



Universidade do Minho

UMinho

Mestrado Engenharia Informática Requisitos e Arquiteturas de Software (2023/24)

PROBUM v.1

Grupo 1.4, PL1/Entrega 1

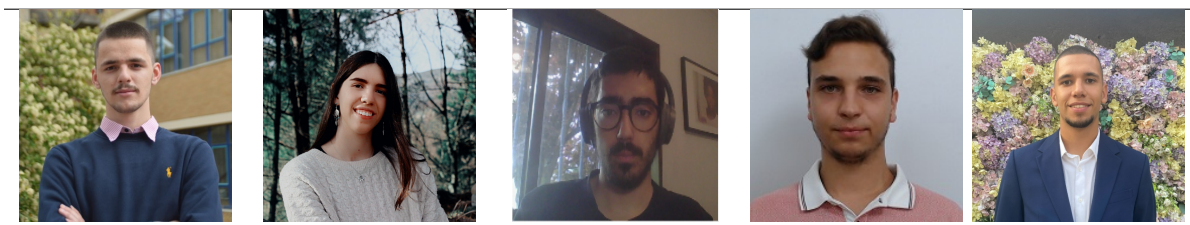
Diogo Filipe da Costa Marques, PG52678

Joana Isabel Freitas Pereira, PG53895

João Afonso Alvim Oliveira Dias de Almeida,
PG53902

João Miguel Pinheiro Machado, PG53926

Tiago Luís Pereira Ferreira, PG54256



Paulo R. Sousa e João M. Fernandes

Braga, 20 de outubro de 2023

Prefácio

Este documento resulta de um esforço conjunto dedicado à primeira fase de criação do produto de software Probum. Este projeto nasceu da visão e do apoio do Reitor da Universidade de Vigo, na Espanha, com o objetivo de facilitar a realização de provas de avaliação académicas no ensino superior.

O Probum é um sistema projetado para atender às necessidades das instituições de ensino superior, permitindo que alunos de diferentes cursos e unidades curriculares realizem provas académicas de forma eficiente e flexível. Um dos principais desafios enfrentados pelas instituições de ensino superior é a gestão de provas em grande escala, e o Probum visa resolver esses desafios.

Este sistema é uma solução inovadora que permite que os professores criem provas de avaliação, agendem datas de realização e, o mais importante, oferece a capacidade de correção, muitas vezes de forma automatizada. Isso não apenas economiza tempo, mas também promove a sustentabilidade, eliminando o desperdício de papel usualmente associado às avaliações tradicionais.

Destacamos o compromisso com a qualidade, eficiência e inovação que orientam o desenvolvimento do Probum. Esperamos que este sistema contribua para a melhoria da experiência académica de todos os envolvidos.

Agradecemos a todos os que contribuíram para este projeto e a todos os que agora embarcam nesta jornada de início de descoberta do Probum.

O contributo de cada elemento do grupo foi essencial no desenvolvimento do trabalho. No entanto, este deve ser analisado de forma a criar uma diferença entre elementos que contribuíram significativamente acima da média (classificado com '+') e os que, por outro lado, contribuíram abaixo da média (classificado com '-'). Se o contributo do aluno se encontrar dentro da média, é classificado com '0':

- Diogo Marques: 0
- Joana Pereira: 0
- João Almeida: 0
- João Machado: 0
- Tiago Ferreira: 0

Resumo

Este documento surge na sequência do contacto do Reitor da Universidade de Vigo (Espanha) com o objetivo de criar um produto de software que permita a realização de provas de avaliação académicas.

Este produto, que se designa Probum, permite que alunos de uma dada unidade curricular de um curso universitário ou politécnico (i.e. do ensino superior) realizem as suas provas académicas, utilizando as infraestruturas informáticas da sua própria instituição de ensino superior (IES), mesmo que estas sejam muito limitadas quanto à sua dimensão, disponibilidade e capacidade. Assim, o Probum deve incluir requisitos funcionais que permitam a sua utilização em diversas IES e permitir alguma parametrização e configuração. No essencial, o Probum permite que os professores criem provas de avaliação e que as calendarizem, que os alunos as realizem (de forma devidamente calendarizada) e que essas provas sejam corrigidas, tendencialmente de forma automática.

Documento criado por Frederico Dias para a empresa XPTO.

Introdução

Este relatório de serve de suporte ao desenvolvimento do projeto Probum, uma solução de software concebida para revolucionar a realização de provas de avaliação acadêmicas no ensino superior. O presente documento representa uma etapa importante no ciclo de vida deste projeto, fornecendo uma visão abrangente do que foi alcançado e do que está por vir.

Ao longo das próximas páginas, exploraremos as diversas fases e componentes que compõem o Probum. Começando pelo nosso modelo de domínio, que é a espinha dorsal do sistema, apresentaremos as principais entidades, relações e elementos que moldam nossa solução. Isso proporcionará uma compreensão sólida do contexto em que o Probum opera.

Ao longo deste relatório, abordaremos detalhadamente o propósito do projeto, delineando a sua razão de ser e a lacuna que visa preencher no cenário das avaliações acadêmicas no ensino superior. Discutiremos os stakeholders envolvidos no projeto, destacando a importância de sua contribuição e envolvimento contínuo.

Com uma ênfase particular nos utilizadores, exploraremos como o Probum atende às necessidades dos docentes, alunos e técnicos. Discutiremos as restrições e desafios que enfrentamos, bem como a taxonomia que guia a nossa estruturação de informações em questão.

Em seguida, mergulharemos no âmbito do produto, apresentando uma visão geral e o seu potencial para transformar a realização de provas acadêmicas. Exploraremos requisitos funcionais e não funcionais que moldam o comportamento do Probum, fornecendo diretrizes para seu desenvolvimento e utilização.

O processo de levantamento de requisitos, incluindo as técnicas utilizadas, será também analisado. Isso proporcionará insights sobre como as necessidades dos utilizadores foram identificadas e documentadas de maneira abrangente.

No âmbito das atividades e comportamento do sistema, apresentaremos o exemplo de um diagrama de atividades, que descreve os principais fluxos de ações e interações do Probum. Além disso, o diagrama de máquinas de estados revelará como o sistema responde a eventos e transições de estados.

Por fim, concluiremos este relatório com uma reflexão sobre o progresso feito até o momento e uma visão do que está por vir no desenvolvimento do Probum.

1. Propósito do Projeto

Contexto

Nos últimos anos, tem-se assistido a uma crescente integração da tecnologia no ambiente educativo e esse fenómeno não passa despercebido no ensino superior. Com o objetivo de otimizar processos e melhorar a experiência de Alunos e Docentes, muitas organizações têm explorado a digitalização de várias atividades, incluindo a realização de provas de avaliação.

Um exemplo interessante desta tendência é aquele que se observa em algumas provas de aferição do ensino básico português, que tentam explorar o potencial de transformação que as soluções digitais podem proporcionar. No entanto, aplicar no imediato estes métodos numa instituição de ensino superior levanta novos desafios, que estão essencialmente relacionados com as limitações das estruturas informáticas. O crescente número de Alunos inscritos em muitas unidades curriculares é também um obstáculo à aplicação desses métodos. Dentro desta realidade, destacam-se os seguintes desafios:

- **Gestão de espaços e recursos:** A realização de provas de avaliação para um grande número de Alunos coloca uma pressão significativa sobre os espaços disponíveis. As salas revelam-se muitas vezes insuficientes, resultando em desafios logísticos na calendarização e alocação destes espaços.
- **Eficiência na correção:** A correção manual de um grande número de provas é um processo demorado e sujeito a erros humanos. Fazê-la, especialmente em cursos com muitos Alunos, torna-se ineficiente e onerosa em termos de tempo e custo.
- **Sustentabilidade ambiental:** O uso de papel na impressão e resposta às provas de avaliação resulta num desperdício muito significativo, que tem um impacto negativo no ambiente. A necessidade de encontrar alternativas mais sustentáveis para a realização das provas é um aspeto crítico.

Objetivos do Projeto

Para responder a estes desafios, propomos o desenvolvimento de um produto, o Probum, que tenha algumas características diferenciadoras e até inovadoras, que aborda todos estes problemas, relacionados com a realização de provas de avaliação, de forma integrada. O Probum permitirá:

- **Criação de provas de avaliação digitais:** Criar provas de avaliação personalizadas será simples e fácil e permitirá incorporar questões de diferentes tipos, temas e níveis de dificuldade. Deve ser possível criar questões alternativas, com respostas também diferenciadas, se for conveniente cada Aluno ter uma prova diferente das dos restantes colegas.
- **Calendarização inteligente das provas:** Utilizando ferramentas matemáticas (e.g., algoritmos de otimização), o Probum poderá gerar automaticamente um calendário para uma dada prova de avaliação, tendo em consideração os espaços disponíveis, o número de

Alunos inscritos, e as necessidades específicas de cada unidade curricular ou curso. Se uma dada prova tiver, por exemplo, 96 Alunos inscritos e a sala para a realização dessa prova só tiver capacidade para 20 Alunos, então têm que ser organizadas pelo menos 5 rondas de realização do teste. Este facto pode obrigar a que as provas não sejam iguais para todos os Alunos, como atrás se referiu.

- **Realização das provas:** Um Aluno poderá realizar as provas de avaliação de forma digital, através de uma plataforma dedicada, proporcionando assim uma experiência simples, robusta, e flexível. Esta plataforma será disponibilizada em espaços e equipamentos da própria IES, garantindo todos os requisitos exigidos (e.g., autenticidade, confidencialidade, equidade, duração, plágio). Deve ser analisada e explorada a possibilidade de compaginar a realização de provas em computadores da IES com provas realizadas em computadores dos Alunos, desde que se mantenham todas as garantias de confidencialidade e autenticidade.
- **Correção:** Na fase de correção, todas as questões de resposta fechada serão facilmente avaliadas e pontuadas de forma automática. O Probum deverá poder ser estendido com componentes baseados em técnicas de processamento de linguagem natural, para auxiliar a correção de questões de resposta livre. Se tal funcionalidade não estiver disponível, cabe ao Docente pontuar essas questões abertas.
- **Sustentabilidade:** A substituição do papel pelo produto contribuirá para a sustentabilidade ambiental. Além disso, evitam-se sobras de papel, i.e., as cópias da prova de avaliação que se tiraram a mais devido ao facto de o número de Alunos que apareceram para a realizar ser menor que o esperado.

Se assumirmos que:

- um curso (de 3 anos) tem 6 unidades curriculares em cada semestre (ou seja 36 unidades curriculares);
- 90% das provas de avaliação desse curso podem ser realizadas de forma digital;
- em cada unidade curricular se realizam 3 provas de avaliação, com uma média de 100 Alunos por prova;
- cada Aluno consome 4 folhas de papel A4 em cada prova de avaliação em que participa;

então, em cada ano letivo, são poupadas 38.800 folhas de papel por curso, se for usado o produto Probum. Se cada impressão custar 0,05 EUR, então há uma poupança de quase 4.000 EUR (assume-se que cada folha é impressa em ambos os lados). Se uma universidade tiver o equivalente a 100 cursos destes, então a poupança cifra-se em 400.000 EUR anuais. Trata-se de um número bem expressivo que mostra a relevância deste produto. A isto há ainda que acrescentar os ganhos de tempo de correção, aspecto que não pode ser negligenciado e que consome muito do tempo de um professor nos períodos, por vezes longos, em que tem de corrigir provas.

- **Consulta das provas:** Assim que todas classificações de uma dada prova forem divulgadas, o Docente poderá permitir que cada Aluno consulte a sua prova. Cada prova (que possa ser consultada) ficará disponível, no perfil do respetivo Aluno, para consulta, durante dois anos, altura em que poderá fazer sentido eliminá-la.

2. Cliente, Consumidor e *Stakeholders*

Ao longo do desenvolvimento deste projeto verificamos a existência de diversas partes envolvidas, nomeadamente: clientes, consumidores e *stakeholders*.

- **Clientes:** Os nossos clientes serão os departamentos pedagógicos de IESs com a necessidade de disponibilizar aos seus corpos docente e discente um sistema informático que automatize a criação, calendarização, realização e correção de provas de avaliação (escritas e individuais).
- **Consumidores:** Os consumidores do nosso produto são os docentes de ensino superior. Eles estão interessados numa ferramenta informática que facilite a criação, distribuição e correção das provas de avaliação, ao mesmo tempo que os mantém capacitados para a resolução de situações anómalas durante as suas realizações. Será através dos docentes que devemos definir como deverá ser a dinâmica de criação, calendarização, realização e correção das provas de avaliação. Durante o desenvolvimento do projeto, os nossos clientes participarão de forma ativa no planeamento e validação da solução.
- **Outros *stakeholders***
 - Alunos: Uma das principais partes interessadas no nosso produto são os estudantes universitários, que engloba todas as pessoas que estão inscritas num curso superior e que têm que realizar provas de avaliação. Pode considerar-se que as suas idades são superiores (ou iguais) a 18 anos, e que estão familiarizados com o uso de ferramentas informáticas. Deve ser-lhes disponibilizado um produto simples, eficiente, robusto, amigável, e tolerante a falhas para que possam realizar as suas provas.
 - Técnicos: Os técnicos de informática são também fundamentais para o funcionamento do produto. São eles que instalam e fazem a manutenção da plataforma nos equipamentos informáticos da IES. Eles esperam que o produto seja fácil de instalar e configurar, e que disponibilize métricas e *logs* que permitam antecipar e diagnosticar eventuais problemas.

3. Utilizadores do Produto

Nesta secção apresentam-se todos os utilizadores que utilizarão efetivamente o produto, listando-se as funções de cada um.

Docente

- **Função:** Responsável por contactar com os alunos, criar provas de avaliação, garantir que as mesmas se realizam nos espaços da sua IES e coordenar o processo de correção (quando necessário).
- **Experiência no contexto:** mestre
- **Experiência tecnológica:** mediana

Aluno

- **Função:** Responsável por dar respostas às questões das provas de avaliação em que se apresentar.
- **Experiência no contexto:** mediana
- **Experiência tecnológica:** mediana

Técnico

- **Função:** responsável por instalar a plataforma nos equipamentos informáticos da IES e mantê-la em funcionamento.
- **Experiência no contexto:** mediana
- **Experiência tecnológica:** mestre

Prioridades atribuídas aos Utilizadores

- **Utilizadores principais:** Docente e Aluno
- **Utilizadores secundários:** Técnico

O sucesso do produto depende diretamente dos Docentes e dos Alunos, e, por esse motivo, é nas suas necessidades e expectativas que se deve focar o esforço de levantamento de requisitos. Os Técnicos têm relevância nos requisitos, no entanto, caso estes colidam com os dos Docentes e Alunos, será dada preferência a estes.

