

Relatório 1º Milestone

Realizado por:

Artur Gentil Silva Pinto nº8230138

Willkie Bianchi Parahyba Filho nº8230127

Relatório 1º Milestone

Índice

| | |
|--|----|
| Índice..... | 2 |
| Índice de Figuras..... | 4 |
| Índice de Tabelas..... | 6 |
| Lista de Abreviaturas, siglas e símbolos | 7 |
| Introdução | 8 |
| Planeamento | 9 |
| APP.js | 10 |
| Routers..... | 12 |
| Index..... | 12 |
| Admin | 12 |
| Categorias..... | 14 |
| Porções..... | 16 |
| Lista de Restaurantes | 17 |
| Lista dos Users | 19 |
| Restaurantes | 20 |
| Restaurante | 22 |
| Encomendas | 24 |
| Registo de Restaurante..... | 25 |
| Login..... | 26 |
| SignUp..... | 27 |
| Perfil | 28 |
| Checkout | 29 |
| Funcionamentos das Paginas Web..... | 30 |
| HomePage..... | 30 |
| Registo de Restaurante | 31 |
| NavBars..... | 33 |

Relatório 1º Milestone

| | |
|--|----|
| NavBar | 33 |
| NavBar do Restaurante..... | 34 |
| NavBar do Admin | 35 |
| Login..... | 36 |
| SignUp..... | 37 |
| Admin Page | 38 |
| Página Editar Dados | 38 |
| Página alterar password | 39 |
| Lista de Restaurantes | 40 |
| Lista de Utilizadores..... | 44 |
| Lista de Categorias | 49 |
| Lista de Porções | 52 |
| Página com todos os Restaurantes | 54 |
| Página do restaurante | 56 |
| Páginas de Erro | 61 |
| Página de Erro 404 | 61 |
| Página de erro personalizada | 62 |
| Conteúdo a melhorar..... | 64 |
| Conclusão | 65 |

Índice de Figuras

| | |
|--|----|
| Figura 1 - Plano Visily | 9 |
| Figura 2 - Conexão MongoDb Atlas | 10 |
| Figura 3 - Sessão uso geral | 10 |
| Figura 4 - Declaração variáveis globais | 11 |
| Figura 5 - Middleware pagina 404 personalizada..... | 11 |
| Figura 6 - Rotas do Index..... | 12 |
| Figura 7 - Variáveis rotas do admin..... | 13 |
| Figura 8 - Aplicação de middleware rotas do admin..... | 13 |
| Figura 9 - rotas do admin | 13 |
| Figura 10 - importação rotas, na route admin | 14 |
| Figura 11 - rotas das categorias..... | 15 |
| Figura 12 - rotas da portions | 16 |
| Figura 13 - Variáveis Lista de Restaurantes | 17 |
| Figura 14 - 1ºimg rotas da lista de restaurantes..... | 17 |
| Figura 15 - 2º img rotas da lista de restaurantes..... | 18 |
| Figura 16 - Importação de rotas na lista de restaurantes | 18 |
| Figura 17 - Rotas da lista de aprovação de restaurantes..... | 18 |
| Figura 18 - 1º img rotas da lista dos users | |
| Figura 19 - 2º img rotas da | |
| lista dos users | 19 |
| Figura 20 - rotas restaurantes sem autentificação | 20 |
| Figura 21 - Lista de Restaurantes autentificacao Tokens | 20 |
| Figura 22 - 1ºimg rotas restaurantes aut. | 21 |
| Figura 23 - 2ºimg rotas restaurantes aut. | 21 |
| Figura 24 - 1º img rota do restaurante..... | 22 |
| Figura 25 - 2º img rota do restaurante..... | 23 |
| Figura 26 - 3º img rota do restaurante..... | 23 |
| Figura 27 - rota tipo utilizar, rota das orders..... | 24 |
| Figura 28 - rotas das encomendas..... | 24 |
| Figura 29 - rotas do registo de restaurantes | 25 |
| Figura 30 - rotas do login | 26 |
| Figura 31 - Rotas do signup | 27 |
| Figura 32 - rotas do router do perfil | 28 |
| Figura 33 - Rota do carrinho de compras..... | 29 |
| Figura 34 - aceder pagina registrar restaurante | 31 |
| Figura 35 - Imagem para criação de um restaurante | 32 |
| Figura 36 - NavBar base | 33 |
| Figura 37 - NavBar autenticado | 33 |
| Figura 38 - NavBar aut. Restaurante..... | 34 |
| Figura 39 - NavBar fora da homePage..... | 34 |
| Figura 40 - Nav Bar do restaurante | 34 |
| Figura 41 – navbarAdmin | 35 |

Relatório 1º Milestone

| | |
|--|----|
| Figura 42 - Página de login | 36 |
| Figura 43 - Página de SignUp | 37 |
| Figura 44 - Pagina do admin | 38 |
| Figura 45 - Página edição dados do admin..... | 39 |
| Figura 46 - Página alterar password admin..... | 39 |
| Figura 47 - Página lista de restaurantes..... | 41 |
| Figura 48 – Página dos restaurantes por aprovar | 41 |
| Figura 49 - Página com informação dos restaurantes..... | 43 |
| Figura 50 - Lista de Utilizadores..... | 45 |
| Figura 51 - Página adicionar utilizador | 46 |
| Figura 52 - Prioridade Dono | 47 |
| Figura 53 - Mais Info utilizador | 48 |
| Figura 54 - Lista de categorias..... | 49 |
| Figura 55 - Página para criar uma categoria | 51 |
| Figura 56 - Lista de Porções | 52 |
| Figura 57 - Página dos restaurantes | 54 |
| Figura 58 - Lista dos restaurantes aut. admin | 55 |
| Figura 59 - Informação do restaurante | 56 |
| Figura 60 - Editar Restaurante | 56 |
| Figura 61 - menus do restaurante | 57 |
| Figura 62 - Página criação dos menus..... | 57 |
| Figura 63 - 2º img. Página Criação menu | 58 |
| Figura 64 - 3º img. Página Criação menu | 58 |
| Figura 65 - Dados Nutricionais do menu | 59 |
| Figura 66 - Imformação do menu | 59 |
| Figura 67 - Informação dos pratos | 60 |
| Figura 68 - Adicionar novo prato | 60 |
| Figura 69 - Filtros dos pratos | 61 |
| Figura 70 - Card de um prato | 61 |
| Figura 71 - Pagina error 404 | 62 |
| Figura 72 - Pagina de erro personalizada para error 403..... | 63 |
| Figura 73 - Pagina de erro personalizado para o error 500..... | 63 |

Índice de Tabelas

| | |
|---|---|
| Tabela 1 - Lista de Abreviaturas, siglas e símbolos | 7 |
|---|---|

Lista de Abreviaturas, siglas e símbolos

| Sigla | Descrição |
|--------------|--|
| ID | Identificador único |
| CRUD | <i>Create, Read, Update and Delete</i> |
| ejs | Embedded Javascript |
| WEB | World Wide Web |
| | |

Tabela 1 - Lista de Abreviaturas, siglas e símbolos

Introdução

No âmbito do desenvolvimento de aumentar a produtividade, capacidade de gestão e vendas de restaurantes, pretendemos desenvolver uma plataforma web para auxiliar múltiplos restaurantes a gerir os pedidos e os seus menus.

O desenvolvimento desta plataforma tem como objetivo permitir as seguintes atividades, registo e autenticação de utilizadores, gestão de menus por parte dos restaurantes, sistema de encomendas online com opções de pagamento, notificações para encomendas, e gestão de conteúdo, também foram realizadas algumas atividades consideradas adicionais, como, utilização da API openFood facts para requisição de dados nutricionais dos ingredientes dos pratos e armazenamento de dados em cache para redução do número de pedidos necessários a esta API, visto que a mesma é extremamente lenta.

Este relatório inclui todas os requisitos e funcionalidades propostos para a primeira milestone

Planeamento

A metodologia de desenvolvimento escolhida pelos dois membros da nossa equipe foi a metodologia Scrum seguindo princípios ágeis, pois é a metodologia que mais dominamos e que se adequa ao nosso projeto, realizamos então SCRUM meetings diárias e o tempo de ciclo do sprint era semanal.

O arrancar deste projeto foi marcado pelo desenvolvimento de um mockup para termos uma base do número possíveis de páginas necessárias e uma ideia inicial do possível design que implementaríamos ao decorrer do projeto, como é possível ver na figura a seguir, para ter uma visão mais detalhada do mockup, é necessário aceder ao link afixado no fim do relatório

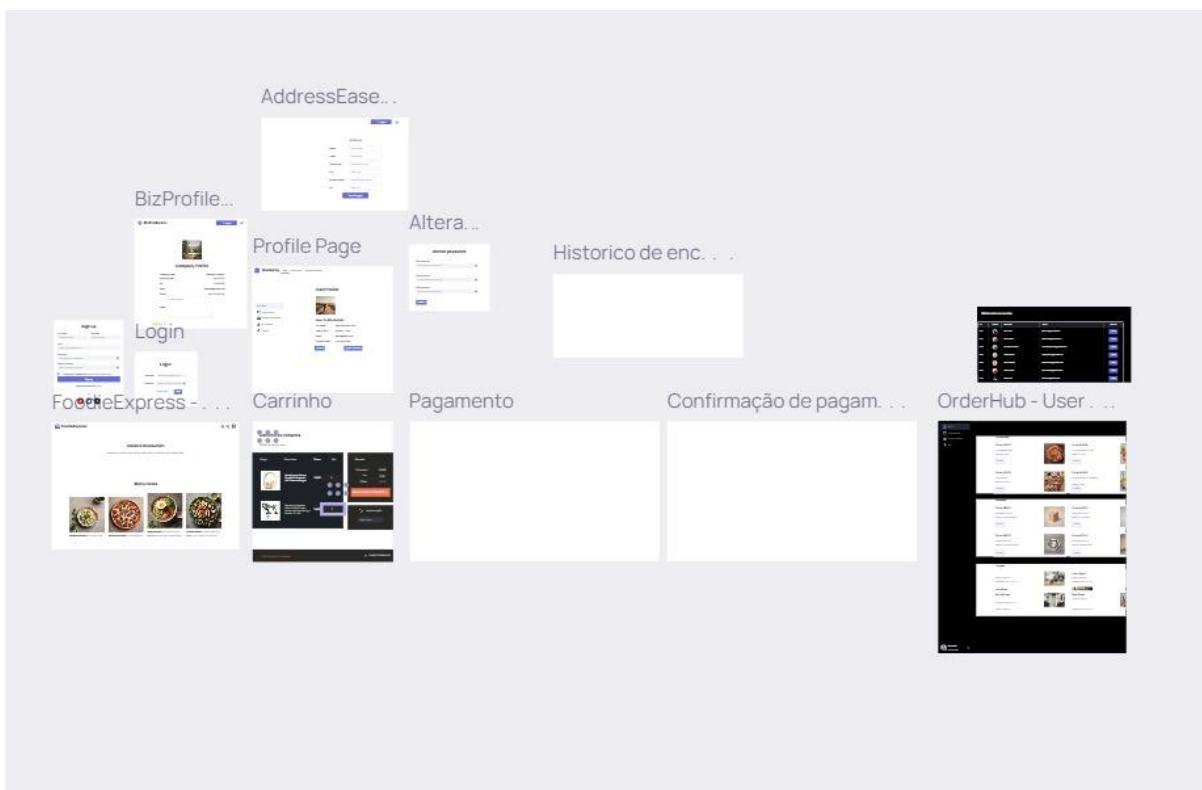


Figura 1 - Plano Visibly

O segundo passo do planeamento foi a criação de um SharePoint para melhor gestão do projeto, onde podemos fazer uma excelente divisão de tarefas e ter um bom controle de tempo. A divisão de tarefas foi feita de maneira uniforme para que ambos os elementos do grupo pudessem provar conhecimentos em todas as etapas do desenvolvimento.

Relatório 1º Milestone

Após toda a parte de preparação da gestão do projeto foi iniciado o desenvolvimento do mesmo, onde demos prioridade a todos os requisitos obrigatórios para a primeira milestone e em seguida realizamos os requisitos extras.

APP.js

A conexão com a base de dados, MongoDB Atlas, é assegurada conforme a imagem a seguir

```
17 mongoose.Promise = global.Promise;
18
19 //Conexão com o Atlas
20 mongoose.connect('mongodb+srv://UserGeral:1234@cluster0.rbiey8q.mongodb.net/TrabalhoPAW', {
21   useNewUrlParser: true,
22 }
23   .then(() => console.log('connection succesful'))
24   .catch((err) => console.error(err));
```

Figura 2 - Conexão MongoDB Atlas

Também definimos a sessão para uso geral nesse js.

```
43 app.use(session({
44   secret: 'TrabalhoPaw',
45   resave: true,
46   saveUninitialized: true,
47   cookie: { secure: false }
48 }));
```

Figura 3 - Sessão uso geral

Declaração das variáveis globais, para ter acesso a página atual, a anterior, e aos dados do usuário de maneira fácil e uniforme.

