**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA “PAULA SOUZA”**

**FACULDADE DE TECNOLOGIA DE TAQUARITINGA**

**GERENCIA E PROJETOS**

**CHATBOT PARA LANCHONETES**

**Brenda da Silva Lorençon**

**Eder Ferreira de Matos**

**Taquaritinga – 2020**

**ESTE DOCUMENTO FOI ELABORADO DE ACORDO COM O REGULAMENTO GERAL DOS CURSOS DE GRADUAÇÃO, DE MARÇO DE 2009, DAS FACULDADES DE TECNOLOGIA DO ESTADO DE SÃO PAULO DO CENTRO “PAULA SOUZA”.**

**Sumário**

[1 – Introdução 6](#__RefHeading___Toc2886_3779691969)

[​ 2 – Desenvolvimento teórico 7](#__RefHeading___Toc2888_3779691969)

[​ 2.1 – Gerenciamento da integração 7](#__RefHeading___Toc2890_3779691969)

[2.1.1 - Objetivos do projeto 7](#__RefHeading___Toc2892_3779691969)

[2.1.2 - Situação atual e justificativa do projeto 7](#__RefHeading___Toc2894_3779691969)

[2.1.3 - Objetivos e critérios de sucesso do projeto 8](#__RefHeading___Toc2896_3779691969)

[2.1.4 - Estrutura analítica do projeto – Fases e principais entregas 8](#__RefHeading___Toc2898_3779691969)

[2.1.5 - Principais requisitos das principais entregas 8](#__RefHeading___Toc2900_3779691969)

[2.1.6 - Marcos 8](#__RefHeading___Toc2902_3779691969)

[2.1.7 - Partes interessadas do projeto 9](#__RefHeading___Toc2904_3779691969)

[2.1.8 – Restrições 9](#__RefHeading___Toc2906_3779691969)

[2.1.9 – Premissas 9](#__RefHeading___Toc2908_3779691969)

[2.1.10 – Riscos 10](#__RefHeading___Toc2910_3779691969)

[2.1.11 – Orçamento do projeto 10](#__RefHeading___Toc2912_3779691969)

[​ 2.2 – Gerenciamento do escopo 10](#__RefHeading___Toc2914_3779691969)

[2.2.1 – Abordagem de desenvolvimento 10](#__RefHeading___Toc2916_3779691969)

[2.2.2 – Levantamento de requisitos 10](#__RefHeading___Toc2918_3779691969)

[2.2.4 – EAP 11](#__RefHeading___Toc2920_3779691969)

[​ 2.3 – Gerenciamento do tempo 11](#__RefHeading___Toc2922_3779691969)

[2.3.1 – Rede do projeto 12](#__RefHeading___Toc2924_3779691969)

[2.3.2 – Tabela de precedência 12](#__RefHeading___Toc2926_3779691969)

[2.3.3 – Caminhos 13](#__RefHeading___Toc2928_3779691969)

[2.3.4 – Caminho crítico 13](#__RefHeading___Toc2930_3779691969)

[2.3.5 – Descrição das atividades 13](#__RefHeading___Toc2932_3779691969)

[​ 2.4 – Gerenciamento dos custos 14](#__RefHeading___Toc2934_3779691969)

[2.4.1 - Planejar o gerenciamento dos custos 14](#__RefHeading___Toc2936_3779691969)

[2.4.2 - Estimar os custos 14](#__RefHeading___Toc2938_3779691969)

[2.4.3 - Determinar o orçamento 15](#__RefHeading___Toc2940_3779691969)

[2.4.4 – Controlar custos 15](#__RefHeading___Toc2942_3779691969)

[​ 2.5 – Gerenciamento da qualidade 16](#__RefHeading___Toc2944_3779691969)

[2.5.1 - Diagramas de causa e efeito 16](#__RefHeading___Toc2946_3779691969)

[​ 2.5.1.1 - Demora no atendimento 16](#__RefHeading___Toc2948_3779691969)

[​ 2.5.1.2 - Transmissão de valores incorretos 17](#__RefHeading___Toc2950_3779691969)

[​ 2.5.1.3 - Anotação de pedidos errados 17](#__RefHeading___Toc2952_3779691969)

[​ 2.5.1.4 - Inversão de chats na resposta 18](#__RefHeading___Toc2954_3779691969)

[2.5.2 - Ciclo PDCA 19](#__RefHeading___Toc2956_3779691969)

[​ 2.5.2.1 – Planejamento 19](#__RefHeading___Toc2958_3779691969)

[​ 2.5.2.2 - Reuniões Administrativas 20](#__RefHeading___Toc2960_3779691969)

[​ 2.5.2.3 - Entrevista com usuário 21](#__RefHeading___Toc2962_3779691969)

[​ 2.5.2.4 - Definição das atividades 22](#__RefHeading___Toc2964_3779691969)

