Olá, estudante.

A seguir, você dará continuidade ao desenvolvimento da sua solução através dos campos específicos para a resolução dos 3 desafios propostos, lembrando que eles se complementam.

**Nome: Felipe Ferrandin**

**RGM:** 29339502

**Documentação:**

Revisite a documentação do projeto desenvolvido na PIT I e insira a seguir toda a documentação atualizada e melhorada:

|  |
| --- |
| ID:FE001 |
| Título: Cadastro de usuário |
| Requerente:Cliente |
| **Comentários:** |
| Critérios de aceitação:  [CA#1] O usuário deve ser capaz de inserir seu nome, endereço de e-mail e senha válidos.  [CA#2] O sistema deve validar se o endereço de e-mail inserido já está cadastrado.  [CA#3] O sistema deve armazenar as informações do usuário no banco de dados.  [CA#4] O usuário deve ser capaz de fazer login após o cadastro. |
| Regras de negócio:  [RN #1] O usuário deve fornecer informações válidas durante o cadastro.  [RN #2] O usuário deve ter um endereço de e-mail exclusivo para cadastrar-se.  [RN #3] O usuário deve ser capaz de fazer login com as credenciais corretas.  [RN #4] As credenciais de login devem ser mantidas em segurança.  [RN #5] O usuário deve ser capaz de editar suas informações pessoais no perfil. |
| Requisito não funcional:  [RN F#1] O aplicativo deve ser responsivo e funcionar bem em diferentes tamanhos de tela.  [RN F#2] O aplicativo deve ser intuitivo e fácil de usar para usuários de diferentes idades e habilidades.  [RN F#3] O aplicativo deve garantir a segurança das informações dos usuários, incluindo senhas e informações pessoais. |
| Prioridade: 5 pontos  [ ] A  [ X ] B  [ ] C  [ ] D  [ ] E |
| Pontos de história: 3 pontos |

|  |
| --- |
| ID:FE002 |
| Título: Login de Usuário |
| Requerente:Cliente |
| **Comentários:** |
| Critérios de aceitação:  [CA#1] O usuário deve ser capaz de inserir seu endereço de e-mail e senha válidos.  [CA#2] O sistema deve validar se as informações inseridas estão corretas.  [CA#3] O usuário deve receber uma mensagem de erro se as informações inseridas estiverem incorretas. |
| Regras de negócio:  [RN #1] O usuário deve fornecer informações válidas durante o login.  [RN #2] O sistema deve validar as credenciais de login do usuário.  [RN #3] O sistema deve impedir o login de usuários que fornecerem credenciais inválidas.  [RN #4] As credenciais de login devem ser mantidas em segurança. |
| Requisito não funcional:  [RN F#1] O aplicativo deve ter um processo de login seguro para proteger as informações do usuário.  [RN F#2] O aplicativo deve ter uma interface intuitiva para facilitar o processo de login.  [RN F#3] O aplicativo deve permitir o acesso seguro ao perfil do usuário após o login. |
| Prioridade:  [ ] A  [ X ] B  [ ] C  [ ] D  [ ] E |
| Pontos de história:3 pontos |
|  |

|  |
| --- |
| ID:FE003 |
| Título: Visualização de Produtos |
| Requerente:Cliente |
| **Comentários:** |
| Critérios de aceitação:  [CA#1] O usuário deve ser capaz de ver uma lista de produtos disponíveis para compra.  [CA#2] O sistema deve exibir o nome, imagem e preço de cada produto.  [CA#3] O sistema deve exibir o número de itens disponíveis para cada produto.  [CA#4] O usuário deve ser capaz de clicar em um produto para ver mais detalhes. |
| Regras de negócio:  [RN #1] O sistema deve exibir uma mensagem de erro se não houver produtos disponíveis.  [RN #2] O usuário deve ser capaz de clicar em um produto para ver mais detalhes. |
| Requisito não funcional:  [RN F#1] O aplicativo deve ser responsivo e funcionar bem em diferentes tamanhos de tela.  [RN F#2] O aplicativo deve ter uma interface intuitiva e fácil de navegar para os usuários.  [RN F#3] O aplicativo deve exibir imagens de alta qualidade dos produtos para os usuários. |
| Prioridade:  [ ] A  [ X ] B  [ ] C  [ ] D  [ ] E |
| Pontos de história:3 pontos |

