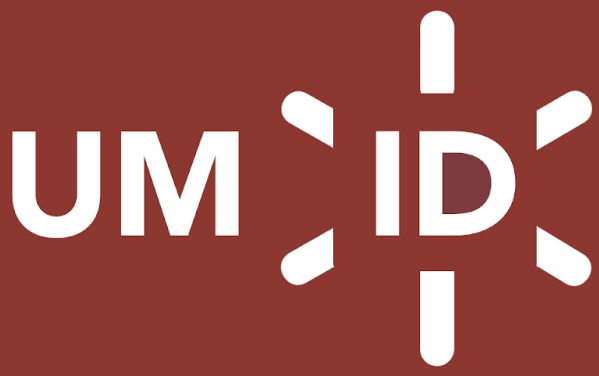


"Todas as pessoas têm direito a participar de forma ativa na economia e na sociedade em que vivem."

Universidade do Minho

Mestrado Integrado em Engenharia Informática

Mestrado em Engenharia Informática

Escola de Engenharia - Departamento de Informática

UM ID
Relatório Técnico

A82145 - Filipa Correia Parente

A81986 - Nuno Afonso Gonçalves Solha Moreira Valente

PG37159 - Mateus da Silva Ferreira

A81403 - Pedro Henrique de Passos Ferreira

A81451 - Alexandre Rzepecki Rodrigues

PG39261 - Leonardo de Jesus Silva

PG41843 - Diogo Rafael Ferraz Duarte

A74572 - Rui Pedro Barbosa Rodrigues

PG39295 - Ricardo Cunha Dias

Janeiro 2021

Sumário executivo

O seguinte documento visa documentar todas as fases de desenvolvimento do projeto UM-ID. Este projeto visa substituir os meios tradicionais de identificação por uma alternativa digital e centralizar serviços institucionais relacionados com a identificação do indivíduo.

Introdução

Nos últimos tempos tem havido uma tendência em direção à digitalização de documentos, isto deve-se a uma maior segurança nestes meios e ao pragmatismo devido a haver uma redução dos elementos físicos que um indivíduo necessita de ter em sua posse.
Este projeto surge como uma alternativa à identificação móvel agregando valor através da centralização de serviços prestados pela instituição ligada ao documento de identificação em causa.

Oportunidades

O desenvolvimento deste projeto surgiu a partir da problemática dos elementos, discentes e docentes, da Universidade do Minho (UM) para identificar-se enquanto participantes desta instituição de ensino e de igual forma aceder aos serviços da mesma com as suas respectivas identificações. O processo de identificação apenas pode ser feito pessoalmente nos diversos serviços institucionais, para além disso diferentes serviços utilizam plataformas distintas entre si. Isto de certa forma é inconveniente para o cliente, pois requer que este interaja com diversas plataformas ou se desloque para o serviço que tenta aceder, tendo de esperar em filas em horários restritos.

É de notar que os incómodos sofridos pela comunidade académica da não se limita à UM, de facto estes incómodos são comuns a muitas instituições não só académicas visto que atualmente, a maioria das instituições não utilizam nenhuma aplicação digital para a identificação dos seus membros, sendo que todo o processo utiliza o método tradicional envolvendo cartões físicos.

A ideia

De forma a resolver estes constrangimentos, surgiu a ideia da criação de uma aplicação digital que terá por base a capacidade de identificar o utilizador na respetiva instituição em que este está inserido, oferecendo também a possibilidade de centralizar outros serviços que esta disponibilize para os seus clientes proporcionando-lhes uma maior satisfação ao utilizar-los, de forma a reduzir burocracias inerentes a eles, simplificando os processos. Esta abordagem, além de ser extremamente conveniente para todas as partes envolvidas, proporciona uma melhor gestão dos dados, otimizando o tempo dos mesmos e agilizando o processo para aceder aos serviços de forma segura.

Requisitos

De forma a podermos definir os requisitos da aplicação tivemos que definir quais seriam os seus utilizadores. Uma vez que o nosso Minimum Viable Product (MVP) será a Universidade do Minho, os utilizadores foram identificados com este cenário em mente mas podem ser facilmente adaptados a outros casos.

- **Administrador:** O administrador do sistema pode ser tanto o gestor de recursos, quanto a equipa de manutenção e aprimoramento.
- **Membros da comunidade académica:** Os membros da comunidade académica utilizam as suas credenciais para se autenticar na aplicação. Isto dá-lhe acesso a identificação i.e. substituto de cartão físico, compra de senhas de cantina online(ao contrário da compra física existente) e reserva/acesso a salas de estudo.
- **Entidade Verificadora:** Utiliza uma versão própria da aplicação que permite verificar e consumir as senhas de cantina.
- **Equipa de Aprimoramento da aplicação:** Para garantir o funcionamento e a implementação de serviços adicionais e novas funcionalidades à aplicação, faz-se necessário que a equipa de aprimoramento da aplicação tenha acesso ao sistema. Recebe feedback dos utilizadores.
- **Equipa de Manutenção:** Lida com eventuais problemas que possam ocorrer na aplicação como falha de servidor, entre outros.

A partir destes utilizadores foram definidos os seguintes requisitos.

Requisitos funcionais

Requisitos do Utilizador

Requisito nº: 1
Descrição: O utilizador pode utilizar a aplicação para se identificar como membro da Instituição de Ensino.
Razão: Para a aplicação substituir o cartão de identificação físico, sendo a funcionalidade fulcral do sistema.
Prioridade: Must

Requisito nº: 2
Descrição: O utilizador ao identificar-se pode optar se o verificador pode guardar a sua informação.
Razão: Para garantir a privacidade do utilizador.
Prioridade: Could

Requisito nº: 3
Descrição: O utilizador pode fazer pagamentos para obter senhas eletrónicas para utilização na cantina da Universidade.
Razão: Funcionalidade básica do sistema.
Prioridade: Must

Requisito nº: 4
Descrição: O utilizador pode consultar o número de senhas que possui.
Razão: Para informar o utilizador de quantas senhas possui. Informação necessária para utilização do serviço da cantina.
Prioridade: Should

Requisito nº: 5

Descrição: O utilizador pode utilizar as senhas para aceder à cantina, desde que seja compatível com o seu tipo de perfil.

Razão: Funcionalidade básica do sistema.

Prioridade: Must

Requisito nº: 6

Descrição: O utilizador pode partilhar senhas com outros utilizadores.

Razão: Funcionalidade básica do sistema.

Prioridade: Could

Requisito nº: 7

Descrição: O utilizador pode utilizar a aplicação para reservar salas de estudo.

Razão: Funcionalidade básica do sistema.

Prioridade: Must

Requisito nº: 8

Descrição: O utilizador pode consultar as suas reservas de salas de estudo.

Razão: Informar o utilizador para que este usufrua do serviço.

Prioridade: Should

Requisito nº: 9

Descrição: O utilizador pode cancelar reservas previamente feitas de salas de estudo.

Razão: Para tornar a sala reservada disponível a outros alunos.

Prioridade: Should

Requisito nº: 10

Descrição: O utilizador deve aceder à sala reservada e identificar-se no local para validar a reserva.

Razão: Para garantir que as reservas são cumpridas pelo utilizador que a reservou.

Prioridade: Must

Requisitos da Entidade Verificadora

Requisito nº: 11

Descrição: A entidade verificadora pode validar a identificação de membros da Universidade.