Relatório 1º Milestone

```
67 app.use((req, res, next) => {
68   // Se a requisição for para arquivos estáticos (imagens, CSS, JS), não redefine currentPage
69   if (req.path.startsWith('/images') || req.path.startsWith('/css') || req.path.startsWith('/js')) {
70     return next();
71   }
72
73   res.locals.currentPage = req.path === '/' ? 'homepage' : req.path;
74   res.locals.previousPage = req.headers.referer || '/';
75   res.locals.user = req.user || null;
76
77   console.log('User:', res.locals.user);
78   console.log('Current Page:', res.locals.currentPage);
79   console.log('Previous Page:', res.locals.previousPage);
80   next();
81 });

});
```

Figura 4 - Declaração variáveis globais

Na nossa APP.js criamos este middleware, que qualquer rota não encontrado receba uma página de erro 404 personalizada.

```
/* Middleware que mostra a pagina 404 personalizada
app.use((req, res, next) => {
  const isStatic = req.path.match(/^(?!index\.html)(\.(jpg|jpeg|png|gif|css|js|svg|ico|webp|woff2?|ttf)$)/i);
  |
  if (isStatic) {
    return res.sendStatus(404);
  }

  res.status(404).render('errors/error404', {
    currentPage: req.path,
    user: req.user || null
  });
});
```

Figura 5 - Middleware pagina 404 personalizada

Routers

Index

Nesta router, “index.js”, tem todas as rotas, que dizem respeito à página principal do site.

```
var express = require('express');
var router = express.Router();

//Controller
const indexController = require("../Controllers/IndexController.js");

//middleware
const autoLoginMiddleware = require('../Middleware/AutoLoginMiddleware');

/**
 * * Rota que carrega a página inicial
 *
 * * autoLoginMiddleware --> realiza automaticamente o login do utilizador, caso ele no login
 * * tenha clicado na opção "Lembrar-me"
 */
router.get('/', autoLoginMiddleware, indexController.indexPage);

/**
 * * Rota que permite ao utilizador pesquisar por um restaurante específico
 */
router.get('/search', indexController.search);

module.exports = router;
```

Figura 6 - Rotas do Index

Admin

Nesta router “admin.js”, definimos todas as rotas/páginas que um admin pode aceder no seu perfil.

Relatório 1º Milestone

```
const express = require("express");
const router = express.Router();

//Controllers
const adminController = require("../Controllers/AdminController.js");

//Middlewares
const {authenticateToken} = require("../Middleware/AuthTokenMiddleware.js");
const {isAdmin} = require("../Middleware/TypeUserMiddleware.js");

//Routers
const restaurantRouter = require("./adminRoutes/listrestaurants.js");
const userRouter = require("./adminRoutes/listusers.js");
const categoriesRouter = require("./adminRoutes/categories.js");
const passwordController = require("../Controllers/PasswordController.js");
const portionsRouter = require("./adminRoutes/portions.js");
```

Figura 7 - Variáveis rotas do admin

Na imagem abaixo é possível observar os middlewares que aplicamos a todas as rotas deste route, através do router.use.

```
/** 
 * * Rota que aplicar a todas as rotas deste js os seguintes middlewares
 *
 * * authenticateToken --> valida se o token do utilizador é valido.
 * * isAdmin --> verifica se o utilizador autenticado é um Admin
 * */
router.use(authenticateToken, isAdmin);
```

Figura 8 - Aplicação de middleware rotas do admin

Estas são as rotas que criamos para este router.

```
/** 
 * * Rota que carrega a pagina inicial de um admin
 * */
router.get("/", adminController.homePage);

/** 
 * * Rota que renderiza a página de edição dos dados do admin
 *
 * * adminId --> id do admin, cujo os dados vão ser editados.
 * */
router.get("/editDados/:adminId", adminController.editPage);

/** 
 * * Rota que atualiza os dados de um admin
 *
 * * accountId --> id do admin, cujo os dados serão atualizados
 * */
router.post("/updateAdmin/:accountId", adminController.updateAdmin);

/** 
 * * Rota que renderiza a página, de alteração da password, da conta do admin
 *
 * * accountId --> id do admin, cujo os dados serão atualizados
 * */
router.get("/editPassword/:accountId", passwordController.editPassword);

/** 
 * * Rota que guarda a nova password do perfil do admin
 *
 * * accountId --> id do admin, que a password foi atualizada
 * */
router.post("/changePassword/:accountId", passwordController.updatePassword);

/** 
 * * Rota que permite dar delete há conta do admin
 *
 * * adminId --> id do admin, que a conta foi eliminada
 * */
router.post("/deleteAccount/:adminId", adminController.deleteAdm);
```

Figura 9 - rotas do admin

Relatório 1º Milestone

Neste router realizamos a importação de quatro rotas principais que organizam o roteamento da aplicação admin em sub-routers.

```
/**
 * Importação do router com as rotas que permitem ao admin realizar o CRUD dos restaurantes
 */
router.use("/listRestaurants", restaurantRouter);

/**
 * Importação do router com as rotas que permitem ao admin realizar o CRUD dos utilizadores do site
 */
router.use("/listUsers", userRouter);

/**
 * Importação do router com as rotas que permitem ao admin realizar o CRUD das categorias
 */
router.use("/listCategories", categoriesRouter);

/**
 * Importação do router com as rotas que permitem ao admin realizar o CRUD das porções
 */
router.use("/listPortions", portionsRouter);

module.exports = router;
```

Figura 10 - importação rotas, na route admin

Categorias

Este router, “categories.js”, define todas as rotas do perfil do admin, que o permitem gerir todas as categorias, de pratos e menus do site. Na imagem abaixo é possível observar todas as rotas deste router.

Relatório 1º Milestone

```
const express = require("express");
const router = express.Router();

//Models
const categoriesController = require("../Controllers/Admin/CategoriesController.js");

/**
 * * Rota que carrega a pagina, com uma lista de todas as categorias
 */
router.get("/", categoriesController.homePage);

/**
 * * Rota que permite filtrar e ordenar dados na pagina das categorias
 */
router.get("/search", categoriesController.search);

/**
 * * Rota que carrega a pagina para criar uma categoria
 */
router.get("/createCategory", categoriesController.createCategory);

/**
 * * Rota que guarda uma categoria, criada na página de criação de uma categoria
 */
router.post("/saveCategory", categoriesController.saveCategory);

/**
 * * Rota que carrega a pagina para editar uma categoria
 *
 * * categoryId --> ID da categoria, que vai ser editada
 */
router.get("/editCategory/:categoryId", categoriesController.editCategory);

/**
 * * Rota que faz o update dos dados de uma categoria
 *
 * * categoryId --> ID da categoria, a ser atualizada
 */
router.post("/updateCategory/:categoryId", categoriesController.updateCategory);

/**
 * * Rota que faz o delete de uma categoria específica
 *
 * * categoryId --> ID da categoria, a ser eliminada
 */
router.post("/deleteCategory/:categoryId", categoriesController.deleteCategory);

module.exports = router;
```

Figura 11 - rotas das categorias

Relatório 1º Milestone

Porções

Este router, “portions.js”, define todas as rotas do perfil do admin, que o permitem gerir todas as porções, de pratos e menus do site. Na imagem abaixo é possível observar todas as rotas deste router, que permitem o CRUD das porções e o render da pagina de gestão das porções, e somente é permitido aceder a qualquer uma dessas rotas no caso de quem está a tentar aceder a rota está autenticado e se o tipo de perfil é de administrador

```
const express = require("express");
const router = express.Router();

//Controllers
const portionsController = require("../Controllers/ControllersAdmin/PortionController.js");

/**
 * * Rota que renderiza pagina com a lista das porções existentes
 */
router.get("/", portionsController.homePage);

/**
 * * Rota que permite filtrar e ordenar dados na pagina das categorias
 */
router.get("/search", portionsController.search);

/**
 * * Rota que carrega a página para criação de uma porção
 */
router.get("/createPortion", portionsController.createPortion);

/**
 * * Rota que permite guardar uma nova porção
 */
router.post("/savePortion", portionsController.savePortion);

/**
 * * Rota que carrega a página para edição de uma porção
 *
 * * portionId --> Id da porção a ser editada
 */
router.get("/editPortion/:portionId", portionsController.editPortion);

/**
 * * Rota que atualiza os dados de uma porção, já existente
 *
 * * portionId --> Id da porção a ser atualizada
 */
router.post("/updatePortion/:portionId", portionsController.updatePortion);

/**
 * * Rota que elimina uma porção
 *
 * * portionId --> Id da porção a ser eliminada
 */
router.post("/deletePortion/:portionId", portionsController.deletePortion);

module.exports = router;
```

Figura 12 - rotas da portions

Relatório 1º Milestone

Lista de Restaurantes

Este router, “listrestaurants.js”, define todas as rotas do perfil do admin, que o permitem gerir todas as categorias, de pratos e menus do site. Na imagem abaixo é possível observar todas as rotas deste router.

```
const express = require("express");
const router = express.Router();

//Controllers
const restaurantsController = require("../Controllers/RestaurantsController.js");
const listrestaurantController = require("../Controllers/ControllersAdmin/ListRestaurantController.js");

//Routers
const aprovesRouter = require("./listaproves.js");
```

Figura 13 - Variáveis Lista de Restaurantes

Nas imagens abaixo, poderá observar todas as rotas que constituem este router.

```
/**
 * * Rota que carrega a pagina com a lista de todos os restuarantes
 */
router.get('/', listrestaurantController.homePage);

/**
 * * Rota que carrega a pagina para visualizar os dados de um restaurante especifico
 *
 * * restaurantId --> id do restaurante a ser visualizado
 */
router.get('/showRest/:restaurantId', listrestaurantController.showRestaurant);

/**
 * * Rota para puder filtrar por restaurantes, e definir um criterio de ordenação
 */
router.get('/search', restaurantsController.search);

/**
 * * Rota para renderizar a pagina para criar um restaurante
 */
router.get('/createRestaurant', restaurantsController.createRestaurant)

/**
 * * Rota que guarda um restarante criado na pagina "createRestaurant"
 */
router.post('/saveRestaurant', restaurantsController.saveRestaurant)

/**
 * * Rota para renderizar a pagina para editar restaurante
 *
 * * restaurantId --> Id do restuarante que vai ser editado
 */
router.get('/editRestaurant/:restaurantId', restaurantsController.editRestaurant)

/**
 * * Rota que guarda as alterações realizas a um restaurante
 *
 * * restaurantId --> Id do restuarante que vai ser atualizado
 */
router.post('/updatRestaurant/:restaurantId', restaurantsController.updatRestaurant);
```

Figura 14 - 1ºimg rotas da lista de restaurantes

Relatório 1º Milestone

```
/**  
 * * Rota que elimina um restaurante  
 *  
 * * restaurant --> Id do restaurante que vai ser removido  
 * */  
router.post('/deleteRestaurant/:restaurant', restaurantsController.removeRestaurant);
```

Figura 15 - 2º img rotas da lista de restaurantes

Neste router realizamos a importação de uma rota principal, que daremos mais informação sobre o mesmo no próximo tópico.

```
/**  
 * * Importação do router com as rotas que permitem ao admin ver os restaurantes que precisam  
 * * de aprovação e aprovar ou rejeitar  
 * */  
router.use('/aproves', aprovesRouter);  
  
module.exports = router;
```

Figura 16 - Importação de rotas na lista de restaurantes

Lista Aprovação de Restaurantes

Esta rota, “listaproves.js”, define todas as rotas necessárias, para o admin, ver os restaurantes que estão há espera de aprovação, que o admin pode aprovar ou rejeitar. Na imagem abaixo é possível observar todas as rotas deste router.

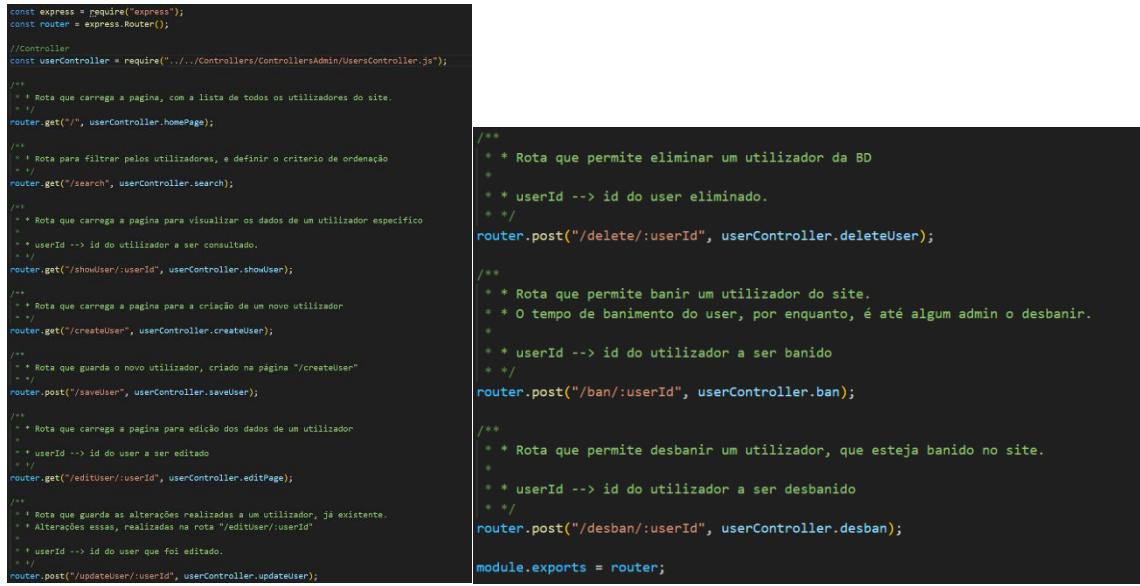
```
const express = require("express");  
const router = express.Router();  
  
//Controller  
const aprovacaoRestController = require("../../Controllers/Admin/AprovacaoRestController.js");  
const listrestaurantController = require("../../Controllers/Admin/ListRestaurantController.js")  
  
/**  
 * * Rota que carrega a página para aprovar ou rejeitar restaurantes  
 * */  
router.get('/', aprovacaoRestController.aprovePage);  
  
/**  
 * * Rota que carrega a com a informação de um restaurante específico  
 *  
 * * restaurantId --> Id do restaurante, do restaurante a ser consultado  
 * */  
router.get('/showRestaurant/:restaurantId', listrestaurantController.showRestaurant);  
  
/**  
 * * Rota para aprovar um restaurante, atualizando o seu estado como aprovo, permitindo com que o mesmo  
 * * possa ser visualizado pelos os utilizadores  
 *  
 * * restaurantId --> Id do restaurante que vai ser aprovado  
 * */  
router.post('/:restaurantId', aprovacaoRestController.aproveRestaurant);  
  
/**  
 * * Rota que rejeita um restaurante, removendo-o da DB  
 *  
 * * restaurantId --> Id do restaurante que vai ser desaprovado  
 * */  
router.post('/reject/:restaurantId', aprovacaoRestController.rejectRestaurant);  
  
module.exports = router;
```

