



Universidade do Minho

Desenvolvimento de Sistemas de Software

Ano Letivo de 2017/2018

Sistema de Gestão de Turnos

Primeira Fase – Levantamento de Requisitos



David Kramer
77849



Joaquim Simões
77653



José Menezes
79187



Rafael Costa
61799

Índice

Introdução	2
Requisitos	2
Aluno.....	2
Professor	2
Direção de curso	3
Requisitos do sistema.....	3
Descrição e modelos	4
Modelo de domínio	4
Diagramas de use case.....	5
Uses cases mais importantes.....	6
Interface	9
Página inicial do aluno	9
Página de dos vários turnos do aluno	10
Página de Gestão de Pedidos de troca do aluno	11
Página para trocar de turno do aluno	12
Página para gerir um pedido do aluno	13
Página inicial do professor	14
Página de modificação de turno	15
Página inicial da direção de curso	16
Análise crítica.....	17

Introdução

Este trabalho tem como objetivo o desenvolvimento de uma aplicação de gestão de turnos no contexto de um curso universitário. No início de cada ano letivo existe sempre uma atribuição aleatória de diferentes horários aos alunos, mas nem sempre estes correspondem com aquilo que os alunos desejam. Esta aplicação trata de tentar satisfazer os alunos permitindo que estes efetuem trocas de turnos entre eles, dando prioridade aos alunos com estatuto especial. Adicionalmente, também permite aos professores responsáveis das unidades curriculares fazerem gestão dos respetivos turnos, segundo a administração da direção de curso.

Este relatório pretende apresentar, em termos gerais, os aspetos mais importantes da aplicação: o seu âmbito, requisitos, aspetos funcionais, interface, tal como fornecer uma análise crítica da mesma.

Requisitos

Esta aplicação é orientada para alunos, professores regentes das unidades curriculares, e direção de curso. Como tal, os requisitos são diferentes para cada um deles. De seguida são enumerados os requisitos, para cada uma das entidades mencionadas, que se espera que sejam cumpridos pelo Sistema de Gestão de Turnos.

Aluno

Relativamente a um aluno o sistema deve suportar dois tipos desta entidade: um aluno de estatuto normal e um aluno de estatuto trabalhador estudante. O segundo tipo apresenta privilégios relativamente às trocas de turnos. De seguida são enumeradas as funcionalidades que o sistema deve suportar no que toca à entidade Aluno:

1. Registrar-se em unidades curriculares.
2. Consultar as unidades curriculares inscritas e seus respetivos turnos atribuídos.
3. Solicitar uma troca de turno (no caso do aluno de estatuto normal).
4. Efetuar uma troca de turno direta, caso seja possível (no caso do aluno de estatuto trabalhador estudante).
5. Cancelar um pedido de uma troca de turno.
6. Verificar o estado de uma troca que entrou em fila de espera.

Professor

1. Consultar as unidades curriculares que leciona e verificar os seus respetivos turnos.
2. Definir a capacidade dos turnos das unidades curriculares que leciona.

3. Remover alunos dos seus turnos.
4. Efetuar trocas de turnos a alunos.
5. Adicionar alunos aos seus turnos.

Direção de curso

1. Configurar o sistema fornecendo um ficheiro contendo a informação das unidades curriculares de um curso, respetivo horário e turnos, lista de professores pertencentes a esse curso e respetivos alunos inscritos nesse curso (este ficheiro deve estar num formato válido conhecido pelo sistema).
2. Informar o sistema que deve proceder ao início da atribuição de turnos aos alunos.
3. Informar o sistema do início das aulas, sendo a partir desse momento apenas possível aos professores responsáveis alterar a composição dos turnos.