Razão: Funcionalidade básica do sistema.

Prioridade: Must

Requisito nº: 12

Descrição: A entidade verificadora pode validar senhas da cantina.

Razão: Funcionalidade básica do sistema.

Prioridade: Must

Requisito nº: 13

Descrição: A entidade verificadora pode validar reservas de salas de estudo.

Razão: Funcionalidade básica do sistema.

Prioridade: Must

Requisitos do Administrador (gestor ou equipa de manutenção da aplicação)

Requisito nº: 14

Descrição: Os responsáveis pela manutenção e aprimoramento da aplicação podem adicionar novas funcionalidades à aplicação.

Razão: Para permitir às Instituições de Ensino alterar a aplicação consoante as suas necessidades.

Prioridade: Could

Requisito nº: 15

Descrição: O administrador pode adicionar salas à aplicação.

Razão: Funcionalidade básica do sistema.

Prioridade: Must

Requisito nº: 16

Descrição: O administrador pode remover salas da aplicação.

Razão: Funcionalidade básica do sistema.

Prioridade: Must

Requisito nº: 17

Descrição: O administrador pode alterar o preço das senhas da aplicação.

Razão: Funcionalidade básica do sistema.

Prioridade: Must

Requisito nº: 18

Descrição: O administrador pode adicionar informação relativa à identificação dos utilizadores na aplicação.

Razão: Funcionalidade básica do sistema.

Prioridade: Must

Requisito nº: 19

Descrição: O administrador pode editar informação relativa à identificação dos utilizadores na aplicação.

Razão: Funcionalidade básica do sistema.

Prioridade: Must

Requisito nº: 20

Descrição: O administrador pode remover informação relativa à identificação dos utilizadores na aplicação.

Razão: Funcionalidade básica do sistema.

Prioridade: Must

Requisito nº: 21

Descrição: O administrador pode adicionar senhas a um utilizador.

Razão: Funcionalidade básica do sistema.

Prioridade: Must

Requisito nº: 22

Descrição: O administrador pode remover senhas de um utilizador.

Razão: Funcionalidade básica do sistema.

Prioridade: Must

Requisitos do Sistema

Requisito nº: 23

Descrição: O sistema valida o documento de identificação a partir dos dados recebidos do utilizador.

Razão: É necessário validar o utilizador sempre que utilizar algum serviço disponibilizado pela plataforma de forma a impedir irregularidades.

Prioridade: Must

Requisito nº: 24

Descrição: O sistema oferece pelo menos um modo de pagamento.

Razão: Adiciona-se o modo de pagamento para permitir utilizar a funcionalidade de compra de senhas da cantina.

Prioridade: Must

Requisito nº: 25

Descrição: O sistema atualiza o número de senhas disponíveis.

Razão: A aplicação tem que manter o número de senhas disponíveis sempre atualizado.

Prioridade: Must

Requisito nº: 26

Descrição: O sistema atualiza a disponibilidade das salas.

Razão: A aplicação tem que manter o número de salas disponíveis sempre atualizado.

Prioridade: Must

Requisito nº: 27

Descrição: O sistema mantém um registo de reservas das salas.

Razão: A aplicação deverá manter este registo de forma a evitar irregularidades no acesso às salas.

Prioridade: Must

Requisito nº: 28

Descrição: O sistema dá 15 minutos de tolerância para cada reserva de sala.

Razão: Caso o utilizador não compareça nos 15 minutos seguintes à hora marcada, a sala deverá tornar-se disponível.

Prioridade: Should

Requisito nº: 29

Descrição: O sistema guarda a informação do utilizador referente a sua identificação na instituição.

Razão: É importante que o sistema guarde esta informação de forma a permitir que o utilizador se identifique quando não estiver ligado à rede.

Prioridade: Must

Requisitos não funcionais

Requisitos de aparência

Requisito nº: 30

Descrição: A interface gráfica deve estar em conformidade com os padrões de cores da universidade.

Razão: É conveniente que o tema das aplicações esteja de acordo com o logótipo da instituição, para o utilizador rapidamente reconhecer a aplicação.

Prioridade: Could

Requisito nº: 31

Descrição: O produto deve ser atraente para o público jovem.

Razão: É conveniente que a aplicação do cartão de estudante seja atrativa para o nosso público alvo, maioritariamente estudantes das instituições de ensino.

Prioridade: Could

Requisito nº: 32

Descrição: O produto deve parecer confiável.

Razão: Para garantir a maior aderência de novos utilizadores ao produto é necessário que dê confiança.

Prioridade: Should

Requisitos de desempenho

Requisito nº: 33

Descrição: O tempo máximo de transferência de dados entre 2 dispositivos deve ser de 2 minutos.

Razão: É conveniente que o processo da transferência de dados seja feito de forma rápida para garantir que a utilização do produto seja agradável.

Prioridade: Must

Requisito nº: 34

Descrição: O produto deve operar em modo offline, se a conexão com o servidor for perdida.

Razão: O sistema deve garantir que não haja condicionalidade na comunicação online para funcionalidade do produto.

Prioridade: Must

Requisito nº: 35

Descrição: O produto deve ter uma infra estrutura de alto desempenho (Disponibilidade, Escalabilidade, Adaptabilidade, Confiabilidade).

Razão: O sistema deve garantir a rapidez na resolução de possíveis problemas e capacidade de adaptar-se de acordo com a demanda necessária.

Prioridade: Must

Requisitos operacionais

Requisito nº: 36

Descrição: O produto deve ter uma comunicação interoperável entre as plataformas.

Razão: A aplicação deve ser capaz de efetuar a comunicação na transferência de dados independente do sistema operativo em que está a ser executada.

Prioridade: Must

Requisito nº: 37

Descrição: O produto deve ser capaz de ser instalado por um utilizador não treinado.

Razão: Para uma maior adesão da comunidade académica independentemente do seu grau de familiaridade tecnológica.

Prioridade: Must

Requisitos de manutenção e suporte

Requisito nº: 38

Descrição: O produto deve funcionar nas principais plataformas móveis (Android e iOS).

Razão: É necessário dar suporte para a maioria do público alvo.

Prioridade: Must

Requisito nº: 39

Descrição: O código fonte do produto deve ser devidamente documentado.

Razão: É necessário que o código esteja bem documentado para facilitar a sua interpretação e manutenção.

Prioridade: Should

Requisitos de segurança

Requisito nº: 40

Descrição: Os dados transferidos devem ser autênticos, íntegros e confiáveis.

Razão: É importante que os dados sejam íntegros para garantir os princípios de uma comunicação segura.

Prioridade: Must

Requisito nº: 41

Descrição: O produto deve rejeitar a receção de dados inválidos.

Razão: O sistema deve proporcionar confiabilidade às entidades verificadoras ao acederem os dados dos utilizadores para evitar abusos intencionais.

Prioridade: Must

Requisito nº: 42

Descrição: Os utilizadores devem estar pré-registados no sistema.

Razão: O sistema deve garantir que o utilizador possua um grau de privilégios superior em relação a um utilizador comum.

Prioridade: Could

Requisito nº: 43

Descrição: O produto deve proteger as informações privadas de acordo com as leis de privacidade relevantes e a política de informações da organização.

Razão: O sistema deve garantir a segurança de acordo com as leis da organização.

Prioridade: Must

Requisitos legais

Requisito nº: 44

Descrição: O produto deve se adequar à política de privacidade RGPD.