4. Restrições do Projeto

Restrições à Solução

Requisito #:	Rest1	Tipo:	Restrição	Use cases #:	n.a.
Descrição	A aplicação deve executar na infraestrutura atual da respectiva IES				
<i>Rationale</i>	Para que não seja necessário investir em novo equipamento				
Origem	Cliente				
<i>Fit criterion</i>	Todos os componentes de software devem estar instalados em máquinas da IES e todas as funcionalidades da plataforma para os Alunos devem executar em pleno nas máquinas que que forem disponibilizadas para as provas de avaliação				
Prioridade	Must				

Tabela 1: Restrição quanto à infraestrutura informática

Requisito #: Rest2	Tipo: Restrição	Use cases #: n.a.
Descrição	O computador disponibilizado a cada Aluno, no momento em que realiza uma prova de avaliação, apenas deve permitir acesso ao Probum	
<i>Rationale</i>	Para evitar que o Aluno recorra a outras aplicações (email, navegadores web, WhatsApp, Skype, etc.) durante a realização da sua prova de avaliação; O produto tem que estar preparado para funcionar em computadores instalados em salas da IES	
Origem	Cliente	
<i>Fit criterion</i>	Enquanto uma prova de avaliação estiver a decorrer não deve ser possível aceder a nenhuma outra aplicação que não o Probum	
Prioridade	Must	

Tabela 2: Restrição quanto ao isolamento da aplicação de resposta às provas

Restrições Temporais

- **Descrição:** O documento presente terá de ser entregue, numa fase inicial, até dia 20 de outubro de 2023.

Justificação: De forma a poder ser avaliado o estado do projeto numa fase inicial, é necessário que seja feita uma entrega que contenha a primeira fase deste projeto, que abarca a contextualização e a definição dos requisitos da solução.

Restrições Orçamentais

- **Descrição:** O orçamento total para o desenvolvimento do projeto é de 20 000€ (vinte mil euros), durante um período de 4 meses.

Justificação: A equipa responsável pelo desenvolvimento do projeto é constituída por quinze engenheiros de *software*. Para além de ter em conta os salários dos elementos, é preciso também a compra de um domínio, bem como de um computador para hospedar todos os dados da aplicação.

5. Taxinomia e definições

Aluno

Ator do sistema. Responsável por se autenticar e responder às Questões de uma Prova; O mesmo que estudante ou discente.

Classificação

O processo de avaliação e atribuição de pontuações ou notas às respostas dos alunos em uma Prova de Avaliação. A classificação é realizada pelo Docente ou pelo Sistema com base nos critérios de avaliação definidos. Pode incluir a revisão e atribuição de pontuações às respostas dos alunos, bem como o cálculo da pontuação total da prova.

Computador

Equipamento informático no qual um Aluno dá as Respostas às Questões da sua Prova. Ainda que eventualmente intermitente, estes equipamentos têm ligação à *intranet* da IES.

Correção

A correção envolve a ação de analisar minuciosamente as respostas dos alunos em uma prova para identificar e avaliar erros, imprecisões ou incoerências, de acordo com os critérios de avaliação estabelecidos.

Critério de Avaliação

Os padrões ou diretrizes específicos usados para avaliar e pontuar as respostas dos alunos em uma prova. Esses critérios são essenciais para garantir a objetividade e consistência na avaliação das respostas dos alunos.

Docente

Ator do sistema. Responsável por criar e dar seguimento a uma prova. Na criação de uma Prova, entre outras tarefas, define as Questões e os respetivos Critérios de Avaliação. O mesmo que professor.

Erro

Condição ou situação anormal que ocorre durante a execução do Probum. Essas condições anormais podem resultar em comportamentos indesejados ou não planeados e podem afetar o funcionamento correto do software.

Instituição de Ensino Superior

Organização de ensino que tem interesse em disponibilizar a plataforma aos seus docentes, para que seja possível utilizar a infraestrutura informática já existente para a realização de provas de avaliação; O mesmo que Universidade ou Instituto Politécnico; Sigla: IES.

Prova de avaliação

Avaliação de competências e de conhecimentos, em que cada Aluno responde, de forma individual e durante um período de tempo previamente estabelecido, a um conjunto de Questões, que foram previamente preparadas pelos Docentes; O mesmo que teste escrito, prova escrita ou exame.

Questão

Elementos da prova criados pelos docentes para testar os conhecimentos e competências dos alunos. Elas podem abranger vários tipos, temas e níveis de dificuldade. As questões são apresentadas aos alunos durante a prova, e os alunos devem fornecer respostas apropriadas com base nas instruções fornecidas em cada questão.

Resposta

As respostas consistem nas soluções ou soluções parciais fornecidas pelos alunos às questões da prova. Essas respostas podem abranger desde textos escritos até seleção de escolha múltipla. Estas podem posteriormente ser corrigidas pelos docentes ou pelo sistema, a fim de determinar a pontuação final da prova.

Sala

Espaço físico equipado com computadores, onde se realizam as Provas. Geralmente têm capacidade para um número reduzido de Alunos em simultâneo (por exemplo, cerca de 20).

8. Âmbito do Produto

Atores

Como está representado no diagrama anterior, o nosso sistema suporta quatro tipos de atores: o *Docente*, o *Técnico*, o *Aluno*, e o *Sistema*.

O *Docente* é o utilizador central do nosso sistema, pois é ele que vai gerir todo o conteúdo e funcionamento da plataforma.

O *Aluno*, por sua vez, é o utilizador mais crítico e sensível do sistema, dado o contexto em que utilizará a plataforma.

O *Técnico* é o indivíduo que ajudará a instalar, configurar, e manter a plataforma nos servidores e Computadores da IES.

O *Sistema* é uma ou mais peças de *software* cujo propósito é automatizar processos e tarefas, tais como a correção total ou parcial de Respostas.

Modelo de Domínio

Consideramos também pertinente desenvolver o modelo de domínio, um modelo abstrato capaz de representar os comportamentos e informações gerais da nossa aplicação e que nos ajudará a estabelecer quais serão as entidades principais e relações do sistema "Probum". Este diagrama, apesar de simples, desempenha então um papel crucial na representação visual das principais entidades e relações no contexto do *Probum*. Ele ajuda a simplificar ideias complexas e a estabelecer uma base sólida para análises posteriores. O Diagrama de Domínio permite-nos compreender o ecossistema do *Probum*, destacando as interações entre professores, alunos e o sistema. É uma ferramenta fundamental para perceber o problema e as suas necessidades, proporcionando uma visão geral do ambiente de avaliação flexível e altamente personalizável que o *Probum* oferece.

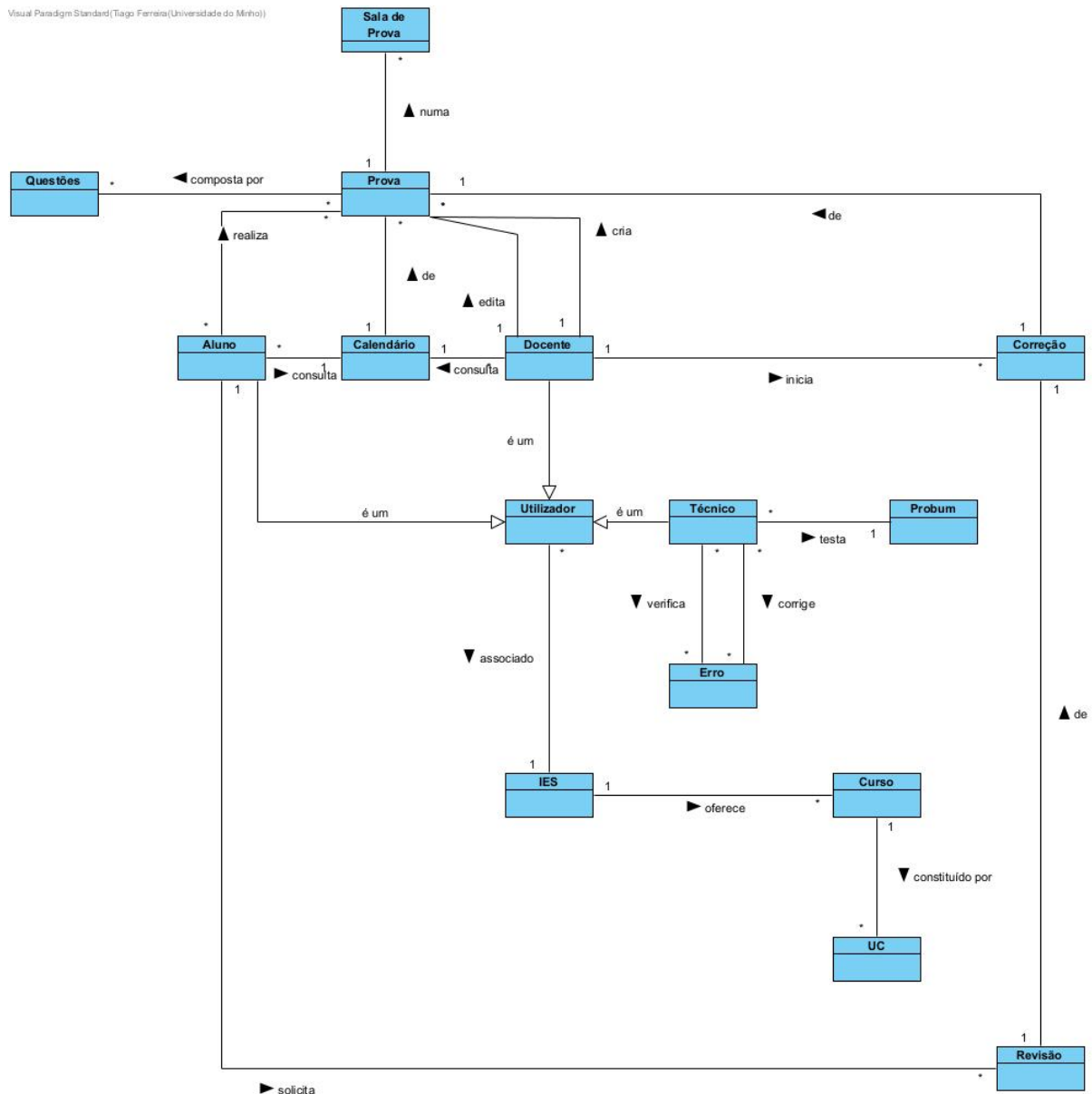


Figura 1: Modelo de Domínio do Sistema

- **Utilizador:** O utilizador é a entidade central do sistema "Probum". Pode representar diferentes tipos de utilizadores, incluindo Docentes, Técnicos e Alunos.
- **Instituição de Ensino Superior (IES):** Uma Instituição de Ensino Superior (IES) representa uma instituição académica, como uma universidade ou politécnico. Cada IES está associada a vários cursos e unidades curriculares.
- **Curso:** Um curso representa um programa de estudo específico oferecido por uma IES. Cada curso está associado a várias unidades curriculares.
- **Unidade Curricular:** Uma unidade curricular representa uma disciplina académica de um curso. Cada unidade curricular está associada a vários exames.
- **Exame:** Um exame representa uma avaliação específica. Cada exame é criado por um Docente e está relacionado a uma Unidade Curricular. Pode haver várias instâncias desse exame, cada uma associada a um Aluno.

- **Aluno:** Um Aluno é um indivíduo matriculado numa Unidade Curricular, pertence a uma IES e é inscrito para realizar exames.
- **Prova:** Uma prova representa uma instância específica de um exame realizado por um Aluno. A prova contém as respostas dos Alunos e informações relacionadas à sua calendarização, bem como, posteriormente, uma correção.
- **Sala:** Uma sala de prova é um local onde os exames são realizados. Cada sala tem uma capacidade máxima de alunos que podem ser acomodados.
- **Calendário:** O Calendário de Exames é uma entidade que contém informações sobre as datas e horários dos exames. É gerado automaticamente com base nas informações sobre exames, salas e disponibilidade.
- **Correção:** A Componente de Correção representa um módulo ou componente do sistema que auxilia na correção de exames, especialmente questões de resposta livre. Pode ser estendido com técnicas de processamento de linguagem natural.

Diagrama de Use Cases

Após a criação do Diagrama de Domínio, seguimos para a elaboração de um Diagrama de Use Cases. Este diagrama permite identificar os principais atores e as funcionalidades do Probum. Ele ajuda a visualizar como os utilizadores interagem com o sistema e quais ações podem ser realizadas. Assim, posteriormente, serão efetuados de forma mais clara os requisitos funcionais e não funcionais do Probum, ou seja, as funcionalidades que o software deve oferecer para atender às necessidades dos professores e alunos. Isso nos ajuda a construir um software eficiente e amigável, alinhado com as expectativas dos utilizadores.

