



Fatec São Caetano do Sul
Curso de Análise e Desenvolvimento de Sistemas

Sistema Web e Mobile para eventos

Felipe Pasianot Rodrigues

Kaue Ribeiro

São Caetano do Sul

2020

FOLHA DE ROSTO

Felipe Pasianot Rodrigues

Kaue Ribeiro

Sistema Web e Mobile para eventos

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de São Caetano do Sul, sob a orientação do Professor Humberto Luiz de Toledo, como requisito para a obtenção do diploma de Graduação no Curso de Análise e Desenvolvimento de Sistemas

São Caetano do Sul / Sp

2020

RESUMO

O projeto consiste em um aplicativo para eventos facilitando a venda de ingressos, inscrição para eventos, cobrança de contribuições e a gestão do bar. Todos os eventos publicados na plataforma estão sob responsabilidade de seus Organizadores.

Nosso foco foi agilizar os processos que interferem no atendimento ao cliente, buscando automatizar e organizar processos que ainda são lentos e feitos manualmente, podendo assim melhorar a experiência do cliente em eventos.

Palavras-chave: Aplicativo de Eventos, Ingresso digital

Abstract

The project consists of an application for events facilitating the sale of tickets, registration for events, collection of contributions for events and the management of the bar. All events published on the platform are under the responsibility of their Organizers.

Our focus was to streamline processes that interfere with customer service, seeking to automate and organize processes that are still slow and done manually, thus being able to improve the customer experience at events.

Keyword: Events application, Digital ticket

SUMÁRIO

INTRODUÇÃO	8
1.PLANEJAMENTO DO SISTEMA.....	8
1.1 Descrição do Projeto.....	8
1.2 Cronograma.....	10
1.3 Recursos.....	11
1.3.1 Recursos Humanos.....	11
1.3.2 Recursos Físicos.....	11
2.ESPECIFICAÇÃO DOS REQUISITOS DO SISTEMA	11
2.1 O Produto.....	11
2.1.1 O Propósito.....	11
2.1.2 Clientes.....	11
2.1.3 Usuários.....	12
2.2 Escopo do Projeto.....	12
2.3 Escopo dos Requisitos.....	14
2.3.1 Funcionais.....	14
2.3.2 Não Funcionais.....	15
2.4 Protótipo do Produto.....	16
2.4.1 Modelo de Navegação.....	16
2.4.1.1 Raia do Consumidor.....	16
2.4.1.2 Raia do Cliente.....	17
2.4.1.3 Raia do Administrador.....	18
2.4.2 Protótipo das telas.....	19
3 Modelo de Caso de Uso do Sistema	21
3.1 Diagrama de Caso de uso.....	21
3.2 Documentação dos Atores.....	22
3.2.1 Ator Consumidor.....	22
3.2.2 Ator Cliente.....	23
3.2.3 Ator Administrador.....	23
3.3 Documentações dos Casos de Uso.....	24
3.3.1 Acesso ao sistema.....	24
3.3.2 Comprar Ingresso.....	25
3.3.3 Comprar Produto no Evento.....	26
3.3.4 Encerrar Comanda.....	27
3.3.5 Cadastrar Evento.....	28
4. MODELOS DE ANÁLISE DO SISTEMA.....	29
4.1 Especificações das Classes de Análise do Sistema.....	29

4.1.1 Visão geral dos Objetos de Análise.....	29
4.1.2 Visão geral dos Pacotes de Análise.....	31
4.1.2.1 Classe Cliente.....	32
4.1.2.2 Classe Consumidor.....	33
4.1.2.3 Classe Administrador.....	34
4.1.2.4 Classe Autenticação.....	35
4.2 Realização de Casos de Uso.....	36
4.2.1 Cenário do Consumidor.....	36
4.2.2 Cenário do Cliente.....	37
4.2.3 Cenário Administrador.....	37
5. MODELOS DE PROJETO DO SISTEMA	38
5.1 Classe de Projeto.....	38
5.2 Refinamento nos Relacionamentos.....	39
6. MODELOS DE IMPLEMENTAÇÃO E IMPLANTAÇÃO DO SISTEMA.....	41
6.1 Programação do Sistema.....	41
6.2 Testes.....	41
6.3 Elaboração do Procedimento de instalação.....	41
Conclusões Finais.....	42
REFERÊNCIAS BIBLIOGRÁFICAS	43

Sumario Imagens

Imagem 01: Raia Consumidor.....	16
Imagem 02: Raia Cliente.....	17
Imagem 03: Raia Administrador.....	18
Imagem 04: Protótipo da Página Inicial.....	19
Imagem 05: Protótipo da Página Eventos.....	20
Imagem 06: Protótipo da Página de Cadastro.....	20
Imagem 07: Protótipo da Página Sobre Nos.....	20
Imagem 08: Protótipo da Página de Login.....	21
Imagem 09: Diagrama de Casos de uso.....	22
Imagem 10: Diagrama de Pacotes de Analise.....	28
Imagem 11: Diagrama de Classe.....	30
Imagem 12: Diagrama de Sequência Consumidor.....	34
Imagem 13: Diagrama de Sequência Cliente.....	34
Imagem 14: Diagrama de Sequência Administrador.....	35
Imagem 15: Diagrama de Classe Refinado.....	37

1. INTRODUÇÃO

A proposta do Projeto é desenvolver um sistema web e um aplicativo para eventos com pagamentos automatizado assim diminuindo o transtorno do cliente em filas e com segurança de ter o seu ingresso digital, facilitando assim a compra e venda dentro dos estabelecimentos.

Pretende-se dinamizar e agilizar os processos que interferem no atendimento ao cliente, buscando automatizar e organizar processos que ainda são lentos e feitos manualmente.

1.1 PLANEJAMENTO DO SISTEMA

Descrição do Projeto

Este projeto foi realizado com ênfase em Eventos. Pois Pessoas que frequentam esses lugares com enormes demandas de público, se deparam com uma enorme fila para fazer o cadastro e comprar o ingresso. Após este, o Consumidor consegue fazer o cadastro e abrir uma comanda, um número aonde fica registrado todo seu consumo naquele lugar, porém pode haver transtornos, pois quando resolve-se adquirir algo, este encontra uma enorme fila. Com isso, o cliente ficará um longo período em espera para ser atendido, diminuindo sua experiência no local.

Após realizar a compra, o cliente encontra-se em uma outra fila para a retirada do produto. Nessa ocasião a demora é um pouco maior, pois terá que esperar o funcionário realizar o seu pedido, assim, perdendo-se grande parte do evento em uma fila.

Após as ações supracitadas, ao ir embora, o cliente se depara com uma outra fila para realizar a quitação de sua comanda. Esse problema poderia ser impedido se: ele fechasse a comanda mais cedo, mas o cliente não poderia consumir mais nada até a sua saída do estabelecimento, ou se realizasse a abertura de uma nova comanda. Então, por fim, depois de muita espera e pouco entretenimento o cliente

deixa o local, levando com ele uma má experiência do estabelecimento. Infelizmente esse é o cenário de muitos lugares em São Paulo.

Devido ao ocorrido acima, resolvemos desenvolver um aplicativo, no qual o cliente poderá realizar seu cadastro e comprar o ingresso antecipadamente, para evitar a fila ao chegar no estabelecimento. Por meio do aplicativo, o cliente terá acesso ao cardápio pelo celular e estará apto a realizar o seu pedido. E será avisado quando este estiver pronto. Com isso, será gerado um código e ele poderá retirá-lo através desse número, sem a necessidade de esperar na fila. Mudando completamente a experiência do cliente no estabelecimento.

Como desenvolver um aplicativo para dinamizar e reduzir problemas de insatisfação de clientes em baladas e bares?

Quais são as principais causas de insatisfação em baladas?

Como o desenvolvimento de um aplicativo resolverá os problemas citados anteriormente?

Pretende-se desenvolver um aplicativo para dinamizar e agilizar os processos que interferem no atendimento ao cliente, buscando automatizar e organizar processos que ainda são lentos e feitos manualmente, pode-se melhorar a experiência do cliente em eventos, já que as principais insatisfações encontradas hoje em dia são as perdas de tempo em filas desnecessárias.

1.2 Cronograma

1. DATA	DESCRIÇÃO
01/06/2019	Planejamento do Escopo do Trabalho.
15/06/2019	Análise e Definições do Escopo do Trabalho
01/07/2019	Levantamento de Requisitos
15/07/2019	Validação dos Requisitos
01/08/2019	Definições e Elaboração dos diagramas
15/08/2019	Definição e elaboração da estrutura do banco de dados
01/09/2019	Definições da Metodologia de Implementação
15/09/2019	Definições e Planejamento de Padrões da Interface Gráfica
01/10/2019	Definições de Ferramentas Utilizadas para Desenvolvimento
15/10/2019	Reunião Sobre Início da Implementação do Sistema
01/11/2019	Desenvolvimento da Interface Gráfica dos Sistema
15/11/2019	Validação das Interfaces do Sistema
01/12/2019	Início da Programação das interfaces
15/12/2019	Validação da interface desenvolvida na semana
01/01/2020	Início da Programação do Banco de Dados
15/01/2020	Validação do Banco de Dados
01/02/2020	Conexão das interfaces e do banco de dados
15/02/2020	Validação do sistema
07/03/2020	Teste e aprimoramento de Funcionalidade
15/03/2020	Teste e aprimoramento de segurança
01/04/2020	Teste e aprimoramento de desempenho
15/04/2020	Revisão dos requisitos funcionais e não funcionais
01/05/2020	Revisão da monografia
15/05/2020	Revisão do trabalho
26/06/2020	Entrega do TCC

1.3 Recursos

1.3.1 Recursos Humanos

Kauê Ribeiro (Programação): 16 horas semanais .

