

UNIVERSIDADE FEDERAL DO PARANÁ
AMANDA NAITO ALVES CHAGAS
HARRISON BRUNNO SANTOS
MARCELO BENIGNO FELTRAN

**SISTEMA GERENCIADOR DE COMANDAS:
MELLOW SUÍTE**

CURITIBA
2017

AMANDA NAITO ALVES CHAGAS
HARRISON BRUNNO SANTOS
MARCELO BENIGNO FELTRAN

**SISTEMA GERENCIADOR DE COMANDAS:
MELLOW SUÍTE**

Trabalho de conclusão de curso do curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná.

Orientador: Prof. Me. Dr. João Eugênio Marynowski.

CURITIBA
2017

TERMO DE APROVAÇÃO

AMANDA NAITO ALVES CHAGAS
HARRISON BRUNNO DOS SANTOS
MARCELO BENIGNO FELTRAN

MELLOW SUÍTE: SISTEMA GERENCIADOR DE COMANDAS

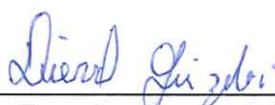
Trabalho apresentado como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas da Universidade Federal do Paraná.

Curitiba, 07 de Dezembro de 2017.

BANCA EXAMINADORA



Orientador: Prof. João Eugenio Marynowski
SEPT/UFPR



Examinador: Prof. Dieval Guizelini
SEPT/UFPR



Examinador: Prof. Mauro A Castro
SEPT/UFPR

Dedicamos os esforços aplicados na elaboração deste trabalho primeiramente a todos aqueles que, de alguma forma, incentivaram na busca por essa realização. Aos nossos pais e demais familiares pelo apoio recebido nos momentos difíceis no decorrer do curso, assim como pela compreensão nos momentos de ausência. E aquelas pessoas insubstituíveis que perdemos nessa trajetória.

AGRADECIMENTOS

Agradecimento aos nossos colegas de sala pelo convívio que nos permitiu crescer, aos professores que nos guiaram nessa trajetória. À equipe pelo trabalho em conjunto. Aos nossos pais, irmãos e à Universidade Federal do Paraná, pelo incentivo aos diversos saberes. E por fim, mas não menos importante, aos nossos companheiros e amigos que nos aturaram nos momentos de estresse.

“O sucesso nasce do querer, da determinação e persistência em se chegar a um objetivo. Mesmo não atingindo o alvo, quem busca e vence obstáculos, no mínimo fará coisas admiráveis.”

José de Alencar

RESUMO

Este trabalho visa realizar a descrição e desenvolvimento de um sistema cuja intenção é assessorar o gerenciamento de entradas e saídas de clientes em estabelecimentos comerciais. Através da criação e automatização de comandas por usuários previamente cadastrados, torna-se possível que clientes acompanhem, em tempo real, a quantidade consumida e verifiquem as opções disponibilizadas no cardápio do estabelecimento. Este procedimento pode ser realizado via smartphone, desde que esteja conectado à internet. Ao analisar esse campo, nota-se uma espera excessiva na realização destas ações, tornando-se prejudicial tanto aos clientes, que desejam usufruir do estabelecimento, quanto ao estabelecimento, na realização de cadastros e fechamento de contas. O intuito do Mellow Suite é prevenir essa possível deficiência, otimizando o tempo de obtenção dos dados de novos clientes, análise de lançamentos e pedidos realizados e facilitar o fechamento e faturamento de contas, diminuindo a formação de filas de espera. O trabalho de conclusão de curso proposto tem como objetivo, também, demonstrar as mais novas tecnologias e métodos de desenvolvimento, disponíveis em várias áreas da programação e análise de sistemas. Para isso, serão utilizadas tecnologias gratuitas e de grande reconhecimento.

Palavras-chave: fila, estabelecimento, comanda, automatização.

ABSTRACT

This work aims to perform the description and development of a system whose purpose is to advise the management of entrances and exits of clients in commercial establishments. Through the creation and automation of tabs by previously registered users, it is possible for customers to monitor the quantity consumed in real time and check the options available on the menu of the establishment. This procedure can be performed by smartphone, as long as it is connected to the internet. When analyzing this field, there is an excessive wait in the accomplishment of these actions, becoming detrimental both to the clients, who wish to enjoy the establishment, as to the establishment, in the accomplishment of registers and closure of accounts. Mellow Suite is designed to prevent this potential deficiency by optimizing the time it takes to obtain new customer data, review launched and completed orders, and facilitate account closure and billing, thereby reducing the formation of waiting queues. The proposed course completion work also aims to demonstrate the latest technologies and development methods available in various areas of programming and systems analysis. For this, free and highly recognized technologies will be used.

Keywords: queue, business, tab, automation.

LISTA DE FIGURAS

Figura 4.1: Menu de contexto do WampServer.....	35
Figura 4.2: Arquivo do modelo.	36
Figura 4.3: Menu Database do MySQL Workbench.	36
Figura 4.4: Arquivo de configuração do banco de dados.	37
Figura 4.5: Tela de login no sistema.	38
Figura 4.6: Tela de cadastro de estabelecimentos.....	39
Figura 4.7: Tela de verificação de abertura de comanda.	39

LISTA DE TABELAS

Tabela 3.1: Requisitos não funcionais.....	22
Tabela 3.2: Requisitos funcionais.....	22
Tabela 3.4: Aplicações da suíte.....	23
Tabela 3.5: Lista de atividades.....	26
Tabela 3.6: Lista de responsabilidades.....	28
Tabela 4.1: Usuários e senhas padrões.....	38

LISTA DE ABREVIATURAS E SIGLAS

AJAX – Asynchronous JavaScript and XML

CGI – Common Gateway Interface

CRUD – Create, Read, Update, Delete

CSS – Cascading Style Sheets

HTML – HyperText Markup Language

IDE – Integrated Development Environment

MER – Modelo Entidade-Relacionamento

MVC – Model-view-controller

PHP – PHP: Hypertext Preprocessor

PHP/FI – PHP: Hypertext Preprocessor/ Form Interpreter

SQL – Structured Query Language

UPFR – Universidade Federal do Paraná

UML – Unified Modeling Language

VIP – *Very Important Person*

WAMP – Windows, Apache, MySQL e PHP

WBS – Work Breakdown Structures

XML – Extensible Markup Language

SUMÁRIO

1	INTRODUÇÃO.....	14
1.1	CONTEXTUALIZAÇÃO	14
1.2	PROBLEMATIZAÇÃO	14
1.3	OBJETIVOS DO PROJETO	15
1.3.1	Objetivos Gerais	15
1.3.2	Objetivos Específicos.....	15
1.4	PROPOSTA.....	15
2	FUNDAMENTAÇÃO TEÓRICA.....	17
2.1	CONCEITOS.....	17
2.1.1	Estabelecimento comercial.....	17
2.1.2	Bar e discoteca	17
2.1.3	Comanda	18
2.2	PROCESSO ATUAL	18
2.3	PÚBLICO ALVO.....	19
2.4	MERCADO	19
3	ANÁLISE E METODOLOGIA.....	21
3.1	MODELO DE PROCESSO DE ENGENHARIA DE SOFTWARE	21
3.2	REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS.....	21
3.3	PERFIS DE USUÁRIO.....	23
3.4	MODELAGEM DOS DADOS	23
3.5	PLANO DE ATIVIDADE	24
3.6	PLANO DE RISCOS	26
3.7	RESPONSABILIDADES	26
3.8	FERRAMENTAS E MATERIAIS	28
3.8.1	Astah.....	28
3.8.2	Ajax	28
3.8.3	Bitbucket	29
3.8.4	Codelgniter.....	29
3.8.5	CSS 3	29
3.8.6	Git.....	30
3.8.7	Google Material.....	30

3.8.8 HTML 5	30
3.8.9 JQuery	31
3.8.10 PHP 7.0	31
3.8.11 QRCode	32
3.8.12 MVC	32
3.8.13 MySQL	32
3.8.14 MySQL Workbench	33
3.8.15 Trello	33
3.8.16 UML	33
3.8.17 WampServer 3.0	34
4 APRESENTAÇÃO DO SOFTWARE	35
4.1 INSTALAÇÃO	35
5 CONSIDERAÇÕES FINAIS	40
REFERÊNCIAS	42
APÊNDICES	45

1 INTRODUÇÃO

Com a homologação da lei municipal 13.407/2009, que "dispõe sobre a obrigatoriedade da identificação dos frequentadores de casas noturnas e similares localizadas no Município de Curitiba e dá outras providências", um grande de formação de enormes filas de entrada e saída foi gerado. Fato que ainda pode ser identificado atualmente. Este trabalho trata da descrição e do desenvolvimento de um sistema que propõe melhorias neste gerenciamento.

Ao analisar soluções similares, foi observado grande burocracia e complexidade, tanto no uso da aplicação quanto na organização dos serviços prestados, o que nos levou a tentar simplificar ao máximo a relação cliente-estabelecimento.

Através do acesso via smartphone conectado à internet, o sistema permite o cadastro de pessoas, estabelecimentos e seus serviços disponíveis. Com a abertura de comanda sendo realizada através de um código de acesso, o sistema possibilita o acompanhamento em tempo real dos produtos consumidos e do valor total a ser pago. Permite, também, que o cliente acesse serviços do estabelecimento, como cardápio e realize pedidos mostrando o código da sua comanda.

Para esse fim, o restante deste trabalho está estruturado da seguinte maneira: na seção 1, é introduzido o tema e são apresentados os objetivos do projeto, na seção 2, a fundamentação teórica para a pesquisa é detalhada, na seção 3, a metodologia aplicada no projeto é demonstrada, na seção 4 é feita a apresentação do software e, por fim, na seção 5, são pontuadas as considerações finais.

1.1 CONTEXTUALIZAÇÃO

A entrada em estabelecimentos comerciais como casas noturnas ou bares, nos casos pesquisados, é feita a partir de um cadastro realizado por funcionários do estabelecimento, solicitando-se um documento que contenha dados como: nome completo, CPF, RG, foto, telefone. Em seguida, é entregue uma comanda, por muitas vezes uma folha de papel ou plástico, para que o consumo no estabelecimento seja controlado.

1.2 PROBLEMATIZAÇÃO

Para melhor entendimento do problema e a fim de desenvolver uma melhor solução, foram realizadas diversas pesquisas em redes sociais, como Facebook, FourSquare e Kekanto, para analisar as opiniões dos clientes que avaliaram

negativamente os estabelecimentos. Os resultados mostram que grande parte das avaliações negativas são decorrentes de tempos longos de espera na fila para entrar e sair dos estabelecimentos, conflitos com o caixa por conta de possíveis cobranças indevidas e, em alguns casos, a limitação das formas de pagamento disponíveis, como não aceitar determinada bandeira de cartão de crédito ou débito.

Com os dados encontrados, decidiu-se criar um sistema gerenciador de comandas que entrega ao usuário uma maneira simples, rápida e segura para controlar seus gastos, agilizando a entrada no estabelecimento e diminuindo a hipótese de extravio da comanda. Desta maneira, otimizando a qualidade de vida do cliente e conseqüentemente a avaliação do estabelecimento.

1.3 OBJETIVOS DO PROJETO

1.3.1 Objetivos Gerais

O objetivo do projeto é desenvolver um sistema computacional para o cadastramento de pessoas, estabelecimentos, produtos, gerenciamento de comandas e gerenciamento de entrada e saída de pessoas. Tudo isso com o intuito de aprimorar o atual processo utilizado por estabelecimentos comerciais. Este sistema é desenvolvido através das tecnologias aprendidas durante o curso de Tecnologia em Análise e Desenvolvimento de Sistemas (UFPR).

1.3.2 Objetivos Específicos

Para atingir os objetivos gerais propostos, são identificados os seguintes objetivos específicos:

- Cadastro de clientes;
- Cadastro de estabelecimentos;
- Cadastro de produtos;
- Gerenciamento de comandas;
- Gerenciar entrada e saída de pessoas.

1.4 PROPOSTA

Com embasamento nesse contexto, visto que há uma necessidade de realizar esse tipo de operação de forma rápida e eficaz, foi optado a automatização do acesso através de aplicação que permita abrir comandas automaticamente. Esta aplicação permitirá a administração e gerenciamento de estabelecimentos

comerciais, abertura de comandas para usuários cadastrados e acompanhamento de comandas abertas.

O sistema será desenvolvido para ambiente WEB, adotando a linguagem de programação PHP e recursos como JQuery, HTML e CSS para a visualização do conteúdo. Para a manipulação de dados, a aplicação faz uso de comandos SQL do sistema gerenciador de banco de dados MySQL para requerimentos e conexão.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 CONCEITOS

Para a melhor compreensão do texto, faz-se necessário a definição de alguns termos específicos da aplicação.

2.1.1 Estabelecimento comercial

Segundo Eliel Geraldino, em artigo publicado à Jus, um “estabelecimento comercial é o conjunto de bens corpóreos e incorpóreos reunidos pelo empresário para o desenvolvimento de sua atividade econômica, ou seja, o estabelecimento comercial é o instrumento da atividade do comerciante. Para que se exerça a atividade empresarial é necessário o capital, o trabalho e a organização” (SILVA, 2015).

Simplificadamente, um estabelecimento comercial é, de modo geral, um intermediário entre um produtor e um consumidor. É o fornecedor de certo produto procurado por determinado público. O empresário deseja instigar o cliente e incentivá-lo à compra, buscando diversas maneiras de atraí-lo para seu local de venda. A relação de compra e venda com o consumidor deve ser maior que o investimento de compra com o produtor. Dessa forma, o empresário gera lucro para sua empresa e garante seu sustento e mantimento do estabelecimento.

2.1.2 Bar e discoteca

Um bar é um local onde vendem-se bebidas podendo ou não vender também alimentos. Geralmente possui restrição de entrada para maiores de 18 anos por conta da venda de bebidas alcoólicas devido ao Segundo Artigo 243 do Estatuto da Criança e do Adolescente (PLANALTO, 1990), onde é considerado crime “vender, fornecer ainda que gratuitamente, ministrar, ou entregar, de qualquer forma, a criança ou adolescente, sem justa causa, produtos cujos componentes possam causar dependência física ou psíquica, ainda que por utilização indevida”. O público varia, dependendo da ambientação, abrangendo desde os mais jovens até idades mais avançadas.

Um bar pode estar conectado à uma discoteca ou boate, onde o objetivo geral é a venda de bebidas e a proporção de música e dança para seus consumidores. Ligeiramente diferente dos bares, as discotecas contêm poucas cadeiras ou locais

para descanso e seus ambientes são escurecidos para a utilização de jogos de luzes ritmados e música dançante e animada. Estes locais são frequentados, majoritariamente, por pessoas mais jovens.

2.1.3 Comanda

Uma comanda é um utensílio amplamente utilizado em bares e casas noturnas para organização dos produtos consumidos por cada cliente. Esta é entregue ao cliente na entrada do estabelecimento juntamente com a realização de um cadastro digital. O cadastro, no geral, contempla nome completo, RG (carteira de identidade) e telefone pessoal. A comanda pode ser feita de papel, a qual descreve os produtos fornecidos pelo local visitado, ou de plástico, exibindo apenas um código identificador para direcionar o processo. A cada pedido, um funcionário registra manualmente o produto pedido e sua quantidade.

2.2 PROCESSO ATUAL

Por padrão, os estabelecimentos comerciais utilizam comandas físicas para controlar o consumo de seus clientes e mesmo que a casa tenha o controle do que foi vendido através de um sistema que realiza o registro de consumo de cada consumidor em um banco de dados, podem surgir problemas como a perda da comanda por parte do cliente. Quanto isso acontece, fica impossível saber o que de fato foi consumido pela pessoa e o que pode ter sido consumido por outro como ato de má fé utilizando a comanda perdida.

Além disso, é fácil para o consumidor gastar mais do que o planejado, pois não existem meios de fácil acesso para o cliente verificar o que foi consumido até o momento. É necessário dirigir-se até um funcionário e pedir que lhe informem o valor total da conta até dado momento. Essa falta de controle pode resultar em desconfortos tanto para o estabelecimento quanto para quem está usufruindo dos serviços disponíveis.

Utilizando esse método de controle via papel com um número gerado por um sistema ligado à um registro no banco de dados e contendo dados pessoais, que devem ser cadastrados no momento de entrada na casa noturna, torna-se inevitável a aglomeração de pessoas aguardando em filas, ocasionando na insatisfação dos consumidores devido ao longo tempo de espera.

2.3 PÚBLICO ALVO

O número de smartphones no Brasil é de aproximadamente um por habitante (CAPELAS, 2017), índice que não para de crescer, devido à sua facilidade de acesso, mobilidade e praticidade.

Não obstante, tornou-se cada vez mais fácil estar conectado via esses aparelhos, seja através de *hotspots* disponibilizados pelo comércio local ou tecnologia 3G/4G de operadoras móveis. Este fato possibilita a troca instantânea de dados, a qualquer hora e em qualquer lugar, entre diversas pessoas conectadas à rede.

Graças a estes elementos, torna-se viável a criação de um sistema *mobile* que atenda a necessidade de agilizar cadastros, entradas e saídas de estabelecimentos sem a necessidade de um investimento pesado em equipamentos, visto que cada cliente pode utilizar seus equipamentos pessoais para o acesso.

2.4 MERCADO

Existem muitas soluções de controle de acesso nos dias de hoje. Uma delas é o “Sem Hora”, aplicativo que permite a compra de ingressos para eventos de entretenimento utilizando o smartphone, com um cartão de crédito ou débito cadastrado. Neste aplicativo, o usuário pode mostrar o *QRCode* pelo aparelho celular ou levar uma cópia impressa para substituir o ingresso após a compra. Apesar da facilidade, o aplicativo não atende aqueles que gostam de frequentar os mesmos estabelecimentos, pois eles possuem apenas alguns eventos cadastrados, e não fazem parcerias com casas ou bares noturnos. (SEM HORA, 2017)

Encontra-se também, em polos gastronômicos, aplicativos para ver o cardápio e preço dos produtos. É o caso do “Mercado Sal”, que disponibiliza um aplicativo de mesmo nome no qual é possível fazer e pagar os pedidos pelo *smartphone*. Entretanto, é limitado aos quiosques do estabelecimento e alguns ainda se encontram fora do serviço.

Voltado para a vida noturna, existe o “ViApp”, um aplicativo exclusivamente *mobile* que facilita a inserção de nomes nas listas VIP de casas noturnas, garantindo descontos para os usuários. O principal objetivo do “ViApp” é aproximar o usuário de eventos noturnos previamente cadastrados no sistema, possibilitando a compra de ingressos de festas antecipadamente ou adicionar o nome em listas VIP de baladas.

Ótimo aplicativo, mas suas funcionalidades são limitadas, não dando ao usuário segurança após adentrar o evento desejado. (VIAPP, 2017)

3 ANÁLISE E METODOLOGIA

De modo a fundamentar o desenvolvimento do trabalho, foram levados em conta alguns princípios da Engenharia de Software. Esta engenharia auxilia justamente a situar os processos, métodos, técnicas e ferramentas utilizadas no desenvolvimento de um software, assegurando, assim, a qualidade do produto (FALBO, 2005).

As principais atividades realizadas foram (FILGUEIRAS; MELNIKOFF, 2000):

- Identificação das necessidades do usuário;
- Identificação dos requisitos do sistema;
- Análise de viabilidade;
- Definição dos componentes do sistema;
- Alocação das funções nos componentes;
- Interfaces do sistema;
- Estruturação da informação.

3.1 MODELO DE PROCESSO DE ENGENHARIA DE SOFTWARE

Devido ao escopo pequeno e pela ausência de um cliente real para suprir as demandas, para o desenvolvimento da aplicação foi selecionado o modelo em cascata. Este modelo é um processo sequencial, chamada desta forma por seguir em suas diferentes etapas em declive como uma cascata (ROSSBERG, 2014). Este modelo é composto por sete fases:

- Especificação de requisitos;
- Design;
- Implementação;
- Testes;
- Implantação;
- Manutenção.

3.2 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

Para a estruturação do trabalho, foram levantados os requisitos funcionais e não funcionais do sistema. Ou seja, os requisitos que descrevem as funcionalidades do sistema e os requisitos que não dizem respeito a sua funcionalidade, porém são necessários (CARDOZO, 2008). Segue abaixo a relação de requisitos:

Requisitos não funcionais

O sistema deverá ser desenvolvido em PHP, utilizando a framework CodeIgniter em modelo MVC.

- O sistema deverá ser desenvolvido com a linguagem de marcação HTML.

- O sistema deverá ser complementado com a linguagem de programação JQuery.

- O sistema deverá ser complementado com a linguagem de estilo CSS.

- O sistema deverá possuir uma interface amigável para todos os usuários.

- O sistema deverá rodar igualmente nos navegadores: Mozilla Firefox, Google Chrome, Internet Explorer, Opera e Safari.

- O servidor do sistema deverá possuir instalado o PHP 7.0 ou versão superior.

