**Instituto Politécnico de Beja**

**Escola Superior de Tecnologia e de Gestão**

**CTeSP de Curso de Tecnologias Web e Dispositivos Móveis**

**Relatório de Programação de Aplicação para Servidor (PAS)**

**Gestão de Senhas de Refeitório para o IPBEJA**

**Jorick Meijde nº 21138**

**Ricardo Coelho nº 21140**

**Beja**

**2021/2022**

**Instituto Politécnico de Beja**

**Escola Superior de Tecnologia e Gestão**

**CTeSP de Curso de Tecnologias Web e Dispositivos Móveis**

**Relatório de Programação de Aplicação para Servidor (PAS)**

**Gestão de Senhas para Refeitório do IPBEJA**

**Projeto de PAS apresentado na Escola Superior de Tecnologia e de Gestão**

**Elaborado por:**

**Jorick Van Der Meijde**

**Ricardo Emanuel Neto Coelho**

**Orientadores:**

**Eng. Luís André Rosário**

**Lic. Carlos Sancho**

**Beja**

**2020/2021**

**Resumo**

Neste relatório, é demonstrada, a forma de como elaboramos o projeto de PAS, as dificuldades que surgiram, o que foi feito para contornar o mesmo, alterações a planificação inicial, e muito mais.

O nosso programa consiste numa aplicação de Gestão de Senhas para o Refeitório do IPBEJA, realizado em Java, com suporte ao Android Studio, com a Ligação de uma API (Interface de Programação de Aplicações) criada através do Laravel, e com Blades (forma de apresentar certas páginas). Tanto a aplicação Android como o BackOffice foram pensados em ser o mais parecido quanto possível da realidade, com o propósito de em termos práticos, serem utilizados pelo Instituto Politécnico.

Mesmo com as dificuldades que foram surgindo, ambos sentimos que aprendemos bastante com a elaboração deste projeto. Um dos objetivos com a elaboração de tudo isto, é qualquer pessoa que pegue neste relatório, conseguir perceber o que foi feito, e se houver necessidade adaptá-lo, o programa.

**Palavras-Chave:** Senhas Refeitório; Laravel; Java; Android Studio; API; Realidade; Consumidor; Dificuldades;

**Índice**

[Introdução 8](#_Toc94622462)

[Tema 9](#_Toc94622463)

[Análise do Problema 9](#_Toc94622464)

[Requisitos 10](#_Toc94622465)

[Protótipo 11](#_Toc94622466)

[Base de Dados 16](#_Toc94622467)

[Aplicação Final 18](#_Toc94622468)

[API 26](#_Toc94622469)

[BackOffice (CRUD) 27](#_Toc94622470)

[Futuras Implementações 31](#_Toc94622471)

[Conclusão 32](#_Toc94622472)

**Índice de Figuras**

[Figura 1 - Login 11](#_Toc94622506)

[Figura 2 - Menu Inicial 12](#_Toc94622507)

[Figura 3 - Menu do Perfil 13](#_Toc94622508)

[Figura 4 – RecyclerView com os dias da Semana 14](#_Toc94622509)

[Figura 5 – Detalhes da Refeição 15](#_Toc94622510)

[Figura 6 - Tabela para as Logins 16](#_Toc94622511)

[Figura 7 - Tabela das Refeições 16](#_Toc94622512)

[Figura 8 - Tabela dos Dias da Semana 17](#_Toc94622513)

[Figura 9 - Tabela das Compras 17](#_Toc94622514)

[Figura 10 - Login da Aplicação 18](#_Toc94622515)

[Figura 11 - Menu 19](#_Toc94622516)

[Figura 12 - Perfil 20](#_Toc94622517)

[Figura 13 – Lista de Dias da Semana 21](#_Toc94622518)

[Figura 14 - Lista das Refeições Disponíveis 22](#_Toc94622519)

[Figura 15 - Detalhes de uma Refeição 23](#_Toc94622520)

[Figura 16 – Confirmar Compra 24](#_Toc94622521)

[Figura 17 – Lista das Senhas (com QR code) 25](#_Toc94622522)

[Figura 18 - Endpoints da nossa API 26](#_Toc94622523)

[Figura 19 - Página para Iniciar Sessão 27](#_Toc94622524)

[Figura 20 - Registar um Novo Utilizador 27](#_Toc94622525)

[Figura 21 – Index 28](#_Toc94622526)

[Figura 22 - Apresentação das tabelas dos Logins 28](#_Toc94622527)

[Figura 23 - Mostrar um Login Específico 29](#_Toc94622528)

[Figura 24 - Adicionar um novo dia de semana 29](#_Toc94622529)

[Figura 25 - Atualizar as Refeições 30](#_Toc94622530)

# Introdução

Este relatório pertence ao projeto desenvolvido nas aulas de Programação de Aplicação para Servidor (PAS).