Felipe Pasianot (Documentação): 10 horas semanais.

1.3.2 Recursos Físicos

Utilizaremos ferramentas como o Visual Studio, WebServer.

Notebook Intel Core i5 4GM RAM.

2.ESPECIFICAÇÃO DOS REQUISITOS DO SISTEMA

2.1 O Produto

2.1.1 Proposito

Um Evento, sendo ele em um bar, restaurante, arena, ou espaços grandes, necessitam de filas para fazer o pedido e esperar em outra fila para poder tira-lo, nossa missão é poder trazer mais comodidade as pessoas que frequentam esses lugares, com isso, a implantação desse software ajudará na redução de grandes, agilidade no pedido, facilidade de compra, e organização de filas nos estabelecimentos.

2.1.2 Clientes

Nossos Clientes são todos aqueles que procuram um sistema para melhorar a experiencia de seus consumidores no seu evento. Um sistema que controla o estoque, realiza vendas de ingressos e produtos, e gera relatórios para ele com informações de vendas. Com isso, nosso cliente terá maior comodidade e menor preocupação com todo gerenciamento do seu negócio.

2.1.3 Usuários

O sistema conta com três tipos de usuário, sendo eles o administrador, o dono do negócio, e o usuário final.

O administrador, será responsável por realizar o controle do dono do negócio, cadastrando-o no sistema.

O dono do negócio, atuará com nosso aplicativo para cadastrar o seu evento, como preço, local, descrição e informações adicionais, ter o controle do estoque em tempo real e por fim receber relatórios das suas finanças adquiridas.

O usuário final, ele sim terá a maior interação com o sistema, pois ele realiza seu pedido e verifica em tempo real como está o andamento do preparo, e recebe uma notificação quando estiver pronto para ser retirado.

2.2 Escopo do Projeto

Nossa missão através desse produto é entregar para nossos clientes maior experiência em gerenciamento, sendo ele de pessoas, estoque, venda, espera e feedback dos consumidores. Conforme apresentado na descrição do projeto.

Esse projeto conta com diagramas e requisitos tirados de uma entrevista com um stakeholder, que concedeu seu tempo para tirar dúvidas sobre seu negócio e sobre as dificuldades enfrentadas com o público, com essas informações tivemos novas ideias e organizamos toda as informações para realizar um produto completo que supre todas as suas necessidades.

Pautas retiradas da entrevista:

- Como podemos reduzir a fila na hora de comprar o ingresso?

A pergunta é respondida fácil, pois hoje em dia existem várias empresas que trabalham com tecnologias de QR code. Mas nós pensamos em algo maior e mais rápido para identificar um comprador, sem o auxílio de um funcionário para autenticar um ingresso, é a leitura facial.

- Como podemos diminuir filas na hora de fazer um pedido?

Essa pergunta foi a mais difícil, pois hoje em dia, são raros os lugares que usam essa estratégia com os cliente, um atendimento rápido para que seu cliente saia satisfeito do local e possa aproveitar o evento, muitas empresas hoje estão preocupadas com o lucro somente, e deixa de lado o conforto com o cliente.

Com isso decidimos implementar algo inovador, iremos aplicar uma tecnologia de Qr code, para retirar o produto que pode ser pedido online pelo celular, do lugar que o consumidor estiver. Após a aprovação do pedido, nosso sistema irá gerar um código e com este, o consumidor pode ir até o balcão e retirar seu produto assim que o aplicativo avisar que está pronto, através de notificação. Com essa tecnologia implementada nos eventos, o Consumidor poderá aproveitar melhor o evento e ter mais comodidade ao realizar um pedido.

Juntamente com essa tecnologia, implementaremos o estoque inteligente, que após a compra de um pedido, o estoque é atualizado sozinho de forma rápida e prática, gerando um relatório no final para o cliente. Incluindo tudo que foi gasto, e a quantidade que falta, sem precisar contar as unidades no final do evento e fazer a contabilidade. Tudo rápido e prático.

Outra pergunta questionada pelo nosso stakeholder foi:

- Como o Consumidor pagará pelos itens consumidos, com ou sem o aplicativo?

É uma pergunta que pode variar de acordo com o gosto do cliente, pois hoje em dia é dividido, alguns lugares trabalham com comanda, uma espécie de cartão onde fica armazenado todo seu consumo, e no final é quitado em um guichê.

Outra forma também é pagar na hora do pedido, mas neste caso, a demora é um pouco maior, pois há um gerenciamento da fila por conta do tempo perdido para realizar o pagamento do produto.

Essas foram as principais questões levantadas na entrevista, a partir gerou os requisitos funcionais e os requisitos não funcionais.

2.3 Dos Requisitos

Segundo somerville, requisitos Funcionais de um sistema descreve o que ele deve fazer, depende do tipo de software e dos usuários que vão utilizar. Ou seja, não podem faltar, são extremamente importantes para que seja desenvolvido o software

Dos requisitos não funcionais, segundo Somerville, são requisitos que não estão diretamente ligados ao software e aos seus serviços específicos aos usuários, intendendo-se que são ferramentas que acrescentam indiretamente no software, agregando características específicas e qualidade.

Após discussões, chegamos à conclusão que as ferramentas importantes para o nosso software são aquelas que estão ligadas diretamente com cadastro, pagamento e controle de estoque do nosso sistema, são as principais ferramentas do nosso aplicativo e os diferenciais.

Os requisitos não funcionais, como explicado acima, são aqueles que agregam no nosso aplicativo, sendo eles de qualidade ou funcionalidade, entre outras categorias que somerville aborda.

2.3.1 Funcionais

ID do Requisito	Descrição	Caso de Uso Relacionado
R001	O Sistema deve ter a área de segurança de credenciais do usuário.	CSU001
R002	O Sistema deve realizar um pedido feito pelo usuário.	CSU003
R003	O sistema deve mostrar ao usuário a fase que está o pedido dele	CSU003
R004	O Sistema deve informar o usuário quando o pedido estiver pronto	CSU003
R005	O Sistema deve mostrar ao cliente todas as vendas realizadas no dia	CSU002

R006	O sistema deve controlar o estoque do estabelecimento, avisando o cliente quando o produto acabar	CSU003
R007	O Sistema deve cadastrar, excluir, e editar um cliente	CSU001
R008	O Sistema deve cadastrar, excluir, editar um evento	CSU002
R009	O Sistema deve validar todo cadastro	CSU001
R010	O Sistema deve validar todo login	CSU001
R011	O Sistema deve mostrar a cozinha o produto que está sendo pedido	CSU003

2.3.2 Não funcionais

ID do Requisito não funcional	Descrição	Caso de uso relacionado
RNF001	O Sistema deve ter mais de uma forma de pagamento	CSU004
RNF002	O Sistema deve ter o tempo de resposta com menos de 1 minuto para o acompanhamento do pedido	CSU003
RNF003	O Sistema deve mostrar de forma rápida e eficiente o cardápio.	CSU002
RNF004	O Sistema deve ser adaptado para IOS e Android	TODOS
RNF005	O Sistema deve pedir usuário e senha para qualquer tipo de usuário que interage com o sistema	CSU001
RNF005	O Sistema deve mostrar para a cozinha somente os pedidos realizados na determinada data	CSU003
RNF006	O Sistema deve mostrar através de uma notificação que o produto está pronto.	CSU003

2.4 Protótipo do Produto

2.4.1 Modelo de navegação

O modelo de navegação é responsável por mostrar ao leitor como será a navegabilidade do software, e o que cada usuário vai interagir com o software, a partir disso é possível realizar os protótipos de tela, pois temos uma noção de como vai funcionar o software, o que cada página pode trazer para o usuário e a sua interação com o software, de acordo com as suas escolhas.