Razão: O sistema deve estar em conformidade com as normas e leis vigentes.

Prioridade: Must

Estrutura do relatório

O presente relatório está dividido em 7 capítulos.

O segundo e próximo capítulo, estrutura aplicacional, irá expor a arquitetura geral elaborada para o desenvolvimento do projeto, ou seja, será feita uma descrição técnica do funcionamento da aplicação.

O terceiro capítulo, interface, descreve a visão que o utilizador terá da aplicação, será feito também uma descrição do desenvolvimento da interface assim como as ferramentas utilizadas.

O quarto capítulo, lógica de controle, descreve os algoritmos mais relevantes para o desenvolvimento e funcionamento da aplicação.

O quinto capítulo, Instalação e manutenção, é feita uma descrição da instalação da aplicação, também é abordado o trabalho necessário com a manutenção do produto final.

O capítulo final, conclusão, analisa o produto final relatando os requisitos conseguidos e os requisitos em falta assim como apresenta um plano de trabalho para melhorar a aplicação e adaptá-la para outros contextos.

Estrutura aplicacional

A arquitetura a ser seguida foi modelada com base na norma ISO/IEC DIS 18013-5 que padroniza soluções de identificação móvel digital. Assim podemos separar o modo de funcionamento da aplicação das seguintes funcionalidades principais:

1. Associação do Utilizador na Autoridade Emissora;
2. Aquisição de senhas;
3. Reserva de espaços;
4. Verificação dos dados apresentados pelo utilizador;

O diagrama seguinte descreve a arquitetura da nossa aplicação:

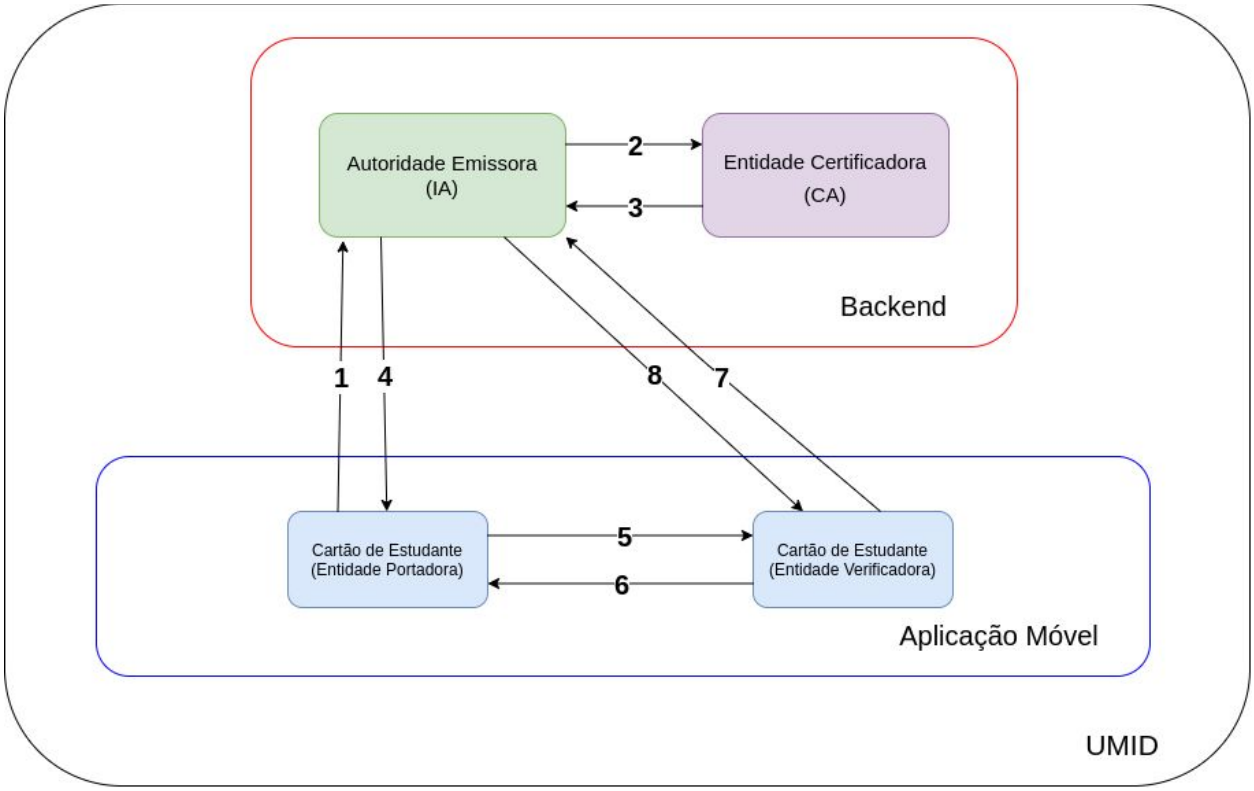


Figura 1: arquitetura da aplicação UMID

Como se pode observar no diagrama, a aplicação encontra-se dividida em quatro entidades, sendo estas:

- **Entidade Portadora** dos dados, isto é, do cartão.

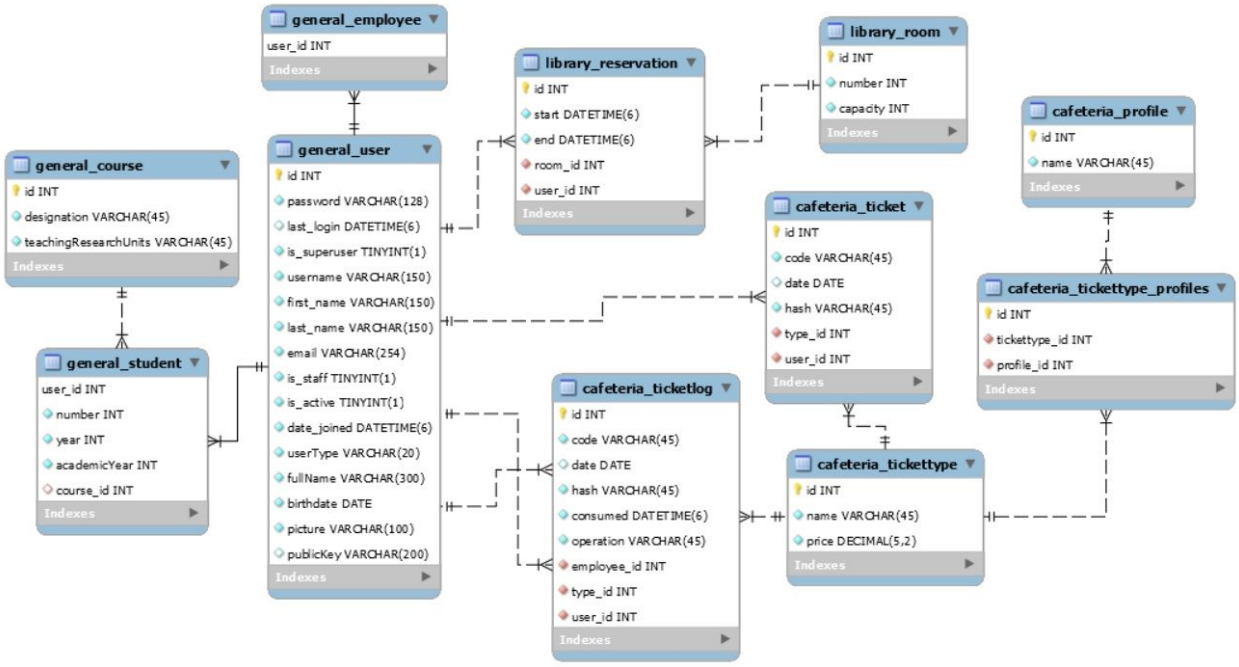
- **Entidade Verificadora** dos dados, responsável pela verificação dos dados provenientes da Entidade Portadora.

