

UNIVERSIDADE ABERTA,
UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO



**Desenvolvimento de uma aplicação móvel para acesso às informações do
portal da Universidade Aberta**

Francisco Jaime Chimbinde

**Trabalho de Projeto para Mestrado em Engenharia Informática e Tecnologia
Web**

2022

UNIVERSIDADE ABERTA,
UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO



**Desenvolvimento de uma aplicação móvel para acesso às informações do
portal da Universidade Aberta**

Francisco Jaime Chimbinde

**Trabalho de Projeto para Mestrado em Engenharia Informática e Tecnologia
Web**

Trabalho de Projeto orientado pelo Professor Doutor Vítor Rocio

Resumo

O avanço das tecnologias de informação e comunicação (TICs) tem sido um processo evolutivo e com várias melhorias. As tecnologias passaram a suportar grande parte dos negócios em todo mundo, também tem influenciado nas diferentes formas de trabalho no seio da população.

O perfil e exigências sociais do trabalhador atual, tem tendências a uma aposta crescente ao uso de dispositivos móveis devido as facilidades proporcionadas por estes, como exemplo da aplicação destes dispositivos, tem o processo de ensino a distância onde estudantes podem aceder aos conteúdos académicos, na saúde a realização de consultas medicas online, e os pacientes podem interagir com os profissionais de saúde por meio de um celular.

Os dispositivos moveis tornaram-se ferramentas poderosas de acesso e partilha de informação. Devido a sua facilidade de mobilidade, passaram a efetuar o papel de salas de aulas virtuais, postos de trabalho, ambientes de laser, casinos e etc.

A Universidade Aberta (UAb) vem acompanhando essa evolução, tanto que atualmente usa o moodle e outros sistemas Web como suporte as atividades internas, e para conectar-se aos estudantes.

Apesar das aplicações Webs serem executadas em dispositivos moveis, as aplicações moveis nativas podem ser usadas como extensão de acesso aos conteúdos das aplicações webs.

No caso da UAb, ainda não dispõe de um aplicativo móvel focado as necessidades dos alunos, na gestão das informações a nível da instituição, por exemplo, os dados do sistema moodle, do sistema de gestão académica (SGA), financeiros e outras, são acedidos de aplicações webs diferentes, o que obriga ao aluno visitar vários endereços eletrónicos e compilar informações de várias plataformas.

Pretende-se desenvolver uma aplicação móvel capaz de incorporar informações integradas, existentes no portal e outras plataformas num único aplicativo.

A parte *front-end* da aplicação será desenvolvida com *React Native*, devido a sua simplicidade, facilidade e possibilidade de desenvolvimento paralelo para as versões iOS e Android, através de um único projeto, enquanto que o *back-end* será usado o *Nodejs*, associado as bibliotecas *express*.

Com a implementação desta solução espera-se criar facilidades de conexão aluno versus UAb, de modo que acedam as informações dos sistemas Webs por meio de um aplicativo móvel.

Abstract

The advancement of information and communication technologies (TICs) has been an evolutionary process with several improvements. Technologies have come to support a large part of business worldwide and have also influenced the different forms of work within the population.

The profile and social requirements of the current worker, has a tendency to a growing bet on the use of mobile devices due to the facilities provided by them, as an example of the application of these devices, there is the distance learning process where students can access academic content, consultations medics online, and can interact with health professionals and teachers through a cell phone.

Mobile devices have become powerful tools for accessing and sharing information, however, due to their ease of mobility, they have taken on the role of virtual classrooms, workstations, laser environments, casinos, etc.

The Open University (UAb) has been following this evolution, so much so that it currently uses moodle and other Web systems to support internal activities and to connect with students.

Although web applications run on mobile devices, native applications can be used as an extension of access to the contents of web applications.

In the case of the UAb, it does not yet have a mobile application focused on the needs of the students, in the student-institution connection, for example, the data from the moodle system, the academic management system (SGA), financial and others, are accessed from applications different webs, which forces the student to visit and compile information from various platforms.

It is intended to develop a mobile application capable of incorporating integrated information existing on the portal and other platforms in a single application.

The front-end part of the application will be developed with React Native, due to its simplicity, ease and possibility of parallel development of the IOS and Android versions through the same project, while the back-end will be used by Nodejs.

With the implementation of this solution, it is expected to create facilities for connecting students from the UAb to the institution, so that they can access the information on the Webs systems.

Índice

ÍNDICE DE TABELAS	VIII
ÍNDICE DE FIGURAS.....	IX
LISTA DE ABREVIATURAS E SIGLAS.....	X

1. INTRODUÇÃO..... 1

1.1. ENQUADRAMENTO TEÓRICO	2
1.2. REVISÃO DE LITERATURA	3
1.3. PROBLEMA DE INVESTIGAÇÃO	7
1.4.1. OBJETIVO GERAL.....	7
1.4.2. OBJETIVOS ESPECÍFICOS.....	7
1.5. RESULTADOS ESPERADOS	8
1.6. TOMADAS DE DECISÕES.....	8
1.7. OPÇÕES METODOLÓGICAS.....	11
1.8. ESTRUTURA DO DOCUMENTO	11

2. DESENVOLVIMENTO DO PROJETO..... 13

2.1. REQUISITOS FUNCIONAIS E DIAGRAMA DE CASOS DE UTILIZAÇÃO.....	14
2.2. DIAGRAMAS DE CASOS DE UTILIZAÇÃO DO SISTEMA.....	15
2.3. MODELO RELACIONAL DE BASE DE DADOS.....	17
2.4. ARQUITETURA APLICACIONAL	20
2.5. <i>BACK-END</i> (FUNCIONAMENTO E ESTRUTURA DA API).....	22
2.5.1. LISTA DE ALGUNS PROCEDIMENTOS ARMAZENADOS	23
2.5.2. ORGANIZAÇÃO DOS PROCEDIMENTOS ARMAZENADOS	25
2.5.3. ORGANIZAÇÃO DA API	26
2.5.3.1. RESTFUL.....	29
2.5.3.2. INSTALAÇÃO DA API	30
2.6. DESCRIÇÃO DO <i>FRONT-END</i>	31
2.7. FUNCIONAMENTO DO SISTEMA.....	32
2.7.1. ACESSO AO SISTEMA.....	33
2.7.2. TELA INICIAL	34
2.7.3. CURSOS.....	35
2.7.4. UNIDADES CURRICULARES.....	36
2.7.5. CALENDÁRIO ACADÊMICO	37
2.7.6. HISTÓRICO DAS NOTAS:	39
2.7.7. CARTÃO DE APRENDIZAGEM	39

3. AVALIAÇÃO DO PROTÓTIPO (INQUÉRITO)..... 41

3.1. QUALIDADE DO DESIGN	43
3.2. CONFORMIDADE (SATISFAÇÃO DOS USUÁRIOS)	44
3.3. TESTE E FUNCIONAMENTO	44
3.4. ANÁLISE DE RESULTADOS.....	44
3.4.1. TESTE DE COMPATIBILIDADE	44
3.4.2. TESTES DE USABILIDADE	46
3.4.3. TESTES DE DESEMPENHO.....	47
3.4.4. TESTES DE RESPONSABILIDADE.....	47
3.4.5. OUTROS RESULTADOS.....	48

4. CONCLUSÕES E TRABALHO SUBSEQUENTE.....	56
5. BIBLIOGRAFIA	59
5.1. REFERÊNCIAS BIBLIOGRÁFICAS	60
5.2. DOCUMENTAÇÃO ADICIONAL E PÁGINAS WEB	62

Índice de Tabelas

Tabela 1- Adaptação da tabela “Funcionalidades das aplicações mobile de algumas instituições” de Carvalho (2017).	10
Tabela 2-Lista dos requisitos funcionais em função do ator	15
Tabela 3-Lista dos requisitos funcionais em função das entidades da base de dados	17
Tabela 4-Contendo a lista dos procedimentos armazenados	25
Tabela 5-Lista dos endpoints da API.....	27
Tabela 6-Resultado Json da chamada de um endpoint	29
Tabela 7-Resumo dos dados dos utilizadores.....	42
Tabela 8-: Resumo das respostas (marca/modelo)	45
Tabela 9- Resumo das respostas (tamanho de telas)	45
Tabela 10- Resumo das respostas (tamanho de telas)	46
Tabela 11-Resumo de respostas para o teste de usabilidade	47
Tabela 12- Resumo da resposta (Responsividade)	48
Tabela 13- resumo da resposta (problemas no funcionamento)	49
Tabela 14-Tabela de eficiência e eficácia.....	50
Tabela 15-Tabela ligada à velocidade de processamento.....	50
Tabela 16-Resultados da combinação de cores e estética da interface.....	51
Tabela 17-Resultados das respostas sobre os cartões de aprendizagem.....	52
Tabela 18-Resultados das respostas sobre a visualização dos calendários académicos.....	52
Tabela 19-Resultados das respostas sobre segurança da aplicação	53
Tabela 20-Resultados das respostas sobre segurança do Aplicativo	54

Índice de Figuras

Figura 1- Diagrama de casos de uso	16
Figura 2-Modelo relacional do sistema	19
Figura 3-Arquitetura Aplicacional.....	21
Figura 4-Estrutura do back-end	23
Figura 5-Parâmetros de entrada para o procedimento armazenado proc_curso	25
Figura 6- Estrutura do diretório app	26
Figura 7- Estrutura do diretório mãe da API	26
Figura 8- Chamadas dos procedimentos armazenados.....	28
Figura 9 - Chamadas dos procedimentos armazenados.....	28
Figura 10-Teste online de acesso a API	30
Figura 11-Ambiente de instalação e gestão de aplicações Node.js	31
Figura 12-Diretória raiz do aplicativo.	32
Figura 13-Diretório lib, estrutura dos aplicativos.....	32
Figura 14-Interface para o login do aluno	33
Figura 15-Menu inicial	34
Figura 16-Página inicial.....	34
Figura 17-Lista de cursos participados	35
Figura 18-Lista de disciplinas de um curso	36
Figura 19-Seleção de cursos	36
Figura 20-Calendarário acadêmico.....	37
Figura 21- Eeventos de um dia selecionado	38
Figura 22-Formulário de cadastro de eventos	38
Figura 23-Lista das disciplinas em função das notas	39
Figura 24-Lista das atividades agendadas e terminadas.....	40
Figura 25-Lista das atividades agendadas	40
Figura 26-Ilustração gráfica dos testes feitos em Android em função do iOS	45
Figura 27-Ilustração gráfica do resultado dos tamanhos das telas efetuadas testes	45
Figura 28-Resumo gráfico dos resultados ligados a facilidade de navegação	46
Figura 29-Resumo gráfico dos resultados ligados ao tempo de resposta do aplicativo	47
Figura 30-Resumo gráfico dos resultados ligados a responsividade.....	48
Figura 31-Resumo gráfico dos resultados ligados aos problemas no momento dois testes	49
Figura 32-Resumo gráfico dos resultados ligados aos erros encontrados em teste.....	50
Figura 33-Resumo gráfico dos resultados ao processamento.....	50
Figura 34-Resumo gráfico dos resultados sobre a estética da interface	51
Figura 35-Resumo gráfico dos resultados sobre a visualização do cartão de aprendizagem ..	52
Figura 36-Resumo gráfico dos resultados sobre a visualização do calendário acadêmico	52
Figura 37-Resumo gráfico dos resultados sobre a segurança da aplicação	53
Figura 38-Resumo gráfico dos resultados sobre a disponibilidade da Aplicativo	54
Figura 39-Resumo gráfico dos resultados sobre as recomendações e observações dos usuários	54

Lista de Abreviaturas e Siglas

TICs - tecnologias de informação e comunicação

UAb - Universidade Aberta

SGA - sistema de gestão académica

SITCON -

SISCAD - Sistema Académico *Online*

EaD - Ensino a Distância

IES – Instituições de Ensino Superior

ISEC - Instituto Superior de Engenharia e Coimbra

API - Application Programming Interface

PLA - Personal *learning assistant*

UML - Linguagem Unificada de Modelagem

SGBD - Sistema de gerenciamento de base de dados

1. Introdução

1.1. Enquadramento teórico

O uso das Tecnologias da Informação e Comunicação (TICs) aumentou as possibilidades de ampliação de acesso à formação, tornando-se parte integrante de projetos educacionais. Essas mudanças estão fornecendo novas possibilidades de práticas docentes, visando melhorar a qualidade do processo educativo (Elias 2011).

Nos dias de hoje, o uso de dispositivos moveis tem sido um processo com aderência notável no mundo, devido as facilidades oferecidas por estas tecnologias, várias atividades tornaram-se exequíveis com recursos celulares, tabletes e outros dispositivos moveis.

Segundo (Carvalho, 2017), outras universidades nacionais dispõem de aplicações móveis, como é o caso da Universidade de Trás-os-Montes e Alto Douro que permite que o aluno consulte os seus dados académicos, a Universidade da Beira Interior, a Universidade Lusófona, a Universidade Católica (“myCatólica”), a Universidade Nova de Lisboa, etc.

Carvalho ilustra na sua dissertação, uma lista contendo a relação nominal, de algumas universidades e as devidas aplicações disponibilizadas, as plataformas usadas e as funcionalidades implementadas.

Este trabalho centra-se no desenvolvimento de um aplicativo móvel híbrido, integrado aos sistemas da UAb, com a finalidade de expandir o modelo atual de acesso as informações académicas na UAb.

A criação deste protótipo foi motivada de uma proposta da UAb e alguns projetos de dissertação da mesma universidade, que propuseram a implementação de dispositivos moveis como um meio facilitador as interações estudantes versos Universidade UAb, portanto, estes abordaram o mesmo tema, mas noutros contextos, alguns na parte teórica, outros no enquadramento desta tecnologia ao modelo pedagógico da UAb.

Atualmente são usadas as tecnologias *web* para aceder as informações e aderir alguns serviços, sejam eles financeiros assim como pedagógicos. A universidade oferece aos seus estudantes diversas plataformas como moodle, para interação pedagógica, portal académico

para visualização de notas, faturas, assim como o estado financeiro do estudante. Todas elas com suporte ao serviço de email institucional.

Estas ferramentas foram desenvolvidas e implantadas sobre a arquitetura *web*, mas podem ser acedidas com recursos aos dispositivos moveis.

A implementação do aplicativo, para além de expandir a disponibilidade de serviços já existentes nessas plataformas *web*, vai dinamizar o acesso a informação por parte do estudante, que com recurso a esta solução poderá aceder aos dados ou informação existente nas demais plataformas da Universidade e em tempo real.

A ideia da implementação desta solução deve-se a facilidade de acesso aos dados que ela pode trazer aos estudantes, uma vez que terá a mobilidade como um recurso ao seu favor, desde que possua *internet*. Poderá receber alertas, mensagens, acompanhar os eventos institucional, assim como agendar nas suas atividades. Parte destas vantagens não são oferecidas com a mesma facilidade quando são usados só recursos *web*, visto que um programa com arquitetura *web* é executado a nível do servidor, o cliente só poderá rodar ao aceder o link e em primeiro plano.

1.2. Revisão de literatura

A aprendizagem móvel é um novo paradigma educacional, o qual é mais flexível do que a aprendizagem que utiliza computadores tradicionais. Ela permite que qualquer pessoa tenha acesso as informações e materiais de aprendizagem de qualquer lugar e a qualquer hora (Demirbilek 2010; Earnshaw et al. 2011).

Kukulska-Hume & Traxler (2005), definem *mobile learning (m-learning)* como um conceito novo que envolve a mobilidade e o estudante, o qual pode realizar atividades educacionais sem limitação do espaço físico de um ambiente de ensino. Pode-se aprender em ambientes fora das salas de aula, basta apenas, motivação de aprender e um dispositivo móvel, leve e que caiba no bolso, como um celular ou tablete,

Carvalho (2017) propõe o desenvolvimento de uma aplicação móvel para o modelo pedagógico virtual da Universidade Aberta, que segundo o autor, irá preencher a lacuna

apresentada pela falta de uma aplicação móvel direcionada para o modelo pedagógico de ensino a distância da UAb.

Segundo o autor, a implementação do mesmo permitirá que os alunos trabalhem com a tecnologia *web* e *mobile* simultaneamente, deixando a cargo as condições ou vontade do utilizador.

O autor optou pela alteração da plataforma *moodle* mobile já existente, que oferece o recurso *addons* como mecanismo de alargamento e adaptação do aplicativo para eventuais necessidades. No trabalho, o autor propõe a disponibilização do acesso as funcionalidades principais dos espaços virtuais das funções a distância da UAb tais como consulta de agenda, de classificações ou mesmo *upload* de trabalhos.

No entanto, o mesmo focou-se mais na componente pedagógica, o que difere da proposta apresentada neste trabalho, uma vez que as mesmas necessidades de expansão do acesso, são verificadas também a nível das consultas de informações administrativas, serviços de *suporte*, dentre outros.

Na proposta apresentada pelo autor, há necessidade de alinhar as outras partes com acesso via a tecnologia *mobile*, de modo que os estudantes consigam ver a parte financeira, ver os eventos institucionais, receber notificações pelo celular assim como aceder as funcionalidades da plataforma SITCON fornecida pela UAb, e acompanhar a evolução das tecnologias.

Oliveira (2017), no seu trabalho desenvolve um aplicativo para controle académico com dispositivos móveis. No trabalho, com base neste, o docente pode lançar as notas, frequências por aulas, envia mensagens para as turmas e consulta as áreas de conhecimento. A proposta contém os módulos do professor, aluno, avisos dos docentes para as turmas e são usados serviços *web* para sincronização da informação entre os módulos, permitindo que o usuário trabalhe mesmo sem *internet* e de forma automática é feita a sincronização da informação, basta que a ligação de internet seja estabelecida.

Para o autor, a implementação deste aplicativo vem como extensão do sistema SISCAD (Sistema Acadêmico *Online*) que é disponibilizado na *web* e não para substituir o sistema *Web*, todavia, este aplicativo vem no intuito de tornar a manipulação das informações do meio académico mais eficiente e prática.

Engelmann et al. (2016), propõem a incorporação de aplicativos móveis voltados para a disponibilização de serviços das instituições de Ensino superior para a comunidade acadêmica, no artigo apresentam resultados sobre o desenvolvimento de um aplicativo móvel integrado denominado MobileS, que segundo os autores, tem como finalidade conectar instituição ao aluno, disponibilizando a comunidade acadêmica serviços institucionais, de modo a facilitar o processo de acompanhamento de informações e de atendimento.

Engelmann et al, usam o sistema de gestão acadêmica da própria instituição como fonte de dados que são integrados por serviços *web*.