Figura 17 - Rotas da lista de aprovação de restaurantes

Relatório 1º Milestone

Lista dos Users

Este router, “listusers.js”, define todas as rotas do perfil do admin, que o permitem gerir todos os utilizadores do site. Na imagem abaixo é possível observar todas as rotas deste router. Na imagem abaixo podes consultar todas as rotas deste router.



```
const express = require("express");
const router = express.Router();

//controller
const userController = require("../Controllers/ControllersAdmin/UsersController.js");

/**
 * Rota que carrega a pagina, com a lista de todos os utilizadores do site.
 */
router.get("/", userController.homePage);

/**
 * Rota para filtrar pelos utilizadores, e definir o criterio de ordenação
 */
router.get("/search", userController.search);

/**
 * Rota que carrega a pagina para visualizar os dados de um utilizador específico
 *
 * @param {String} userId --> id do utilizador a ser consultado.
 */
router.get("/showUser/:userId", userController.showUser);

/**
 * Rota que carrega a pagina para a criação de um novo utilizador
 */
router.get("/createUser", userController.createUser);

/**
 * Rota que guarda o novo utilizador, criado na página "/createUser"
 */
router.post("/saveUser", userController.saveUser);

/**
 * Rota que carrega a pagina para edição dos dados de um utilizador
 *
 * @param {String} userId --> id do user a ser editado
 */
router.get("/editUser/:userId", userController.editPage);

/**
 * Rota que guarda as alterações realizadas a um utilizador, já existente.
 * Alterações essas, realizadas na rota "/editUser/:userId"
 *
 * @param {String} userId --> id do user que foi editado.
 */
router.post("/updateUser/:userId", userController.updateUser);

/**
 * Rota que permite eliminar um utilizador da BD
 *
 * @param {String} userId --> id do user eliminado.
 */
router.post("/delete/:userId", userController.deleteUser);

/**
 * Rota que permite banir um utilizador do site.
 * O tempo de banimento do user, por enquanto, é até algum admin o desbanir.
 *
 * @param {String} userId --> id do utilizador a ser banido
 */
router.post("/ban/:userId", userController.ban);

/**
 * Rota que permite desbanir um utilizador, que esteja banido no site.
 *
 * @param {String} userId --> id do utilizador a ser desbanido
 */
router.post("/desban/:userId", userController.desban);

module.exports = router;
```

Figura 18 - 1º img rotas da lista dos users

Figura 19 - 2º img rotas da lista dos users

Restaurantes

Este router, “restaurants.js”, ela define todas as rotas, que os utilizadores podem aceder na página, que mostra todos os restaurantes aprovados no site. Na imagem a baixo é possível ver as rotas, que não precisam do utilizador estar autenticado para aceder.

```
var express = require('express');
var router = express.Router();

//Controllers
const restaurants = require("../Controllers/RestaurantsController.js");
const passwordController = require("../Controllers/PasswordController.js");

//Middlewares
const {authenticateToken} = require("../Middleware/AuthTokenMiddleware.js");
const {isDonoClienteOrAdmin, isDonoOrAdmin, isRestaurant} = require("../Middleware/TypeUserMiddleware.js");

//Router
const routeRestaurant = require("./restaurant.js");

/**
 * * Rota que carrega a pagina com todos os restaurantes do site, aprovados.
 */
router.get('/', restaurants.restaurantsPage);

/**
 * * Página que filtrar pelos restaurantes, e definir o criterio de ordenação.
 */
router.get('/search', restaurants.search);

/**
 * * Rota que carrega o router com as rotas padrão para a pagina de cada restaurante
 */
router.use('/:restaurant', routeRestaurant);
```

Figura 20 - rotas restaurantes sem autentificação

Na imagem abaixo é possível observar a rota que aplica a autentificação de tokens nas restantes, obrigando os utilizadores a estarem a estar autenticados para aceder as restantes rotas.

```
/**
 * * Rota que aplicar a todas as rotas deste js o seguinte middleware
 *
 * * authenticateToken --> valida se o token do utilizador é valido.
 */
router.use(authenticateToken);
```

Figura 21 - Lista de Restaurantes autentificacao Tokens

Relatório 1º Milestone

Nas restantes imagens é possível observar as rotas que precisam de autentificação do utilizador e em cada uma delas o tipo de utilizador que pode aceder às mesmas

```
/**  
 * * Rota que renderiza a página de criação de um restaurante, caso o utilizador seja admin ou  
 * * a página de pedido de criação de um restaurante caso o utilizador seja um dono ou cliente  
 *  
 * * isDonoClienteOrAdmin --> verifica se o user é um dono, cliente ou admin  
 */  
router.get('/createRestaurant', isDonoClienteOrAdmin, restaurants.createRestaurant);  
  
/**  
 * * Rota que guarda um novo restaurante, caso o utilizador seja admin ou guarda um pedido de  
 * * criação de um restaurante caso o utilizador seja um dono ou cliente  
 *  
 * * isDonoClienteOrAdmin --> verifica se o user é um dono, cliente ou admin  
 */  
router.post('/saveRestaurant', isDonoClienteOrAdmin, restaurants.saveRestaurant);  
  
/**  
 * * Rota que renderiza a pagina de edição dos caso o utilizar seja o dono do próprio restaurante  
 * * ou um admin.  
 *  
 * * isDonoOrAdmin --> verifica se o user é um dono ou admin  
 */  
router.get('/editRestaurant/:restaurantId', isDonoOrAdmin, restaurants.editRestaurant);  
  
/**  
 * * Rota que permite guardar os dados, alterados de um restaurante, pelo dono ou admin  
 *  
 * * isDonoOrAdmin --> verifica se o user é um dono ou admin  
 */  
router.post('/updateRestaurant/:restaurantId', isDonoOrAdmin, restaurants.updateRestaurant);
```

Figura 22 - 1ºimg rotas restaurantes aut.

```
/**  
 * * Rota que permite ao restaurante alterar a password da sua conta  
 *  
 * * isRestaurant --> verifica se o user é um restaurante  
 */  
router.get('/editRestaurant/editPassword/:accountId', isRestaurant, passwordController.editPassword);  
  
/**  
 * * Rota que permite ao restaurante guardar a sua nova password  
 *  
 * * isRestaurant --> verifica se o user é um restaurante  
 */  
router.post('/editRestaurant/changePassword/:accountId', isRestaurant, passwordController.updatePassword);  
  
/**  
 * * Rota que dá delete de um restaurante no site, por parte do dono ou de um admin  
 *  
 * * isDonoOrAdmin --> verifica se o user é um dono ou admin  
 */  
router.post('/deleteRestaurant/:restaurant', isDonoOrAdmin, restaurants.removeRestaurant);  
  
module.exports = router;
```

Figura 23 - 2ºimg rotas restaurantes aut.

Restaurante

O router “restaurant.js” é responsável por ter as rotas, que importam os router das encomendas e pratos, para alem de ter todas as rotas responsáveis por realizar o CRUD dos menus e pratos de um restaurante específico.

```
var express = require('express');
var router = express.Router({ mergeParams: true });

//Controller
var restaurant = require("../Controllers/RestaurantController.js");
var menu = require("../Controllers/ControllersRestaurant/MenuController.js");

//Middlewares
const {authenticateToken} = require("../Middleware/AuthTokenMiddleware.js");
const {isDonoOrAdmin} = require("../Middleware/TypeUserMiddleware.js");

//Rotas
const order = require("./restaurantRoutes/orders.js");
const dish = require("./restaurantRoutes/dish.js");
const comments = require("./restaurantRoutes/comments.js");

/**
 * * Rota que renderiza a home page de um restaurante
 */
router.get('/', restaurant.homePage);

/**
 * * Importação das rotas dos comentários
 */
router.use('/comments', comments);

/**
 * * Importação das rotas das dish (pratos)
 */
router.use('/menu', dish);

/**
 * * Rota para a página que mostra informação sobre o menu
 */
router.get('/showMenu/:menu', menu.showMenu);

/**
 * * Rota que permite filtrar por menus, de um restaurante, e também ordenar os menus
 */
router.get('/search', restaurant.searchMenu);
```

Figura 24 - 1º img rota do restaurante

Relatório 1º Milestone

```
/**  
 * * Rota que permite filtrar pelos pratos, de um menu, e também ordenar os menus  
 * */  
router.get('/showMenu/:menu/search', menu.searchMenu);  
  
/**  
 * * Rota que aplicar as rotas abaixo o seguinte middleware  
 *  
 * * authenticateToken --> valida se o token do utilizador é valido.  
 * */  
router.use(authenticateToken);  
  
/**  
 * * Importação das rotas das encomendas  
 * */  
router.use('/orders', order);  
  
/**  
 * * Rota que aplicar as rotas abaixo o seguinte middleware  
 *  
 * * isDonoOrAdmin --> verifica se o utilizador é um dono ou Admin.  
 * */  
router.use(isDonoOrAdmin);  
  
/**  
 * * Rota que carrega a pagina para criar um menu  
 * */  
router.get('/createMenu', menu.createMenu);  
  
/**  
 * * Rota que permite guardar um novo menu  
 * */  
router.post('/saveMenu', menu.saveMenu);  
  
/**  
 * * Rota que carrega a pagina para editar um menu  
 *  
 * * menuId --> id do menu a ser editado  
 * */  
router.get('/editMenu/:menuId', menu.editMenu);
```

Figura 25 - 2º img rota do restaurante

```
/**  
 * * Rota guardar as alterações realizadas num menu  
 *  
 * * menuId --> id do menu a ser atualizado  
 * */  
router.post('/updateMenu/:menuId', menu.saveEditMenu);  
  
/**  
 * * Rota que elimina um menu  
 *  
 * * menuId --> id do menu a ser eliminado  
 * */  
router.post('/deleteMenu/:menuId', menu.deleteMenu);  
  
module.exports = router;
```

Figura 26 - 3º img rota do restaurante

Relatório 1º Milestone

Encomendas

Este router, “orders.js”, constitui todas as rotas para que um restaurante possa gerir todas as suas encomendas, passadas e presentes. Na imagem abaixo é possível ver a route que aplica as restantes a necessidade do utilizador ser o dono ou um restaurante.

```
var express = require('express');
var router = express.Router({ mergeParams: true });
const mongoose = require('mongoose');

//Controllers
const order = require("../Controllers/ControllersRestaurant/OrderController.js");

//Middlewares
const {isDonoOrRestaurant} = require("../Middleware/TypeUserMiddleware.js");

/**
 * * Rota que aplicar a todas as rotas deste js os seguintes middlewares
 *
 * * isDonoOrRestaurant --> verifica se o utilizador é um dono ou restaurante
 */
router.use(isDonoOrRestaurant);
```

Figura 27 - rota tipo utilizar, rota das orders

Na imagem abaixo é possível observar as rotas deste router.

```
/**
 * * Rota que carrega a pagina para gestão das encomendas do restaurante
 */
router.get('/', order.addOrder);

/**
 * * Rota que carrega a pagina para o restaurante poder criar uma encomenda
 */
router.get('/createOrder', order.createOrder);

/**
 * * Rota que carrega a pagina para o restaurante gerir as encomendas que estão em andamento
 */
router.get('/orderManagement', order.orderManagement);

/**
 * * Rota que carrega a pagina, com o historico de encomendas do restaurante
 */
router.get('/historic', order.historicOrder);

/**
 * * Rota que carrega a pagina, com informação de uma encomenda específica
 */
router.get('historic/:orderId', order.showOrder);

/**
 * * Rota que altera o estado de uma encomenda
 */
router.post('/:orderId/status', order.updateOrderStatus);

module.exports = router;
```