- **Autoridade Emissora** dos dados, responsável tanto pela emissão dos dados para a entidade portadora, como pelos mecanismos de segurança necessários para o desenvolvimento de uma aplicação de identificação digital móvel segura.

- **Entidade Certificadora**, responsável pela gestão de certificados do tipo X.509 atribuídos a cada entidade portadora. Estes certificados servem para garantir a autenticidade dos dados no processo de verificação.

Nas próximas cinco seções serão abordadas as implementações das várias camadas da aplicação, desde o processo da construção da base de dados, até ao desenvolvimento das aplicações móveis.

Base de dados



Para o funcionamento da aplicação é necessário guardar informação relativamente aos utilizadores (quer seja entidade portadora ou verificadora), senhas e salas. Para este efeito foi criada uma base de dados relacional capaz de suportar a informação necessária para o funcionamento da aplicação.

A base de dados guarda o nome, imagem e data de nascimento para cada utilizador, atribuindo um identificador único. Dependendo do tipo de utilizador terá informação adicional, alunos terão curso, ano académico, número de matrículas e número de aluno. Verificadores terão um cargo ao qual estão associadas um número de permissões.

Cada senha terá uma entrada na base de dados, esta entrada terá informação sobre o tipo de senha, o seu dono e data de utilização se tiver, cada senha é única e terá um identificador. Informação adicional como os tipos de utilizadores que podem usar cada senha será mantido na base de dados e poderá ser mudado por um administrador.

As salas disponibilizadas pela instituição para os seus utilizadores serão guardadas nesta base de dados com um número identificador e a sua lotação. Uma lista de reservas será mantida que compromete um utilizador com uma sala numa hora específica.

Extração, Transformação e Carregamento (ETL)

Este processo visa povoar inicialmente a base de dados com os dados já existentes que estão a ser utilizados pela instituição em causa. Será único para cada instituição e será desenvolvido únicamente para cada. Irá depender do método utilizado atualmente pela instituição, tais métodos podem ser: bases de dados e/ou ficheiros json ou excel.

O primeiro passo é a extração de todos os dados necessários de todas as fontes usadas. O segundo passo será a modificação destes dados e alguns dos processos que podem ser usados são: modificação do tipo de dados, juntar dados de várias fontes, adição de identificadores. O terceiro passo consiste no carregamento dos dados previamente modificados na base de dados.

Para a criação do MVP foi simulada uma possível base de dados de uma instituição de ensino e desenvolvido um processo de ETL para a instituição simulada. A informação dos utilizadores foi extraída de uma base de dados e transformada para se adaptar à base de dados da aplicação. Alguns dos processos usados na transformação foram:

- Concatenação do primeiro nome com o apelido;
- Junção de várias tabelas para a povoação dos utilizadores.
- Processamento de datas.

Para além deste processo certos objetos como tipo de senhas utilizadas e as salas foram inseridas manualmente, devido à falta de um formato digital dos mesmos.

Entidade Certificadora (CA)

Como já foi referido anteriormente, a CA (certificate authority) ou entidade certificadora é responsável pela gestão dos certificados digitais utilizando uma infraestrutura de Chave Pública (PKI).

Segundo o RFC-3647, uma infraestrutura de Chave Pública (PKI) é um conjunto de regras, políticas, hardware, software e procedimentos necessários para criar, gerir e armazenar certificados digitais. Neste projeto, o objetivo da PKI consiste em garantir a confidencialidade, privacidade e autenticidade das informações transmitidas entre as entidades do sistema. Esta segurança é garantida através da utilização de criptografia de chave pública.

A implementação da PKI na CA foi baseada nas especificações contidas no standard internacional ISO/IEC DIS 18013-5 que padroniza soluções de identificação móvel digital. Posto isto, foi utilizado a aplicação CFSSL que consiste num toolkit open source, desenvolvido pela CloudFlare para a criação e gestão de PKI. Para armazenar os certificados gerados pela CA foi criado uma base de dados utilizando o postgresql. De forma a verificar se os certificados encontram-se válidos ou revogados foi criado, através do CFSSL, um servidor OCSP que se encontra acessível através de um endpoint disponibilizado pelo backend.

Para facilitar a implementação com o backend, a base de dados e o cfssl foram configurados em diferentes containers Docker e integrados utilizando o docker-compose. Por questões de segurança, os containers Docker só podem ser acedidos através do backend.

Outra funcionalidade implementada na instância da CA consiste num script que realiza assinaturas digitais e que pode ser acedido apenas via SSH por parte do backend.

Backend API

A API foi desenvolvida utilizando o Django Rest Framework, que facilita a interação com a base de dados e possibilita a utilização de objetos python, sendo criados modelos que são manipulados pelo Django para servir pedidos.
Desta forma, foram programados endpoints para os seguintes tipos de pedidos:

- General - Para pedidos relativos a utilizadores e suas informações
- Library - Para pedidos relativos a salas de estudo e reservas destas
- Cafeteria - Para pedidos relativos a senhas da cantina

General

Foram criados dois tipos de utilizador, estudantes e funcionários. Os dois tipos têm atributos diferentes, mas compartilham a mesma classe do utilizador geral. O estudante contém atributos adicionais, como ano e curso. O curso também é uma classe, que contém atributos como a sua designação.

Para listar a informação sobre os utilizadores, foram criados três endpoints, um para o utilizador geral, outro para os estudantes e outro para os funcionários. Para utilizadores normais, só é possível obter informação sobre o seu perfil.

Caso seja necessário obter certos atributos de um utilizador, foi criado um endpoint que recebe um token e, depois de validado, são retornados todos os atributos do utilizador que o token contém.

Também foi criado um endpoint para obter todas as informações sobre um utilizador. Este endpoint recebe um CSR que é usado para obter o certificado do utilizador, MSO e guardar a sua chave pública. Após isso, são enviados todos os dados sobre o utilizador, incluindo as suas reservas de sala e senhas da cantina.
Finalmente, foi criado um endpoint para receber a CA.

Library

Para a biblioteca, foram criadas duas classes, a das salas e a das reservas. As salas contêm o seu número e capacidade e as reservas contêm o tempo de começo e fim, a sala respetiva e o utilizador que reservou.

Foram criados endpoints para obter as informações sobre as salas e sobre as reservas de um utilizador.

Também foi necessário implementar um endpoint que enviasse as salas que se encontram atualmente livres e outro que enviasse os intervalos de tempo em que certa sala está livre. Para isso, foi implementada uma função que obtém todas as reservas efetuadas a uma sala nas próximas 24 horas e outra que recebe esse resultado e subtrai todos os intervalos de tempo ao longo do dia.

Cafeteria

Esta parte lida com as senhas de cantina. Desde a sua criação, até ao consumo. Permite ao utilizador ver as suas senhas como também as senhas que foram gastas.