Requisitos do sistema

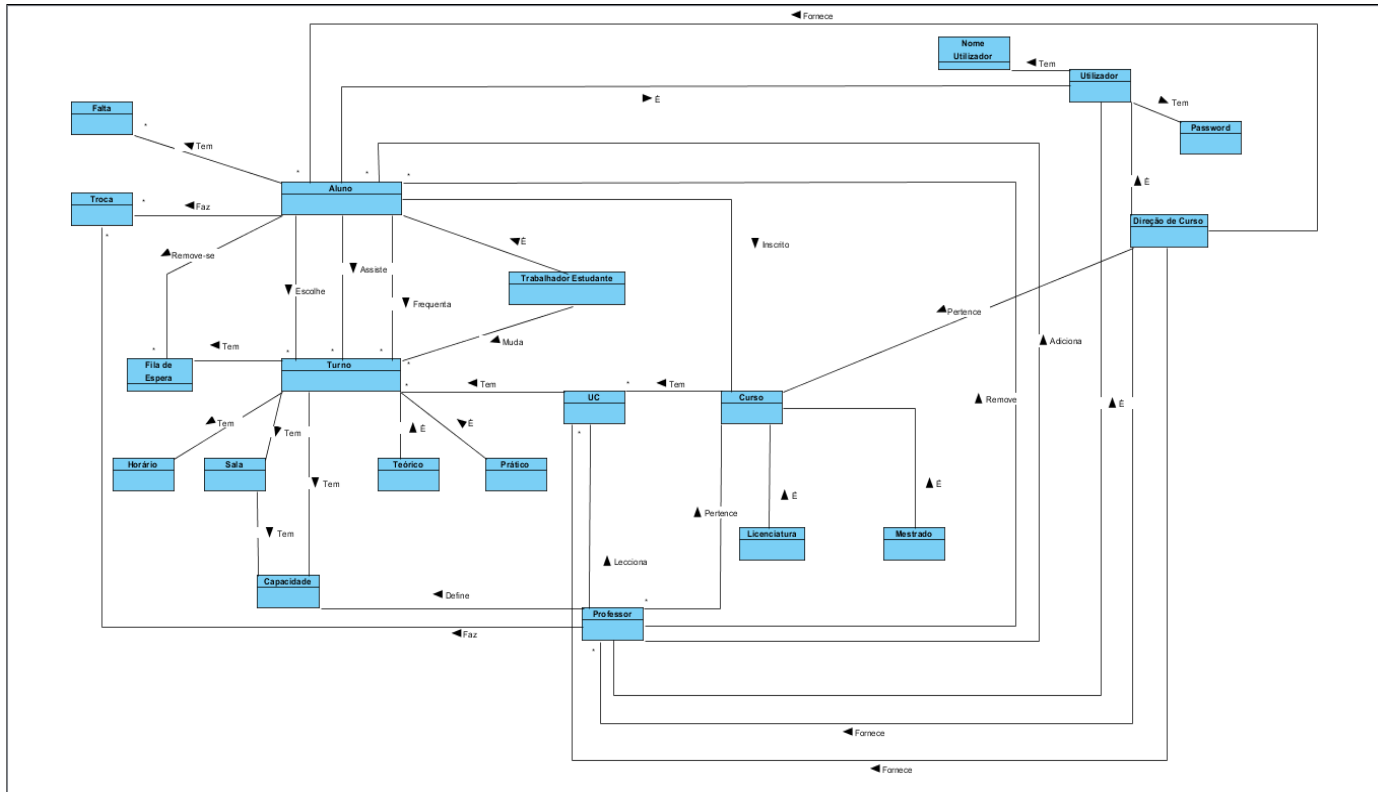
1. Deve permitir que cada uma das entidades descritas acima possa registar-se e efetuar *log in*.
2. Ser capaz de extrair a informação necessária do ficheiro de configuração fornecido pela direção de curso.
3. Atribuir horários (num modo aleatório) aos alunos. Os horários devem ser robustos e flexíveis, evitando que os alunos tenham aulas sobrepostas.
4. Gerir os pedidos de trocas de turnos através de filas de espera.
5. Conceder aos alunos com o estatuto de trabalhador estudante prioridade aos alunos de estatuto normal no que diz respeito às trocas de turnos.
6. Impedir que os alunos efetuem ou solicitem trocas entre si, depois de aulas já terem começado.

Além destes requisitos que apresentam funcionalidades do sistema, espera-se que outros requisitos não funcionais sejam cumpridos. Espera-se que a *User Interface* da aplicação seja intuitiva, que o tempo de resposta das diferentes funcionalidades do sistema aos diversos tipos de clientes sejam o menor possível. Finalmente, os dados armazenados pela aplicação devem ocupar o menor espaço de memória possível.