Será enumerado, uma análise do tema escolhido, os objetivos, indicar os contratempos que possam ter surgido, as suas respetivas soluções, todos os processos realizados ao longo do mesmo, o funcionamento do programa, a sua utilização, e possíveis melhorias.

O projeto foi elaborado no 1º semestre com as matérias nas cadeiras de Programação para Dispositivos Móveis 2 e de Programação de Aplicação para Servidor.

# Tema

O nosso grupo, decidiu escolher este tema, pois pareceu-nos bastante apelativo, visto ser algo existente na vida real, também possibilitou a existência de algo mais possível com a realidade envolvente. Já que ambos estamos inseridos na comunidade académica do Instituto Politécnico, tentamos aproveitar ao máximo a motivação para este tema, já que em termos práticos, a aplicação poderia ter mesmo um uso destinado, e não seria apenas mais um projeto para fins académicos. O nosso tema, é um tema que envolve um peso muito grande na parte dos consumidores, já que estes são os utilizadores da nossa aplicação. O resto dos utilizadores, poderá fazer as modificações que achar necessárias através do BackOffice. De seguida, será explicado melhor cada nível de utilizador existente, de uma forma mais detalhada.

# Análise do Problema

O Refeitório do IPBEJA tem várias refeições para cada dia, e por sua vez, cada refeição tem as respetivas compras. Ou seja, dia 30/01/2022 tem 3 refeições, e uma refeição tem 5 compras registadas.

Várias das coisas que podem ser feitas nesta aplicação é que, os utilizadores conseguem ver quais são os dias disponíveis para comprar senhas. comprar as senhas que pretenderem, e observarem todas as senhas que contenham.

Já que consideramos que o sistema que o IPBEJA usa esta desatualizado, uma máquina para o politécnico inteiro, decidimos que é muito mais fácil os estudantes conseguirem comprar as senhas através da aplicação no telemóvel.

# Requisitos

Falando com os docentes da cadeira, falou-se sobre os requisitos do projeto, em conjunto com os mesmos definiu-se os seguintes requisitos:

* Logins;
* Criar uma “Home Page”;
* Ver os dias da semana;
* Marcar as Senhas;
* Ver os detalhes do menu;
* Ver as senhas possuídas;
* Implementar uma API;
* Criar uma aplicação Web para o BackOffice.

Basicamente, estes consistem na aplicação ter um sistema de login, uma página inicial, algo que nos permita ver os dias da semana disponíveis, comprar as senhas que o utilizador pretender, ver os detalhes dos mesmos, ver as senhas compradas e implementar uma API funcional.

# Protótipo

Nesta parte, é demonstrada o protótipo da aplicação estabelecida em conjunto com os docentes no início do semestre. Isto é, o que estava estabelecido de modo a responder aos requisitos apresentados. Conforme o desenvolvimento do projeto, procedeu-se a algumas mudanças, os quais serão demonstrados mais tarde, na versão final.



Figura 1 - Login

Nesta parte, conseguimos observar um ecrã, onde o utilizador poderá introduzir o seu email, e a sua palavra-passe. Caso o login seja feito com sucesso, o utilizador é levado para a próxima página, a página principal. Caso o utilizador coloque os dados errados, o utilizador é informado do mesmo, e permanece no mesmo ecrã. Escolheu-se fazer o design assim, para ser minimalista, e não confundir os utilizadores desnecessariamente.



Figura 2 - Menu Inicial

Tal como referido anteriormente, após efetuar o login, conseguimos observar 2 botões, um para ir para o submenu dos dias da semana, um para o ecrã que irá mostrar as senhas que o utilizador possui e carregando na fotografia do utilizador no canto superior direito, será levado para um ecrã de onde poderemos terminar sessão.

Uma imagem com texto

Descrição gerada automaticamente

Figura 3 - Menu do Perfil

Nesta imagem está representado o ecrã do perfil do utilizador. Basicamente no início estava pensado, que o utilizador aqui poderia terminar a sua sessão, editar os meios de pagamento e observar o seu histórico das senhas. Mais uma vez, pretende se um design minimalista, mas eficiente.



Figura 4 – RecyclerView com os dias da Semana

Aqui é apresentado um ecrã onde serão listados os dias da semana disponíveis. Através deste ecrã, o utilizador poderá escolher um dia para o qual pretenda adquirir as suas senhas. Para de seguida clicar num dia de semana, e poder comprar, ou voltar para o início caso o utilizador assim o pretenda.

Uma imagem com texto

Descrição gerada automaticamente

Figura 5 – Detalhes da Refeição

Nesta imagem, está representado os detalhes da refeição que o utilizador escolheu do ecrã anterior. O ecrã demonstra uma foto do prato principal, a sopa, e a sobremesa associado a mesma. De seguida, o utilizador poderá escolher entre comprar a refeição ou não, caso compre, o mesmo irá proceder a compra do mesmo, e adicionar a refeição ao utilizador.