Foram criados endpoints básicos que permitem o utilizador aceder aos seus dados, ver o tipo de senhas que se pode comprar. Os outros dois endpoints são um para consumir a senha, da parte do funcionário e colocá-la no registo, a outra adicionar senhas ao utilizador.

Lógica de Controlo

A lógica de controlo da aplicação foi feita de modo a modularizar as funcionalidades, permitindo assim termos a mesma aplicação a ser utilizada pelos estudantes e funcionários. Estes perfis possuem funcionalidades distintas, a aplicação tem a capacidade de ajustar-se a elas. Para utilizar a aplicação UMId é necessário a criação de pin de acesso por parte do utilizador, em seguida é realizado o processo de associação da identificação do elemento da comunidade académica a aplicação.

Este processo é uma simulação do mecanismo, o qual a Universidade do Minho (UM) faz uso para os seus participantes acederem aos serviços já existentes da universidade, para isso o utilizador faz uso do número mecanográfico e palavra passe.

A gestão, relacionada às senhas da cantina que o utilizador possui, é feita de acordo com o tipo de senha. A aplicação coloca à disposição do utilizador três tipos de senhas, dois destes tipos são uma representação das senhas físicas existentes. A terceira opção é a senha promocional do dia, senha com data definida que fará a refeição, a qual tem como proposta beneficiar o perfil de utilizador que a comprar.

A gestão da aplicação garante que ao utilizador querer utilizar uma senha e tiver a senha promocional para aquele dia, mesmo que tenha outro tipo de senha a aplicação só permitirá consumir a senha promocional. Desta forma, evitamos possível equívoco do utente ao utilizar uma senha para refeição. Ainda sobre as senhas promocionais do dia, mesmo que o utente venha a esquecer-se de utilizar naquele determinado dia a senha fica armazenada e é informado que a senha está vencida. Esta abordagem tem como objetivo o pagamento de uma multa caso ele não a use no dia devido para resgatar para usar em outro determinado dia.

A compra de senhas é realizada de forma online na aplicação. O utilizador pode escolher a quantidade de senhas que quer seja a senha completa e/ou prato simples. Para as senhas promocionais do dia, o utente escolhe os dias que quer e o tipo de

senha, completa ou simples. A aplicação controla a quantidade de senhas do tipo promocional para aquele dia, pois um utente só pode possuir duas senhas promocionais para um determinado dia, caso já possua uma senha promocional para esse dia só poderá comprar mais uma senha promocional para esse dia. Uma vez realizado a compra das senhas é atualizado no backend e no guardado na aplicação, na zona segura já mencionada .

Ao consultar as senhas que possui o utente é informado o número de senhas que possui do tipo senha completa e senha prato simples de igual forma quantas senhas possui do tipo senha promocional do dia e também qual a data da próxima senha a ser utilizada deste tipo.

Para realizar uma reserva o utente é necessário conexão com a internet, pois é feito uma busca das disponibilidade de cada sala o utilizador insere o intervalo de tempo o qual deseja fazer pesquisa de disponibilidade para uma determinada sala. Uma vez estipulado esse intervalo temporal é apresentado ao utente slots de 30 minutos para aquela sala se houver disponibilidade. O utente não pode escolher slots intercalados se escolher mais de um slot para uma reserva, estes devem ser consecutivos.

Na aplicação pode consultar qual a sua próxima reserva, bem como realizar pela aplicação o check-in e check-out da mesma. Aplicação pode-se alterar o código de acesso como também acedê-la como o recurso biométrico do telemóvel.

Segurança Aplicacional

Segurança dos dados (relação entre Entidade portadora e Autoridade Emissora)

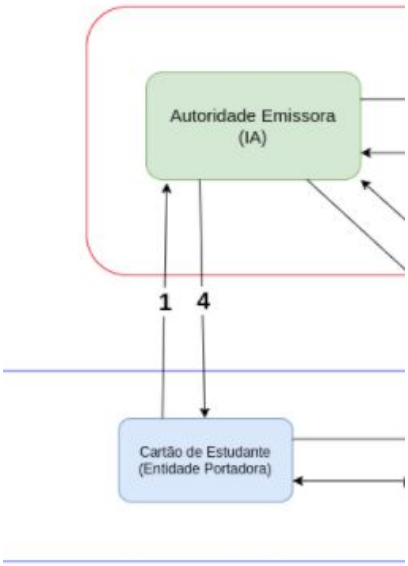


Figura 2: Esquema explicativo da relação entre a Entidade Portadora e a Autoridade Emissora

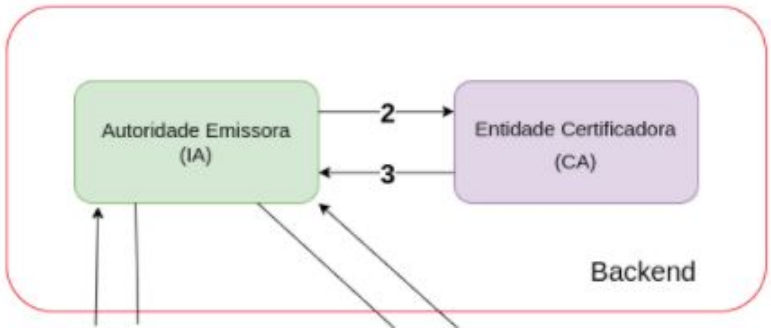


Figura 3: Esquema explicativo da relação entre a Autoridade Emissora e a Entidade Certificadora + Public Key Infrastructure

De forma a garantir uma comunicação segura entre a Entidade Portadora e a Autoridade Emissora, o processo de associação da identificação com o dispositivo é realizado através dos seguintes passos:

1. Envio de um pedido proveniente da aplicação através de um Certificate Signing Request (CSR);
2. Envio do CSR à Entidade Certificadora (CA). Este CSR será posteriormente utilizado pela CA para gerar o certificado associado ao utilizador que estiver a se associar. Este certificado servirá para garantir a autenticidade e a integridade de todos os dados provenientes da Entidade

Portadora. É nesta fase que a estrutura com os dados do utilizador é enviada para a Public Key Infrastructure (PKI) onde será gerada um Mobile Security Object (MSO). O MSO consiste na geração das assinaturas em cada atributo relacionado a identificação do utilizado, essa estrutura é devolvida pelo Backend para aplicação. Esta estrutura é fundamental para garantir a integridade e autenticidade dos dados no modo de comunicação offline (sem recurso a pedidos a Autoridade Emissora);

3. Envio do certificado gerado e do MSO à Autoridade Emissora dos dados;
4. Envio do certificado gerado, do MSO e dos dados associados à entidade portadora.
5. Verificação da autenticidade dos dados recebidos com recurso à validação da cadeia de certificados (ver Nota);

Nota: Tanto a Entidade Verificadora como a Entidade Portadora possuem localmente o certificado raiz/root da Entidade Certificadora. Este certificado é essencial para garantir a autenticidade da Autoridade Emissora.

Dado a natureza dos dados referentes à identificação do utilizador, de nível crítico, estes estão a ser armazenados em um local seguro nos respectivos sistemas operacionais dos telemóveis. Desta forma, foi utilizado um plugin do Ionic, Secure Storage, para esse determinado fim. Este plugin faz uso no iOS do recurso keychain, um container criptografado que armazena com segurança em pequenos pedaços de dados em nome de aplicativos e serviços seguros. Em relação ao Android utiliza-se uma abordagem similar para o sistema operacional iOS.