Figura 28 - rotas das encomendas

Registo de Restaurante

Este router, “registRestaurant.js”, ele possui todas as rotas necessários para que um cliente ou um já dono de(s) restaurante(s), puder registar um novo restaurante. Sendo que para o dono ou utilizador puder registar um restaurante ele precisa de estar autenticado.

```
const express = require("express");
const router = express.Router();

//Controller
const restaurantsController = require("../Controllers/RestaurantsController.js");

//Middlewares
const {authenticateToken} = require("../Middleware/AuthTokenMiddleware.js");
const {isDonoOrCliente} = require("../Middleware/TypeUserMiddleware.js");

/**
 * * Rota que aplicar a todas as rotas deste js os seguintes middlewares
 *
 * * authenticateToken --> valida se o token do utilizador é valido.
 * * isDonoOrCliente --> verifica se o utilizador autenticado é um Dono ou cliente
 * */
router.use(authenticateToken, isDonoOrCliente)

/**
 * * Rota que renderiza a página de pedido de criação de um restaurante, no site. Que após
 * * a criação do mesmo o admin poderá aprova-lo ou rejeita-lo.
 * */
router.get("/", restaurantsController.createRestaurant);

/**
 * * Rota que guarda o pedido de criação de um restaurante
 * */
router.post("/saveRestaurant", restaurantsController.saveRestaurant);

module.exports = router;
```

Figura 29 - rotas do registo de restaurantes

Login

Na imagem abaixo é possível observar o nosso router “login.js”, que é o responsável por ter todas as rotas que permitem com que o utilizador realize o login e o logout no nosso site.

```
const express = require("express");
const router = express.Router();

//Controllers
const loginController = require("../Controllers/LoginController.js");

//Middlewares
const {possibleBlockLogin} = require("../Middleware/ValidateLoginMiddleware.js");

/**
 * * Rota que permite ao utilizador fazer logout da sua sessão
 */
router.get("/logOut", loginController.logout);

/**
 * * Aquila o middleware para block, nas rotas abaixo
 *
 * * possibleBlockLogin --> caso o utilizador já esteja autenticado, impedio de aceder as rotas de login
 */
router.use(possibleBlockLogin);

/**
 * * Rota que carrega a página de login
 */
router.get("/", loginController.login);

/**
 * * Rota que realiza o login do utilizador e atribui-lhe um token de autentificação
 */
router.post("/", loginController.loginToken);

module.exports = router;
```

Figura 30 - rotas do login

SignUp

Na imagem abaixo é possível observar as rotas do router “signup.js”, que permite com que um utilizador possa criar uma conta no nosso site.

```
var express = require('express');
var router = express.Router();

//Controllers
const signUpController = require("../Controllers/SignUpController.js");

//Middlewares
const {possibleBlockLogin} = require("../Middleware/ValidateLoginMiddleware.js");

/**
 * Aquila o middleware para block, nas rotas abaixo
 *
 * * possibleBlockLogin --> caso o utilizador já esteja autenticado, impedio de aceder as rotas de login
 */
router.use(possibleBlockLogin)

/**
 * Rota que carrega a pagina de signUp.
 */
router.get("/", signUpController.signup);

/**
 * Rota que cria a conta do utilizador
 */
router.post("/", signUpController.save);

module.exports = router;
```

Figura 31 - Rotas do signup

Perfil

Este router, “perfil.js”, ela possui as rotas para cada tipo de utilizador puder aceder as páginas do seu perfil. Para aceder a qualquer uma destas rotas é obrigatório o utilizador estar autenticado. Na imagem abaixo é possível observar todas as rotas, que criamos para este router para esta 1º milestone.

```
var express = require('express');
var router = express.Router();

//Middlewares
const {authenticateToken} = require("../Middleware/AuthTokenMiddleware.js");
const {isAdmin} = require("../Middleware/TypeUserMiddleware.js");

//Routers
const adminRouter = require("../routes/admin.js")

/**
 * * Rota que aplicar a todas as rotas deste js os seguintes middlewares
 *
 * * authenticateToken --> valida se o token do utilizador é valido.
 * */
router.use(authenticateToken);

/**
 * * Importação do router do perfil do admin
 * */
router.use('/admin', isAdmin, adminRouter);

module.exports = router;
```

Figura 32 - rotas do router do perfil

Checkout

O router, “checkout.js”, possui a rota responsável por carregar a página do carrinho de compras do utilizador.

```
const express = require("express");
const router = express.Router();

//Controllers
const carrinhoController = require("../Controllers/CarrinhoController");

/**
 * * Rota que carrega a pagina do carrinho de compras do utilizador
 */
router.get('/', carrinhoController.carrinhoPage);

module.exports = router;
```

Figura 33 - Rota do carrinho de compras

Funcionamentos das Páginas Web

HomePage

Na homepage é possível observar uma barra de navegação adaptativa com o logo e nome da empresa fictícia para qual estamos a desenvolver o portal web. A barra de navegação varia para o caso de o utilizador estar logado ou não como é possível observar nas imagens a seguir



Também é possível observar na imagem acima que a página possui uma barra de pesquisa, que atualmente só permite que sejam buscados restaurantes, mas futuramente deve permitir buscar também por pratos. Ainda na mesma página, temos uma secção de restaurantes recomendados somente para ilustração atualmente, mas futuramente será implementado, onde mostra os 3 restaurantes que mais se destacam cada um por um parâmetro definido posteriormente

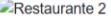
Relatório 1º Milestone

Recomendados para você



Restaurante 1

Comida deliciosa e entrega rápida.



Restaurante 2

Opções variadas com preços acessíveis.



Restaurante 3

Especialidades exclusivas e sabores únicos.

[VER TODOS OS RESTAURANTES](#)

Também possuímos uma secção a contar uma breve história da nossa empresa.

O que é o Click&Eat?

O Click&Eat é uma plataforma de entrega de refeições que conecta restaurantes e clientes, oferecendo uma experiência prática e deliciosa.

Com o Click&Eat, você pode explorar uma variedade de opções gastronômicas, fazer pedidos online e receber suas refeições no conforto da sua casa ou escritório.

Registo de Restaurante

Para poder registar um utilizador é necessário o utilizador ter uma conta e estar autenticado e depois na secção “Junte-se a nós” clicar no botão “Adicione o seu restaurante”, que pode ser observado na imagem abaixo.

Junte-se a nós!



Fornça refeições aos seus colaboradores

[Crie uma conta empresarial](#)



As refeições do seu restaurante entregues ao domicílio

[Adicione o seu restaurante](#)



Entregue com o Click&Eat

[Registre-se para fazer as entregas](#)

Figura 34 - aceder pagina registar restaurante

Relatório 1º Milestone

Após clicar nesse botão será redirecionado para a seguinte página, que pode ser observada na imagem abaixo. Onde nela para adicionar um restaurante é necessário preencher todos os dados, exceto os campos Sigla e Descrição, e introduzir uma foto para o restaurante, após isso é só clicar no botão de guardar. Caso um dos dados esteja mal introduzido o utilizador é redirecionado de novo para a página, para poder corrigir o problema. Após todos os dados obrigatórios estarem corretamente introduzidos e o utilizador clicar no botão guardar ele terá de aguardar um dos admin do site aceitar o seu restaurante para que ele possa ser adicionado ao site. Uma imagem da página para criação de um restaurante pode ser observada na imagem abaixo.

The screenshot shows a dark-themed web page titled 'Adicionar novo Restaurante'. The form contains the following fields:

- Nome:** Input field placeholder: 'Introduza o nome do restaurante'
- Sigla:** Input field placeholder: 'Introduza a sigla do restaurante'
- NIF:** Input field placeholder: 'Introduza o NIF'
- Telefone:** Input field placeholder: 'Introduza o número telefónico'
- Email:** Input field placeholder: 'Introduza o email do restaurante'
- Password:** Input field placeholder: 'Introduza a password do restaurante'
- Confirmar Password:** Input field placeholder: 'Introduza a password novamente'
- Morada:** Input field placeholder: 'Introduza a sua morada'
- Código postal:** Input field placeholder: 'Ex: 1234-567'
- Cidade:** Input field placeholder: 'Introduza o nome da sua cidade'
- Descrição:** Input field placeholder: 'Introduza uma descrição'
- Foto do restaurante:** A button labeled 'Escolher ficheiro' and a message 'Nenhum ficheiro selecionado'.
- Action Buttons:** Three buttons at the bottom: 'Guardar' (highlighted in blue), 'Limpar', and 'Voltar'.

Figura 35 - Imagem para criação de um restaurante

NavBars

Para o nosso site nos decidimos implementar um sistema de navBar de maneira a que o utilizador tenha acesso a algumas páginas de maneira rápida, onde para isso nos desenvolvemos 3 navBars que são utilizadas em partes distintas do nosso site, sendo elas:

- NavBar
- NavBar Restaurante
- Nav Bar do admin

NavBar

Esta NavBar ela vai estar disponível desde a homePage até há pagina dos restaurantes. A primeira NavBar que mostramos abaixo acontece quando o utilizador se encontra na homePage. Quando o utilizador não estiver autenticado ela terá o mesmo aspeto da imagem abaixo.



Figura 36 - NavBar base

Nesse formato da navbar o utilizador só poderá aceder a duas páginas, a página de Sign Up, para criar uma conta e a página de Login, para poder iniciar sessão na sua conta.

Quando o utilizador está autenticado com prioridade de cliente, dono ou admin, a NavBar já contará com a opção perfil, que permite ao utilizador aceder há página do seu perfil (de momento só está funcional o perfil de admin), a opção carrinho que redireciona o utilizador ao seu carrinho de compras (que de momento também não está funcional) e a opção de Logout que faz com que o utilizador termine a sua sessão atual. Na imagem abaixo é possível observar essa NavBar.



Figura 37 - NavBar autenticado

Na imagem abaixo é possível observar a NavBar quando uma conta do tipo Restaurante está autenticado, onde nesse caso ao clicar no botão perfil o utilizador será redirecionado, para a homePage do restaurante em questão.

Relatório 1º Milestone



Figura 38 - NavBar aut. Restaurante

Agora quando o utilizador ele vai para uma pagina diferente da homePage a navBar ela mantem o mesmo formato das restantes apresentadas, a única diferença é que ela possui novos botões que são:

- Voltar, que permite o utilizador voltar há página anterior;
- Main Page, que ao clicar no mesmo o utilizador é redirecionado para a pagina principal



Figura 39 - NavBar fora da homePage

Uma coisa em comum em todas as navBar apresentadas é que no lado esquerda tem a logotipo do site e o nome do mesmo.

NavBar do Restaurante

A nav Bar dos restaurantes, é igual à navBar falada no tópico anterior a única diferença é que ela no lado esquerdo em vez de mostrar o nome e logotipo do site mostra o nome e logotipo do restaurante que o utilizador selecionou. Esta navBar estará presente ao utilizador em todas as páginas dentro de cada página do restaurante.



Figura 40 - Nav Bar do restaurante

Relatório 1º Milestone

NavBar do Admin

Esta NavBar ela vai estar presente apenas na página do perfil do admin que facilita ao admin, gerir todo o conteúdo do site. A NavBar do admin ele possui vários botões, sendo eles:

- Voltar → permite ao utilizador voltar diretamente há homepage do site;
- Perfil → permite ao admin visualizar os dados do seu perfil;
- Restaurantes → permite ao admin aceder há página de gestão dos restaurantes;
- Utilizador → permite ao admin aceder há página de gestão de utilizadores;
- Categorias → permite ao admin aceder há página de gestão das categorias;
- Porções → permite ao admin aceder há página de gestão de porções;
- Log Out → permite ao admin fazer logout da sua conta;
- Apagar Conta → permite ao admin de maneira fácil apagar a sua conta.

A imagem abaixo mostra a NavBar do admin.

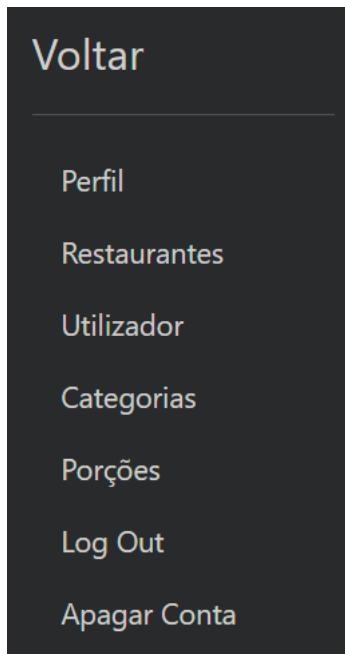


Figura 41 – navbarAdmin

Login

Na imagem abaixo é possível observar a nossa página de login, que permite que o utilizador autentique a sua conta no site.