2.4.1.1 Raia do Consumidor

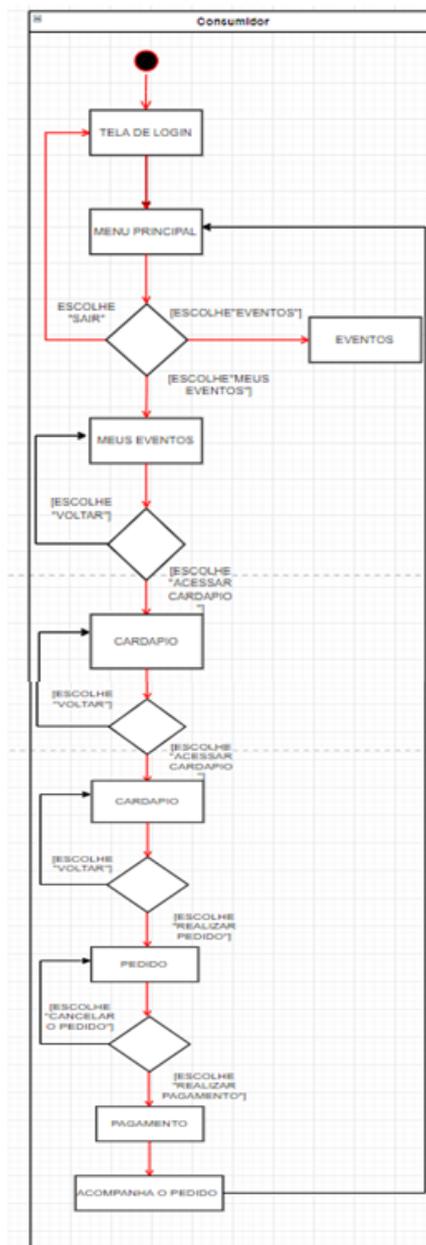


Imagem 01: Raia Consumidor

2.4.1.2 Raia do Cliente

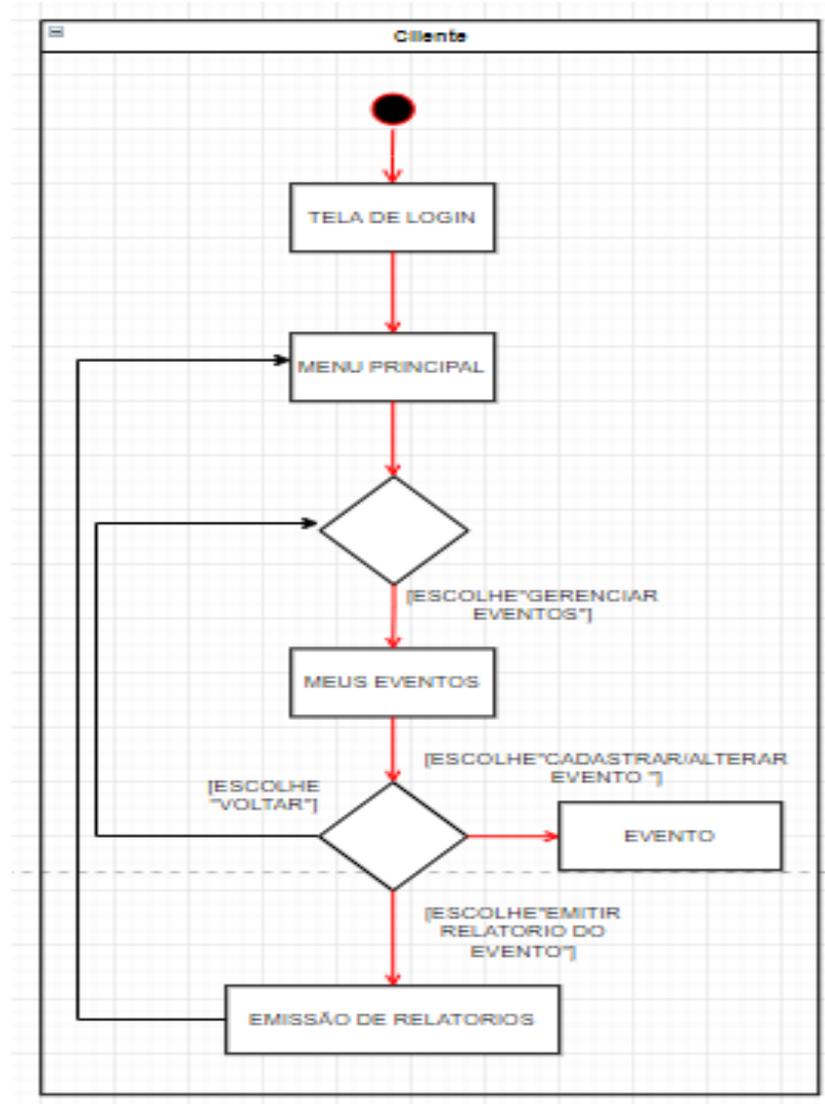


Imagem 02: Raia do Cliente

2.4.1.3 Raia do Administrador

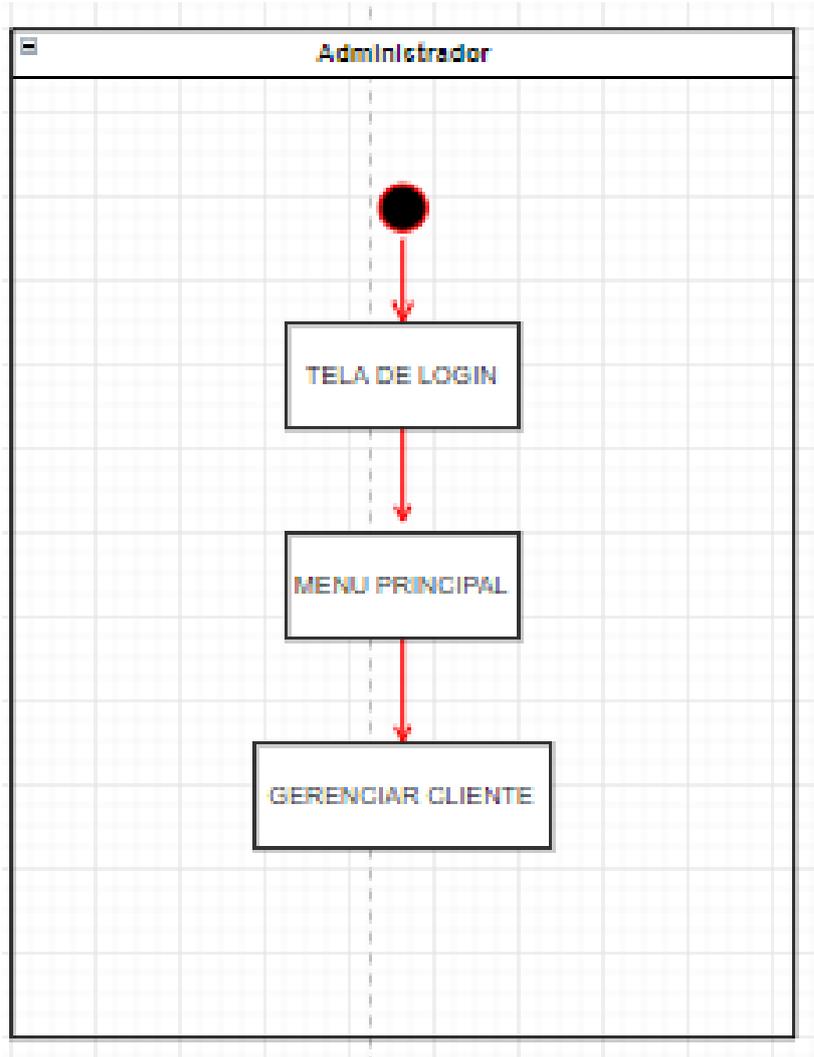


Imagem 03: Raia do Administrador

2.4.2 Protótipo das telas

Neste modulo temos os protótipos das telas.

