Instituto Federal do Piauí

Departamento de Informação, Ambiente, Saúde e Produção Alimentícia

Coordenação do Curso de Tecnoligia em Análise e Desenvolvimento de Sistemas

Desvendando a Inovação: Relatório Técnico do Desenvolvimento de Software

HelloPay

Aluno: **Vinicius Roosevelt Rodrigues Borges**

Professor Orientador: Rogério da Silva

Junho

2023

Instituto Federal do Piauí

Departamento de Informação, Ambiente, Saúde e Produção Alimentícia

Coordenação do Curso de Tecnoligia em Análise e Desenvolvimento de Sistemas

Relatório Técnico de Desenvolvimento de Software

HelloPay

Relatório Técnico de Desenvolvimento de Software apresentado para a conclusão do Curso de Tecnoligia em Analise e Desenvolvimento de Sistemas deste instituto.

Aluno: Vinicius Roosevelt Rodrigues Borges

Professor Orientador: Rogério da Silva

Junho

2023

Lista de Figuras

Lista de Tabelas

A lista de tabelas pode ser encontrada na página 1.

LISTA DE ABREVIATURAS E SIGLAS

API Interface de Programação de Aplicativos

WWW Rede mundial de computadores

UML Linguagem de Modelagem Unificada

UI Interface de Usuário

MVP Mínimo Produto Viável

UX Experiência do Usuário

HTTP Protocolo de Transferência de Hipertexto

Sumário

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

O Brasil tem atualmente mais de um smartphone por habitante, segundo levantamento anual divulgado pela FGV. São 242 milhões de celulares inteligentes em uso no país, que tem pouco mais de 214 milhões de habitantes, de acordo com o IBGE [4].Essa estatística mostra que os smartphones se tornaram uma parte essencial do cotidiano dos brasileiros, tornando-se uma ferramenta indispensável tanto para fins pessoais quanto profissionais.Essa proliferação de dispositivos móveis cria um ambiente propício para o empreendedorismo, abrindo novas oportunidades de negócios e transformando a forma como as empresas se conectam com os consumidores.

O empreendedorismo no contexto dos smartphones oferece uma ampla gama de possibilidades. Com um mercado tão grande e uma demanda crescente por serviços e aplicativos móveis, os empreendedores podem explorar diversas áreas para inovar e atender às necessidades dos usuários. Um exemplo claro é o desenvolvimento de aplicativos móveis. Com tantos smartphones em circulação, há uma demanda constante por aplicativos que ofereçam soluções úteis, entretenimento, serviços financeiros, educação, saúde e muito mais. Os empreendedores podem aproveitar essa demanda para criar e lançar aplicativos que atendam às necessidades específicas dos usuários brasileiros, seja criando novas formas de entretenimento, facilitando o acesso a serviços essenciais ou fornecendo soluções inovadoras para problemas cotidianos. Em resumo, o Brasil está vivenciando uma realidade em que o número de smartphones ultrapassa a marca de um por habitante, oferecendo uma infinidade de oportunidades para empresas.

A partir desses contexto (uso de celulares no Brasil, e novas formas de negócios através do uso de smartphones), é mais fácil para empresas criarem seus próprios aplicativo de pagamento de serviços, pois entrega aos seus clientes uma maior integração e facilidade no momento do pagamento,dos serviços prestados. Além de facilitar a vida dos usuários e também das empresas, fazendo com que o foco de uma empresa seja em outras áreas que possibilitam um maior conforto na hora de gerência/pagamento dos serviços prestados. Para contribuir com essas empresas é proposto um software mobile para dar dinamicidade para esses pagamentos, mas também um interface confortável para empresa e seu usuário

1.2 JUSTIFICATIVA

Na Atualidade é muito difícil para pessoas que tão iniciando na vida do empreendedorismo ou já são micro-empresas nascerem grande e concorrerem com outras empresas, logo para se destacar no mercado é obrigatório trabalhar no diferencial de que as concorrentes não oferecem, mantendo assim um padrão de qualidade, logo esse software impacta positivamente na qualidade de atendimento,porquê através de poucos cliques o usuário já estará fazendo o pagamento de um serviço e usufruindo sem precisar ligar para um atendente ou até mesmo o dono do negócio,desse empreendedor ou da micro empresas que estão procurando ter dinamicidade na empresa ou no pequeno negócio. Seguindo essa ideia, esse software busca criar também soluções de pagamento para que seja ofertado a empresa um app de gerenciamento (acompanhamento das assinaturas de um cliente) de pagamento, para que assim haja facilidade na hora da comprança dos clientes e pagamento por eles.A administração de pagamentos em aplicativos mobile refere-se à capacidade de processar e facilitar transações financeiras dentro do contexto de um aplicativo móvel.Essa funcionalidade é fundamental para aplicativos que oferecem serviços de comércio eletrônico, assinaturas, serviços on-demand, entre outros.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Solução de Software para gestão de pagamento de um serviço,ao criar um software mobile, permite-se criar uma integração de recursos de gestão de pagamentos permitindo usuários realizem transações de forma conveniente e segura diretamente pelo aplicativo, o que também contribui para que o dono do software consigar agradar seus clientes, e ter maior controle sobre o seu negócio,possibilitando assim uma melhoria na área de serviços da sua empresa ou pequeno negócio.

