+

**Técnico de Informática – Sistemas 2019/2022**

2021/2022

**Prova de Aptidão Profissional**

GESRES 1.0

**Vítor Pereira Albernaz**

Ponta Delgada, março de dois mil e vinte e dois



**Técnico de Informática – Sistemas 2019/2022**

2021/2022

**Prova de Aptidão Profissional**

GESRES 1.0

**Formando:** Vítor Albernaz

**Orientador:** Duarte Cota

Ponta Delgada, março de dois mil e vinte e dois

**AGRADECIMENTOS**

|  |
| --- |
| Primeiramente queria agradecer à instituição da Escola de Novas Tecnologias dos Açores pela oportunidade de tirar o meu curso de nível IV nesta instituição.  De seguida agradecer a todos os formadores e professores que pelo meu progresso formativo passaram. Da área técnica os formadores Jorge Soares, Jorge Costa, Luís Carreira, Daniel Oliveira, Nuno Melo, Duarte Cota e Óscar Neto. Na área sociocultural e científica os professores/as Pedro Fonseca, Paulo Martinho, Mariana Silva, Henrique Medeiros, Renato Lameirão, Silvia Gouveia, Ana Pereira e Anabela Ferreira. Também a todo o trabalho do staff da escola, a dona Gabriela Viveiros, doutora Patrícia Ribeiro e ao doutor João Lima, que sempre fizeram o melhor possível pela escola.  Um especial obrigado ao formador Duarte Cota, que me ajudou sempre que precisava enquanto orientador da minha prova de aptidão profissional. Também à diretora de turma, Ana Pereira, por nos ajudar no processo de arranjar estágio e de preparar-nos para da melhor maneira para as nossas provas finais de curso.  Obrigado pela confiança! |

Assinatura

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Resumo:**

GESRES é um software principalmente de vendas num estabelecimento bar ou restaurante.

Este, por sua vez, exerce alguns extras, tais como: ligação com um programa para a cozinha, gestão de stock e de faturas em tempo real, e controlo de funcionários.

Para o desenvolvimento do mesmo foi utilizada as linguagens JavaFX e SQl, com técnicas de trabalho ganhas em sala de aula no decorrer do curso.

//Resultados finais

//conclusão

Índice geral:

[Índice geral: 6](#_Toc93852039)

[Índice de Imagens: 7](#_Toc93852040)

[Introdução: 8](#_Toc93852041)

[Objetivos: 8](#_Toc93852042)

[Sentido e importância: 8](#_Toc93852043)

[Escolha: 8](#_Toc93852044)

[Metodologia: 8](#_Toc93852045)

[Desenvolvimento: 9](#_Toc93852046)

[Vista inicial- login 9](#_Toc93852047)

[Dashboard- Admin 10](#_Toc93852048)

[Dashboard- Funcionário 15](#_Toc93852049)

Índice de Imagens:

[Imagem 1 Modelo MVC 8](#_Toc93769155)

Introdução:

GESRES é programa de vendas e gestão de um estabelecimento de Bar e ou Restauração.

Objetivos:

Este projeto tem como objetivos: a aplicação de conhecimentos obtidos nas aulas e/ou pesquisas na área da programação com interfaces gráficas, e gestão de bases de dados. Construir uma ferramenta de fácil utilização e aprendizagem da mesma. Também permitir que a mesma consiga gerir pedidos entre o setor de Bar e Cozinha, integrar um sistema de gestão de stock através dos pedidos dos clientes, e gerir a origem dos pedidos (estabelecimento, takeaway ou entregas ao domicílio, se existirem).

Sentido e importância:

Um programa desses tem muita importância no mundo dos negócios, pois pode facilitar muito o trabalho dos empregados e até mesmo do administrador do estabelecimento.

Escolha:

Escolhi um projeto deste calibre pois sempre foi do meu interesse como era gerido um estabelecimento deste tipo.

Metodologia:

Para o desenvolvimento do projeto foi utilizado um método de arquitetura MVC (“Model-View-Controller”) utilizando as linguagens Java e FXML (JavaFx).

MVC é um padrão de aquitetura de sofware responsavel por contribuir para a otimização da velocicade de resposta aos pedidos do utilizador.

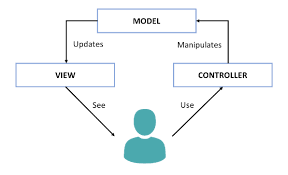


Imagem Modelo MVC

Desenvolvimento:

Começando por desenhar o programa, que inicialmente ia ser um programa com três dahsboards, “Admin”, “Funcionário” e “Cozinha, porem decidi separar a dashboard da cozinha para um programa separado. Realizei as vistas do programa com a ajuda do SceneBuilder, ferramenta que gerou o código FXML para uma interface gráfica no meu programa. Ao todo foram criadas quinze vistas para o programa:

-Uma de login;

-Sete dedicadas à dashboard de “Admin”;

-Sete dedicadas à dashboard de “Funcionário”;

-Uma para o programa da “Cozinha”;

Todas as vistas foram feitas para uma medida de aproximadamente, ou não exceder, as 15 polegadas, medida vulgarmente utilizada nos monitores de estabelecimentos do tipo bar/restaurante,

Vista inicial- login

Vista composta por uma ComboBox, onde os seus valores são os users disponíveis para o programa, no caso “Admin” e “Funcionário”. E uma PasswordField para colocar a respetiva palavra-passe.

//imagem do login

Dashboard- Admin

A principal de Admin foi elaborada para: adicionar novos funcionários e remover caso necessário, controlar a atividade dos mesmos, e nesta temos acesso a opções expostas no MenuBar para: Controlo de stock, receitas e faturas.