Os autores mencionam a diferença das regras de negócio entre instituições como um dos motivos que impulsionou a implementação do projeto, uma vez que existem projetos com o mesmo foco, mas com regras de funcionamento distintas, o aplicativo foi denominado MobileS, foi desenvolvido com arquitetura híbrida. Para suportar algumas funcionalidades como alteração de dados de cadastro, consulta ao histórico escolar, inscrição em cursos de extensão e acesso a informações financeiras.

Os autores propõem esta solução como mecanismo de resolução do problema causado pelas diferenças das regras de negócio existentes, no entanto, segundo o autor, o aplicativo é compatível com qualquer instituição de ensino desde que os sistemas das IES sejam adaptados para tal.

Mencionam a existência de várias aplicações móveis EaD (Ensino a Distância) para instituições de ensino, mas vários destes tem finalidade suportar a parte didática entre os professores e os alunos, porém, o trabalho não está focado nesta vertente, mas sim na conexão aluno e a instituição em áreas como parte financeira e etc.

Na proposta os autores tiveram como resultados uma aplicação híbrida capaz de dar as IES benefícios como, oferecer aos alunos serviços institucionais através de um meio de comunicação muito usado no-dia-a-dia da comunidade acadêmica, este que passa a ser uma forma alternativa ao atendimento convencional.

Costeira, S. A. B. (2019), propõe a implementação de uma aplicação móvel para que estudantes do ISEC pudessem efetuar todo tipo de consultas e operações relacionadas as atividades escolares, desta forma, ele oferece o desenho e desenvolvimento de uma

aplicação móvel com funcionalidades similares ao NetP que é um portal acadêmico do Instituto Superior de Engenharia e Coimbra (ISEC).

Segundo o autor, o portal tem um peso enorme na vida acadêmica dos alunos, pois são consultados regularmente para operações como visualização de notas, inscrições em exames e consulta de referências multibanco para pagamento de propinas.

Nos dias de hoje, a tendência de separação do *back-end* ao *front-end* torna-se comum em vários projetos, devido a dinâmica de trabalho proporcionada aos projetos.

Segundo Gonsalves(2014) et al, relatam uma experiência relacionada ao uso de *WebSockets* para atender as demandas dos sistemas de informação que necessitam de respostas em tempo real, estes integram os *WebSockets* a uma arquitetura *RESTful* de modo a comunicar os módulos utilizados para prover serviços aos usuários

Silva et al(2020) , apresentam uma proposta para incorporar a API (Application Programming Interface) desenvolvida em as funcionalidades relacionadas ao calculo da pegada de carbono de uma propriedade rural. Segundo os autores, qualquer sistema que deseje incorporar novas funcionalidades somente precisa configura-las, adicionando a API desejada em suas biblioteca.

A API desenvolvida, em conjunto com o banco de dados presente no servidor da Embrapa Pecuária Sul, possibilita que informações possam ser armazenadas para posterior leitura e compartilhamento em diversos dispositivos, possibilitando o intercambio de dados entre sistemas Silva et al(2020).

Tendo em conta as fontes mencionadas anteriormente, pode-se notar uma procura de implementação de serviços aos dispositivos moveis, como algo de extrema importância, parte dos trabalhos foi mencionado a implementação de uma solução móvel como mecanismo para expansão dos serviços, esta que tem sido uma tendência notável pelo mundo das tecnologias. O uso da arquitetura *RESTful*, para além de dar outra dinâmica organizacional ao projeto, ajuda o compartilhamento do mesmo *back-end* para dispositivos móveis, assim como web e aplicações *Desktop*.

1.3. Problema de Investigação

Para Miguel (2017), a UAb já dispõe de sistemas *web* para interação com os alunos, entretanto, os mesmos aplicativos são integrados a nível de dados, todavia, nenhum destes está disponível na versão para dispositivos móveis sem contar que as informações se encontram em sistemas diferentes, algumas na aplicação moodle outras no portal do estudante. Além de mais, o sistema de atendimento ficará mais abrangente se for acessado também por dispositivos móveis nativos.

O uso de uma Interface PLA (*Personal Learning Assistant*) poderá ajudar no processo de atendimento básico em tempo real, entretanto, para o direcionamento da pesquisa criou-se a seguinte questão:

- A implementação de um aplicativo móvel capaz de disponibilizar informações básicas do portal UAb e outros sistemas da Universidade poderá facilitar a interação aluno versus UAb?

Para o desenvolvimento desta solução existem mais fatores condicionantes, dentre eles, organização dos dados a nível da própria UAb (APIs), o entendimento do protocolo de partilha de dados usado para o consumo das aplicações existentes a partir de aplicações externas. Como pretende-se alimentar a aplicação por meio de APIs, então é fundamental que estas fontes de informação já existam, caso contrário será fundamental desenvolver esta parte.

1.4.1. Objetivo geral

O principal objetivo deste trabalho é desenvolver uma proposta de aplicativo móvel para interação aluno versus Universidade Uab, com vista a facilitar a partilha das informações básicas dos sistemas web existentes na Instituição.

1.4.2. Objetivos específicos

- Desenvolver um *back-end* capaz de responder ao problema
- Permitir a consulta de eventos no calendário académico
- Consultar o cartão de aprendizado das disciplinas
- Visualizar histórico das notas das disciplinas concluídas

- Receber alertas e mensagens da instituição
- Receber informe e notícias publicadas na Universidade

Para alcançar os objetivos mencionados existe um conjunto de requisitos ao nível de tecnologias que devem ser cumpridos, recorrendo ao uso de APIs com autenticação *RESTful* que será usada para o desenvolvimento do *back-end* da aplicação, sendo responsável pela interação com o *front-end*, assim como as APIs e serviços disponibilizados pelos sistemas da UAb, porém, os sistemas atuais da Universidade devem fornecer esta abertura de informações por meio de *endpoints* autenticados.

1.5. Resultados esperados

Com a implementação deste projeto, espera-se a criação de uma aplicação móvel, integrada aos sistemas da Universidade Aberta. A mesma deverá ser usada pelos estudantes que poderão efetuar as tarefas feitas nos sistemas *Webs* fornecidos pela instituição em seus dispositivos *smartphones* ou *tabletes*.

Os utilizadores poderão consultar facilmente os eventos no calendário académico, consultar as informações do cartão de aprendizagem das disciplinas, o histórico das notas finais das disciplinas que participou, em função dos cursos do estudante. Poderá também, ver o seu estado financeiro, receber alertas e mensagens da instituição, acompanhar as notícias da página da Universidade dentre outros e etc.

1.6. Tomadas de decisões

Como foi mencionado na parte introdutória do trabalho, a tendência da migração ao uso dos dispositivos móveis é uma realidade notável nos dias de hoje. Segundo Jonhson et al. (2011), em breve, 80% das pessoas navegarão na Internet a partir de um dispositivo móvel.

Atualmente, esta tecnologia torna possível a concretização de vários processos do nosso dia a dia, sem contar que esta temática é tida em conta na maioria das discussões que abordam sobre as tecnologias educativas e nas áreas de inovação.

Atualmente, as pessoas esperam poder trabalhar, aprender e estudar em qualquer lado e a qualquer hora, porque as tecnologias móveis, que usam diariamente, são cada vez mais “*cloud-based*” (Johnson et al., 2011).

Para além destas tecnologias ajudarem a economizar recursos em empresas, elas ajudam também a economizar tempo e isto tende a ajudar no aumento da produtividade. Tendo em conta ao mundo atual, onde o recurso tempo tem sido uma variável fundamental no sucesso das organizações, o acesso a informação associado a mobilidade torna os processos mais eficientes.

A Computação Móvel é um paradigma da computação no qual usuários portando dispositivos móveis têm acesso a serviços via redes sem fio independente de sua localização (Forman e Zahorjan, 1994).

Segundo Teixeira Júnior (2005), executivos estão trocando *notebooks* por telefones inteligentes, que buscam e exibem on-line, todas as informações importantes para a tomada de decisões longe do escritório

Uma vez que varias entidades vem aderindo a esta tecnologia, há uma tendência competitiva para as instituições, em relação a sua implementação para o uso associado aos sistemas de gestão Universitária várias instituições já vêm aderindo estes recursos.

Segundo Carvalho (2017), que ilustra uma tabela com as Universidades portuguesas e internacionais em função das plataformas usadas para suporte assim como a arquitetura das mesmas. Nesta tabela pode-se notar que muitas instituições de ensino tendem a implementar essa tecnologia, todavia, é um ponto importante para Universidade Aberta expandir também.

Instituição	Educação a distância	Nome da aplicação	Plataforma		Funcionalidades
			Android	iOS	
Universidade Católica	Não	myCatolina	X	X	-
Universidade da Beira Interior	Não	mUBI	X		-
Universidade de Aveiro	Não	UAMobile	X	x	-
Universidade de Trás-os-Montes e Alto Douro (UTAD)	Não	alunosUTA D	X	X	-
Universidade Lusófona	Não	Lusófona Mobile	X	x	-
Nova SBE	Não	Nova SBE	X	X	-
American Public University System	Sim	APUS Mobile	X	x	-
DeVry University	Sim	DeVry University	X	x	-
Harvard University	Não	Liberty Today	X	x	-
Massachusetts Institute of Technology	Não	MIT Mobile	X	x	-
Princeton University	Não	Princeton Mobile	X	x	-
Stanford University	Não	Stanford Mobile	X	x	-
The Open University	Sim	OU Anywhere	X	X	-
UNIR	Sim	CampusUN IR	X	X	-
University of Oxford	Sim	Mobile Oxford	X	x	-
University of Phoenix	Sim	Yale	X	x	-
Yale University	Não	University of Phoenix Mobile	X	x	-

Tabela 1- Adaptação da tabela “Funcionalidades das aplicações mobile de algumas instituições” de Carvalho (2017).

1.7. Opções metodológicas

Para o desenvolvimento da solução foram seguidos alguns passos recomendados pelas metodologias ágeis de desenvolvimento, numa primeira fase, o estado atual da Universidade aberta. Fez-se o Levantamento do estado atual dos sistemas e recursos da Universidade, com vista a perceber melhor o negócio, com base em observação aos sistemas disponíveis. A Modelação e Desenvolvimento da proposta de solução, consistiu no desenho e desenvolvimento de um aplicativo capaz de resolver o problema.

Uma vez definida a solução foi feita a implementação da mesma, seguida de testes de execução, onde foram efetuados inquéritos a alguns usuários escolhidos, de modo a colher sensibilidades sobre a qualidade da proposta e as melhorias por adicionar.

1.8. Estrutura do documento

Este trabalho é composto por 4 capítulos, que descrevem e documentam o processo de desenvolvimento de uma aplicação mobile para a expansão da aplicação web. No capítulo 1, é descrito um enquadramento teórico junto a uma revisão bibliográfica de modo a apresentar o uso e a relevância do trabalho, tendo em conta a escolha de outras organizações e instituições e tendências do mercado atual. Neste capítulo são descritos ainda o panorama atual da implementação desta solução, os resultados esperados deste trabalho e uma breve explicação das decisões tomadas para a escolha deste projeto.

No Segundo capítulo descreve-se o desenvolvimento da aplicação, aqui são ilustrados e descritos o funcionamento da aplicação, desde os diagramas, casos de uso e modelos de dados. São detalhados aspetos relacionados as etapas de instalação da aplicação assim como uma breve apresentação das ferramentas usadas para o desenvolvimento, API *RESTful*, Nodejs, a base de dados e o *Framework* React Native, expo e no final o resultado esperado.

O terceiro capítulo é centrado na descrição dos resultados dos testes da aplicação desenvolvida, aqui são descritas as reações e dos utilizadores após terem experimentado o protótipo disponibilizado.

O quarto capítulo, faz-se a descrição conclusiva do trabalho, onde são descritos aspetos como desafios identificados e as soluções de alguns inconvenientes. Nesta parte são mencionados também, alguns desafios futuros para a continuação deste trabalho.

2. Desenvolvimento do projeto

2.1. Requisitos funcionais e Diagrama de Casos de Utilização

Para melhor compreensão do projeto, partiu-se dos requisitos funcionais dos sistemas já existentes, em função dos atores também existentes. Tratando-se uma aplicação para o uso dos estudantes foram modelados apenas 3 tipos de atores, o usuário, o estudante e o *UAb_Web_services*.

- **Usuário**- é o usuário criado para a efetivação dos dados gerais de todos os tipos futuros de dados. Este usuário tem a capacidades de efetuar o login, gerir as palavras passes. Um estudante herda as funcionalidades deste ator, assim como, em casos futuros esta entidade será importantíssima para a adição de mais categorias de usuários, por exemplo, caso se deseja expandir o aplicativo para os docentes, alumnis, ou mesmo candidatos a UAb.
- **Estudante**- é o ator que a partir do aplicativo poderá visualizar os seus dados pessoais, visualizar o cartão de aprendizagem de cada disciplina, consultar o histórico das notas, entre outras funcionalidades.
- **UAb_Web_services**- este ator é um sistema/módulo externo contendo serviços(scripts) webs, que será responsável na sincronização dos dados do aplicativo aos sistemas webs existentes. Este poderá atualizar as notas, o histórico das notas, enviar emails e todos os outros dados que são recebidos dos sistemas web e de todos estudantes, de modo que os dados do aplicativo mantenham-se atualizados.

A figura1 descreve os requisitos funcionais escolhidos para esta aplicação. diagrama de casos de uso do sistema, desta forma, ilustra-se as funcionalidades, os atores e as suas relações.

Código	Requisito funcional	Ator
RF01	Efetuar o login	Usuário
RF02	Visualizar cartão de aprendizagem	Estudante
RF03	Listar eventos e calendário acadêmico	Estudante
RF04	Consultar histórico das notas	Estudante
RF05	Agendar eventos pessoais	Estudante
RF06	Consultar notificações	Estudante
RF07	ver inscrições	Estudante
RF08	Consultar mensagens	Estudante
RF09	Visualizar notas de avaliações	Estudante
RF10	Mudar password	Usuário
RF11	Atualizar pagamentos inscr	Uab_web-services
RF12	Atualizar registro usuário	Uab_web-services
RF13	Atualizar calendário	Uab_web-services
RF14	Editar eventos pessoais	Estudante
RF15	Atualizar histórico notas	Uab_web-services

Tabela 2-Lista dos requisitos funcionais em função do ator

2.2. Diagramas de casos de utilização do sistema

A figura 1 representa o diagrama de casos de utilização usado para o desenvolvimento do projeto, nele estão representadas as funcionalidades do sistema.

Com base no diagrama de casos de uso, pode-se ver que o ator usuário executa a funcionalidade “efetuar Login” e a funcionalidade “mudar a palavra chave”. Existem outras que não são ilustradas no diagrama.

As funcionalidades executáveis pelo ator usuário, são herdadas pelos usuários estudantes e poderão ser herdadas por outros tipos de usuários em caso de necessidades futuras, porém, a ligação entre estes atores é “herança”.

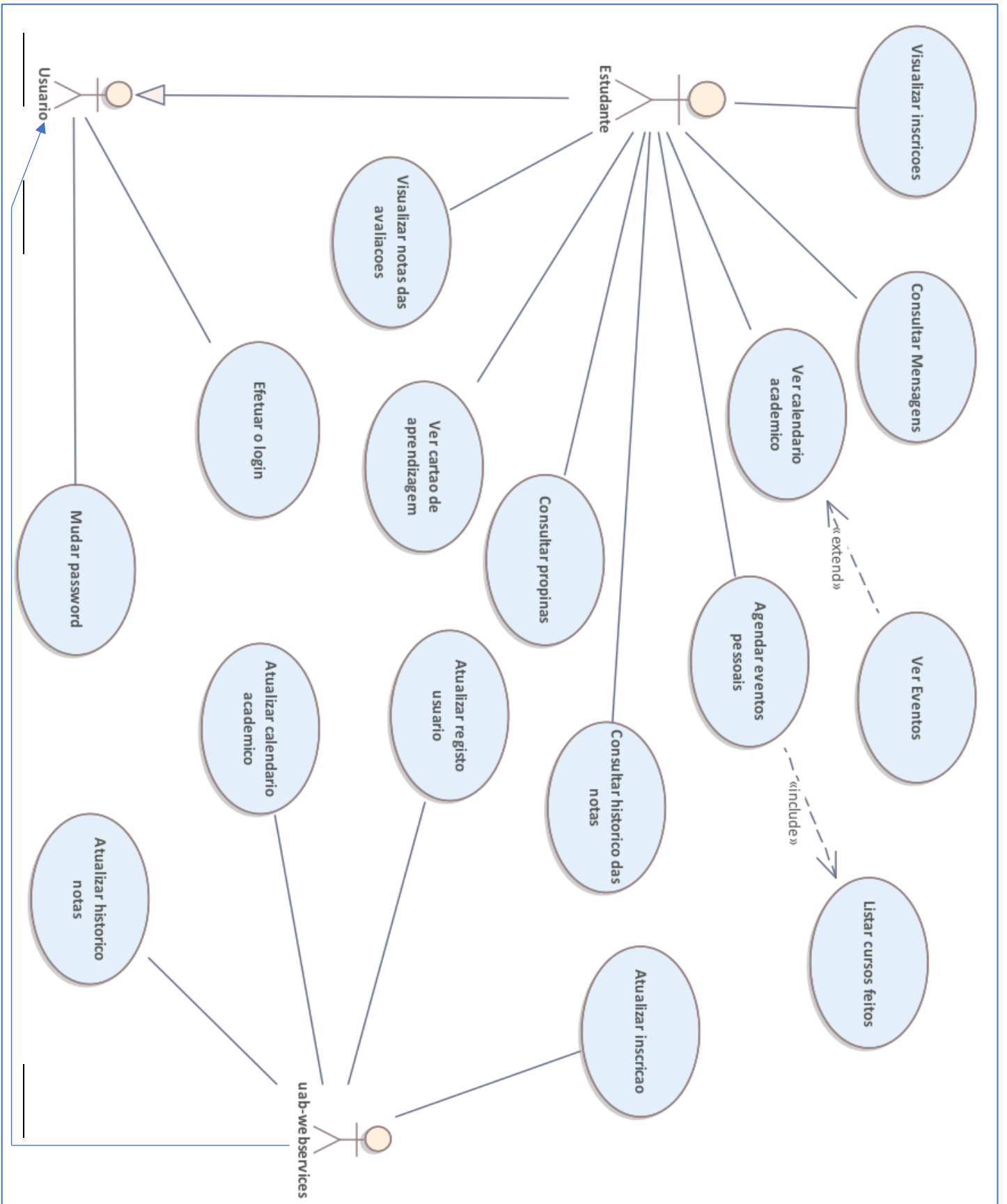


Figura 1- Diagrama de casos de uso

2.3. Modelo relacional de base de dados

Antes de mencionar sobre o modelo de dados, é importante relacionar os requisitos funcionais as entidades que são ilustradas no modelo de dados, na tabela 3, acrescentou-se a relação dos casos de uso em função das entidades da base de dados.