1.3.2 Objetivos Específicos

- Gestão de Pagamentos desacoplado (e com possiblidades de novas integrações de pagamentos) que possibilita as ações de criação, emissão e acompanhamento de compensação em diversas modalidades, tais como Cartão, PIX e Boleto.
- Disponibilizar um Sistema Mobile para tornar o aplicativo mais acessível para as

Empresas.

- Oferecer uma experiência de compra ou pagamento integrada, sem a necessidade de redirecionar os usuários para sites externos ou aplicativos de terceiros.

Resumo

Neste capítulo,foi apresentado uma introdução sobre os aumentos do uso de celulares no Brasil,como justificativa para a criação de um aplicação mobile.Logo depois,foi mostrada a justificativa do trabalho em si,que se concentra na ideia dos benéficios a união dos temas celulares e como a tecnologia pode trazer benéficios para empresas de menores porte concorrerem com as grandes no mercado,centrado na temática Gestão de Pagamentos.Por último, foram pontuados os objetivos geral e específicos do projeto.

Os próximos capítulos estão apresentados na seguinte sequência:

- **Fundamentação Teórica:** Capítulo voltado para a exibição dos conceitos teóricos utilizados no desenvolvimento da solução proposta no trabalho;
- **Metodologia:** Capítulo dedicado a apresentação da metologia utilizada na criação do projeto da solução, através do levantamento de requisitos do software proposto, da listagem das ferramentas que foram e das que serão utilizadas, e apresentação do projeto de aplicação.
- **Conclusão:** Capítulo direcionado a conclusão do projeto, com a sugestão de trabalhos futuros, os quais se resumem a iniciar o desevolvimento prático da solução.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 TECNOLOGIA,EMPRENDEDORISMO E PAGAMENTOS

Com o advento das redes sociais, a exemplo do Orkut, em janeiro de 2004, a internet ganhou outros contornos,começou a gerar novos empregos, novas empresas sugiram no ramo da tecnologia, o marketing digital começou a ganhar força, os bancos começaram a ter novas formas de transações. Toda essa mudança proporcionada pela internet começou com o aparecimento de sites como Kazaa e P2P, que tinham como propósito compartilhar dados, o que deu abertura para a entrada de outras comunidades online, como Twitter, Skype, Orkut e, mais recentemente, o Facebook e o Instagram. As redes sociais evoluíram tanto ao longo do tempo que hoje também são utilizadas para negócios [17]. Um oportunidade interessante é a economia compartilhada, impulsionada por aplicativos móveis. Plataformas de compartilhamento de caronas, aluguel de imóveis e serviços diversos têm se tornado cada vez mais populares, permitindo que empreendedores criem negócios baseados na conexão entre usuários e provedores de serviços. Essa abordagem tem o potencial de transformar setores inteiros da economia, gerando oportunidades para empreendedores inovadores. O mundo dos pagamentos é um ecossistema complexo e diversificado que envolve várias partes e tecnologias para facilitar as transações financeiras entre indivíduos, empresas e instituições.Geralmente os Pagamentos Giram em Torno de alguns atores:Participantes,Modalidades de Pagamento,Processamento do pagamento e Segurança.

2.1.1 Participantes

Geralmente são pessoas ou empresas que realizam pagamentos para adquirir bens ou serviços

2.1.2 Modalidades de Pagamento

Vária desde a dinheiro em espécie há cartões de crédito/débito e criptomoedas.

2.1.3 Processamento do pagamento

São comumentes conhecidos como os status(estado de uma transação).

- Autorização: O processo de verificação e validação do pagamento pelo emissor do cartão ou instituição financeira envolvida. utilizados no desenvolvimento da solução proposta no trabalho;
- Liquidação: Transferência do valor do pagamento do cliente para o comerciante, geralmente por meio das instituições financeiras envolvidas.
- Compensação: Verificação das transações e ajuste dos saldos entre as instituições financeiras.
- Conciliação: Processo de reconciliação entre as transações registradas pelos comerciantes e as transações processadas pelas instituições financeiras.

2.1.4 Segurança

Garante que dados sensiveis não sejam expostos

- Criptografia: Uso de algoritmos criptográficos
- Tokenização: Substituição dos dados sensíveis por um token único
- Autenticação: Verificação da identidade do cliente por meio de senhas, códigos de segurança, biometria, entre outros.
- Monitoramento: Uso de sistemas e tecnologias para identificar atividades suspeitas e prevenir fraudes.

É importante destacar que o mundo dos pagamentos está em constante evolução, impulsionado pelo avanço tecnológico e pela demanda por soluções mais rápidas, seguras e convenientes. Novas tecnologias, como blockchain, inteligência artificial e Internet das Coisas, estão sendo aplicadas para melhorar os processos de pagamento e proporcionar experiências mais eficientes aos usuários.