Segurança dos dados (relação entre Entidade Portadora e Entidade Verificadora)

O mecanismo de transferência dos dados recorre à tecnologia Bluetooth Low Energy. De forma a compreender melhor o processo de transferência dos dados na Figura 4 encontra-se o diagrama explicativo:

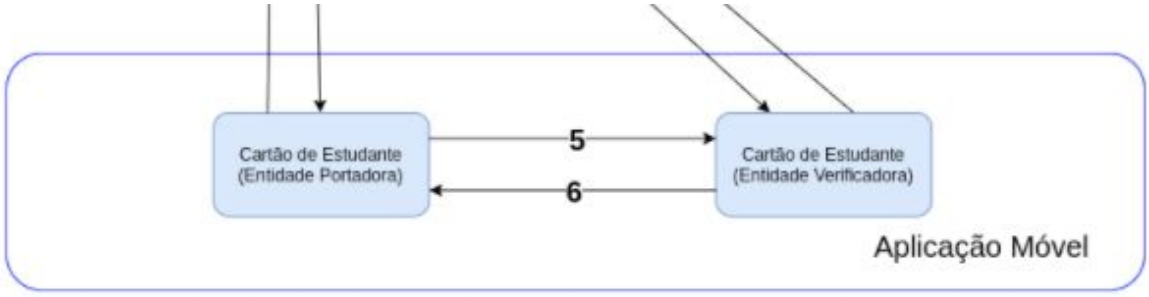


Figura 4: Esquema explicativo da relação entre a Entidade Portadora e a Entidade Verificadora

A transferência procede-se da seguinte forma:

6. A Entidade Portadora apresenta um QR-Code com os dados necessários para o estabelecimento da sessão segura de transferência;
7. A Entidade Verificadora procede à sua leitura, estabelecendo uma sessão segura com a Entidade Portadora. Posteriormente envia um pedido à Entidade Portadora com os dados que pretende, dependendo do tipo de pedido;
8. A Entidade Portadora recebe esse pedido e, mediante o consentimento, do utilizador final, envia ou não os dados pretendidos para a Entidade Verificadora;
9. A Entidade Verificadora recebe os dados provenientes da Entidade Portadora e procede à validação dos mesmos.

A comunicação entre as entidades Portadora e Verificadora é feita através de mecanismos criptográficos que garantem a autenticidade, integridade e validade dos dados. Note-se que para tipos diferentes de comunicação os mecanismos de validação dos dados são distintos. Tendo isso em conta, os mecanismos são os seguintes:

Comunicação online:

1. Geração de um token no formato JWT na Entidade Portadora cuja estrutura usada para representar o payload se encontra em **a.** para obter a identificação e **b.** para validar/consumir uma senha:
    a. token: {
        "username": 'pg32333',
        "namespaces": ['username','picture', 'birthdate']
    }
    b. token: {
        "username": 'pg32333',
        "type": "senha completa",
        "date": true,
        "debugdate": "2021-02-02T01:28:31.164501Z"
    } // para as senhas promocionais
    token: {
        "username": 'pg32333',
        "type": "senha prato simples",
        "date": false
    } // para as senhas não promocionais

Comunicação offline:

1. Criação de uma estrutura baseada na norma ISO/IEC DIS 18013-5. Esta estrutura contém o Mobile Security Object (MSO) e os dados pedidos pela Entidade Verificadora.

Para a realização da transferência bluetooth, a cada nova transferência/sessão os seguintes mecanismos de segurança são efetuados, tanto na comunicação offline, como online:

1. Geração de chaves efémeras usando o algoritmo Elliptic Curve Diffie Hellman (ECDH). Estas chaves são usadas para derivar em chaves de sessão que vão encriptar todos os dados que serão transmitidos entre as duas entidades;
2. Encriptação dos dados a serem transmitidos através da chave privada de sessão;
3. Desencriptação dos dados recebidos através da chave pública de sessão.

De referir que o pedido da entidade emissora e a resposta da entidade portadora, baseiam-se na norma ISO/IEC DIS 18013-5. Nesta estrutura, a assinatura tanto da Entidade Emissora como da Entidade Verificadora e da Entidade Portadora encontra-se incluída numa mensagem do tipo CBOR Object Signing and Encryption (COSE). Nessa mensagem também se encontra em formato Concise Binary Object Representation (CBOR) os dados dos atributos pedidos (caso da Entidade Verificadora), bem como os atributos pedidos pela Entidade Verificadora (caso da Entidade Portadora).

Segurança dos dados (relação entre Entidade Verificadora e a Autoridade Emissora)

A Entidade Verificadora possui um papel fundamental na intermediação entre a Entidade Portadora e a Autoridade Emissora, no caso da comunicação online. Neste tipo de comunicação esta entidade é responsável por:

- Enviar o token proveniente da Entidade Portadora para a Autoridade Emissora;
- Fazer o tratamento da resposta proveniente da Autoridade Emissora. No caso específico da UM Id, os dados pedidos são enviados pela Autoridade Emissora no formato JWT. O payload do JWT varia consoante os atributos que são pedidos. Dentro do payload também se encontra o certificado de assinatura do token, que garante a autenticação da Autoridade Emissora e a integridade dos dados recebidos.

Já na comunicação offline esta é a responsável por garantir os mecanismos de segurança dos dados recebidos. Neste tipo de comunicação esta entidade é responsável por:

- Fazer o tratamento da resposta proveniente da Entidade Portadora. Este tratamento inclui mecanismos de segurança que passam por:
    1. Validação de cada elemento recebido através das validação das assinaturas presentes no MSO;
    2. Validação da cadeia de certificados, constituída pelo certificado raiz da CA e pelo certificado recebido da Entidade Portadora;

Interface

O frontend deste projeto foi planeado com uso da ferramenta Figma, a qual possui recursos visuais para construir os mockups das telas e seus respectivos fluxos de usabilidade e funcionalidades das aplicações. A Figura abaixo mostra o esquema criado para as interfaces das aplicações usando o Figma.

Consequentemente, uma vez de posse destes mockups foi desenvolvida a interface gráfica das aplicações, para isso utilizou-se do recurso do ambiente de desenvolvimento adotado neste projeto nomeadamente framework Ionic. Está sendo uma framework de desenvolvimento cross-platform, deste modo garantimos a interoperabilidade em diferentes sistemas operacionais móveis.

O Ionic é composto por blocos de construção de alto nível chamados componentes, os quais permitem construir rapidamente a User Interface (UI) da sua aplicação. Desta forma, o desenvolvimento do frontend para este projeto foi baseado em componentes

visuais com o propósito de obter uma maior generalização da interface gráfica das aplicações. Desta forma, para adaptar em outros contexto de identificação móvel conseguimos responder prontamente a essa necessidade em relação ao frontend.