Requisito funcional	Entidades na base de dados
Efetuar o login	Tabela Usuário
Visualizar cartão de aprendizagem	Tabelas inscritas, est_actividades
Listar eventos e calendário académico	Evento, tipoevento
Consultar histórico das notas	Históricodisciplinas
Agendar eventos pessoais	Evento
Consultar notificações	Usuario_grupo, mensagem
ver inscrições	Inscritas
Consultar mensagens	Mensagem
Visualizar notas de avaliações	Est_actividades
Mudar password	Usuário
Atualizar pagamentos inscr	Pagamentoinscr
Atualizar registo usuário	Usuario
Atualizar calendário	Evento
Atualizar histórico notas	Historicodisciplinas

Tabela 3-Lista dos requisitos funcionais em função das entidades da base de dados

O modelo relacional do trabalho é apresentado na figura 2, nele são ilustradas as combinações entre as tabelas que tornam possíveis a realização das funcionalidades do aplicativo.

Segundo o diagrama, que foi desenhado em função do negócio em estudo, registaram-se algumas regras de negócio:

- A conta de um usuário do sistema pode estar ativa ou inativa
- Um estudante é um usuário
- Um usuário pode estar ou não associado a um grupo de usuários
- Um estudante pode estar associado a um ou mais cursos
- Um estudante pertencente a um curso pode se inscrever as disciplinas pertencentes ao plano curricular do curso.
- Para as disciplinas de um curso, são planificadas uma ou mais atividades

- Um estudante participa de atividades criadas nas disciplinas pelo qual está inscrito
- Um professor leciona uma ou mais disciplinas de um curso
- Um usuário regista eventos pessoais
- Um usuário visualiza eventos publicados pela instituição
- Um curso possui um plano curricular
- Um usuário de um grupo publica mensagens a um ou mais usuários
- Um usuário administrador envia alerta a um usuário ou mais
- Um plano curricular de um curso agrega várias disciplinas

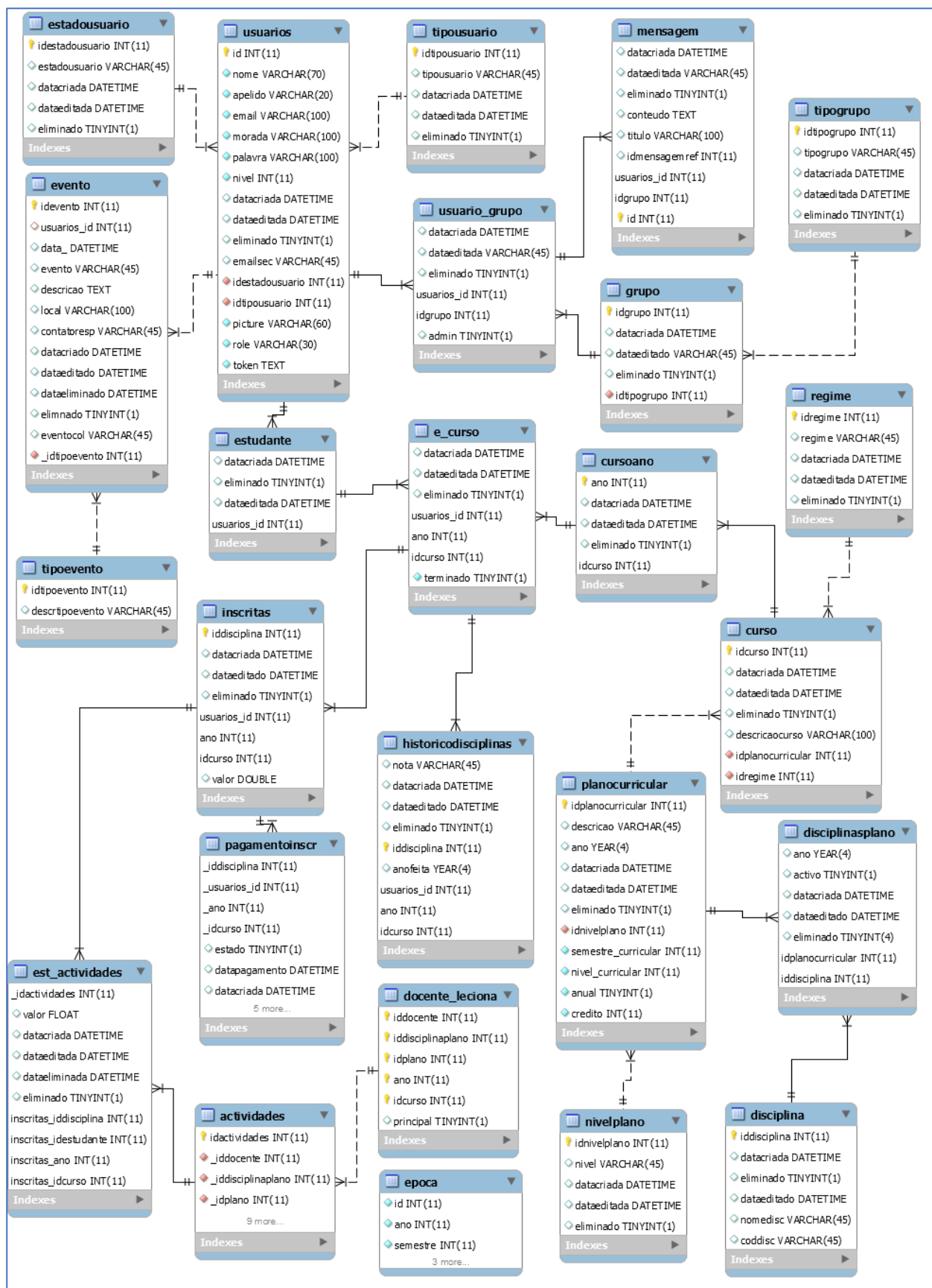


Figura 2-Modelo relacional do sistema

2.4. Arquitetura Aplicacional

A figura 3, ilustra a arquitetura aplicacional do problema em questão. A nível da aplicação as funcionalidades podem iniciar nos usuários, que podem requisitar serviços por meio de um dispositivo móvel (tablete ou celular). Nestes dispositivos são usados recursos da aplicação *React Native* para receber dados ou captar o que o usuário precisa.

Os dados requisitados são enviados ao servidor *web* por meio do recurso internet que é responsável pela ligação do “*fron-tend*” ao “*back-end*”. No servidor está hospedada uma API responsável pelo atendimento as requisições dos clientes. A API interage com o sistema gerenciador de base de dados (MySQL) que efetua de imediato a consulta da requisição solicitada por meio de procedimentos armazenados existentes na base de dados.

A API retorna os dados requisitados ao servidor *web*, que por meio de serviços *web* fornece ao dispositivo cliente o resultado através de um endereço de *endpoint* solicitado e a resposta é divulgada na tela cliente do utilizador.

A outra forma de circulação de dados a nível da aplicação é por meio de serviços *web* usados para a sincronização da informação da aplicação local em função aplicação mãe da UAb.

Tendo em conta a figura, os serviços *web* que são executados automaticamente por meio de agendamento de scripts, podem aceder aos dados existentes nos diversos servidores *web* da UAb e atualizar a base de dados usada pela API. Este poderá ser executado em intervalos de 30 minutos ou mesmo a cada 10 minutos.

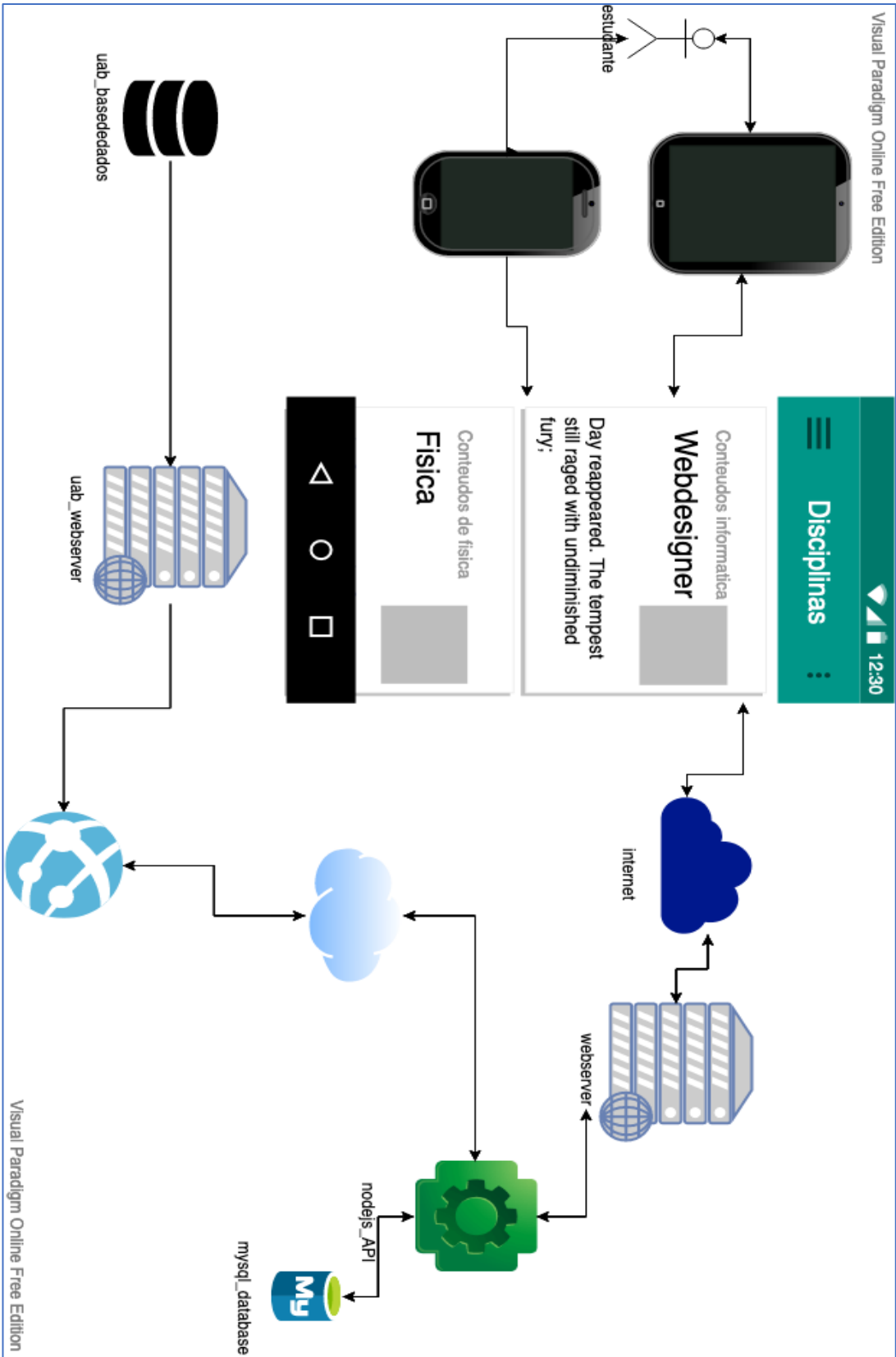


Figura 3-Arquitectura Aplicacional

2.5. *Back-end* (Funcionamento e estrutura da API)

A tecnologia *Nodejs*, que usa a linguagem *Javascript*, foi usada para o desenvolvimento do *back-end* da aplicação. No desenvolvimento da API, foi aplicado o conceito MVC (modelo vista controlo) para facilitar a integração de futuras equipas de trabalho e desenvolvimento do mesmo projeto.

O *Nodejs* pode ser definido como um ambiente de execução *Javascript server-Side* (*Nodejs*, 2020).

Quando se diz que *Nodejs* é *Server-Side*, significa que com o *Node.js* é possível criar aplicações *Javascript* para rodar como uma aplicação standalone em uma máquina, não dependendo de um browser para a execução, como estamos acostumados. (*Nodejs*, 2020).

A tecnologia *Javascript* é muito conhecida pela sua execução na parte cliente desde o início da sua existência, contudo, atualmente vem ganhando espaço por ter sido implementada também no lado do servidor. A implementação do *Javascript* no lado do servidor, é suportada pelo *Nodejs*, que permite que os desenvolvedores possam desenvolver aplicações que serão executadas do lado servidor.

O principal motivo de sua adoção é a sua alta capacidade de escala. Além disso, sua arquitetura, flexibilidade e baixo custo, o tornam uma boa escolha para implementação de Micro-serviços e componentes da arquitetura *Serverless* (modelo de desenvolvimento nativo em nuvem para criação e execução de aplicações sem o gerenciamento de servidores). Inclusive, os principais fornecedores de produtos e serviços *Cloud* já têm suporte para desenvolvimento de soluções escaláveis utilizando o *Node.js*.

A principal característica que diferencia o *Node.js* de outras tecnologias, como *PHP*, *Java*, *C#*, é o fato de sua execução ser *single-thread*. Ou seja, apenas uma *thread* é responsável por executar o código *Javascript* da aplicação, enquanto que nas outras linguagens a execução é *multi-thread* (*Nodejs*, 2020).

Outros aspetos que foram relevantes para a escolha desta ferramenta para o desenvolvimento *back-end*, foi o facto de usar a biblioteca *npm* (*Node Package Manager*),

que é o maior repositório de software do mundo. Isto ajuda bastante no desenvolvimento, porque a tecnologia permite reutilizar uma variedade de códigos do repositório, Node.js pode facilitar mais pelo facto de permitir que desenvolvedores trabalhem com *front-end* assim como o *back-end* sobre a mesma linguagem de programação.

O *back-end* é dividido em duas partes, a base de dados e a API. Para melhor organização e processamento, foi desenhado o esquema representado na figura 4.

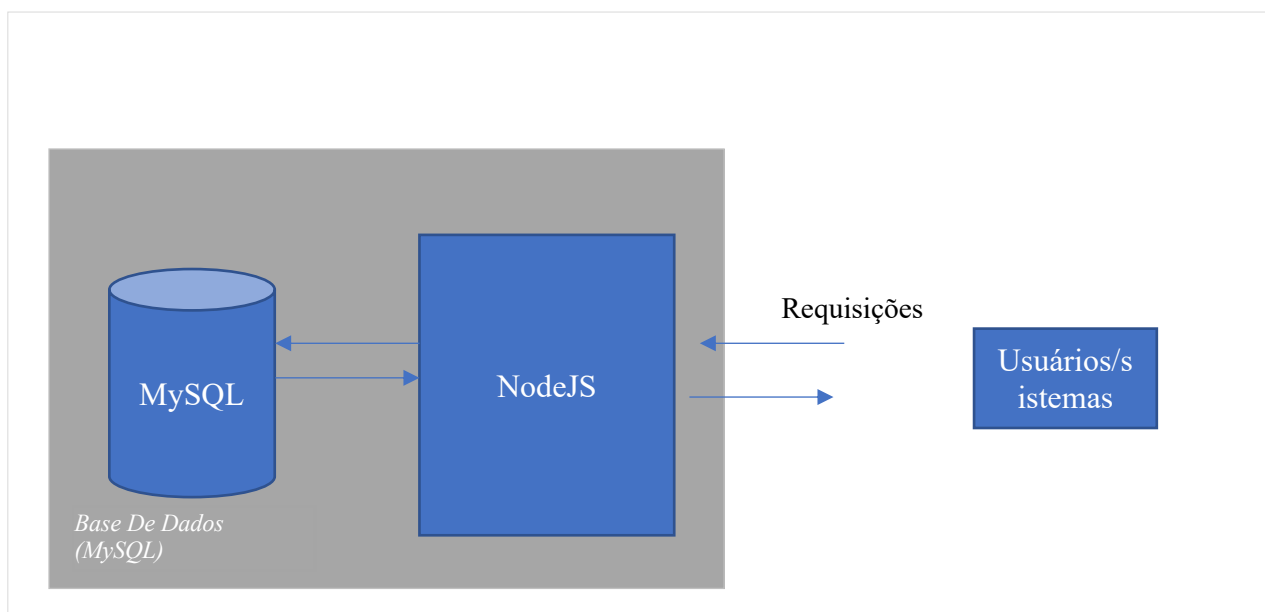


Figura 4-Estrutura do back-end

Na base de dados, cujo modelo foi ilustrado anteriormente, são usados procedimentos armazenados, vistas e gatilhos. O uso dos procedimentos armazenados permite que parte da lógica do *back-end* seja resolvido a nível da base de dados, ajudando desta forma a diminuir a carga de processamento no servidor web.

2.5.1. Lista de alguns procedimentos armazenados

Tendo em conta a arquitetura e a organização da aplicação, foram usados procedimentos armazenados para no ao invés de consultas SQL embutidas nas classes modelo das APIs. A principal motivação para tal foi a reutilização das consultas, e divisão da carga das requisições

entre a aplicação Node.js e o sistema de gerenciamento de base de dados MySQL. A tabela 4 ilustra procedimentos armazenados e uma breve descrição.

ordem	Nome do procedimento	Descrição
1	Proc_activ_disc_est	Responsável pela visualização das atividades do estudante por estudante, esta funcionalidade
2	proc_calendario_academico_	Procedimento responsável pela gestão das atividades agendadas no calendário académico, seja elas individuais assim como gerais.
3	proc_cartão_de_aprendizagem	Procedimento responsável pela visualização das notas dos estudantes
4	proc_curso	Procedimentos responsáveis pela gestão dos cursos na plataforma
5	proc_curso_ano	Procedimentos responsáveis pela gestão dos cursos em função dos anos
6	proc_cursos_do_estudante	Procedimento responsável pela gestão dos cursos no qual o estudante esta alocado
7	proc_disci_inscritas	Procedimentos responsáveis pelas inscrições das disciplinas dos estudantes
8	proc_disciplina	Procedimento responsáveis pela gestão básica das disciplinas,
9	proc_disciplinas_no_plano	Procedimentos responsável pela visualização dos planos das disciplinas, créditos e etc
10	proc_evento_final	Procedimento responsável pela visualização e gestão dos eventos
11	proc_historico_notas_disciplinas	Procedimentos usados para a gestão e visualização das notas dos estudantes em todos cursos e disciplinas participadas na Universidade

12	proc_planocurricular	Procedimento responsável por gerir os planos curriculares e as relações de cursos e disciplinas pertencentes
13	proc_planocurricular	Procedimento para gerir os planos curriculares usados pelos cursos
14	proc_unidade_curricular	Procedimentos para gerir as unidades curriculares dos cursos

Tabela 4-Contendo a lista dos procedimentos armazenados

2.5.2. Organização dos procedimentos armazenados

O procedimento armazenado contém todas as operações necessárias para a manipulação da cada entidade chamada. Como exemplo, procedimento `proc_curso`, que é usado para manipulação dos cursos do sistema, nele encontram-se escritas as consultas para editar, inserir, eliminar e visualizar os dados. Este procedimento recebe parâmetros de entrada, como é ilustrado na figura 5.

```

1 CREATE DEFINER='root'@'localhost' PROCEDURE `proc_curso` (
2     in op char(20),
3     in pag1 int,
4     in pag2 int, in _descricao curso char(200),
5     in _idplanocurricular int,
6     in _idregime int,
7     in _idcurso int
8
9

```

Figura 5-Parâmetros de entrada para o procedimento armazenado proc_curso

Para cada procedimento armazenado, estão escritas todas as operações usadas para a entidade referente ao mesmo, porém, quando chamados, são passados os parâmetros necessários para a execução do caso de uso.