# Base de Dados

Depois do protótipo elaborado, pensamos sobre a estrutura da base de dados. A nossa base de dados foi baseada no que era necessário para a nossa API e para a nossa aplicação do Android Studio.



Figura 6 - Tabela para as Logins

Na figura acima representado, conseguimos observar o esquema que a nossa tabela das Logins representa, no nosso caso era para ser a tabela dos Utilizadores/Users, mas neste caso já que numa próxima etapa o Laravel gera uma tabela dos Users para poder gerir o BackOffice, tivemos que mudar o nome da nossa tabela, de modo a evitar possíveis conflitos.

Dá para ver que neste caso, guardamos o código de utilizador, o nome, o último nome, o email, a palavra-passe, a url para poder inserir uma foto, e quando é que o registo foi criado/atualizado pela última vez.



Figura 7 - Tabela das Refeições

Na figura acima representado, está estruturado a tabela das refeições, isto permite nos guardar as refeições que depois serão colocadas a venda. Certas informações serão públicas ao utilizador, outras nem por isso, que servirão para a gestão interna.

Neste caso, a tabela guarda, o código da refeição, o código do dia da semana, uma ligação de uma para muitos, é por esse motivo que consta aqui a chave estrangeira.

A designação dada ao prato principal, a sopa, e a sobremesa, e uma hiperligação para uma foto da refeição de modo ao mesmo poder ser apresentado ao utilizador mais para a frente.



Figura 8 - Tabela dos Dias da Semana

Nesta tabela, encontra-se os dias da semana registados para a nossa aplicação. Neste caso, só guardamos mesmo os dias uteis.

A tabela guarda então o código do dia da semana, o tal que irá fazer referência as refeições, guarda o nome do dia, a data, e como sempre a informação de quando foi criado/atualizado.



Figura 9 - Tabela das Compras

Nesta tabela, conseguimos ver a tabela que irá armazenar as informações de uma compra. Neste caso como um utilizador pode fazer várias compras, criou-se a tabela de relação de muitos para muitos. Os campos codMeal, e codUser estabelecerão a devida conexão as suas entidades. Também conseguimos ver o campo flgUsed, neste caso o mesmo corresponde a uma booleana, no sentido de num desenvolvimento futuro ser utilizado a senha com sucesso, se registar a senha como usada, e assim a flgUsed passará a ser usada com sucesso. Por fim, a tabela também tem o campo do código da compra, para se poder consultar uma compra em específico mais tarde.

# Aplicação Final

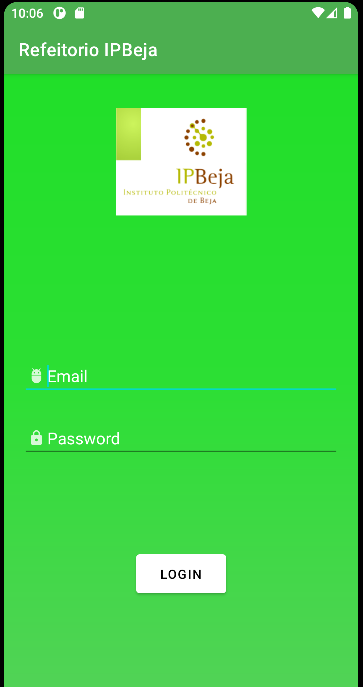


Figura 10 - Login da Aplicação

Na imagem em cima ilustrado, conseguimos observar o aspeto final do ecrã do login, assim como algumas semelhanças com o protótipo anteriormente apresentado. Também é possível verificar que existem 2 introduções de texto, um para o email do utilizador da pessoa em questão, e um para a sua respetiva palavra-passe. Após a introdução dos dados, os mesmos serão validados através da API, caso exista a conta do utilizador, o login será efetuado, caso contrário terá que tentar proceder ao início de sessão outra vez.

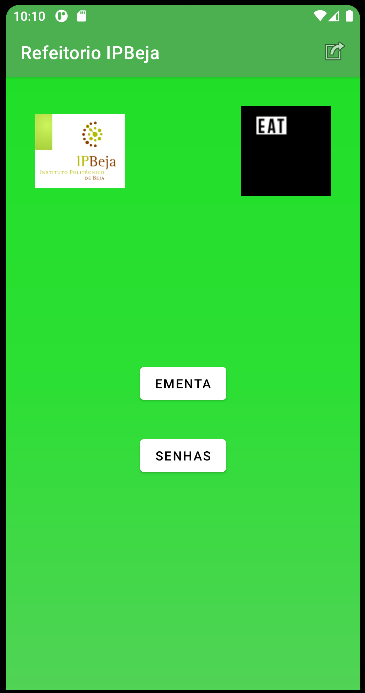


Figura 11 - Menu

Na imagem acima, conseguimos observar algumas mudanças em relação ao protótipo inicial, isto porque nos parecia mais otimizado, e mais user friendly, um design simples, mas eficaz. Encontram-se dois botões, um para nos levar para a lista dos dias da semana, e o outro para ver as senhas possuídas pelo utilizador. No canto superior direito o utilizador poderá escolher terminar a sua sessão. Caso não termine sessão, mas reinicie a aplicação a sessão mantém-se, já que o utilizador não o terminou. O utilizador também poderá carregar na sua própria fotografia, e isto levara nos para um ecrã com algumas opções do utilizador.