|  |
| --- |
| ID:FE004 |
| Título: Adição de Produtos ao Carrinho |
| Requerente:Cliente |
| **Comentários:** |
| Critérios de aceitação:  [CA#1] O usuário deve ser capaz de adicionar um produto ao carrinho.  [CA#2] O sistema deve exibir o número de itens no carrinho após a adição de um produto.  [CA#3] O usuário deve ser capaz de visualizar o conteúdo do carrinho a qualquer momento.  [CA#4] O sistema deve exibir o preço total dos itens no carrinho.  [CA#5] O usuário deve ser capaz de remover um item do carrinho. |
| Regras de negócio:  [RN #1] O usuário deve ser capaz de adicionar um produto ao carrinho.  [RN #2] O sistema deve exibir o número de itens no carrinho após a adição de um produto.  [RN #3] O usuário deve ser capaz de visualizar o conteúdo do carrinho a qualquer momento.  [RN #4] O sistema deve exibir o preço total dos itens no carrinho.  [RN #5] O usuário deve ser capaz de remover itens do carrinho. |
| Requisito não funcional:  [RN F#1] O aplicativo deve permitir que os usuários adicionem produtos ao carrinho de forma rápida e fácil.  [RN F#2] O aplicativo deve exibir informações claras sobre os produtos adicionados ao carrinho, incluindo quantidade e preço.  [RN F#3] O aplicativo deve ter um desempenho rápido e responsivo para a adição de produtos ao carrinho. |
| Prioridade:  [ ] A  [ X ] B  [ ] C  [ ] D  [ ] E |
| Pontos de história:3 pontos |

|  |
| --- |
| ID:FE005 |
| Título: Finalização da Compra |
| Requerente:Cliente |
| **Comentários:** |
| Critérios de aceitação:  [CA#1] O usuário deve ser capaz de finalizar a compra.  [CA#2] O sistema deve exibir o valor total da compra. |
| Regras de negócio:  [RN #1] O sistema deve exibir um resumo detalhado da compra antes da finalização.  [RN #2] O usuário deve ser capaz de editar as informações do carrinho antes da finalização da compra. |
| Requisito não funcional:  [RN F#1] O aplicativo deve permitir que o usuário finalize a compra de forma fácil e rápida.  [RN F#2] O aplicativo deve exibir informações claras sobre o preço total da compra.  [RN F#3] O aplicativo deve permitir que o usuário revise e edite as informações da compra antes de finalizar o pagamento.  [RN F#5] O aplicativo deve ter um desempenho rápido e responsivo para a finalização da compra. |
| Prioridade:  [ X ] A  [ ] B  [ ] C  [ ] D  [ ] E |
| Pontos de história: 8 pontos |

|  |
| --- |
| ID:FE006 |
| Título: Acompanhamento do Pedido |
| Requerente:Cliente |
| **Comentários:** |
| Critérios de aceitação:  [CA#1] O usuário deve ser capaz de visualizar o status do seu pedido em tempo real.  [CA#2] O sistema deve exibir o número. |
| Regras de negócio:  [RN #1] O usuário deve ser capaz de acompanhar o status do seu pedido em tempo real.  [RN #2] O sistema deve exibir o histórico de pedidos do usuário. |
| Requisito não funcional:  [RN F#1] O aplicativo deve permitir que o usuário visualize o status do seu pedido em tempo real.  [RN F#2] O aplicativo deve fornecer informações claras sobre o tempo estimado de entrega.  [RN F#3] O aplicativo deve garantir a segurança das informações do pedido, incluindo informações pessoais do usuário.  [RN F#4] O aplicativo deve ter um desempenho rápido e responsivo para o acompanhamento do pedido. |
| Prioridade:  [ X ] A  [ ] B  [ ] C  [ ] D  [ ] E |
| Pontos de história: 5 pontos |