Descrição e modelos

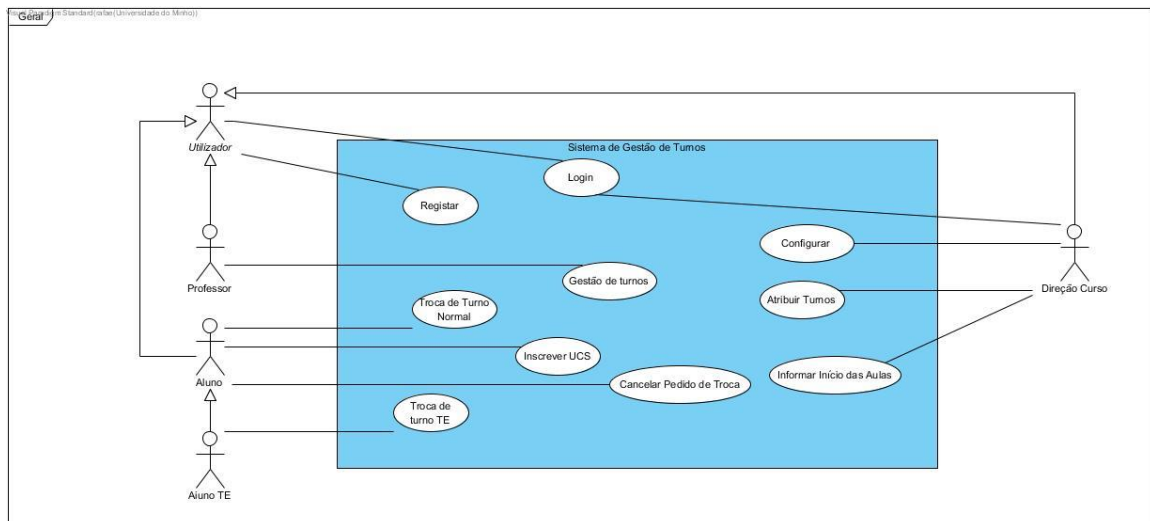
Para esta aplicação realizamos vários modelos e diagramas: modelo de domínio, diagramas de *use case*, especificação dos *use case* e diagramas de sequência.

Modelo de domínio

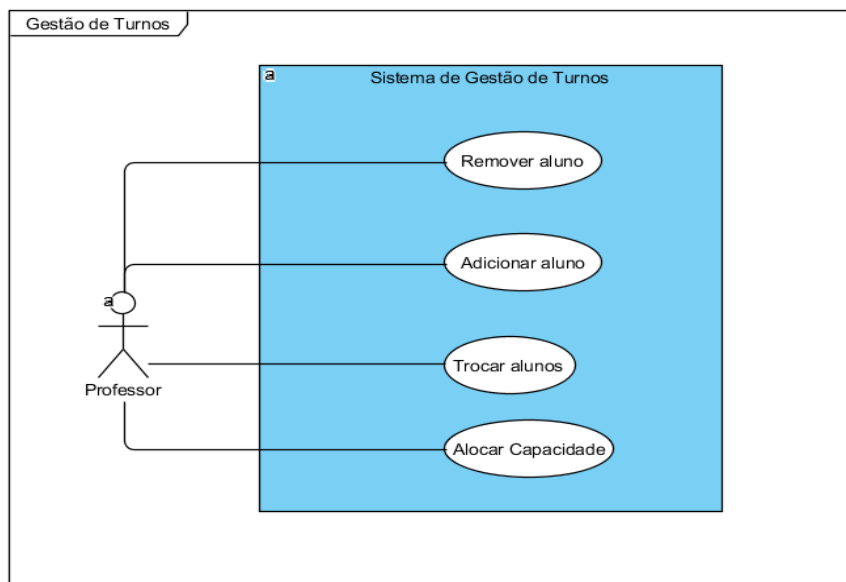


Este modelo retrata as entidades da aplicação e os relacionamentos entre elas. Salienta-se a importância do uso de filas de espera associadas a cada turno, o qual será usado no caso de os alunos quererem efetuar trocas de turnos entre eles. Este mecanismo será abordado posteriormente na especificação dos *use cases*. Os restantes relacionamentos são facilmente interpretados através da observação do esquema.

Diagramas de use case



Todas as funcionalidades que os diferentes tipos de utilizadores podem realizar encontram-se neste diagrama.



Dentro da gestão de turnos, o professor pode remover, adicionar ou trocar alunos e também alocar a capacidade do mesmo.