Entre aos parâmetros, são passados os atributos necessários assim como a operação que se pretende executar.

2.5.3. Organização da API

A figura 6 ilustra os ficheiros do diretório mãe do projeto da API, nele contém o ficheiro `server.js` que é o primeiro ficheiro acessado tirando, depois do `access`, A pasta `node_modules`, `well-known` e `tmp` correspondem as bibliotecas disponibilizadas pelo Node.js.

Segundo Chen et al, View Model.Controller (mvc) é um padrão desejável para o desenvolvimento de software para aplicações Web, o mesmo, considera a separação em três elementos ou camadas todo o projeto, sendo: a lógica de controle (saber quais elementos o projeto tem e o que fazer, mas não como implementado), lógica de negócios (saber desenvolver a aplicação) e lógica de apresentação (saber como ela interage o usuário com o aplicativo).

O diretório `app` foi usado para a criação da estruturação do projeto, é nele onde localizam-se as classes, controle, modelo, rotas e mais outras, como ilustra a figura 7. Tratando-se de uma API (*back-end*), não foram criadas quaisquer vistas, uma vez que esta fornece os seus serviços por meio de *endpoints*.

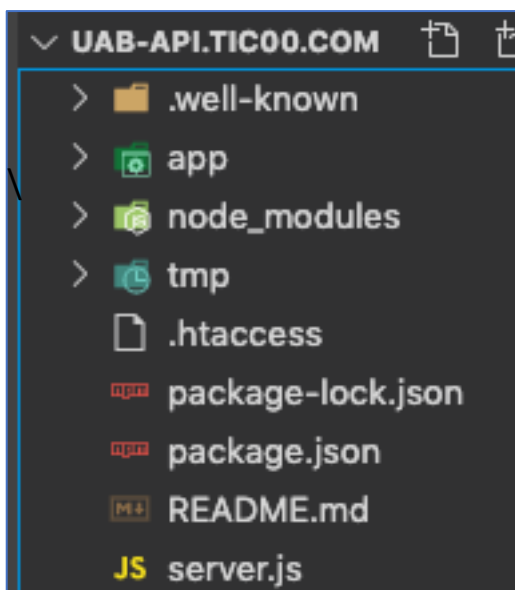


Figura 6- Estrutura do diretório app

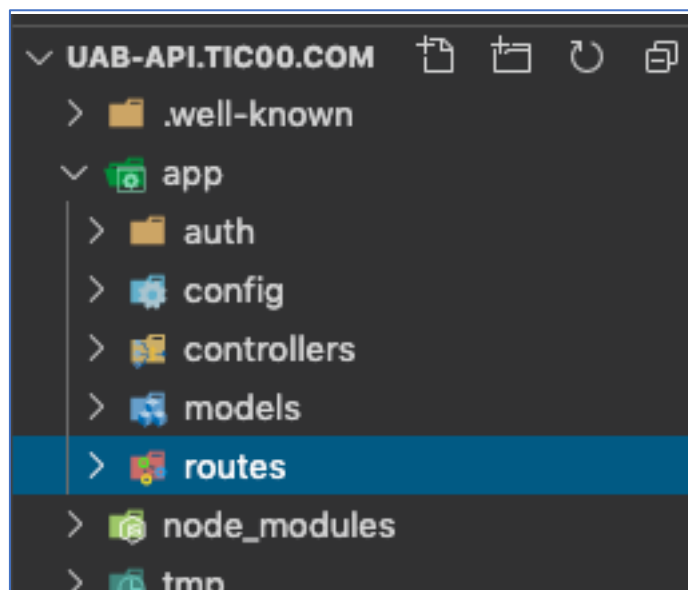


Figura 7- Estrutura do diretório mãe da API

A classe `server` é também responsável pela declaração dos ficheiros com as rotas declaradas no projeto.

Para o desenvolvimento da API, foram usadas as funcionalidades mencionadas nos requisitos funcionais, na figura 1 e, com base nestas, foram criados os *endpoints* da aplicação que são listados na tabela 5.

#	Tipo de rota	Nome do endpoint
01	POST	/usuários
02	POST	/login
03	POST	/usuarios/find
04	GET	/usuarios/todos"
05	PUT	/usuarios/findby/:email
06	GET	/usuarios/:usuariosId
07	PUT	/usuarios/:usuariosId
08	GET	/plano_tematico/:idcurso
09	GET	/inscricoes/:idcurso/:idestudante
10	GET	/historico_notass/:idcurso/:ano/:idestudante
11	GET	/historico_notas/:idcurso/:idestudante
12	PUT	/eventos/atualizar/:id
13	POST	/eventos/inscricao
14	GET	/eventos/:usuariosId
15	GET	/eventos/buscar-por/:id
16	GET	/cursos_feitos_estudante/:usuariosId
17	GET	/disciplinas_curso/:idcurso
18	GET	/uab_cursos
19	GET	/plano_curricular_/:idcurso

Tabela 5-Lista dos endpoints da API

As classes modelos são usadas para interagir com a base de dados, porém, nestas são efetuadas chamadas aos procedimentos armazenados. Foram usados procedimentos na maioria das requisições com intenção de partilhar a carga de processamento de requisições entre os servidores de base de dados e o servidor web, sendo assim, as consultas são efetuadas pelo SGBD (Sistema de gerenciamento de base de dados) MySQL.

Para ilustrar melhor a interação entre as classes modelo e a base de dados observe a figura 8 e 9 que mostra o objeto “evento” e os seus atributos, assim como a sua conexão com a base de dados, por meio dos procedimentos, como foi mencionado anteriormente.

```

bknd_uab > app > models > JS eventos.model.js > ...
1  const sql = require("./db.js")
2  const {genSaltSync, hashSync} = require("bcrypt")
3  const salt = genSaltSync(10)
4
5  // constructor
6
7  const Eventos = function(eventos) {
8    this.data = eventos.data
9    this.idevento = eventos.idevento
10   this.descricao = eventos.descricao
11   this.evento = eventos.evento
12   this.local = eventos.local
13   this.usuarios_id = eventos.usuarios_id
14   this.nome = eventos.nome
15   this.apelido = eventos.apelido
16   this.datacriado = eventos.datacriado
17   this.horafim=eventos.horafim
18   this.horainicio = even
19 }
20
21 Eventos.inserir= (evento, result) => {
22   consulta = "call proc_evento(0,'" + evento.evento+"', '"+evento.descricao+"', '"+evento.local+"', '"+evento.data+"', "+
23   evento.idevento+"', '"+evento.usuarios_id+"');";
24   sql.query(consulta , (err, res) => {
25     if (err) {
26       console.log("Ocorreu um erro: ", err)
27       result(err, null)
28     }
29     console.log("Usuário criados: ", { id: res.usuarios_id })
30     result(null, { id: res.usuarios_id})
31   })
32 }
33

```

Figura 8- Chamadas dos procedimentos armazenados

```

34
35 Eventos.atualizar= (evento, result) => {
36   consulta = "call proc_evento(1,'" + evento.evento+"', '"+evento.descricao+"', '"+evento.local+"', '"+evento.data+"', "+
37   evento.idevento+"', '"+evento.usuarios_id+"');";
38   sql.query(consulta , (err, res) => {
39     if (err) {
40       console.log("Ocorreu um erro: ", err)
41       result(err, null)
42     }
43     console.log("Usuário criados: ", { id: res.usuarios_id })
44     res.message="Atualizado com sucesso.";
45     result(null, res)
46   })
47 }
48
49
50
51 Eventos.add= (evento, result) => {
52
53   let consulta1 = `call proc_evento_final('inserir',${evento.usuarios_id},0,'${evento.evento}','${evento.descricao}','${evento.local}`
54   //console.log(consulta1);
55   sql.query(consulta1 , (err, res) => {
56     if (err) {
57       console.log("Ocorreu um erro: ", err)
58       result(err, null)
59     }
60     console.log("Usuário criados: ", { id: res.usuarios_id })
61     result(null, { id: res.usuarios_id})
62   })
63 }
64
65 Eventos.edit= (evento,id, result) => {
66
67   let consulta1 = `call proc_evento_final('update',${evento.usuarios_id},${id},'${evento.evento}','${evento.descricao}','${evento.lo}`
68   console.log(consulta1);
69   //console.log(evento);

```

Figura 9 - Chamadas dos procedimentos armazenados

A adição de eventos pessoais, as atualizações representam uma das atividades importantíssimas da aplicação, para o exemplo do procedimento, são passados os dados necessários em conjunto com um parâmetro que indica o tipo de operação que o procedimento vai efetuar.

Os resultados dos dados provenientes do acesso a API são dados do tipo Json, por exemplo, para o *endpoint* na tabela 6 obtemos os dados no formato correspondente.

EndPoint	Json resultado
https://uab-api.tic00.com/usuarios/todos	<pre>[{ "id":1, "nome":"Francisco Jaime", "apelido":"Chimbinde", "email":"chimbinde001@gmail.com", "morada":"Matacuane", "palavra":"\$2a\$10\$OU8nc.AXt2BNEOKGrDyeu/BiRlZan2: "nivel":1, "datacriada":"2021-01-25T20:30:17.000Z", "dataeditada":"2021-01-25T20:30:17.000Z", "eliminado":0, "emailsec":"francisco.chimbinde@uzambeze.ac.mz", "idestadosuario":2, "idtipousuario":1, "picture":"", "role":"", "token":"", "tipousuario":"Estudante", "estadosuario":"Activo" }, { "id":2, "nome":"Elio Made",</pre>

Tabela 6-Resultado Json da chamada de um endpoint

2.5.3.1. RESTful

Para o projeto foi usado a tecnologia RESTful API, de modo a dar mais segurança ao aplicativo. Para os acessos a API o usuário deve autenticar-se com um nome de usuário e senha de acesso. Uma vez confirmado os dados do login, a aplicação recebe um *token* válido por 2 horas e é a partir deste que a aplicação cliente consegue ter acesso aos conteúdos do terminal.

As requisições enviadas ao servidor de aplicação onde estará hospedado a API devem conter um cabeçalho https, o tipo de requisição escolhida, assim como o critério de autorização escolhido, para o nosso caso deste projeto foi usado o *bearer token*.

O Bearer token é uma chave para aceder aos recursos protegidos por um OAuth2, este representa uma autorização do servidor, emitida para o cliente. Porém, o cliente deve possuir mecanismos próprios para identificar e validar o token.

Na figura10, está ilustrada o acesso ao endpoint todos usuários a partir da aplicação <https://reqbin.com/> que permite o efetuar testes sobre APIs.

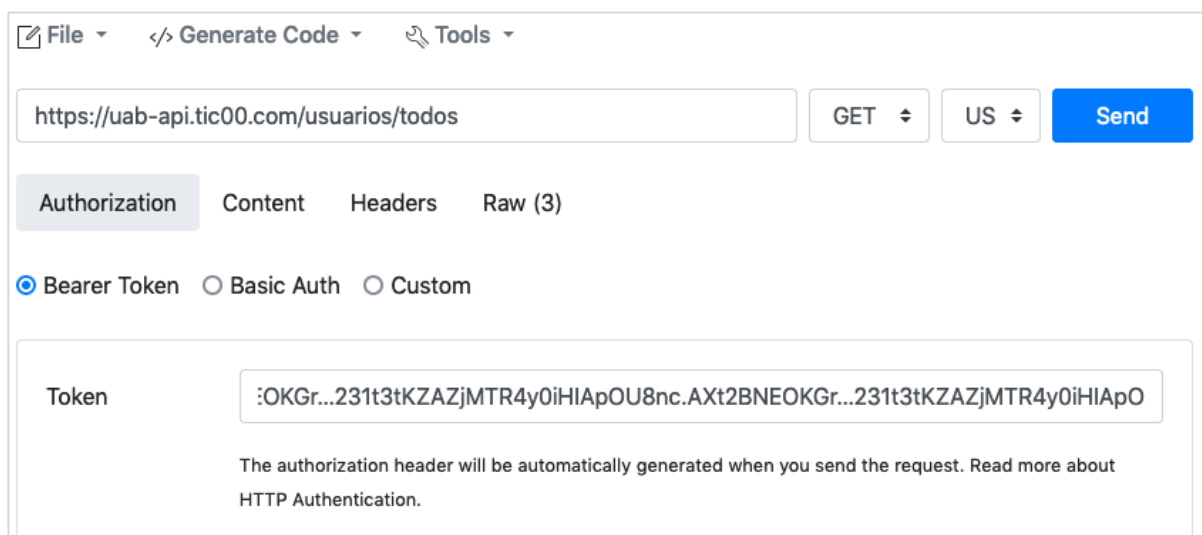


Figura 10-Teste online de acesso a API

2.5.3.2 Instalação da API

O *back-end* da aplicação esta instalada num servidor *cloud* compartilhado e é no momento acessado usando o domínio <https://uab-api.tic00.com>. Para a instalação da API foram seguidos os seguintes passos.

- Criação da base de dado, após a criação desta, foi criado um usuário específico com o devido acesso a base de dados

- Criação do subdomínio, como foi mencionado anteriormente, para aceder a API é necessário que ela esteja disponível num domínio, para o nosso caso foi usado um subdomínio do domínio principal do servidor
- O servidor partilhado da *mochahost* (*empresa de hospedagem de servidores*), possui um serviço denominado *softwares*, nele, são disponibilizadas as ferramentas para instalação do *Framework Node.js*. A figura 11 ilustra o ambiente de gestão das ferramentas *Node.js* usada.

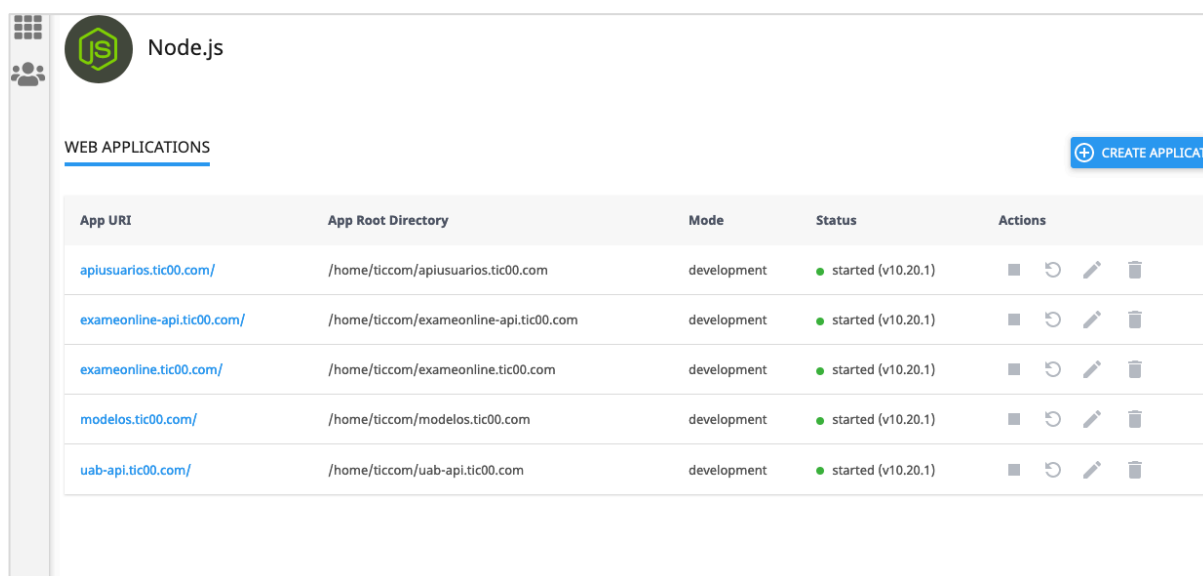


Figura 11-Ambiente de instalação e gestão de aplicações Node.js

2.6. Descrição do *front-end*

Uma vez descrita a parte *back-end* da aplicação, esta parte está reservada para descrição do funcionamento do aplicativo. O *front-end* do aplicativo foi desenvolvido com o *Framework React Native*, que permite a integração e chamadas de APIs *RESTful* de outras tecnologias.

A figura 13 ilustra a estrutura do projeto, o diretório raiz, nela pode-se ver as pastas criadas pelo *Framework*, assim como os recursos criados para suportar os sistemas operativos iOS assim como Android. Esta capacidade de efetuar projetos Android e iOS faz com que React Native firme ao mercado de tecnologias de desenvolvimento de aplicativos móveis.

A pasta `lib` é responsável por guardar todo o código da aplicação, a figura 12 representa a estrutura do diretório `lib`. As classes controladoras, modelos, rotas, vistas dentre outras, estão localizadas em `lib`.

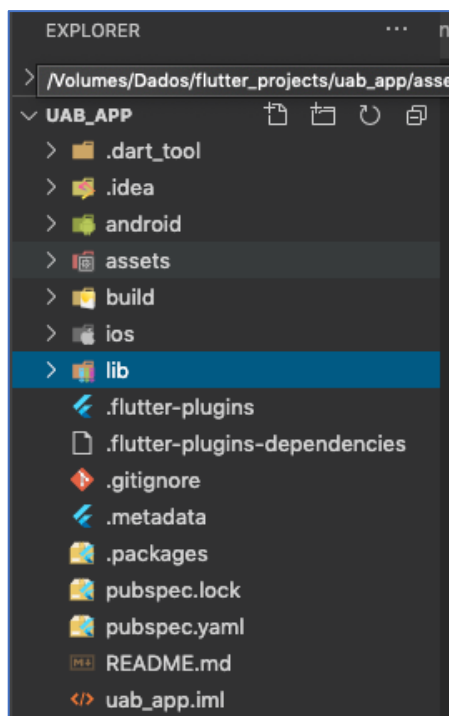


Figura 13-Diretório `lib`, estrutura dos aplicativos

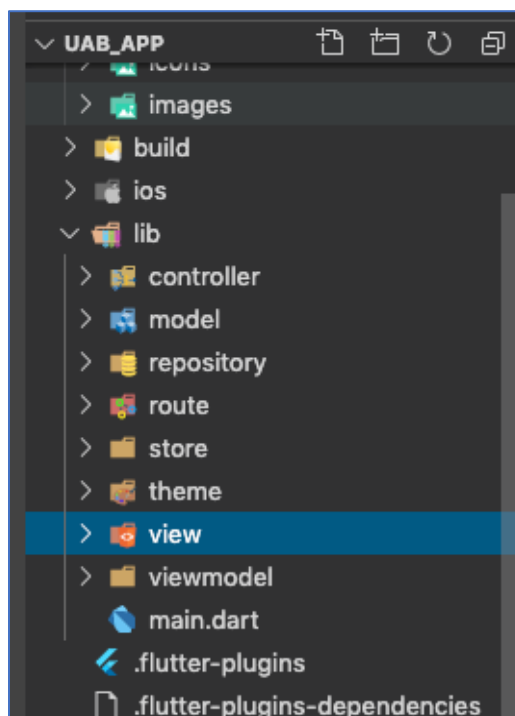


Figura 12-Diretória raiz do aplicativo.