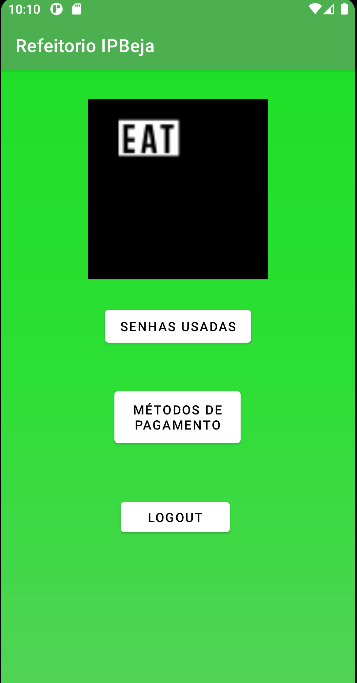


Figura 12 - Perfil

Tal como referido anteriormente, ao clicar na foto de utilizador, o mesmo leva-nos a um ecrã que contem algumas opções por onde o mesmo poderá escolher o que pretender fazer. Neste caso observamos um quadrado preto no qual o ecrã se encontra focado, pois é a “foto” de utilizador, da pessoa em questão. Temos um botão de senhas usadas, que mostra uma lista de senhas já usadas. Conseguimos observar também um botão de métodos de pagamento, nesta versão da aplicação, não se focamos na implementação do mesmo, é uma referência para uma possível atualização. E por fim, mais uma vez teremos um botão para o utilizador poder finalizar a sua sessão.

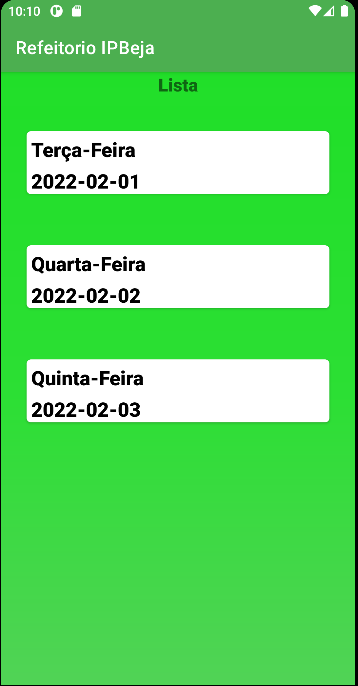


Figura 13 – Lista de Dias da Semana

A interface dos dias da semana pode ser acedida através do botão “ementa”, o qual se encontra no menu principal. Isto aqui como indicado, mostra todos os dias da semana registadas na nossa base de dados. Neste caso só demonstramos 3 dias, mas muitos mais podem ser adicionadas, que serão apresentados a mesma. Ao selecionar um dia da semana, o utilizador é redirecionado para uma interface com a lista de refeições para o dia escolhido.

Uma imagem com texto

Descrição gerada automaticamente

Figura 14 - Lista das Refeições Disponíveis

Na interface das refeições, o utilizador tem as opções da escolha da refeição do dia. Neste ecrã o utilizador encontra então informação sobre qual a sopa nesse dia, qual o prato principal, e qual a sobremesa incluída na refeição.

Após selecionar uma refeição, o mesmo é redirecionado para uma interface mais detalhada sobre a senha.



Figura 15 - Detalhes de uma Refeição

Na interface com a lista de refeições selecionada anteriormente, o utilizador é então levado para o ecrã com os detalhes. É apresentado uma imagem do prato principal, a sopa do dia, e a sobremesa incluída. O utilizador consegue escolher entre comprar a senha, encontrando-se com um alerta de confirmação de compra demonstrado já de seguida, ou caso escolha o botão “voltar” para regressar ao menu principal.

Uma imagem com texto

Descrição gerada automaticamente

Figura 16 – Confirmar Compra

Tal como referido anteriormente, é nesta parte do ecrã, na confirmação que o utilizador escolhe entre comprar a senha ou não. Caso decida comprá-la, a mesma fica registada, e poderá ser consultada mais tarde.