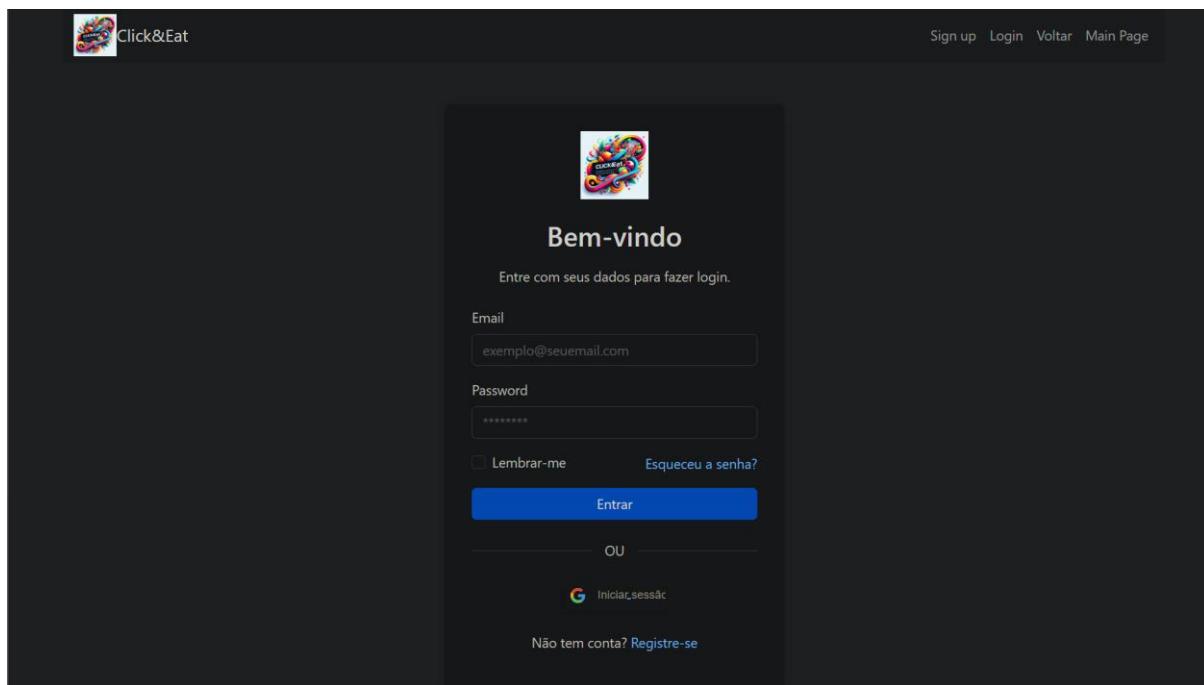


Figura 42 - Página de login

Para o utilizador realizar o login é obrigatório que o mesmo tenha uma conta no site, onde caso não tenha ele poderá clicar na navBar na opção “Sign up”, ou na pagina no login na hiperligação “Registre-se”. Após o utilizador criar a sua conta, para realizar o login ele necessita de introduzir o email e a sua password, caso ele queira que o seu login esteja ativo durante 30 dias ele pode clicar na opção “Lembrar-me” fazendo assim com que nesse período ele não necessite de fazer login para se autenticar no site. Por fim após o utilizador preencher todos esses campos é só ele clicar no botão “Entrar”, para poder se autenticar no site, se o email e password introduzidos forem válidos.

Relatório 1º Milestone

SignUp

Na imagem abaixo é possível observar a nossa página de Sign Up, que permite ao utilizador criar a sua conta no nosso site.

The screenshot shows the 'Sign up' page for the Click&Eat website. At the top, there is a logo of a colorful donut and the text 'Click&Eat'. To the right are links for 'Sign up', 'Login', 'Voltar' (Back), and 'Main Page'. The main title 'Sign up' is centered above several input fields. Each field has a placeholder text and a descriptive label: 'First name' (placeholder: 'Input first name'), 'Last name' (placeholder: 'Input last name'), 'Email' (placeholder: 'example@gmail.com'), 'Telefone:' (placeholder: 'Introduza o número telefónico'), 'Password' (placeholder: 'Enter at least 8+ characters'), and 'Repeat password' (placeholder: 'Enter at least 8+ characters'). Below these fields is a checkbox labeled 'Para poder criar uma conta tem de concordar com Terms of Use & Privacy Policy'. A large blue button at the bottom left contains the text 'Sign up'. Below it, a link 'Já tem uma conta? Log in' is visible. In the center, the word 'OR' is flanked by horizontal lines. At the bottom center is a circular icon containing the Google 'G' logo, indicating a Google sign-in option.

Figura 43 - Página de SignUp

Para o utilizador poder criar a sua conta no nosso site ele precisa de preencher todos os campos apresentados na imagem, que são:

- First Name, o primeiro nome do utilizador;
- Last Name, o ultimo nome do utilizador;
- Email, o email da conta do utilizador, que não pode ser igual ao email de outra conta já existente;

Relatório 1º Milestone

- Telefone, o número de telefone do utilizador, que não pode ser igual ao número de telefone de outra conta já existente;
- Password, que é a password da conta do utilizador;
- Repeat Password, que tem de ser igual há password anteriormente introduzida;
- Termos e Condições, onde antes do utilizador poder criar a conta ele necessita de aceitar com os mesmos.

Por fim quando o utilizador preencher todos os campos e aceitar com os termos e Condições do site ele só precisa de clicar no botão Sign up, que a sua conta será criada como sendo a conta de um cliente, e será redirecionada para a página de login.

Admin Page

A página de admin, ela só está disponível para os utilizadores que são do tipo “admin”. Ela possui uma navBar, que já foi falada no subtópico “NavBarAdmin” do tópico: “NavBar”. Esta página, ainda mostra os dados do admin, sendo o seu nome, número de telefone, email e a sua foto de perfil. A página ainda possui dois botões um para poder editar os dados do admin e outro para o admin poder alterar a sua password.

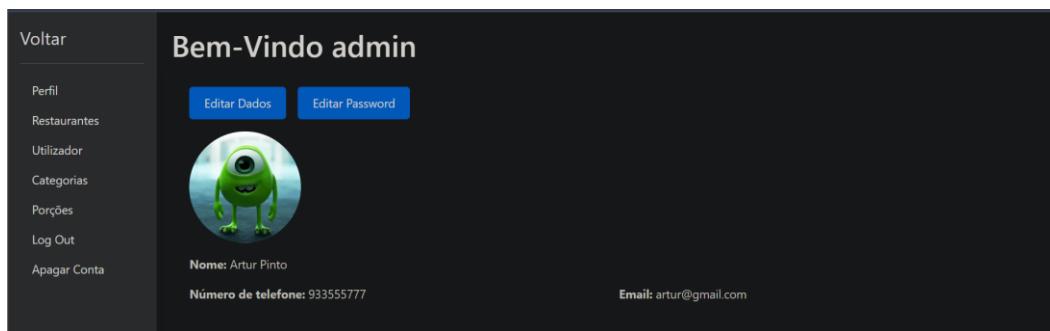


Figura 44 - Pagina do admin

Página Editar Dados

Está é a página, para editar os dados, onde na mesma os campos com os dados do admin, já se encontram preenchidos com os dados do mesmo. Para o admin alterar os seus dados, ele não deve deixar nenhum campo vazio, exceto o da foto do admin. Após o utilizador alterar os seus dados, corretamente, ele só necessita de clicar no botão “alterar” que será redirecionado para a pagina de perfil do admin.

Relatório 1º Milestone

Editar conta

First name
Arthur

Last name
Pinto

Email
arthur@gmail.com

Telefone:
93355777

Foto de perfil

Foto do admin:
Escolher ficheiro Nenhum ficheiro selecionado

Alterar

Voltar

Figura 45 - Página edição dados do admin

Página alterar password

Está é a página para alterar a password para alterar a password da conta do admin onde para alterar o mesmo tem de introduzir a sua password atual, onde caso não seja a mesma surgirá uma mensagem de erro e inserir e confirmar a nova password para o seu perfil, onde caso não sejam compatíveis surgirá uma mensagem de erro.

Click&Eat

Alterar Password

Password Atual:
Introduza a password atual

Nova Password:
Introduza a nova password

Confirmar Nova Password:
Confirme a nova password

Guardar

Limpar

Voltar

Figura 46 - Página alterar password admin

Listas de Restaurantes

Está é a página, com a lista de restaurantes do admin, onde ele consegue observar todos os restaurantes já aprovados. Nesta página ela ainda possui campos para a realização de filtragem e ordenação dos restaurantes sendo os campos de filtragem os seguintes:

- Nome de restaurante, permite pesquisar pelo nome de um restaurante;
- Cidade, permite pesquisar pelo nome da cidade dos restaurantes que o mesmo pretende consultar.

Já para ordenação possuímos os seguintes campos:

- Nome (A-Z), permite ordenar pelo nome dos restaurantes de maneira ascendente;
- Nome (Z-A), permite ordenar pelo nome dos restaurantes de maneira descendente.

Para o utilizador puder filtrar e/ou ordenar ele precisa preencher os campos que pretende e clicar no botão “Filtrar”, após filtrar se ele quiser apagar o filtro/ordenação que aplicou é só clicar no botão “Clear”.

Esta página ainda possui o botão adicionar que permite ao admin adicionar um novo restaurante, cujo design é igual ao da página de adição de um restaurante que o registo de restaurante e um botão que redireciona para a página com a lista dos restaurantes por aprovar, que só aparece caso existem restaurantes por ser aprovados que iremos falar no próximo tópico.

Cada restaurante da lista possui 3 botões, sendo eles:

- Remover, que permite ao admin remover um restaurante;
- Editar, que permite ao admin editar os dados de um restaurante, idêntico há pagina de edição do tópico “Página com todos os Restaurantes”;
- Mais Info, permite ao utilizador consultar todos os dados de um restaurante.

Relatório 1º Milestone

Figura 47 - Página lista de restaurantes

Página de Aprovação

Esta é a página com todos os restaurantes por aprovar, onde cada restaurante possui 3 botões:

- Aprovar, botão que aprova um restaurante;
- Rejeitar, botão que rejeita um restaurante;
- Mais Info, botão que mostra mais informação sobre aquele restaurante.

Quando o admin, acaba de aprovar ou rejeitar todos os restaurantes da lista ele é automaticamente redirecionado para a página com a lista de restaurantes.

Figura 48 – Página dos restaurantes por aprovar

Relatório 1º Milestone

Criar Restaurante

Está é a página, que é igual e tem um funcionamento igual à página de registo de um restaurante, que se encontra na HomePage, a única coisa que muda é que quando o admin, preenche todos os dados e clica em “Guardar”, o restaurante é aprovado automaticamente.

Adicionar novo Restaurante

Nome:
Introduza o nome do restaurante

Sigla:
Introduza a sigla do restaurante

NIF:
Introduza o NIF

Telefone:
Introduza o número telefónico

Email:
Introduza o email do restaurante

Password:
Introduza a password do restaurante

Confirmar Password:
Introduza a password novamente

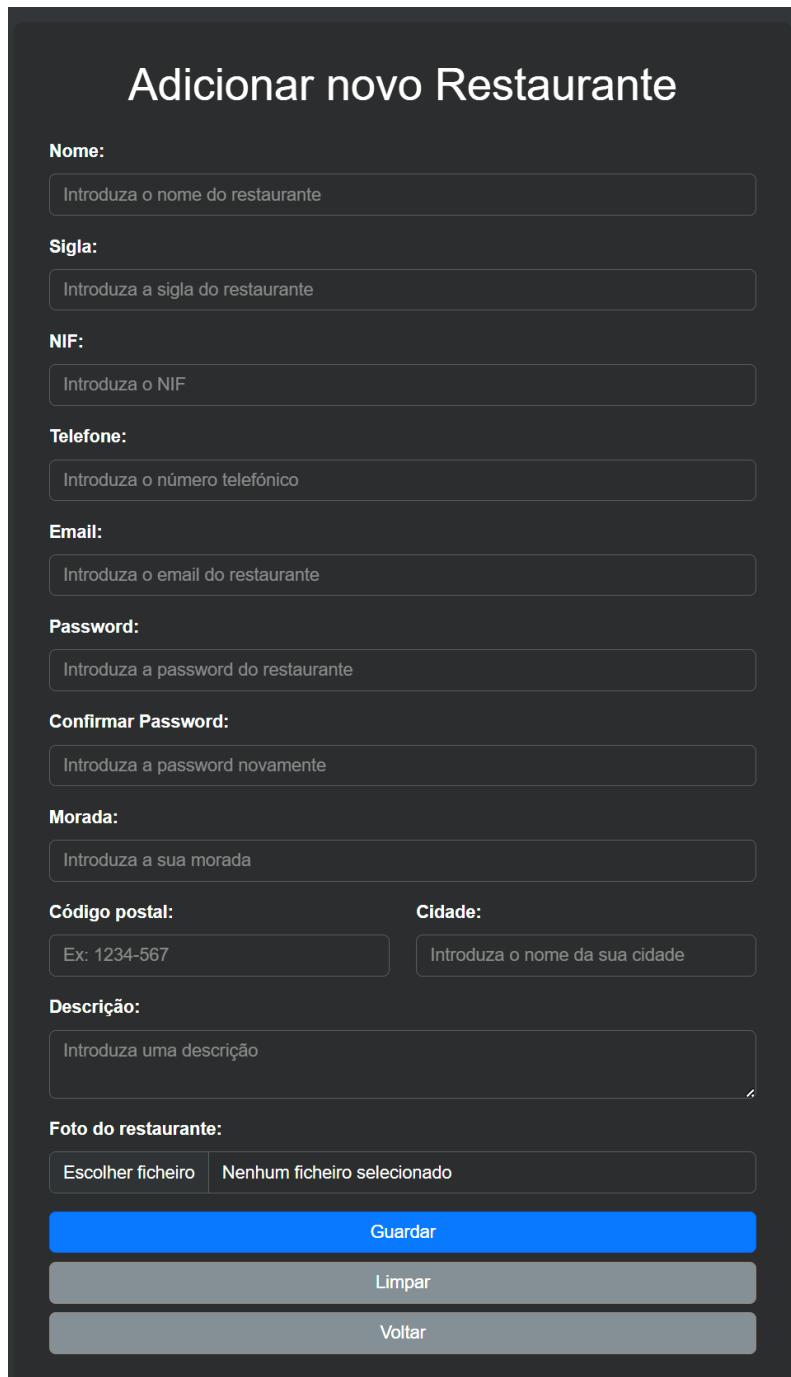
Morada:
Introduza a sua morada

Código postal: Ex: 1234-567 **Cidade:** Introduza o nome da sua cidade

Descrição:
Introduza uma descrição

Foto do restaurante:
Escolher ficheiro | Nenhum ficheiro selecionado

Guardar | **Limpar** | **Voltar**

A screenshot of a web-based form titled "Adicionar novo Restaurante". The form consists of several input fields and buttons. At the top, there's a title "Adicionar novo Restaurante". Below it, there are input fields for "Nome" (Name), "Sigla" (Abbreviation), "NIF" (Tax ID), "Telefone" (Phone), "Email", "Password", "Confirmar Password" (Confirm Password), "Morada" (Address), "Código postal" (Postal Code), and "Cidade" (City). Each input field has a placeholder text. At the bottom of the form, there are three buttons: "Guardar" (Save) in blue, "Limpar" (Clear) in grey, and "Voltar" (Back) in grey.