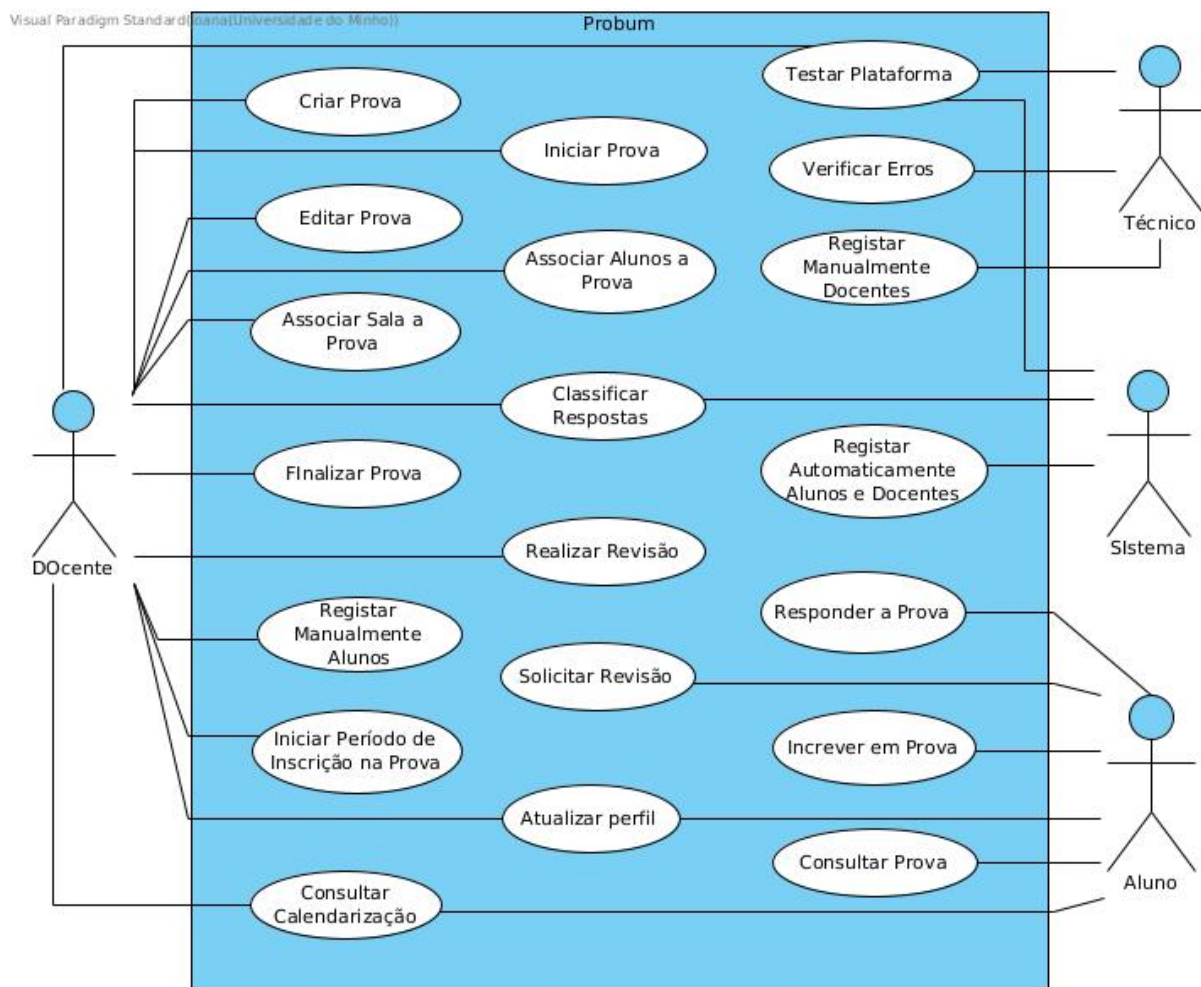


Figura 2: Diagrama de Use Cases

Breve Descrição dos *Use Cases*

Esta secção apresenta uma especificação tabellar de cada *use case* considerado, de modo a facilitar o processo de implementação de cada funcionalidade do nosso sistema.

Deste modo, consideramos que é bastante perceptível o fluxo sequencial da interação do ator com o sistema.

Testar plataforma

O *use case* "Testar plataforma", cujo o ator principal é o *Docente* ou o *Técnico* e o ator secundário é o *Sistema*, consiste na execução de uma série de testes automatizados da plataforma *Probum* para verificar o bom funcionamento da mesma na administração de uma prova assim como a sua correção, para confirmar que não existam eventuais problemas durante ou após a realização de uma prova. Caso haja algum problema é criado um aviso e um *log* do erro em questão para que posteriormente o problema seja resolvido, caso contrário o Sistema informa que a Plataforma está a funcionar devidamente.

<i>Use case</i>	1	
Ator principal	Docente, Técnico	
Ator secundário	Sistema	
Pré-Condições	O Docente/Técnico está autenticado na plataforma.	
Pós-Condições	Plataforma é testada e dá o seu estado de funcionamento.	
Cenário Normal	Input do Ator	Resposta do Sistema
	1 - O Docente/Técnico seleciona a opção para testar a Plataforma. 4 - O docente recebe uma notificação informando-o do resultado da testagem.	2 - O Sistema corre uma série de testes automatizados previamente definidos. 3 - O Sistema recebe e verifica os resultados dos testes e gera uma notificação.
Exceção 1 - Resultado de um teste é inválido - Passo 4	3.3 - O Docente/Técnico recebe a notificação que o teste não foi bem sucedido e o ficheiro com os <i>logs</i> .	3.1 - O Sistema verifica que um ou mais resultados dos testes não é válido. 3.2 - O Sistema gera um ficheiro de <i>logs</i> e envia o ficheiro juntamente com a notificação que a testagem foi mal sucedida.

Tabela 3: Especificação do *use case* "Testar Plataforma"

Editar Prova

O *use case* "Editar Prova", cujo ator principal é o *Docente*, consiste na edição de uma prova de avaliação já criada. Esta edição implica a mudança de todos os parâmetros da Prova, como as questões da prova, que à altura da sua criação, possam conter gralhas ou outro tipos de erros (e.g., formatação, interpretação), data, hora ou até mesmo a lista de alunos inscritos.

Caso o erro seja encontrado durante a realização da prova, o *Docente* será capaz de também proceder à edição das questões, sendo que o sistema sincroniza automaticamente a prova para todos os alunos.

<i>Use case</i>	2	
Ator principal	Docente	
Ator secundário	-	
Pré-Condições	O Docente está autenticado na plataforma.	
Pós-Condições	Prova de avaliação é modificada.	
Cenário Normal	Input do Ator	Resposta do Sistema
	1 - O Docente clica na lista de Provas disponíveis.	2 - O Sistema mostra Provas disponíveis.
	3 - O Docente seleciona a Prova que quer modificar.	4 - O Sistema mostra a Prova selecionada justamente com um menu com as opções disponíveis.
	5 - O docente seleciona a opção que pretende modificar.	6 - O Sistema regista a opção selecionada e pede os dados necessários associados à opção selecionada.
	7 - O Docente fornece os dados pedidos pelo Sistema.	8 - O Sistema modifica a Prova de acordo com os dados fornecidos.
	9 - O Docente volta ao passo 5.	
Exceção 1 - Informações inválidas - Passo 8		8.1 - O Sistema informa o Docente que as informações fornecidas são inválidas e fornece orientações para corrigi-las e volta ao passo 7.
Alternativa 1 - Voltar à lista de Provas - Passo 9	9 - O Docente seleciona a opção de sair da Prova selecionada e voltar para a lista de Provas. 10 - O Docente volta ao passo 3.	

Tabela 4: Especificação do *use case* "Editar Prova"

Registar Automaticamente Alunos e Docentes

O *use case* "Registar Automaticamente Alunos e Docentes", cujo ator principal é o *Sistema*, consiste no registo automático de alunos e docentes na plataforma, permitindo acesso às funcionalidades da plataforma. Este registo é *triggered* pelo sistema quando é feito um novo registo de um docente ou aluno, no Sistema de gestão de base de dados da *IES* sendo este que automaticamente manda os dados para o Probum para ele registar os novos utilizadores.

<i>Use case</i>	3	
Ator principal	Sistema	
Ator secundário	-	
Pré-Condições	O Sistema está operacional e a base de dados está acessível.	
Pós-Condições	Atualização dos alunos e docentes registados .	
Cenário Normal	Input do Ator	Resposta do Sistema
	1 - O Sistema verifica se a base de dados contém novos alunos ou docentes. 2 - O Sistema manda os dados dos novos utilizadores para o Probum.	3 - O Sistema verifica as informações fornecidas. 4 - O Sistema regista os alunos na plataforma. 5 - O sistema gera senhas temporárias para as contas dos novos alunos. 6 - O sistema envia notificações para os novos alunos informando sobre a criação das suas contas e fornecendo informações de acesso temporárias.
	Alternativa 1 - Nenhum novo aluno ou docente - Passo 1	1.1 - O Sistema termina a verificação.
Exceção 1 - Informações inválidas - Passo 3		3.1 - O Sistema informa o Sistema que as informações fornecidas são inválidas e fornece orientações para corrigi-las num ficheiro de <i>logs</i> e termina o processo.

Tabela 5: Especificação do *use case* "Registar Automaticamente Alunos e Docentes"

Registar Manualmente Alunos

O *use case* "Registar Manualmente Alunos", cujo ator principal é o *Docente* ou o *Técnico*, permite que o *Docente/Técnico* registre alunos na plataforma para que possam acessar as funcionalidades.

<i>Use case</i>	4	
Ator principal	Docente/Técnico	
Ator secundário	-	
Pré-Condições	O Docente/Técnico está autenticado na plataforma	
Pós-Condições	Alunos são registrados na plataforma	
Cenário Normal	Input do Ator	Resposta do Sistema
	1 - O Docente/Técnico fornece informações dos alunos, como nome, curso, ano, etc.	2 - O Sistema verifica as informações fornecidas. 3 - O Sistema registra os alunos na plataforma. 4 - O sistema gera senhas temporárias para as contas dos novos alunos. 5 - O sistema envia notificações para os novos alunos informando sobre a criação das suas contas e fornecendo informações de acesso temporárias.
Exceção 1 - Informações inválidas - Passo 2		2.1 - O Sistema informa o Docente/Técnico que as informações fornecidas são inválidas e fornece orientações para corrigi-las e volta ao passo 1.

Tabela 6: Especificação do *use case* "Registrar Manualmente Alunos"

Registrar Manualmente Docentes

O *use case* "Registrar Manualmente Docentes", cujo ator principal é o *Técnico*, permite que o Técnico registre docentes na plataforma para que possam acessar as funcionalidades.

<i>Use case</i>	5	
Ator principal	Técnico	
Ator secundário	-	
Pré-Condições	O Técnico está autenticado na plataforma	
Pós-Condições	Docentes são registrados na plataforma	
Cenário Normal	Input do Ator	Resposta do Sistema
	1 - O Técnico fornece informações dos docentes, como nome, departamento, contato, etc.	2 - O Sistema verifica as informações fornecidas. 3 - O Sistema registra os docentes na plataforma. 4 - O sistema gera senhas temporárias para as contas dos novos docentes. 5 - O sistema envia notificações para os novos docentes informando sobre a criação de suas contas e fornecendo informações de acesso temporárias.
Exceção 1 - Informações inválidas - Passo 2		2.1 - O Sistema informa o Técnico que as informações fornecidas são inválidas e fornece orientações para corrigi-las e volta ao passo 1.

Tabela 7: Especificação do *use case* "Registrar Manualmente Docentes"

Atualizar Perfil

O *use case* "Atualizar Perfil" é acionado pelo Aluno ou pelo Docente e tem como finalidade permitir ao utilizador atualizar as informações do seu perfil na plataforma, incluindo a alteração de senha.

<i>Use case</i>	6	
Ator principal	Aluno/Docente	
Ator secundário	-	
Pré-Condições	O utilizador (Aluno ou Docente) está autenticado na plataforma	
Pós-Condições	As informações de perfil do utilizador são atualizadas	
Cenário Normal	Input do Ator	Resposta do Sistema
	1 - O utilizador Aluno/Docente fornece as informações que deseja alterar.	2 - O Sistema verifica as informações fornecidas. 3 - O Sistema atualiza as informações do perfil. 4 - O Sistema notifica o utilizador que as informações foram atualizadas com sucesso.
Exceção 1 - Informações inválidas - Passo 2		2.1 - O Sistema informa o utilizador que as informações fornecidas são inválidas e fornece orientações para corrigi-las e volta ao passo 1

Tabela 8: Especificação do *use case* "Atualizar Perfil"

Consultar Calendarização de Provas

O *use case* "Consultar Calendarização de Provas", com o ator principal sendo o Aluno ou Docente, permite que os utilizadores autorizados acessem a plataforma e visualizem a lista de provas criadas para o curso. Esta funcionalidade possibilita que os utilizadores conheçam as datas, horários e locais das provas (se já tiverem sido requisitados), facilitando o planeamento de seu calendário académico. O *use case* cobre tanto a situação normal, em que as provas estão agendadas e são exibidas ao utilizador, assim como a exceção de nenhum agendamento de prova disponível no momento.

<i>Use case</i>	7	
Ator principal	Aluno, Docente	
Ator secundário	-	
Pré-Condições	O Aluno/Docente está autenticado na plataforma	
Pós-Condições	O Aluno/Docente visualiza a calendarização de provas	
Cenário Normal	Input do Ator	Resposta do Sistema
	1 - O Aluno/Docente acessa a opção de consultar a calendarização de provas	2 - O Sistema exibe a lista de provas agendadas com detalhes (data, hora, local)
Exceção 1 - Nenhuma prova agendada - Passo 2		2.1 - O Sistema informa o Aluno/-Docente que não há provas agendadas no momento

Tabela 9: Especificação do *use case* "Consultar Calendarização de Provas"

Criar Prova

O *use case* "Criar Prova", cujo ator principal é o *Docente*, consiste na criação de uma nova prova de avaliação. Esta criação implica a introdução da duração e a criação de questões. Existe um cenário de exceção quando o sistema não consegue interpretar o ficheiro submetido pelo Docente. Nesse caso são devolvidos os detalhes do erro, para que o ficheiro possa ser retificado e posteriormente resubmetido.

<i>Use case</i>	8	
Ator principal	Docente	
Ator secundário	-	
Pré-Condições	O Docente está autenticado na plataforma	
Pós-Condições	A Prova está criada no sistema com algumas informações e questões registadas	
Cenário Normal	Input do Ator	Resposta do Sistema
	1- O Docente fornece a duração e o dia da realização da prova. 3- O Docente cria uma Questão, selecionando o seu tipo, tema, dificuldade e a sua descrição. 4 - O Docente define os Critérios de Avaliação da Questão. 6- O Docente volta ao passo 3.	2- O Sistema associar as informações à prova. 5 - O Sistema cria a Questão e adiciona-a à prova.
Alternativa 1 - terminado - Passo 5	5.1 - O Docente define o número de questões por tipo, tema e dificuldade que a prova irá ter.	

Tabela 10: Especificação do *use case* "Criar Prova"

Associar Alunos a Prova

O use case "Associar Alunos a Prova" em que o Docente é o ator principal, permite a inscrição manual dos alunos desejados a uma determinada prova.

<i>Use case</i>	9	
Ator principal	Docente	
Ator secundário	-	
Pré-Condições	A Prova está criada.	
Pós-Condições	Os alunos são associados manualmente pelo Docente a uma dada Prova.	
Cenário Normal	Input do Ator	Resposta do Sistema
	1 - O Docente fornece as informações dos alunos, como nome, curso, ano, etc., para associá-los à Prova.	2 - O Sistema verifica as informações fornecidas. 3 - O Sistema associa os alunos à Prova.
Exceção 1 - Informações inválidas - Passo 2		2.1 - O Sistema informa o Docente que as informações fornecidas são inválidas e fornece orientações para corrigi-las.

Tabela 11: Especificação do *use case* "Associar Alunos a Prova"

Iniciar Período de Inscrição na Prova

O *Use Case* "Iniciar Período de Inscrição na Prova" permite que o Docente abra um período de inscrição para uma prova previamente criada. Durante este período, os alunos podem se inscrever na prova.