No diretório `views` estão os ficheiros codificados, as interfaces gráficas, como é o exemplo da interface do login, a página `home`, a página das unidades curriculares, histórico das notas e etc. Estas páginas interagem com a outra parte do programa por meio das classes controladoras, `views` e `model`. Para a comunicação com o *back-end*, a aplicação usa a classe `repositor`, nesta estão desenvolvidos todos métodos necessários.

2.7. Funcionamento do sistema

Nesta parte do trabalho aborda-se o funcionamento do aplicativo em função da aplicação vista da interface gráfica do utilizador.

O *front-end* da aplicação foi desenvolvido usando o *React Native*, testado com o simulador do Xcode, e um dispositivo Huawei Y2019.

O Xcode é um ambiente de desenvolvimento integrado (IDE) usado para criar aplicativos para *Mac, iPhone, iPad, Apple Watch e Apple TV*. Esta ferramenta facilita o desenvolvedor nos testes.

O Huawei Y9 2019 é um *smartphone Android*, que tem uma tela *Touchscreen* de 6.5 polegadas com uma resolução de 2340x1080 pixel. O Huawei Y9 2019 possui, um processador 4x 2.2 GHz Cortex-A73 + 4x 1.7 GHz Cortex-A53 com uma memória RAM de 4Gb.

2.7.1. Acesso ao sistema

O acesso à aplicação é feito a nível da interface de login, que por sua vez consulta o *back-end* e a base de dados, porém, esta informação de acesso deve ser explorada do sistema atual da UAb, que fornece ao Aplicativo a possibilidade de consultar se um estudante é ou não da Universidade Aberta.



Figura 14-Interface para o login do aluno

Para que usuário acesse a plataforma basta digitar o seu nome de usuário UAb no campo 1, seguido da sua senha no campo 2. Porém, as credenciais devem ser requisitadas por um formulário de registro ilustrado na figura 14.

2.7.2. Tela inicial

A tela inicial é a primeira tela vista depois de acessar o aplicativo, no entanto, ela inicia com um breve resumo de informações recebidas nas diferentes funcionalidades, tais como, alertas, mensagens recebidas, notícias, ou mesmo informações de eventos.

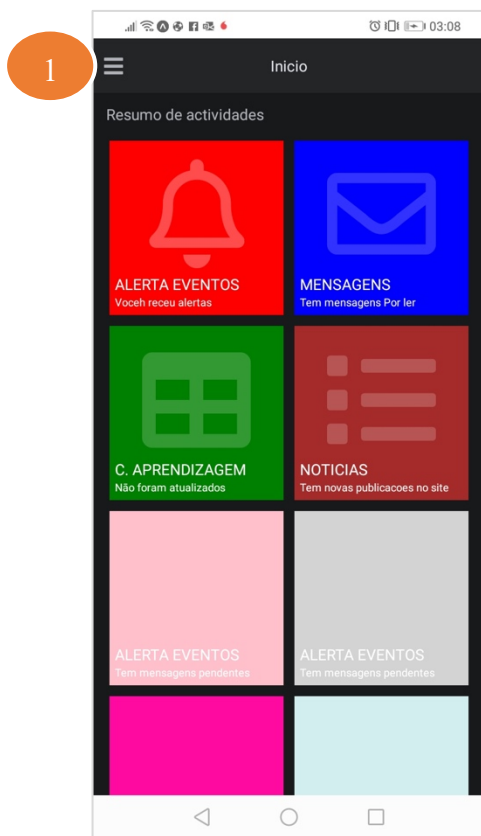


Figura 16-Página inicial

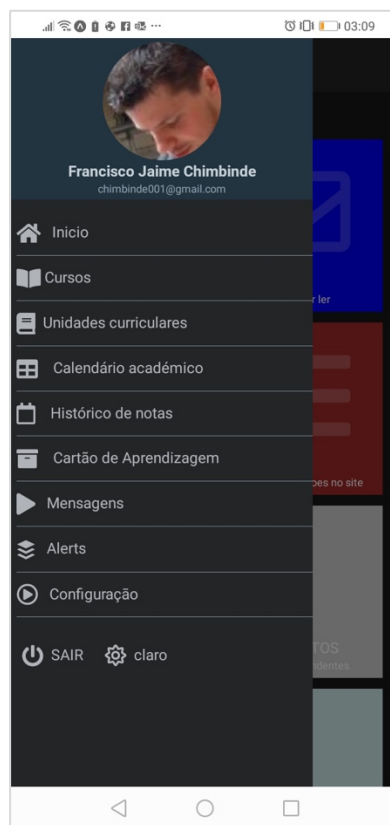


Figura 15-Menu inicial

A página inicial exibe apenas os resumos, porém, para acessar as outras funcionalidades o usuário deve acessar o menu por meio do botão 1 da figura 16, que por sua vez abrirá o menu ilustrado na figura 15.

O *sidebar menu* (menu lateral), é o mecanismo usado para a navegação entre as páginas. As funcionalidades do aplicativo podem ser vistas e selecionadas a partir deste menu. Dentre as funcionalidades podemos selecionar:

- Os cursos, unidades curriculares
- As unidades curriculares
- Calendário académico
- Histórico das notas
- Cartão de aprendizagem
- Alertas
- Mensagens
- Configurações

2.7.3. Cursos

Esta funcionalidade oferece ao usuário a possibilidade de visualizar todos os cursos em que participa na UAb em forma de lista.

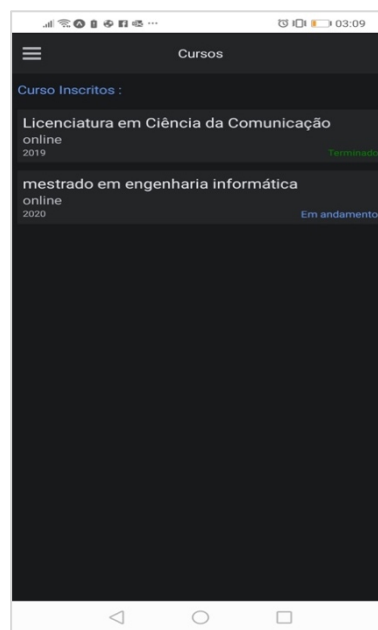


Figura 17-Lista de cursos participados

Na figura 17, pode-se observar mais detalhes sobre os cursos que o estudante se inscreveu, como por exemplo, o estado atual do curso, se o curso já foi terminado ou se ainda está a decorrer.

Pode-se ver também o ano do curso, isto é, como cada curso é lançado em função do ano, associa-se ao estudante o curso, mas também a variável ano, correspondente ao ano civil em que o estudante entrou para o curso.

Pode-se observar o regime do curso também, é importante saber se o curso é exclusivamente online ou tem momentos presenciais.

2.7.4. Unidades curriculares

Os cursos estão ligados a planos de estudos, e por sua vez esses planos ditam as disciplinas relacionadas a estes cursos, assim como as suas horas de contacto. Esta funcionalidade permite que o usuário visualize todas disciplinas de um curso.

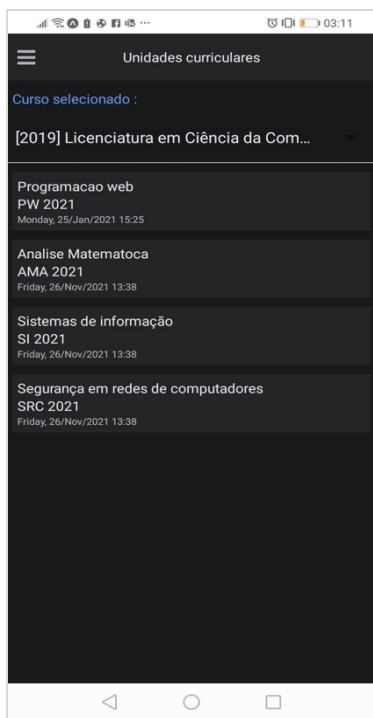


Figura 18-Lista de disciplinas de um curso

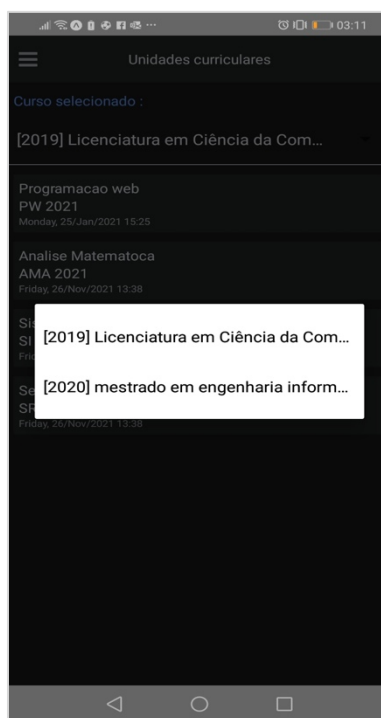


Figura 19-Seleção de cursos

A partir da figura 18, pode-se ver mais detalhes relacionados aos cursos, tal como, o código das disciplinas e a descrição das disciplinas. Para os utilizadores como mais cursos na plataforma, o sistema dá a possibilidade de seleccionar um dos cursos inscritos para visualizar a informação.

A figura 19 ilustra a caixa de combinação “selecione cursos”, que é usada para selecionar os cursos.

2.7.5. Calendário acadêmico

Esta funcionalidade permite que o utilizador tenha sob controle todas as atividades agendadas pela instituição em função da agenda académica para todo o ano. Ela permite também, que o usuário associe a sua agenda individual.

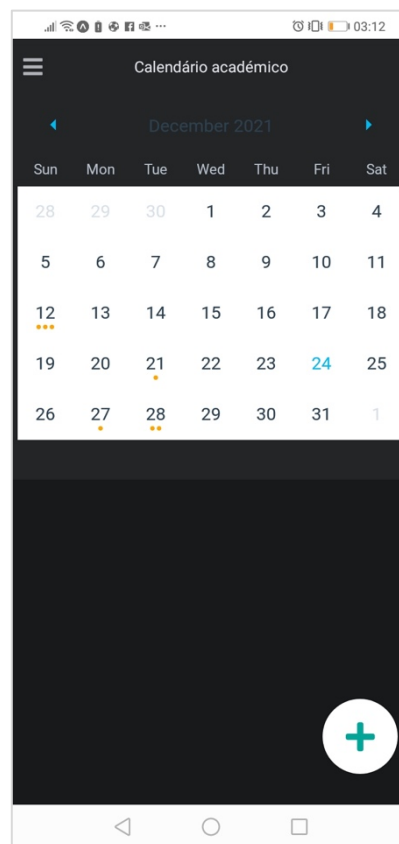


Figura 20-Calendário académico

O calendário académico é representado pelo recurso calendário, contudo, os eventos são marcados nos dias do calendário. Dependendo da quantidade de eventos existentes na data, são usados pontos para assinalar essa relação.

Na figura 20, os dias 27 e 21 estão marcados eventos únicos durante o dia, enquanto que para os dias 12 e 28 existem vários eventos marcados.

Para ver mais detalhes sobre as datas marcadas sobre o calendário, basta selecionar nos dias.

Na figura 21 pode se observar a lista dos eventos marcados num dia.

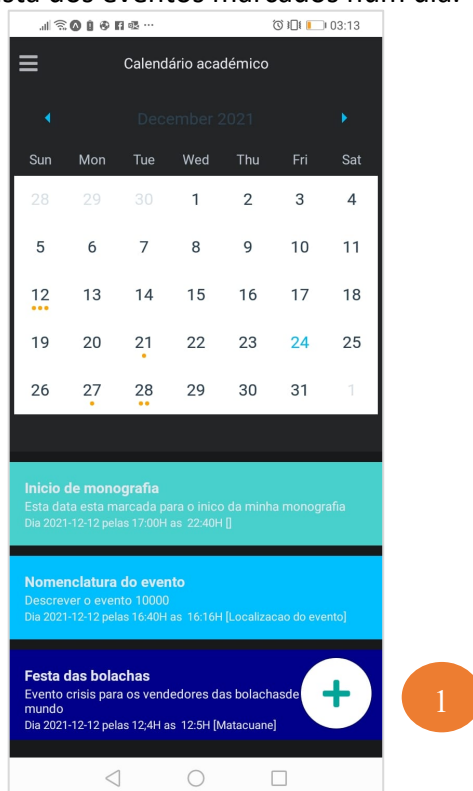


Figura 21- Eeventos de um dia selecionado

A partir da lista de eventos ilustrada na figura 21, pode-se editar os eventos assim como criar eventos pessoais. Os eventos criados pelas contas da UAb são visíveis por todos os usuários, mas somente editados e eliminados por usuários administradores, isto é, com privilégios elevados. O botão ilustrado em 1, é usado para criação de um novo evento.

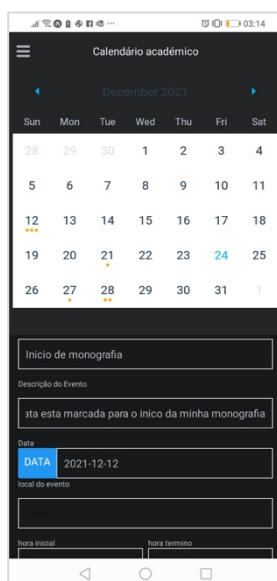


Figura 22-Formulário de cadastro de eventos

O evento pode ser modificado, caso o utilizador digite sobre o item da lista e pode ser criado quando selecionado o evento na lista exibida na figura 21.

2.7.6. Histórico das notas:

Esta funcionalidade, é responsável pela visualização das notas finais adquiridas pelo aluno nas demais disciplinas que participou nos cursos. Esta pode ser usada pelo aluno, para consultar as notas adquiridas.



Figura 23-Lista das disciplinas em função das notas

2.7.7. Cartão de aprendizagem

O cartão de aprendizagem é a funcionalidade responsável pelo registo das atividades de avaliação realizadas pelo aluno em função de uma disciplina. O professor cria várias atividades que são realizadas pelos alunos no decorrer do semestre ou ano letivo.

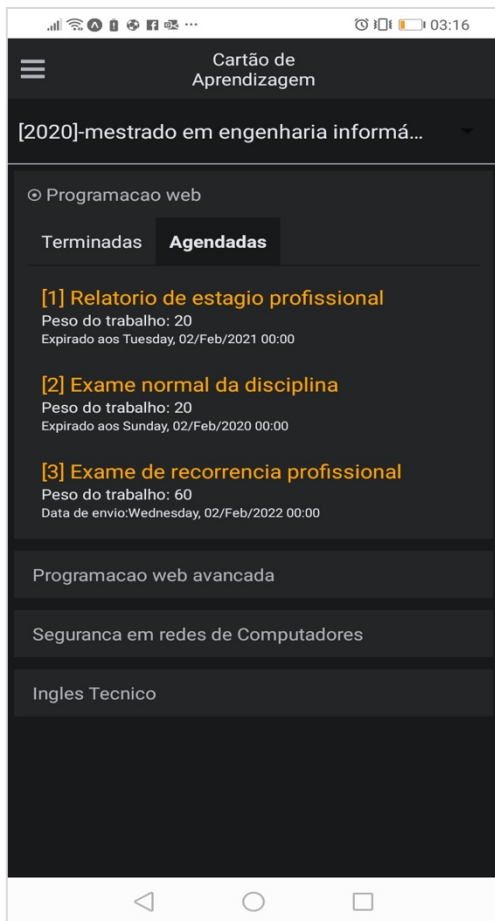


Figura 25-Lista das atividades agendadas

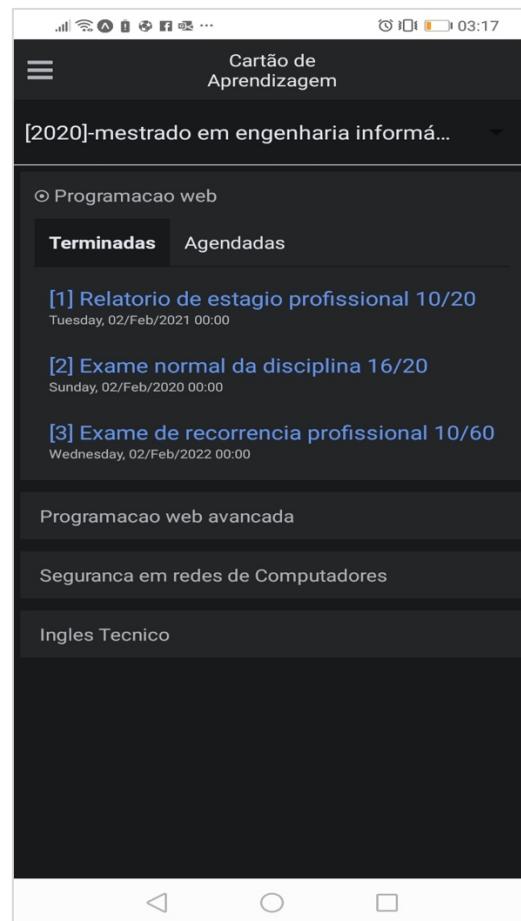


Figura 24-Lista das atividades agendadas e terminadas

3.Avaliação do Protótipo (Inquérito)

Após o desenvolvimento do protótipo procedeu-se à realização de testes com usuários, acompanhado de preenchimento de um questionário relacionado a experiência do utilizador vivida no processo do teste.

Foram inquiridos 20 estudantes da UAb, por meio de um questionário disponibilizado pela ferramenta Google Forms, que foi escolhida pela facilidade de uso que oferece.