|  |
| --- |
| ID:FE007 |
| Título: Controle de Estoque |
| Requerente:Cliente |
| **Comentários:** |
| Critérios de aceitação:  [CA#1] O sistema deve ser capaz de controlar o estoque de cada tipo de cupcake  [CA#2] O sistema deve impedir que um pedido seja aceito se não houver estoque suficiente do cupcake solicitado  [CA#3] O estoque deve ser atualizado automaticamente após a conclusão de cada pedido |
| Regras de negócio:  [RN #1] O sistema deve monitorar o nível de estoque de cada produto e atualizá-lo em tempo real. |
| Requisito não funcional:  [RN F#1] O aplicativo deve exibir informações claras sobre a quantidade de cada produto em estoque. |
| Prioridade:  [ X ] A  [ ] B  [ ] C  [ ] D  [ ] E |
| Pontos de história: 5 pontos |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **História do usuário** | **Estimativa em pontos** | **Prioridade** |
| FE001 | Cadastro de usuário | 5 pontos | B |
| FE002 | Login de usuário | 3 pontos | B |
| FE003 | Visualização de produtos | 3 pontos | B |
| FE004 | Adição de produtos ao carrinho | 3 pontos | B |
| FE005 | Finalização da compra | 8 pontos | A |
| FE006 | Acompanhamento do pedido | 5 pontos | A |
| FE007 | Controle de estoque | 5 pontos | A |

**Diagrama de casos de uso**

A diagram of a person with white text

Description automatically generated

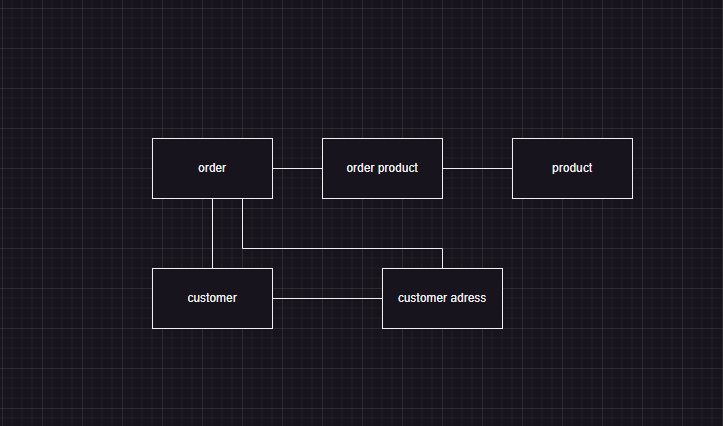
Diagrama de classe

A diagram of a service

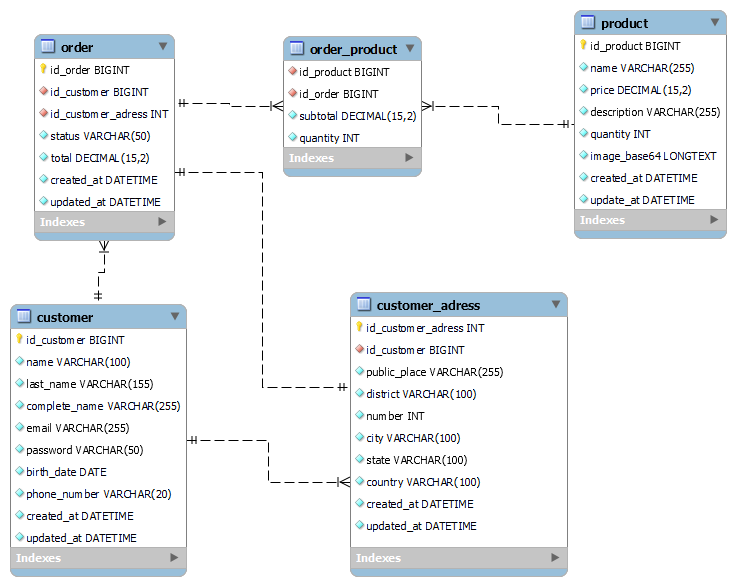
Description automatically generated**Banco de Dados:**

Insira a seguir o Modelo Conceitual, Modelo ER e o Projeto Físico do banco de dados desenvolvidos a partir do Diagrama de Classes:

**MODELO CONCEITUAL**



**MODELO ER**



**PROJETO FISICO**

CREATE SCHEMA IF NOT EXISTS `commerce` DEFAULT CHARACTER SET utf8 ;

USE `commerce` ;

-- -----------------------------------------------------

-- Table `commerce`.`product`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `commerce`.`product` (