- O banco de dados a ser utilizado deverá ser o MySQL 5.7 ou versão superior.

Tabela 3.1: Requisitos não funcionais.

Requisitos funcionais

- O sistema deverá possibilitar o gerenciamento dos dados referentes aos usuários.

- O sistema deverá possibilitar o gerenciamento dos dados referentes aos estabelecimentos.

- O sistema deverá possibilitar o gerenciamento dos dados referentes aos produtos.

- O sistema deverá possibilitar o gerenciamento dos dados referentes às comandas.

- O sistema deverá possuir um mecanismo contra invasões e ações malignas ao banco.

- O sistema deverá possuir hierarquia de permissão.

- O sistema deverá validar todos os campos inseridos para evitar erros.

- O sistema deverá manter os dados dos usuários.

- O sistema deverá permitir somente ao administrador a manutenção dos dados dentro do sistema.

- O sistema deverá requisitar login e senha.

- O sistema deverá ser acessível via smartphone.

- O sistema deverá requisitar login e senha para abertura de comandas

- O sistema deverá possibilitar a validação de acesso via leitor de códigos.

- O sistema deverá possibilitar a visualização do extrato da comanda.

- O sistema deverá possibilitar a visualização dos produtos do estabelecimento enquanto a comanda estiver aberta.

Tabela 3.2: Requisitos funcionais.

A partir desses requisitos e com o propósito de compreender o funcionamento da aplicação, foi elaborado o diagrama de caso de uso encontrado no apêndice 2, junto com sua respectiva documentação. Além deste e ainda com o mesmo intuito, encontra-se o diagrama de classes no apêndice 3, os diagramas de sequência no apêndice 4 e os diagramas de atividades no apêndice 5.

3.3 PERFIS DE USUÁRIO

A suíte é subdividida em três aplicativos diferentes, cada qual com objetivo funcionalidades diferentes de acordo com o perfil do usuário que acessou, conforme tabela abaixo.

Aplicação	Descrição
MellowAdmin	Aplicação voltada para o gerenciamento de todo o sistema, com todas as permissões de acesso
MellowBusiness	Aplicação voltada para o estabelecimento, permitindo o gerenciamento e manutenção das informações desse determinado estabelecimento, assim como a validação e abertura de comandas para ele
MellowApp	Aplicação voltada para o público geral, permitindo cadastro e utilização de comandas em estabelecimento cadastrados

Tabela 3.3: Aplicações da suíte.

Onde, apenas na aplicação MellowBusiness, existe uma subdivisão de perfil de gerente e funcionário, no qual o usuário gerente tem permissão de alterar os dados do próprio estabelecimento ao passo que o usuário funcionário apenas pode gerenciar as comandas.

3.4 MODELAGEM DOS DADOS

O modelo de banco de dados nada mais é do que uma descrição dos tipos de dados armazenados em um banco (HEUSER, 2000) em diferentes níveis de abstração dependendo do modelo selecionado de acordo com a intenção do modelador.

Para a fundamentação deste trabalho, foi levado em consideração os aspectos necessários para que a aplicação armazene os dados dos estabelecimentos, seus

produtos, seus clientes e comandas. Através da ferramenta MySQL Workbench, foi possível criar um modelo de dados, encontrado no apêndice 1, já possuindo o relacionamento entre suas entidades e inserções necessárias para que o software execute corretamente.

3.5 PLANO DE ATIVIDADE

Seguindo o modelo cascata de desenvolvimento, foi elaborada a lista de atividades necessárias para que o projeto funcione corretamente. De modo geral, como observado na tabela 3.5.1, os nós são as etapas da metodologia selecionada e suas folhas são as atividades que necessitam ser realizadas naquela determinada etapa.

#	Atividade	Pré-requisitos
1	Gerenciamento de projeto	
2	Definição da equipe	
3	Definição do tema	2
4	Planejar cronograma	2, 3
5	Definição do workflow	2, 3
6	Análise	
7	Especificação dos requisitos	3
8	Definição das tecnologias a serem utilizadas	3, 7
9	Pesquisa da fundamentação teórica do projeto	8
10	Elaboração da fundamentação teórica do projeto	9
11	Especificação do diagrama de casos de uso	7, 15
12	Especificação do diagrama de classes	7, 15
13	Especificação dos diagramas de sequência	11, 12
14	Design	
15	Definição e elaboração da marca	3
16	Prototipação das telas	8
17	Modelagem do Banco de Dados	8
18	Codificação	
19	Módulo Login	16, 17
20	Módulo Configuração	

21	Alergênico	16, 19
22	Perfil	16, 19
23	Tipo de estabelecimento	16, 19
24	Tipo de produto	16, 19
25	Usuário	16, 19
26	Módulo Portal	1, 6
27	Módulo Estabelecimento	
28	Explorar	16, 19, 23
29	Gestão	16, 19, 23
30	Módulo Produto	
31	Cardápio	16, 19, 28
32	Gestão	16, 19, 27
33	Módulo Sessão	
34	Verificador	16, 19, 35
35	Check-in	16, 19, 28
36	Módulo Comanda	
37	Extrato	16, 19, 33, 39
38	Histórico	16, 19
39	Gestão	16, 19
40	Lançamento	16, 19, 33, 39
41	Módulo Home	
42	Administrador	16, 19
43	Gerente	16, 19
44	Funcionário	16, 19
45	Cliente	16, 19
46	Testes	
47	Geração do pacote do produto para testes	18
48	População do banco de dados	17
49	Testes e correções	47
50	Atualização do desenvolvimento do trabalho escrito	49
51	Integração	
53	Geração do pacote final do produto	46
54	Finalização e entrega do trabalho escrito	50

55	Apresentação final do trabalho de conclusão de curso	54
56	Revisão final e entrega oficial	55

Tabela 3.4: Lista de atividades.

Durante a etapa de codificação, suas atividades foram divididas em módulos do sistema, e estes módulos foram subdivididos em funcionalidades específicas de cada módulo, conforme exibido no diagrama WBS encontrado no apêndice 6. Com a WBS pronta, foi elaborado também o gráfico Gantt da execução do projeto, encontrado no apêndice 7.

3.6 PLANO DE RISCOS

Para a implementação do projeto foram identificados inicialmente alguns riscos, os quais, para evitar problemas futuros, definiu-se ações específicas para evitar que o estrago seja maior do que o esperado (Apêndice 8).

3.7 RESPONSABILIDADES

Abaixo a lista de atividades executadas por cada membro da equipe durante o desenvolvimento do trabalho:

#	Atividade	Responsável
1	Gerenciamento de projeto	
2	Definição da equipe	Equipe
3	Definição do tema	Equipe
4	Planejar cronograma	Amanda
5	Definição do workflow	Harrison
6	Análise	
7	Especificação dos requisitos	Amanda e Harrison
8	Definição das tecnologias a serem utilizadas	Equipe
9	Pesquisa da fundamentação teórica do projeto	Harrison e Marcelo
10	Elaboração da fundamentação teórica do projeto	Harrison e Marcelo
11	Especificação do diagrama de casos de uso	Marcelo
12	Especificação do diagrama de classes	Amanda
13	Especificação dos diagramas de sequência	Harrison

14	Design	
15	Definição e elaboração da marca	Amanda
16	Prototipação das telas	Amanda e Marcelo
17	Modelagem do Banco de Dados	Harrison e Marcelo
18	Codificação	
19	Módulo Login	Amanda
20	Módulo Configuração	
21	Alergênico	Amanda
22	Perfil	Amanda
23	Tipo de estabelecimento	Amanda
24	Tipo de produto	Amanda
25	Usuário	Amanda
26	Módulo Portal	Amanda e Marcelo
27	Módulo Estabelecimento	
28	Explorar	Amanda
29	Gestão	Amanda
30	Módulo Produto	
31	Cardápio	Amanda
32	Gestão	Amanda
33	Módulo Sessão	
34	Verificador	Amanda
35	Check-in	Amanda
36	Módulo Comanda	
37	Extrato	Amanda
38	Histórico	Amanda
39	Gestão	Amanda
40	Lançamento	Amanda
41	Módulo Home	
42	Administrador	Amanda
43	Gerente	Amanda
44	Funcionário	Amanda
45	Cliente	Amanda
46	Testes	

47	Geração do pacote do produto para testes	Amanda
48	População do banco de dados	Marcelo
49	Testes e correções	Harrison e Marcelo
50	Atualização do desenvolvimento do trabalho escrito	Harrison e Marcelo
51	Integração	
53	Geração do pacote final do produto	Amanda
54	Finalização e entrega do trabalho escrito	Harrison e Marcelo
55	Apresentação final do trabalho de conclusão de curso	Equipe
56	Revisão final e entrega oficial	Harrison e Marcelo

Tabela 3.5: Lista de responsabilidades.

3.8 FERRAMENTAS E MATERIAIS

Para melhor compreensão, será explicado sucintamente a seguir, as ferramentas e métodos adotados para o desenvolvimento do Mellow Suíte.

3.8.1 Astah

Astah, anteriormente denominado JUDE (*Java and UML Developers' Environment*), é um software de modelagem UML desenvolvida pela empresa japonesa Change Vision (CARNEIRO, 2015).

É um software de uso livre e multiplataforma voltado para o desenvolvimento de diversos diagramas da UML.

3.8.2 Ajax

Ajax, ou *Asynchronous JavaScript and XML* no original, é um método utilizado durante a programação para tornar o navegador mais interativo com o usuário. Com a mescla e o uso sistemático do JavaScript e XML, é possível fazer com que solicitações sejam enviadas ao servidor web sem que haja a necessidade de recarregar da página (NIEDERAUER, 2007).

O modelo anterior e geralmente mais usado é, ao contrário do Ajax, necessário que cada etapa procedida necessite o envio de todas as solicitações, através de um link, recarregando a página inteira para o recebimento das requisições ao servidor. Já com o Ajax é diferente, a ferramenta serve de intermédio entre a interface gráfica e o servidor web. Como é acionado através de comandos JavaScript, permite que o

usuário permaneça visualizando a página normalmente enquanto ocorre a comunicação com o servidor.

O Ajax é um método flexível e eficiente, onde seu maior beneficiário é o usuário final. Como o conceito inteiro do Mellow Suíte é focado justamente na economia de tempo em que o usuário, seja ele cliente ou estabelecimento, consegue realizar seus procedimentos cruciais, é importante que a interface seja amigável e entregue as informações da maneira o mais eficaz possível.

3.8.3 Bitbucket

O Bitbucket é um sistema de controle de versão cujo objetivo de gerenciar diferentes versões de um mesmo projeto. É possível obter históricos das modificações, permitir que vários programadores trabalhem no mesmo projeto e permitir um comparativo entre várias versões do projeto (APRENDENDO WWW, 2015).

O projeto fica armazenado no servidor (repositório), onde ficam registrados os históricos e modificações de cada versão. O desenvolvedor pode baixar a última versão e trabalhá-la e posteriormente atualizar a versão contida no servidor.

Essa sincronização de arquivos entre a estação de trabalho e o servidor é feito através dos comandos *commit* e *update*.

O *commit* envia o conjunto de arquivos alterados ao servidor, gerando um novo histórico de atualização. O *update* faz o inverso, mandando a última versão dele ao computador do desenvolvedor.

3.8.4 CodeIgniter

O CodeIgniter é um framework para desenvolvimento web baseado na arquitetura MVC (VARGAS, 2015). Foi escolhido por ser leve, ter uma segurança íntegra apesar de simples e por apresentar uma ótima documentação.

3.8.5 CSS 3

As folhas de estilo em cascata, ou simplesmente CSS (*Cascading Style Sheets*), são ferramentas utilizadas no desenvolvimento de páginas e aplicativos para web e utilizadas em paralelo a linguagens de marcação como o HTML. O CSS informa ao navegador como que as informações devem aparecer na interface gráfica, definindo

e formatando as fontes, margens, cores, imagens, alinhamentos entre outros (POUNCEY; YORK, 2011).

O CSS ajuda a padronizar a formatação das páginas web, fornecendo uma facilidade durante o desenvolvimento e possibilitando que o conteúdo da parte visual do sistema seja mais fácil de separar e, quando necessário, atualizar. Trazendo, desta forma, benefícios para o desenvolvedor configurar a interface apresentada ao usuário de maneira mais simples e fácil.

3.8.6 Git

Construir um sistema em equipe é um grande desafio. Nosso código tem que se integrar de maneira transparente e sem emendas com o código de todos os outros membros.

Existem ferramentas que funcionam como máquinas do tempo e robôs de integração para o seu código. Elas nos permitem acompanhar as alterações desde as versões mais antigas. Também é possível detectar e mesclar alterações nos mesmos arquivos, além de identificar conflitos, tudo de maneira automática.

Essas ferramentas são chamadas de sistemas de controle de versão e o Git é um desses sistemas de controle de versão (AQUILES; FERREIRA, 2014).

3.8.7 Google Material

A linguagem Material Design, desenvolvido pelo Google, impressiona pela facilidade de desenvolvimento e pelo design. Envolve todas as plataformas, tendo sido iniciado no *Android*. O Material Design utiliza cor, luz e movimento para gerar interfaces bonitas e para proporcionar uma navegação intuitiva para o usuário (MEW, 2016).

3.8.8 HTML 5

HyperText Markup Language, ou simplesmente HTML, é a linguagem padrão utilizada nos ambientes de web. Diferentemente do que muitos possam dizer, HTML não se trata de uma linguagem de programação e sim de uma linguagem de marcação. Ou seja, é um conjunto de elementos que podem incorporar textos.

Servem como indicador, dizendo-se determinadas palavras devem começar com maiúscula, minúscula, sublinhada e por assim vai. É basicamente o mesmo que uma folha de estilos faz em documentos.

É uma linguagem versátil e neutra, que trabalha principalmente com hipertextos. O hipertexto, por sua vez, é uma ferramenta que vincula as informações, criando laços entre os textos em si com imagens, fontes, imagens ou até mesmo links (SAVOLA, 1996).

De modo geral, com a HTML, é possível fazer inúmeras tarefas, pois essa linguagem é capaz de enviar qualquer tipo de informação de diversas maneiras. Isto demonstra sua abrangência e capacidade, permitindo criar do mais simples documento ao mais complexo, desta maneira, tornando-se ideal para o desenvolvimento do sistema Mellow Suite.

3.8.9 JQuery

JQuery é uma poderosa biblioteca JavaScript criada para simplificar a criação de efeitos visuais e de interatividade em web sites. Ele propicia a criação de scripts de uma forma simples, reduzindo a 6 linhas um código de aproximadamente 40 desenvolvido com JavaScript tradicional (SILVA, 2010).

3.8.10 PHP 7.0

O PHP: Hypertext Preprocessor, ou simplesmente PHP, é uma linguagem de script dinâmica e de código aberto, que é executada no lado servidor da página.

Quando embutido o PHP em uma página HTML, o servidor retorna uma página contendo somente HTML, e esta, é interpretada pelo navegador do usuário. Esta linguagem foi criada em 1994 por Rasmus Lerdorf (DALL'OGGIO, 2007), servindo, inicialmente, como um pacote de programas CGI, para substituir um conjunto de scripts na linguagem Perl que o seu criador usava em sua página pessoal. Anos mais tarde, foi lançado o pacote PHP/FI que continha um interpretador de comandos SQL.

Somente em sua versão 3, lançada no ano 2000, o PHP ganhou orientação a objetos.

Para elaboração do gestor do presente trabalho, utiliza-se a linguagem PHP em sua versão mais atual, 7.1, pelos seguintes fatores (GILMORE, 2010):

- Custo: O PHP é gratuito e está disponível para download no site do seu fornecedor;
- Portabilidade: A linguagem PHP está disponível para diversos sistemas operacionais diferentes;

- Integração com o Banco de Dados: O PHP tem conexões nativas com muitos sistemas de banco de dados, e principalmente, com o MySQL.

3.8.11 QRCode

Significa Código de resposta rápida, foi criado para substituir o código de barras e tem esse nome por ser rapidamente reconhecido pelas pessoas.

3.8.12 MVC

O modelo de organização adotado neste trabalho foi MVC, que basicamente significa Modelo/Visão/Controlador (Model-View-Controller, no original).

O objetivo desse modelo é separar as linhas de código em etapas interligadas, dessa forma, facilitando a organização e reutilização dos códigos do projeto. Esse modelo é composto por três objetos principais: O Modelo é o objeto de aplicação, a Visão é a interface com o usuário e o Controlador são as regras de negócios (GAMMA, E. et al, 1995).

A camada Modelo (Model) trabalha diretamente com os dados e com o banco de dados integrado no sistema através de comandos SQL.

A camada Visão (View) trabalha com a interface gráfica, é a tela que o usuário vê e interage.

Já a camada Controlador (Controller) trabalha com as regras de negócio. Nela que estão chamadas para retorno do dados que camada Modelo irá executar para acessar os bancos de dados e as chamadas para a exibição da interface gráfica.

Com a divisão do código nessas três partes, a programação do sistema torna-se mais flexível, permitindo uma melhor abordagem dos desenvolvedores durante a implementação, tratamento de erros e manutenção do software, além de permitir a reutilização de partes dos códigos. Para este projeto, a implementação deste modelo é nativa do framework CodeIgniter, facilitando a codificação.

3.8.13 MySQL

MySQL é um sistema de gerenciamento de banco de dados relacional poderoso e muito rápido (MILANI, 2007), criado pelos suecos David Axmark e Allan Larsson, e pelo finlandês Michael Monty Widenius. O servidor de MySQL controla o acesso aos dados para assegurar que vários usuários possam trabalhar com os dados ao mesmo tempo, fornecendo acesso rápido a eles e assegurando que apenas

usuários autorizados obtenham acesso. Utiliza da linguagem SQL, a linguagem de consulta padrão de banco de dados em todo o mundo.

Suas principais vantagens são:

- Custo: O MySQL disponível sem nenhum custo, sob uma licença de código-fonte aberto, e a baixo custo, sob uma licença comercial;
- Desempenho: O MySQL é inegavelmente rápido, apresentando em testes feitos em 2002 comparando o desempenho de banco de dados de código aberto (PIRES; NASCIMENTO; SALGADO, 2002), resultados iguais ao Oracle, que por sua vez, é pago;
- Portabilidade: O MySQL pode ser utilizado em muitos sistemas Unix diferentes, bem como sob o Microsoft Windows.

3.8.14 MySQL Workbench

O MySQL Workbench é uma ferramenta visual que auxilia no desenvolvimento de bancos de dados baseados em MySQL. A partir dele pode-se modelar e compreender melhor a modelagem de dados.

3.8.15 Trello

O Trello é uma ferramenta que auxilia no gerenciamento do projeto, organizando as tarefas em listas, de forma versátil.

3.8.16 UML

A UML (Unified Modeling Language - Linguagem de Modelagem Unificada) é uma linguagem de modelagem, um padrão de mercado para modelagem de sistemas com programação orientada a objeto (MELO, 2010). Serve para estruturar modelos de desenvolvimento de softwares.

No Mellow Suíte, foi adotado o uso da UML para especificar, descrever e visualizar a construção do mesmo. Os diagramas utilizados foram:

- Diagrama de Classes – Representação da estrutura do sistema, recorrendo ao conceito de classe e suas relações (CARVALHO; LOPES; MORAIS, 2009);
- Diagrama de Caso de Uso - Descrição da sequência de ações que representam um cenário principal e cenários alternativos, demonstrando o comportamento de um sistema através de atores;
- Diagrama de Sequência - Representação e modelo do fluxo de mensagens, eventos e ações entre objetos e componentes do sistema (MAIA, 2008);
- Diagrama de Atividades – Representação do fluxo de atividades de um processo, demonstrando a relação de dependências entre essas atividades.

Além disso, também fora documentado o Diagrama de Caso de Uso, para ter um melhor detalhamento do sistema. Todos os diagramas e documentações estão em anexo ao trabalho.

3.8.17 WampServer 3.0

O WAMP (Windows, Apache, MySQL e PHP) é um software com um conjunto de programas independentes, muito utilizado em servidores web. Foi criado em 2004 pela PHP Team com o objetivo de instalar rapidamente no computador os softwares PHP, MySQL e Apache (WAMPSEVER).

Para o trabalho corrente, utiliza-se esta tecnologia devido à sua facilidade de instalação e constante atualização dos softwares nele contidos.

4 APRESENTAÇÃO DO SOFTWARE

Capítulo destinado à apresentação do software, demonstrando telas e funcionalidades do Mellow Suite, assim como sua instalação e estrutura básica da codificação.

4.1 INSTALAÇÃO

Para fazer com que o sistema funcione, são necessários ao menos dois programas para instalá-lo: o WampServer 3.0 e o MySQL Workbench. Através do menu de contexto do WampServer (Figura 4.1), iniciar todos os serviços.