Para este processo disponibilizou-se aos usuários uma página *web* contendo a explicação sobre o teste, e instalação do aplicativo via expo:

A Expo é uma ferramenta utilizada no desenvolvimento mobile com React Native, que permite o fácil acesso às API's nativas do dispositivo sem precisar instalar qualquer dependência ou alterar código nativo (expo,2021).

Para efetuar o teste do Aplicativo desenvolvido, o usuário deve instalar o aplicativo expo, que está disponível na Play Store e na App Store. Uma vez instalada, pode aceder ao link da aplicação em teste por meio do código ou endereço disponibilizado pelo expo.

O critério de escolha dos usuários usado foi apenas o fato de ser estudante da UAb, porém, este tem a capacidade de comparar com o funcionamento atual da Universidade.

Data início	Data fim	Utilizadores	Respostas dos questionários
01 de março	27de março	20	14

Tabela 7-Resumo dos dados dos utilizadores

Na fase de análise, foram levados em conta alguns pontos fundamentais, dentre eles:

- Qualidade do *design* (usabilidade, acessibilidade, responsividade)
- Conformidade (satisfação dos usuários)
- Desempenho (ubiquidade, disponibilidade, velocidade)

3.1. Qualidade do design

Para (Ferreira,2008), a usabilidade estuda a relação entre as ferramentas e os seus utilizadores. Para que uma ferramenta seja eficaz, ela deve permitir que os utilizadores realizem as tarefas desejadas e necessárias da melhor forma possível. O estudo da usabilidade procura a utilização fácil e o mapeamento claro das funcionalidades e dos conteúdos de um sistema interativo.

Nesta parte do teste, é importante apurar a usabilidade do Aplicativo, isto porque as interfaces gráficas do utilizador tendem a influenciar no uso. No entanto, pretende-se saber se o utilizador teve ou não dificuldades, deparou-se com erros ao aceder ao Aplicativo, se a interface gráfica do utilizador facilitou.

- Usabilidade:

Para a parte da usabilidade, foram destacados três pontos:

- i. A facilidade de aprendizagem: o usuário pode ou não realizar as suas tarefas e aprender facilmente o funcionamento do aplicativo
- ii. Facilidades de memorização: pretende-se apurar se o utilizador poderá enfrentar as mesmas dificuldades num segundo contacto com o aplicativo.
- iii. Baixa taxa de erros: Uma vez disponibilizado o aplicativo, até em que nível o utilizador realiza as suas tarefas sem transtornos, e em caso de erros, até que nível influenciam na realização das atividades do usuário.

- Disponibilidade:

Outro aspeto fundamental analisado nos testes, foi a questão de acessibilidade, é importante analisar se o Aplicativo esteve acessível para todos usuários, os alunos da UAb que representam o público alvo, assim como para os demais, com acesso aos endereços disponibilizados, mas sem as credenciais disponibilizadas.

- Responsividade:

Uma vez que se trata de uma aplicação que será executada em diversos ambientes ou dispositivos e com dimensões de telas diferentes, é importante ter informação do comportamento nesses ambientes, principalmente, se a interface será ajustável em função do ecrã do utilizador.

3.2. Conformidade (satisfação dos usuários)

Neste ponto, analisa-se o facto de os usuários estarem satisfeitos ou não com a solução, se sentem falta de algo ou se foi difícil se ajustar a solução apresentada.

É fundamental também, saber do utilizador, se esta solução contribui ou não para melhoria de qualidade de serviço na UAb.

3.3. Teste e Funcionamento

O último ponto, levado em consideração foi o desempenho, aqui pretende-se avaliar do aplicativo, questões como desempenho da aplicação, velocidade de processamento, tempo de respostas e o tempo que este ficará disponível

Uma vez que o aplicativo está disponível para Android e iOS, também se procurou saber as características do dispositivo usado pelo usuário nos testes.

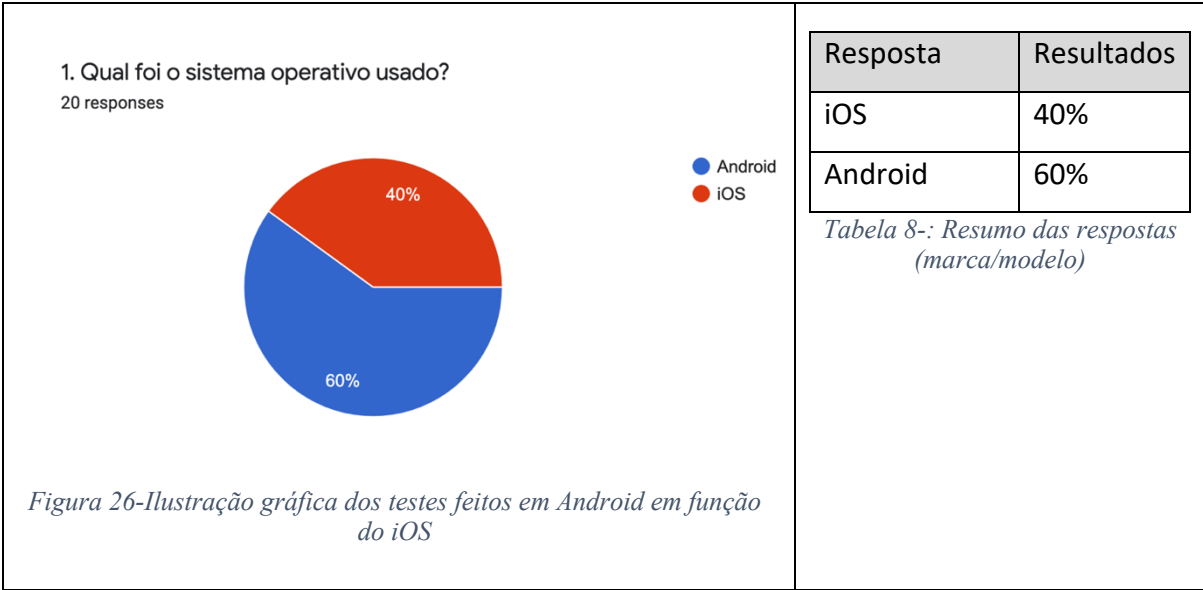
3.4. Análise de Resultados

Com base no questionário e testes efetuados pelos usuários no Aplicativo proposta neste trabalho, agrupou-se os testes para melhor tratamento dos resultados em estudo.

3.4.1. Teste de compatibilidade

Como foi mencionado, foi usado o React Native para o desenvolvimento do aplicativo, sendo que esta ferramenta permite que o desenvolvedor trabalhe em duas versões (Android e iOS) simultaneamente. Tendo em conta esta vertente, foi fundamental saber do usuário:

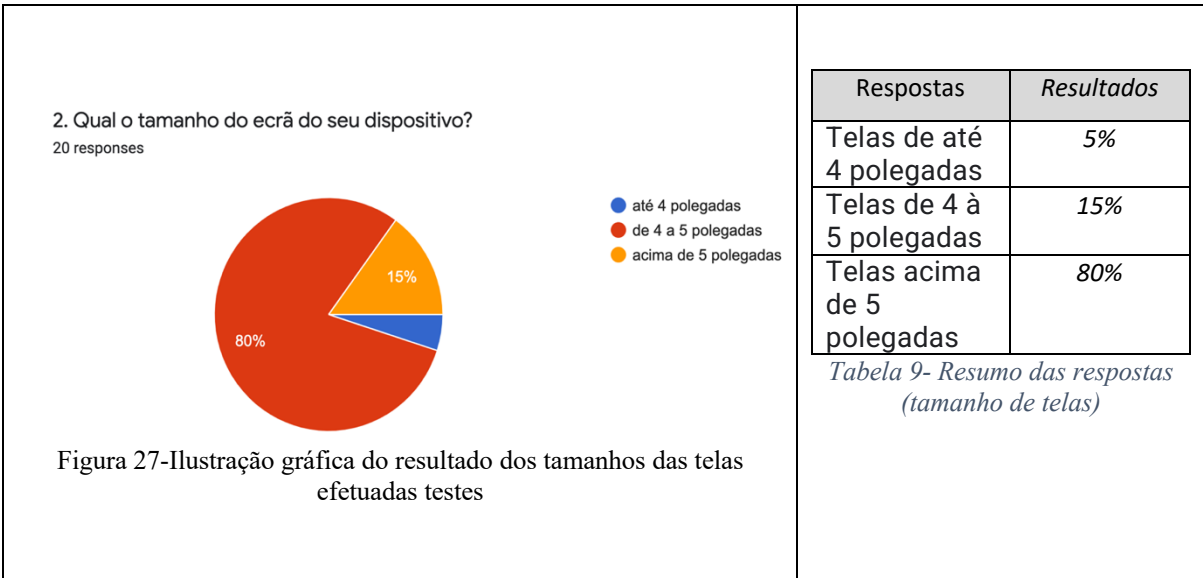
Questão 01: Qual foi o sistema operativo usado no teste?



Tendo em conta os resultados encontrados, pode-se notar que apesar dos usuários de android apresentarem maior número, os usuários do iOS também apresentam um número considerável, quase a metade, contudo, significa que os resultados dos testes são validos para as duas plataformas.

Quando se fala da compatibilidade dos dispositivos, um dos pontos chaves é a responsividade da interface, é importante que os conteúdos ilustrados pela interface dos usuários sejam ajustáveis aos diferentes tamanhos de ecrã, pelo que surgiu a necessidade de questionar:

Questão 02: Qual o tamanho do ecrã do seu dispositivo?



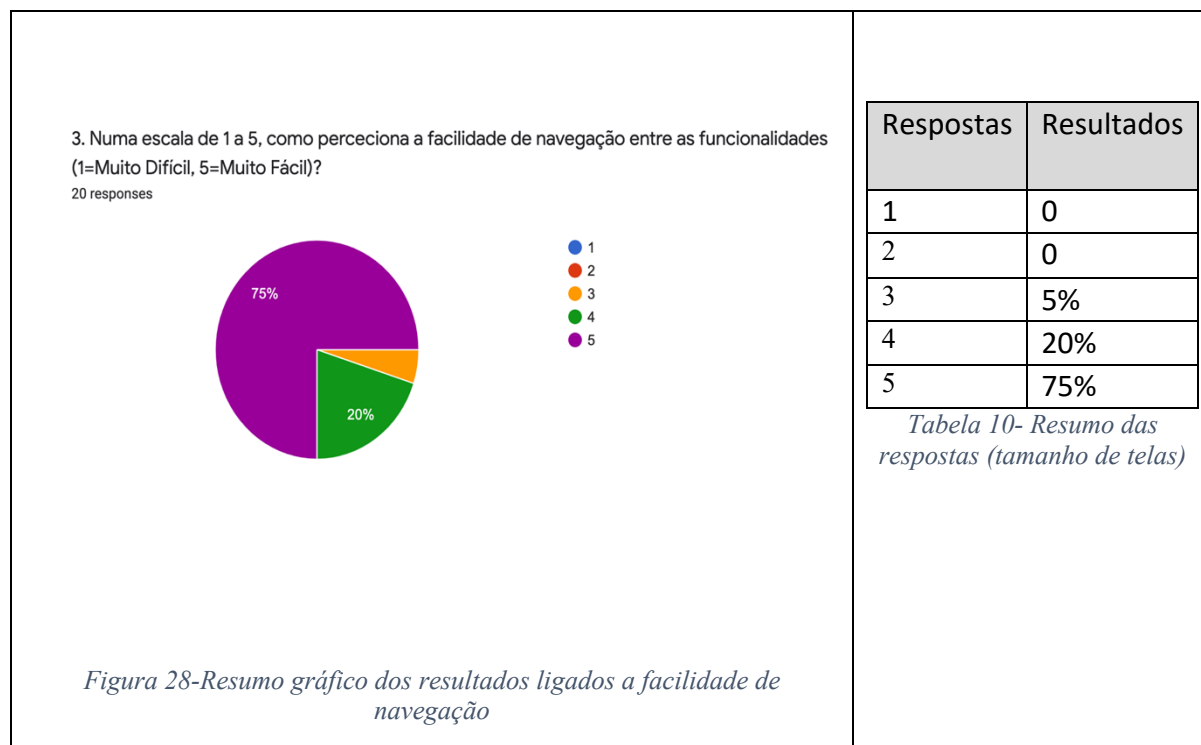
Segundo o gráfico dos dados dos questionários, 80% representam ecrã de telas de 4 a 5 polegadas, enquanto 15% representam telas acima de 5 polegadas(tabletes), os restantes 5% correspondem a telas abaixo de 4 polegadas.

Devido ao menor número de telas de 4 polegadas, não foram validados os dados para telas abaixo de 4 polegadas, contudo, usar-se-ão apenas para telas de dimensão igual ou superior a 4 polegadas.

3.4.2. Testes de Usabilidade

Em termos de usabilidade, é importante que o utilizador não enfrente dificuldades ao usar o Aplicativo, contudo, achou-se fundamental saber dos usuários o grau de dificuldade enfrentadas no momento da navegação das diferentes funcionalidades desenvolvidas, daí que se elaborou a seguinte questão:

Questão 03: Numa escala de 1 a 5, como perceciona a facilidade de navegação entre as funcionalidades (1=Muito Difícil, 5=Muito Fácil)?



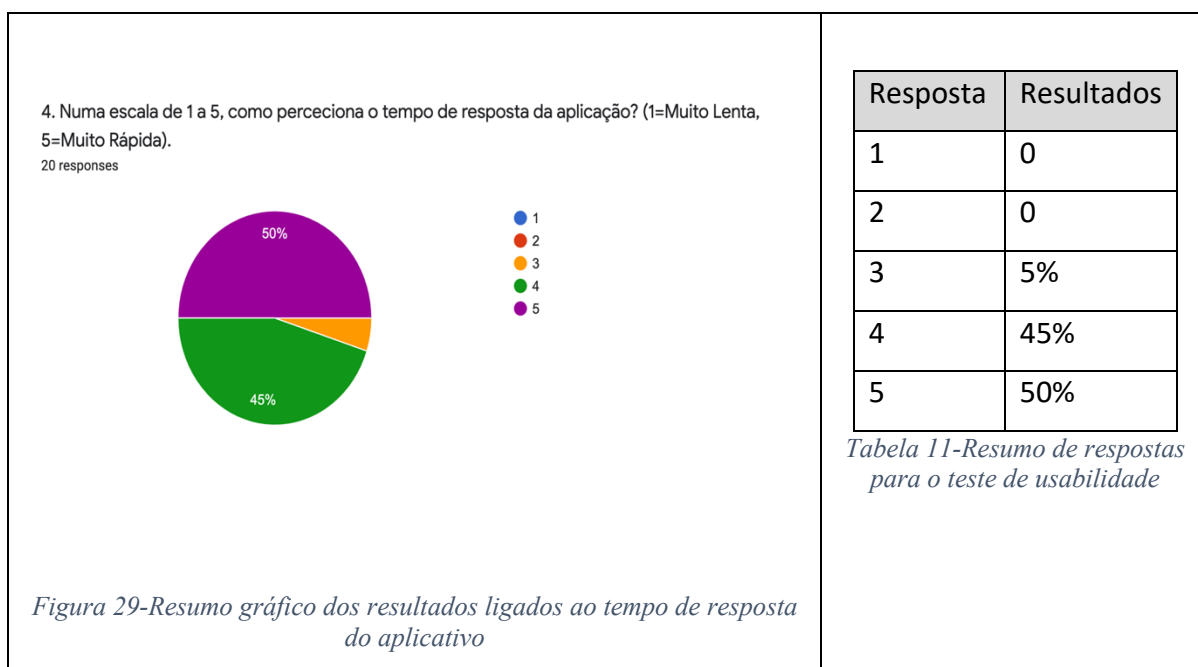
Segundo os dados, 75% dos usuários considera a navegabilidade do Aplicativo muito fácil, 20% considera fácil. Apenas 5% estão divididos entre fácil e difícil.

Tendo em conta esses dados, pode-se dizer que o aplicativo apresenta uma navegabilidade aceitável.

3.4.3. Testes de Desempenho

Para um produto aceitável é fundamental ter em conta o seu desempenho, levar em conta o tempo de processamento pode melhorar a experiência do utilizador e permitir que o mesmo confie mais no Aplicativo. Dai que se achou fundamental elaborar a seguinte questão:

Questão 04: Numa escala de 1 a 5, como perceciona o tempo de resposta da aplicação? (1=Muito Lenta, 5=Muito Rápida).



Em termos de processamento, pode-se notar que 50% dos usuários considera muito rápida, 45% consideram rápida, enquanto que os 5% restantes são neutros.

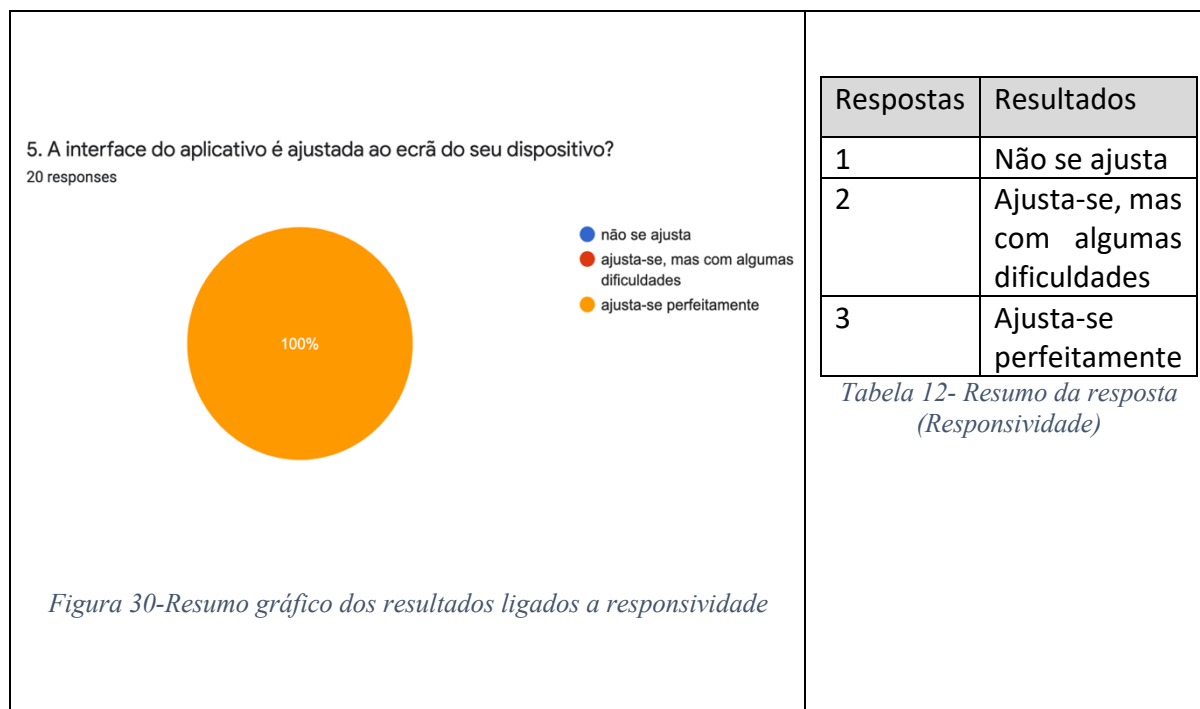
Pode-se concluir que, em média, o tempo de resposta do Aplicativo é aceitável.