Uses cases mais importantes

Name		Troca de turno (aluno normal)	
Brief Description		Um aluno efetua a troca de turno de uma UC	
Preconditions		Está autenticado pelo sistema, é um aluno e tem um horário atribuído	
Post-conditions		O aluno fica inserido no turno a que solicitou troca	
		Actor Input	System Response
Flow of Events	1		Mostra a lista de UCs que o aluno está inscrito
	2	Seleciona UC	
	3		Mostra turnos dessa UC
	4	Seleciona turno	
	5		Verifica se na fila de espera do turno de origem existe algum aluno pertencente ao turno de destino
	6		Remove o "outro" aluno – o que pertença ao turno de destino e que se encontre na menor posição da fila de espera do turno de origem - de eventuais filas de espera relativas à mesma UC
	7		Remove o outro aluno do seu turno de origem (turno de destino do aluno que pretende efetuar
	8		Remove o aluno de eventuais filas de espera de outros turnos da mesma UC em que possa estar incluído.
	9		Remove o aluno do turno de origem
	10		Coloca o outro aluno no turno de origem
	11		Coloca o aluno no turno de destino
	12		Informa que a troca foi efetuada com sucesso
Exceção 1 (passo 5) [na fila de espera do turno de origem não existe nenhum aluno pertencente ao turno de destino]		Actor Input	System Response
	1		Coloca o aluno no final da fila de espera do turno destino
	2		Informa que o aluno foi colocado na fila de espera do turno destino

Preconditions		O Utilizador está autenticado, tem um horário e é do tipo Aluno TE	
Postconditions		O Aluno TE efetua uma troca de turno com sucesso	
		Actor Input	System Response
Flow of Events	1		Mostra a lista de UCs que o aluno está inscrito
	2	Seleciona unidade curricular (UC)	
	3		Mostra os turnos dessa UC
	4	Seleciona o turno de destino	
	5		Verifica se o turno de destino tem capacidade
	6		Remove o aluno de eventuais filas de espera de outros turnos da mesma UC em que podesse estar contido
	7		Remove o aluno do turno de origem
	8		Insere o aluno no turno de destino
			Informa que a troca de turno foi efetuada com sucesso
Alternativa 1 (passo 4) [turno não tem capacidade]		Actor Input	System Response
	1		Verifica se na fila de espera do turno de origem existe algum aluno que pertence ao turno de destino
	2		Seleciona o aluno que pertence ao turno de destino com maior prioridade nessa fila de espera
	3		Remove esse aluno de eventuais filas de espera de outros turnos da mesma UC em que podesse estar contido
	4		Remove esse aluno do turno de destino
	5		Remove o aluno TE de eventuais filas de espera de outros turnos da mesma UC em que podesse estar contido
	6		Remove o aluno TE do turno de origem
	7		Insere esse aluno no turno de origem
	8		Volta ao passo 7
Exceção 1 (passo 4.1) [na fila de espera do turno de origem não existe nenhum aluno pertencente ao turno de destino]		Actor Input	System Response
	1		Coloca o aluno TE na fila de espera do turno de destino com prioridade acima de todos os alunos do tipo normal
	2		Informa que o aluno está em fila de espera do turno de destino

Name		Troca de turno (efetuada por um professor)	
Brief Description		Professor troca um aluno de turno	
Preconditions		Está autenticado pelo sistema como um professor e já existem horários atribuídos aos alunos	
Postconditions		É efetuada uma troca de turno de um aluno por parte do professor da UC	
		Actor Input	System Response
Flow of Events	1		Mostra a lista de UCs que o professor leciona
	2	Seleciona UC	
	3		Mostra a lista de alunos inscritos a essa UC
	4	Seleciona o primeiro aluno a trocar	
	5	Seleciona o segundo aluno a trocar	
	6		Verifica os turnos dos dois alunos selecionados
	7		Remove o primeiro aluno do seu turno de origem
	8		Remove o primeiro aluno de eventuais filas de espera da UC em que podesse estar contido
	9		Remove o segundo aluno do seu turno de origem
	10		Remove o segundo aluno de eventuais filas de espera da UC em que podesse estar contido
	11		Inserir o primeiro aluno no turno de origem do segundo aluno
	12		Inserir o segundo aluno no turno de origem do primeiro aluno
	13		Informa que a troca foi efetuada com sucesso
Exceção 1 (passo 6) [os dois alunos pertencem ao mesmo turno]		Actor Input	System Response
	1		Informa que os dois alunos pertencem ao mesmo turno

Estas especificações detalham os diferentes processos de troca de turno por parte de um aluno, aluno com estatuto especial e professor.