<i>Use Case</i>	10	
Ator principal	Docente	
Ator secundário	-	
Pré-Condições	A prova existe.	
Pós-Condições	Alunos são permitidos a se inscreverem na Prova.	
Cenário Normal	Input do Ator	Resposta do Sistema
	1 - O Docente indica o período que deseja que a inscrição para a prova fique aberta.	2 - O Sistema notifica os alunos sobre a disponibilidade e o período da inscrição. 4 - Após o término do período indicado pelo Docente, o Sistema associa automaticamente os alunos inscritos à prova.

Tabela 12: Especificação do *Use Case* "Iniciar Período de Inscrição na Prova"

Inscrição na Prova

O caso de uso "Inscrição na Prova" permite que os alunos se inscrevam em uma prova previamente criada pelo Docente durante o período de inscrição.

<i>Use Case</i>	11	
Ator principal	Aluno	
Ator secundário	-	
Pré-Condições	O Docente criou uma prova e abriu um período de inscrição	
Pós-Condições	O Aluno está inscrito na prova	
Cenário Normal	Input do Ator	Resposta do Sistema
	1 - O Aluno acessa a plataforma e visualiza as provas disponíveis com período de inscrição aberto. 2 - O Aluno seleciona a prova desejada e inscreve-se.	3 - O Sistema confirma a inscrição.

Tabela 13: Especificação do *Use Case* "Inscrição na Prova"

Iniciar Prova

O *use case* "Iniciar Prova", cujo ator principal é o *Docente*, consiste no início de uma prova de avaliação.

Esta iniciação implica que o sistema seja capaz de verificar se todos os recursos necessários à realização da prova estejam disponíveis (e.g., o sistema a correr em todos os equipamentos, a comparência dos alunos nas salas). Uma vez iniciada a prova, o sistema regista o *timestamp* de início da prova, atribui aleatoriamente as questões para cada aluno e notifica os alunos que podem proceder à resolução da mesma.

Existem duas exceções neste *use case*:

- **Recursos Indisponíveis** - O sistema informa o Docente que alguns recursos necessários não estão disponíveis de momento e a prova não é possível ser iniciada.
- **Prova já Iniciada** - O sistema informa o Docente que a Prova já foi iniciada anteriormente.

Existe, por fim, uma alternativa neste *use case*, que é a cessação da prova, caso alguma circunstância alheia à mesma assim o justifique.

<i>Use case</i>	12	
Ator principal	Docente	
Ator secundário	-	
Pré-Condições	O Docente está autenticado na plataforma e a Prova está criada.	
Pós-Condições	A Prova é iniciada e os alunos podem responder à mesma.	
Cenário Normal	Input do Ator	Resposta do Sistema
	1 - O Docente seleciona a Prova que deseja iniciar.	2 - O sistema verifica a disponibilidade dos recursos necessários para a iniciação da prova. 3 - O sistema confirma que a prova pode ser iniciada.
	4 - O Docente inicia a prova.	5 - O sistema regista o <i>timestamp</i> do início da prova. 6 - O Sistema atribui aleatoriamente as questões para cada aluno e notifica os de que a prova foi iniciada.
Exceção 1 - Recursos indisponíveis - Passo 2		2.1 - O Sistema informa que alguns recursos não estão disponíveis de momento e volta ao passo 1.
Exceção 2 - Prova já iniciada - Passo 3		3.1 - O Sistema informa o Docente que a prova já foi iniciada anteriormente e volta ao passo 1.
Alternativa 1 - Abortar início - Passo 3	3.1 - O Docente aborta o início da prova e volta ao passo 1.	

Tabela 14: Especificação do *use case* "Iniciar Prova"

Finalizar Prova

O *use case* "Finalizar Prova", cujo ator principal é o *Docente*, consiste na finalização de uma prova de avaliação.

Esta finalização implica que o sistema verifique que a prova está em andamento e que, ou por decisão do docente, ou por outros fatores (e.g., fim do tempo estipulado para a realização da prova, todos os alunos terminarem a prova antecipadamente). Confirmando-se a finalização da prova, o sistema regista o *timestamp* de término da mesma.

Existe uma exceção neste *use case*, que é, tal como no *use case* anterior, o sistema informa o docente de que a prova já foi finalizada anteriormente.

Por fim, existe uma alternativa neste *use case*, que é a reiniciação da prova, caso o docente assim o deseje.

<i>Use case</i>	13	
Ator principal	Docente	
Ator secundário	-	
Pré-Condições	O Docente está autenticado na plataforma e a Prova está iniciada.	
Pós-Condições	A Prova é terminada e os alunos não podem responder a mais nenhuma pergunta.	
Cenário Normal	Input do Ator	Resposta do Sistema
	1 - O Docente seleciona a prova que deseja terminar.	2 - O sistema confirma que a prova está iniciada.
	3 - O Docente decide finalizar a prova.	4 - O sistema regista o <i>timestamp</i> de término da prova.
		5 - A prova é marcada como finalizada no sistema.
Exceção 1 - Prova já terminada - Passo 2		2.1 - O Sistema informa que a prova já foi terminada.
Alternativa 1 - Cancelar término - Passo 3	3.1 - O Docente volta ao passo 1.	
Alternativa 2 - Prova já terminada - Passo 10		4.1 - O sistema deteta que a prova tem de ser finalizada devido a diversos fatores e transita para o passo 4.

Tabela 15: Especificação do *use case* "Finalizar Prova"

Responder Prova

O *use case* "Responder Prova", com o ator principal sendo o Aluno, envolve a interação do aluno com o sistema para responder a uma prova de avaliação. Neste cenário, o aluno acessa uma prova agendada, visualiza as questões e fornece respostas às mesmas. O objetivo principal deste *use case* é registar as respostas na base de dados do sistema para posterior avaliação. Abrange situações normais, como a conclusão bem-sucedida da prova, bem como exceções, como a expiração da prova ou a opção de finalizar a prova antes de responder a todas as questões.

<i>Use case</i>	14	
Ator principal	Aluno	
Ator secundário	-	
Pré-Condições	O Aluno está autenticado na plataforma e a Prova está agendada.	
Pós-Condições	As respostas do Aluno são registradas no sistema.	
Cenário Normal	Input do Ator	Resposta do Sistema
	1 - O Aluno seleciona a Prova que deseja responder. 2 - O Aluno visualiza as questões da Prova. 3 - O Aluno seleciona uma questão para responder. 4 - O Aluno fornece a sua resposta à questão. 6 - O Aluno repete os passos 3 a 5 para as demais questões da Prova. 7 - O Aluno finaliza a Prova.	5 - O Sistema registra a resposta do Aluno. 8 - O Sistema confirma que a Prova foi respondida com sucesso.
	Exceção 1 - Prova expirada - Passo 2	2.1 - O Sistema informa o Aluno que a Prova está expirada e não pode ser respondida.
Alternativa 1 - Prova parcialmente respondida - Passo 7	7 - O Aluno decide finalizar a Prova mesmo que não tenha respondido a todas as questões.	8 - O Sistema registra as respostas fornecidas e informa que a Prova foi respondida parcialmente.

Tabela 16: Especificação do *use case* "Responder Prova"

Associar Sala à Prova

O caso de uso "Associar Sala à Prova" permite que o Docente associe uma sala à prova previamente criada. O sistema, no entanto, realiza uma alocação automática de salas com base no número de alunos inscritos na prova. Após a alocação automática, o sistema fornece ao Docente uma lista de salas sugeridas para revisão e confirmação.

<i>Use case</i>	15	
Ator principal	Docente	
Ator secundário	-	
Pré-Condições	O Docente criou uma prova e os alunos estão associados à mesma.	
Pós-Condições	A sala é associada à prova.	
Cenário Normal	Input do Ator	Resposta do Sistema
	1 - O Docente indica a data e hora da realização da prova. 5 - O Docente aceita as salas.	2 - O Sistema verifica a data associada à prova anteriormente. 3 - O Sistema verifica a disponibilidade das salas na data especificada. 4 - O Sistema fornece a lista das salas com os alunos associados e as horas da prova. 6 - O Sistema associa as salas,o/s horário/s e data à prova.
Exceção 1 - Sem salas disponíveis - Passo 3		3.1 - O Sistema informa que não há salas disponíveis para todos os alunos. 3.2 O Sistema verifica a disponibilidade de mais salas em outro horário. 3.3 O Sistema indica os dois horários e as salas com os alunos associados. 3.4 Volta ao passo 4.
Exceção 2 - Sem salas disponíveis - Passo 3.2	3.2.2 O Docente volta ao passo 1.	3.2.1 - O Sistema informa que não há salas disponíveis e sugere ao Docente escolher outra data.
Exceção 3 - Rejeição salas - Passo 5	5.1 - O Docente recusa as salas e termina o processo.	
Alternativo 1 - Terminar Processo - Passo 3.2.2	3.2.2.1 O Docente termina o processo de associar uma sala à prova.	

Tabela 17: Especificação do *use case* "Associar Sala à Prova"

Solicitar revisão

O *use case* "Solicitar revisão", cujo ator principal é o *Aluno*, consiste na solicitação da revisão de um teste. Para solicitar o aluno precisa de identificar que perguntas é que acha que merecem revisão e explicar o porque.

Use case	16	
Ator principal	Aluno	
Ator secundário	-	
Pré-Condições	O Aluno está autenticado na plataforma, realizou o teste	
Pós-Condições	Foi pedido uma revisão do teste	
Cenário Normal	Input do Ator	Resposta do Sistema
	1- O Aluno consulta a prova. 3 - O Aluno seleciona a opção de pedir revisão de prova. 5 - O aluno aponta as questões que acha que estão mal corrigidas e escreve uma pequena nota a explicar. 6 - O aluno submete a prova.	2 - O Sistema mostra a prova corrigida. 4 - O sistema aceita rever. 7 - O sistema confirma submissão.

Tabela 18: Especificação do *use case* "Solicitar revisão"

Verificar Erros

O *use case* "Verificar Erros", cujo ator principal é o *Técnico*, consiste na verificação de erros que possam ocorrer ao longo de todo o sistema. Assim que surja um erro no sistema, este é enviado para a secção de monitorização, onde é dividido em diferentes prioridades, dependendo do impacto do mesmo no funcionamento geral do sistema. O técnico poderá analisar o erro e a sua prioridade e de seguida agir em conformidade.

Use case	17	
Ator principal	Técnico	
Ator secundário	-	
Pré-Condições	O Técnico está autenticado na plataforma e uma ação gerou um erro	
Pós-Condições	O Técnico verifica o erro e age em conformidade	
Cenário Normal	Input do Ator	Resposta do Sistema
	1- O Técnico verifica os erros na secção de monitorização. 3 - O Técnico seleciona um erro disponível na lista. 5 - O Técnico toma as medidas necessárias para corrigir o erro (e.g., configuração, alterações no <i>script</i>). 6 - O Técnico documenta o erro e como executou a correção.	2 - O Sistema lista os erros registados no sistema. 4 - O Sistema apresenta os detalhes do erro. 7 - O sistema confirma submissão.

Tabela 19: Especificação do *use case* "Verificar Erros"

Consultar provas

O *use case* "Consultar provas", cujo ator principal é o *Aluno*, consiste na solicitação da consulta de um teste. Para consultar, o aluno precisa de ter operado o teste. Pressupõe que o teste já tenha sido corrigido.

<i>Use case</i>	18	
Ator principal	Aluno	
Ator secundário	-	
Pré-Condições	O Aluno está autenticado na plataforma, realizou o teste, teste corrigido.	
Pós-Condições	O Aluno consegue ver as suas respostas no teste	
Cenário Normal	Input do Ator	Resposta do Sistema
	1- O Aluno Consulta Calendarização. 3 - Aluno seleciona prova.	2 - Sistema mostra calendarização. 4 - O sistema mostra o prova corrigido.
Exceção 1 - Prova não corrigida - Passo 3		2.1 - O Sistema informa o Aluno que não há uma correção da provas.

Tabela 20: Especificação do *use case* "Consultar Prova"

Realizar Revisão

O *use case* "Realizar Revisão", cujo ator principal é o *Docente*, consiste na revisão da consulta de um teste. Para rever a prova, o aluno precisa de ter submetido a prova e tenha feito um pedido de revisão da mesma. O Docente irá ajustar a prova com base nas observações submetidas pelo aluno.

<i>Use case</i>	19	
Ator principal	Docente	
Ator secundário	-	
Pré-Condições	O Aluno terá realizado a prova e submetido uma revisão de prova corrigida.	
Pós-Condições	O Docente corrigiu as coisas erradas	
Cenário Normal	Input do Ator	Resposta do Sistema
	1- Docente clica em listar pedidos de revisão de prova. 3 - Docente escolhe prova. 5 - Docente corrige potenciais erros e submete prova corrigida.	2 - Sistema mostra pedidos. 4 - O sistema carrega o pedido de revisão de prova. 6 - O sistema atualiza nota no aluno.

Tabela 21: Especificação do *use case* "Realizar Revisão"

Classificar Respostas

O *use case* "Classificar Respostas", com o ator principal sendo o Sistema e com o ator secundário sendo o *Sistema*, aborda a correção automática das respostas dos alunos numa prova de avaliação. Neste cenário, o Sistema, inicia o processo de correção automática e pede intervenção do Docente caso seja necessário. O objetivo principal deste *use case* é agilizar o processo de correção,

aplicando critérios de correção previamente definidos e registrando as pontuações das respostas dos alunos. O *use case* inclui cenários normais, como correção bem-sucedida, bem como exceções, como a impossibilidade de correção devido a erros na correção automática ou à prova não estar ainda encerrada.

<i>Use case</i>	20	
Ator principal	Sistema	
Ator secundário	Docente	
Pré-Condições	A prova está encerrada.	
Pós-Condições	As respostas dos alunos são corrigidas e pontuadas.	
Cenário Normal	Input do Ator	Resposta do Sistema
	1 - O Sistema recebe notificação que uma prova foi encerrada. 2 - O Sistema inicia o processo de correção automática pedindo ao Probum os critérios. 4 - O Sistema atribui pontuações às respostas com base na correção automática. 5 - O Sistema verifica a existência de perguntas por corrigir pois não têm critérios de correção automática, .	3 - O Probum manda os critérios de correção previamente definidos às respostas dos alunos. 6 - O Probum regista as pontuações finais das respostas.
Alternativo 1 - Perguntas por corrigir - Passo 5	4.2 - O Sistema recebe o input do Docente para realizar a correção. 4.3 - Volta ao passo 6.	4.1 - O Probum informa o Docente que existem perguntas que necessitam da sua intervenção para serem corrigidas.

Tabela 22: Especificação do *use case* "Classificar Respostas"

Diagrama de Atividades e Autômato de *Use Cases*

Após a elaboração dos Use Cases, o grupo realizou Diagramas de Atividades e Autômato de *Use Cases* para aprofundar a compreensão dos Use Cases e dos seus detalhes de implementação. Os Diagramas de Atividades ajudam a visualizar o fluxo de trabalho em um Use Cases, destacando as ações e decisões que ocorrem. Por outro lado, o Autômato de *Use Cases* mostram a interação entre os Use Cases.