2.2 DIAGRAMA DE CLASSES

Diagrama de classes é um tipo de diagrama Linguagem de Modelagem Unificada (UML) muito usado para demonstrar as relações de classes que serão usados para modelagem de objetos quando o código for gravado. A forma de classe em si consiste em um retângulo com três linhas. A linha superior contém o nome da classe, a linha do meio, os atributos da classe e a linha inferior expressa os métodos ou operações que a classe pode utilizar. Classes e subclasses são agrupadas juntas para mostrar a relação estática entre cada objeto.[8]. Decerto existem alguns benefícios em utilizar o diagrama de classes, dentre eles, ilustrar modelos de dados para sistemas, sejam eles simples ou complexos, e expressar visualmente as necessidades de um sistema [8].

2.3 CASOS DE USO

Os casos de usos são um tipo de diagrama comportamental de UML [3], usando para descrever os detalhes das funcionalidades e interação com um sistemas. Indentifica-se várias ultilidades neses diagrama desde a documentação dos detalhes de um software, permitir melhor entendimento do fluxo feito no sistema e além de dar melhores detalhes para o desenvolvedor que vai implementar. Existe ainda uma estrutura de especificação do caso de uso, com mais detalhes sobre o mesmo. É possível modificar a estrutura de tópicos conforme necessário em cada situação. Os diagramas de caso de uso consistem em 4 objetos:Ator,Caso de uso,Sistema,Pacote [3]. Embora os diagramas de caso de uso possam ser usados para vários fins, há algumas diretrizes comuns que você precisa seguir ao desenhar casos de uso.

Estes incluem padrões de nomear, direções de setas, colocação de casos de uso, uso de caixas de sistema e também uso adequado de relacionamentos [3].

2.4 FIGMA

Figma é um software para edição de designs gráficos, vetores e protótipos, baseado em nuvem, que permite a criação de artes individualmente ou de forma colaborativa em tempo real, e, desde então, permite que designers e desenvolvedores se comuniquem e transfiram produtos criados através de códigos com maior agilidade e facilidade [1], as que se destacam são: versionamento automático, painel de camadas e objetos, biblioteca

compartilhável de componen- tes, possibilidade de compartilhar o projeto e colocá-lo em modo de apresentação, e também permite criar fluxo de navegação entre telas.

2.5 VISUAL STUDIO CODE

O Visual Studio Code é um editor de código-fonte lançado em 2015 e desenvolvido pela empresa Microsoft [16]. A IDLE é baseada no framework Electron e está disponível para diferentes plataformas de sistemas operacionais Windows, Linux e macOS. É uma ferramenta que vem com suporte integrado para algumas linguagens de programa- ção, bem como uma variada lista de extensões para outras linguagens [16]. Além de conta com ferramentas de versionamento de código como Git

2.6 GIT E GITHUB

São dois software que atuam junto o Git como orquestrador e o GitHub como armazenador entre outras gestões de código. Primeiramente, o Git é um sistema de controle de versão distribuído [7], e um recurso que o destaca dentre os outros do mesmo tipo é seu recurso de ramificações, as quais podem ser independentes entre si, ou seja, pode ser feita uma alteração em uma ramificação específica e não vai haver mudanças nas outras. Já o GitHub é um local de hospedagem de código-fonte e arquivos com controle de versão utilizando o Git [18].

2.7 DART

Dart é uma linguagem de programação usada para desenvolver aplicações de várias plataformas a partir do mesmo código fonte. Ela foi criada pela empresa Google em 2011 [11]. É uma linguagem fortemente tipada que auxilia no desenvolvimento de software e evitar erros de sintaxe, Porém, também possui desvantagens como ter pouco tempo de mercado

2.8 FLUTTER

Flutter é um framework de código aberto desenvolvido pela Google em 2015. É baseado na linguagem de programação Dart, o que possibilita criação de aplicações para mobile, web e desktop [12]. Uma grande vantagem do Flutter é sua alta produtividade, já que a mesma base de código vai servir tanto para um aplicativo Android como para um iOS [6].

Assim, não há necessidade de desenvolver uma aplicação para cada arquitetura Por isso, grandes empresas utilizam o framework como o Nubank, iFood, Ebay e o próprio Google [6].

2.9 JAVASCRIPT

JavaScript (frequentemente abreviado como JS) é uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinâmica fraca e multiparadigma (protótipos, orientado a objeto, imperativo e funcional) [13].

2.10 TYPESCRIPT

TypeScript é uma linguagem de programação de código aberto desenvolvida pela Microsoft. É um superconjunto sintático estrito de JavaScript e adiciona tipagem estática opcional à linguagem. Tipos fornecem uma maneira de descrever a forma de um objeto, fornecendo melhor documentação e permitindo que o TypeScript valide se seu código está funcionando corretamente. Como TypeScript é um superconjunto de JavaScript, os programas JavaScript existentes também são programas TypeScript válidos [15].

2.11 NODE

Node.js é um ambiente de execução JavaScript que permite executar aplicações desenvolvidas com a linguagem de forma autônoma, sem depender de um navegador. Com ele, é possível criar praticamente qualquer tipo de aplicações web, desde servidores para sites estáticos e dinâmicos, até APIs e sistemas baseados em microserviços [9].

2.12 API

A sigla Interface de Programação de Aplicativos (API) deriva da expressão inglesa Application Programming Interface que, traduzida para o português, pode ser compreendida como uma interface de programação de aplicação. Ou seja, API é um conjunto de normas que possibilita a comunicação entre plataformas por meio de uma série de padrões e protocolos. Por meio de APIs, desenvolvedores podem criar novos softwares e aplicativos capazes de se comunicar com outras plataformas. Por exemplo: caso um desenvolvedor queira criar um aplicativo de fotos para Android, ele poderá ter acesso à câmera do celular

através da API do sistema operacional, sem ter a necessidade de criar uma nova interface de câmera do zero [5].