O frontend foi desenvolvido considerando o uso do modo dark e light dos telemóveis adaptando-se de acordo com o modo ao qual o utilizador está a usar no momento. Isso proporciona ao utilizador das aplicações uma maior melhor experiência visual ao utilizá-las. Segue abaixo a interface gráfica da tela de notificações em modo dark e light:

Modo Dark

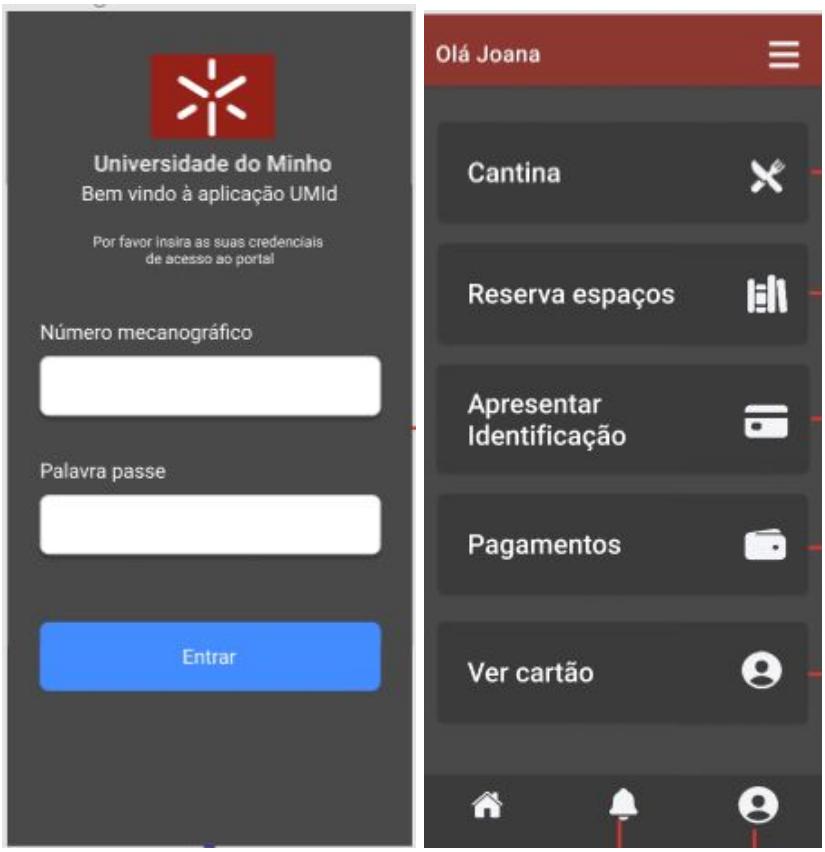
Modo Light

Views

Inicialmente o aluno começa com no ecrã de login, onde terá de inserir as credenciais para entrar na aplicação.

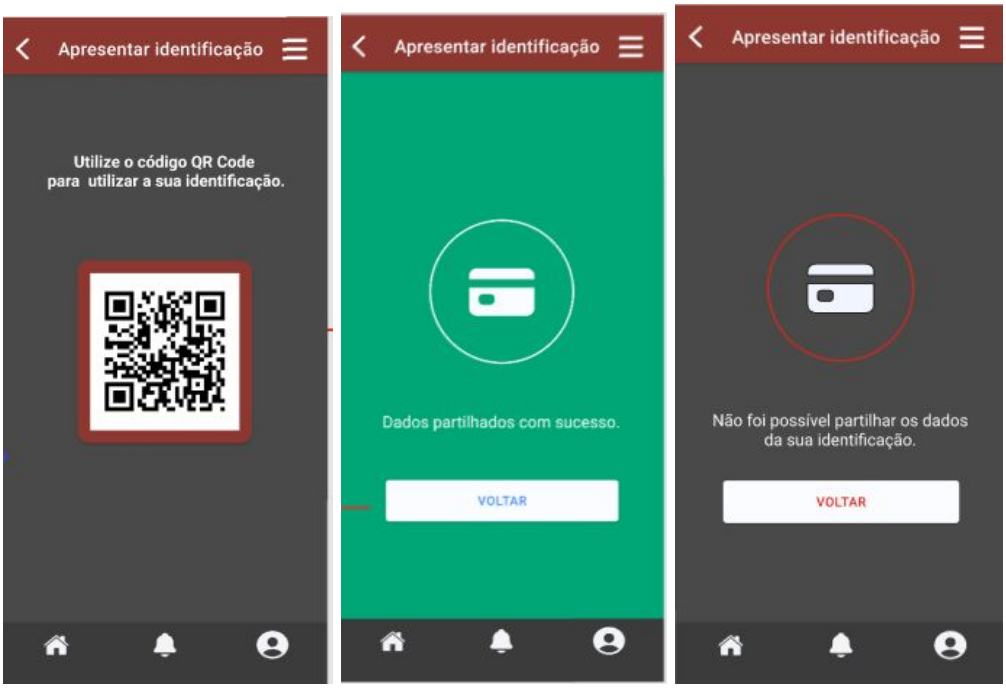
Após inserir as credenciais será redirecionado para a tela inicial. Na tela inicial o utilizador terá as seguintes escolhas:

- Apresentar a sua Identificação;
- Gerir as suas senhas de cantina;
- Gerir as suas reservas de espaços;



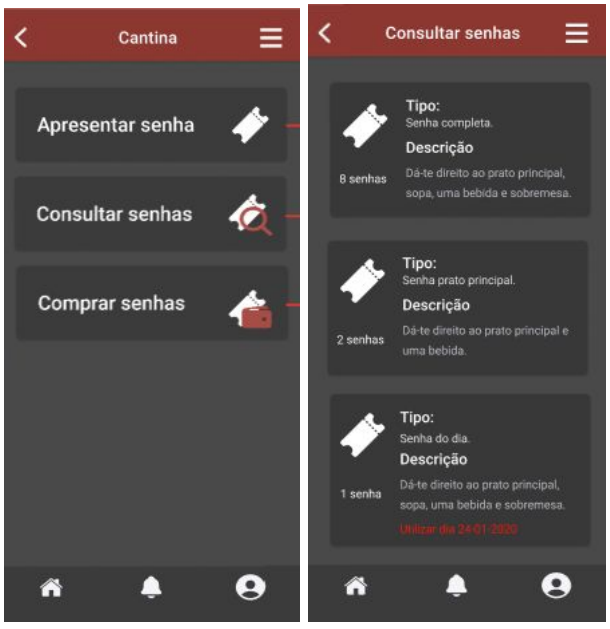
Utilizando a opção de identificação o ecrã do utilizador passará a ser composto por um código QR, a ser lido por uma entidade verificador.

Dependendo do sucesso da operação a aplicação irá mostrar duas telas diferentes

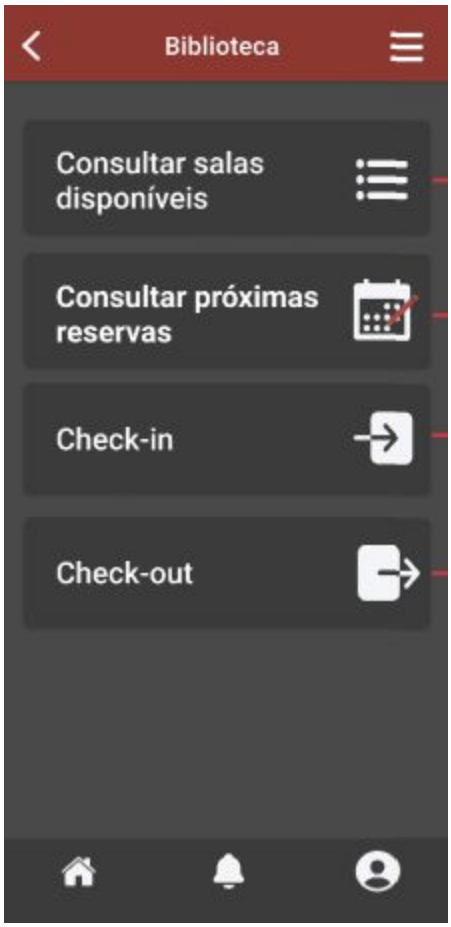


A opção de cantina no menu principal irá redirecionar a um submenu de cantina onde é feita toda a gestão de senhas, tendo opções de comprar, consultar e utilizar.