Um dos Diagramas de Atividades está relacionado com o Use Case "Associar Sala à Prova", que o grupo considerou um dos mais complexos. Esse diagrama permitiu uma visualização detalhada da interação entre o Docente e o Sistema e a forma como o Sistema faz a atribuição de salas a uma prova, de modo a permitir vários horários para uma mesma prova caso seja necessário.

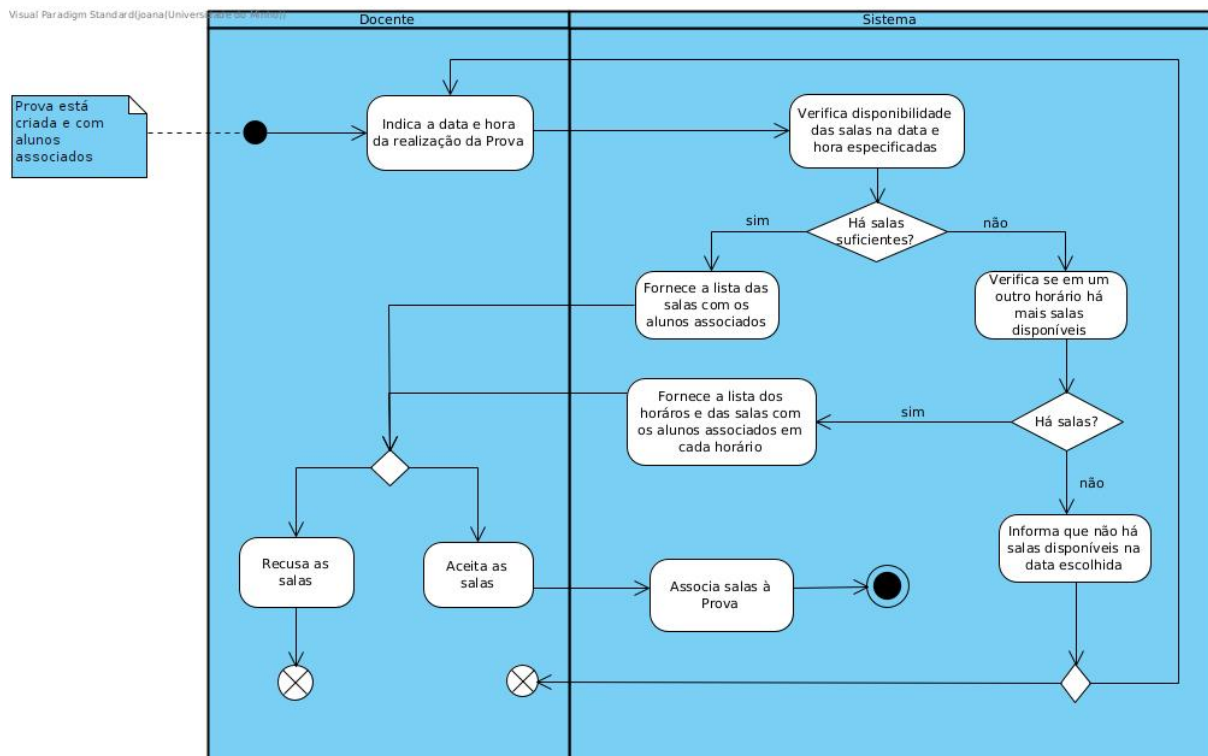


Figura 3: Diagrama de Atividades - Associar Sala à Prova

Outro Diagrama de Atividade que o grupo achou pertinente está relacionado com o Use Case "Editar Prova". Esse diagrama permitiu uma visualização detalhada das possibilidades que o Docente possui quanto efetua a edição de uma prova.

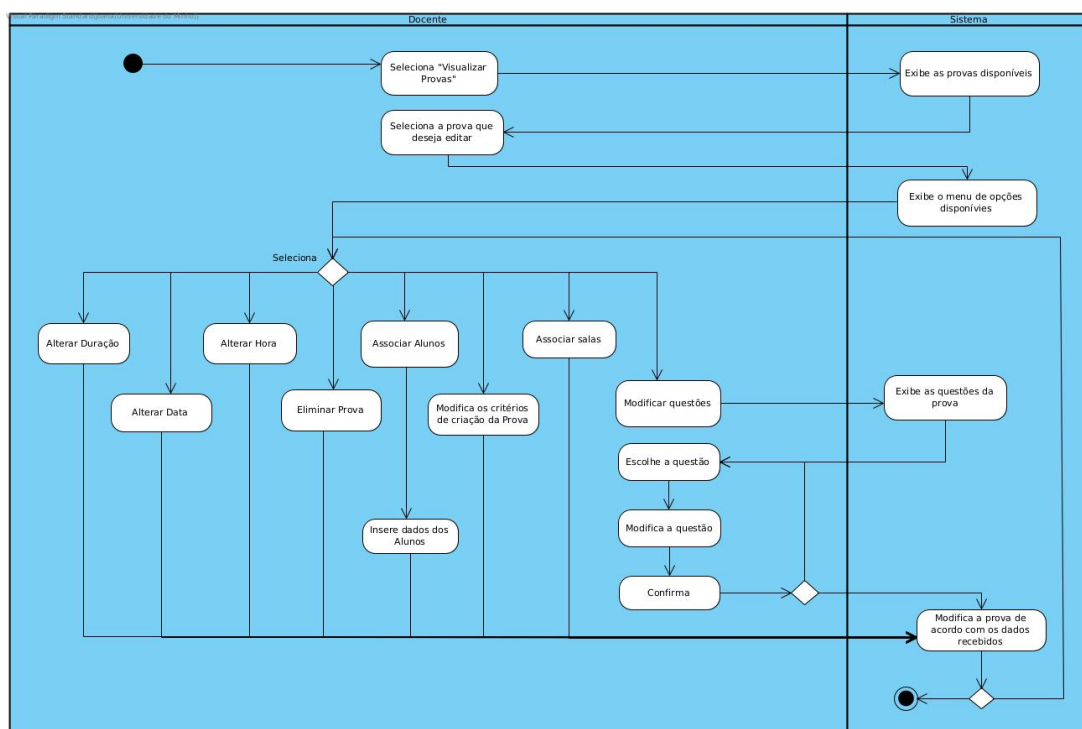


Figura 4: Diagrama de Atividades - Editar Prova

O Autômato de *Use Cases* temos uma demonstração de como os mesmos se relacionam entre si, sendo assim mais explícito as dependências existentes entre eles assim como uma visão da ordem de como os *Use Cases* iram ocorrer.

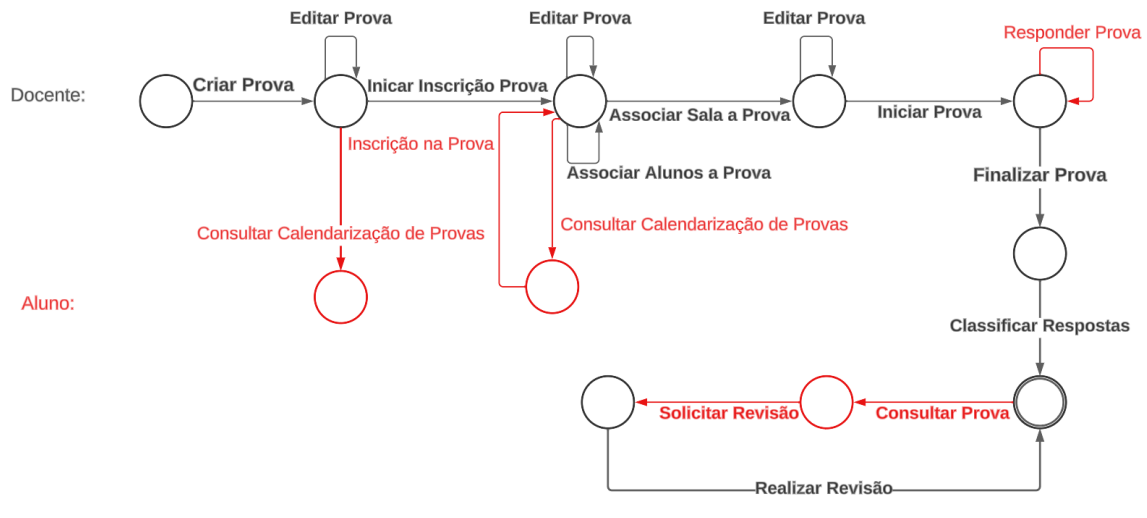


Figura 5: Autômato de use cases

9. Requisitos Funcionais

Modelação de Requisitos

Para o levantamento de requisitos foi utilizada a *requirement shell* do modelo *Volere* como forma de representação, para os descrever concisamente.

Requisito #:	12	Tipo:	Funcional	<i>Use cases</i> #:	1, 7, 20
Descrição	O produto deve registar todas as estradas que foram repavimentadas				
<i>Rationale</i>	Para permitir agendar o reparo de estradas não repavimentadas e identificar potenciais perigos				
Origem	Equipa				
<i>Fit criterion</i>	O registo das estradas repavimentadas deve seguir a especificação da IP e ser feito até 30 minutos após concluída a repavimentação da estrada				
Prioridade	Must				

Tabela 23: Exemplo de especificação de um requisito

Como caracterização da tabela de representação de requisitos, é necessário descrever os campos:

- **Requisito:** número de identificação do requisito.
- **Tipo:** tipo de requisito, considerando o modelo de *Volere*.
- **Use Cases:** número dos Use Cases associados.
- **Descrição:** descrição clara e concisa do requisito.
- **Rationale:** razão para a existência do requisito.
- **Origem:** quem originou o requisito.
- **Fit criterion:** critério para validar cumprimento do requisito.
- **Prioridade:** índice de prioridade para a implementação do requisito:
 - **Must:** requisito obrigatório;
 - **Should:** requisitos que deve ser implementados;
 - **Could:** requisito que não é necessário, mas é desejado;

- Won't: requisito que pode ser considerado posteriormente.
- **Data:** data da especificação do requisito.

Requisitos Funcionais

Requisito #:	RF1	Tipo:	Funcional	Use cases #:	1, 2, 4, 6, 7, 8, 9, 10, 12, 13, 15, 19
Descrição	Autenticação do Docente no sistema				
<i>Rationale</i>	Este requisito permite que o docente apenas consiga utilizar o sistema se estiver autenticado no mesmo				
Origem	Cliente				
<i>Fit criterion</i>	O docente só consegue utilizar o sistema se estiver autenticado.				
Prioridade	Must				

Tabela 24: Requisito funcional para a autenticação do docente no sistema.

Requisito #:	RF2	Tipo:	Funcional	Use cases #:	6, 7, 11, 14, 16, 18
Descrição	Autenticação do Aluno no sistema				
<i>Rationale</i>	Este requisito permite que o aluno apenas consiga utilizar o sistema se estiver autenticado no mesmo				
Origem	Cliente				
<i>Fit criterion</i>	O aluno só consegue utilizar o sistema se estiver autenticado.				
Prioridade	Must				

Tabela 25: Requisito funcional para a autenticação do aluno no sistema.

Requisito #:	RF3	Tipo:	Funcional	Use cases #:	1, 4, 5, 17
Descrição	Autenticação do Técnico no sistema				
<i>Rationale</i>	Este requisito permite que o técnico apenas consiga utilizar o sistema se estiver autenticado no mesmo				
Origem	Cliente				
<i>Fit criterion</i>	O docente só consegue utilizar o sistema se estiver autenticado.				
Prioridade	Must				

Tabela 26: Requisito funcional para a autenticação do técnico no sistema.

Requisito #:	RF4	Tipo:	Funcional	Use cases #:	8
Descrição	O Docente cria uma prova de avaliação				
<i>Rationale</i>	Para permitir associar questões que serão respondidas pelos alunos e, posteriormente, classificadas				
Origem	Cliente				
<i>Fit criterion</i>	A criação de uma prova implica introduzir toda a informação necessária à sua realização e inequívoca identificação				
Prioridade	Must				

Tabela 27: Requisito funcional quanto à criação de uma prova de avaliação

Requisito #: RF5	Tipo: Funcional	Use cases #: 4
Descrição	O Técnico regista Alunos no Sistema	
<i>Rationale</i>	Para permitir que Alunos possam participar na prova, associando-lhes também um método de autenticação. Em alternativa, o Docente também deve poder registar alunos, que não estejam, por alguma razão, ainda inscritos	
Origem	Cliente	
<i>Fit criterion</i>	O processo de registo de alunos deve permitir definir que alunos podem participar na prova, e garantir que cada um deles pode ser autenticado pela plataforma aquando da sua realização	
Prioridade	Must	

Tabela 28: Requisito funcional quanto ao registo de alunos numa prova de avaliação

Requisito #: RF6	Tipo: Funcional	Use cases #: 2, 8
Descrição	O Docente adiciona Questões de escolha múltipla a uma prova de avaliação	
<i>Rationale</i>	Para permitir avaliar o conhecimento dos Alunos numa dada temática, facultando um conjunto de possíveis respostas, onde apenas uma é a correta	
Origem	Cliente	
<i>Fit criterion</i>	As Provas de avaliação podem ser compostas por Questões de escolha múltipla	
Prioridade	Must	

Tabela 29: Requisito funcional quanto à possibilidade de criar Questões de escolha múltipla

Requisito #: RF7	Tipo: Funcional	Use cases #: 2, 8
Descrição	O Docente deve ser capaz de adicionar Questões de Resposta Aberta a uma prova de avaliação	
<i>Rationale</i>	Este requisito possibilita que os docentes criem questões que exigem respostas escritas em texto, permitindo avaliar o conhecimento dos alunos de forma mais aberta	
Origem	Cliente	
<i>Fit criterion</i>	O sistema deve fornecer a opção de adicionar questões de resposta aberta a uma prova de avaliação, que não são limitadas a escolhas pré-determinadas	
Prioridade	Must	

Tabela 30: Requisito funcional para a adição de Questões de Resposta Aberta numa prova

Requisito #: RF8	Tipo: Funcional	Use cases #: 18
Descrição	O Aluno deve ser capaz de consultar a sua prova de avaliação	
<i>Rationale</i>	Este requisito permite que os alunos acessem informações detalhadas sobre o desempenho numa prova de avaliação, incluindo a pontuação total e a avaliação específica de cada questão	
Origem	Cliente	
<i>Fit criterion</i>	O sistema deve disponibilizar uma opção que permita aos alunos visualizar a classificação total e as pontuações específicas por questão de uma prova de avaliação que tenham realizado e que se apresenta classificada	
Prioridade	Must	