Relatório 1º Milestone

Info dos Restaurantes

Esta página ela apenas mostra todos os dados sobre o restaurante, como o nome, sigla, email, número de telefone, morada e descrição e demonstra, caso o restaurante tenha algum dono, os dados de todos os donos daquele restaurante, sendo eles o nome do dono, o email e o telefone.

The screenshot displays a mobile application interface for viewing a restaurant. At the top, the title "Visualizar Restaurante" is centered. Below it, the restaurant's details are listed in a vertical format:

- Nome:** McDonalds
- Sigla:** MC
- Email:** mcdonalds@gmail.com
- Telefone:** 756432123
- Morada:** rua do McDonalds, 4000-000, Guimaraes
- Descrição:** PARAPAPA

Below these details, a section titled "Donos do Restaurante" shows the information for one owner:

Total de donos: 1

Adriano Santos

Email: donomcdonalds@gmail.com

Telefone: 897987231

A large grey button at the bottom is labeled "Voltar".

Figura 49 - Página com informação dos restaurantes

Lista de Utilizadores

Está é a lista de utilizadores, que mostra todos os utilizadores do site. Esta página possui filtros e ordenações para facilitar a pesquisa do admin por um utilizador ou um conjunto de utilizadores, possuindo os seguintes filtros:

- First Name, que permite pesquisar pelo primeiro nome dos utilizadores;
- Last Name, que permite pesquisar pelo último nome dos utilizadores;
- Prioridade, que permite pesquisar pelos utilizadores de uma determinada prioridade (cliente, dono, admin ou todas);
- Banido, que permite pesquisar pelos utilizadores banidos, não banidos ou todos.

Já para ordenar é possível o admin ordenar por:

- Nome (A-Z), permite ordenar pelo nome dos restaurantes de maneira ascendente;
- Nome (Z-A), permite ordenar pelo nome dos restaurantes de maneira descendente;
- Prioridade (A-Z), que mostra as prioridades ordenadas ascendentemente;
- Prioridade (Z-A), que mostra as prioridades ordenadas descendetemente;
- Desbanned, que mostra primeiro os utilizadores desbanidos e depois os banidos;
- Banned, que mostra primeiro os utilizadores banidos e depois os desbanidos.

Para o utilizador puder filtrar e/ou ordenar ele precisa preencher os campos que pretende e clicar no botão “Filtrar”, após filtrar se ele quiser apagar o filtro/ordenação que aplicou é só clicar no botão “Clear”.

Esta página ainda possui um botão para adicionar um novo utilizador, que será explicado no próximo tópico.

Cada utilizador possui 4 botões, sendo eles:

- Remover, que elimina o utilizador;
- Editar, que é semelhante à página de adicionar um utilizador, só que desta vez os campos já estão preenchidos com os dados do utilizador, para que eles possam ser editados;

Relatório 1º Milestone

- Desbanir/Banir, onde o botão desbanir quando o utilizador está banido e permite desbanir o utilizador e o botão banir aparece quando o utilizador está desbanido e permite banir o mesmo.
- Mais info, que mostra os dados de um utilizador específico.

The screenshot shows a user management interface titled "Lista de Utilizadores". On the left, there's a sidebar with links: Voltar, Perfil, Restaurantes, Utilizador (selected), Categorias, Porções, Log Out, and Apagar Conta. The main area has search fields for First name and Last name, and dropdowns for Prioridade (Qualquer), Banido? (Todos), and Ordenação (- Selecione uma Opção). Below these are two buttons: "Filtrar" and "Clear". The user list is displayed in a grid of six cards:

- Rodrigo Filipe**
Prioridade: Cliente
Email: rodrigo@gmail.com
Telefone: 777888999
Buttons: Remover, Editar, Desbanir
- Fernando Agustinho**
Prioridade: Cliente
Email: fernando@gmail.com
Telefone: 222333444
Buttons: Remover, Editar, Banir
- Jose Pinto**
Prioridade: Cliente
Email: josepinto@gmail.com
Telefone: 999888111
Buttons: Remover, Editar, Banir
- Maria Albertina**
Prioridade: Cliente
Email: mariaalbertina@gmail.com
Telefone: 888999111
Buttons: Remover, Editar, Desbanir
- Funcionario teste**
Prioridade: Funcionario
Email: funcionario1@gmail.com
Telefone: 888222111
Buttons: Remover, Editar, Banir
- Ze esquina**
Prioridade: Dono
Email: zeesquina123@gmail.com
Telefone: 666777888
Buttons: Remover, Editar, Banir

Figura 50 - Lista de Utilizadores

Relatório 1º Milestone

Adicionar Utilizador

Esta é a página, para criar um novo utilizador onde o admin precisa obrigatoriamente de preencher cada um dos seguintes campos:

- First Name, primeiro nome do utilizador;
- Last Name, último nome do utilizador;
- Email, email do utilizador, que será o email do perfil;
- Telefone, número de telefone do utilizador;
- Password, password da conta do utilizador;
- Repeat Password, repetição da password;
- Prioridade, prioridade do utilizador, que pode ser cliente, dono ou admin;

Após o preenchimento correto de todos os campos é só o admin clicar em criar, que o utilizador será criado e de seguida o mesmo será redirecionado para a lista de utilizadores.

Criar conta

First name
Input first name

Last name
Input last name

Email
example@gmail.com

Telefone:
Introduza o número telefónico

Password
Enter at least 8+ characters

Repeat password
Enter at least 8+ characters

Prioridade
Cliente

Criar

Voltar

Limpar

Figura 51 - Página adicionar utilizador

Relatório 1º Milestone

Caso o campo prioridade seja dono, aparecerá um novo campo abaixo onde o admin terá de inserir o nome do restaurante cujo utilizador é dono, sendo que ainda possui o botão adicionar restaurante, que permite adicionar mais um campo para introduzir o nome do restaurante, permitindo assim que um utilizador, seja dono de vários restaurantes. Ao lado de cada um desse campo existe um botão “Remover”, que permite remover um desses campos.

The screenshot shows a user interface element. At the top, there is a label "Prioridade" followed by a blue-outlined input field containing the text "Dono". Below this is a section labeled "Nome do restaurante" with a text input field containing the placeholder "Introduza o nome do restaurante". To the right of this input field is a red-bordered button labeled "Remover". At the bottom left is a blue-bordered button labeled "Adicionar restaurante".

Figura 52 - Prioridade Dono

Relatório 1º Milestone

Mais Info

Esta é a página que mostra todos os dados do utilizador, primeiro nome, o ultimo, email e telefone, sendo que os campos abaixo “Restaurantes do Utilizador”, só aparece caso o utilizador seja dono de pelo menos um restaurante, e mostra o nome dos restaurantes, a sigla, email, número de telefone, morada e descrição.

The screenshot shows a user profile page titled "Visualizar Perfil". It contains fields for First name (Adriano), Last name (Santos), Email (donomcdonalds@gmail.com), and Telephone (897987231). Below this, a section titled "Restaurantes do utilizador" displays information for one restaurant: McDonalds. The restaurant's details include Sigla (MC), Email (mcdonalds@gmail.com), Telephone (756432123), Morada (rua do McDonalds, 4000-000, Guimaraes), and Descrição (PARAPAPA). A "Voltar" (Back) button is at the bottom.

| First name | Adriano |
|----------------------------|---------------------------------------|
| Last name | Santos |
| Email | donomcdonalds@gmail.com |
| Telephone | 897987231 |
| Restaurantes do utilizador | |
| Total de restaurantes: | 1 |
| McDonalds | |
| Sigla | MC |
| Email | mcdonalds@gmail.com |
| Telephone | 756432123 |
| Morada | rua do McDonalds, 4000-000, Guimaraes |
| Descrição | PARAPAPA |

Figura 53 - Mais Info utilizador

Relatório 1º Milestone

Lista de Categorias

Esta página é a lista de categorias, que mostra todas as categorias que um prato e/ou menu pode ter. Nesta página o admin pode utilizar um filtro, que lhe permite pesquisar pelo nome de uma categoria. E um sistema de ordenação que possui os seguintes critérios de ordenação:

- Nome (A-Z), permite ordenar pelo nome das categorias de maneira ascendente;
- Nome (Z-A), permite ordenar pelo nome das categorias de maneira descendente.

Para o utilizador puder filtrar e/ou ordenar ele precisa preencher os campos que pretende e clicar no botão “Filtrar”, após filtrar se ele quiser apagar o filtro/ordenação que aplicou é só clicar no botão “Clear”.

Esta página possui um botão para adicionar uma nova categoria.

Cada categoria possui dois botões, sendo eles:

- Remover, que permite remover uma categoria;
- Editar, que permite editar o nome da categoria, e possui um visual igual ao adicionar categoria, mas os campos já estão preenchidos com os dados da categoria;

The screenshot shows the 'Lista de Categorias' (Category List) page. The interface is in Portuguese. At the top right, there is a green 'Adicionar Categoria' (Add Category) button. Below it are two input fields: 'Categoria:' with placeholder 'Introduza o nome da categoria' (Enter category name) and 'Ordenação:' with placeholder '- Selecione uma Opção -' (Select an option). Below these are two buttons: 'Filtrar' (Filter) and 'Clear'. The main content area displays a grid of categories in boxes. The categories shown are Peixe, Vegan, Dieta, Prato do dia, Alternativo, Carne, and Hamburguer. Each category box contains its name, a 'Remover' (Remove) button, and an 'Editar' (Edit) button. On the far left, there is a vertical sidebar with navigation links: Voltar, Perfil, Restaurantes, Utilizador, Categorias, Porções, Log Out, and Apagar Conta.

Figura 54 - Lista de categorias

Relatório 1º Milestone

Adicionar Categoria

Esta é a página, para criar uma nova categoria, onde o admin só necessita de meter o nome da categoria, onde o nome tem de ser uma categoria que ainda não existe. Após introduzir o nome é só o utilizador clicar em “Guardar”.

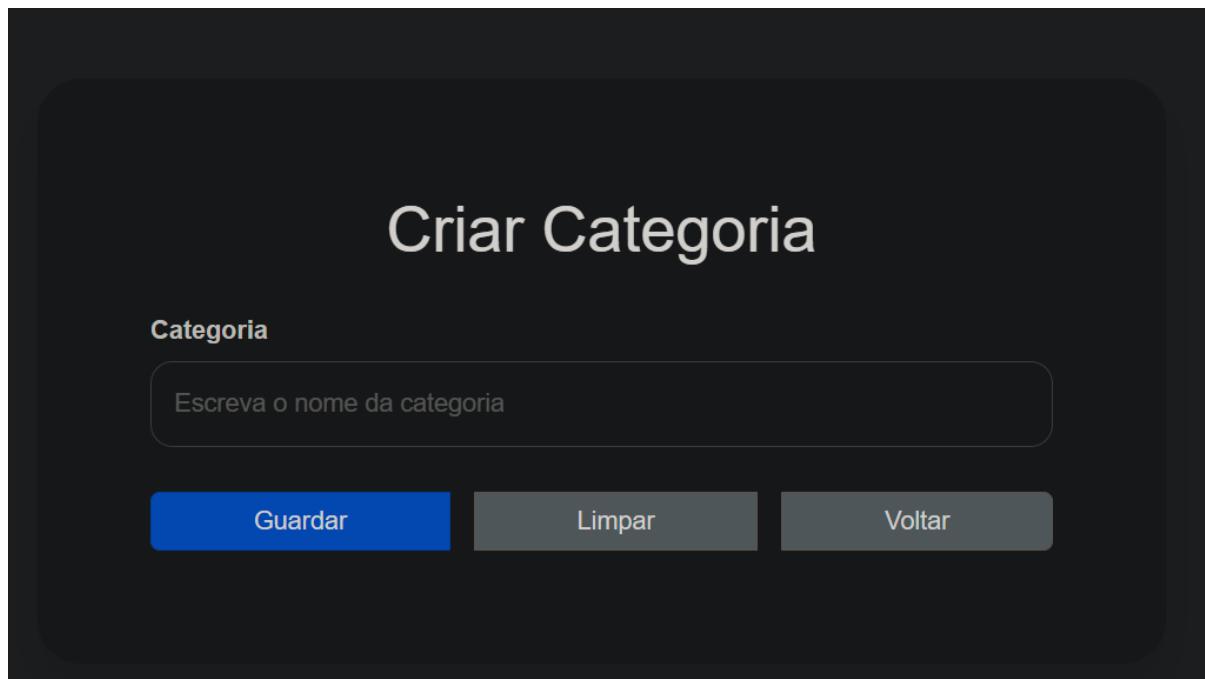


Figura 55 - Página para criar uma categoria

Relatório 1º Milestone

Lista de Porções

Esta página é a lista de porções, que mostra todas as porções que um prato pode ter. Nesta página o admin pode utilizar um filtro, que lhe permite pesquisar pelo nome de uma porção. E um sistema de ordenação que possui os seguintes critérios de ordenação:

- Nome (A-Z), permite ordenar pelo nome das porções de maneira ascendente;
- Nome (Z-A), permite ordenar pelo nome das porções de maneira descendente.