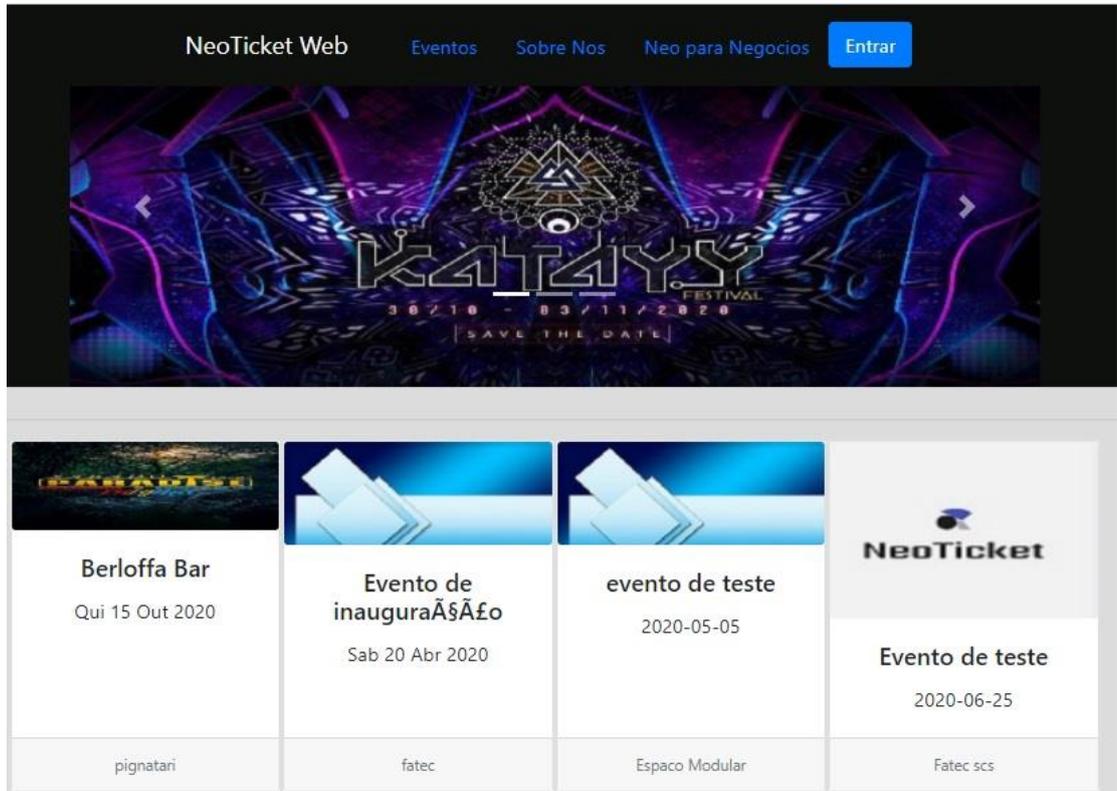


Imagem 04: Protótipo da Página Inicial

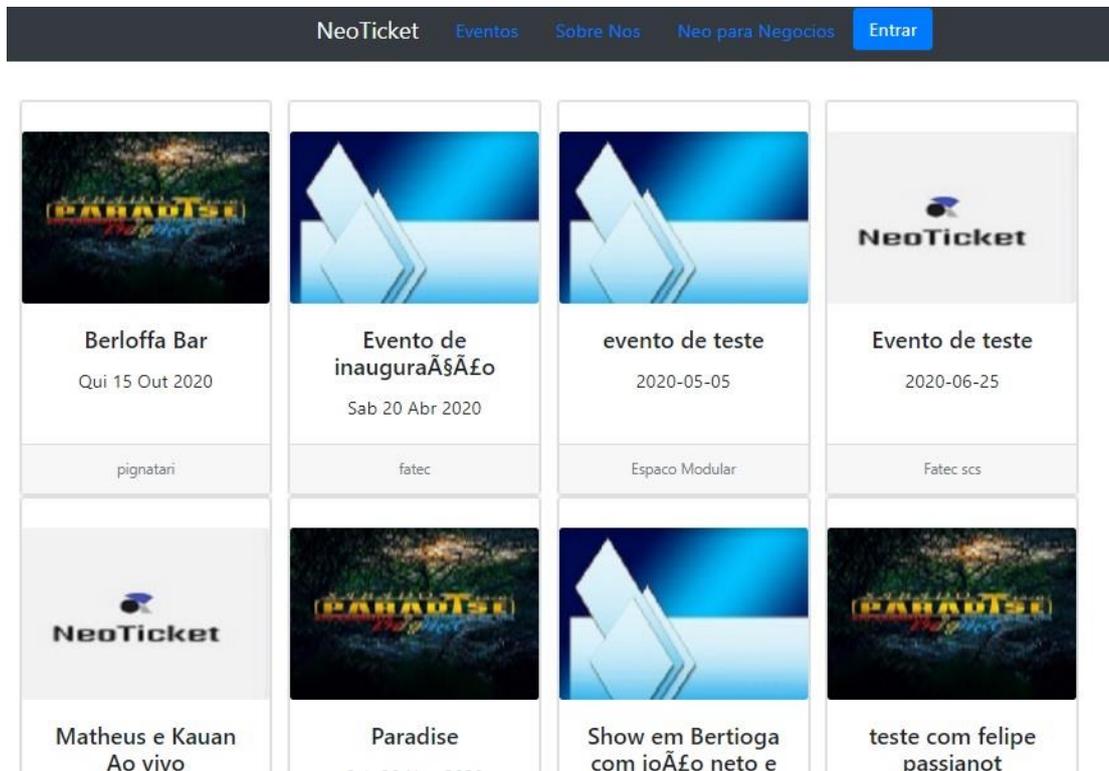


Imagem 05: Protótipo da Página de Eventos.

NeoTicket Eventos Sobre Nos Neo para Negocios Entrar

NeoTicket

Bem Vindo ao Cadastro

Nome: Primeiro Nome Último Nome

Email: root Confirme o Email:

Senha: Confirme a Senha:

Cpf: Ex.: 00000000000 Data de Nascimento: dd/mm/aaaa

Cadastrar

Imagem 06: Protótipo da Página de Cadastro

NeoTicket Eventos Sobre Nos Neo para Negocios Entrar

NeoTicket Web

Somos uma startup criada em 2019 na faculdade de tecnologia de São Caetano (FATEC Antonio Russo), com o intuito de agilizar o processo em eventos, tendo maior comodidade para o consumidor.

NeoTicket For business

Esta ferramenta serve para auxiliar o cliente a ter uma melhor gestão do seu negócio, podendo controlar estoque, ver pedidos sendo realizados por seus consumidores e, por fim, tendo o controle do seu evento e ingressos a toda hora.

NeoTicket Mobile

Temos outra ferramenta, a NeoTicket Mobile, que serve para os consumidores, este produto traz a eles uma nova experiência nos eventos parceiros, pois podem comprar, ver o cardápio e fazer pedidos por lá, quando tiverem no evento comprado pela...

ERP

01 - CONTROLE DE ESTOQUE
02 - CONTROLE DE VENDAS
03 - SERVIÇO DE ATENDIMENTO AO CLIENTE
04 - TELA ÚNICA
05 - SERVIÇO DE ENTREGA DE INGRESSOS
06 - SERVIÇO DE ENTREGA DE INGRESSOS
07 - SERVIÇO DE ENTREGA DE INGRESSOS
08 - SERVIÇO DE ENTREGA DE INGRESSOS

MELHORIA & PRODUTIVIDADE
OUTROS PROCESSOS
AUMENTA A LUCRATIVIDADE
100% ONLINE SEGURANÇA DAS INFORMAÇÕES

Imagem 07: Protótipo da Página Sobre Nós

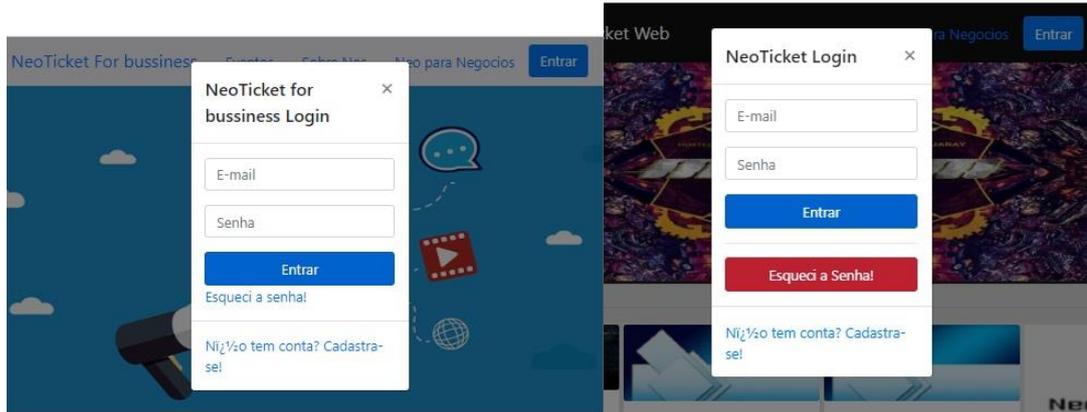


Imagem 08: Protótipo dos Logins

3 MODELO DE CASO DE USO DO SISTEMA

3.1 Diagrama de Caso de Uso

O Diagrama de Caso de uso representa o papel a ser realizado por cada personagem, no software, teremos a interação de três personagens como visto anteriormente.

O Cliente que terá interação com o sistema, no cadastro do seu evento, ao emitir relatórios e ao visualizar seus eventos.

A outra interação que este caso de uso nos mostra é a do Consumidor, que por sua vez, é o personagem que mais terá interação com o software. Ele pode visualizar o evento, comprar ingresso, visualizar cardápio, fazer pedido e por último realizar o pagamento de suas despesas.

E por fim, o administrador que visualiza e gerencia os clientes.

Conforme ilustra a figura abaixo é possível visualizar um pouco melhor os atores e suas responsabilidades.