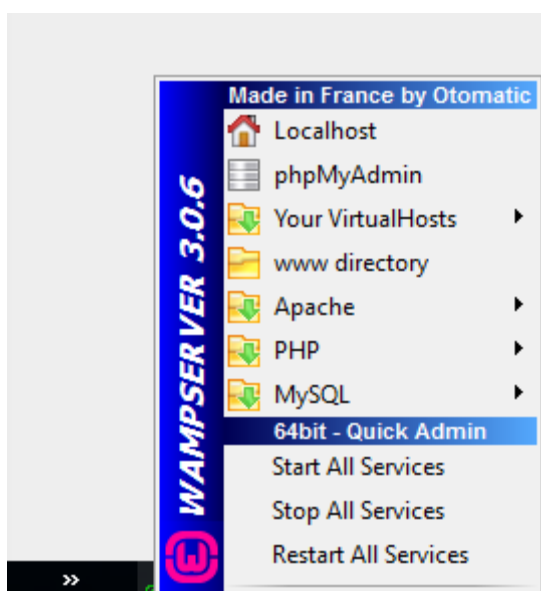


Figura 4.1: Menu de contexto do WampServer.

Fonte: Autores (2017)

Feito isso, descompactar os arquivos de instalação no diretório “www”, o qual pode ser acessado também via menu de contexto do programa. Na pasta “mellow” descompactada, acessar a subpasta “assets” e por fim a subpasta “bd” (Figura 4.2). Nesta pasta “bd”, abrir o arquivo modelo “bd.mwb”.

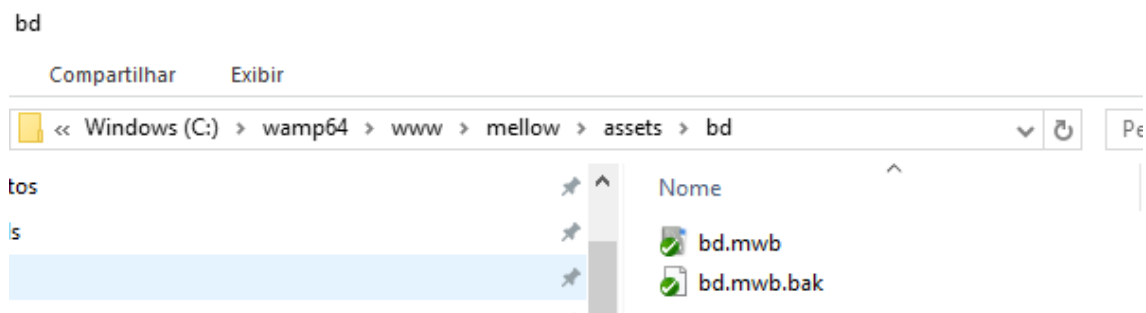


Figura 4.2: Arquivo do modelo.

Fonte: Autores (2017)

Com modelo aberto no MySQL Workbench, acessar o menu “Database” e acionar o botão “Forward Engineer...” (Figura 4.3), ou através da hotkey “Ctrl+G” do software.

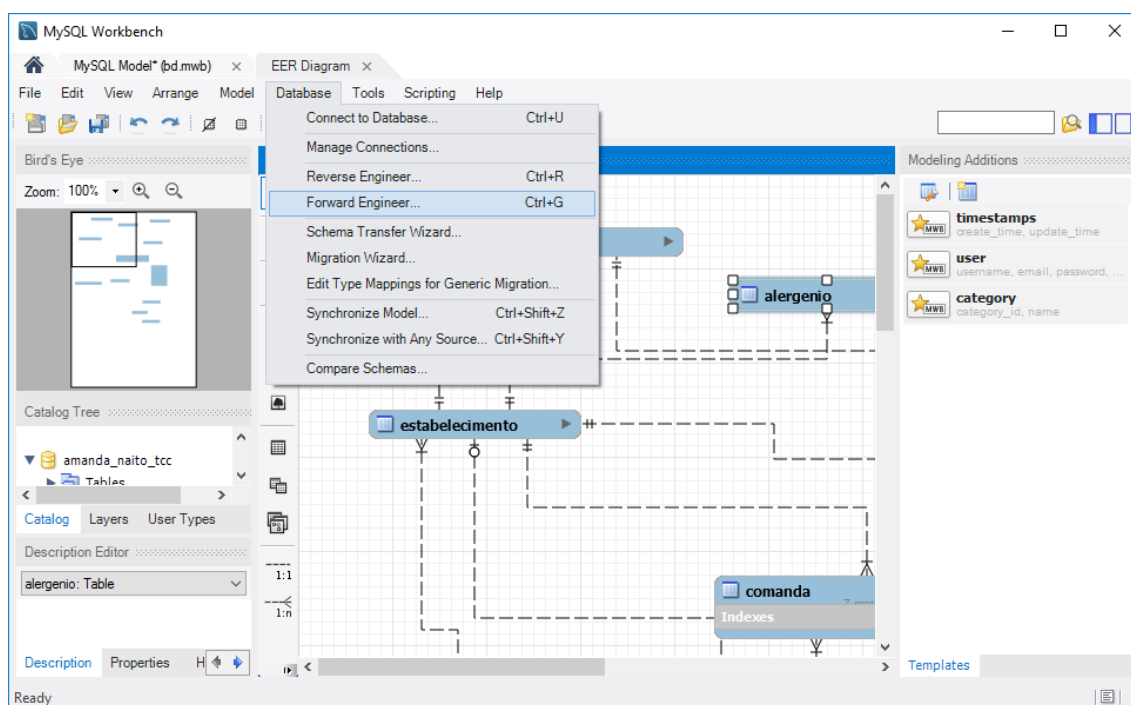


Figura 4.3: Menu Database do MySQL Workbench.

Fonte: Autores (2017)

Na tela de configuração, selecionar os dados da conexão local e prosseguir com o wizard até o final. Feito isso, o banco de dados local já estará apto a ser utilizado.

Após a instalação do banco de dados, acessar o caminho “<diretório wamp> www\mellow\application\admin\config” e abrir o arquivo “database.php”. Ao final do

arquivo, alterar as configurações do hostname, username e password para as configurações do servidor MySQL local. Por padrão, o Wampserver utiliza as seguintes configurações exibidas na figura 4.4.

```

2 |         When you run a query, with this setting set to TRUE (default),
3 |         CodeIgniter will store the SQL statement for debugging purposes.
4 |         However, this may cause high memory usage, especially if you run
5 |         a lot of SQL queries ... disable this to avoid that problem.
6 |
7 |     The $active_group variable lets you choose which connection group to
8 |     make active. By default there is only one group (the 'default' group).
9 |
10 |    The $query_builder variables lets you determine whether or not to load
11 |    the query builder class.
12 |
13 | */
14 | $active_group = 'default';
15 | $active_record = TRUE;
16 | $query_builder = TRUE;
17 |
18 | $db['default'] = array(
19 |     'dsn' => '',
20 |     'hostname' => 'localhost:3306',
21 |     'username' => 'root',
22 |     'password' => '',
23 |     'database' => 'amanda_naito_tcc',
24 |     'dbdriver' => 'mysqli',
25 |     'dbprefix' => '',
26 |     'pconnect' => FALSE,
27 |     'db_debug' => (ENVIRONMENT !== 'production'),
28 |     'cache_on' => FALSE,
29 |     'cachedir' => '',
30 |     'char_set' => 'utf8',
31 |     'dbcollat' => 'utf8_general_ci',
32 |     'swap_pre' => '',
33 |     'encrypt' => FALSE,
34 |     'compress' => FALSE,
35 |     'stricton' => FALSE,
36 |     'failover' => array(),
37 |     'save_queries' => TRUE
38 | );

```

Figura 4.4: Arquivo de configuração do banco de dados.

Fonte: Autores (2017)

Ao final desta etapa, o sistema já reconhecerá o banco de dados importado e permitirá sua utilização.

Por padrão, foram disponibilizados alguns usuários para acesso, conforme descrito abaixo:

Login	Senha	Perfil	Estabelecimento
ana.abbott	111111	Gerente	Caldeirão Furado
neville.longbottom	111111	Funcionário	Caldeirão Furado
madame.rosmerta	111111	Gerente	Três vassouras
catia.bell	111111	Funcionário	Três vassouras
marcelo.benigno	111111	Administrador	

Tabela 4.1: Usuários e senhas padrões.

Utilizando a tela de login, os usuários têm acesso ao sistema, de acordo com seus respectivos perfis.

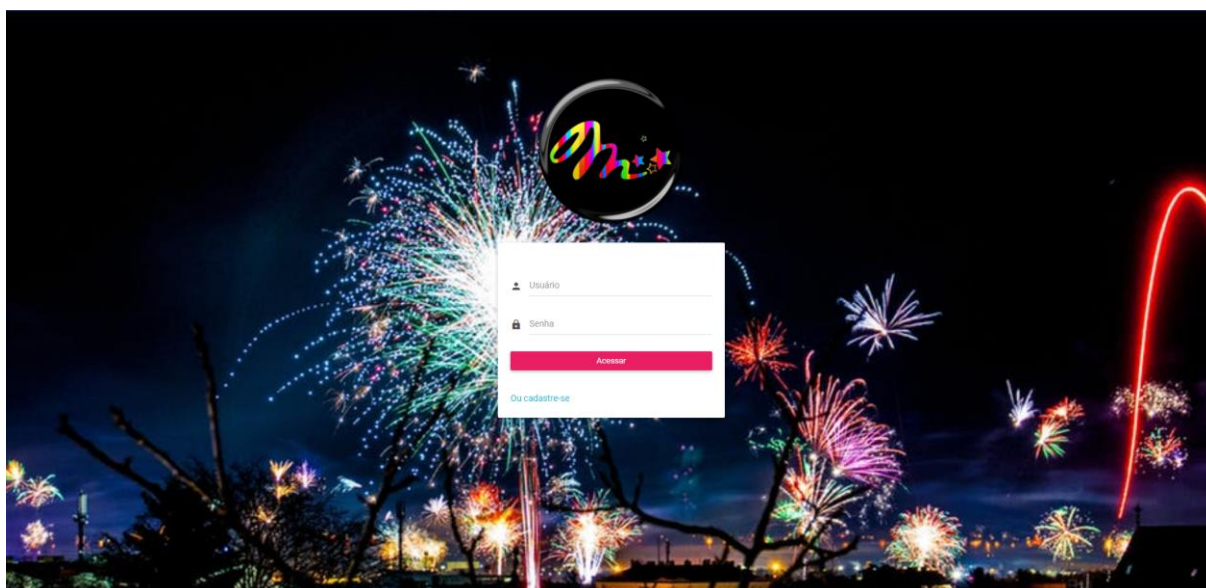


Figura 4.5: Tela de login no sistema.

Fonte: Autores (2017)

O sistema libera para o usuário, opções diversas para os usuários, como cadastrar estabelecimentos, abrir comandas ou criar o cardápio do estabelecimento.

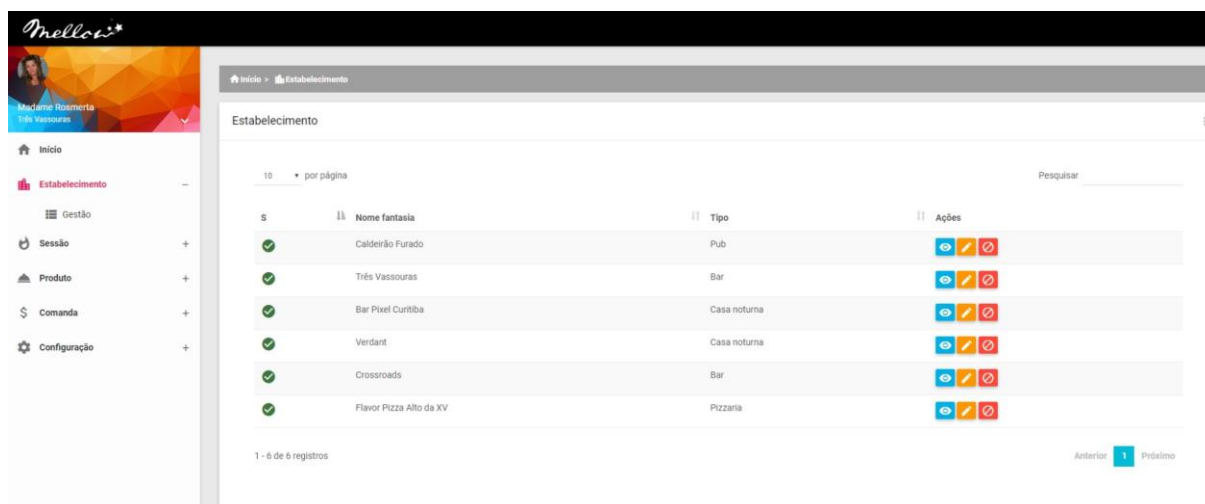


Figura 4.6: Tela de cadastro de estabelecimentos.

Fonte: Autores (2017)

A função de finalizar comandas, encerra a comanda, e não mais permite sua edição, apenas visualização, fazendo com que o usuário tenha que abrir uma nova comanda, caso deseje continuar consumindo.

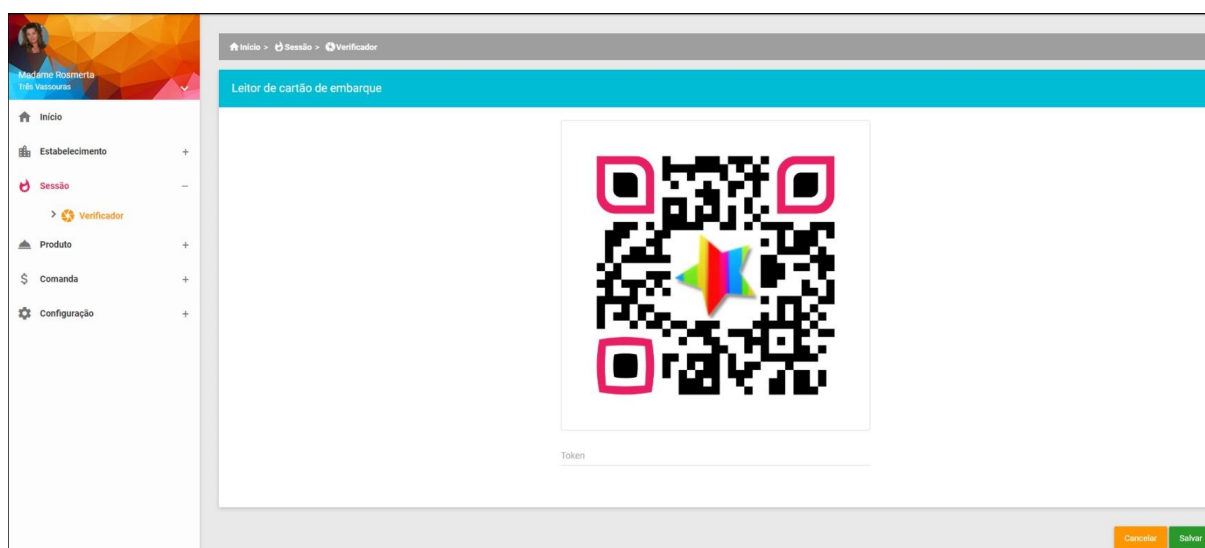


Figura 4.7: Tela de verificação de abertura de comanda.

Fonte: Autores (2017)

5 CONSIDERAÇÕES FINAIS

Com o desenvolvimento do trabalho de conclusão de curso voltado para o gerenciamento de estabelecimentos e comandas, em suma, foi possível notar a dificuldade dos profissionais deste ramo em controlar a entrada e saídas de seus clientes, assim como realizar o fechamento e faturamento do consumo dos mesmos dentro de suas áreas comerciais. Da maneira que essas operações são feitas atualmente, há um grande desperdício de tempo e recursos, além de causar insatisfação por parte dos clientes devido a longas filas de espera.

O projeto tinha por objetivo sanar essa deficiência, agilizando o processo de cadastro, identificação e consumo de bens e mitigar a divergência de dados contidos no sistema do estabelecimento e o registrado na comanda de papel do cliente.

Com a conclusão do trabalho, o objetivo principal foi cumprido. Com o desenvolvimento do Mellow Suite, os estabelecimentos têm a oportunidade de controlar de maneira mais eficiente o fluxo de pessoas e seu consumo e os clientes podem gozar de segurança contra extravios, cobranças indevidas e menor tempo de espera.

Apesar de o sistema estar com quase 100% das funcionalidades propostas no escopo inicial, não finaliza comandas de clientes realizando o pagamento de seu consumo utilizando um cartão de crédito ou débito atrelado ao cadastro do usuário como planejado inicialmente. Devido ao fato da equipe não possuir clientes reais para testes e as operadoras de cartão de crédito não disponibilizarem equipamentos de cobrança para quem não possui CNPJ, não foi possível implementar essa funcionalidade na versão final do sistema. A implementar dessa função futuramente agregaria e muito no valor comercial do produto, pois permite que os estabelecimentos e clientes tenham um meio mais rápido e seguro de realizar pagamentos.

A primórdio, o Mellow Suite foi criado com o intuito de atender casas de entretenimento noturno, mas devido à versatilidade como foi projetado e desenvolvido, pode atender a qualquer estabelecimento que tenha um cardápio de produtos ou serviços à venda. Basta que um estabelecimento interessado cadastre seus serviços ou produtos no sistema e os usuários poderão consultá-los de seus computadores ou smartphones conectados à internet, aumentando o alcance de mercado e divulgação para mais clientes em potencial.

Houve dificuldades na concepção do sistema, devido ao curto espaço de tempo para ser desenvolvido e uma necessidade de desistência da ideia inicial do projeto poucos meses após o começo do mesmo, fazendo com que a equipe reavaliasse a continuidade dele.

Neste último caso, após dias escolhendo e finalmente definindo a marca “*Nighter*” (derivada do termo em inglês “*all-nighter*”), o escopo inicial do projeto e dando início das etapas de análise, tomando sempre um certo cuidado para não selecionar algo já disponível no mercado, a equipe se deparou com um problema na fase final que era a obtenção do domínio para marca definida: o domínio já havia sido comprado por uma empresa paulistana poucos dias antes. Com esse fato, a equipe buscou saber do que se tratava e encontrou um portal para um aplicativo ainda não lançado que possuía o mesmo nome, o mesmo esquema de cores, uma logo parecida e uma proposta que também abordava o tema de casas noturnas e festas. Após avaliar as funcionalidades desse aplicativo e conversas sobre o que fazer a seguir, a equipe decidiu prosseguir com o projeto visto que a proposta dele era algo mais voltado a rede social do que de fato ao gerenciamento de comandas e acessos.

Entretanto, apesar das adversidades, foi possível concluir o projeto no prazo estipulado, apesar de ser apenas um protótipo e com uma bela redução do escopo original.

Fora isso, também precisa ser constado o aprendizado que esse projeto proporcionou aos integrantes da equipe, possibilitando que os mesmos, na prática, aprendessem a trabalhar em um projeto que utilizasse todos os temas abordados durante a aprendizagem cursada, além de incentivar a pesquisa e a busca por conhecimentos que vão além da ementa do curso.

REFERÊNCIAS

ALEXANDRE, Aquiles, RODRIGO Ferreira. **Controlando versões com Git e GitHub**. 2014

APRENDENDO WWW, **Controle de Versão, Git, Github e Bitbucket – Afinal, o que é tudo isso ?**. Disponível em:
<<https://aprendendowww.wordpress.com/2015/02/20/control-de-versaogit-github-e-bitbucket-afinal-o-que-e-tudo-isso/>> Acesso em: 29 nov. 2017.

CAPELAS, Bruno. **Até o fim de 2017, Brasil terá um smartphone por habitante, diz FGV**. 2017. Disponível em:
<<http://link.estadao.com.br/noticias/gadget,ate-o-fim-de-2017-brasil-tera-um-smartphone-por-habitante-diz-pesquisa-da-fgv,70001744407>> Acesso em: 29 nov. 2017.

CARDOZO, Eleri. **Requisitos de Software**. 2008. Disponível em:
<<https://pt.scribd.com/document/102743729/Levantamento-de-Requisitos>> Acesso em: 02 nov. 2017.

CARNEIRO, Bruno Seabra. **O QUE É O ASTAH?**. 2015. Disponível em:
<http://www.startupsstars.com/2015/10/o-que-e-o-astah-posttecnico-por-bruno-seabra/> Acesso em: 17 nov. 2017.

CARVALHO, Armando Jorge; LOPES, Filomena Castro; MORAIS, Maria Paula. **Desenvolvimento de Sistemas de Informação**. 2009.

DALL’OGLIO, Pablo. **PHP: Programando com Orientação a Objetos**. 2007.

FALBO, Ricardo de Almeida. **Engenharia de Software**. 2005. Disponível em:
<<http://www.inf.ufes.br/~falbo/download/aulas/es-g/2005-1/NotasDeAula.pdf>> Acesso em: 05 nov. 2017

FILGUEIRAS, Lúcia V. L.; MELNIKOFF, Selma Shin Shimizu. **Engenharia de Software**. 2000. Disponível em:
<<http://docplayer.com.br/29365784-Engenharia-de-software-teste-de-software-introducao-profa-dra-lucia-v-l-filgueiras-profa-dra-selma-shin-shimizu-melnikoff.html>> Acesso em: 25 out. 2017.

GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; VLISSIDES, John. **Design Patterns: Elements of Reusable Object-Oriented Software**. 1995.

GILMORE, W. Jason. **Beginning PHP and MySQL From Novice to Professional**. 2010

HEUSER, Carlos Alberto. **Projeto de banco de dados : Volume 4 da Série Livros didáticos informática UFRGS**. 2009.

MAIA, Marcio E. F. **Diagramas UML**. 2008. Disponível em:
<<http://disciplinas.lia.ufc.br/engsof081/arquivos/ApresentacaoDiagramas.pdf>>
Acesso em: 31 out. 2017.

MELO, Ana Cristina. **Desenvolvendo Aplicações com UML 2.2**. 2010.

MEW, Kyle. **Aprendendo Material Design**. 2016.

MILANI, André. **MySQL – Guia do Programador**. 2007.

NIEDERAUER, Juliano. **Web Interativa com Ajax e PHP**. 2007.

PIRES, Carlos Eduardo Santos; NASCIMENTO, Rilson Oscar, SALGADO, Ana Carolina. **Comparativo de Desempenho entre Bancos de Dados de Código Aberto**. 2002.

PLANALTO. **LEI Nº 8.069**. 1990. Disponível em:
<http://www.planalto.gov.br/ccivil_03/leis/L8069.htm> Acesso em: 17 set. 2017.

POUNCEY, Ian; YORK, Richard. **Beginning CSS**. 2011.

ROSSBERG, Joachim. **Beginning Application Lifecycle Management**. 2014.

SAVOLA, Tom. **Usando HTML**. 1996.

SEM HORA. **Sem Hora**. Disponível em:
<<https://www.semhora.com.br>> Acesso em: 30 out. 2017.

SILVA, Eliel Geraldino da. **Estabelecimento comercial**. 2015. Disponível em:
<<https://jus.com.br/artigos/35431/estabelecimento-comercial>> Acesso em: 22 nov. 2017.

SILVA, Maurício Samy. **jQuery, a biblioteca do programador JavaScript**. 2010.

VIAPP. **ViApp**. Disponível em:
<<https://itunes.apple.com/br/app/viapp-nomes-na-lista-ingresso/id563925834?mt=8>>
Acesso em: 30 out. 2017.

WAMPSEVER. **WampServer**. Disponível em:
<<http://www.wampserver.com/en/>> Acesso em: 17 set. 2017.

APÊNDICES

APÊNDICE 1 – MER - Modelo Entidade Relacionamento

APÊNDICE 2 – Diagrama de Casos de Uso

APÊNDICE 2.1 – Documentação de Casos de Uso – Realizar Login

APÊNDICE 2.2 – Documentação de Casos de Uso – Tela Principal

APÊNDICE 2.3 – Documentação de Casos de Uso – Manter Estabelecimento

APÊNDICE 2.4 – Documentação de Casos de Uso – Manter Cardápio

APÊNDICE 2.5 – Documentação de Casos de Uso – Manter Usuário

APÊNDICE 2.6 – Documentação de Casos de Uso – Fazer Check-in

APÊNDICE 2.7 – Documentação de Casos de Uso – Verificar Check-in

APÊNDICE 2.8 – Documentação de Casos de Uso – Verificar Extrato

APÊNDICE 2.9 – Documentação de Casos de Uso – Manter Comanda

APÊNDICE 2.10 – Documentação de Casos de Uso – Fazer Pedido

APÊNDICE 2.11 – Documentação de Casos de Uso – Visualizar Cardápio

APÊNDICE 2.12 – Documentação de Casos de Uso – Visualizar Estabelecimento

APÊNDICE 3.1 – Diagrama de Classes - Controllers

APÊNDICE 3.2 – Diagrama de Classes - Models

APÊNDICE 3.3 – Diagrama de Classes – Modelo de Negócios

APÊNDICE 4.1 – Diagrama de Sequência - Cliente

APÊNDICE 4.2 – Diagrama de Sequência - Estabelecimento

APÊNDICE 5 – Diagrama de Atividades

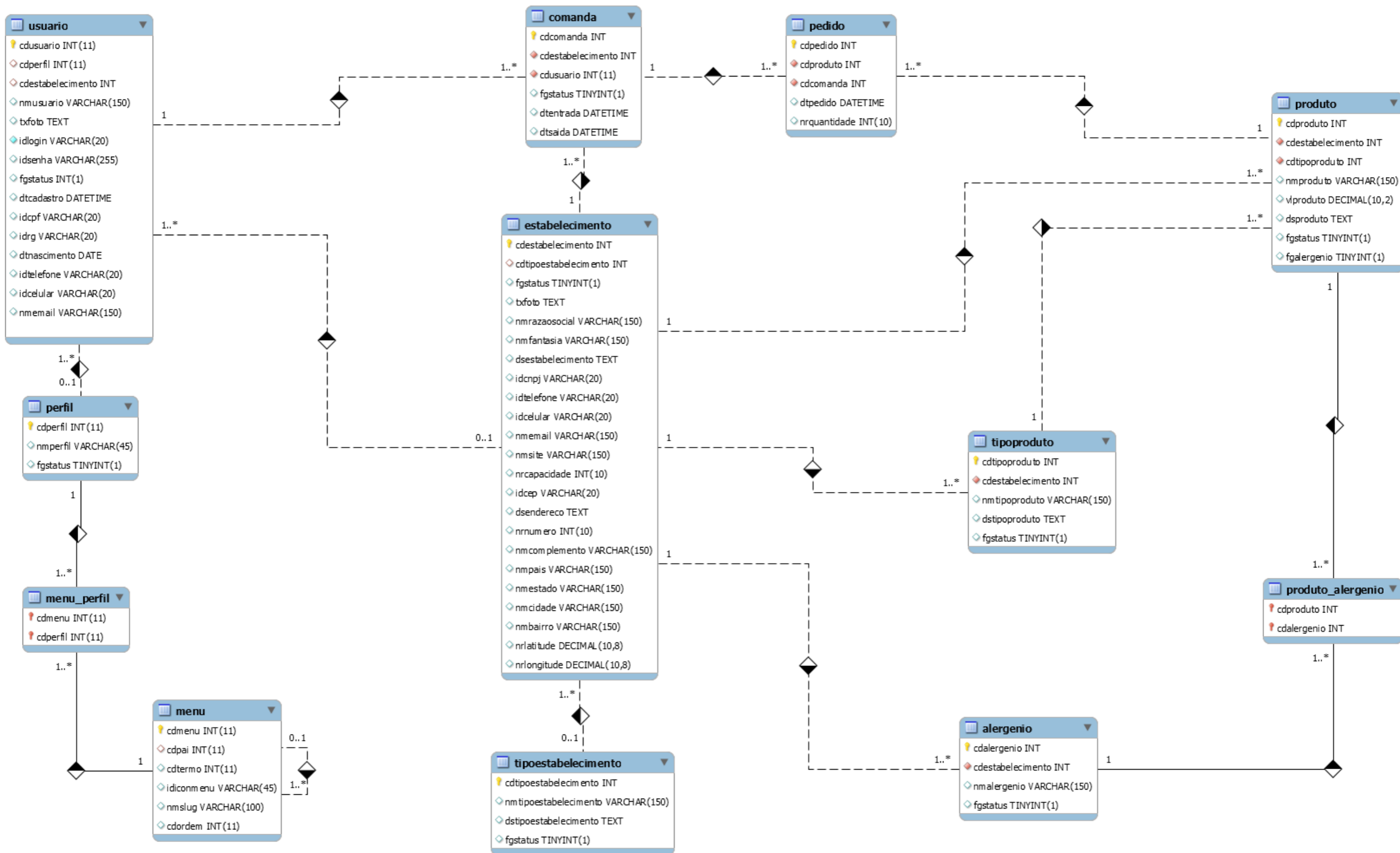
APÊNDICE 6 – WBS

APÊNDICE 7 – Gráfico Gantt

APÊNDICE 8 – Plano de riscos

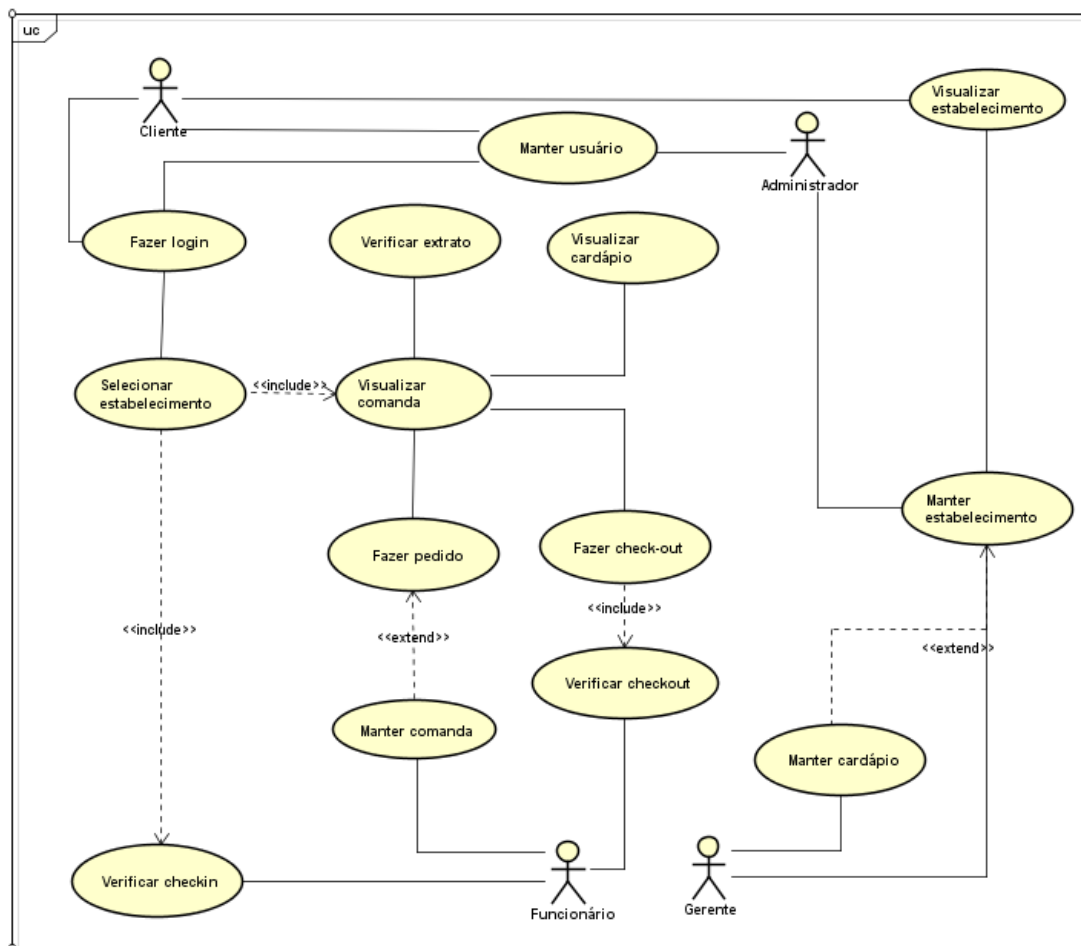
APÊNDICE 1

MER – Modelo Entidade Relacionamento



APÊNDICE 2

Diagrama de Casos de Uso



APÊNDICE 2.1

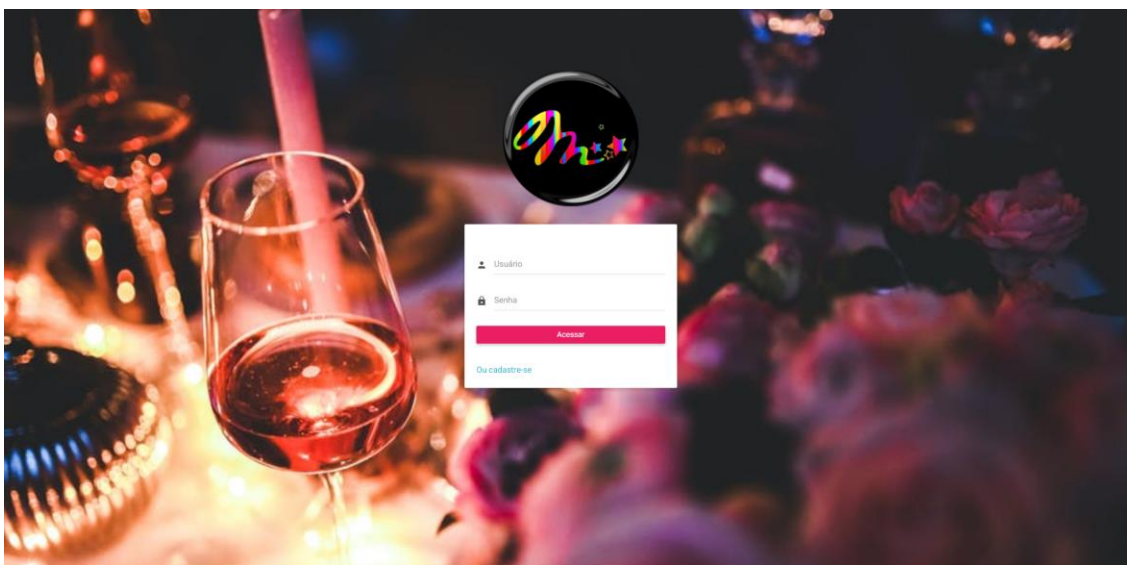
Especificação de Caso de Uso **UC001 – Realizar Login**

Descrição

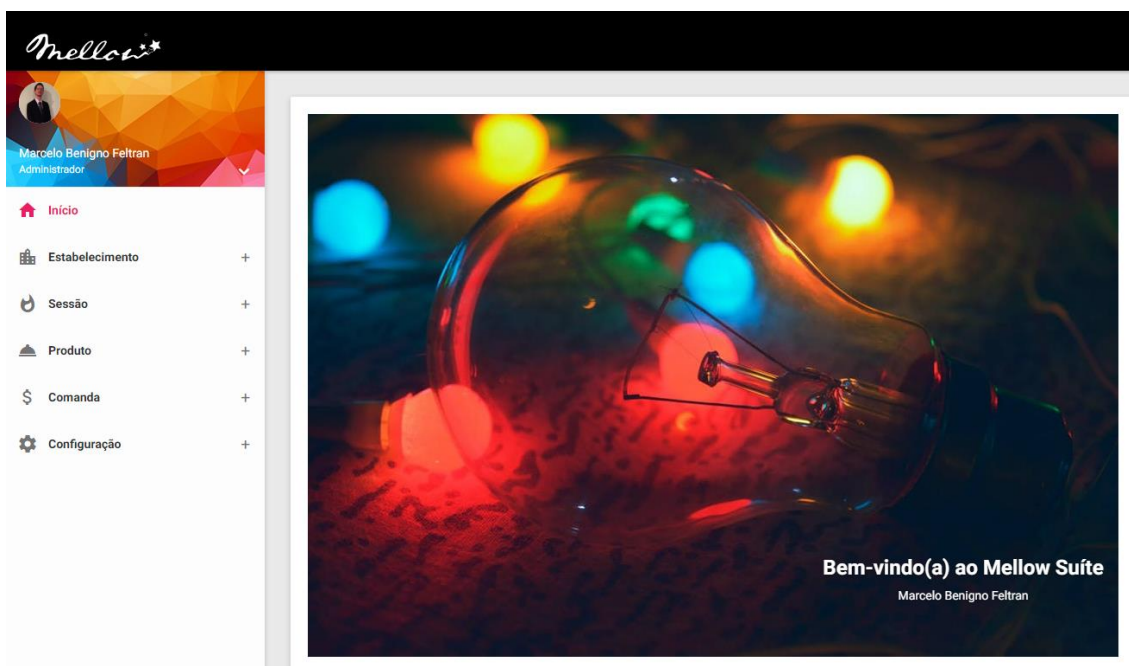
Este é o caso de uso referente ao Login no sistema.

Data View

DV1 – Tela de Login.



DV2 – Tela de Boas Vindas.



Pré-condições

N/A

Pós-condições

Após o fim normal deste caso de uso o usuário deve:

1. Ter logado no sistema.

Ator Primário

Usuário

Fluxo de Eventos Principal

1. O sistema apresenta a tela. [\(DV1\)](#)
2. O usuário preenche o campo login. (A1)
3. O usuário preenche o campo senha.
4. O usuário clica no botão Acessar.
5. O sistema valida as informações inseridas.
6. O sistema realiza o Login do usuário. [\(E1\)\(E2\)\(R1\)](#)
7. O sistema chama o caso de uso [\(UC002 - Tela Principal\)](#)
8. O caso de uso é finalizado.

Fluxos Alternativos

A1. O usuário clica na opção “Cadastrar-se”.

1. O sistema chama o caso de uso (UC005 – Manter Usuário).
2. O caso de uso é encerrado.

Fluxos de Exceção

E1. Campos obrigatórios não preenchidos:

1. O sistema verifica que o campo não foi preenchido.
2. O sistema retorna a mensagem “Preencha o campo X”.
3. O sistema destaca os campos de preenchimento obrigatório que não foram preenchidos em vermelho e negrito.

E2. Usuário errou o login/senha:

1. O sistema verifica que a senha está errada.
2. O sistema retorna a mensagem “login ou senha incorretos”
3. O caso de uso é reiniciado.

Regras de Negócio

R1. O usuário precisa estar cadastrado previamente no sistema.

APÊNDICE 2.2

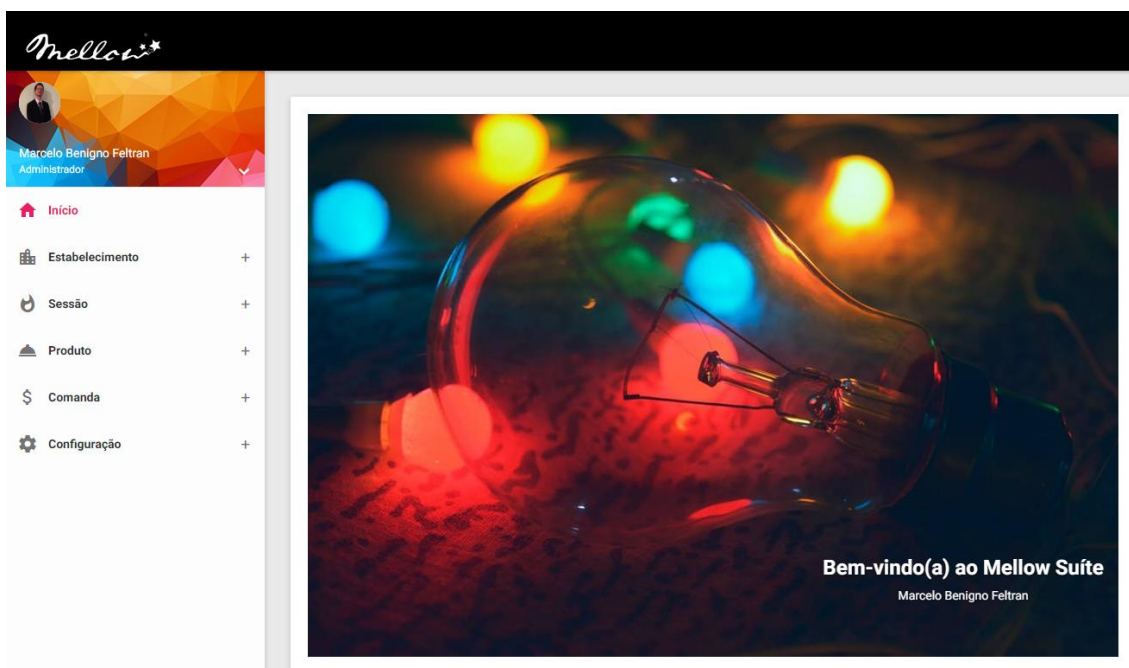
Especificação de Caso de Uso **UC002 –Tela Principal**

Descrição

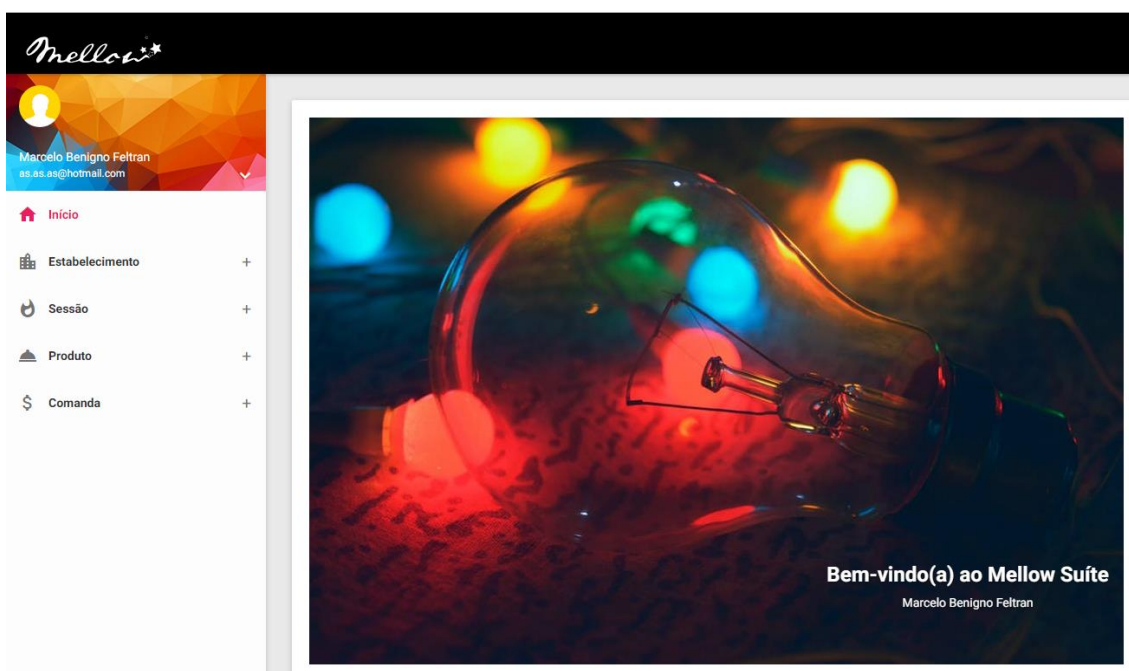
Este é o caso de uso referente a Tela Principal do sistema.

