao diretor de curso registar as unidades curriculares. Para que isso seja possível, é necessário que o diretor de curso se encontre autenticado.

O primeiro passo é inserir o ano correspondente à unidade curricular pretendida, ano este que o sistema irá verificar se existe. Depois disso, é necessário indicar qual a unidade curricular a ser tratada. Em relação a esta unidade curricular, o sistema irá verificar se existe, se corresponde ao ano inserido e se já se encontra registada. De seguida, informa que a unidade curricular foi registada com sucesso.

Caso a ano seja inválido, a unidade curricular não exista, não corresponda ao ano ou já tenha sido registada anteriormente, o sistema informa o diretor de curso que é impossível efetuar o seu registo.

1. **Adicionar docente responsável**

Permite ao diretor de curso adicionar um docente responsável de uma determinada unidade curricular. Para que tal seja possível, o diretor de curso tem que estar autenticado.

Em primeiro lugar, é necessário inserir a unidade curricular para a qual se pretende adicionar o referido docente responsável. O sistema irá efetuar a validação dessa unidade curricular. De seguida, tem que se inserir o número do docente. O sistema verifica se o docente correspondente a esse número existe, se a unidade curricular inserida está inserida no lote de unidades curriculares que esse docente pode lecionar. Após essa verificação, insere o docente responsável e informa do sucesso.

No entanto, se a unidade curricular não existir, informa que não é possível adicionar o docente responsável. O mesmo acontece caso o número do docente não corresponda a um docente desse curso ou a unidade curricular não se encontre no grupo de unidades curriculares que esse docente pode lecionar.

1. **Alocação dos alunos aos turnos das unidades curriculares**

Permite ao diretor de curso alocar os alunos num turno. Para que isso aconteça, o mesmo tem que estar autenticado.

O primeiro passo é inserir o número do aluno, cuja existência irá ser verificada pelo sistema. Após essa verificação, é necessário inserir a unidade curricular. A sua existência irá também ser verificada pelo sistema. De seguida, insere-se um turno, e o sistema verifica se esse turno existe e se não há sobreposição com aulas de outra UC a essa hora. Feita esta verificação, o aluno é inserido no turno e é dada a informação do sucesso da operação.

Caso o aluno, a unidade curricular ou o turno não existam, ou o aluno já esteja inscrito numa unidade curricular a essa hora, o sistema informa do sucedido.

1. **Remover Aluno**

Permite ao diretor de curso remover um aluno. É necessário que o diretor de curso se encontre autenticado.

É necessário inserir o número do aluno, cuja existência será verificada pelo sistema. Depois dessa verificação, o sistema remove o aluno e informa que o aluno foi removido com sucesso.

Caso o número de aluno inserido não exista, o sistema informa do erro sucedido.

1. **Remover Docente Responsável**

Permite ao diretor de curso remover o docente responsável por uma determinada unidade curricular. Para que tal seja possível, é necessário que o diretor de curso se encontre autenticado.

Em primeiro lugar, insere-se o nome de uma unidade curricular, e o sistema verifica se a mesma existe. De seguida, insere-se o número do docente responsável. Neste caso, o sistema verifica se o número corresponde a um docente, se esse docente é o responsável da unidade curricular. Após essas verificações, o sistema remove o docente responsável e informa do sucesso.

Caso a unidade curricular não exista, o número inserido não corresponda a nenhum docente ou corresponda a um docente que não seja responsável pela unidade curricular. o sistema informa da situação ocorrida.

1. **Remover unidades curriculares**

Permite ao diretor de curso remover uma unidade curricular. O diretor de curso tem que estar autenticado.

É necessário inserir uma unidade curricular, cuja existência será verificada pelo sistema. Após essa verificação, o sistema remove a unidade curricular e informa do sucesso.

Caso a unidade curricular inserida não exista, o sistema informa da situação.

1. **Visualizar informação do turno**

Permite a qualquer utilizador visualizar as informações de um determinado turno, desde que esse utilizador se encontre autenticado.