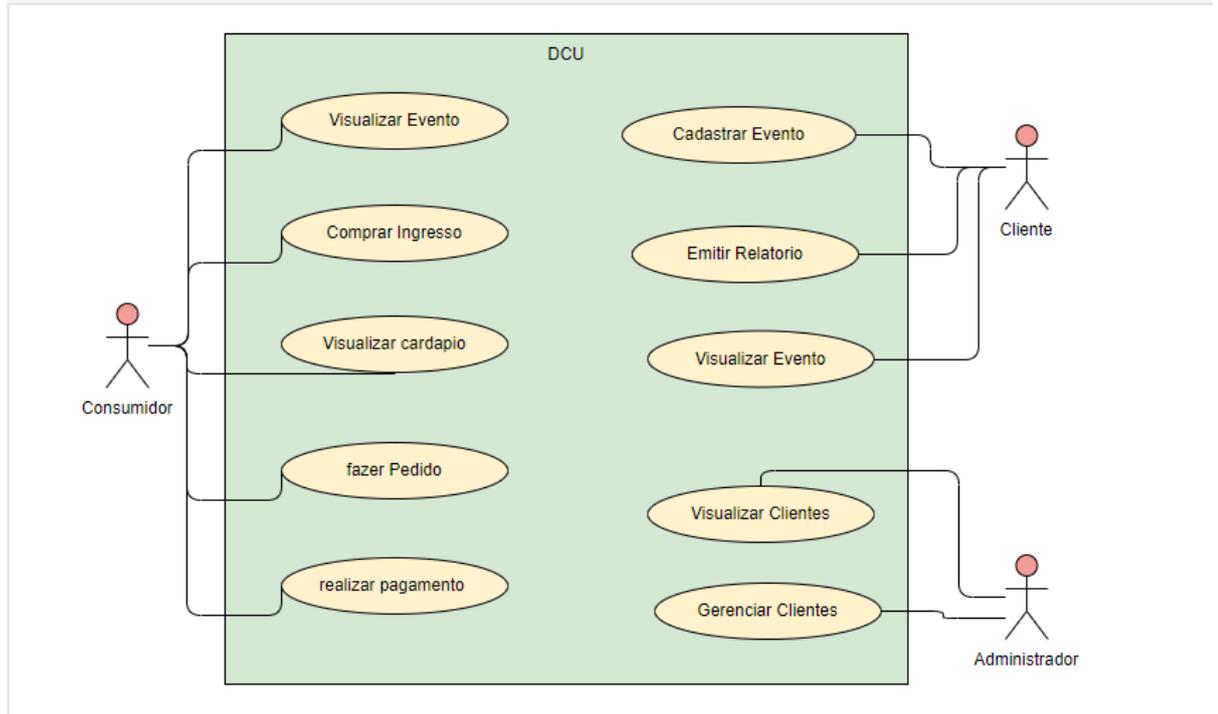


Imagem 09: Diagrama de Caso de Uso

3.2 Documentação dos Atores

Documentação dos atores é uma parte importante no desenvolvimento de um software, porque nele é descrito, cada caso que ocorrerá na utilização do aplicativo de acordo com cada ator.

3.2.1 Ator Consumidor

Ator: Consumidor
ID:001
Resumo: O Consumidor Poderá Visualizar Eventos e realizar Compras dentro do Sistema
Responsabilidade: Visualizar e Realizar Compras quando necessário.
Ambiente Físico: Ele terá que estar no evento para realizar compras, ou em qualquer lugar, somente para compras do ingresso, desde que tenha conexão com a internet.
Número e tipo: Não Há limite de consumidores
Frequência com que uso o Sistema: Diariamente

3.2.2 Ator Cliente

Ator: Cliente
ID: 001
Resumo: O Consumidor Poderá Visualizar Eventos e realizar o Gerenciamento de seus eventos dentro do Sistema
Responsabilidades: Visualizar e gerenciar seu evento
Ambiente Físico: Não há, ele poderá ter acesso ao gerenciamento de qualquer lugar, basta estar com internet.
Número e tipo: Não Há limite de clientes
Frequência com que uso o Sistema: Diariamente

3.2.3 Ator Administrador

Ator: Administrador
ID: 001
Resumo: O Administrador poderá gerenciar todos os clientes e emitir relatórios sobre suas atividades.
Responsabilidade: O administrador deve gerenciar os clientes
Ambiente Físico: Não há, o Administrador poderá gerenciar de qualquer lugar, basta estar com internet.
Número e tipo: Não Há limite de 3 administradores a cada 100 Clientes.
Frequência com que uso o Sistema: Diariamente

3.3 Documentação dos Casos de Usos:

3.3.1 Acesso ao Sistema

Caso de uso: Acesso ao Sistema
ID: CSU001
Resumo: O Sistema deverá ter login e senha para todos os níveis de usuários que desejam ter acesso.
Atores primários: Todos os usuários
Atores Secundários: Não Existe.
Pré-Condição: Estar cadastrado no sistema
Fluxo de Evento Principal: <ol style="list-style-type: none"> 1. O Sistema solicita login e senha 2. O sistema valida as credenciais recebidas do usuário 3. O sistema permite o acesso do usuário 4. O caso de uso termina
Pós condição: Acesso ao Sistema
Fluxo de eventos alternativos: <p>Fluxo 1A: Credenciais invalidas</p> <ol style="list-style-type: none"> a) Caso o login e senha sejam inválidos o sistema não permite o acesso. b) Sistema informa, login e senha inválidos, digite novamente. <p>Fluxo 2A: Usuário não possui cadastro</p> <ol style="list-style-type: none"> A. Sistema não consegue localizar o usuário B. Sistema informa que o usuário não possui cadastro, e informa o mesmo a realizar um. C. volta ao passo do fluxo principal

3.3.2 Comprar Ingresso

Caso de uso: Comprar Ingresso
ID: CSU002
Resumo: O Sistema deve permitir que o usuário realize a compra de ingresso.
Atores primários: Consumidor
Atores Secundários: Não Existe.
Pré-Condição: Estar cadastrado no sistema
<p>Fluxo de Evento Principal:</p> <ol style="list-style-type: none"> 1. O Sistema deve mostrar todos os eventos disponíveis 2. O consumidor escolhe qual evento deseja visitar 3. O sistema informa o preço e horário 4. O Consumidor confirma os dados 5. Sistema solicita dados do cartão do consumidor 6. O consumidor informa os dados 7. O sistema valida os dados inseridos pelo Consumidor 8. O consumidor recebe informações sobre seu ingresso adquirido (número, preço, local, hora). 9. O caso de uso termina.
Pós condição: Acesso ao cardápio do estabelecimento
<p>Fluxo de eventos alternativos:</p> <p>Fluxo 1A: Erro ao realizar a transação com o banco</p> <ol style="list-style-type: none"> A. O Sistema informa ao Consumidor que não foi possível concluir a transação. B. Sistema solicita ao Consumidor confirmar os dados ou trocar o cartão. C. O Fluxo volta ao passo 6 do fluxo principal

3.3.3 Comprar Produtos no Evento

Caso de uso: Consumo
ID: CSU003
Resumo: O Sistema deve permitir que o Consumidor realize a compra de produtos dentro do evento.
Atores primários: Consumidor
Atores Secundários: Não Existe.
Pré-Condição: Estar no evento
<p>Fluxo de Evento Principal:</p> <ol style="list-style-type: none"> 1. O Sistema deve mostrar todos os produtos disponíveis naquele ambiente. 2. O consumidor escolhe qual produto deseja comprar 3. O sistema informa o preço e detalhe 4. O Consumidor confirma os dados 5. Sistema solicita dados do cartão do consumidor 6. O consumidor informa os dados 7. O sistema valida os dados inseridos pelo Consumidor 8. O consumidor recebe informações sobre seu ingresso adquirido (número, preço, local, hora). 9. O caso de uso termina.
Pós condição: Acesso ao cardápio do estabelecimento
<p>Fluxo de eventos alternativos:</p> <p>Fluxo 1A: Erro ao realizar a transação com o banco</p> <ol style="list-style-type: none"> A. O Sistema informa ao Consumidor que não foi possível concluir a transação. B. Sistema solicita ao Consumidor confirmar os dados ou trocar o cartão. C. O Fluxo volta ao passo 6 do fluxo principal

3.3.4 Encerrar a comanda

Caso de uso: Encerrar a Comanda
ID: CSU003
Resumo: O Sistema deve permitir que o Consumidor realize o pagamento de sua dívida.
Atores primários: Consumidor
Atores Secundários: Não Existe.
Pré-Condição: escolher pagamento em dinheiro ao realizar uma compra
<p>Fluxo de Evento Principal:</p> <ol style="list-style-type: none"> 1. O Sistema deve mostrar todos os produtos comprados pelo Consumidor naquele local. 2. O Consumidor escolhe realizar pagamento em dinheiro 3. O sistema informa o número da comanda pertencente, total. 4. O sistema informa para o Consumidor dirige-se até um guichê para realizar o pagamento em dinheiro. 5. O Sistema informa a quitação da dívida após ela ser quitada no guichê. 6. O caso de uso termina
Pós condição: pagamento já quitado
<p>Fluxo de eventos alternativos:</p> <p>Fluxo 1A: Consumidor já pagou 100% da sua dívida</p> <ol style="list-style-type: none"> A. O Sistema informa ao Consumidor que a dívida já está quitada e informa um número para deixar o local. B. Volta ao passo 6 do fluxo principal.