2.13 REST

Na informática e engenharia de software, Representational State Transfer (abreviado REST), em português Transferência de Estado Representacional, é um estilo de arquitetura de software, criado em 2000 por Roy Fielding, que define um conjunto de restrições a serem usadas para a criação de um tipo especial de serviços-Web, denominados Web services RESTful, que fornecem interoperabilidade entre sistemas de computadores na Internet; RESTful permite que os sistemas solicitantes acessem e manipulem representações textuais de recursos da Web usando um conjunto uniforme e predefinido de operações sem estado (requisição e resposta independentes). Outros tipos de Web services, como Web services SOAP, expõem seus próprios conjuntos de operações arbitrários [14].

2.14 POSTGRES

O PostgreSQL é um banco de dados relacional de software livre com suporte de 30 anos de desenvolvimento, sendo um dos bancos de dados relacionais mais estabelecidos disponíveis. A popularidade do PostgreSQL com desenvolvedores e administradores se deve à sua flexibilidade e integridade notáveis. Por exemplo, o PostgreSQL dá suporte a consultas relacionais e não relacionais, além de que sua natureza de software livre significa uma comunidade dedicada de mais de 600 colaboradores que melhora constantemente o sistema de banco de dados [10].

3 METODOLOGIA

A metodologia do projeto foi feita em três momentos. Primeiramente, com o levantamento dos requisitos do software, que definiu os serviços que o software oferecerá. Em seguida, é abordada quais ferramentas foram e quais serão utilizadas no desenvolvimento da aplicação. Por fim, é apresentada a aplicação, com suas caraterísticas e fluxo de telas.

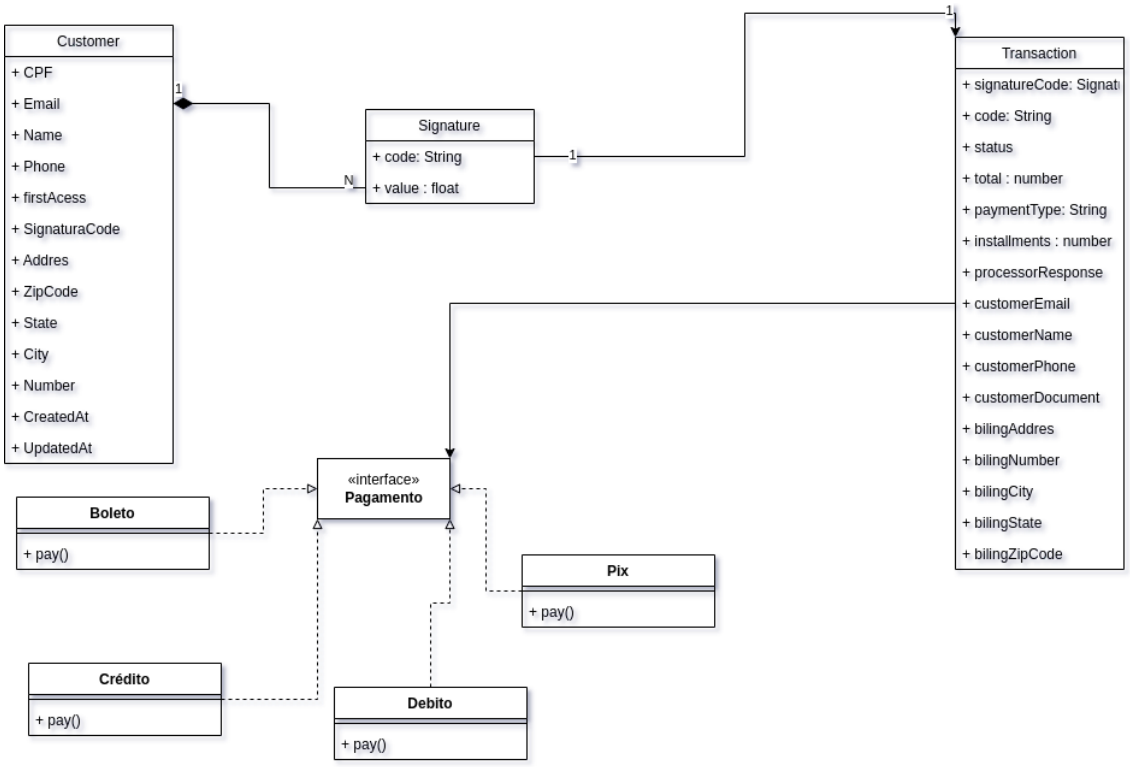
3.1 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos para o desenvolvimento do projeto, foi feito através de dois itens: diagrama de classes e casos de uso.

3.1.1 Diagrama de classes

Nesse tópico, foi aplicado o diagrama de classes, para exibir as classes presentes no sistema e seus respectivos atributos e métodos.