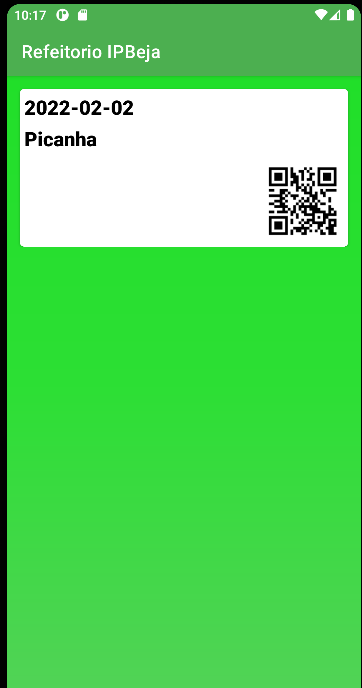


Figura 17 – Lista das Senhas (com QR code)

No menu principal, ao selecionar a opção senhas, o utilizador é redirecionado para uma interface com uma lista de senhas compradas, caso o mesmo pressione a senha, um QR code aparece no ecrã. O QR code tem o objetivo de levar scan quando o utilizador está a ser servido no refeitório, o mesmo código contém o dia da refeição, e qual a refeição comprada. As senhas têm uma propriedade (denominada de flgUsed) para determinar se a senha já foi utilizada anteriormente. Na nossa aplicação não é possível declarar a refeição como consumida, já que não temos uma aplicação totalmente a parte que permite ler QR code e declarar a senha como usada, mas o mesmo é possível fazer-se. Neste caso a lista das refeições consumidas/não consumidas terão um design quase idêntico.

# API

A API tem como objetivo prestar auxílio a aplicação Android, de modo a encher os dados através de Endpoints

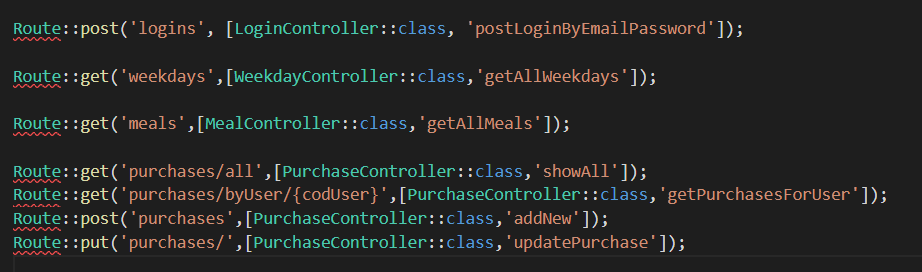


Figura 18 - Endpoints da nossa API

Na figura acima representado, conseguimos observar os endpoints que a nossa API usa. Neste caso a nossa aplicação do Android Studio faz pedidos e através destas routes as mesmas são direcionadas para os devidos métodos dos controladores. O que acontece. depois depende do que cada controlador precisa.

Por exemplo, o Post nos logins, irá mandar o email do utilizador, e a palavra-passe do mesmo, depois dependendo da resposta, a aplicação fará o que for mais correto.

Os Gets vão buscar uma lista de todos os Dias/Refeições registadas na base de dados. Ou seja, mais uma vez faz o endereçamento necessário para o Android Studio.

O mesmo envolve muito mais código por trás, mas o mais importante está mesmo listado em cima.

# BackOffice (CRUD)

O BackOffice tem como objetivo permitir a gestão dos utilizadores, dos dias da semana (não listar os feriados por exemplo), as refeições para cada dia, e até mesmo das compras.

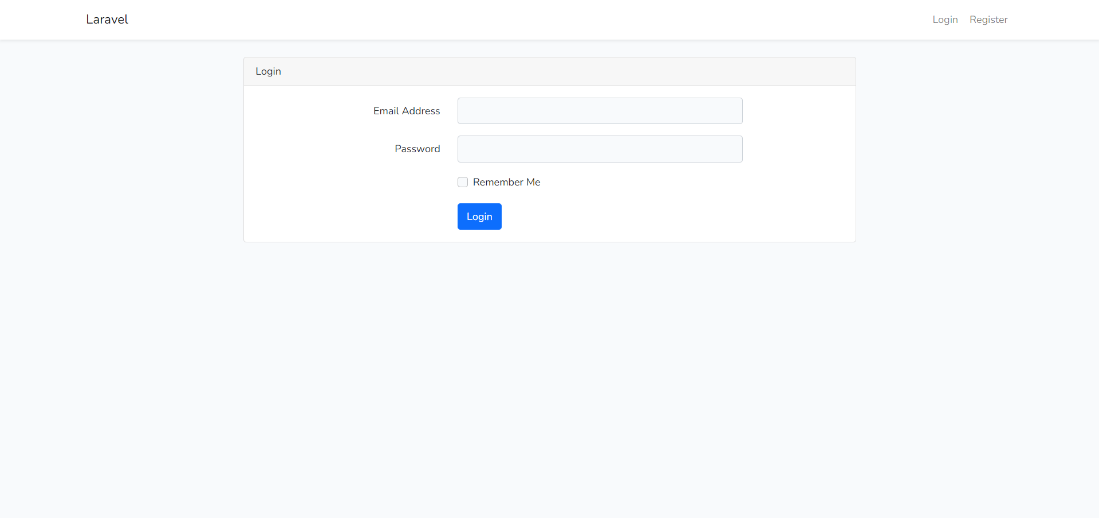


Figura 19 - Página para Iniciar Sessão

Para aceder ao BackOffice é necessária uma autenticação, a mesma tem que ser inserida manualmente na base de dados. Neste caso criamos a conta principal do administrador, e de seguida não deixamos registar mais ninguém para já, já que o objetivo disto tudo não é que qualquer pessoa se registe.