Em primeiro lugar, é necessário inserir uma unidade curricular, cuja existência irá ser verificada pelo sistema. De seguida, tem que se inserir um turno. O sistema verifica a sua existência e, depois disso, apresenta a informação.

Caso a unidade curricular ou o turno inserido não existam, o sistema informa da situação.

1. **Realizar troca entre dois alunos**

Permite ao docente responsável realizar uma troca entre dois alunos. Para que tal aconteça, é necessário que o docente responsável se encontre autenticado.

É necessário identificar os alunos e identificar a unidade curricular, processos estes que podem ser realizados através de *includes*, relacionados com esses use cases já definidos. Depois disso, o sistema realiza a troca entre os dois alunos.

1. **Identificar aluno com estatuto especial**

Permite identificar alunos com estatuto especial, sem qualquer pré-condição associada.

É necessário inserir o número do aluno que queremos identificar. Depois disso, o sistema vai verificar se o aluno existe e se o mesmo tem estatuto especial, informando do sucesso da identificação.

Caso o número de aluno inserido não corresponda a nenhum aluno, ou corresponda a um aluno sem estatuto especial, o sistema informa da situação ocorrida.

1. **Definir tipo de turno**

Permite ao docente responsável definir o tipo do turno (prático ou prático-laboratorial), sendo para isto necessário que o docente responsável se encontre autenticado.

Em primeiro lugar, é necessário inserir uma unidade curricular, cuja existência será verificada pelo sistema. Após isso, deve inserir-se um turno e um tipo de turno e, em ambos os casos, o sistema trata de verificar a existência dos mesmos. Depois de a verificação ser feita, é adicionado o tipo de turno.

Caso a unidade curricular, o turno ou o tipo de turno inseridos não existam, o sistema informa do erro ocorrido.

1. **Identificar alunos**

Permite que seja feita a identificação de dois alunos (necessária para realizar trocas), sem qualquer pré-condição atribuída.

É necessário inserir o número do aluno 1, cuja existência é verificada pelo sistema. O mesmo acontece para o aluno 2. Após essa verificação, o sistema seleciona a informação dos 2 alunos.

Caso o aluno 1 ou o aluno 2 inseridos não existam, o sistema informa da ocorrência do erro.

1. **Identificar unidade curricular**

Permite identificar uma unidade curricular (necessário para realizar trocas de alunos), e não tem qualquer pré-condição associada.

Neste use case, é apenas necessário inserir o nome da unidade curricular e, de seguida, o sistema verifica se a unidade curricular inserida existe. Após esta verificação, a UC é selecionada pelo sistema.

Caso a unidade curricular inserida não exista, o sistema informa disso mesmo.

1. **Remover aluno de um turno**

Permite ao docente responsável remover um aluno de um turno (por ultrapassar o número de faltas que pode ser dado). Para isto, o docente responsável tem que estar autenticado.

Em primeiro lugar, tem que se inserir o número do aluno e o nome da unidade curricular, cuja existência irá ser verificada pelo sistema. Depois disso, é necessário inserir um turno. Em relação ao turno inserido, o sistema irá verificar a existência do mesmo, se o número de faltas foi realmente excedido e, após essa verificação, remover o aluno do turno, informando do sucesso da operação.

Caso o aluno, a unidade curricular ou o turno não existam, o sistema informa do erro ocorrido, o que também acontece caso o número de faltas não tenha sido excedido.

1. **Identificar aluno com estatuto especial**

Permite identificar alunos com estatuto especial, e não tem nenhum pré-condição associada.

É necessário inserir o número do aluno em questão. De seguida, o sistema verifica se o aluno com esse número existe e se esse aluno possui estatuto especial e, após essa verificação, informa do sucesso.

Caso o aluno não exista ou não possua estatuto especial, o sistema informa da situação ocorrida.