Data View

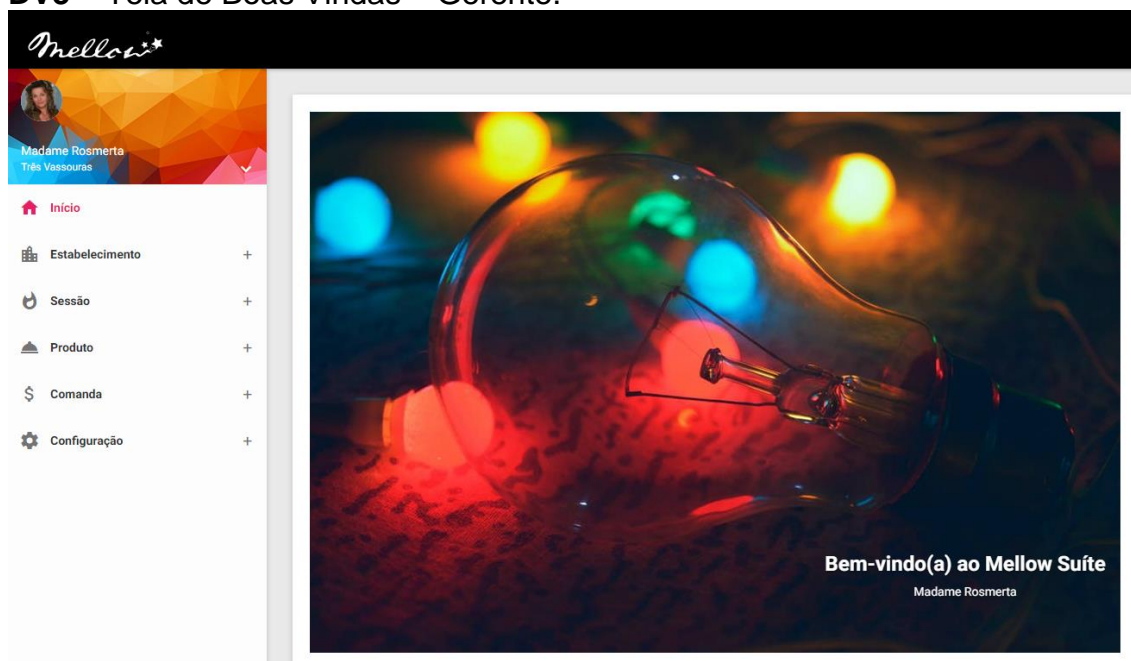
DV1 – Tela de Boas Vindas - Admin.



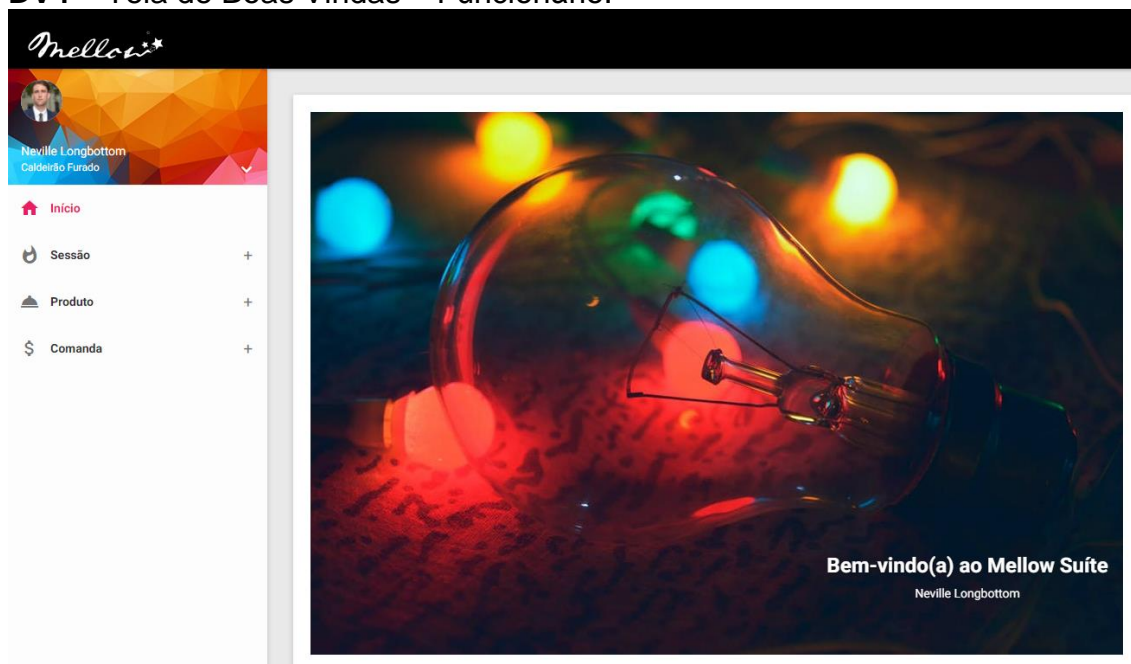
DV2 – Tela de Boas Vindas - Cliente.



DV3 – Tela de Boas Vindas – Gerente.



DV4 – Tela de Boas Vindas – Funcionário.



Pré-condições

Este caso de uso pode iniciar somente se:

1. O usuário estiver logado no sistema.

Pós-condições

1. N/A

Ator Primário

Administrador

Fluxo de Eventos Principal

1. O sistema verifica qual perfil de usuário está logado.
2. O sistema verifica que o usuário é administrador. [\(A1\)\(A2\)\(A3\)](#)
3. O sistema apresenta a tela [DV1](#).
4. O Administrador clica na opção estabelecimento. [\(A4\)\(A5\)](#)
5. O sistema chama o caso de uso [\(UC003 – Manter Estabelecimento\)](#)

Fluxos Alternativos

- A1. O sistema verifica que o usuário é Cliente
1. O sistema apresenta a tela [\(DV2\)](#).
 2. O Cliente clica na opção “Estabelecimentos”. [\(A6\)\(A7\)\(A8\)](#)
 3. O sistema chama o caso de uso [\(UC012 – Visualizar Estabelecimento\)](#)
- A2. O sistema verifica que o usuário é Gerente
1. O sistema apresenta a tela [\(DV3\)](#).
 2. O Gerente clica na opção “Cardápio”. [\(A9\)](#)
 3. O sistema chama o caso de uso [\(UC004 – Manter Cardápio\)](#).
- A3. O sistema verifica que o usuário é Funcionário.
1. O sistema apresenta a tela [\(DV4\)](#).
 2. O Funcionário clica na opção “Comanda”. [\(A10\)](#)
 3. O sistema chama o caso de uso [\(UC009 – Manter Comanda\)](#).
- A4. O Administrador clica na opção Cliente
1. O sistema chama o caso de uso [\(UC005 – Manter Usuário\)](#).
- A5. O Administrador clica no botão sair
1. O sistema invalida a sessão ativa.
 2. O sistema chama o caso de uso [\(UC001 – Login\)](#).
- A6. O Cliente clica na opção “Check-in”.
1. O sistema chama o caso de uso [\(UC006 – Fazer Check-in\)](#).
- A7. O Cliente clica na opção “Visualizar Comanda”.
1. O sistema chama o caso de uso [\(UC013 – Visualizar Comanda\)](#).
- A8. O Cliente clica na opção “Visualizar Cardápio”.
1. O sistema chama o caso de uso [\(UC011 – Visualizar Cardápio\)](#).
- A9. O Gerente clica na opção “Estabelecimento”.
1. O sistema chama o caso de uso [\(UC003 – Manter Estabelecimento\)](#).

A10. O Funcionário clica na opção “Pedido”.

1. O sistema chama o caso de uso [\(UC010 – Fazer Pedido\)](#)

Fluxos de Exceção

1. N/A

Regras de Negócio

1. N/A

APÊNDICE 2.3

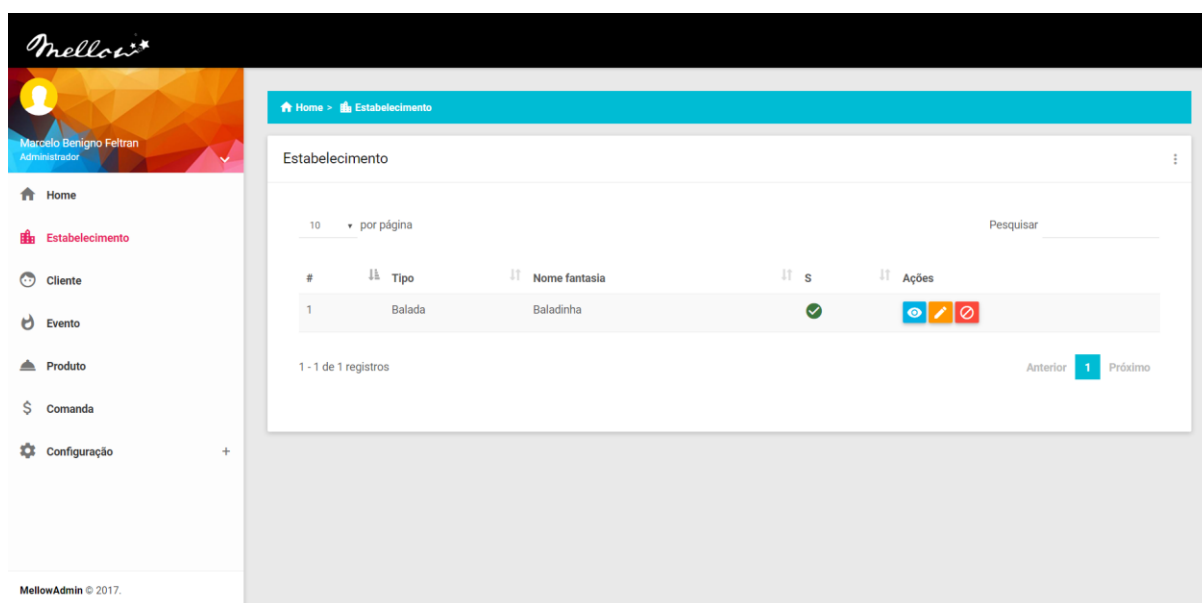
Especificação de Caso de Uso **UC003 – Manter Estabelecimento**

Descrição

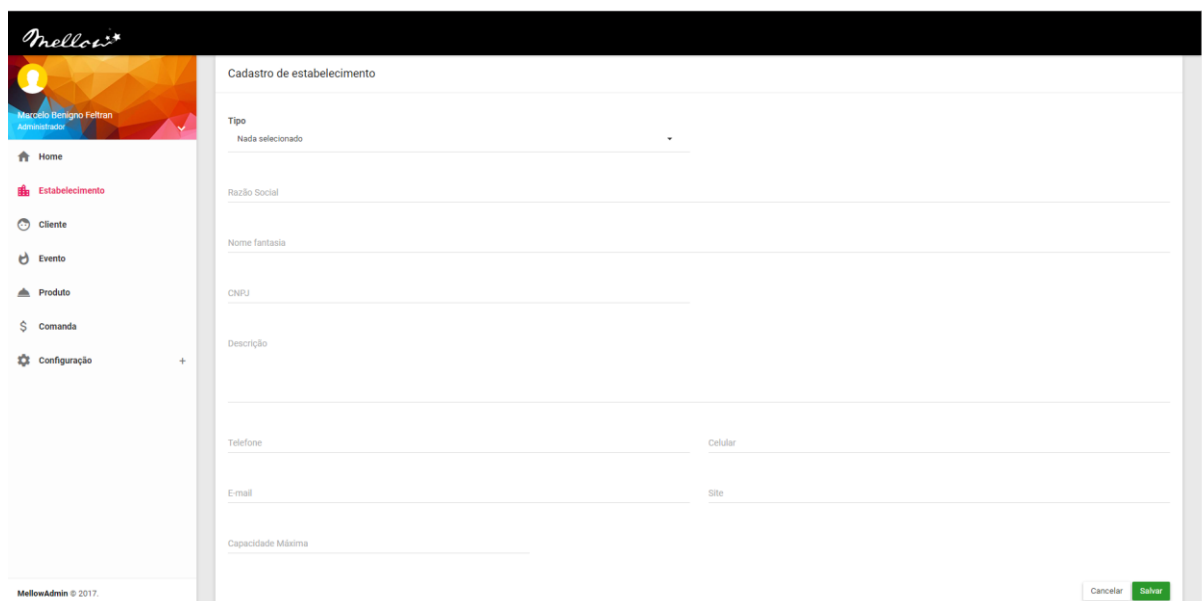
Esse caso de uso tem o propósito de cadastrar ou editar os estabelecimentos no sistema

Data View

DV1 – Tela Visualizar Estabelecimentos.



DV2 – Tela Cadastro Estabelecimento



Pré-condições

Esse caso de uso pode ser iniciado somente se:

1. O usuário for Administrador.
2. O usuário for Gerente.

Pós-condições

O sistema ter salvo quaisquer alterações realizadas.

Ator Primário

Administrador/Gerente

Fluxo de Eventos Principal

1. O sistema busca no banco de dados os estabelecimentos já cadastrados de acordo com o perfil do usuário logado.
2. O sistema preenche a lista “Estabelecimentos” com os dados encontrados.
3. O sistema apresenta a tela (DV1).
4. O Administrador clica no botão “Incluir”.(A1)(A2)
5. O sistema apresenta a tela (DV2).
6. O Administrador preenche os campos apresentados.
7. O Administrador clica no botão “Salvar”.
8. O sistema valida os dados inseridos.
9. O sistema salva os dados contidos no formulário. (E1)(E2)
10. O sistema busca os estabelecimentos já cadastrados no banco.
11. O sistema apresenta a tela (DV1)
12. O caso de uso é encerrado.

Fluxos Alternativos

- A1. O Administrador clica no botão “Editar Estabelecimento”.
1. O sistema busca no banco os dados do estabelecimento escolhido.
 2. O sistema preenche o formulário com os dados buscados.
 3. O sistema apresenta a tela (DV2) com o formulário preenchido.
 4. O Administrador faz as alterações pertinentes.
 5. O Administrador clica no botão “Salvar”.
 6. O sistema valida os dados inseridos.
 7. O sistema salva os dados contidos no formulário. (E1)(E2)
 8. O sistema busca os estabelecimentos já cadastrados no banco.
 9. O sistema preenche a lista “Estabelecimentos” com os dados encontrados.
 10. O sistema apresenta a tela (DV1).
 11. O caso de uso é encerrado.
- A2. O usuário clica no botão “Visualizar Estabelecimento”.
1. O sistema busca no banco os dados do estabelecimento escolhido.
 2. O sistema preenche o formulário com os dados buscados.
 3. O sistema bloqueia os campos do formulário para edição.
 4. O sistema apresenta a tela (DV2) com o formulário preenchido.
 5. O Administrador clica no botão retornar.
 6. O sistema apresenta a tela (DV1).

7. O caso de uso é encerrado.

Fluxos de Exceção

E1. O sistema verifica que um campo obrigatório não foi preenchido.

1. O sistema pinta de vermelho o campo não preenchido.
2. O sistema leva o foco para o campo não preenchido.
3. O sistema emite a mensagem “Campo obrigatório”.
4. Retorna ao fluxo principal.

E2. O sistema verifica que os dados não estão de acordo com a máscara do campo.

1. O sistema pinta de vermelho o campo com dados fora do padrão.
2. O sistema leva o foco para o campo incorreto.
3. O sistema emite a mensagem “Siga o padrão de preenchimento do campo”.
4. Retorna ao fluxo principal.

Regras de Negócio

1. O Estabelecimento precisa ter o CNPJ válido.

APÊNDICE 2.4

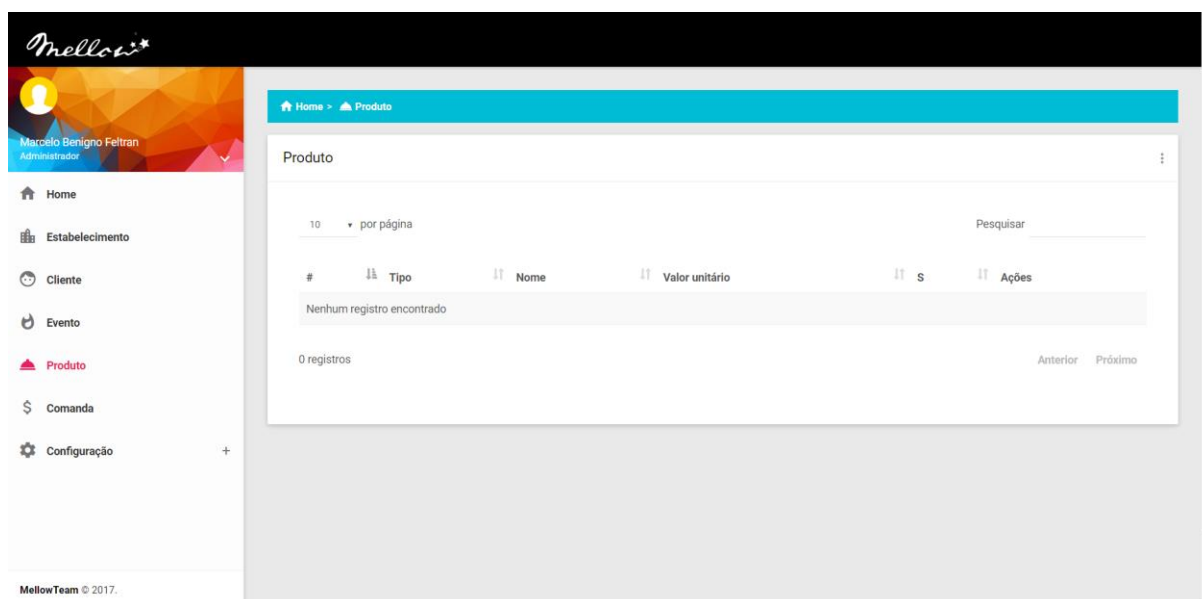
Especificação de Caso de Uso **UC004 – Manter Cardápio**

Descrição

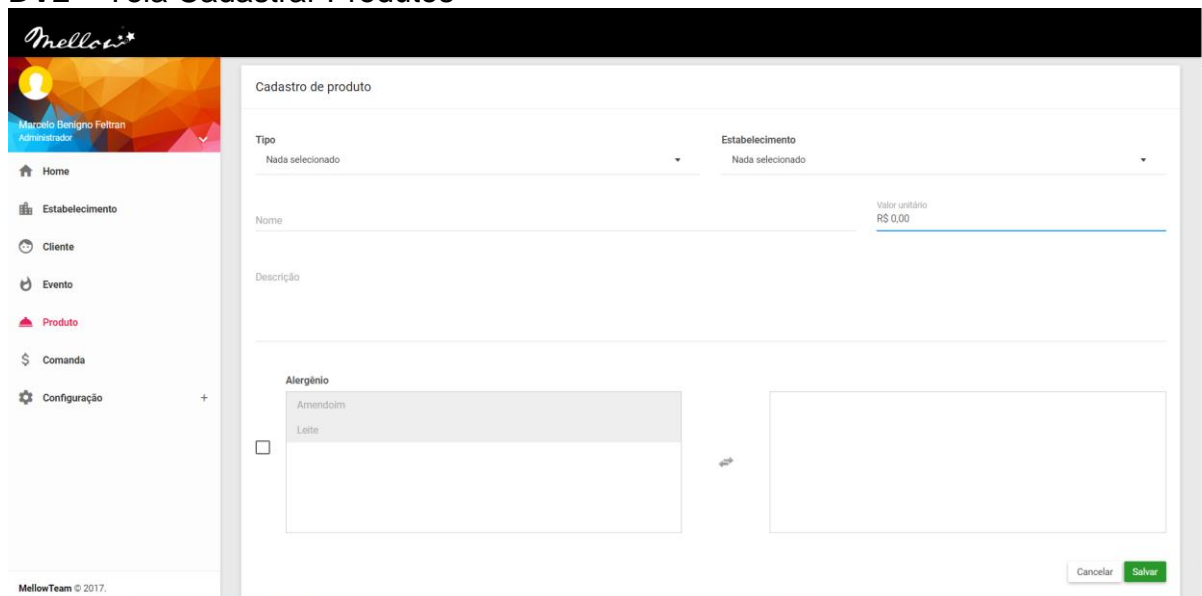
Esse caso de uso tem a finalidade de cadastrar e editar os produtos de um estabelecimento

Data View

DV1 – Tela Visualizar Produtos



DV2 – Tela Cadastrar Produtos



Pré-condições

Esse caso de uso pode ser iniciado somente se:

1. O usuário for Administrador.
2. O usuário for Gerente.

Pós-condições

O sistema ter salvo quaisquer alterações realizadas.