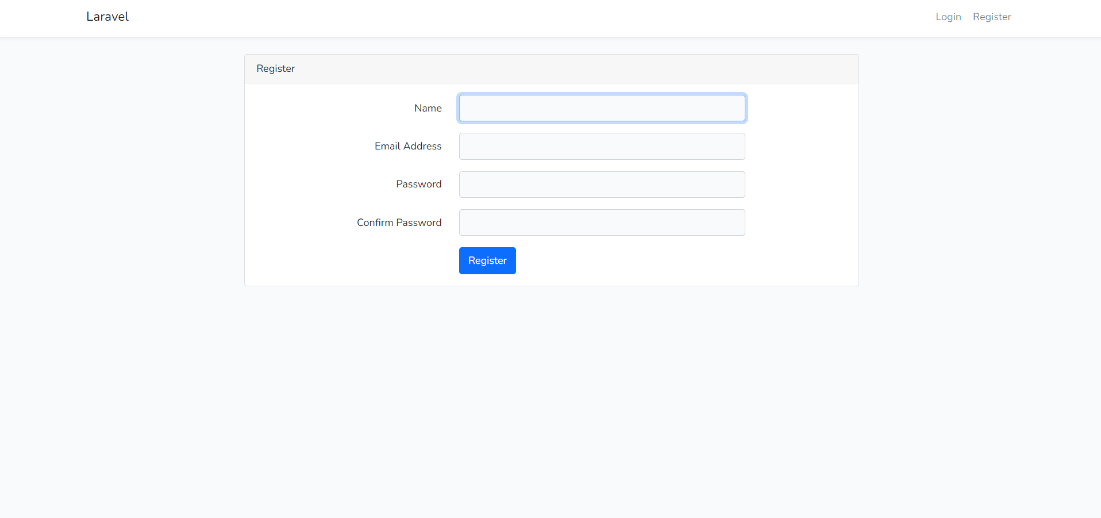


Figura 20 - Registar um Novo Utilizador

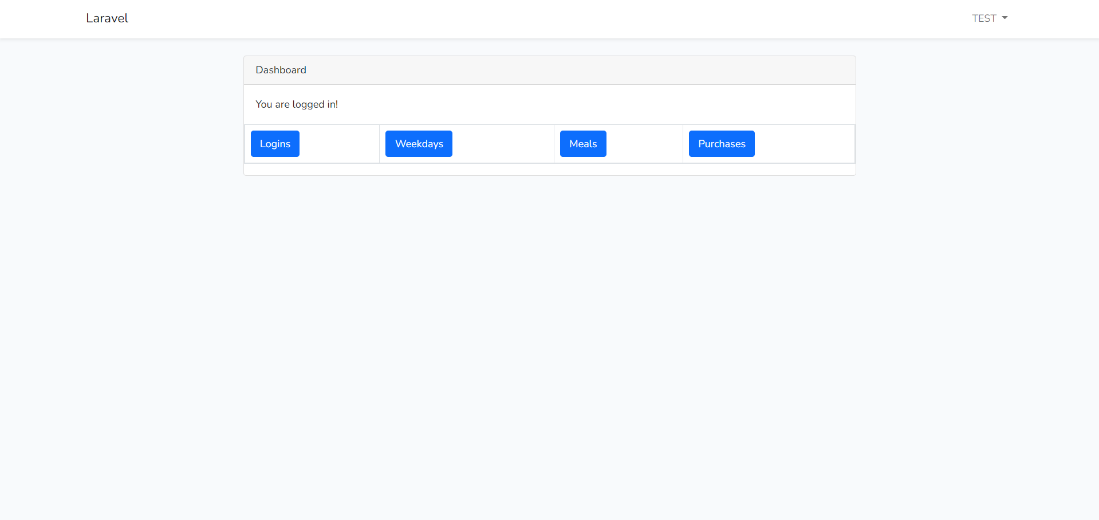


Figura 21 – Index

A Dashboard permite ao utilizador escolher qual é a tabela da base de dados que pretende gerir, através do clique no botão seremos redirecionados para os mesmos. Atenção que todas as hiperligações só de possível acesso caso o administrador tenha a sessão iniciada.