1. **Realizar troca de aluno com estatuto especial**

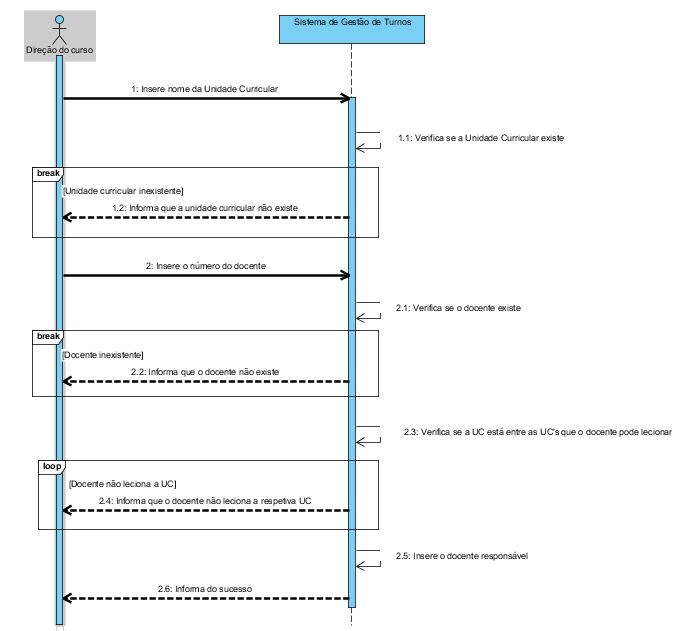
Permite ao docente responsável efetuar a mudança de turno de um aluno com estatuto especial, sendo necessário que o docente se encontre autenticado.

Em primeiro lugar, é necessário identificar a unidade curricular (através de um *include*, visto este use case já estar definido). Depois disso, é necessário inserir um turno, em que o sistema verifica se o mesmo existe e se tem capacidade disponível para “acolher” o aluno em questão. Após essa verificação, resta identificar o aluno com estatuto especial (usando novamente um *include* referente a essa identificação).

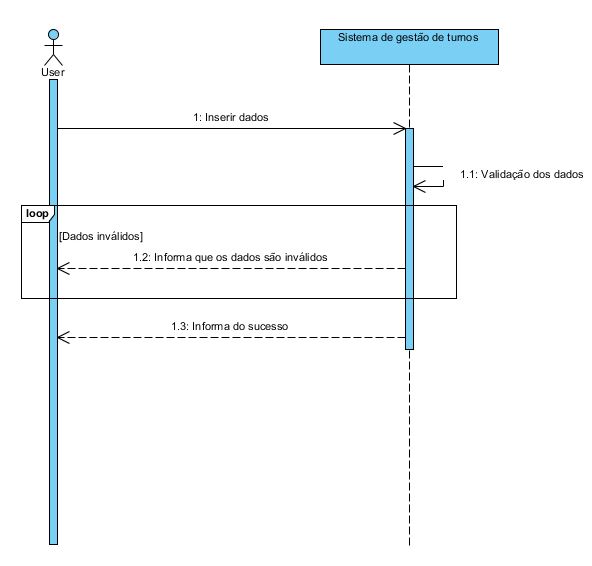
Caso o turno não exista ou não tenha capacidade disponível, o sistema informa da situação ocorrida.

**Diagramas de sequência**

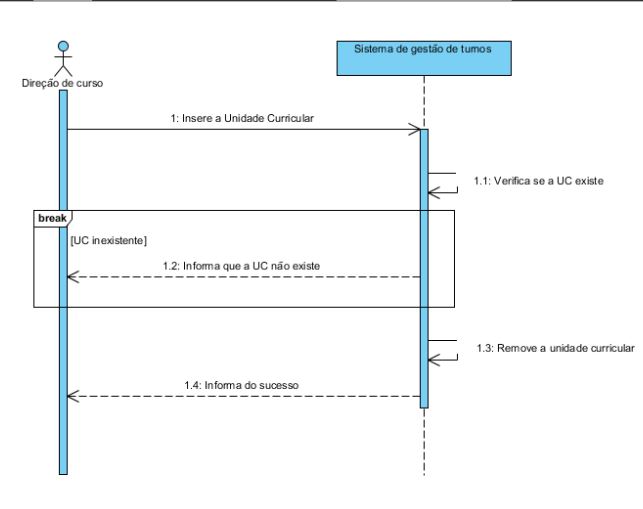
Neste tópico são apresentados alguns exemplos dos diagramas de sequência correspondentes aos use cases acima apresentados.