3.4.4. Testes de Responsividade

Quando se fala de aplicações híbridas, o assunto responsividade tem sido um ponto importante, pois, a capacidade da mesma aplicação se ajustar a diferentes écrans com diferentes tamanhos

chega a contribuir para que os usuários usem com facilidade, sendo assim, se considerou importante saber se:

Questão 05: A interface do aplicativo é ajustada ao ecrã do seu dispositivo?



Segundo os dados, 100% dos usuários responderam a favor da responsividade do Aplicativo, o que leva a concluir que a interface se ajusta, porém, este resultado só não pode ser considerado para os ecrãs menores de 4 polegadas, uma vez que houve insuficiência de testes nessa dimensão.

Uma vez submetidos aos testes, e tendo em conta que se trata de uma aplicação ainda em desenvolvimento, a necessidade de coletar informação sobre problemas ligados ao funcionamento da aplicação foi levada em consideração, pelo que houve a necessidade de questionar o utilizador se teve problemas no durante o uso ou não.

3.4.5. Outros Resultados

Questão 06: Numa escala de 1 a 5, a aplicação funcionou sem problemas no seu dispositivo? (1=Muitos problemas, 5=Sem quaisquer problemas).

6. Numa escala de 1 a 5, a aplicação funcionou sem problemas no seu dispositivo? (1=Muitos problemas, 5=Sem quaisquer problemas).

20 responses

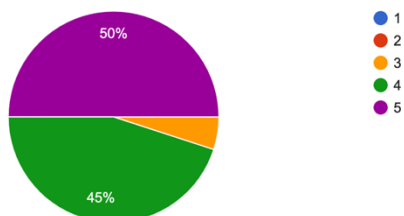


Figura 31-Resumo gráfico dos resultados ligados aos problemas no momento dois testes

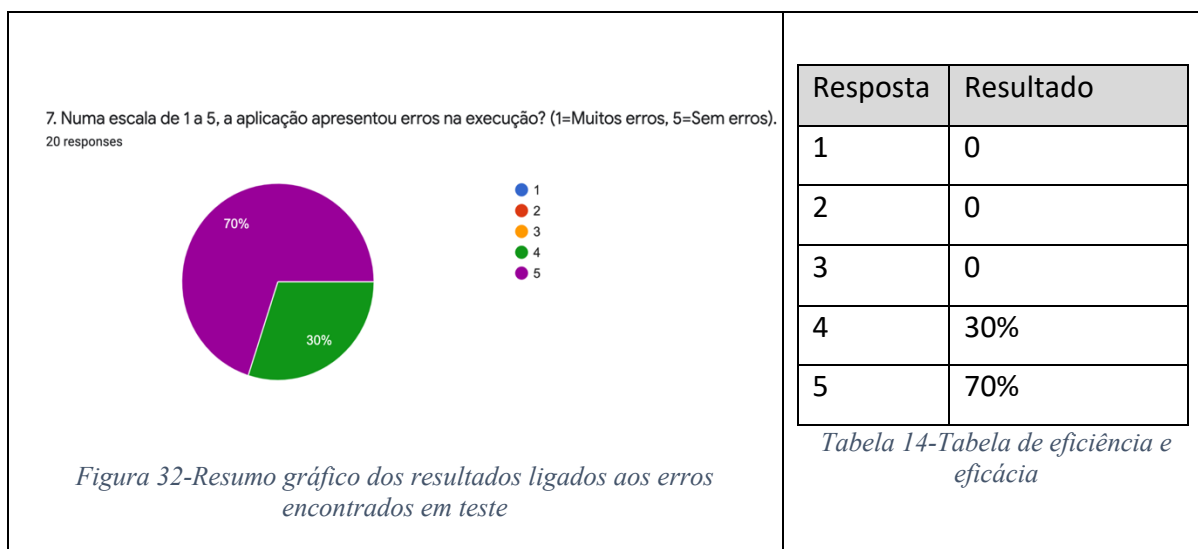
Resposta	Resultado
1	0
2	0
3	5%
4	45%
5	50%

Tabela 13- resumo da resposta (problemas no funcionamento)

Segundo os dados ilustrados, 50% dos inquiridos mencionam não ter quaisquer problemas, 45% também mencionam não ter experienciado grandes problemas. Assim, grande parte dos utilizadores não registou problemas de maior no uso do Aplicativo, no momento dos testes.

Tirando os problemas no processo de instalação do protótipo para os testes, considerou-se importante saber se os usuários obtiveram algum erro durante a execução da Aplicação, pelo que se questionou:

Questão 07: Numa escala de 1 a 5, a Aplicação apresentou erros na execução? (1=Muitos erros, 5=Sem erros).

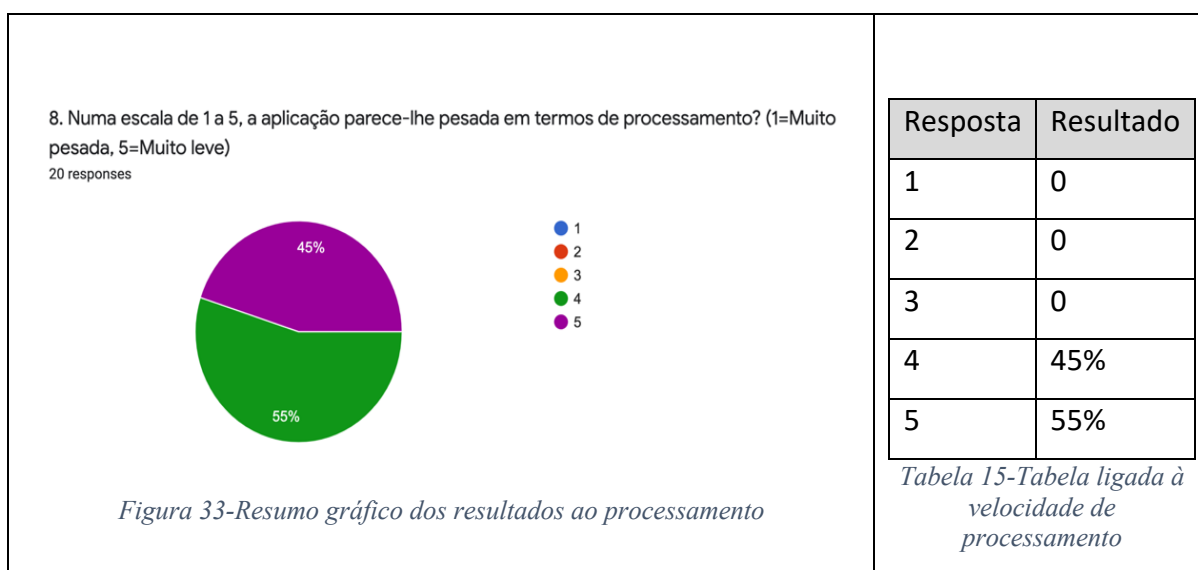


Segundo os dados, 70% dos utilizadores não tiveram erros no momento dos testes, enquanto que os 30% restantes se aproximam a execuções em erros.

Tendo em conta os dados, pode se dizer que não foram detetados erros para uma maioria dos utilizadores que efetuou os testes, no entanto, considerou-se importante determinar a natureza dos erros experienciados por 30% dos utilizadores.

Para consolidação da questão relacionada ao processamento, houve necessidade de ter informações ligadas ao tempo dos pedidos o Aplicativo, tendo-se perguntado aos usuários:

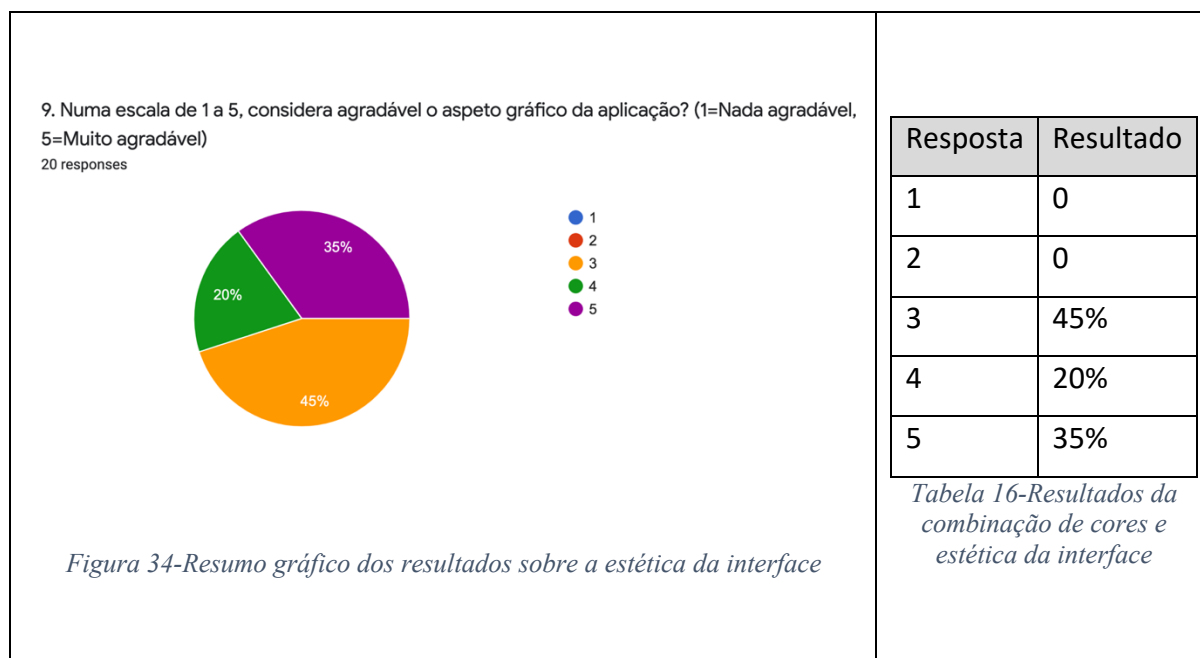
Questão 08: Numa escala de 1 a 5, o Aplicativo parece-lhe pesada em termos de processamento? (1=Muito pesada, 5=Muito leve)



Segundo os dados obtidos, 45% dos usuários consideram o Aplicativo muito leve, ao passo que 55% considera leve.

Pode-se concluir que 100% destes testes indicam que o Aplicativo é leve quando executado. E como a satisfação do utilizador é um indicador para futuras melhorias, considerou-se importante questionar sobre o aspeto gráfico do Aplicativo.

Questão 09: Numa escala de 1 a 5, considera agradável o aspeto gráfico do Aplicativo? (1=Nada agradável, 5=Muito agradável)

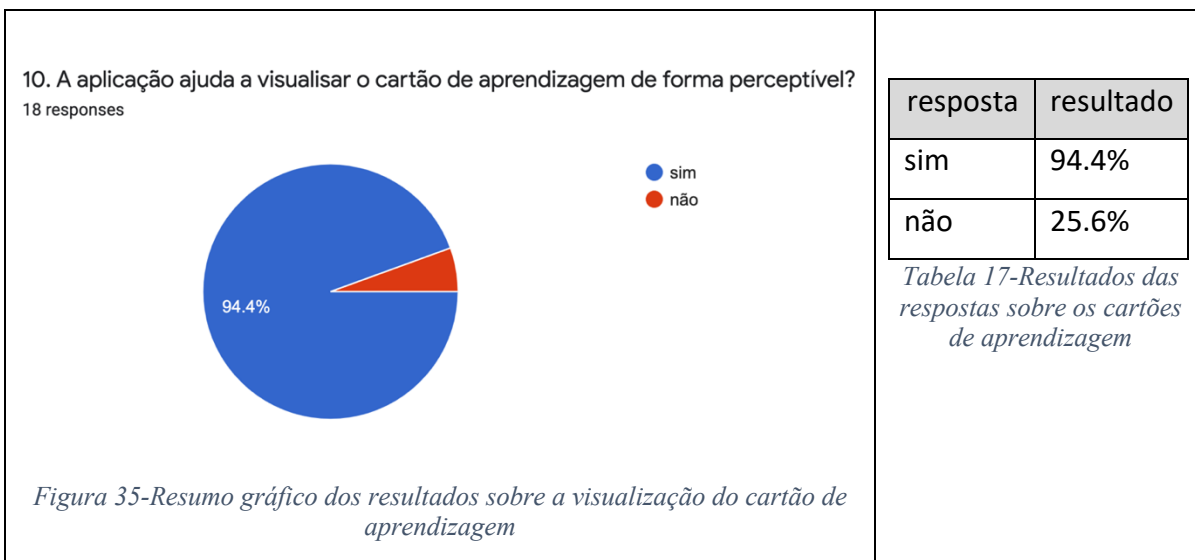


Segundo os resultados obtidos, 45% mostrou-se neutro, 20% considera agradável, enquanto que os restantes 35% considera muito agradável.

Sendo assim, pode dizer que 55% dos utilizadores consideram agradável, mas há aqui um grande número de usuários que não manifestaram tendência.

Tendo em conta as funcionalidades desenvolvidas, é importante verificar o funcionamento saudável do Aplicativo, mas também é fundamental averiguar se as funcionalidades foram concluídas com sucesso, daí que houve necessidade em questionar se:

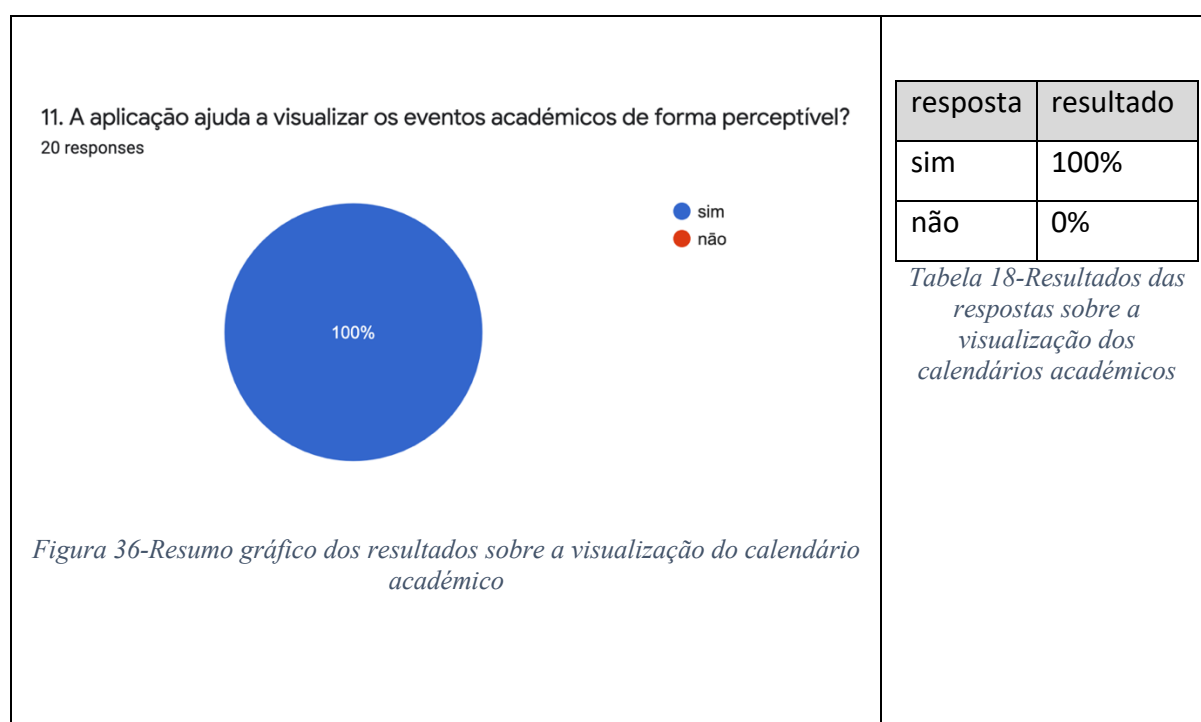
Questão 10: O Aplicativo ajuda a visualizar o cartão de aprendizagem de forma perceptível?



Segundo os dados coletados, 94.4% dos inquiridos mencionaram que o Aplicativo apresenta uma funcionalidade capaz de ilustrar o cartão de aprendizagem de forma perceptível.

Em relação a efetivação dos requisitos funcionais da Aplicativo, foi desenvolvida a funcionalidade que permite a visualização dos eventos acadêmicos, porém é fundamental saber se:

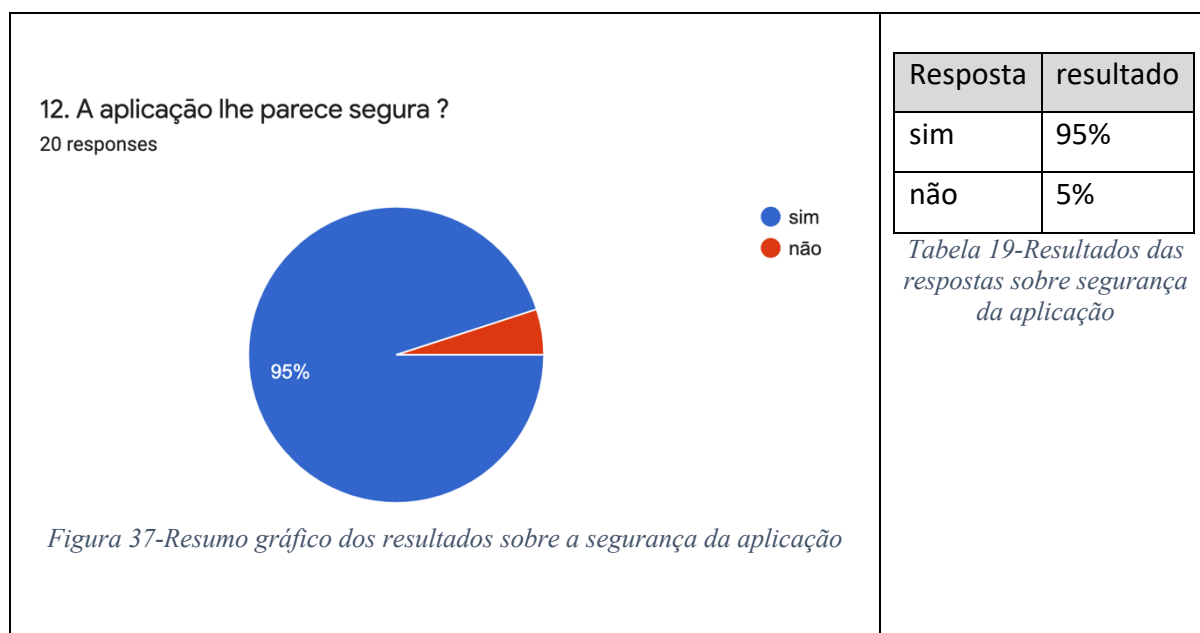
Questão 11: O Aplicativo ajuda a visualizar os eventos acadêmicos de forma perceptível?