[​ 2.5.2.5 - Fluxo da Informações e aprovação 23](#__RefHeading___Toc2966_3779691969)

[​ 2.5.2.6 - Definição do tempo, custo, qualidade e risco 24](#__RefHeading___Toc2968_3779691969)

[​ 2.5.2.7 - Plano de qualidade 26](#__RefHeading___Toc2970_3779691969)

[​ 2.5.2.8 - Ferramentas e definição de detalhes 27](#__RefHeading___Toc2972_3779691969)

[​ 2.5.2.9 - Modelo das atividades 28](#__RefHeading___Toc2974_3779691969)

[​ 2.5.2.10 - Configuração da infraestrutura 29](#__RefHeading___Toc2976_3779691969)

[​ 2.5.2.11 – Back-end 30](#__RefHeading___Toc2978_3779691969)

[​ 2.5.2.12 - Integração com o WhatsApp 31](#__RefHeading___Toc2980_3779691969)

[​ 2.5.2.13 - Teste em geral 32](#__RefHeading___Toc2982_3779691969)

[​ 2.6 – Gerenciamento dos recursos 33](#__RefHeading___Toc2984_3779691969)

[2.6.1 - Planejamento 33](#__RefHeading___Toc2986_3779691969)

[2.6.2 - Recursos e atividades 33](#__RefHeading___Toc2988_3779691969)

[2.6.3 - Adquirir Recursos 33](#__RefHeading___Toc2990_3779691969)

[2.6.4 - Desenvolvimento da equipe 33](#__RefHeading___Toc2992_3779691969)

[2.6.5 - Gerenciamento da Equipe 34](#__RefHeading___Toc2994_3779691969)

[2.6.6 - Controle de Recursos 34](#__RefHeading___Toc2996_3779691969)

[2.6.7 – PMO 34](#__RefHeading___Toc2998_3779691969)

[​ 2.7 – Gerenciamento das comunicações 35](#__RefHeading___Toc3000_3779691969)

[2.7.1 – Matriz de comunicações 35](#__RefHeading___Toc3002_3779691969)

[​ 2.7.1.1 - Registros das partes interessadas 35](#__RefHeading___Toc3004_3779691969)

[​ 2.7.1.2 - Desenvolvimento da aplicação 36](#__RefHeading___Toc3006_3779691969)

[​ 2.7.1.3 - Registros das partes interessadas 36](#__RefHeading___Toc3008_3779691969)

[​ 2.7.1.4 – Cronograma 37](#__RefHeading___Toc3010_3779691969)

[​ 2.7.1.5 - Documentação de riscos 38](#__RefHeading___Toc3012_3779691969)

[​ 2.7.1.6 - Plano do projeto 39](#__RefHeading___Toc3014_3779691969)

[​ 2.7.1.7 - Registro de bugs 39](#__RefHeading___Toc3016_3779691969)

[​ 2.8 – Gerenciamento dos riscos 41](#__RefHeading___Toc3018_3779691969)

[​ 2.8.1 - Riscos do projeto 41](#__RefHeading___Toc3020_3779691969)

[2.8.2 – Matriz de riscos 42](#__RefHeading___Toc3022_3779691969)

[​ 2.9 – Gerenciamento das aquisições e contratações 43](#__RefHeading___Toc3024_3779691969)

[​ 3 – Desenvolvimento prático 43](#__RefHeading___Toc3026_3779691969)

[3.1 – Metodologia 44](#__RefHeading___Toc3028_3779691969)

[3.1.1 – Planejamento da Sprint 44](#__RefHeading___Toc3030_3779691969)

[3.1.2 – Reuniões diárias - Dailys 44](#__RefHeading___Toc3032_3779691969)

[3.2 – Linguagem de programação 45](#__RefHeading___Toc3034_3779691969)

[3.2.1 – Puppeteer 45](#__RefHeading___Toc3036_3779691969)

[3.2.2 – Socket 45](#__RefHeading___Toc3038_3779691969)

[3.2.3 – WAPI 46](#__RefHeading___Toc3040_3779691969)

[3.3 – Licenças 46](#__RefHeading___Toc3042_3779691969)

[3.4 – Funcionalidades 48](#__RefHeading___Toc3044_3779691969)

[3.4.1 – Realizar pedido 48](#__RefHeading___Toc3046_3779691969)

[​ 3.4.1.1 – Pedindo um lanche 49](#__RefHeading___Toc3048_3779691969)

[​ 3.4.1.2 – Pedindo uma bebida 50](#__RefHeading___Toc3050_3779691969)

[​ 3.4.1.3 – Pedindo uma porção 51](#__RefHeading___Toc3052_3779691969)

[3.4.2 – Obter informações sobre o estabelecimento 52](#__RefHeading___Toc3054_3779691969)

[3.4.3 – Deixar elogio/critica 53](#__RefHeading___Toc3056_3779691969)