Tabela 31: Requisito funcional para a consulta de classificação total e por questão numa prova de avaliação

Requisito #: RF9	Tipo: Funcional	Use cases #: 10
Descrição	O Docente define o momento a partir do qual as provas podem ser consultadas	
<i>Rationale</i>	Para permitir ao Docente disponibilizar as respostas dadas por cada Aluno, assim como as respectivas classificações	
Origem	Cliente	
<i>Fit criterion</i>	As Provas de avaliação apenas podem ser consultadas após o momento definido pelo Docente	
Prioridade	Should	

Tabela 32: Requisito funcional quanto à possibilidade de disponibilizar Provas de avaliação para consulta

Requisito #: RF10	Tipo: Funcional	Use cases #: 2, 8
Descrição	O Docente edita textualmente as questões diretas numa prova de avaliação	
<i>Rationale</i>	Para permitir a edição de perguntas de uma prova de avaliação	
Origem	Docente	
<i>Fit criterion</i>	As Provas de avaliação podem ser compostas por Questões de escolha múltipla	
Prioridade	Must	

Tabela 33: Requisito funcional quanto à possibilidade de criar Questões de escolha múltipla

Requisito #:	RF11	Tipo:	Funcional	Use cases #:	2
Descrição	O Docente edita os critérios das questões de avaliação automática				
<i>Rationale</i>	Para permitir a correção de qualquer gralha na correção da avaliação				
Origem	Docente				
<i>Fit criterion</i>	As Provas de avaliação podem ser compostas por Questões de escolha múltipla				
Prioridade	Must				

Tabela 34: Requisito funcional quanto à possibilidade de criar Questões de escolha múltipla

Requisito #:	RF12	Tipo:	Funcional	Use cases #:	14
Descrição	O Aluno deve ser capaz de responder a uma prova de avaliação				
<i>Rationale</i>	Este requisito permite que os alunos facultem as suas respostas às questões de uma prova a fim de a completar e entregar.				
Origem	Cliente				
<i>Fit criterion</i>	O sistema deve permitir que os alunos acessem, visualizem e respondam a uma prova de avaliação agendada.				
Prioridade	Must				

Tabela 35: Requisito funcional para a funcionalidade de responder a uma prova.

Requisito #:	RF13	Tipo:	Funcional	Use cases #:	1
Descrição	O Docente edita textualmente as questões diretas numa prova de avaliação				
<i>Rationale</i>	Para permitir a edição de perguntas de uma prova de avaliação				
Origem	Docente				
<i>Fit criterion</i>	As Provas de avaliação podem ser compostas por Questões de escolha múltipla				
Prioridade	Must				

Tabela 36: Requisito funcional quanto à possibilidade de criar Questões de escolha múltipla

Requisito #:	RF14	Tipo:	Funcional	Use cases #:	1
Descrição	O Sistema deve ser capaz de correr testes que verificam o bom funcionamento da Plataforma				
<i>Rationale</i>	Para garantir que a plataforma funciona na infraestrutura atual da IES				
Origem	Técnico				
<i>Fit criterion</i>	O Sistema deve disponibilizar uma opção para realização de testes para revelar possíveis problemas que possam ocorrer				
Prioridade	Could				

Tabela 37: Requisito funcional para a testagem da Plataforma

Requisito #:	RF15	Tipo:	Funcional	Use cases #:	1, 17
Descrição	O sistema notifica os Técnicos de erros ocorridos				
<i>Rationale</i>	Para permitir que os Técnicos estejam a par de erros prioritários (impacto crítico e alto) no sistema				
Origem	Cliente				
<i>Fit criterion</i>	O sistema envia notificações de erros com impacto crítico ou alto ao Técnico				
Prioridade	Could				

Tabela 38: Requisito funcional para o envio de notificações de erro ao técnico

Requisito #:	RF16	Tipo:	Funcional	Use cases #:	1, 17
Descrição	O sistema regista erros				
<i>Rationale</i>	Para permitir que se possa manter um registo de todos os erros ocorridos no sistema				
Origem	Cliente				
<i>Fit criterion</i>	O sistema regista todos os erros que ocorrem no sistema				
Prioridade	Must				

Tabela 39: Requisito funcional para o registo de erros no sistema

Requisito #:	RF17	Tipo:	Funcional	Use cases #:	1, 17
Descrição	O sistema mantém registo dos erros durante um mês				
<i>Rationale</i>	Para permitir que se possa existir registo de todos os erros ocorridos no último mês e assim permitir ao Técnico verificar os erros decorridos				
Origem	Cliente				
<i>Fit criterion</i>	O sistema mantém os erros no sistema durante um mês				
Prioridade	Must				

Tabela 40: Requisito funcional para manter os erros no sistema durante um mês

Requisito #:	RF18	Tipo:	Funcional	Use cases #:	2
Descrição	O Docente deve ser capaz de eliminar uma prova				
<i>Rationale</i>	Para permitir que o Docente consiga eliminar uma prova se for necessário				
Origem	Docente				
<i>Fit criterion</i>	O sistema mantém os erros no sistema durante um mês				
Prioridade	Must				

Tabela 41: Requisito funcional para eliminar provas se necessário

Requisito #:	RF19	Tipo:	Funcional	Use cases #:	2, 8
Descrição	O Docente fornece a duração e o dia da realização da prova				
<i>Rationale</i>	Isso é necessário para especificar as informações essenciais relacionadas ao agendamento da prova				
Origem	Cliente				
<i>Fit criterion</i>	O sistema deve validar e armazenar as informações de duração e data fornecidas pelo Docente para associação à prova				
Prioridade	Must				

Tabela 42: Requisito funcional para fornecer informações de agendamento da prova

Requisito #:	RF20	Tipo:	Funcional	Use cases #:	2, 8
Descrição	O Docente define Critérios de Avaliação para as Questões que criar				
<i>Rationale</i>	Isso é necessário para estabelecer os padrões de avaliação das Questões na prova				
Origem	Cliente				
<i>Fit criterion</i>	O sistema deve fornecer uma maneira para que o Docente defina Critérios de Avaliação para cada Questão criada				
Prioridade	Must				

Tabela 43: Requisito funcional para a definição de Critérios de Avaliação

Requisito #:	RF21	Tipo:	Funcional	Use cases #:	2, 8
Descrição	O Docente define o número total de questões da prova.				
<i>Rationale</i>	Isso é necessário para que posteriormente o Sistema ao criar provas aleatoriamente possa seguir os critérios definidos pelo Docente				
Origem	Cliente				
<i>Fit criterion</i>	O sistema deve fornecer uma interface que permita ao Docente especificar o número de questões desejadas				
Prioridade	Must				

Tabela 44: Requisito funcional para personalizar a estrutura da prova

Requisito #:	RF22	Tipo:	Funcional	Use cases #:	2, 8
Descrição	O Docente define o número de questões com cada dificuldade				
<i>Rationale</i>	Isso é necessário para que posteriormente o Sistema ao criar provas aleatoriamente siga certos critérios definidos pelo Docente de modo a as provas fiquem com a mesma dificuldade				
Origem	Cliente				
<i>Fit criterion</i>	O sistema deve fornecer uma interface que permita ao Docente especificar o número de questões desejadas				
Prioridade	Must				

Tabela 45: Requisito funcional para personalizar a estrutura da prova

Requisito #:	RF23	Tipo:	Funcional	Use cases #:	2, 8
Descrição	O Docente define o número de questões por tipo e por tema				
<i>Rationale</i>	Isso é necessário para que posteriormente o Sistema ao criar provas aleatoriamente siga certos critérios definidos pelo Docente de modo a as provas fiquem com a mesma estrutura				
Origem	Cliente				
<i>Fit criterion</i>	O sistema deve fornecer uma interface que permita ao Docente especificar o número de questões desejadas				
Prioridade	Must				

Tabela 46: Requisito funcional para personalizar a estrutura da prova

Requisito #:	RF24	Tipo:	Funcional	Use cases #:	8
Descrição	O Sistema deve criar uma prova, associando a data, duração e Questões registadas pelo Docente				
<i>Rationale</i>	Isso é necessário para criar uma prova completa com base nas informações fornecidas pelo Docente				
Origem	Cliente				
<i>Fit criterion</i>	O sistema deve combinar as informações de agendamento, Questões, Critérios de Avaliação e estrutura da prova para criar a prova associada ao Docente				
Prioridade	Must				

Tabela 47: Requisito funcional para a criação da prova associada

Requisito #:	RF25	Tipo:	Funcional	Use cases #:	12
Descrição	Registo de <i>timestamp</i> de início da prova				
<i>Rationale</i>	Este requisito permite que o sistema registe o <i>timestamp</i> de início da prova, para efeitos de registo e possível auditoria				
Origem	Cliente				
<i>Fit criterion</i>	O sistema deve criar um documento de <i>log</i> , em que regista o <i>timestamp</i>				
Prioridade	Could				

Tabela 48: Requisito funcional para o registo do *timestamp* do início da prova

Requisito #:	RF26	Tipo:	Funcional	Use cases #:	12
Descrição	O sistema atribui questões aleatórias aos alunos				
<i>Rationale</i>	Para permitir que cada aluno tenha uma prova diferente e equivalente em dificuldade às restantes				
Origem	Cliente				
<i>Fit criterion</i>	O sistema deve aleatorizar as questões das provas para cada aluno, a partir de um banco de questões, que possui um índice de dificuldade para cada questão. A distribuição deve manter a escolha do número de perguntas e dificuldade do Docente				
Prioridade	Must				

Tabela 49: Requisito funcional para a atribuição aleatória de questões

Requisito #:	RF27	Tipo:	Funcional	Use cases #:	13
Descrição	O sistema produz relatórios da prova				
<i>Rationale</i>	Para permitir que sejam criados relatórios para fins estatísticos				
Origem	Cliente				
<i>Fit criterion</i>	O sistema deve, no final da prova, produzir um documento de <i>log</i> , que possua informações tal como: <ul style="list-style-type: none"> • Data de início e fim da prova • Número de alunos inscritos e número de alunos que responderam à prova • Duração da prova de cada aluno (dados anónimos) • Questões respondidas por cada aluno (dados anónimos) 				
Prioridade	Must				

Tabela 50: Requisito funcional para a produção de relatórios da prova

Requisito #:	RF28	Tipo:	Funcional	Use cases #:	12
Descrição	O Docente inicia a prova de avaliação				
<i>Rationale</i>	Para permitir que o Docente inicie uma prova de avaliação, assim que todas as condições estejam reunidas: <ul style="list-style-type: none"> • O sistema corre em todos os equipamentos • Os alunos encontram-se na sala e preparados para resolver o teste 				
Origem	Cliente				
<i>Fit criterion</i>	O Docente deve ser capaz de iniciar a prova assim que todos os critérios se reunam. Se não se reunirem, a prova não deverá ser iniciada				
Prioridade	Must				

Tabela 51: Requisito funcional para a iniciação da prova

Requisito #:	RF29	Tipo:	Funcional	Use cases #:	12
Descrição	O Docente seleciona a prova				
<i>Rationale</i>	Para permitir que o docente selecione a prova que deseja iniciar a partir das provas que lhe estão disponíveis (e.g. de outras Unidades Curriculares)				
Origem	Cliente				
<i>Fit criterion</i>	O Docente consegue selecionar a prova que pretende a partir da lista de provas disponíveis				
Prioridade	Must				

Tabela 52: Requisito funcional para a seleção das provas

Requisito #:	RF30	Tipo:	Funcional	Use cases #:	12
Descrição	O sistema notifica o aluno do início da prova				
<i>Rationale</i>	Para permitir que o aluno seja notificado do início da prova e de que possa começar a resolvê-la				
Origem	Cliente				
<i>Fit criterion</i>	O aluno recebe uma notificação de que a prova iniciou assim que o docente a inicie				
Prioridade	Must				

Tabela 53: Requisito funcional para a notificação de início da prova

Requisito #:	RF31	Tipo:	Funcional	Use cases #:	13
Descrição	O docente deve ter a capacidade de encerrar a prova a qualquer momento, caso alguma circunstância alheia à mesma assim o justifique				
<i>Rationale</i>	Permite oferecer flexibilidade ao docente para encerrar a prova quando necessário				
Origem	Cliente				
<i>Fit criterion</i>	O sistema permite que o docente encerre a prova.				
Prioridade	Should				

Tabela 54: Requisito funcional para encerramento da prova pelo docente

Requisito #:	RF32	Tipo:	Funcional	Use cases #:	2
Descrição	O docente deve poder adicionar tempo de compensação à prova				
<i>Rationale</i>	Permite oferecer mais tempo aos alunos para resolver a prova, devido a uma circunstância que assim o justifique				
Origem	Cliente				
<i>Fit criterion</i>	O sistema permite que o docente adicione mais tempo à prova				
Prioridade	Should				

Tabela 55: Requisito funcional para adicionar tempo de compensação à prova

Requisito #:	RF33	Tipo:	Funcional	Use cases #:	13
Descrição	O sistema deve permitir que o docente termine a prova a qualquer momento				
<i>Rationale</i>	Permite que a prova seja terminada pelo docente, assim que se justifique				
Origem	Cliente				
<i>Fit criterion</i>	O sistema termina a prova assim que o docente assim o decida				
Prioridade	Should				

Tabela 56: Requisito funcional para encerramento da prova pelo docente

Requisito #:	RF34	Tipo:	Funcional	Use cases #:	13
Descrição	O sistema deve terminar a prova assim que o tempo definido para a resolução da prova termine.				
<i>Rationale</i>	Permite que a prova seja terminada assim que termine o tempo que o docente colocou para a resolução da mesma				
Origem	Cliente				
<i>Fit criterion</i>	O sistema termina a prova assim que o tempo limite seja atingido				
Prioridade	Should				

Tabela 57: Requisito funcional para encerramento da prova pelo docente

Requisito #:	RF35	Tipo:	Funcional	Use cases #:	13
Descrição	O sistema notifica o aluno do término da prova				
<i>Rationale</i>	Para permitir que o aluno seja notificado do término da prova e de que não consegue responder mais à mesma				
Origem	Cliente				
<i>Fit criterion</i>	O aluno recebe uma notificação de que a prova terminou assim que o docente a termine ou quando o tempo limite da mesma for atingido				
Prioridade	Must				

Tabela 58: Requisito funcional para a notificação de término da prova

Requisito #:	RF36	Tipo:	Funcional	Use cases #:	13, 14
Descrição	O sistema deve impedir alterações após o término da prova				
<i>Rationale</i>	Para permitir que o aluno não responda a mais nenhuma questão assim que a prova for terminada.				
Origem	Cliente				
<i>Fit criterion</i>	O aluno não consegue responder a questões da prova, assim que o tempo limite da prova for atingido ou se o docente terminar a mesma.				
Prioridade	Must				