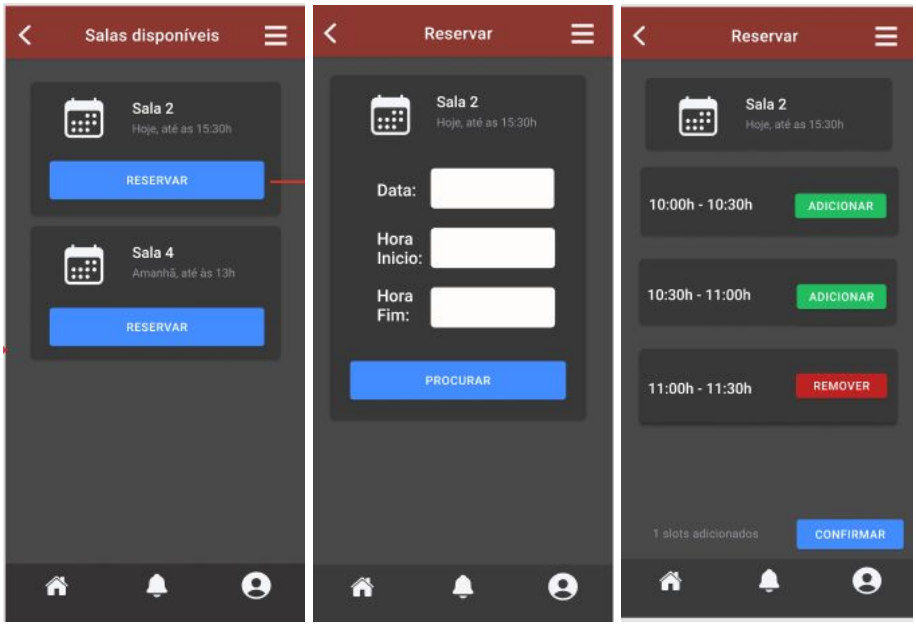
Qualquer uma das opções escolhidas neste menu levará a uma lista de senhas. A compra de senhas será feita através de mbway ou paypal. Para utilizar a senha basta escolher o tipo de senha e será mostrado um código QR com um processo igual ao de identificação.



A terceira opção do menu é a reserva de espaços. Escolher esta opção encaminha para o submenu da biblioteca, dando ao utilizador a opção de: consultar salas disponíveis e reservá-las; ver as reservas efetuadas com opção de cancelar; fazer check-in e check-out numa reserva.



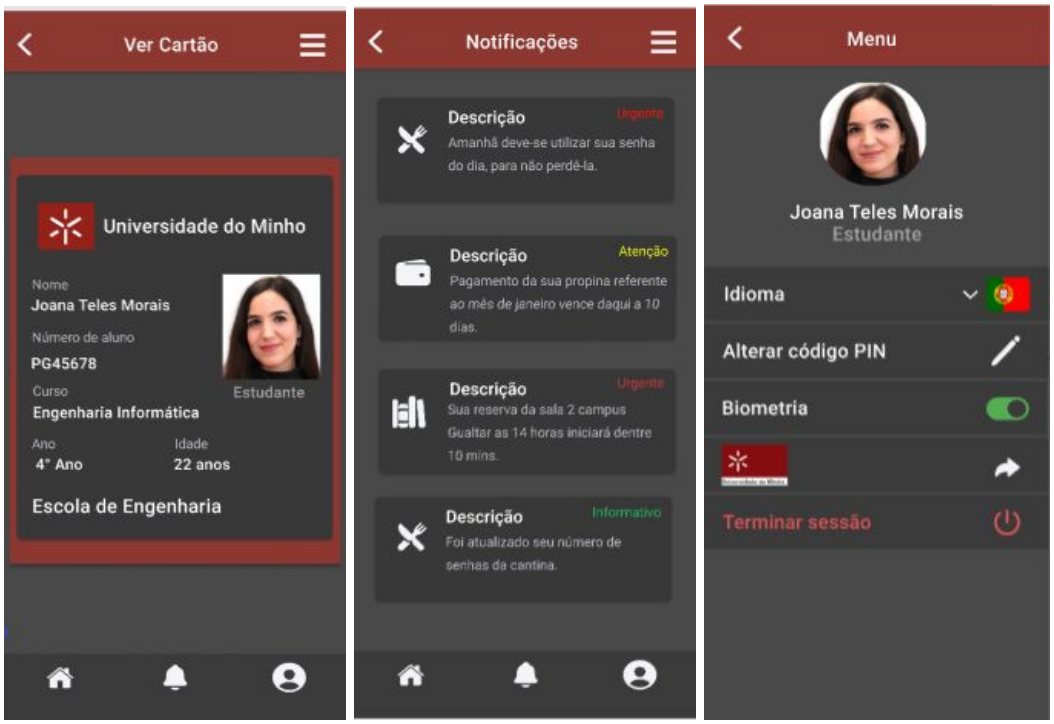
A consulta de salas irá listar todas as salas, após escolher uma o utilizador terá de escolher um horário que pretende. Após estas escolhas será apresentada uma listagem de tempos em que a sala está disponível e será dada a opção de reservar.



A opção de consultar salas irá mostrar a lista de reservas do utilizador, este poderá cancelar ou caso seja uma hora válida efetuar check-in. O processo de check in é semelhante ao de identificação produzindo um código QR a ser validado por uma autoridade verificadora. O check-out poderá ser realizado a qualquer hora depois de check-in.

A qualquer momento de utilização o utilizador poderá:

- Verificar o seu perfil. Clicando na imagem no canto inferior direito;
- Ver as suas notificações (informações importantes ou promoções). Clicando no ícone de notificações localizado no centro da barra inferior;
- Mudar as suas definições (código pin e idioma). Pressionando a opção no canto superior direito;



Instalação e manutenção

A instalação das aplicações será feita, primeiramente, em Android utilizando-se do Android Package (apk). Um executável para instalar em dispositivos com sistema operacional Android. Por se tratar neste momento de uma simulação de uma aplicação para Universidade do Minho, por questões legais optou-se por não registar a aplicação nas lojas de aplicativos dos principais sistemas operacionais móveis do mercado, nomeadamente Android e iOS.

Sobre a manutenção dado a natureza da estrutura da aplicação ser modularizada, este produto possui uma boa manutenibilidade e adaptabilidade para outros clientes.

Conclusão

Isto posto, consideramos que o produto obteve êxito na realização dos objetivos planeados. Os quais resultaram num produto viável e factível de inserção no mercado académico. De igual forma, o produto construído torna possível uma escalabilidade, adaptabilidade do mesmo para diferentes universidades como também para outros nichos de negócio os quais fazem uso de identificação e tem serviços restritos a essas. Para além disso, a aplicação tem em seu cerne o componente de segurança desde da sua infraestrutura como o BackEnd a aplicação em si. Bem como a comunicação entre esses dois elementos do produto. A usabilidade das aplicações também se destaca, pois ela pode atuar em dois modos visuais, o light mode e dark mode, de acordo com o modo o utilizador está a usar no seu telemóvel.