3.3.5 Cadastrar Evento

Caso de uso: Cadastrar Evento
ID: CSU005
Resumo: O Sistema deve permitir que o Cliente realize o cadastro do seu evento.
Atores primários: cliente
Atores Secundários: Não Existe.
Pré-Condição: estar cadastrado no sistema
<p>Fluxo de Evento Principal:</p> <ol style="list-style-type: none"> 1. O Cliente deve solicitar o cadastro de um evento 2. O sistema deve solicitar informações do evento, como, data, hora, local, preço e cardápio. 3. O sistema deve informar ao Cliente que o cadastro foi realizado 4. O caso de uso termina
Pós condição: Disponibilidade de novo evento para todos os Consumidores
<p>Fluxo de eventos alternativos:</p> <p>Fluxo 1A. Já existe um evento com o mesmo nome</p> <ol style="list-style-type: none"> A. O sistema informa ao usuário que o nome escolhido já existe. B. O sistema informa ao Cliente escolher outro nome para o evento. C. Volta ao passo 2 do fluxo principal

4. MODELOS DE ANÁLISE DO SISTEMA

4.1 Especificação das classes de análise do sistema

4.1.1 Visão geral dos pacotes de análise

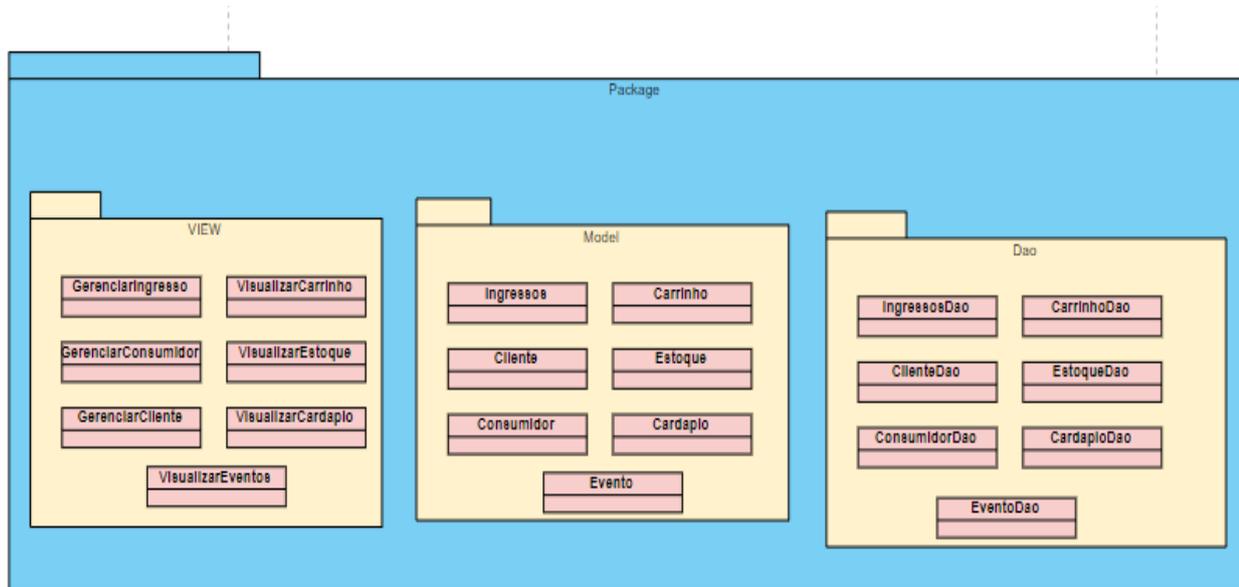


Imagem 10: Diagrama de Pacotes de Análise.

4.1.1.1 Pacotes

DAO - O pacote DAO serve para organizar a hierarquia das classes DAO. As classes DAO servem para ler e gravar a partir da origem de dados, elas são responsáveis por realizar a persistência no banco de dados.

Model – O pacote model é utilizado para realizar os getters e setters dos atributos e invocar um método construtor.

View – O pacote view é onde será implementado o código junto da interface gráfica que será apresentada ao usuário. Por exemplo: na view é apresentada uma interface para Login que recebe informações das classes model e DAO e então implementa os métodos em tempo de execução.

4.1.2 Diagramas e especificação das classes

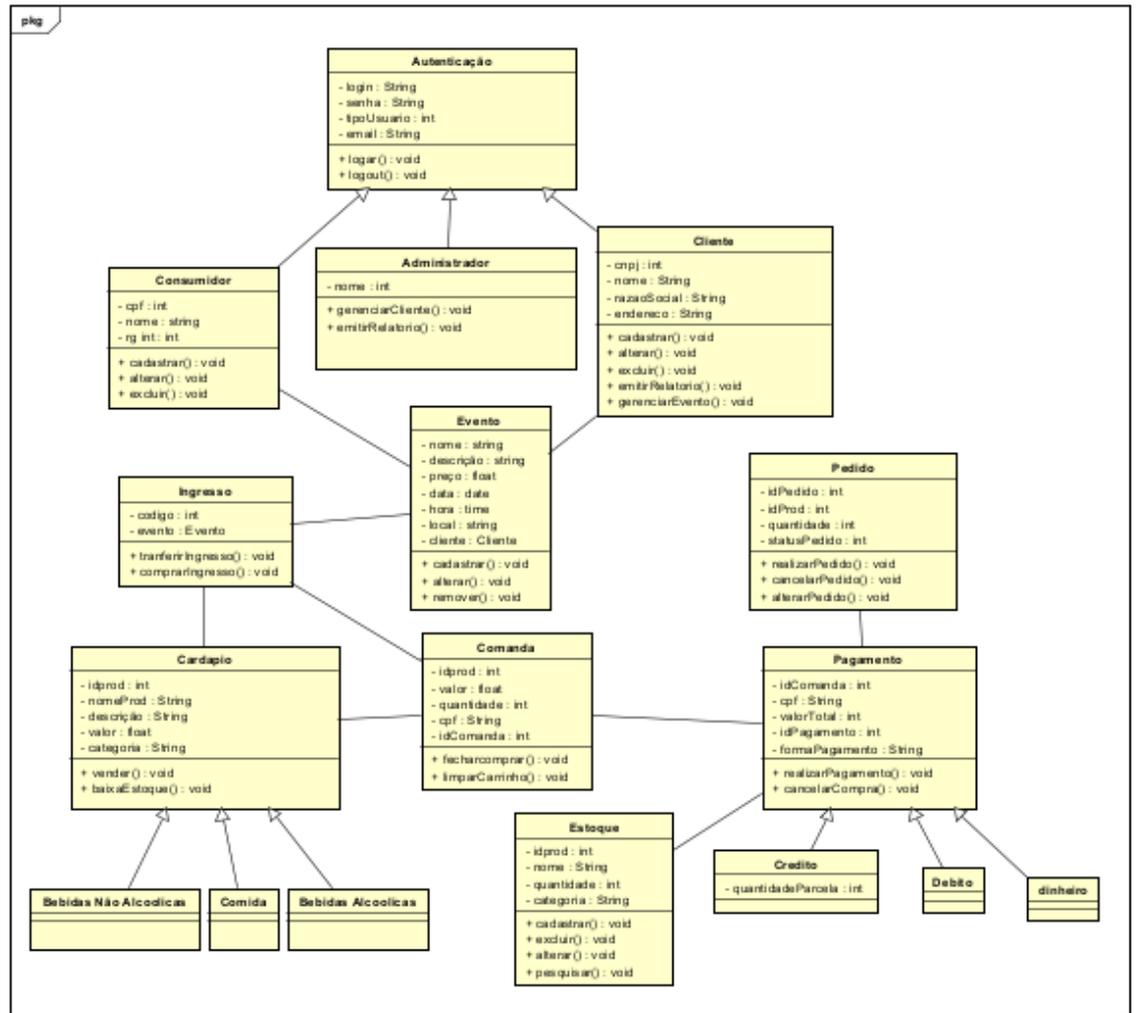


Imagem 11: Diagrama de Classe

4.1.2.1 Classe Cliente

Descrição:

A classe Cliente representa aos Clientes donos dos eventos.

Superclasse(s): Autenticação

Subclasse(s): Nenhuma

Atributos:

Nome:	CNPJ
Descrição:	Representa o CNPJ do cliente
Tipo:	Int
Domínio:	Qualquer Número entre 0 e 9

Nome:	Nome
Descrição:	Representa o nome do cliente
Tipo:	String
Domínio:	Qualquer Algarismo entre A e Z

Nome:	RazaoSocial
Descrição:	Representa a razão Social do Cliente
Tipo:	String
Domínio:	Qualquer Algarismo entre A e Z

Nome:	Endereco
Descrição:	Representa o endereço do cliente
Tipo:	String
Domínio:	Qualquer alfanuméricos entre A e Z e 0 e 9

Nome:	Senha
Descrição:	Representa a Senha do Consumidor
Tipo:	String
Domínio:	Qualquer alfanuméricos entre A e Z e 0 e 9

4.1.2.2 Classe Consumidor

Descrição:

A classe Consumidor representa aos consumidores dos eventos, o público alvo que irá comprar os ingressos dos eventos

Superclasse(s): Autenticação

Subclasse(s): Nenhuma

Atributos:

Nome:	CPF
Descrição:	Representa o CPF do Consumidor
Tipo:	Int
Domínio:	Qualquer Número entre 0 e 9

Nome:	Nome
Descrição:	Representa o nome do Consumidor
Tipo:	String
Domínio:	Qualquer Algarismo entre A e Z