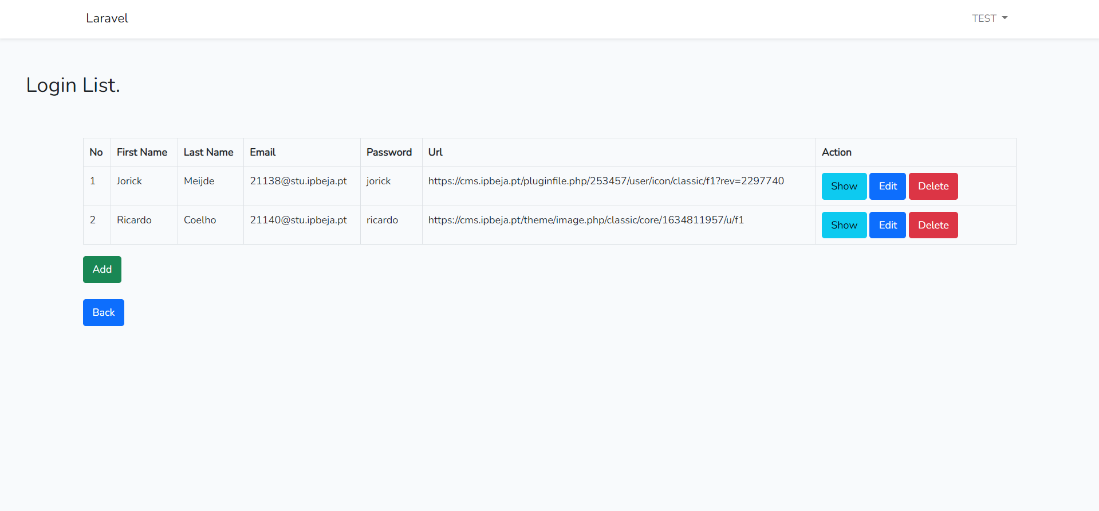


Figura 22 - Apresentação das tabelas dos Logins

Após escolher o que gerir, o utilizador depara-se com uma interface com todos os registos que estão na base de dados. Também tem diversas possibilidades, como por exemplo o botão “Add”, que permite adicionar um novo registo, o “Show” que permite ver os detalhes de um registo, o “Edit”, o qual oferece a possibilidade de alterar um registo atual, o “Delete”, que apaga o registo escolhido, e o botão “Back” que permite voltar ao menu principal. Neste caso está apresentado a tabela dos logins, mas o design das outras tabelas é semelhante.