Ator Primário

Administrador/Gerente

Fluxo de Eventos Principal

1. O sistema busca no banco de dados os produtos cadastrados.
2. O sistema preenche a lista “Produtos” com os dados encontrados.
3. O sistema apresenta a tela (DV1).
4. O usuário clica no botão “Visualizar”. (A1)(A2)(A3)(A4)
5. O sistema busca no banco de dados os dados do produto selecionado.
6. O sistema bloqueia os campos para edição.
7. O sistema preenche os campos com os dados buscados.
8. O sistema apresenta a tela (DV2).
9. O usuário clica no botão retornar. (A1)
10. O sistema apresenta a tela (DV1).
11. O caso de uso é encerrado.

Fluxos Alternativos

- A1. O usuário clica no botão “Editar”.
1. O sistema busca no banco de dados os dados do produto selecionado.
 2. O sistema preenche os campos com os dados encontrados.
 3. O sistema apresenta a tela (DV2).
 4. O usuário edita as informações pertinentes.
 5. O usuário clica no botão “Salvar”. (A5)
 6. O sistema valida os dados inseridos.
 7. O sistema salva os dados no banco. (E1)
 8. O caso de uso é encerrado.
- A2. O usuário clica no botão “Inativar”.
1. O sistema emite o alerta “Deseja inativar esse produto?”.
 2. O usuário clica na opção “Inativar”. (A6)
 3. O sistema altera o status do produto para inativo.
 4. O sistema busca no banco de dados os produtos cadastrados e ativos.
 5. O sistema preenche a lista “Produtos”.
 6. O sistema apresenta a tela (DV1).
 7. O caso de uso é encerrado.
- A3. O usuário clica no campo “Pesquisar”.
1. O usuário digita parte da descrição do produto desejado.
 2. O sistema busca na lista os produtos que batem com a busca.
 3. O sistema preenche a lista “Produtos” com o resultado da busca.
 4. O caso de uso é encerrado.

A4. O usuário clica no botão “Incluir”.

1. O sistema apresenta a tela (DV2).
2. O usuário preenche os campos necessários.
3. O usuário clica no botão “Salvar”.
4. O sistema valida os dados inseridos.
5. O sistema salva as alterações realizadas. (E1)
6. O caso de uso é encerrado.

Fluxos de Exceção

E1. O sistema verifica que um campo obrigatório não foi preenchido.

1. O sistema pinta de vermelho o campo não preenchido.
2. O sistema leva o foco para o campo não preenchido.
3. O sistema emite a mensagem “Campo obrigatório”.
4. Retorna ao fluxo principal.

Regras de Negócio

N/A

APÊNDICE 2.5

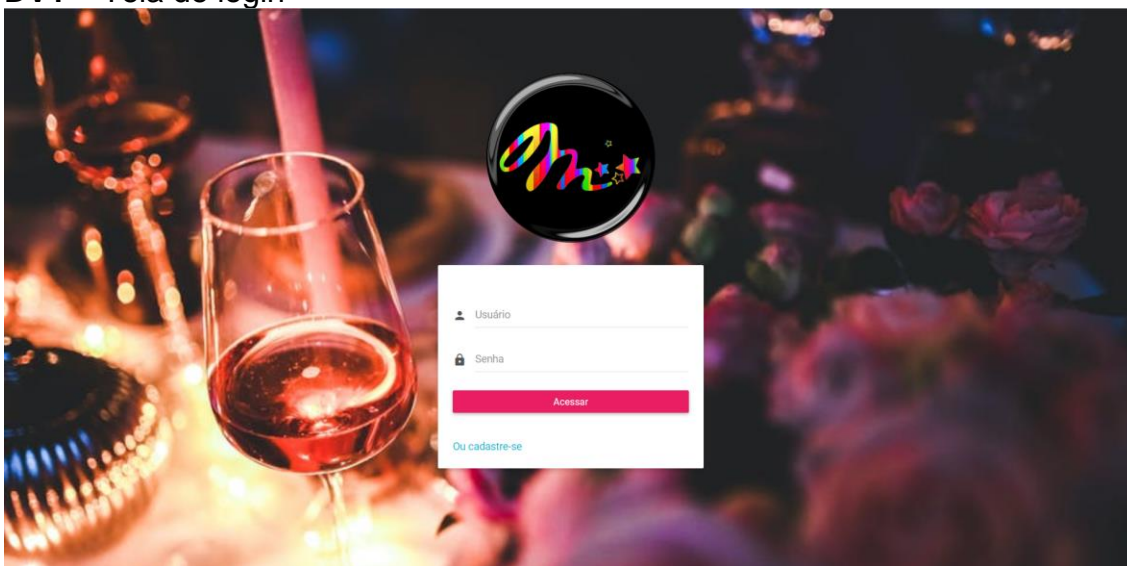
Especificação de Caso de Uso **UC005 – Manter Usuário**

Descrição

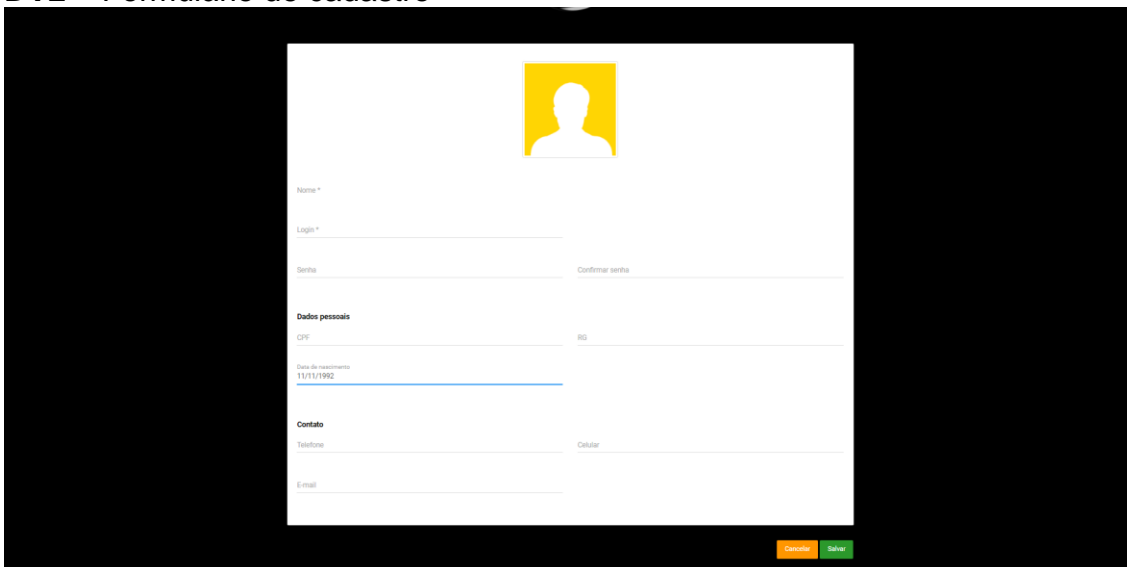
Esse caso de uso tem o objetivo de cadastrar ou editar um usuário no sistema.

Data View

DV1 – Tela de login



DV2 – Formulário de cadastro



Pré-condições

Esse caso de uso pode ser iniciado somente se:

1. O usuário estiver logado no sistema.

Pós-condições

O sistema ter salvo quaisquer alterações realizadas.

Ator Primário

Administrador/Cliente

Fluxo de Eventos Principal

1. O sistema apresenta a tela (DV1).
2. O usuário clica na opção “Cadastrar-se”. (A1)
3. O sistema chama a tela (DV2).
4. O usuário preenche o formulário com seus dados.
5. O sistema valida os dados preenchidos.
6. O sistema salva os dados no banco. (E1)(E2)
7. O caso de uso é encerrado.

Fluxos Alternativos

N/A

Fluxos de Exceção

E1. O sistema verifica que um campo obrigatório não foi preenchido.

1. O sistema emite a mensagem “Campo não preenchido”.
2. O sistema pinta de vermelho o campo não preenchido.
3. O sistema define o foco para o campo não preenchido.
4. Retorna ao fluxo principal.

E2. O sistema verifica que um campo foi preenchido incorretamente.

1. O sistema emite a mensagem “Campo preenchido incorretamente”.
2. O sistema pinta de vermelho o campo incorreto.
3. O sistema define o foco para o campo incorreto.
4. Retorna ao fluxo principal.

Regras de Negócio

1. O cpf deve ser válido, de acordo com os dígitos verificadores.
2. O email deve estar no formato xxx@xxx.com

APÊNDICE 2.6

Especificação de Caso de Uso

UC006 – Fazer Check-in

Descrição

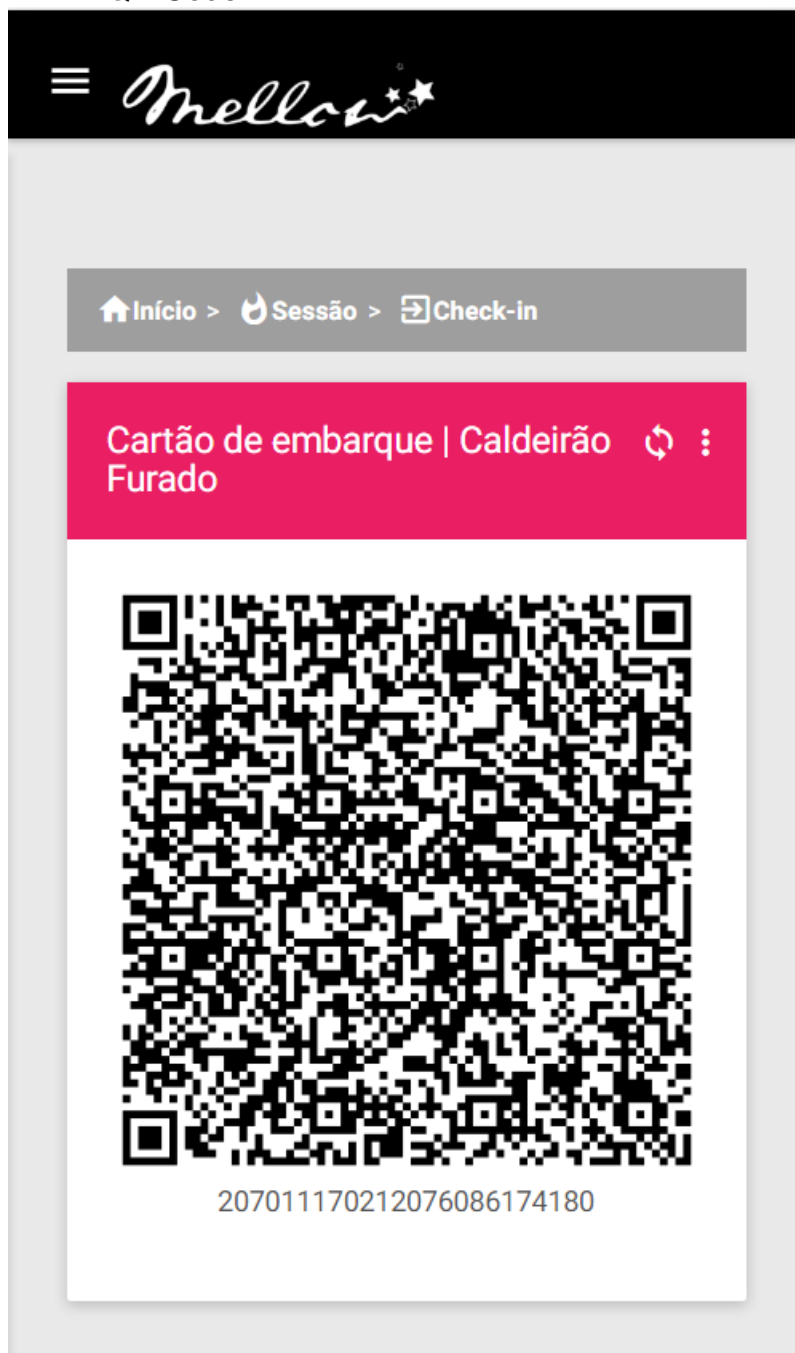
Esse caso de uso tem o propósito de realizar o check-in do usuário dentro de um estabelecimento

Data View

DV1 – Estabelecimentos cadastrados



DV2 – QR Code

**Pré-condições**

Esse caso de uso pode ser iniciado somente se:

1. O usuário ter o perfil cliente.

Pós-condições

O usuário ter uma comanda aberta em seu nome, no estabelecimento desejado.

Ator Primário

Cliente

Fluxo de Eventos Principal

1. O usuário clica na opção "Check-In".
2. O sistema busca no banco de dados os estabelecimentos cadastrados.
3. O sistema preenche a lista "Estabelecimentos".
4. O sistema apresenta a tela (DV1).
5. O usuário clica em um estabelecimento.
6. O sistema gera um código e o liga ao usuário conectado.
7. O sistema gera um QR Code para esse código.
8. O sistema apresenta a tela (DV2).
9. O caso de uso é encerrado.

Fluxos Alternativos

N/A

Fluxos de Exceção

N/A

Regras de Negócio

N/A

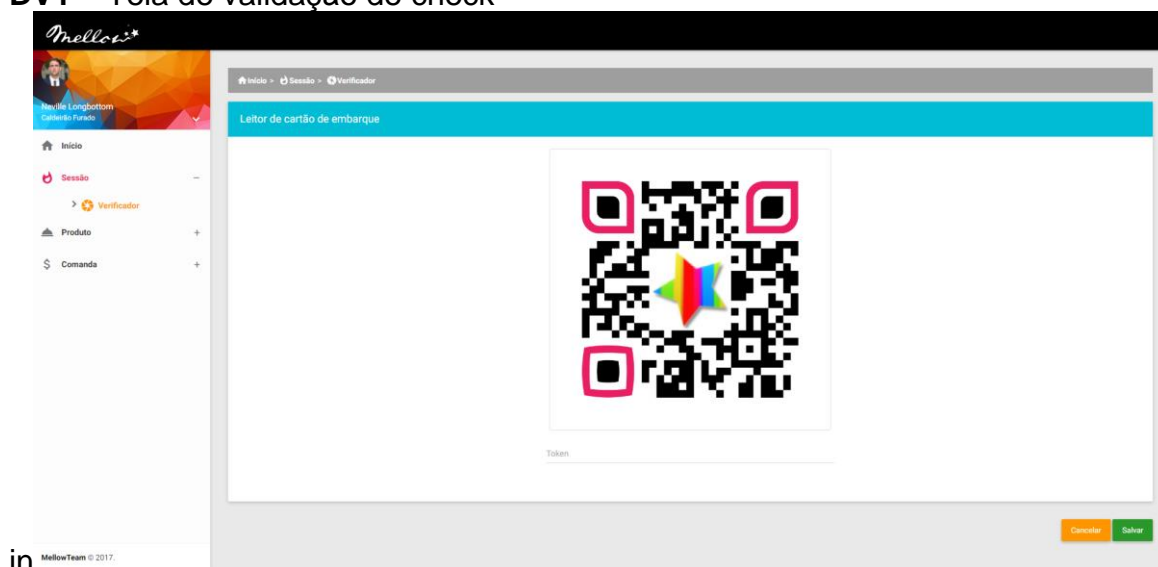
APÊNDICE 2.7

Especificação de Caso de Uso UC007 – Verificar Check-in

Descrição

Data View

DV1 – Tela de validação de check-



Pré-condições

Esse caso de uso pode ser iniciado somente se:

1. O usuário for Funcionário.

Pós-condições

Uma comanda será atrelada ao Cliente.

Ator Primário

Funcionário

Fluxo de Eventos Principal

1. O sistema apresenta a tela (DV1).
2. O Funcionário clica no botão “Escanear código”.
3. O Funcionário tira uma foto do QR Code do Cliente.
4. O sistema lê o QR Code.
5. O sistema cria uma comanda para o Cliente.
6. O sistema emite a mensagem “Comanda criada com sucesso”.
7. O caso de uso é encerrado.

Fluxos Alternativos

N/A

Fluxos de Exceção

N/A

Regras de Negócio

N/A

APÊNDICE 2.8

Especificação de Caso de Uso **UC008 – Verificar Extrato**

Descrição

Data View

DV1 –

Pré-condições

Esse caso de uso pode ser iniciado somente se:

1. O usuário estiver com uma comanda aberta em um estabelecimento.

Pós-condições

O usuário ter o extrato de consumo atual.

Ator Primário

Cliente

Fluxo de Eventos Principal

1. O usuário clica na opção “Extrato”.
2. O sistema busca no banco de dados, o registro de consumo do usuário.
3. O sistema preenche a lista com os dados encontrados.
4. O sistema apresenta a tela (DV1).
5. O caso de uso é encerrado.

Fluxos Alternativos

N/A

Fluxos de Exceção

N/A

Regras de Negócio

N/A

APÊNDICE 2.9

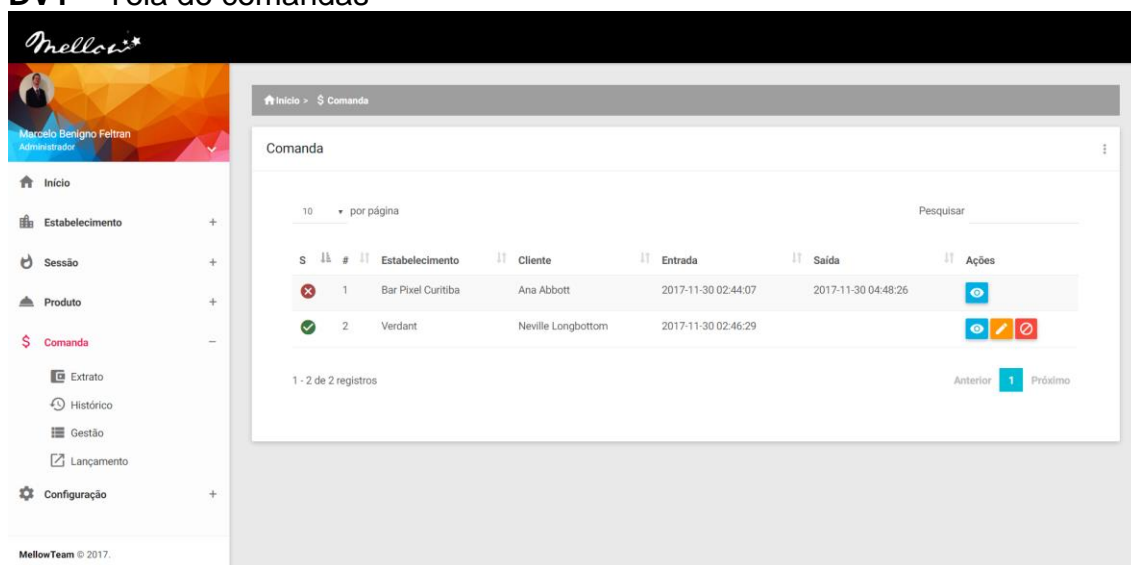
Especificação de Caso de Uso **UC009 – Manter Comanda**

Descrição

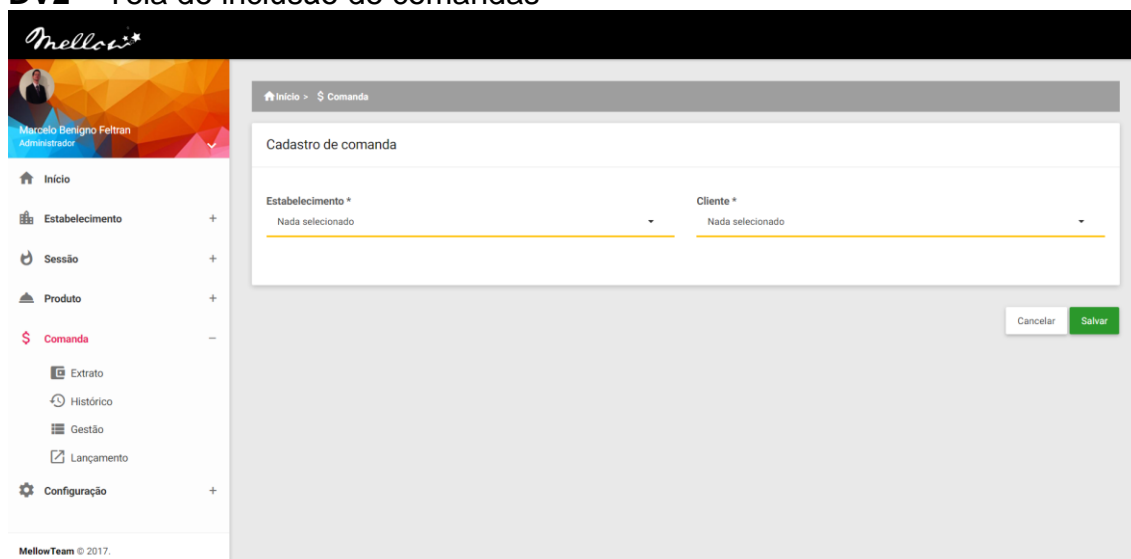
Esse caso de uso tem o propósito de abrir e fechar comandas.