`id\_product` BIGINT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(255) NOT NULL,

`price` DECIMAL(15,2) NOT NULL,

`description` VARCHAR(255) NOT NULL,

`quantity` INT NOT NULL,

`image\_base64` LONGTEXT NOT NULL,

`created\_at` DATETIME NOT NULL,

`update\_at` DATETIME NOT NULL,

PRIMARY KEY (`id\_product`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `commerce`.`customer`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `commerce`.`customer` (

`id\_customer` BIGINT NOT NULL,

`name` VARCHAR(100) NOT NULL,

`last\_name` VARCHAR(155) NOT NULL,

`complete\_name` VARCHAR(255) NOT NULL,

`email` VARCHAR(255) NOT NULL,

`password` VARCHAR(50) NOT NULL,

`birth\_date` DATE NOT NULL,

`phone\_number` VARCHAR(20) NOT NULL,

`created\_at` DATETIME NOT NULL,

`updated\_at` DATETIME NOT NULL,

PRIMARY KEY (`id\_customer`),

UNIQUE INDEX `email\_UNIQUE` (`email` ASC) VISIBLE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `commerce`.`customer\_adress`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `commerce`.`customer\_adress` (

`id\_customer\_adress` INT NOT NULL AUTO\_INCREMENT,

`id\_customer` BIGINT NOT NULL,

`public\_place` VARCHAR(255) NOT NULL,

`district` VARCHAR(100) NOT NULL,

`number` INT NOT NULL,

`city` VARCHAR(100) NOT NULL,

`state` VARCHAR(100) NOT NULL,

`country` VARCHAR(100) NOT NULL,

`created\_at` DATETIME NOT NULL,

`updated\_at` DATETIME NOT NULL,

PRIMARY KEY (`id\_customer\_adress`),

INDEX `fk\_customer\_adress\_customer1\_idx` (`id\_customer` ASC) VISIBLE,

CONSTRAINT `fk\_customer\_adress\_customer1`

FOREIGN KEY (`id\_customer`)

REFERENCES `commerce`.`customer` (`id\_customer`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `commerce`.`order`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `commerce`.`order` (

`id\_order` BIGINT NOT NULL AUTO\_INCREMENT,

`id\_customer` BIGINT NOT NULL,

`id\_customer\_adress` INT NOT NULL,

`status` VARCHAR(50) NOT NULL,

`total` DECIMAL(15,2) NOT NULL,

`created\_at` DATETIME NOT NULL,

`updated\_at` DATETIME NOT NULL,

PRIMARY KEY (`id\_order`),

INDEX `fk\_order\_customer1\_idx` (`id\_customer` ASC) VISIBLE,

INDEX `fk\_order\_customer\_adress1\_idx` (`id\_customer\_adress` ASC) VISIBLE,

CONSTRAINT `fk\_order\_customer1`

FOREIGN KEY (`id\_customer`)

REFERENCES `commerce`.`customer` (`id\_customer`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_order\_customer\_adress1`

FOREIGN KEY (`id\_customer\_adress`)

REFERENCES `commerce`.`customer\_adress` (`id\_customer\_adress`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `commerce`.`order\_product`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `commerce`.`order\_product` (

`id\_product` BIGINT NOT NULL,

`id\_order` BIGINT NOT NULL,

`subtotal` DECIMAL(15,2) NOT NULL,

`quantity` INT NOT NULL,

INDEX `fk\_product\_has\_order\_order1\_idx` (`id\_order` ASC) VISIBLE,

INDEX `fk\_product\_has\_order\_product\_idx` (`id\_product` ASC) VISIBLE,

CONSTRAINT `fk\_product\_has\_order\_product`

FOREIGN KEY (`id\_product`)

REFERENCES `commerce`.`product` (`id\_product`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_product\_has\_order\_order1`

FOREIGN KEY (`id\_order`)

REFERENCES `commerce`.`order` (`id\_order`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;**Codificação:**

Na Tabela a seguir insira as informações referentes ao desenvolvimento do código do processo.