Decidimos recorrer ao uso de filas de espera, de modo a atender a pedidos de troca quando estes não são imediatamente possíveis. Assim, cada turno possui uma fila de espera na qual são colocados os alunos que pretendem trocar para esse turno, até haver um aluno desse turno que deseje trocar para outro, caso em que o primeiro aluno na fila de espera pertencente a esse turno será trocado com o aluno. Deste modo é aplicado um critério de justiça em que os alunos que solicitam a troca primeiro serão os primeiros a ser atendidos quando tal for possível, excetuando os alunos que possuam estatuto especial e que possuem prioridade em relação aos outros, mas que entre si mantêm a mesma relação, isto é: o primeiro a solicitar a mudança de turno será o primeiro a ser atendido. O processo é mais simples para os professores; estes podem trocar alunos diretamente, sem serem colocados em filas de espera e removendo-os de qualquer fila de espera de turnos dessa UC em que estejam inseridos.

Interface

Página inicial do aluno

Sistema de Gestão de Trocas			- X
→ Início	Primeiro Ano	UC	<input type="text"/>
	Semestre	Análise	
	1º		
Trocar Turma			Inscrição
	Segundo Ano	UC	
Gestão de Pedidos	Semestre	Sistemas Operativos	
	2º		Inscrição
Lg out			

Sistema de Gestão de Trocas					X
	Ano	Semestre	UC	Turno	
Início	1º	1º	Cálculo	TP1	
				Trocar	
→ Trocar Turno			0		
			0		
			0		
			0		
Gestão de Pedidos			0		
			0		
			1		
			0		
Log out			0		
			0		
			0		

Página de Gestão de Pedidos de troca do aluno

Sistema de Gestão de Turmas				- X
	UC	Tamanho de fila	Posição	
Início	Cólab	10	3	
				Gein
Trocar Turma				
→ Gestão de Pedidos				
Log out				

Página para trocar de turno do aluno

Troca de Turno	- X
Cálculo - TP2	
Trocar para	<input type="text"/>
<input type="button" value="OK"/>	<input type="button" value="Cancelar"/>

Página para gerir um pedido do aluno

Gerir Pedido	-X
Origem	
TP1	16-18h 5 ^a feira
Destino	
TP4	9-11h 3 ^a feira
<input type="checkbox"/> OK	<input type="checkbox"/> Cancelar Pedido

Página de modificação de turno

Modifican Turno		— X	
Cálculo - TP2			
Capacidade	<input type="text" value="25"/>		
Remover	<input type="text"/>	<input type="checkbox"/>	
Adicionar	<input type="text"/>	<input type="checkbox"/>	
Trocar	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
		<input type="button" value="OK"/>	<input type="button" value="Cancelar"/>

Página inicial da direção de curso

Sistema de Gestão de turnos		- X
→ Carregar ficheiro	Ficheiro: <input type="text"/>	<input type="button" value="Browse"/>
	<input type="button" value="Upload"/>	
Compartilhar aulas		
Atribuir turnos		
Log out		

Análise crítica

Nesta fase de levantamento de requisitos, tentámos ao máximo especificar, com detalhe, todas as funcionalidades que esperamos que a nossa aplicação irá suportar. Começou-se por definir o domínio da nossa aplicação, caracterizando cada uma das diferentes entidades envolvidas e descrever como é que estas se relacionam entre si. De seguida caracterizamos cada um dos diferentes atores que podem entrar em contacto com a nossa aplicação (direção de curso, professor e aluno). Para cada um destes atores foi especificado com detalhe o modo como eles interagem com o sistema. De todas as interações possíveis entre os diferentes atores e a aplicação enumerámos um conjunto de requisitos necessários para o bom funcionamento da aplicação. Entre estes salientámos a funcionalidade de filas de espera para cada turno que consiste num componente da aplicação bastante importante para resolver as trocas de turnos entre alunos.

Finalmente, apresentámos uma proposta da interface da aplicação para cada interação entre os diferentes atores e a aplicação. Optámos por uma interface o mais intuitiva possível, que seja fácil de usar e que permita que todas as funcionalidades da aplicação sejam usufruídas sem necessidade de abrir, recorrentemente, novas janelas. Este assunto foi tomado em conta, já que uma aplicação com demasiadas janelas torna-se muito difícil de usar e tende a tornar-se incómodo para quem a utilizar.

Concluindo, consideramos que esta fase de levantamento de requisitos foi importante de modo a definir e caracterizar bem o domínio, bem como todas as funcionalidades de uma aplicação antes de a implementar. Isto irá permitir que na fase da implementação sejam evitados muitos problemas que poderiam vir a surgir caso a aplicação fosse implementada sem um planeamento prévio.