Segundo os dados coletados, todos os usuários apontaram a existência da funcionalidade para verificar os eventos acadêmicos com sucesso.

Em relação aos diversos acessos efetuados pelos usuários no momento de testes, houve a necessidade de questionar ao utilizador sobre aspetos ligados a segurança da informação, porém, questionou -se:

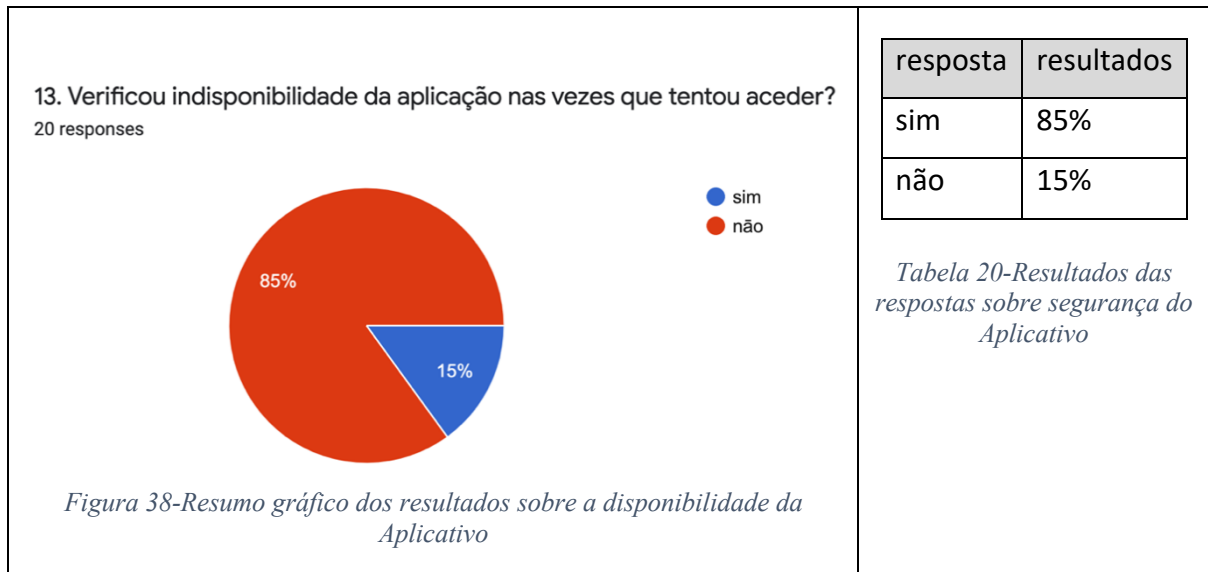
Questão 12: O Aplicativo lhe parece segura?



Segundo os dados obtidos, 95% dos utilizadores consideram o Aplicativo seguro enquanto que os 5% restantes tem opinião diferente. Sendo assim, pode-se considerar o Aplicativo seguro em função aos testes efetuados.

Devido aos recursos usados para testes que obrigam que haja disponibilidade de 3 pontos diferentes para a efetivação de testes com sucesso (servidor da API, servidor das expo e instalação do Aplicativo no dispositivo), houve necessidade de questionar sobre a disponibilidade do protótipo para testes. Dai a seguinte questão:

Questão 13: Verificou indisponibilidade da aplicação nas vezes que tentou aceder?



85% dos usuários menciona ter a plataforma disponível para a efetivação dos testes enquanto que os restantes 15% menciona ter problemas de disponibilidade ao aceder.

Para coletar mais opiniões para melhorias, abriu-se um espaço para que estes deixassem as devidas recomendações futuras.

Questão 14: Que sugestões gostaria de deixar para o melhoramento do Aplicativo?

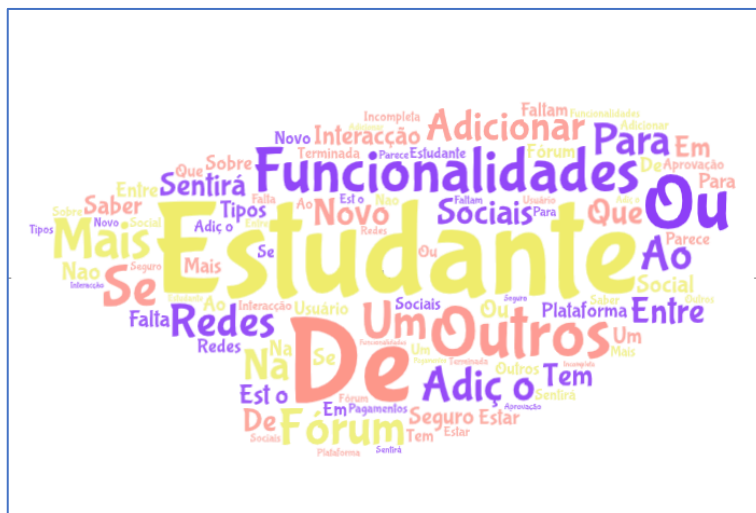


Figura 39-Resumo gráfico dos resultados sobre as recomendações e observações dos usuários

Em relação as sugestões, apenas 15% dos 100% usuários preencheu, pelo que se decidiu invalidar a questão devido a fraca adesão dos usuários.

Tendo em conta os objetivos traçados para o trabalho, o questionário não só permitiu a análise ao nível da completude das metas, mas também permitiu saber se as metas foram efetuadas da melhor forma possível.

Em resposta a questão 10 dos questionários, 94,4% dos utilizadores mencionaram a existência de um cartão de aprendizagem perceptível para o usuário, o que significa que a experiência vivida pelos utilizadores no momento do teste em relação a esta funcionalidade foi muito boa, porém, este ponto representa um dos objetivos traçados para o trabalho. Sendo assim, pode se dizer que o objetivo em questão foi alcançado com sucesso.

Em resposta à questão 11, os usuários inquiridos foram unânimes na existência de uma funcionalidade capaz de visualizar os eventos académicos da UAb de forma perceptível. Isto significa que o impacto criado sobre a experiência dos usuários corresponde aos objetivos traçados para o trabalho foi bom. Contudo, pode se concluir que este objetivo foi atingido com sucesso.

Em resposta as questões 10 e 11, que apresentam indicadores favoráveis do cumprimento dos objetivos parciais do trabalho, e tendo em conta que a camada *front-end* depende totalmente da *back-end*, isto é, o desenvolvimento dos *endpoints* influenciam diretamente o funcionamento da camada cliente, pelo que podemos concluir que o objetivo que recai sobre “o desenvolvimento de *back-end*” fica automaticamente cumprido.

4. Conclusões e trabalho subsequente

O avanço da tecnologia permitiu que os dispositivos móveis ganhassem mais adesão, devido à sua facilidade de uso. O processo de Ensino e aprendizagem aderiu a este recurso como um recurso facilitador, tanto na área académica, onde estudantes podem aceder aos conteúdos, assim como no processo de interação Universidade versus estudante, onde o estudante se mantém informado sobre outros assuntos. Várias Universidades passaram a desenvolver meios para o uso dos dispositivos móveis como uma forma de reforçar as tecnologias já existentes. Tendo em conta os trabalhos feitos na UAb, a realidade mostra que esta ainda não possui uma Plataforma mobile capaz de facilitar explorar a informação por meio de um dispositivo, o que permitiu levantar o problema de pesquisa: a implementação de um Aplicativo móvel capaz de disponibilizar informações básicas do portal UAb e outros sistemas da Universidade poderá facilitar a interação aluno versus UAb?

Sendo assim, o trabalho foi focado no desenvolvimento de um Aplicativo móvel capaz de facilitar a mediação entre o estudante e a UAb.

O trabalho foi desenvolvido em etapas, que consistiram em desenvolver uma API (*back-end*) usando Node.js, uma base de dados em MySQL, modelada a partir de um estudo exaustivo sobre o funcionamento dos sistemas de informação da UAb. Em seguida, desenvolveu-se algumas funcionalidades usando o React Native, e por fim efetuaram-se testes com usuários. Para a avaliação do protótipo foram criados questionários, onde se usou o Google Forms para coleta de dados, a plataforma expo para facilitar a instalação do protótipo de teste do Aplicativo e um servidor partilhado para alojar o *back-end* e a base de dados.

O questionário contém perguntas fechadas relacionadas com as funcionalidades assim como a performance, usabilidade, responsividade, satisfação do usuário dentre outros aspetos, porém, foram desenvolvidas algumas funcionalidades e questionadas sobre o funcionamento destas aos utilizadores, esperavam-se mais testes, mas, só foi possível trabalhar com 20 usuários.

Para a efetivação do teste, se usou como critério único, o facto de ser estudante da UAb.

No que diz respeito à completude dos requisitos traçados, quando questionados aos utilizadores sobre o acesso as funcionalidades visualização dos cartões de aprendizagem, visualização dos eventos, os dados mostraram que a maior parte destes teve acesso as funcionalidades e ficaram satisfeitos com o funcionamento.

Quando questionados sobre a eficiência, a disponibilidade, erros encontrados nos testes, as respostas foram favoráveis, os dados recolhidos mostram que não foram encontrados erros e poucos usuários tiveram problemas no momento dos testes.

Considerando o problema de pesquisa e os objetivos traçados, foi desenvolvido um Aplicativo capaz de facilitar o acesso as informações da UAb aos estudantes com base em serviços web, tal como proposto no trabalho.

A implementação do Aplicativo num ambiente melhorado foi limitada por diversos fatores, dentre eles, a falta de acesso das APIs da UAb, contudo, algumas funcionalidades relevantes que não fizeram parte do foco deste trabalho, podem ser consideradas em trabalhos futuros, em função da disponibilidade de uma API geral que defina a interface com os sistemas da UAb. A realidade da UAb em termos de arquitetura será mais visível desta forma, o que permitirá trabalhar e contribuir com ideias inovadoras, tanto para a API em si como para o Aplicativo móvel.

Entre as funcionalidades futuras no sentido de melhoria do Aplicativo e da prestação de serviços ao estudante, considera-se relevante o desenvolvimento de funcionalidades para gestão de pagamentos de propinas usando o Aplicativo, inscrições, pedidos de levantamentos de certificados, envio de requerimentos, entre outras.

5. Bibliografia

5.1. Referências Bibliográficas

Cevallos, M. del C. (2009). Jóvenes y celulares: las razones de su uso y abuso. Acedido aos 2 de maio de 2012, <http://www.ciespal.net/ciespal/images/docu/produccionacademica/Informe%20celulares.pdf>.

Costeira, S. A. B. (2019). Análise e desenvolvimento de aplicação móvel para portal académico (NetP) (Doctoral dissertation).

Chen, Y., Wang, Y. K., Ugur, K., Hannuksela, M. M., Lainema, J., & Gabbouj, M. (2008). The emerging MVC standard for 3D video services. *EURASIP Journal on Advances in Signal Processing*, 2009, 1-13.

De Almeida, M. E. B. (2012). Integração das tecnologias de informação e comunicação na Educação do Brasil e de Portugal: convergências e especificidades a partir do olhar de professores. *Psicologia da Educação*. Programa de Estudos Pós-Graduados em Educação: Psicologia da Educação. ISSN 2175-3520, (35), 171-196.

Demirbilek, M. (2010). Investigating attitudes of adult educators towards educational mobile media and games in eight European countries. *Journal of Information Technology Education*, v9, p 235-247, 2010.

Engelmann, L. K., & Sápiras, L. A. (2016). MobIES: Aplicativo Integrado de Serviços para Instituições de Ensino Superior. *Anais da III Escola Regional de Sistemas de Informação do Rio de Janeiro*, 2016. v. 1. p. 126-133.

Ferreira, A. C. D. S. M. (2008). Usabilidade e Acessibilidade no design para a Web.

Forman, G. H.; Zahorjan, J. (1994). The Challenges of Mobile Computing. *IEEE Computer*, USA, v.27, n. 4, (Apr.) p.38-47.

Gonçalves, G. S., Bastos, P. H. O., & de Oliveira, D. (2014, November). O Uso de WebSockets no Desenvolvimento de Sistemas Baseados em uma Arquitetura Front-end com API. In Anais da I Escola Regional de Sistemas de Informação do Rio de Janeiro (pp. 57-63). SBC.

Johnson, L., Adams, S., Haywood, K., (2011). The NMC Horizon Report: 2011 K-12 Edition. Austin, Texas: The New Media Consortium.

Kukulska-hulme, A.; Traxler, J. (2005). Mobile Learning: A handbook for educators and trainers. Routledge, 2005.

Lasen, A. (2002). A Comparative Study of Mobile Phones Use in Public Places in London, Madrid and Paris. Acedido ao dia 2 de maio de 2012 em <http://ronik7.files.wordpress.com/2010/01/compstudy.pdf>.

Leite da Silva, F., Nascimento Lamper, V. D., & Ludtke Ferreira, A. P. (2020). Desenvolvimento de API para estimativa da pegada de carbono na pecuária de corte. In XII Congreso de AgroInformática (CAI 2020)-JAIIO 49 (Modalidad virtual).

Miguel, R. (2017). Interface De App Para Dispositivos Móveis De Universidade Virtual, Caso Da UAb. Tese de mestrado. UNIVERSIDADE ABERTA.

Neto, J. F. B., & da Fonseca, F. D. S. (2013). Jogos educativos em dispositivos móveis como auxílio ao ensino da matemática. RNOTE-Revista Novas Tecnologias na Educação, 11(1).

Oliveira, J. M. G. (2017). Desenvolvimento de um aplicativo para controle acadêmico em dispositivos móveis. Dissertação (Mestrado em Inovação Tecnológica) - Programa de Mestrado Profissional em Inovação Tecnológica, Universidade Federal do Triângulo Mineiro, Uberaba, 2017.

Reactnative.dev. (2021). Introdução React Native Disponível em: <https://reactnative.dev/docs/getting-started> [Acedido aos 29 de Dezembro de 2021].

Reactjs.org. (2021). *Primeiros passos - React*. [online] Disponível em: <https://reactjs.org/docs/getting-started.html> Acedido aos 29 de Dezembro de 2021.

Robinson, M.; Kalakota, R.(2001) M-business: The Race To Mobiliy. Business Integration Journal, Dallas

Tarouco, Liane. M. R. et al. (2004). *Objetos de Aprendizagem para M-Learning*. Florianópolis: SUCESU - Congresso Nacional de Tecnologia da Informação e Comunicação.

Teixeira Júnior; 2005. Ele Mudou até a Vida. Revista Exame. Abril; Ed. 844; Ano 39; n.11; 8 de junho de 2005.

5.2. Documentação adicional e páginas web

Documentação da Expo. 2021. *Introdução à Expo - Documentação da Expo*. [online] Disponível em: <<https://docs.expo.dev/>> [Acedido em 29 de dezembro de 2021].

Expressjs.com. 2021. *Express - Node.js framework de aplicações web*. [online] Disponível em: <<https://expressjs.com/>> [Acedido em 29 de dezembro de 2021].

Expressjs.com. 2021. *Roteamento expresso*. [online] Disponível em: <<https://expressjs.com/en/guide/routing.html>> [Acedido aos 29 de Dezembro de 2021].

Node.js. 2021. Documentação Node.js , <https://nodejs.org/en/docs/> Acedido aos 29 de Dezembro de 2021].

Npm. 2021. axios . [online] Disponível em: <https://www.npmjs.com/package/axios> Acedido aos 29 de dezembro de 2021].

Npm. 2021. bcrypt . [online] Disponível em: <<https://www.npmjs.com/package/bcrypt>> [Acessado em 29 de dezembro de 2021].

Npm. 2021. cors . [online] Disponível em: <<https://www.npmjs.com/package/cors>> [Acessado em 29 de dezembro de 2021].

Npm. 2021. *react-native-add-calendar-event* .Disponível em: <https://www.npmjs.com/package/react-native-add-calendar-event>, Acedido em 29 de dezembro de 2021].

Npm. 2021. *jsonwebtoken* . Disponível em: <https://www.npmjs.com/package/jsonwebtoken> Acedido em 29 de Dezembro de 2021].

npm. 2021. *momento-reação* . Disponível em: <<https://www.npmjs.com/package/react-moment?activeTab=readme>> [Acedido aos 29 de Dezembro de 2021].

Npm. 2021. *react-native-animatable* . [online] Disponível em: <<https://www.npmjs.com/package/react-native-animatable>> [Acessado em 29 de dezembro de 2021].

npm. 2021. *react-native-calendars* . [online] Disponível em: <<https://www.npmjs.com/package/react-native-calendars>> [Acessado em 29 de dezembro de 2021].

npm. 2021. *react-native-calendars* . [online] Disponível em: <<https://www.npmjs.com/package/react-native-calendars>> [Acessado em 29 de dezembro de 2021].

npm. 2021. *react-native-dropdown-picker* . [online] Disponível em: <<https://www.npmjs.com/package/react-native-dropdown-picker>> [Acessado em 29 de dezembro de 2021].

npm. 2021. *react-native-date-picker* . [online] Disponível em: <<https://www.npmjs.com/package/react-native-date-picker>> [Acessado em 29 de dezembro de 2021].

npm. 2021. *react-native-datefield* . [online] Disponível em:
<<https://www.npmjs.com/package/react-native-datefield>> [Acessado em 29 de dezembro de 2021].

npm. 2021. *react-native-dotenv* . [online] Disponível em:
<<https://www.npmjs.com/package/react-native-dotenv>> [Acessado em 29 de dezembro de 2021].

npm. 2021. *menu* *suspense* *react-native-element* . [online] Disponível em:
<<https://www.npmjs.com/package/react-native-element-dropdown>> [Acessado em 29 de dezembro de 2021].

W3schools.com. 2021. *Node.js* *MySQL* . [online] Disponível em:
<https://www.w3schools.com/nodejs/nodejs_mysql.asp> [Acessado em 29 de dezembro de 2021].

Reactnavigation.org. 2021. [online] Disponível em:
<<https://reactnavigation.org/docs/getting-started/>> [Acesso em 29 de dezembro de 2021].