Tabela 59: Requisito funcional para impedir alterações na prova após o término da mesma

Requisito #:	RF37	Tipo:	Funcional	Use cases #:	20
Descrição	O sistema classifica as respostas automaticamente				
<i>Rationale</i>	O sistema deve ser capaz de classificar automaticamente as respostas dos alunos com base em critérios pré-definidos pelo Docente (e.g., respostas fechadas, palavras-chave usando expressões regulares)				
Origem	Cliente				
<i>Fit criterion</i>	O sistema classifica automaticamente as respostas do Aluno				
Prioridade	Must				

Tabela 60: Requisito funcional para a classificação automática de provas

Requisito #:	RF38	Tipo:	Funcional	Use cases #:	20
Descrição	O Docente revê e aprova a classificação automática das respostas				
<i>Rationale</i>	Para permitir que o Docente aprove as classificações realizadas automaticamente pelo sistema, se assim o desejar				
Origem	Cliente				
<i>Fit criterion</i>	O Docente revê e aprova todas as respostas classificadas automaticamente pelo sistema				
Prioridade	Must				

Tabela 61: Requisito funcional para a revisão e aprovação de classificações automáticas

Requisito #: RF39	Tipo: Funcional	Use cases #: 20
Descrição	O Sistema mantém o registo de todas as classificações das respostas	
<i>Rationale</i>	Para permitir manter um registo de todas as classificações de respostas, permitindo o acesso a versões anteriores das classificações, caso necessário	
Origem	Cliente	
<i>Fit criterion</i>	O Sistema mantém um registo das classificações das respostas, com recurso a versões	
Prioridade	Must	

Tabela 62: Requisito funcional para p registo das classificações das respostas

Requisito #: RF40	Tipo: Funcional	Use cases #: 7
Descrição	O Aluno consegue visualizar a calendarização de provas na plataforma.	
<i>Rationale</i>	Isso é essencial para permitir que o Aluno planeie adequadamente seu calendário académico, conhecendo as datas, horários e locais das provas para as quais está inscrito	
Origem	Cliente	
<i>Fit criterion</i>	O Aluno deve poder aceder a esta informação de forma clara e concisa, facilitando a compreensão das datas das provas	
Prioridade	Must	

Tabela 63: Requisito funcional para a visualização da calendarização de provas por parte dos Alunos

Requisito #: RF41	Tipo: Funcional	Use cases #: 7
Descrição	O Docente deve ser capaz de visualizar a lista de provas agendadas para todas as unidades curriculares dos cursos associados, a fim de evitar sobreposições e permitir uma melhor gestão de calendário	
<i>Rationale</i>	Isso é necessário para que os Docentes possam garantir que não haja conflitos de horários entre diferentes provas e possam fazer ajustes, se necessário	
Origem	Equipa de Desenvolvimento	
<i>Fit criterion</i>	O sistema deve fornecer uma visão clara e organizada das provas agendadas para facilitar a identificação de potenciais sobreposições	
Prioridade	Must	

Tabela 64: Requisito funcional adicional para a visualização de provas agendadas por parte dos Docentes

Requisito #: RF42	Tipo: Funcional	Use cases #: 7
Descrição	O Docente deve ser capaz de associar manualmente os alunos a uma prova	
<i>Rationale</i>	Isso é necessário para permitir que o Docente escolha e inscreva alunos específicos em uma prova, fornecendo a informação dos mesmos como o nome, curso, ano, etc	
Origem	Cliente	
<i>Fit criterion</i>	A associação manual de alunos à prova deve ser concluída com sucesso e refletida no sistema	
Prioridade	Must	

Tabela 65: Requisito funcional para a associação manual de alunos a uma prova

Requisito #:	RF43	Tipo:	Funcional	Use cases #:	2, 9
Descrição	<p>O sistema deve verificar as informações fornecidas pelo Docente durante o processo de associação manual de alunos a uma prova</p> <ul style="list-style-type: none"> Os campos obrigatórios, como nome, curso, ano, endereço de email e número de telefone não devem estar em branco O campo "curso" e "ano" deve conter um valor válido na base de dados da instituição O campo "nome" deve conter apenas caracteres alfabéticos O campo "endereço de email" deve ser válido O campo "número de telemóvel" deve conter 9 dígitos e ser existente 				
Rationale	Essa verificação é necessária para garantir a precisão e validade das informações fornecidas pelo Docente, prevenindo erros e associações incorretas				
Origem	Cliente				
Fit criterion	O sistema deve fornecer feedback imediato ao Docente no caso de informações inválidas, indicando os campos que precisam ser corrigidos				
Prioridade	Must				

Tabela 66: Requisito funcional para a verificação das informações fornecidas pelo Docente

Requisito #:	RF44	Tipo:	Funcional	Use cases #:	4
Descrição	<p>O sistema deve permitir ao utilizador (Aluno ou Docente) atualizar informações pessoais, incluindo nome, curso, ano, endereço de email e número de telefone, no perfil</p>				
Rationale	Isso é necessário para permitir que os utilizadores mantenham suas informações de perfil atualizadas e precisas				
Origem	Cliente				
Fit criterion	As informações atualizadas devem ser refletidas no perfil do utilizador após a conclusão bem-sucedida do processo de atualização				
Prioridade	Must				

Tabela 67: Requisito funcional para a atualização de informações pessoais no perfil do utilizador

Requisito #:	RF45	Tipo:	Funcional	Use cases #:	6
Descrição	O sistema deve permitir ao utilizador (Aluno ou Docente) alterar sua senha				
<i>Rationale</i>	Isso é necessário para fornecer uma maneira segura e eficaz de os utilizadores alterarem suas senhas quando desejarem				
Origem	Cliente				
<i>Fit criterion</i>	A alteração da senha deve ser bem-sucedida e a nova senha deve ser usada para autenticação após a conclusão do processo				
Prioridade	Must				

Tabela 68: Requisito funcional para a alteração de senha do utilizador

Requisito #:	RF46	Tipo:	Funcional	Use cases #:	6
Descrição	Após a atualização bem-sucedida das informações de perfil, o sistema deve notificar o utilizador que as informações foram atualizadas com sucesso				
<i>Rationale</i>	Isso é necessário para informar o utilizador que suas atualizações foram aceitas e concluídas com sucesso				
Origem	Cliente				
<i>Fit criterion</i>	A notificação deve ser clara e imediatamente visível para o utilizador após a conclusão do processo de atualização				
Prioridade	Must				

Tabela 69: Requisito funcional para notificação de sucesso na atualização do perfil

Requisito #:	RF47	Tipo:	Funcional	Use cases #:	6
Descrição	O sistema deve manter um histórico das atualizações do perfil do utilizador, permitindo a recuperação de versões anteriores das informações				
<i>Rationale</i>	Isso é necessário para fornecer um registro de todas as atualizações anteriores, caso o utilizador deseje consultar ou restaurar informações anteriores				
Origem	Cliente				
<i>Fit criterion</i>	O histórico de atualizações deve estar disponível para o utilizador e deve ser mantido de forma segura e acessível				
Prioridade	Should				

Tabela 70: Requisito funcional para o registo do histórico de atualizações do perfil do utilizador

Requisito #:	RF48	Tipo:	Funcional	Use cases #:	2, 15
Descrição	O Docente deve ser capaz de visualizar a lista de salas sugeridas pelo sistema para a prova				
<i>Rationale</i>	Isso é necessário para que o Docente revise as opções de sala antes de confirmar a associação				
Origem	Docente				
<i>Fit criterion</i>	O sistema deve fornecer ao Docente uma lista de salas sugeridas, juntamente com os alunos associados e as horas da prova, para revisão				
Prioridade	Must				

Tabela 71: Requisito funcional para mostrar as possibilidades de salas para a realização de uma prova

Requisito #: RF49		Tipo: Funcional	Use cases #: 16
Descrição	O Aluno deve ser capaz de solicitar uma revisão da prova, identificando as questões que deseja rever e fornecendo uma explicação para cada uma		
<i>Rationale</i>	Isso permite que o Aluno comunique as suas preocupações sobre a correção da prova		
Origem	Cliente		
<i>Fit criterion</i>	O sistema deve aceitar as solicitações de revisão do Aluno, permitindo que ele identifique as questões e forneça explicações		
Prioridade	Must		

Tabela 72: Requisito funcional para a possibilidade de pedir revisão de prova

Requisito #: RF50		Tipo: Funcional	Use cases #: 14
Descrição	O sistema deve permitir ao Aluno salvar suas respostas durante o tempo de resposta da prova		
<i>Rationale</i>	Isso permite que o Aluno continue a trabalhar na sua prova e salve o seu progresso		
Origem	Cliente		
<i>Fit criterion</i>	O sistema deve fornecer uma opção para o Aluno salvar as respostas durante o período de responder a prova		
Prioridade	Should		

Tabela 73: Requisito funcional para guardar respostas

Requisito #:	RF51	Tipo:	Funcional	Use cases #:	10
Descrição	O Docente indica o período de inscrição para uma prova, especificando datas de início e fim				
<i>Rationale</i>	Isso é necessário para permitir que o Docente defina um período em que os alunos podem se inscrever para a prova				
Origem	Equipa de Desenvolvimento				
<i>Fit criterion</i>	O sistema deve fornecer uma interface amigável que permita ao Docente especificar as datas de início e fim para o período de inscrição				
Prioridade	Must				

Tabela 74: Requisito funcional para permitir ao Docente definir o período de inscrição

Requisito #:	RF52	Tipo:	Funcional	Use cases #:	10
Descrição	O sistema notifica os alunos sobre a disponibilidade do período de inscrição para uma prova e fornece informações sobre as datas de início e fim do período de inscrição				
<i>Rationale</i>	Isso é necessário para informar os alunos sobre a oportunidade de se inscreverem na prova e para fornecer informações claras sobre o período de inscrição				
Origem	Equipa de Desenvolvimento				
<i>Fit criterion</i>	O sistema deve enviar notificações por meio de e-mail e/ou notificações internas na plataforma, garantindo que todos os alunos relevantes sejam informados				
Prioridade	Must				

Tabela 75: Requisito funcional para notificar os alunos sobre o período de inscrição da prova

Requisito #:	RF53	Tipo:	Funcional	Use cases #:	10
Descrição	O sistema associa automaticamente os alunos inscritos à prova após o término do período de inscrição indicado pelo Docente				
<i>Rationale</i>	Isso é necessário para simplificar o processo de associação dos alunos à prova e garantir que todos os alunos inscritos sejam incluídos				
Origem	Equipa de Desenvolvimento				
<i>Fit criterion</i>	O sistema deve realizar essa associação de forma automática, respeitando as restrições do período de inscrição indicado				
Prioridade	Must				

Tabela 76: Requisito funcional para a associação automática dos alunos à prova após o término do período de inscrição

Requisito #:	RF54	Tipo:	Funcional	Use cases #:	11
Descrição	O sistema fornece uma lista de provas disponíveis para inscrição, incluindo detalhes como nome, data e disciplina				
<i>Rationale</i>	Isso é necessário para permitir que os alunos escolham a prova desejada com base em informações relevantes				
Origem	Aluno				
<i>Fit criterion</i>	Os detalhes da prova devem ser exibidos de forma clara e organizada, facilitando a seleção por parte do Aluno				
Prioridade	Must				

Tabela 77: Requisito funcional para listar provas disponíveis para inscrição

Requisito #:	RF55	Tipo:	Funcional	Use cases #:	11
Descrição	O Aluno seleciona uma prova a partir da lista de provas disponíveis				
<i>Rationale</i>	Isso é necessário para permitir que os Alunos escolham uma prova específica na qual desejam se inscrever				
Origem	Aluno				
<i>Fit criterion</i>	O sistema deve fornecer uma interface de seleção intuitiva para os Alunos				
Prioridade	Must				

Tabela 78: Requisito funcional para permitir a seleção de uma prova

Requisito #:	RF56	Tipo:	Funcional	Use cases #:	11
Descrição	O sistema deve permitir que o Aluno se inscreva em uma prova				
<i>Rationale</i>	Isso é necessário para registrar a inscrição do Aluno na prova				
Origem	Aluno				
<i>Fit criterion</i>	O processo de inscrição deve ser simples e orientado				
Prioridade	Must				

Tabela 79: Requisito funcional para permitir a inscrição do Aluno em uma prova

Requisito #:	RF57	Tipo:	Funcional	Use cases #:	11
Descrição	O sistema deve confirmar a inscrição do Aluno após a conclusão bem-sucedida do processo				
<i>Rationale</i>	Isso é necessário para informar o Aluno sobre o status da inscrição e fornecer confirmação				
Origem	Aluno				
<i>Fit criterion</i>	O sistema deve exibir uma mensagem de confirmação e um recibo da inscrição bem-sucedida				
Prioridade	Must				

Tabela 80: Requisito funcional para confirmar a inscrição do Aluno

Requisito #:	RF58	Tipo:	Funcional	Use cases #:	3
Descrição	O sistema deve ser capaz de receber e processar automaticamente os dados de novos alunos e docentes fornecidos pelo Sistema da IES				
<i>Rationale</i>	Este requisito permite a integração automática de novos alunos e docentes na plataforma, garantindo um processo eficiente de registo e acesso às funcionalidades do Probum				
Origem	Interação com o Sistema de Gestão de Base de Dados da IES				
<i>Fit criterion</i>	O sistema deve ser capaz de receber e processar automaticamente as informações de novos alunos e docentes fornecidas pelo Sistema de Gestão de Base de Dados da Instituição de Ensino Superior				
Prioridade	Must				

Tabela 81: Requisito funcional para a integração de novos utilizadores na plataforma de forma automática

10. Requisitos Não-Funcionais de Aparência

Requisito #: RNF1	Tipo: <i>Usability</i>	<i>Use cases #:</i> -
Descrição	O sistema apresenta uma aparência em conformidade com o <i>branding</i> da IES	
<i>Rationale</i>	Para permitir que seja facilmente associada à IES em questão	
Origem	Cliente	
<i>Fit criterion</i>	O Probum deve seguir a mesma linha de estilo de outros produtos e sistemas mantidos e geridos pelo IES.	
Prioridade	Should	

Tabela 82: Requisito não funcional de aparência quanto à aparência do sistema.