Nome:	RG
Descrição:	Representa o RG do Consumidor
Tipo:	Int.
Domínio:	Qualquer alfanuméricos entre A e Z e 0 e 9

4.1.2.3 Classe Administrador

Descrição:

A classe Administrador representa aos consumidores dos eventos, o público alvo que irá comprar os ingressos dos eventos

Superclasse(s): Autenticação

Subclasse(s): Nenhuma

Atributos:

Nome:	Nome
Descrição:	Representa o nome do administrador
Tipo:	String
Domínio:	Qualquer algarismo entre A e Z

4.1.2.4 Classe Autenticação

Descrição:

A classe Autenticação representa a Autenticação dos usuários no sistema, sendo eles Administrador, Consumidor ou Cliente

Superclasse(s): Nenhuma

Subclasse(s): Cliente, Consumidor, Administrador

Atributos:

Nome:	nome
Descrição:	Representa o nome do administrador
Tipo:	string
Domínio:	Qualquer algarismo entre A e Z

Nome:	nome
Descrição:	Representa o nome do administrador
Tipo:	string
Domínio:	Qualquer algarismo entre A e Z

Nome:	nome
Descrição:	Representa o nome do administrador
Tipo:	string
Domínio:	Qualquer algarismo entre A e Z

Nome:	nome
Descrição:	Representa o nome do administrador
Tipo:	string
Domínio:	Qualquer algarismo entre A e Z

4.2 Realização de Casos de Uso

O objetivo nesta seção é demonstrar a simulação dos casos de uso pelas classes descobertas na análise. Cada caso de uso representa um cenário, no qual pretende-se mostrar como funciona a interação das classes em tempo de execução, o mais parecido com a realidade dos objetos. O diagrama utilizado abaixo é conhecido como diagrama de sequência, ele é responsável por mostrar a interação das classes em cada etapa do software.

4.2.1 Cenário do Consumidor

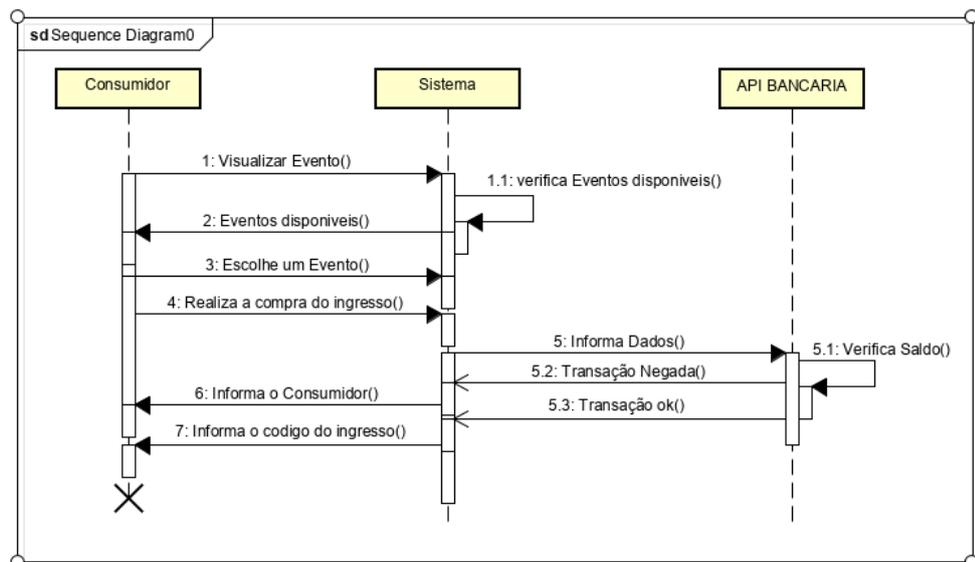


Imagem 12: Diagrama de Sequência Consumidor

4.2.2 Cenário do Cliente

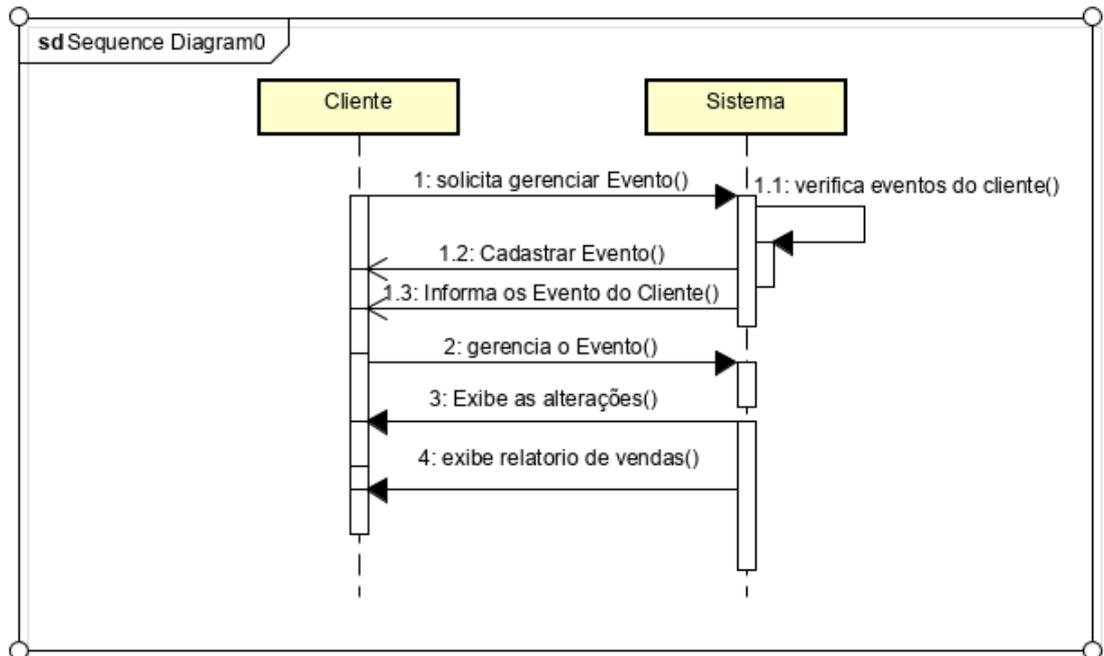


Imagem 13: Diagrama de Sequência Cliente.

4.2.3 Cenário Administrador

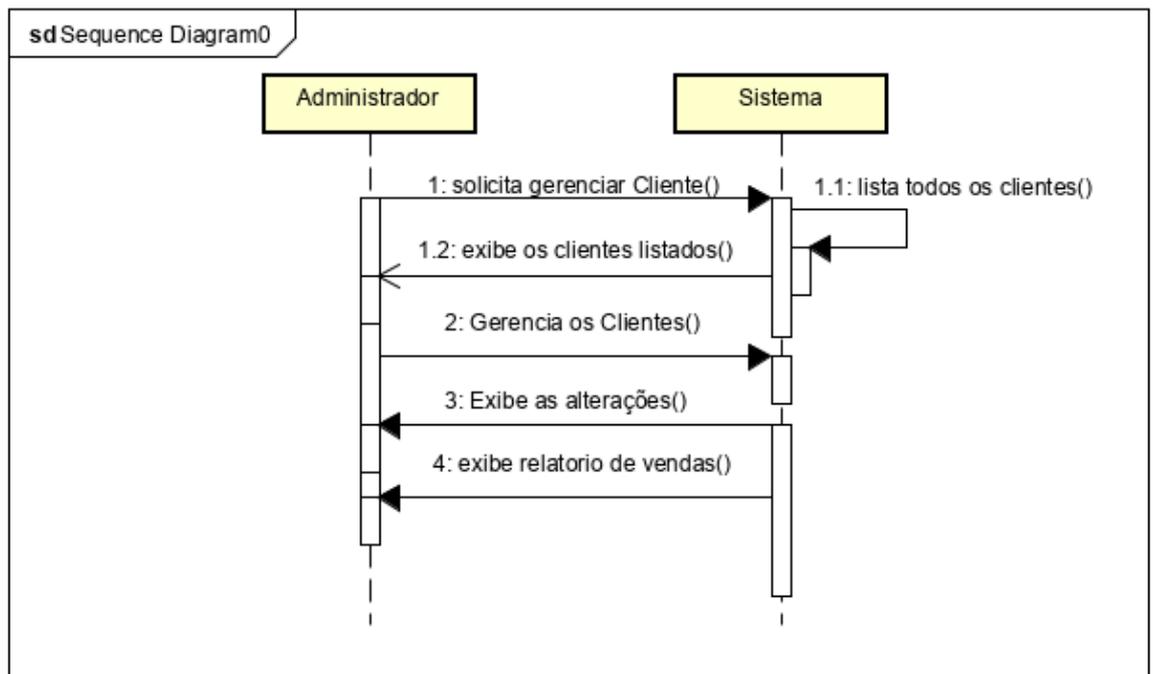


Imagem 14: Diagrama de Sequência Administrador.