|  |  |
| --- | --- |
| **Linguagem** | **Backend Typescript com NEST.js e Front Typescript e VUE.js** |
| **Banco de Dados** | **MySQL** |
| **Hospedagem** | **Frontend http://18.229.138.251:8081/ e Backend http://18.229.138.251:8080/health-check** |
| **Plataforma** | **AWS, Linux** |
| **Modo de Codificação** | (X) Tradicional  ( ) *Low-code*  ( ) *No-code* |
| ***Link* do repositório no [GitHub](https://github.com/login) com os códigos abertos** | https://github.com/FelipeFerrandin/pit-II |
| ***Link* da solução em funcionamento** | **http://18.229.138.251:8081/** |
| ***Link* do vídeo narrado (no mínimo 5 min)** | https://www.loom.com/share/943f59fbd7e24c82958d5ae66cf590cb?sid=4c5a24c7-3faa-406f-b259-9fdc21cb0ad5 |

**Testes da Solução**

Escolha 5 colegas para testar sua aplicação e preencha a tabela a seguir com as informações obtidas:

|  |  |
| --- | --- |
| **Nome:Matheus Vinicius Bianchini** | **Data do teste: 14/11/2023** |
| **O que testou e funcionou:**  **Fluxo cadastro**  **Fluxo login**  **Fluxo compra**  **Fluxo edição dados conta**  **Fluxo para visualização dos pedidos realizados** | |
| **O que testou e não funcionou – O que deve ser corrigido:**  **Precisa formatar valores** | |
| **Funcionalidade não testada (faltou ou não foi implementada):** | |

|  |  |
| --- | --- |
| **Nome: Gustavo Braun** | **Data do teste: 07/11/2023** |
| **O que testou e funcionou:**  **Fluxo cadastro**  **Fluxo login**  **Fluxo compra**  **Fluxo edição dados conta**  **Fluxo para visualização dos pedidos realizados** | |
| **O que testou e não funcionou – O que deve ser corrigido:**  **Quando o produto está no carrinho não consigo alterar a quantidade** | |
| **Funcionalidade não testada (faltou ou não foi implementada):** | |

|  |  |
| --- | --- |
| **Nome:Kaio Plandell** | **Data do teste:07/11/2023** |
| **O que testou e funcionou:**  **Fluxo cadastro**  **Fluxo login**  **Fluxo compra**  **Fluxo edição dados conta**  **Fluxo para visualização dos pedidos realizados** | |
| **O que testou e não funcionou – O que deve ser corrigido:**  **Quando o produto está no carrinho a descrição não aparece inteira deve ter um tooltip para mostrar** | |
| **Funcionalidade não testada (faltou ou não foi implementada):** | |

|  |  |
| --- | --- |
| **Nome: Daniel Kochany** | **Data do teste: 07/11/2023** |
| **O que testou e funcionou:**  **Fluxo cadastro**  **Fluxo login**  **Fluxo compra**  **Fluxo edição dados conta**  **Fluxo para visualização dos pedidos realizados** | |
| **O que testou e não funcionou – O que deve ser corrigido:**  **No carrinho não mostra o total de cada item** | |
| **Funcionalidade não testada (faltou ou não foi implementada):** | |

|  |  |
| --- | --- |
| **Nome:Kauana Bomfim** | **Data do teste: 09/11/2023** |
| **O que testou e funcionou:**  **Fluxo cadastro**  **Fluxo login**  **Fluxo compra**  **Fluxo edição dados conta**  **Fluxo para visualização dos pedidos realizados** | |
| **O que testou e não funcionou – O que deve ser corrigido:** | |
| **Funcionalidade não testada (faltou ou não foi implementada):** | |

**Laudo de Qualidade**

Insira a seguir o laudo de qualidade do sistema, apontando os erros e as correções. Não esqueça de coletar as evidências para inseri-las no laudo:

Todas as adversidades reportadas foram corrigidas com eficiência e qualidade, após as correções, foi novamente testado e aprovado pelo time de qualidade.

**Vídeo da Solução atualizada**

Após levantar os *feedbacks* e executar as correções necessárias e pertinentes, grave um vídeo de até 5 minutos apresentando as modificações realizadas no sistema.

|  |  |
| --- | --- |
| ***Link* para o vídeo** | **https://www.loom.com/share/44d5900097f34c368fe6d78da1b6c272?sid=e5f09b06-fbd1-4f57-8573-e8bc8e63934a** |