Data View

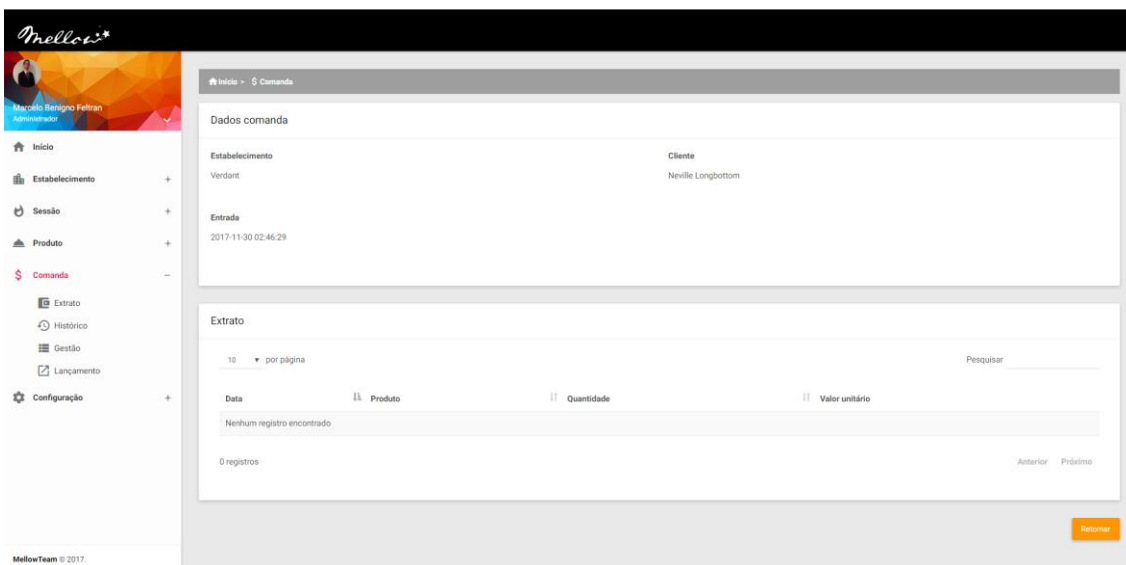
DV1 – Tela de comandas



DV2 – Tela de inclusão de comandas



DV3 – Tela de Detalhes



Pré-condições

Esse caso de uso pode ser iniciado somente se:

1. O usuário tiver o perfil “Funcionário”.

Pós-condições

O sistema ter salvo quaisquer alterações realizadas.

Ator Primário

Funcionário

Fluxo de Eventos Principal

1. O sistema busca no banco de dados as comandas registradas.
2. O sistema preenche a lista “Comandas”.
3. O sistema apresenta a tela (DV1).
4. O usuário clica na opção “Detalhes”. (A1)(A2)(A3)
5. O sistema busca no banco, os dados do registro escolhido.
6. O sistema preenche o formulário com as informações encontradas.
7. O sistema bloqueia os campos para edição.
8. O sistema apresenta a tela (DV3).
9. O usuário clica em retornar.
10. O caso de uso é encerrado.

Fluxos Alternativos

- A1. O usuário clica na opção “Editar”.
 1. O sistema busca no banco, os dados do registro escolhido.
 2. O sistema preenche o formulário com os dados encontrados.
 3. O usuário realiza as alterações pertinentes.
 4. O usuário clica em “Salvar”. (A4)
 5. O sistema valida os dados inseridos.
 6. O sistema salva os dados no banco. (E1)(E2)
 7. O caso de uso é encerrado.

- A2. O usuário clica na opção “Inativar”.
1. O sistema emite o alerta “Tem certeza que deseja inativar este registro?”.
 2. O usuário clica na opção “Sim”. (A4)
 3. O sistema altera o status da comanda para inativo.
 4. O sistema busca no banco de dados as comandas registradas.
 5. O sistema preenche o grid com os dados encontrados.
 6. O sistema apresenta a tela (DV3).
 7. O caso de uso é encerrado.
- A3. O usuário clica na opção “Incluir”.
1. O sistema apresenta a tela (DV2)
 2. O usuário preenche os campos pertinentes.
 3. O usuário clica na opção “Salvar” (A4).
 4. O sistema valida os dados inseridos.
 5. O sistema salva os dados no banco. (E1)(E2);
 6. O caso de uso é encerrado.
- A4. O usuário clica na opção “Cancelar”.
1. O sistema descarta as alterações realizadas.
 2. Retorna ao fluxo principal.

Fluxos de Exceção

- E1. O sistema verifica que campos obrigatórios não foram preenchidos.
1. O sistema emite a mensagem “Campo não preenchido”.
 2. O sistema pinta o campo não preenchido de vermelho.
 3. O sistema define o foco no campo não preenchido.
 4. Retorna ao fluxo principal.
- E2. O sistema verifica que algum campo foi preenchido incorretamente.
1. O sistema emite a mensagem “Preenchimento incorreto”.
 2. O sistema pinta o campo incorreto de vermelho.
 3. O sistema define o foco no campo preenchido incorretamente.
 4. Retorna ao fluxo principal.

Regras de Negócio

N/A

APÊNDICE 2.10

Especificação de Caso de Uso **UC010 – Fazer Pedido**

Descrição

Esse caso de uso tem a finalidade de salvar um pedido no sistema.

Data View

DV1 –

DV2 –

Pré-condições

Esse caso de uso pode ser iniciado somente se:

1. O usuário for Funcionário.

Pós-condições

O sistema ter salvo o pedido realizado na comanda do cliente.

Ator Primário

Funcionário

Fluxo de Eventos Principal

1. O sistema busca no banco de dados os produtos disponíveis.
2. O sistema preenche a lista de produtos com os dados buscados.
3. O sistema apresenta a tela (DV1).
4. O usuário clica em “Incluir Produto”.
5. O sistema insere o produto selecionado na comanda do cliente.
6. O usuário clica em “Salvar”. (A1)
7. O sistema salva as alterações no banco.
8. O caso de uso é encerrado.

Fluxos Alternativos

- A1. O usuário clica na opção “Cancelar”.
1. O sistema descarta as alterações.
 2. O caso de uso é encerrado.

Fluxos de Exceção

N/A

Regras de Negócio

N/A

APÊNDICE 2.11

Especificação de Caso de Uso **UC011 – Visualizar Cardápio**

Descrição

Esse caso de uso tem como objetivo mostrar o cardápio disponível.

Data View

DV1 –



Pré-condições

Esse caso de uso pode ser iniciado somente se:

1. O usuário tiver realizado o Login.

Pós-condições

O usuário terá uma lista com todos os produtos disponíveis para compra.

Ator Primário

Cliente.

Fluxo de Eventos Principal

1. O sistema busca no banco de dados os produtos cadastrados.
2. O sistema preenche a lista “Produtos” com os dados buscados.
3. O sistema apresenta a tela (DV1).
4. O caso de uso é encerrado.

Fluxos Alternativos

N/A

Fluxos de Exceção

N/A

Regras de Negócio

N/A

APÊNDICE 2.12

Especificação de Caso de Uso
UC012 – Visualizar Estabelecimento

Descrição

Esse caso de uso tem como objetivo mostrar os estabelecimentos disponíveis.

Data View

DV1 – Estabelecimentos Cadastrados



Pré-condições

Esse caso de uso pode ser iniciado somente se:

1. O usuário tiver realizado o Login.

Pós-condições

O usuário terá uma lista com todos os estabelecimentos disponíveis.

Ator Primário

Cliente.

Fluxo de Eventos Principal

1. O sistema busca no banco de dados os estabelecimentos cadastrados.
2. O sistema preenche a lista “Estabelecimentos” com os dados buscados.
3. O sistema apresenta a tela (DV1).
4. O caso de uso é encerrado.