Figura 1: Figura 1: Diagrama de Classe



Fonte: Elaborada pelo Autor

Primeiramente,o usuário pode assumir o papel de Admin ou Cliente,nunca os dois ao mesmo tempo, que define desempenham funções diferentes no sistema. Adiante, as assinaturas faz o papel de gestor entre as transações e o cliente, logo um cliente pode ter apenas uma assinatura e assinatura pode pertencer a diferentes clientes, Ademais, as transações recebem uma assinatura e dados de um cliente para serem realizadas e dependendo da interface de pagamento recebe essa transações seus atributos podem ser alterados Por fim, a interface de pagamento depende do tipo de pagamento que o cliente deseja fazer, assim se adaptando a diferentes contextos de pagamentos

3.1.2 Casos de uso

Nesse item, foram utilizados os casos de uso, uma forma existente para especificar as funcionalidades do software. Nas tabelas abaixo, foram abordadas sobre cada função o ator principal, a descrição, o propósito, a pré-condição, o fluxo principal, e, se existir, o fluxo alternativo e o fluxo exceção.

UC-01	**EFETUAR LOGIN**
Ator principal	Usuário
Descrição	Autenticação do usuário cadastrado no sistema, permitindo a realização de operações na área restrita do site.
Propósito	Acessar o sistema para ver pagamentos, sua assinatura..
Pré-condição	O usuário deve estar cadastrado no sistema..
Fluxo principal	Entrar pelo login; O sistema solicita informações obrigatórias para a autenticação: e-mail e senha; O usuário informa os dados de autenticação; O sistema habilita as ações relacionadas ao software.
Fluxo Alternativo	Opção voltar • O usuário seleciona a opção voltar; • O sistema retorna a tela anterior. • O usuário seleciona a opção esqueci a senha; • O sistema mostra uma tela para atualizar a senha; • O usuário informa a nova senha; • A senha é atualizada; • O sistema retorna a página de login.
Fluxo de Exceção	Informações obrigatórias para a autenticação incorretas: • O sistema informa ao usuário o erro.

Tabela 1: Tabela de Caso de Uso de Realizar Login

UC-02	**Realizar cadastro**
Ator principal	Admin
Descrição	Cadastro do cliente, que vai permitir o login e acesso ao software.
Propósito	Acessar o sistema para ver pagamentos, sua assinatura..
Pré-condição	O Cliente deve ter email unico, que não consta na base de dados e informações válidas Admin Deve conhecer uma senha especial para fornecer ao sistema
Fluxo principal	• Entrar pelo login; • O sistema retorna a página inicial e possuindo a opção de cadastrar novos clientes com • O Admin informa os dados de cadastro; • O sistema habilita as ações relacionadas ao software para o usuário.
Fluxo Alternativo	N/A
Fluxo de Exceção	Campos obrigatórios • O sistema verifica que os campos obrigatórios não forma preenchi- dos; • O sistema exibe mensagem de alerta. E-mail já existente • O sistema verifica que o e-mail informado já está cadastrado no sistema; • O sistema exibe mensagem de alerta.

Tabela 2: Tabela de Caso de Uso de Realizar Cadastro

UC-03	**Realizar cadastro de uma Assinatura**
Ator principal	Admin
Descrição	Cadastro das assinaturas, que permite indentificar o plano do usuário
Propósito	Permitir o usuário paga sua assinatura..
Pré-condição	O Admin deve está logado para criar as assinaturas
Fluxo principal	• Entrar pelo login; • A Home Possui um botão que indica a criação de assinaturas • O Admin informa os dados da assinatura; • O sistema habilita as ações relacionadas ao usuário poder tem aquela assinatura.
Fluxo Alternativo	N/A
Fluxo de Exceção	Caso o Admin não esteja logado ou quem quer criar a assinatura não seja admin • O Sistema verifica a identidade do usuário e caso ele não seja admin não deixa continuar e exibi uma mensagem; • O sistema exibe mensagem de alerta.

Tabela 3: Tabela de Caso de Uso de Criar uma Assinatura

UC-04	**Apagar nos registro uma Assinatura**
Ator principal	Admin
Descrição	Excluir dos registros uma assinatura que não faz mais parte do contexto do sistema
Propósito	Manter Limpo os Registros que não vão mais ser ultilizados..
Pré-condição	O Admin deve está logado para apagar as assinaturas
Fluxo principal	• Entrar pelo login; • A Home Possui um botão que leva para ver todas as assinaturas e caso o haja a necessidade de excluir a assinatura é apagada do sistemas sem prejudicar outras entidades
Fluxo Alternativo	N/A
Fluxo de Exceção	Caso o Admin não esteja logado ou Usuário qualquer deseje apagar a assinatura • O Sistema verifica a identidade do usuário e caso ele não seja admin não deixa continuar • O sistema exibe mensagem de alerta.

Tabela 4: Tabela de Caso de Uso de Apagar uma Assinatura

UC-05	**Alterar dos registros uma Assinatura**
Ator principal	Admin
Descrição	Alterar dos registros uma assinatura que atualizando o seu contexto no sistema
Propósito	Manter as assinaturas de acordo com o contexto do software..
Pré-condição	O Admin deve está logado para alterar as assinaturas
Fluxo principal	• Entrar pelo login; • A Home Possui um botão que leva para ver todas as assinaturas • E nessa listagem possui o botão de alterar assinatura • O Admin informa os campos a serem alterados • Por fim o Admin salva as alterações
Fluxo Alternativo	N/A
Fluxo de Exceção	Caso o Admin não esteja logado ou Usuário qualquer deseje alterar alguma assinatura • O Sistema verifica a identidade do usuário e caso ele não seja admin não deixa continuar • O sistema exibe mensagem de alerta.