**Figura 3** – Adicionar Docente Responsável- diagrama de sequência

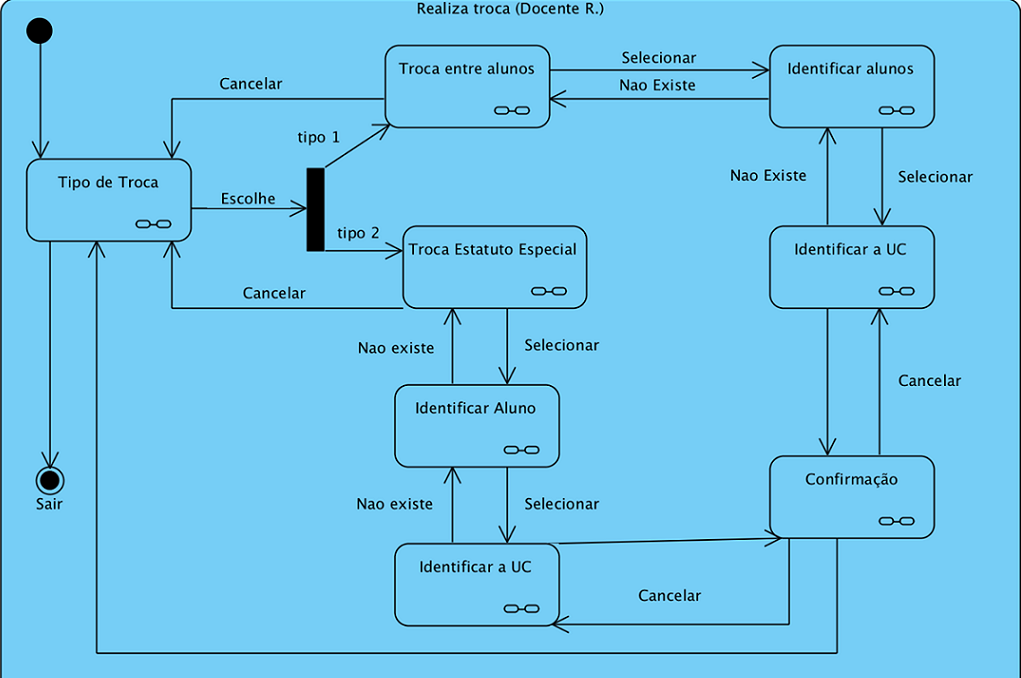


**Figura 4** – Login- diagrama de sequência



**Figura 4** – Remover UC- diagrama de sequência

**Diagrama de estado**

****

**Conclusão**

Nesta primeira fase do projeto, foi-nos requerida a modelação do problema apresentado. O objetivo é ajudar-nos a compreender melhor a realidade do projeto e a sua estrutura essencial, de forma a planear de forma rigorosa o projeto final e prevenir possíveis riscos do código a implementar

Para tal, elaborámos para esta parte o Modelo de Domínio, a definição dos Use Cases e respetivos Diagramas de Sequência e o Diagrama de Estado.

O Modelo de Domínio serviu para planear as várias entidades que poderemos vir a precisar e a relação entre elas.

Depois partimos para a elaboração dos *Use Case*, planeando os mesmos em tabelas de Excel, para facilitar a organização e futura implementação do código. Aqui tentámos prever o comportamento do programa às diferentes ações de um utilizador. Em relação aos Use Case definidos, realizamos também os respetivos diagramas de sequência.

Em modo de conclusão, pensámos que esta primeira parte é bastante útil para o projeto final, visto já estar tudo organizado e estruturado sendo que, na parte final, será necessário implementar o código correspondente.