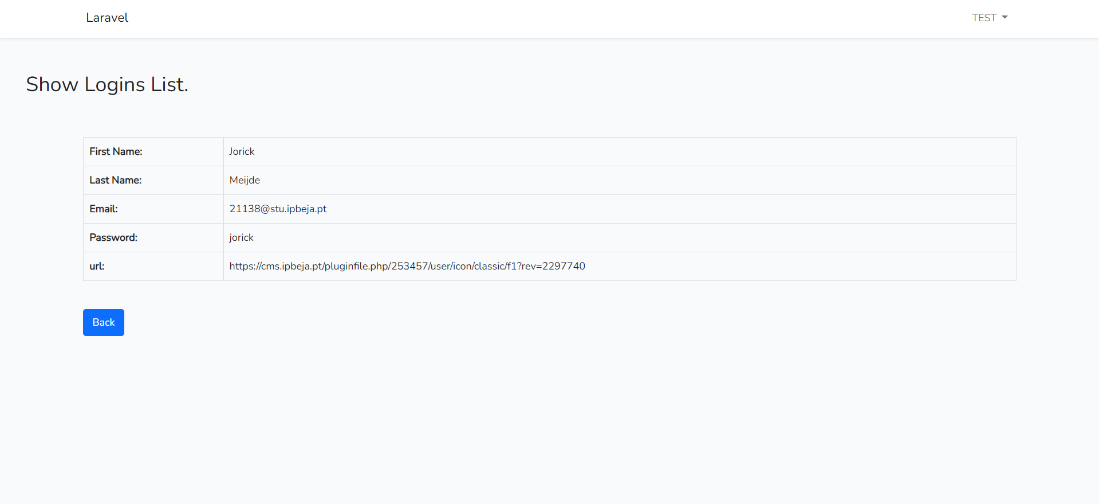


Figura 23 - Mostrar um Login Específico

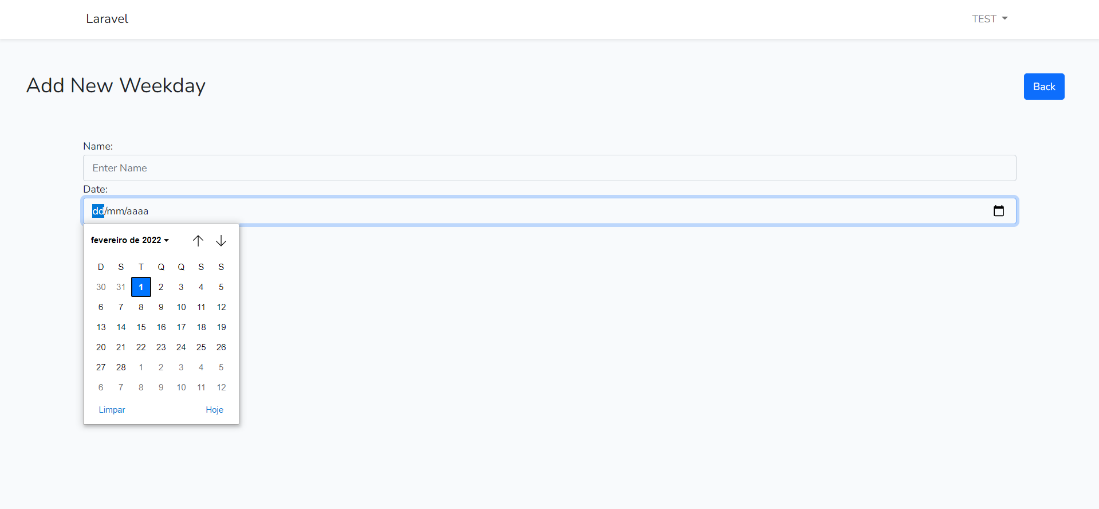


Figura 24 - Adicionar um novo dia de semana

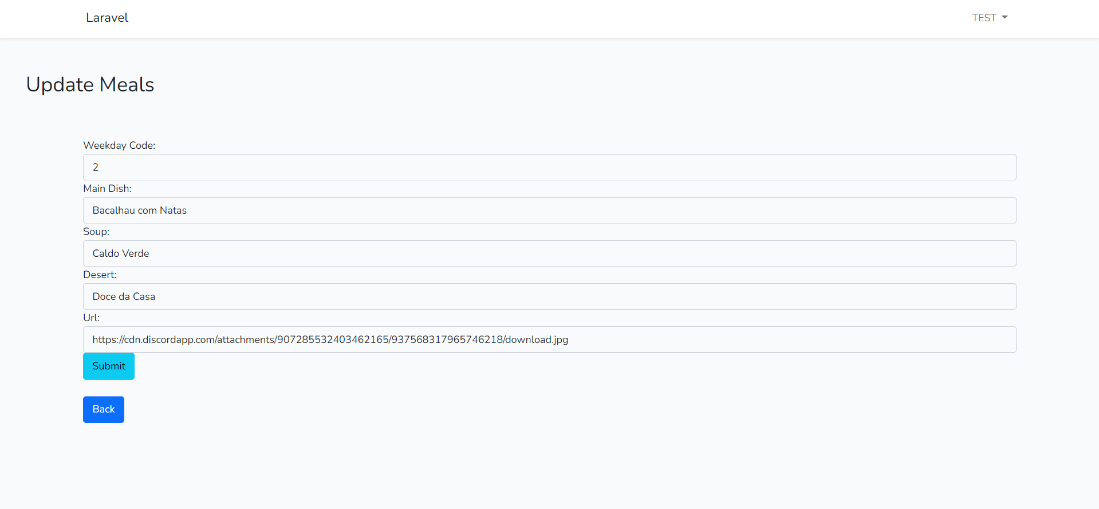


Figura 25 - Atualizar as Refeições

# **Futuras Implementações**

Nesta parte do relatório, são apresentadas algumas ideias que nos foram ocorrendo ao longo da elaboração deste projeto, mas que por algum motivo não foram aplicadas, ou por falta de tempo, ou porque a equipa assim o decidiu…

Algumas ideias, que poderão ser implementadas no futuro:

* Gerir métodos de pagamento;
* Carregar as purchases por user, e não no geral e depois filtrar tudo (tentou se fazer de outra maneira, mas estávamos a ter dificuldades com o mesmo);
* Evitar um erro, onde ao abrir logo os purchases o programa crashar, carregar primeiro as meals/weekdays;

# Conclusão

Ao longo da realização este projeto, foi possível reter melhor os conhecimentos provenientes obtidos neste curso.

Podemos adicionar, que através das cadeiras lecionadas anteriormente, conseguimos melhorar o projeto em questão, já que em todas as respetivas cadeiras, se pensou em formas de melhorar o mesmo, respeitando as regras todas em questão, por exemplo regras de usabilidade, ou questões de design.

Também podemos concluir, que ambos sentimos que aprendemos bastante com a elaboração deste projeto, já que é um trabalho mais autónomo, o que nos permitiu praticar as nossas aprendizagens, ajudando nos um ao outro quando necessário, e aprendendo mais os dois.