Tabela 5: Tabela de Caso de Uso de Alterar uma Assinatura

UC-06	**Listar dos registros todas as Assinaturas**
Ator principal	Admin
Descrição	Visualiza dos registros todas as Assinaturas do Sistema
Propósito	Visualizar todas as Assinaturas do Software
Pré-condição	O Admin deve está logado para visualizar todas as assinaturas
Fluxo principal	• Entrar pelo login; • A Home Possui um botão que leva para ver todas as assinaturas • E nessa nova tela é possível ver todas as assinaturas
Fluxo Alternativo	N/A
Fluxo de Exceção	Caso o Admin não esteja logado ou Usuário qualquer deseje visualizar todas as assinaturas • O Sistema verifica a identidade do usuário e caso ele não seja admin não deixa continuar • O sistema exibe mensagem de alerta.

Tabela 6: Tabela de Caso de Uso de Visualizar todas as Assinaturas

UC-07	**Criar uma Transação**
Ator principal	Usuário
Descrição	O Usuário está preparando o caminho para pagar sua assinatura
Propósito	Realizar o pagamento da assinatura
Pré-condição	O Usuário deve está logado, e possuí uma assinatura
Fluxo principal	• Entrar pelo login; • A Home Possui que levar o usuário a visualizar sua assinatura • Caso ele queria ele pode efetuar o pagamento de sua assinatura • Criando uma Transação no Sistema • Que fica ouvindo as modifiações caso haja um pagamento daquela transação
Fluxo Alternativo	Caso o Usuário não deseje continuar • É criado a transação,porém com status de cancelada pelo usuário
Fluxo de Exceção	Caso o email informado não for o mesmo que do usuário • A transação é cancelada • O Sistema verifica a identidade do usuário e do email passado para criacão da transaction • O sistema exibe mensagem de alerta.

Tabela 7: Tabela de Caso de Uso de Criar uma transação

UC-08	**Altera o Status de uma Transanção de Pendente para Pago**
Ator principal	Sistema
Descrição	Quando o Sistema é notificado é alterado o status da transação do Usuário
Propósito	Atualizar o Status da Transação para Pago
Pré-condição	O usuário deve ter feito iniciado o pagamento da Transação e ter seguido as etapas de colocar seus dados para o pagamento
Fluxo principal	• O Sistema é notificado que o usuário quer fazer o pagamento e muda o status de Iiniciado para Pendente • O Cliente Paga a Transação • A Gateway de Pagamento notificar o sistema • E Alterar status de Pendente para Pago
Fluxo Alternativo	N/A
Fluxo de Exceção	Caso o usuário não insira dados validos para o pagemnto ou não foi houve algo problema de saldo • A transação volta para o status de Pendente • O Sistema exibe uma mensagem para o usuário de alerta avisando o ocorrido Caso haja algum problema com o Sistema ou Gateway • O Sistema verifica o Erro • Caso seja problema na gateway o status fica pendente até o sistema ser notificado que houve sucesso ou falha na transação • Caso seja sucesso para aprovado • Caso seja falha para erro • O Sistema mostra uma mensagem de erro ou sucesso

Tabela 8: Tabela de Caso de Uso de Alterar Status de Uma Transação

3.2 FERRAMENTAS

Essa seção compartilha,todas as **coisas** que contribui e contribuirão para criação do software, que váriam desde a linguagem de progamação,criação de Interface de Usuário (UI),frameworks, e criação de diagramas.

3.2.1 Ferramentas utilizadas

- **Node**: Adotado como motor do JavaScript, que permite a execução do Javascript fora de um navegador web,possibilitando o desenvolvimento de aplicativos do lado do servidor.

- **TypeScript**: Aplicado devido a ser uma linguagem de progamação tipada, o que dar vantagens para o compilador enterpretar melhor variáveis,métodos e classes,dando melhor produtividade.

- **JavaScript**: Após o projeto ser publicado o TypeScript e transpilado em JavaScript para poder rodar no motor do Node.

- **Draw.io**: Empregado para criação de diagrmas online como os da UML, sendo um editor de diagrams de código aberto que permite criar uma gama de outros diagramas

- **Postgres**: Postgres, é um sistema de gerenciamento de banco de dados relacional de código aberto. Ele é conhecido por sua confiabilidade, escalabilidade e recursos avançados, tornando-se uma escolha popular para muitas aplicações e projetos.

3.2.2 Ferramentas a Serem utilizadas

- **Dart**: O Dart é uma linguagem moderna e eficiente, com suporte a recursos como tipagem estática, gerenciamento de memória eficiente e compilação just-in-time (JIT) para um desenvolvimento e execução rápidos.

- **Flutter**: Flutter é um framework de código aberto desenvolvido pelo Google, usado para criar aplicativos nativos de alta qualidade para dispositivos móveis, web e desktop. Ele permite que os desenvolvedores escrevam o código uma vez e o executem em diferentes plataformas, utilizando uma única base de código.