Fluxos Alternativos

N/A

Fluxos de Exceção

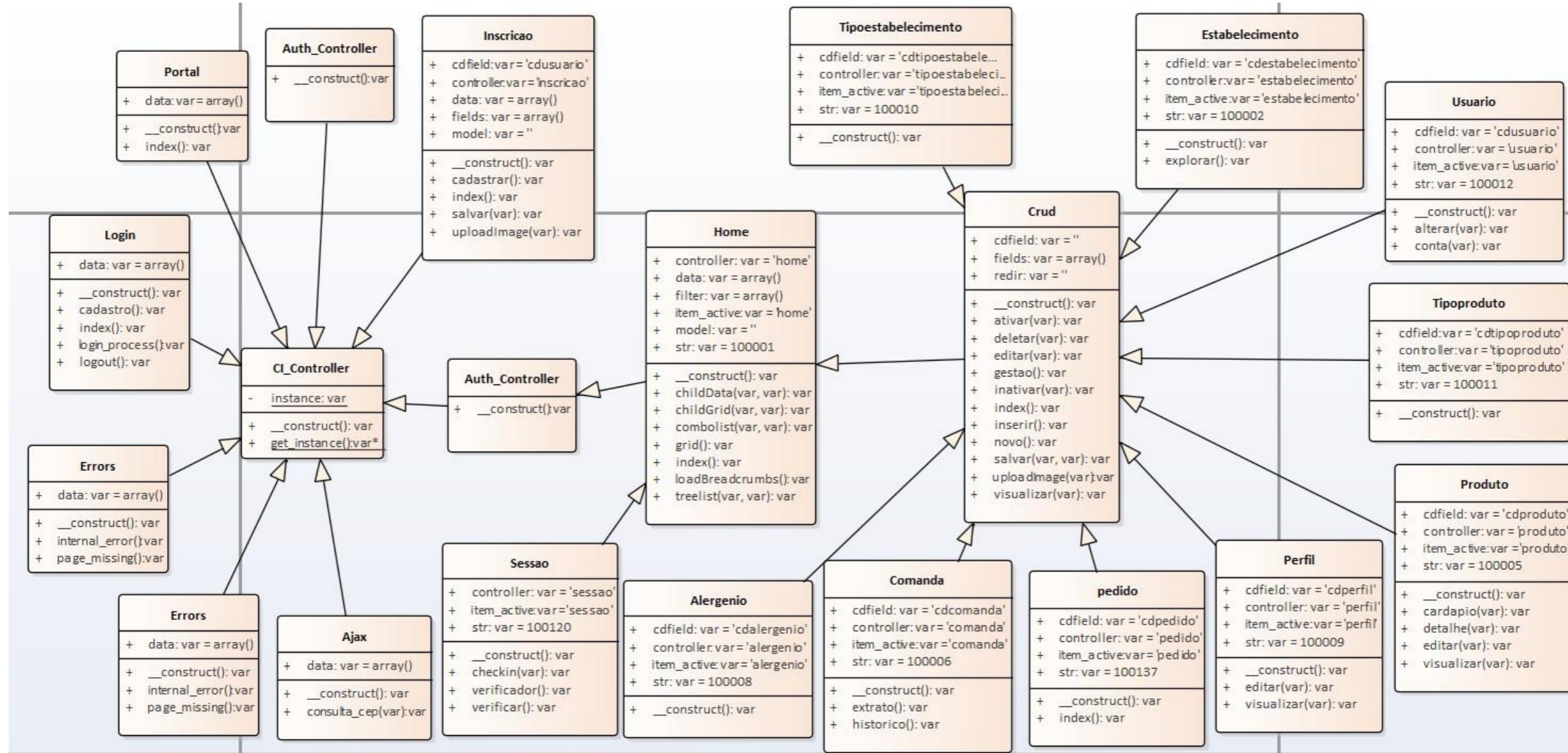
N/A

Regras de Negócio

N/A

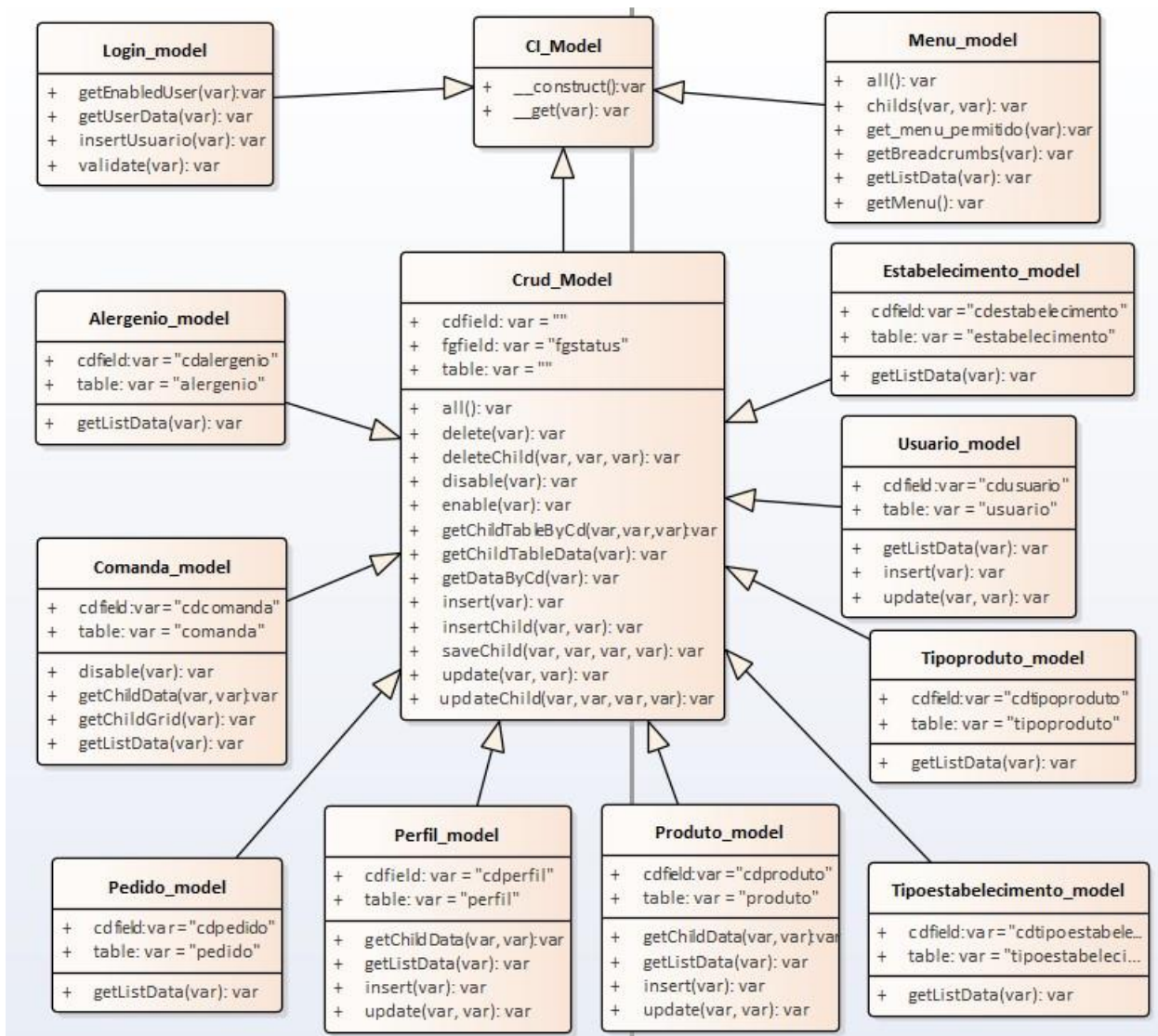
APÊNDICE 3.1

Diagrama de Classes – Controllers



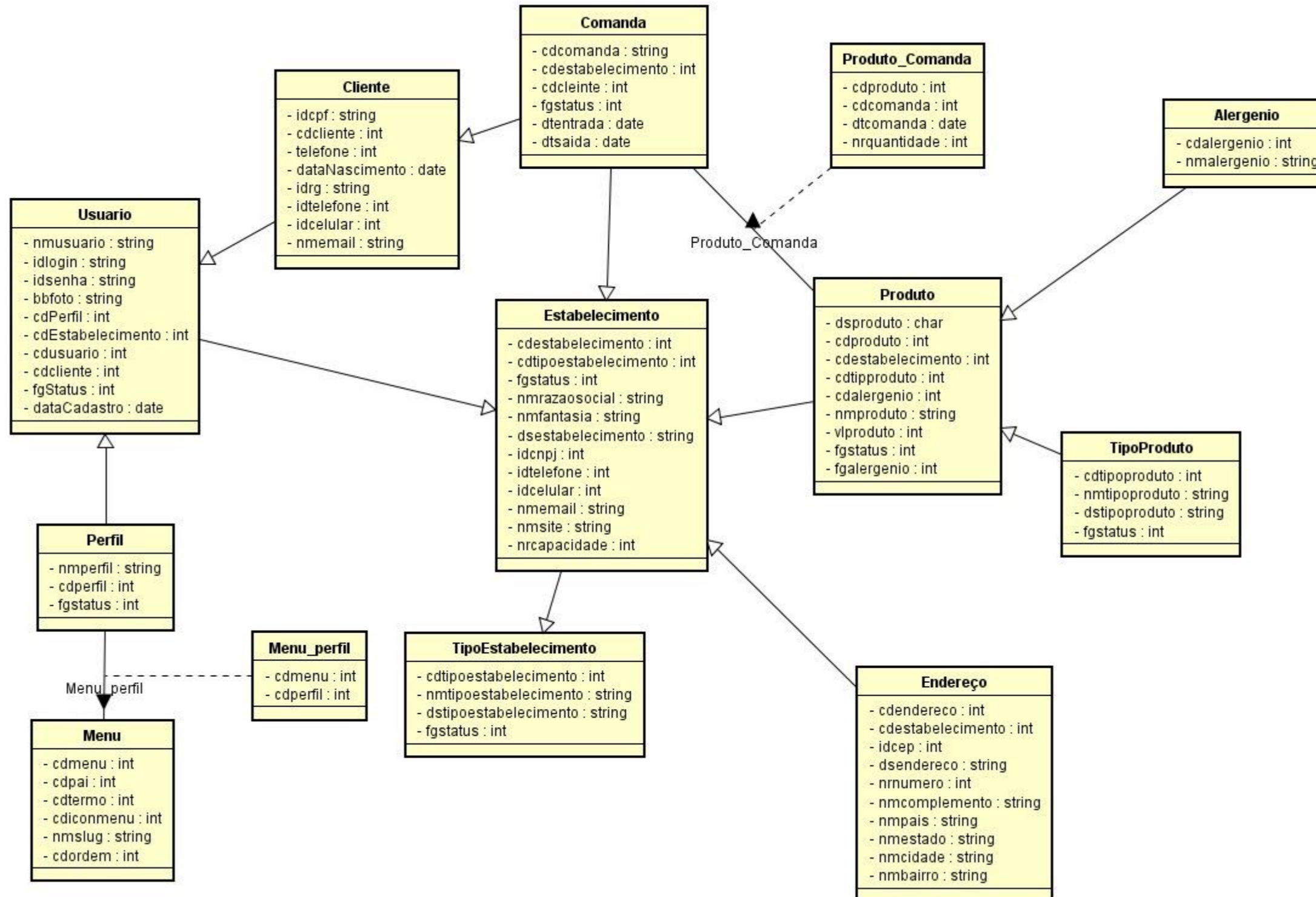
APÊNDICE 3.2

Diagrama de Classes - Models



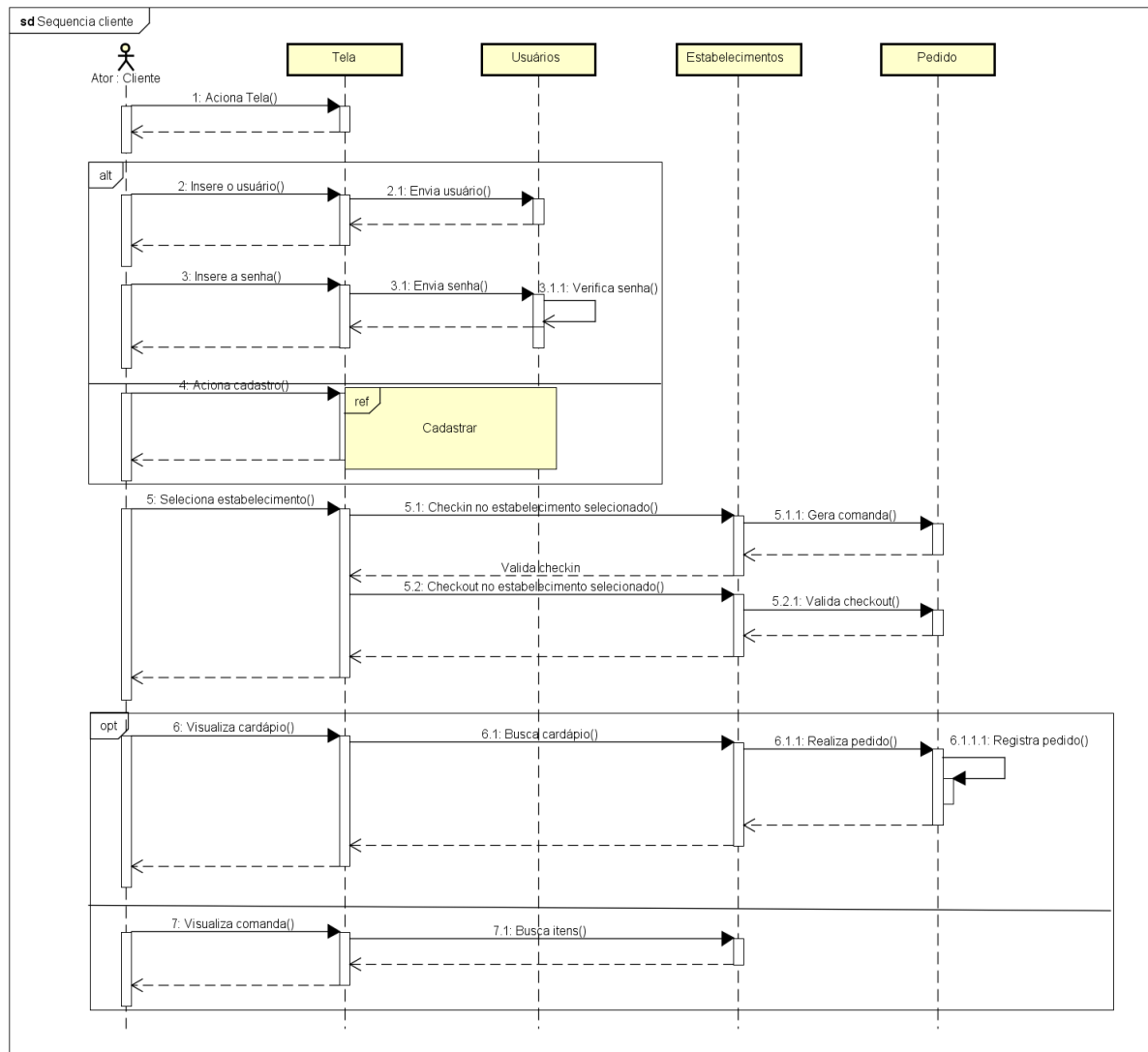
APÊNDICE 3.3

Diagrama de Classes – Modelo de Negócios



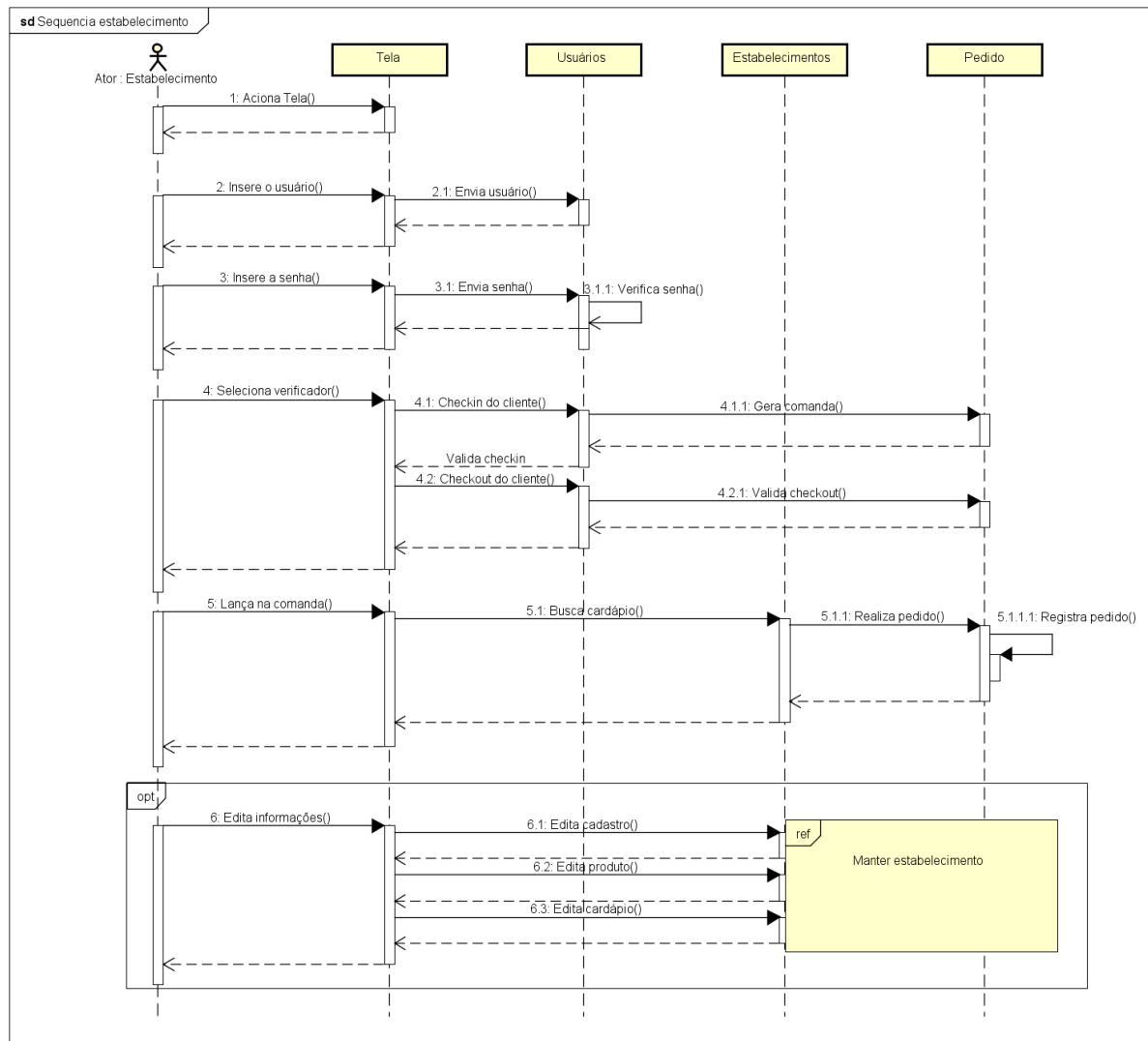
APÊNDICE 4.1

Diagrama de Sequência – Cliente



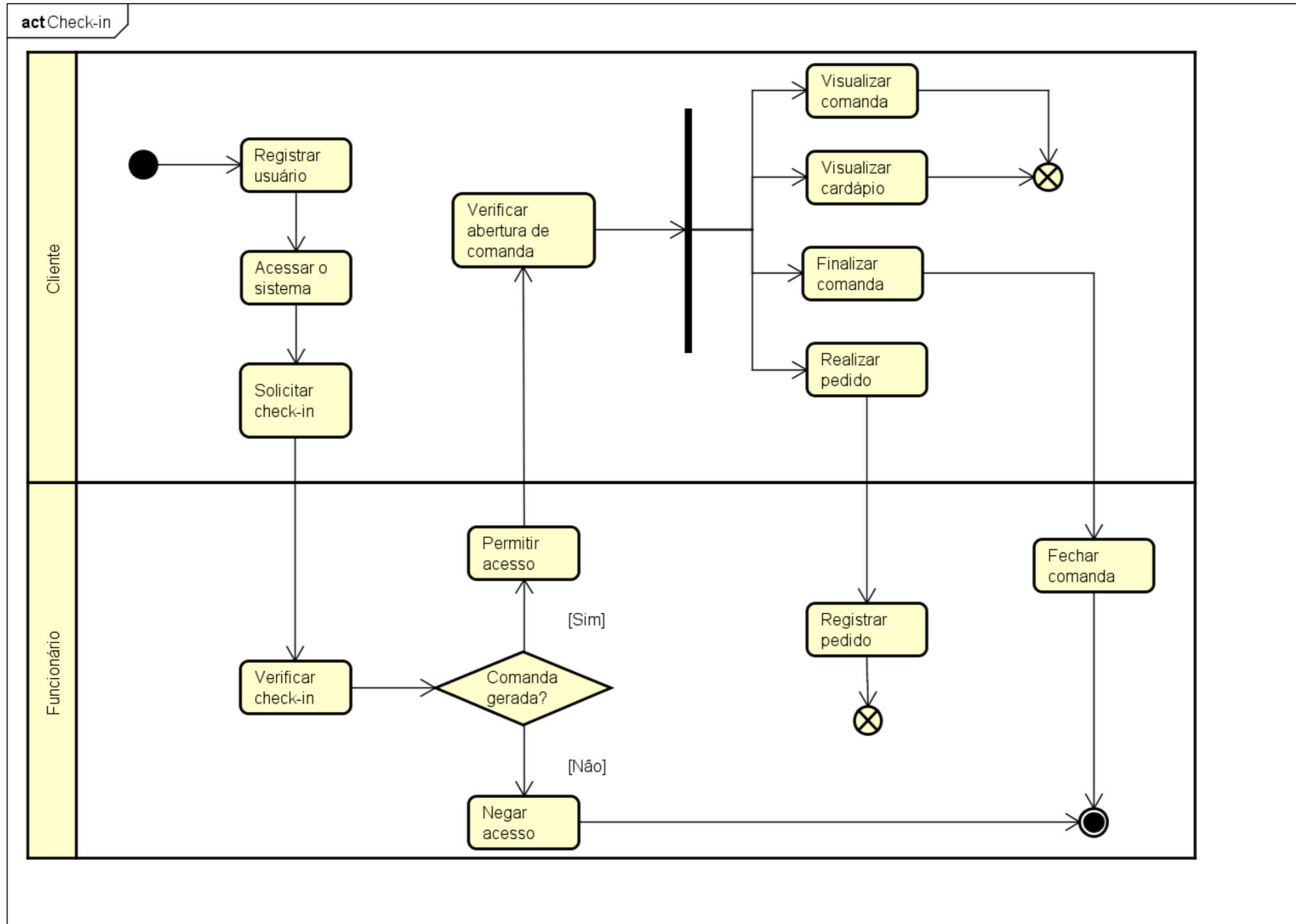
APÊNDICE 4.2

Diagrama de Sequência - Estabelecimento



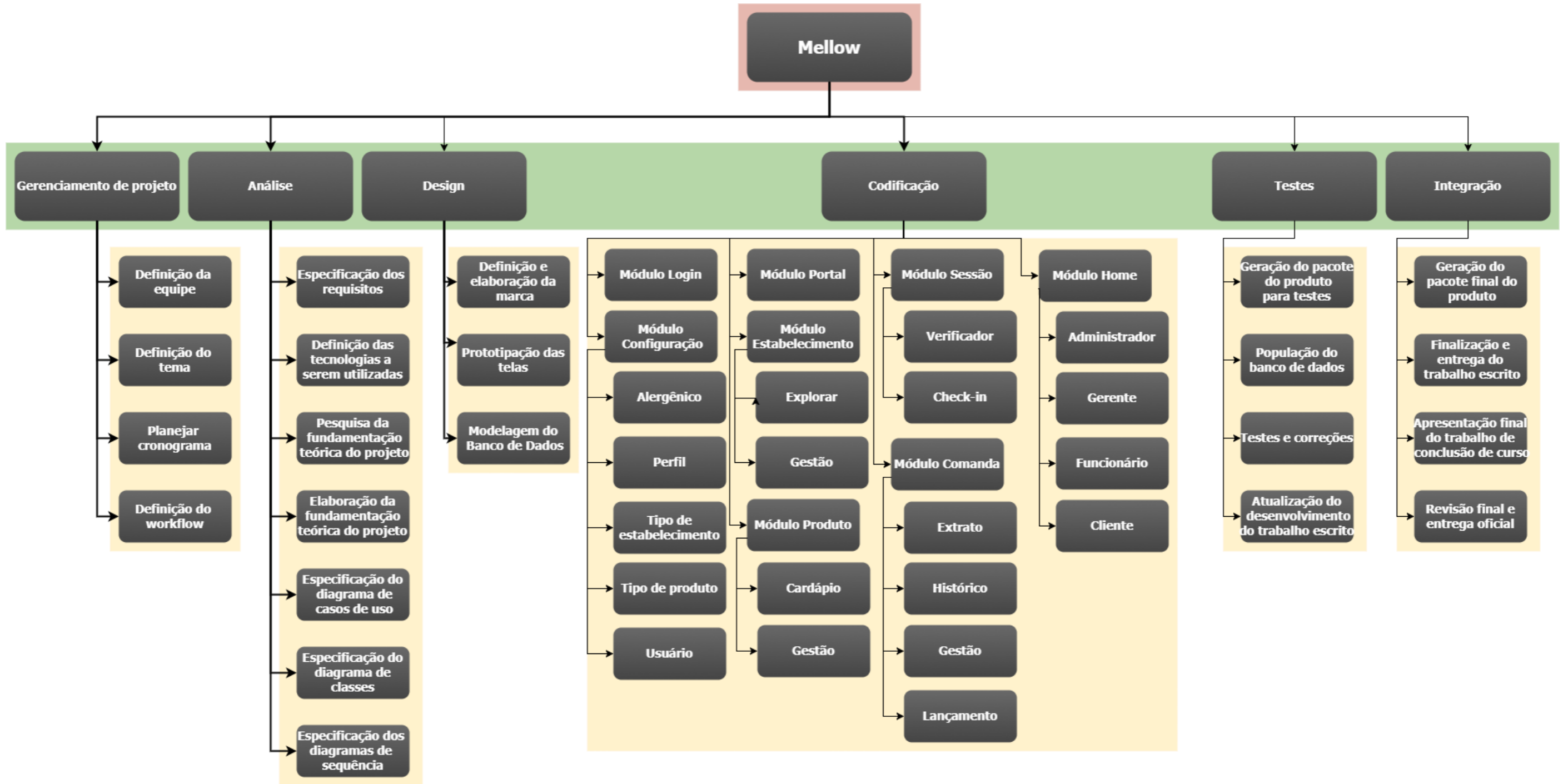
APÊNDICE 5

Diagrama de Atividades



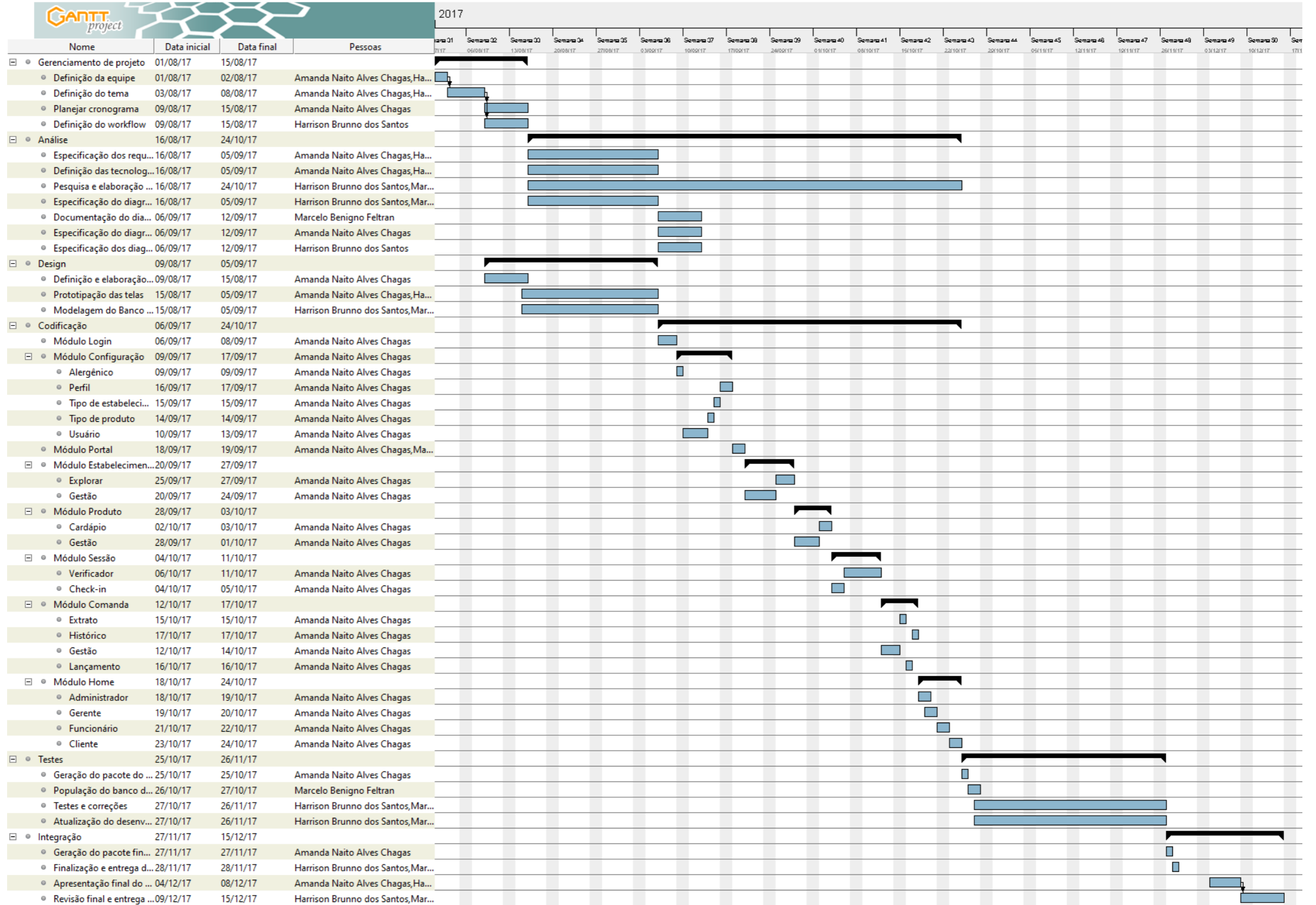
APÊNDICE 5

WBS



APÊNDICE 6

Gráfico Gantt



APÊNDICE 7

Plano de riscos

#	Condição	Data Limite	Consequência	Ação	Monitoramento	Probabilidade	Impacto	Classificação
1	Excesso de mudança nos requisitos	Sem data	Atraso na implementação das funcionalidades; Possível necessidade de treinamento da equipe;	Garantir desde o início do projeto que todos os requisitos sejam muito bem definidos;	Gerente de projetos, através de feedback reforçado com o orientador durante a o planejamento do projeto;	Moderada	Alto	6
2	Decisões externas causam mudanças forçadas no projeto	Sem data	Necessidade de reestruturação do projeto; Reiniciar o projeto desde a pesquisa;	Pesquisa referente à viabilidade de manter o projeto conforme definido no início;	Gerente de projetos, via feedbacks constantes do orientador, para garantir que o projeto não necessite de reestruturação;	Baixa	Muito Alto	6
3	Indisponibilidade de recursos humanos	20/10/2017	Atraso na entrega do projeto final;	Organizar o cronograma e garantir que todo o tempo livre seja utilizado no desenvolvimento do projeto;	Todos os envolvidos no desenvolvimento, com feedbacks e análises constantes do cronograma, a fim de garantir que o projeto não atrase	Alta	Alto	7
4	Cronograma não realista	15/11/2017	Atraso na entrega do projeto final;	Estudar os prazos e a velocidade de implementação de todas as fases do projeto;	Gerente de projetos, através de monitoramento constante das datas de entrega e andamento do projeto, para garantir que não hajam atrasos;	Moderada	Alto	6
5	Uso de novas tecnologias de hardware e software	30/09/2017	Necessidade de treinamento da equipe; Alta curva de aprendizado;	Estudo prévio das novas tecnologias a serem utilizadas; Cursos rápidos e completos sobre o que deve ser aprendido;	Gerente de projetos, verificando como anda o aprendizado de cada membro da equipe;	Moderada	Moderado	5