Para o utilizador puder filtrar e/ou ordenar ele precisa preencher os campos que pretende e clicar no botão “Filtrar”, após filtrar se ele quiser apagar o filtro/ordenação que aplicou é só clicar no botão “Clear”.

Esta página possui um botão para adicionar uma nova categoria.

Cada categoria possui dois botões, sendo eles:

- Remover, que permite remover uma porção;
- Editar, que permite editar o nome da porção, e possui um visual igual ao adicionar porção, mas os campos já estão preenchidos com os dados da porção;

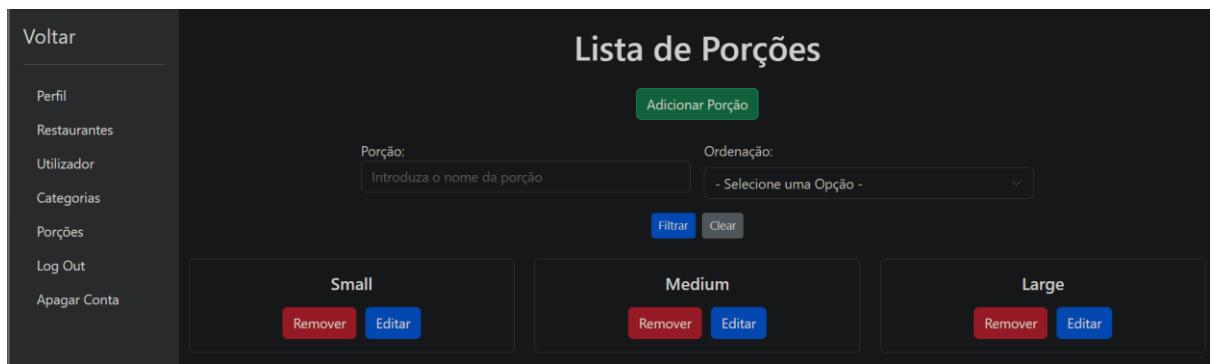
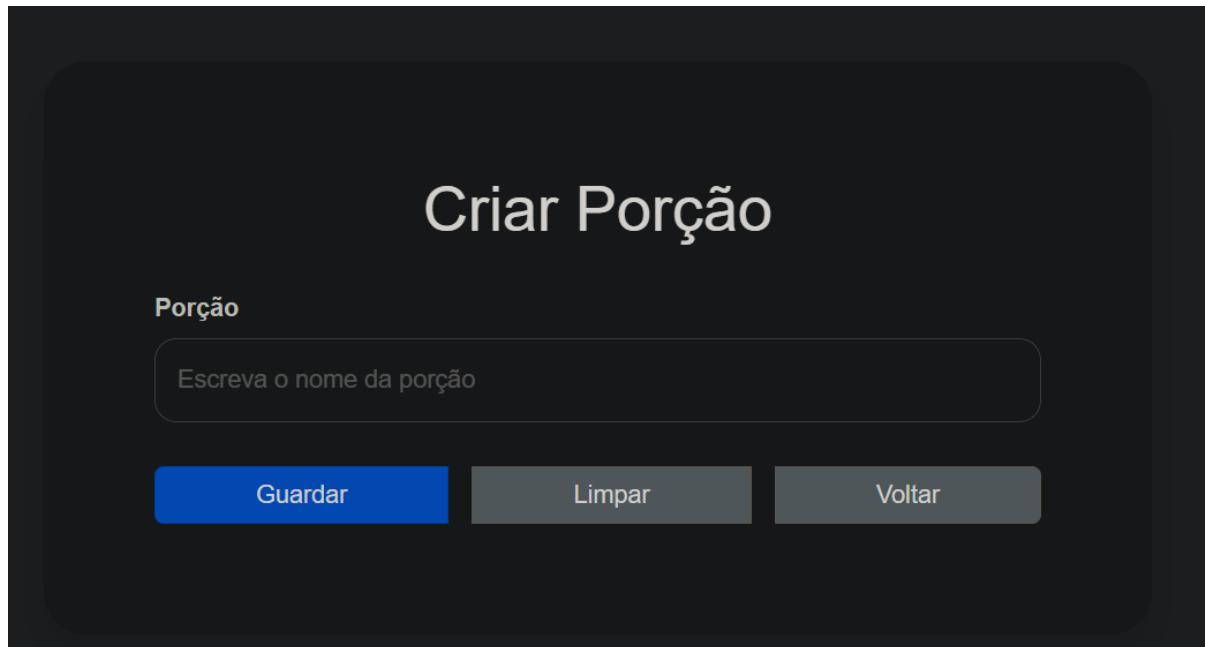


Figura 56 - Lista de Porções

Relatório 1º Milestone

Adicionar Porção

Esta é a página, para criar uma nova porção, onde o admin só necessita de meter o nome da porção, onde o nome tem de ser uma porção que ainda não existe. Após introduzir o nome é só o utilizador clicar em “Guardar”.



Página com todos os Restaurantes

Esta é a pagina, que permite ao utilizador, visualizar todos os restaurantes, aprovados, disponíveis no site. Nesta página estarão todos os restaurantes que o utilizador pode visitar para fazer uma encomenda. Onde para entrar na pagina do site ele terá apenas de clicar no botão “Mais Info”. Para facilitar a pesquisa de um restaurante por parte do utilizador o mesmo terá à sua disposição um conjunto de filtros e um mecanismo de ordenação de restaurantes, que possui os filtros:

- Nome de restaurante, permite pesquisar pelo nome de um restaurante;
- Cidade, permite pesquisar pelo nome da cidade dos restaurantes que o mesmo pretende consultar.

Já para ordenação possuímos os seguintes campos:

- Nome (A-Z), permite ordenar pelo nome dos restaurantes de maneira ascendente;
- Nome (Z-A), permite ordenar pelo nome dos restaurantes de maneira descendente.

Para o utilizador puder filtrar e/ou ordenar ele precisa preencher os campos que pretende e clicar no botão “Filtrar”, após filtrar se ele quiser apagar o filtro/ordenação que aplicou é só clicar no botão “Clear”.

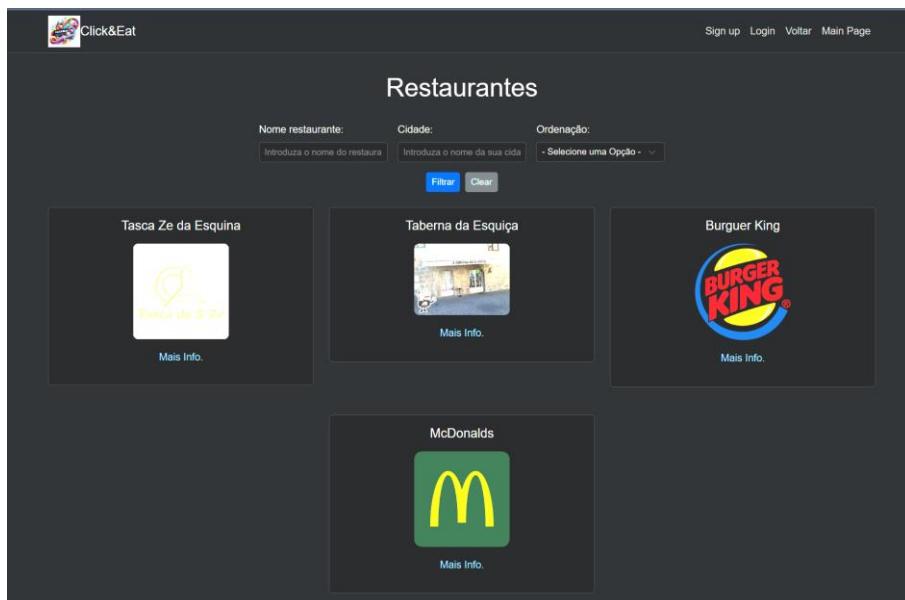


Figura 57 - Página dos restaurantes

Relatório 1º Milestone

Caso o utilizador esteja autenticado e seja do tipo admin esta página, terá um visual diferente, podendo observá-lo na imagem abaixo.

The screenshot shows the Click&Eat application's restaurant management page. At the top, there is a logo for 'Click&Eat' and a navigation bar with links for 'Perfil', 'Carrinho', 'Log Out', 'Voltar', and 'Main Page'. The main title is 'Restaurantes' with a 'Adicionar Restaurante' button. Below this, there are search fields for 'Nome restaurante', 'Cidade', and 'Ordenação', along with 'Filtrar' and 'Clear' buttons. The page displays four restaurant entries in cards:

- Tasca Ze da Esquina**: Shows a logo with a stylized 'Z' and 'E'. Buttons: 'Remover' (red), 'Editar' (blue), and 'Mais Info.'.
- Taberna da Esquiça**: Shows a thumbnail image of a building. Buttons: 'Remover' (red), 'Editar' (blue), and 'Mais Info.'
- Burguer King**: Shows the Burger King logo. Buttons: 'Remover' (red), 'Editar' (blue), and 'Mais Info.'
- McDonalds**: Shows the McDonald's golden arches logo. Buttons: 'Remover' (red), 'Editar' (blue), and 'Mais Info.'

Figura 58 - Lista dos restaurantes aut. admin

Quando o user é um admin, a página ela ganha um botão para adicionar restaurantes igual ao que tem na página do admin, e cada restaurante ganha mais dois botões, sendo eles:

- Remover, que permite remover um restaurante;
- Editar, que permite editar os dados de um restaurante.

Relatório 1º Milestone

Página do restaurante

Na página principal de um restaurante é possível observar três campos principais, que variam conforme o tipo de utilizador que acede a página, os dados do restaurante, as avaliações que o mesmo tem, e os menus que o mesmo possui.

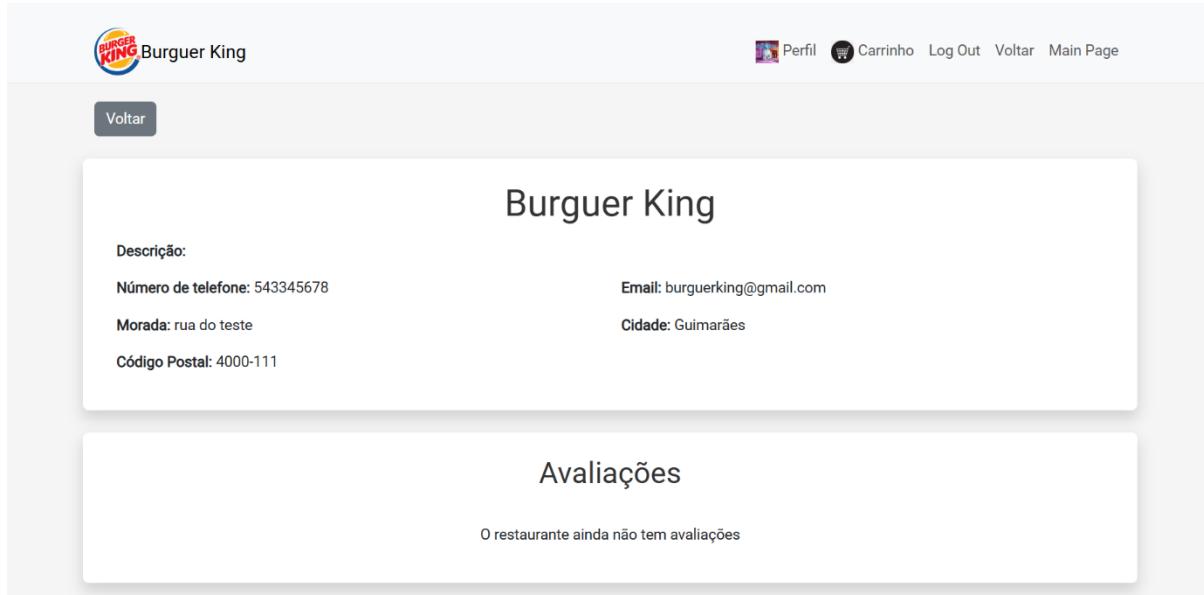


Figura 59 - Informação do restaurante

Caso o utilizador seja um dos donos do restaurante ou um admin, aparecerá na informação do restaurante um botão, que permitirá ao mesmo editar os dados do restaurante, como é possível observar na imagem abaixo.