- **Figma**: O Figma é uma plataforma de design de interface do usuário UI e experiência do usuário Experiência do Usuário (UX) baseada na nuvem.

- **Draw.io**: Empregado para criação de diagrmas online como os da UML, sendo um editor de diagrams de código aberto que permite criar uma gama de outros diagramas

- **Postgres**: Postgres, é um sistema de gerenciamento de banco de dados relacional de código aberto. Ele é conhecido por sua confiabilidade, escalabilidade e recursos avançados, tornando-se uma escolha popular para muitas aplicações e projetos.

4 HelloPay

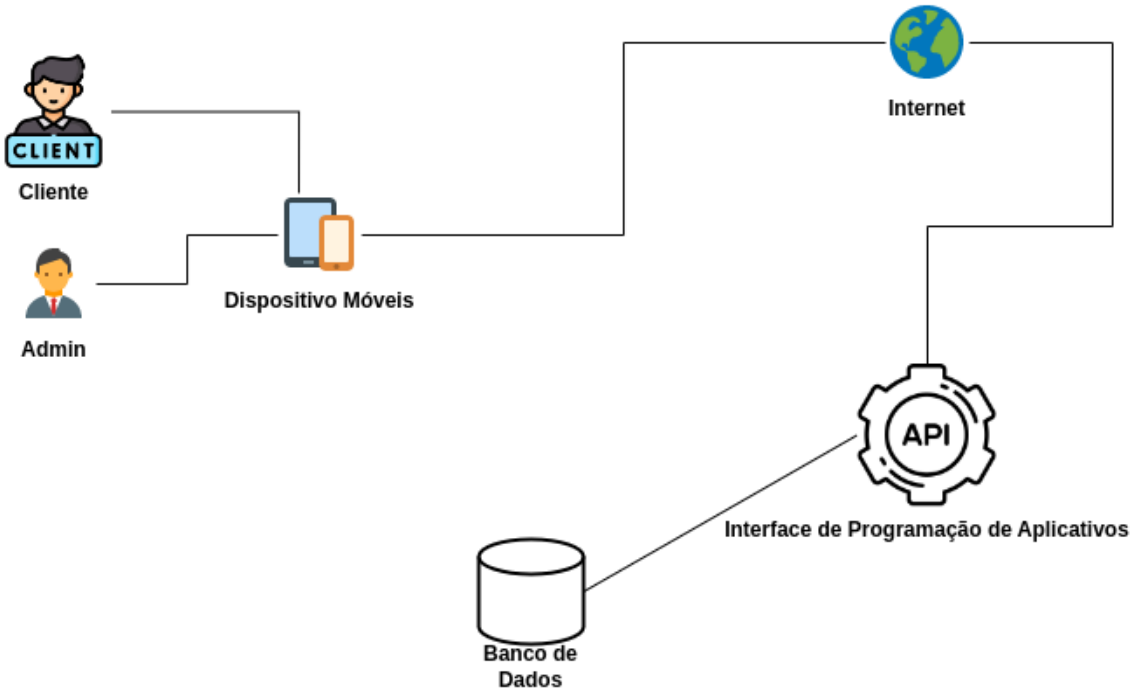
O HelloPay tem a proposta de ser mobile e somente para android inicialmente.Do mesmo modo, o sofware será baseado em interfaces famosas como nubank,ifood,uber,entre outros aplicativos, devido a ter uma interface boa e rápida para o usuário.A ideia é juntar o crescente uso dos celulares com empreendedorismo e pagamentos que possibilite a empresas menores competirem com as grandes.

4.1 Resultados

Essa seção é reservada para demonstrar os resultados dos casos de usos unidos e alguns detalhes singelos da aplicação,como arquitetura do sistema e como ele é divido e a função de cada parte dele.

A aplicação HelloPay tem como objetivo ser um app voltado para gestão de pagamentos, esse app é altamente flexível, permitindo a integração de novos métodos de pagamento de forma independente. Ele oferece funcionalidades abrangentes para a criação, emissão e acompanhamento de transações em diversas modalidades, incluindo Cartão, PIX e Boleto. O sistema é dividido em três partes principais, que garantem um fluxo eficiente e seguro de pagamentos.

Figura 2: Estrutura do Software



Fonte: Elaborada pelo Autor

- **API**: As APIs são um conjunto de padrões que fazem parte de uma interface [5].A Api do HelloPay também contém conjunto de regras e protocolos que permite troca de informações entre um sistema(app de celular) para mitigar a problematica citada no Capítulo 1.Essa API também conta com a ultilização do protocolo Protocolo de Transferência de Hipertexto (HTTP), que é um protocolo muito famoso que especifica como será a comunicação entre um navegador e um servidor web,sendo um dos principais da Rede mundial de computadores (WWW) [2].

- **Aplicação móvel**: Será a parte mais ultilizada do sistema,além de conter a interface do usuário UI além de ser o principal ponto de ligação entre o usuário e os dados providos pela API. Será ultilizado o framework Flutter, devido ao seu desenvolvimento ser compartilhado em diferente plataformas a partir do mesmo código fonte e criação de interfaces bem concisas.