Logo ao entrar na dashboard, deparamo-nos com três TextFields, para o primeiro nome, último nome, e número de funcionário, um DatePicker para a data de nascimento, e uma ComboBox para selecionar o sexo do funcionário. Estes servem para adicionar um funcionário após clicar no botão “Adicionar”, que irá é preparado um objeto da classe “Funcionarios” para ser introduzido na base de dados.

Também tem uma tabela, onde apresentam alguns dados do funcionário: nome completo, atividade, número de funcionário. Esta é preenchida logo ao entrar na vista através do método “initialize()”, que preenche a mesma com os respetivos valores presentes na base de dados.

A atividade é possível ser mudada através de uma a ComboBox e do botão “Atualizar” abaixo da tabela, que atualiza na base de dados a atividade através de um querry SQl de “UPDATE”, e seguidamente atualiza a tabela limpado todos os dados e voltando a colocá-los.

No MenuBar presente há três menus: “STOCK” com dois submenus “Consultar Stock” e “Adicionar Produto”, “FICHA TÉCNICA” com um submenu “Receitas”, e “FATURAS” com três submenus “Diárias”, “Semanais” e “Mensais”.

//foto menu bar com opções todas abertas

No primeiro submenu “Consultar Stock”, ao clicar lá é aberta outra vista composta por uma tabela com o nome de cada produto e a respetiva quantidade em stock. Esta informação também é posta ao iniciar a vista através do método “initialize()”, que através de um querry SQl é apresentada a lista de produtos presente na base de dados sendo guardada num objeto da classe “Produtos”.

Também é possível editar o stock nessa vista, selecionando o produto que pretende editar o stock na tabela, quando é selecionado aparece o nome do produto na primeira TextField e fica outra TextField para o utilizador poder por a nova quantidade de stock. Após carregar no botão “Editar” é atualizada a informação da tabela limpado todos os dados e voltando a colocá-los.

//print vista editar stock

No segundo submenu “Adicionar Produto”, ao clicar lá é aberta outra vista composta por duas TextFields e duas ComboBoxs, estas para por o nome do produto a querer adicionar, o preço, selecionar o fornecedor e o tipo do produto, se é “Café”, “Sumos”,” Bebidas Alcoólicas”, “Cozinha”, “Snacks”, “Doces/Sobremesas” ou “Retalho”. Apos clicar no botão “Concluir” os dados que são recolhidos dos quatro elementos são guardados num objeto da classe “Produtos” e de seguida adicionados na base de dados.

//print vista addProduto

No terceiro submenu “Receitas”, ao clicar lá é aberta outra vista composta por uma tabela, com o nome de cada receita recolhidos da base de dados e guardados num objeto da classe “Receita” e mal a vista é aberta, através do método “initialize()”, é mostrado na tabela o nome de todas as recitas existentes na base de dados. Também tem três botões: “Consultar” que abre uma vista com uma tabela, após escolher a receita, com todos os ingredientes e respetivas quantidades da receita selecionada. “Adicionar” que abre outra vista com três TextFields, uma para por o nome da receita, uma para colocar o ingrediente outra para colocar a quantidade, um botão “Adicionar”, uma tabela para ir mostrando todos os produtos adicionados a um objeto da classe “Receita”. Após clicar no botão “Finalizar” é adicionada a receita na base de dados utilizando outro objeto criado.

// print vista Receitas

//falar sobre as vistas das faturas

//// print vista faturas

Dashboard- Funcionário

A dashboard funcionário é composta por um total de sete vistas diferentes. A primeira vista da mesma é composta por dez botões numerados de um a dez, correspondendo ao número das mesas do estabelecimento, ao clicar num dos botões é passado para a próxima vista, para mostrar os detalhes da mesa em questão, o numero da mesa correspondente à mesa. Quando a mesa está livre o botão fica da cor verde, já quando está ocupada fica vermelho. E no final um botão servindo para registar os pedidos de balcão, ou seja, para quem não precisar de usufruir de mesa, este quando selecionado abre a vista introduzir o código de funcionário.

//print funcViewMesas

A vista de detalhes da mesa, é composta por uma tabela, para que caso o pedido esteja ativo mostra na mesma todo o pedido. Também é composta quatro botões: “Novo Pedido” que quando selecionado abre a vista introduzir o código de funcionário, “Editar Pedido” que é aberta novamente a vista de realizar o pedido novo, mas com a função de acrescentar ao pedido antigo, uma forma arranjada para não realizar uma vista diferente, “Despachar Mesa” quer servirá para por o botão, que corresponde ao numero da mesa, novamente verde assinalando que a mesa está livre, e por ultimo o botão “Pagamento” que é apresentado outra vista com o total do pedido escrito numa TextField, outra TextField para colocar o contribuinte do cliente caso queira, e uma ComboBox para selecionar o método de pagamento, apos clicar no botar “Finalizar” o pedido é posto noutra tabela da base de dados “faturas” e a mesa fica automaticamente livre.

//print vistaDetalhesPedidos e pagamento

A vista de introduzir um código de utilizador é sempre mostrada antes de iniciar um novo pedido, de modo a ficar registado o funcionário que fez o pedido. Esta é composta por uma TextField, que serve para introduzir um número, e após clicar no botão concluir, o número é validado verificando de existe o número na base de dados. Caso não exista é lançado um alerta referindo que o código é inexistente. Se existir o código é aberta a vista para realizar o pedido com o nome do funcionário correspondente ao código.

//print vista código func