5.MODELOS DE PROJETO DO SISTEMA

5.1 Classes de Projeto

O objetivo aqui é, explicar as classes de domínio do sistema e justificar seu uso no sistema, utilizou-se a linguagem UML, que são padrões para desenvolver diagramas de classes orientado a objeto. Classe de análise orientada a objetos, essa parte, representa ao “design” do projeto e todas suas funcionalidades na implementação.

Herança: a herança é o tipo mais forte de acoplamento que existe, com isso, sabendo-se que quanto maior o acoplamento menor a flexibilidade. Mas as três heranças usadas na figura 7 do projeto, possuem características únicas. Como por exemplo a classe Pagamento, ela tem várias formas de pagamento, tendo valores diferente ao utilizar elas, aplicando-se juros(se for crédito) ou descontos(dinheiro ou debito) com isso é utilizado o princípio de Liskov, um conceito importante na engenharia de software que fala sobre o polimorfismo, poli = muito, morfismo = forma, ou seja, muitas formas de implementar as filhas de uma superclasse, como o exemplo a cima, já a outra classe que apresenta herança que é a classe cardápio, pode ser substituída ou excluída e deixar os produtos somente com categoria, assim saberemos quando ele for da categoria do não alcoólicos, comida, entre outras categorias.

Segundo o que dito acima, eliminamos uma herança, retiramos um acoplamento forte, com isso, o software ficou mais flexível, com uma maneabilidade melhor e com melhor desempenho.

5.2. Refinamento dos relacionamentos

Esse tópico consiste em refinar os relacionamentos do modelo de análise de modo que posteriormente seja implementado na linguagem utilizada que é o *c#*, uma linguagem que se permite realizar o desenvolvimento do software voltado para os usuários de Android e IOS.

Ao realizar um refinamento na figura 7 deste projeto, chegou-se a conclusão que seria mais adequado e vantajoso realizar algumas alterações nas classes. Com isso surgiu outro diagrama, conforme mostrado na figura abaixo.

Ele foi utilizado para realizar a implementação do sistema pois com o refinamento podemos ter uma visão melhor de como será o funcionamento do software, usou-se dependência estruturada para as classes do tipo model que interagem com as outras classes desse tipo, e para dependência não estrutural que foi utilizada para as classes controles, foram implementadas no código passando as informações por parâmetros, sendo eles variáveis locais ou globais.

As classes view, no padrão utilizado que é o mvc, servem exatamente para representarem as interações gráficas com o usuário, elas possuem a parte em que ocorre as ações, sendo elas por botões ou por processos do sistema. Com isso é invocado o método da classe controle, que serve justamente para executar as funções e buscar as informações na model, a classe controle fica a maior parte do processamento do sistema, pois é ela que realiza as funções do sistema, as ações que são essenciais para o sistema realizar.

Foram escolhidos a relação de composição entre a classe view e a controle pois se não estiver a classe controle para fazer o gerenciamento é perdido a view pois não tem ação!

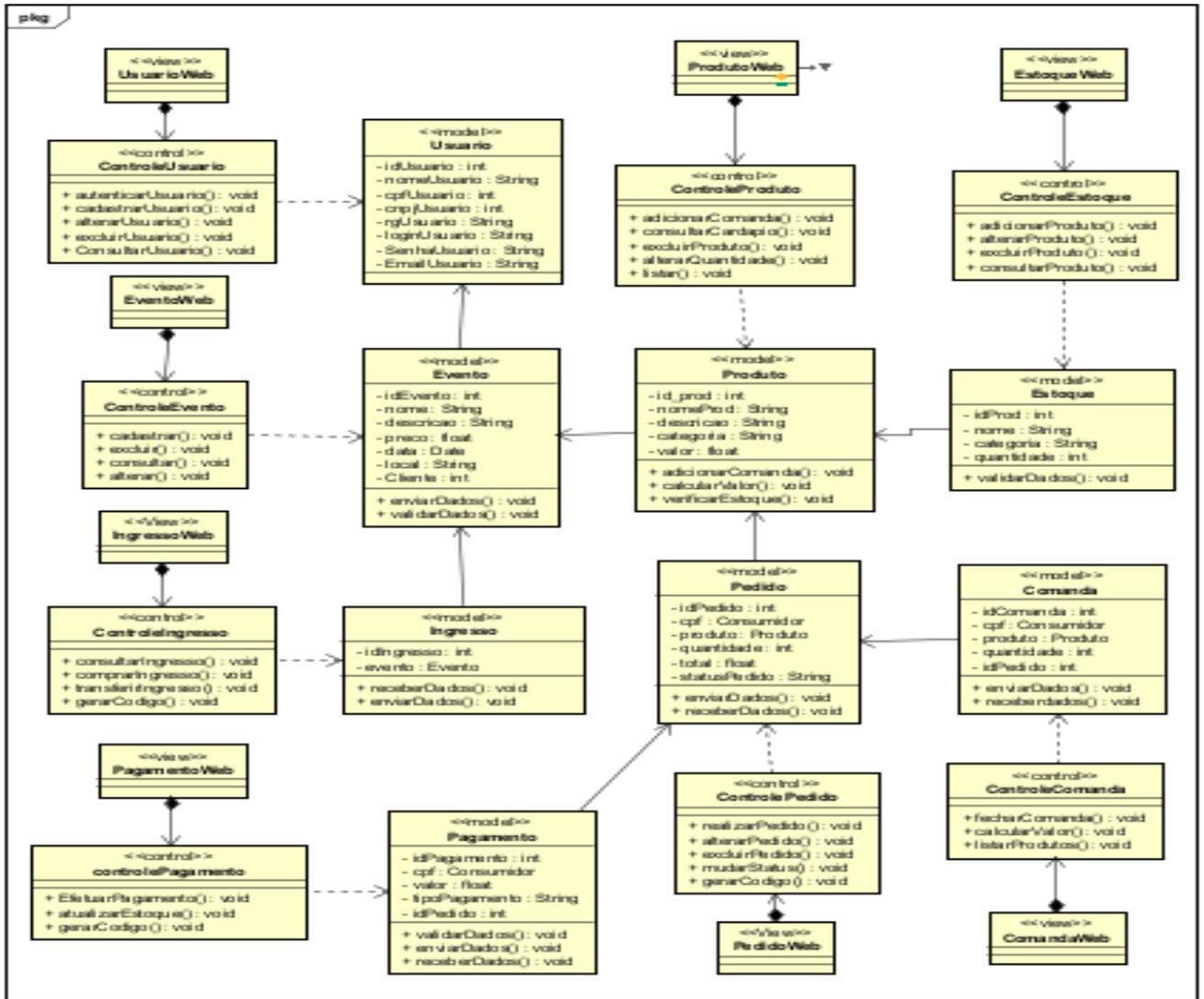


Imagem 15: Diagrama de Classe Refinado

6. MODELOS DE IMPLEMENTAÇÃO E IMPLANTAÇÃO DO SISTEMA

6.1. PROGRAMAÇÃO DO SISTEMA

6.2. Testes

Os testes foram realizados em todas as funções do sistema e não foi encontrado nenhum tipo de falha.

6.3. Elaboração do procedimento de instalação

O processo de instalação é simples, você encontra o aplicativo no Google Play(caso for android) ou AppleStore(caso seja IOS).

Após o download da aplicação já estará instalado no seu sistema o aplicativo, uma vez feito isso, só seguir os passos de manuseio do sistema.

Conclusões Finais

Este trabalho relata o processo de desenvolvimento de uma solução de gerenciamento de eventos, denominada Neoticket. O sistema resultante deste trabalho poderá facilitar o cotidiano dos clientes de eventos em São Paulo. Se corretamente utilizado, o Neoticket permite que dentro dos eventos tenha visibilidade em um meio eletrônico. Além disso, também é possível controlar todos os pedidos feitos em um único meio, fornecendo integridade nas informações e controle gerencial.

Os clientes, por sua vez, têm a possibilidade de realizar pedidos sem problemas causados por comunicação oral (falta de entendimento entre atendentes e clientes) e sem ter que aguardar em filas para realizar o pedido. Também é fornecido um meio de acompanhar o andamento do pedido.

REFERÊNCIAS BIBLIOGRÁFICAS

NIEDERAUER, Juliano. **Desenvolvendo websites com PHP: aprenda a criar Websites dinâmicos e interativos com PHP e bancos de dados**. Novatec Editora, 2017.

DE LIMA MORAES, André Luiz et al. Flexibilização do ensino de cálculo de esforços a partir do uso de protótipo e software em linguagem PHP em cursos de Engenharia. **Revista Brasileira de Ensino Superior**, v. 3, n. 3, p. 114-127, 2017.

PORTUGAL, Flávia Batista et al. Qualidade de vida em pacientes da atenção primária do Rio de Janeiro e São Paulo, Brasil: associações com eventos de vida produtores de estresse e saúde mental. **Ciência & Saúde Coletiva**, v. 21, p. 497-508, 2016.

RIBEIRO, LEANDRO. O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML.

IAN SOMMERVILLE. Engenharia de Software. 8ª edição Venezuela: Editora Pearson Education, 2008.

CAELUM. Apostila Desenvolvimento WEB com HTML, CSS e JavaScript.