- **Banco de Dados**: Um sistema organizado para armazenar, gerenciar e recuperar informações de maneira estruturada. Ele é projetado para armazenar grandes quantidades de dados de forma eficiente e permitir o acesso rápido e seguro a esses dados quando necessário.

4.2 Cronograma

Inicialmente foi criado totalmente a criação e manipulação de assinaturas,usários e transações, que já colaboram para administração de usuário e de suas transações pelo administrador além do controle dos serviços oferecidos pela empresa.As Próximas etapas para o projeto, é a produção do aplicativo mobile e conexão com gatewaty de pagamento,que possiblita fazer as transações com segurança e diferentes métodos de pagamentos, o que possiblita atingir o objetivo de pagamento desacoplado, ou seja um Mínimo Produto Viável (MVP) para já poder ser valida pelos usuários.

Todo o código produzido é versionado no GitHub e cada parte do processo de contrução do software será documentada.É planejado a entrega de um aplicativo para celulares Android totalmente integrado com API do HelloPay, além da criação de teste para garantir que as funcionalidades oferecidas pelo HelloPay são funcioanis e possui uma maior segerunça graças aos testes.

5 CONCLUSÃO E TRABALHOS FUTUROS

Esse trabalho apresentou o projeto do aplicativo HelloPay,um app de gestão e pagamento com foco em serviços disponibilizados por uma empresa. Para poder justificar esse trabalho, foi feito uma pesquisa sobre o uso de celulares,a tecnologia como auxilidora do homem.

Dessa forma foi proposto uma aplicação que une as duas temáticas com administração de pagamentos, um sistema de gerenciamento de pagamentos desvinculado que possibilita as etapas de criação, emissão e acompanhamento de compensação em diversas modalidades. Primeiramente foi levantado objetivos gerais e especificos da aplicação,contribui para o gerenciamento das transações dos clientes e pagamento de serviços. Em seguida foi realizado o levantamento de requisitos, para melhor compreensão do software.Por conseguinte foi exposto as ferramentas a serem ultilizados,desde linguagens de progamação a frameworks e banco de dados, que foram e serão utilizadas no futuro da aplicação.

Referências

[1] BLOGB2BSTACK. “Entenda o que é e como usar o Figma para criar designs”. Em: 30.2 (2021). Acesso em: 22 jun. 2023, pp. 10–20.

[2] Matheus Bigogno Costa. “O que é HTTP”. Em: 30.2 (2021). Acesso em: 22 jun. 2023, pp. 10–20.

[3] CREATELY. “Tutorial do diagrama de caso de uso (guia com exemplos)”. Em: 30.2 (2021). Acesso em: 22 jun. 2023, pp. 10–20.

[4] Cruzeirofm. “Brasil tem mais smartphones que habitantes”. Em: 30.2 (2022). Acesso em: 22 jun. 2023, pp. 10–20.

[5] Clara Fabro. “O que é API e para que serve? Cinco perguntas e respostas”. Em: 30.2 (2020). Acesso em: 22 jun. 2023, pp. 10–20.

[6] GEEKHUNTER. “Flutter: por que aprender o framework da Google é uma boa ideia em um mercado mobile crescente”. Em: 30.2 (2021). Acesso em: 22 jun. 2023, pp. 10–20.

[7] HOSTGATOR. “Conheça o Git, o sistema de versionamento que protege os projetos”. Em: 30.2 (2020). Acesso em: 22 jun. 2023, pp. 10–20.

[8] LUCIDCHART. “O que é um diagrama de classe em UML?” Em: 30.2 (2019). Acesso em: 22 jun. 2023, pp. 10–20.

[9] Diego Melo. “O que é Node.js? [Guia para iniciantes]”. Em: 30.2 (2021). Acesso em: 22 jun. 2023, pp. 10–20.

[10] Azure Microsoft. “O que é PostgreSQL”. Em: 30.2 (2023). Acesso em: 22 jun. 2023, pp. 10–20.

[11] WIKIPÉDIA. “Dart (linguagem de programação)”. Em: 30.2 (2023). Acesso em: 22 jun. 2023, pp. 10–20.

[12] WIKIPÉDIA. “Flutter”. Em: 30.2 (2023). Acesso em: 22 jun. 2023, pp. 10–20.

[13] WIKIPÉDIA. “JavaScript”. Em: 30.2 (2023). Acesso em: 22 jun. 2023, pp. 10–20.

[14] WIKIPÉDIA. “REST”. Em: 30.2 (2023). Acesso em: 22 jun. 2023, pp. 10–20.

[15] WIKIPÉDIA. “TypeScript”. Em: 30.2 (2022). Acesso em: 22 jun. 2023, pp. 10–20.

[16] WIKIPÉDIA. “Visual Studio Code”. Em: 30.2 (2022). Acesso em: 22 jun. 2023, pp. 10–20.

[17] WOMAKERSCODE. “O que é, o que é? O que são as ferramentas de versionamento e por que utilizar em seus projetos.” Em: 30.2 (2021). Acesso em: 22 jun. 2023, pp. 10–20.

[18] WOMAKERSCODE. “O que é, o que é? O que são as ferramentas de versionamento e por que utilizar em seus projetos.” Em: 30.2 (2021). Acesso em: 22 jun. 2023, pp. 10–20.