11. Requisitos Não-Funcionais de Usabilidade

Requisito #:	RNF2	Tipo:	<i>Usability</i>	Use cases #:	-
Descrição	O sistema apresenta uma interface intuitiva e de fácil utilização para os docentes e alunos				
<i>Rationale</i>	Para permitir que os seus utilizadores a consigam utilizar e ter proveito do máximo de funções disponíveis				
Origem	Cliente				
<i>Fit criterion</i>	85% dos utilizadores do Probum devem ser capazes de explorar todas as funcionalidades, fazendo recurso do tutorial disponibilizado pelo sistema.				
Prioridade	Should				

Tabela 83: Requisito não funcional de usabilidade quanto à facilidade de utilização do sistema

Requisito #:	RNF3	Tipo:	<i>Usability</i>	Use cases #:	-
Descrição	O sistema deve ser capaz de ser usado por pessoas com NEE (Necessidades Educativas Específicas)				
<i>Rationale</i>	Para permitir que os utilizadores com NEE consigam utilizar o sistema.				
Origem	Cliente				
<i>Fit criterion</i>	Utilizadores com NEE conseguem utilizar o sistema tal como os utilizadores sem NEE				
Prioridade	Should				

Tabela 84: Requisito não funcional de usabilidade quanto a utilizadores com NEE.

12. Requisitos Não-Funcionais de Performance

Requisito #:	RNF4	Tipo:	<i>Performance</i>	Use cases #:	-
Descrição	O sistema deverá apresenta os erros rapidamente				
<i>Rationale</i>	Para permitir que os técnicos consigam resolver o erro o mais rapidamente possível				
Origem	Cliente				
<i>Fit criterion</i>	Assim que um erro surja, deve ser disponibilizado na lista de erros em menos de um segundo, e deve ser enviado uma notificação ao técnico em menos de 5 segundos				
Prioridade	Should				

Tabela 85: Requisito não funcional de performance relativo à apresentação de erros

Requisito #:	RNF5	Tipo:	<i>Performance</i>	Use cases #:	-
Descrição	O sistema deverá guardar os dados sem comprometer o funcionamento normal do sistema e de forma segura				
<i>Rationale</i>	Para permitir que o sistema seja capaz de armazenar todos os dados do sistema, mesmo em momentos de utilização elevada, criando backups regulares				
Origem	Cliente				
<i>Fit criterion</i>	O sistema não apresenta mais de 20% de duração da realização de processos normais, tendo de registar 1 a 300 provas e a cópia de segurança é realizada em menos de 5 minutos.				
Prioridade	Should				

Tabela 86: Requisito não funcional de performance relativo ao armazenamento de dados.

Requisito #: RNF6		Tipo: <i>Performance</i>	Use cases #: 13, 20
Descrição	O sistema deverá classificar as respostas automáticas de forma rápida e eficaz		
<i>Rationale</i>	Para permitir que o sistema classifique as respostas automaticamente, com uma baixa taxa de erro e da forma mais rápida possível		
Origem	Cliente		
<i>Fit criterion</i>	O sistema classifica automaticamente as respostas em menos de 10 segundos, por prova e apresenta uma taxa de acerto de 100% em perguntas de resposta fechada e de 70% em perguntas de resposta aberta		
Prioridade	Should		

Tabela 87: Requisito não funcional de performance relativo à classificação automática de respostas.

Requisito #: RNF7		Tipo: Não Funcional	Use cases #: 4
Descrição	As atualizações do perfil devem refletir em tempo real, garantindo que as informações sejam atualizadas imediatamente após a conclusão bem-sucedida do processo.		
<i>Rationale</i>	Isso é necessário para proporcionar feedback imediato aos utilizadores e garantir que as informações do perfil estejam sempre atualizadas.		
Origem	Cliente		
<i>Fit criterion</i>	As atualizações feitas no perfil do utilizador devem ser imediatamente visíveis para o próprio utilizador e outros utilizadores autorizados.		
Prioridade	Must		

Tabela 88: Requisito não funcional para atualização em tempo real do perfil.

Requisito #:	RNF8	Tipo:	Não Funcional	Use cases #:	4
Descrição	O sistema deve ser capaz de iniciar e terminar a prova rapidamente				
<i>Rationale</i>	Isso é necessário para proporcionar um rápido feedback do sistema				
Origem	Cliente				
<i>Fit criterion</i>	O sistema inicia e termina a prova para todos os alunos em menos de 5 segundos.				
Prioridade	Must				

Tabela 89: Requisito não funcional para o início e término das provas.

13. Requisitos Não-Funcionais Operacionais

Requisito #:	RNF9	Tipo:	Operacional	Use cases #:	Todos
Descrição	O sistema deverá estar disponível num formato de executável instalado através de um <i>script</i>				
<i>Rationale</i>	Para permitir que o Técnico consiga instalar o programa nos dispositivos necessários de forma rápida e eficaz.				
Origem	Técnico				
<i>Fit criterion</i>	O sistema dispõe de um <i>script</i> de instalação.				
Prioridade	Should				

Tabela 90: Requisito não funcional de Operacional relativo ao suporte dos técnicos

14. Requisitos Não-Funcionais de Manutenção e Suporte

Requisito #:	RNF10	Tipo:	Manutenção e Suporte	Use cases #:	1, 17
Descrição	O sistema deverá oferecer mecanismo de submissão de <i>bugs</i>				
<i>Rationale</i>	Para permitir que o Técnico consiga detetar <i>bugs</i> no sistema, que não sejam necessariamente erros ou exceções reportadas automaticamente pelo sistema.				
Origem	Cliente				
<i>Fit criterion</i>	O sistema dispõe de uma secção dedicada a reportar erros detetados pelo Docente ou pelo Aluno				
Prioridade	Should				

Tabela 91: Requisito não funcional de manutenção e suporte relativo ao suporte dos utilizadores

Requisito #:	RNF11	Tipo:	Não Funcional	Use cases #:	8
Descrição	O sistema deve manter um registo das atividades relacionadas à criação de provas, incluindo quem criou a prova, quando foi criada e quais Questões foram associadas.				
<i>Rationale</i>	Isso é necessário para fins de organização e auditoria.				
Origem	Cliente				
<i>Fit criterion</i>	Must				
Prioridade	Requisito não funcional para atualização de registo.				

Tabela 92:

Requisito #:	RNF12	Tipo:	Não Funcional	Use cases #:	-
Descrição	O <i>source code</i> do programa deve estar documentado				
<i>Rationale</i>	É necessário para que novos técnicos consigam perceber todo o funcionamento do programa, lendo os comentários no código				
Origem	Cliente				
<i>Fit criterion</i>	Todo o <i>source code</i> do Probum deve estar documentado e comentado.				
Prioridade	Must				

Tabela 93: Requisito não funcional para atualização de registo.

15. Requisitos Não-Funcionais de Segurança

Requisito #: RNF13		Tipo: Não Funcional	Use cases #: 10, 11
Descrição	O sistema deve garantir a integridade e a segurança dos dados de inscrição dos alunos durante o período de inscrição na prova.		
<i>Rationale</i>	Isso é necessário para proteger as informações confidenciais dos alunos e evitar qualquer acesso não autorizado durante o processo de inscrição.		
Origem	Equipa de Segurança		
<i>Fit criterion</i>	O sistema deve utilizar mecanismos de criptografia e autenticação adequados para proteger os dados de inscrição dos alunos. Além disso, deve ter sistemas de controle de acesso para evitar violações de segurança.		
Prioridade	Must		

Tabela 94: Requisito não funcional para garantir a segurança e integridade dos dados de inscrição dos alunos durante o período de inscrição na prova.

Requisito #: RNF14		Tipo: Não Funcional	Use cases #: 8
Descrição	O sistema deve garantir a segurança das informações associadas à prova, protegendo-as contra acesso não autorizado.		
<i>Rationale</i>	Isso é essencial para manter a confidencialidade das informações das provas.		
Origem	Cliente		
<i>Fit criterion</i>	Must		
Prioridade	Requisito não funcional para proteção de dados da prova.		

Tabela 95:

Requisito #: RNF15		Tipo: Não Funcional	Use cases #: 3, 4, 6
Descrição	O sistema deve garantir a segurança dos dados dos alunos ao receber e armazenar informações pessoais.		
<i>Rationale</i>	Isso é necessário para proteger a privacidade e confidencialidade das informações pessoais dos alunos, cumprindo regulamentações de privacidade de dados.		
Origem	Cliente		
<i>Fit criterion</i>	O sistema deve implementar medidas de segurança, como criptografia de dados e controle de acesso, para garantir a proteção dos dados dos alunos durante o processo de associação à prova.		
Prioridade	Must		

Tabela 96: Requisito não funcional para garantir a segurança dos dados dos alunos durante a associação à prova.

16. Requisitos Não-Funcionais Culturais e Políticos

Requisito #:	RNF16	Tipo:	<i>Cultural e Político</i>	Use cases #:	-
Descrição	O produto está preparado para suportar vários idiomas (e.g., português, espanhol, inglês)				
<i>Rationale</i>	Para permitir que os seus utilizadores o possam usar utilizando a língua que mais dominam				
Origem	Cliente				
<i>Fit criterion</i>	Utilizadores que dominam diferentes idiomas têm a mesma experiência de utilização				
Prioridade	Should				

Tabela 97: Requisito não funcional cultural e político quanto ao produto suportar vários idiomas

Requisito #:	RNF17	Tipo:	<i>Cultural e Político</i>	Use cases #:	2, 7, 8
Descrição	O sistema apresenta todos os feriados nacionais na calendarização				
<i>Rationale</i>	Para permitir que os Docentes saibam os dias em que não podem agendar provas				
Origem	Cliente				
<i>Fit criterion</i>	O sistema apresenta no seu calendário todos os feriados nacionais.				
Prioridade	Should				

Tabela 98: Requisito não funcional cultural e político quanto à calendarização de feriados.

17. Requisitos Não-Funcionais Legais

Requisito #: RNF18		Tipo: <i>Legal</i>	<i>Use cases #</i> : Todos
Descrição	O sistema cumpre com a legislação do RGPD em vigor		
<i>Rationale</i>	Para permitir que o sistema esteja de acordo com as normas previstas de RGPD		
Origem	Cliente		
<i>Fit criterion</i>	O sistema cumpre com as normas de privacidade indicadas pelo RGPD		
Prioridade	Should		

Tabela 99: Requisito não funcional legal quanto ao cumprimento do RGPD.

Requisito #: RNF19		Tipo: <i>Legal</i>	<i>Use cases #</i> : Todos
Descrição	O sistema deve armazenar as provas dos alunos durante 2 anos		
<i>Rationale</i>	Para cumprir com a legislação em vigor e permitir tratar eventuais reclamações		
Origem	Cliente		
<i>Fit criterion</i>	O sistema armazena as provas dos alunos durante 2 anos após a disponibilização da mesma		
Prioridade	Should		

Tabela 100: Requisito não funcional legal quanto ao armazenamento das provas do alunos.

Requisito #: RNF20		Tipo: Não Funcional	Use cases #: X
Descrição	O sistema deve confirmar a submissão da solicitação de revisão após o Aluno concluir a submissão.		
Rationale	Isso fornece feedback ao Aluno de que a solicitação foi registrada com sucesso.		
Origem	Cliente		
Fit criterion	Must		
Prioridade	Requisito não funcional para confirmação de submissão de prova		

Tabela 101:

21. Levantamento de Requisitos

O processo de levantamento de requisitos para o projeto foi cuidadosamente planeado para atender às necessidades dos utilizadores finais e garantir a eficácia do sistema. Para este fim, foram adotadas diversas técnicas e abordagens de levantamento de requisitos, a fim de garantir que os requisitos fossem bem definidos e documentados, cobrindo qualquer particularidade que quisessemos expressar. As técnicas utilizadas incluíram:

- **Entrevistas com Utilizadores:** Foram realizadas entrevistas individuais e em grupo com Docentes, Alunos, Técnicos e outros intervenientes para discutir as suas necessidades, expectativas e requisitos específicos para o sistema Probum.
- **Pesquisa de Mercado:** Foi conduzida uma pesquisa de mercado para identificar outras soluções semelhantes que podem servir como referência. Isto permitiu a comparação e a recolha de ideias relevantes para o desenvolvimento do Probum.
- **Análise de *Use Cases*:** Foram elaborados os casos de uso para representar as interações entre os utilizadores e o sistema. Isto ajudou-nos a definir os principais cenários de uso e vários requisitos funcionais e não funcionais, quase diretamente.
- **Observação em Instituições de Ensino Superior:** A equipa de projeto visitou Instituições de Ensino Superior para observar de perto como as avaliações académicas são conduzidas e identificar os desafios enfrentados pelos utilizadores no processo.
- **Feedback Direto dos Utilizadores:** Foram realizadas sessões de feedback com Docentes, Alunos e Técnicos para recolher opiniões e sugestões sobre o sistema Probum, a fim de refinar os requisitos.

À medida que o processo de levantamento de requisitos avançava, todas as informações coletadas eram registadas no nosso documento de requisitos com a estrutura indicada anteriormente, projetado especificamente para esse fim.

Compreendemos que o processo de levantamento de requisitos foi uma etapa crítica no desenvolvimento do Probum, garantindo que as necessidades dos utilizadores fossem compreendidas e documentadas de forma abrangente.

Conclusão

Em retrospectiva, consideramos que o nosso desenho atual para este sistema satisfaz as condições propostas e possui o que precisa para uma possível boa continuação da evolução sustentada do projeto, resultado de um processo de aperfeiçoamento das definições que nos eram pedidas com base em muitos momentos de discussão de parâmetros e dinâmicas da aplicação.

Temos em mente todos os pontos fortes do nosso trabalho, mas contemplamos também os pontos fracos, pois sabemos que contemos uma quantidade de Use Cases e Requisitos relativamente elevada, que apesar de nos ajudarem a definir e cobrir especificidades do funcionamento da nossa aplicação, podem tornar-se áridos e excessivos a certo ponto.

Foi importante ter sempre em consideração que estas etapas de definição e modelação de arquiteturas são imprescindíveis num ciclo de vida de desenvolvimento de software, e que várias decisões tomadas agora se irão refletir com importância em fases mais avançadas de implementação.

Olhando ao processo deste trabalho prático até ao momento, consideramos que se tem revelado enriquecedor pois dá-nos ferramentas e hábitos importantes para o desenvolvimento de software robusto, com um planeamento cuidado, sem saltar etapas. Neste ponto, a linguagem UML foi, sem dúvida, bastante útil na construção dos mais diversos modelos e diagramas que suportam a codificação e que se têm revelado fundamentais no desenvolvimento de um projeto sólido e sustentado, que garanta os requisitos propostos.

Em suma, relativamente a esta fase do projeto, o grupo considera que fez um bom planeamento e construiu uma boa base capaz de suportar, evoluir e prosperar no futuro deste trabalho.