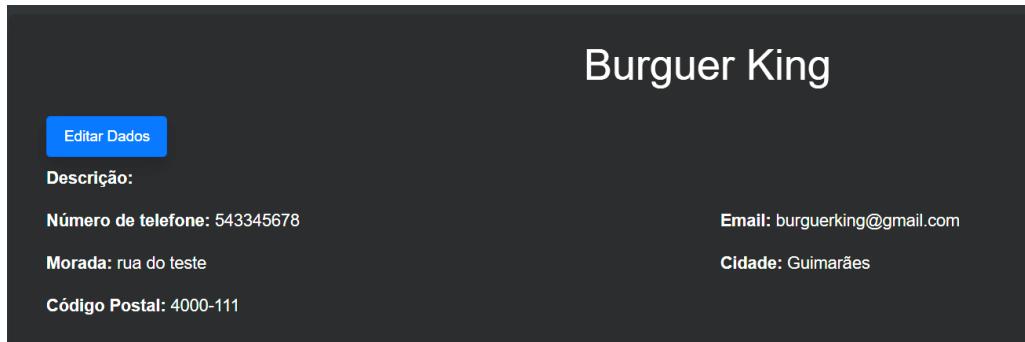


Figura 60 - Editar Restaurante

Como é possível observar na imagem a seguir, é possível o filtro nos menus, por nome, tipo do Menu e é até possível ordenar os menus

Relatório 1º Milestone

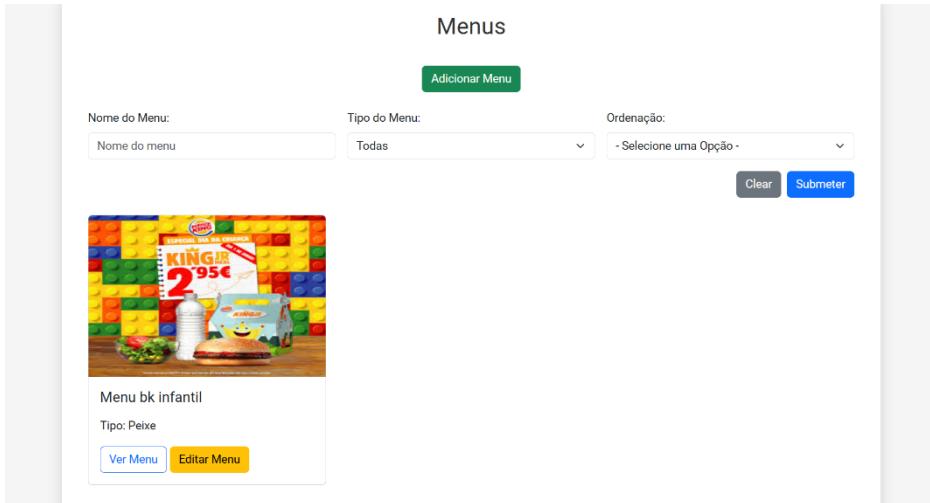


Figura 61 - menus do restaurante

Caso o utilizador seja do tipo restaurante, esta página terá apenas um botão a mais que redirecionara para a página de gestão dos pedidos caso o mesmo seja selecionado, caso o utilizador seja do tipo Dono, e o id de um dos restaurantes associados ao dono seja o do restaurante que o mesmo acedeu, para além de ver o botão da gestão de pedidos, também irá ver o botão para criar um menu e um para editar cada menu individualmente, para os restantes tipos de utilizadores, estas opções não estarão disponíveis.

O CRUD do menu e dos pratos também é assegurado aqui, ao selecionar o botão criar menu o dono do restaurante é direcionado a página criar menu, aonde tem um formulário para preencher para a criação do menu

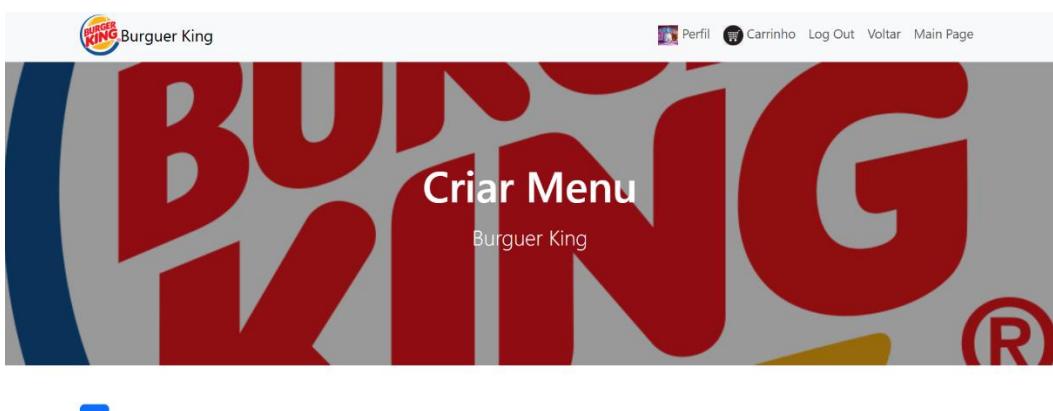


Figura 62 - Página criação dos menus

Relatório 1º Milestone

Nome do Menu:

Foto do Menu:

Tipo do Menu:

Pratos

Prato 1

Nome:

Descrição:

Categoria:

Preço:

Incidências:

Figura 63 - 2º img. Página Criação menu

Ingredientes:

Por Nome ▾ Nome do ingrediente

Apenas um ingrediente por caixa

Porções:

Selecionar uma porção...

Foto do Prato:

Figura 64 - 3º img. Página Criação menu

Caso o Utilizador deseje informar dos dados calóricos do prato é possível, é somente preciso por os ingredientes e os dados serão assegurados pela API openFoodFacts, entretanto como esta API é extremamente lenta e imprecisa, implementamos um sistema de cache, que armazena os dados pedidos por uma hora, e criamos uma página para validação dos dados obtidos, como é possível ver na imagem seguinte

Relatório 1º Milestone

Confirme os Dados Nutricionais

asjaodad

Produto sugerido pela API: Horno Al punto de sal

| Nutriente | Per 100g |
|------------------|----------|
| Calorias | 442 |
| Proteína (g) | 5.9 |
| Gordura (g) | 13 |
| Carboidratos (g) | 73 |
| Açúcares (g) | 5.3 |

[Confirmar e Continuar](#) [Re-pesquisar](#) [Editar Manualmente](#)

[Salvar Menu](#)

Figura 65 - Dados Nutricionais do menu

Na página para editar o menu, é possível editar todos os campos que foram preenchidos no formulário da página de criar, adicionar mais pratos ou remover já existentes, e até mesmo apagar o menu por completo.

Informações do Menu

Nome do Menu:

bk infantil



Alterar foto do Menu:

Choose File No file chosen

Tipo do Menu:

Peixe

Figura 66 - Imformação do menu

Relatório 1º Milestone

Pratos Atuais

Nome do Prato:
nuggets



Alterar Imagem:
Choose File No file chosen

Descrição:
mmm

Categoria: Peixe Preço: 2

Porções:
Small 2.00 X

Figura 67 - Informação dos pratos

Adicionar Novos Pratos

Adicionar Prato

Salvar Alterações

Apagar Menu Completo

 Cancelar

Figura 68 - Adicionar novo prato

Na pagina para ver um menu, é possível ver os pratos que o mesmo tem, e utilizar filtros para procurar por pratos

Relatório 1º Milestone

DSANASDJ

Peixe pratos

| | | |
|---|--|--------------------------------------|
| Nome do Menu: | Tipo do Prato: | Preço: |
| <input type="text" value="Nome do prato"/> | <input type="text" value="Todos"/> ▼ | <input type="text" value="Preço"/> ▼ |
| Porção: | Ordenação: | |
| <input type="text" value="Todos"/> ▼ | <input type="text" value="- Selecione uma Opção -"/> ▼ | |
| Clear Filtrar | | |

Figura 69 - Filtros dos pratos



Figura 70 - Card de um prato

Páginas de Erro

Neste tópico iremos falar das páginas de erro, que criamos. Para quando ocorre um erro no site, causado pelo utilizador ou um erro interno.

Página de Erro 404

A página de erro 404, ela aparece quando o utilizador tenta aceder a uma página que não existe, sendo que quando isso acontece aparecerá a página que pode observar na imagem abaixo.

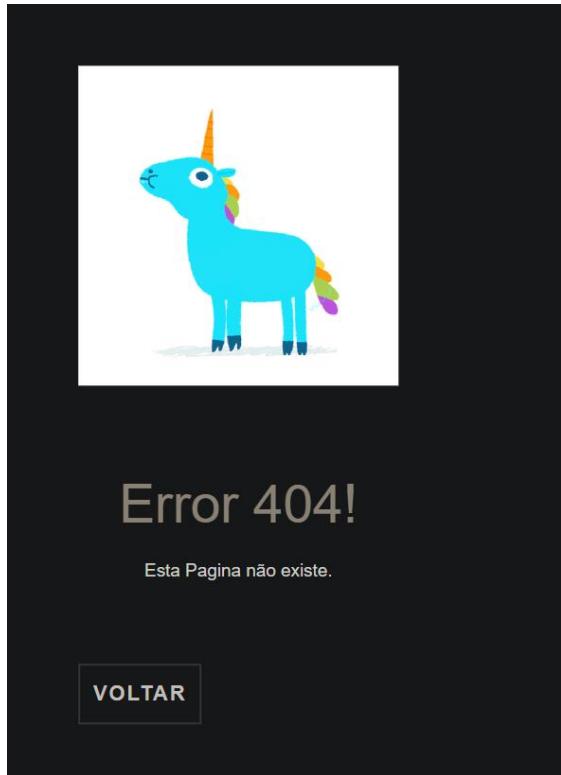


Figura 71 - Pagina error 404

O botão de “Voltar” da mesma, ele permite o utilizador voltar há página onde ele se encontrava.

Página de erro personalizada

Para as restantes páginas de erros, que existentes decidimos criar um ejs, que de acordo com o numero de Erro a mostrar e a mensagem ele criar uma pagina para o mesmo, ou seja, nos criamos uma pagina default que de acordo com o tipo de erro ela adapta a informação que precisa de mostrar para o mesmo, exemplos dessas paginas pode ser observado nas imagens abaixo.



Figura 72 - Pagina de erro personalizada para error 403

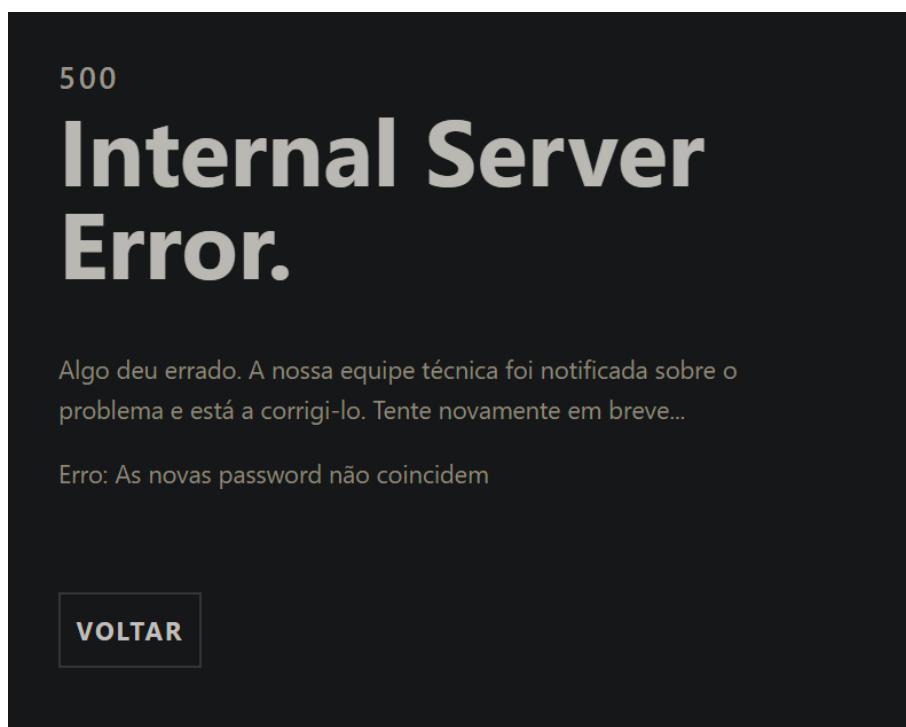


Figura 73 - Pagina de erro personalizado para o error 500

O botão de “Voltar” da mesma, ele permite o utilizador voltar há página onde ele se encontrava.

Conteúdo a melhorar

Nesta milestone conseguimos realizar todos os pontos obrigatórios e faltou-nos apenas dos opcionais o opcional para o restaurante poder registar encomendas, onde esse é um ponto que iremos finalizar logo no início da próxima milestone, visto que já possuímos os ejs necessários para essa funcionalidade, faltando apenas o código js para a mesma. Uma coisa que nós pretendemos melhorar no nosso login e signup é meter com que o utilizador possa criar uma conta e autenticar-se com uma conta google, através da Api da mesma marca. Referente aos dados nutricionais, mesmo com a implementação de timeOut e do cache, não é sempre certo de que iremos receber a resposta ao pedido que fizemos, ou por demorar de mais a responder, ou por um pequeno instante sem rede, então, pretendemos implementar um mecanismo de retry, para tentar cobrir esses casos.

Conclusão

Nesta milestone, conseguimos concluir todos os campos obrigatórios que foram pedidos, ficando apenas por concluir dos pontos opcionais a funcionalidade para os restaurantes poderem registar encomendas e gerir as mesmas. Também reconhecemos que há pontos que poderiam ser melhor como cada dono na página dos restaurantes poder ter um botão no seu restaurante para poder editar os dados do mesmo.