[3.4.4 – Finalizar atendimento 53](#__RefHeading___Toc3058_3779691969)

[​ 3.4.4.1 – Finalização de atendimento de um pedido 53](#__RefHeading___Toc3060_3779691969)

[3.4.4.2 – Finalização de atendimento sem pedido 55](#__RefHeading___Toc3062_3779691969)

[​ 3.4.4.3 – Finalização de atendimento por falta de interação 56](#__RefHeading___Toc3064_3779691969)

[​ 4 – Considerações finais 57](#__RefHeading___Toc3066_3779691969)

[​ Referências 58](#__RefHeading___Toc3068_3779691969)

**Lista de Figuras**

[Tabela 1 - Termo de Abertura 5](#_Toc54726884)

[Tabela 2 - Controle de Versões 5](#_Toc54726885)

[Tabela 3 – Marcos 7](#_Toc54726886)

Tabela 4 - Salário 32

# 1 – Introdução

A tecnologia é algo que vem revolucionando o mundo, e de forma abrangente em todos os meios sociais.

O nosso projeto tem como foco o desenvolvimento de um chatbot, como objetivo proporcionar maior facilidade e agilidade com pedidos de uma empresa (lanchonetes, restaurantes, fastfood etc.)

Onde se torne algo mais rápido e prático tanto para cliente quanto para empresa, podendo fazer bom uso de mensagens automáticas e bem variadas de acordo com a necessidade do cliente.

Em geral o chatbot terá muitas condições e será versátil para qualquer modificação. De acordo com a variedades para se fazer um pedido em grande massa, foi analisado a necessidade no mercado de automatizar e agilizar este processo. É a área de tecnologia que mostra toda a evolução do mundo, de forma que resolva muitos problemas durante a vida no mundo e proporciona muito recursos.

Toda a conclusão do projeto está sendo feita conforme o prazo estimado.

# 2 – Desenvolvimento teórico

## 2.1 – Gerenciamento da integração

|  |
| --- |
| **Termo de abertura do projeto** |
| Projeto: Chatbot para lanchonetes |

Tabela 1 - Termo de Abertura

**Controle de versões**

| **Versão** | **Data** | **Autores** | **Notas de revisão** |
| --- | --- | --- | --- |
| 1 | 08/09/2020 | Brenda Lorençon  Eder Matos | Sem revisões |

Tabela 2 - Controle de Versões

### 2.1.1 - Objetivos do projeto

Desenvolver e implantar um chatbot para uma lanchonete, para atender os clientes de forma mais rápida e poder ter um acompanhamento dos pedidos em determinados espaço de tempo. Esse chatbot deve permitir a realização de pedidos, e obtenção de informações relacionadas à lanchonete.

### 2.1.2 - Situação atual e justificativa do projeto

Hoje na lanchonete Remy’s Burguer possuímos muitos pedidos realizados pelo WhatsApp. Em alguns dias existem diversos pedidos feitos ao mesmo tempo, o que ocasiona uma lentidão no atendimento, logo gerando também insatisfação por parte dos clientes.

### 2.1.3 - Objetivos e critérios de sucesso do projeto

O objetivo desse projeto é o desenvolvimento de um chatbot para WhatsApp onde possa ser realizado pedidos de lanches, e bebidas, além de obter informações sobre localização do estabelecimento.

Essa aplicação deve ter uma rápida ao cliente, para que ele não fique esperando e consequentemente venha desistir de realizar um pedido.

### 2.1.4 - Estrutura analítica do projeto – Fases e principais entregas

A estrutura analítica do projeto irá contemplar as seguintes áreas:

* Gerenciamento do projeto
* Requisitos do produto
* Design de detalhes
* Construção
* Integração e testes

### 2.1.5 - Principais requisitos das principais entregas

* Realização de pedido de lanches, com quantidade e observação
* Realização de pedido de bebidas com quantidades
* Finalização de atendimento
* Criação de opção para comentários
* Criação de opção de obter informações do estabelecimento

### 2.1.6 - Marcos

|  |  |  |
| --- | --- | --- |
| **Fase** | **Marcos** | **Previsão** |
| Iniciação do projeto | Projeto Aprovado | 08/09/2020 |
| Planejamento | Planejamento e gerenciamento do projeto | 13/09/2020 |
| Desenvolvimento | Fluxo de pedidos | 02/10/2020 |
| Desenvolvimento | Fluxo de Informações | 17/10/2020 |
| Desenvolvimento | Fluxo de comentários | 20/10/2020 |
| Desenvolvimento | Inclusão de testes | 24/10/2020 |
| Desenvolvimento | Integração com WhatsApp | 27/10/2020 |
| Monitoramento | Entrega Validada | 30/10/2020 |
| Encerramento | Projeto entregue e encerrado | 03/11/2020 |

Tabela 3 – Marcos

### 2.1.7 - Partes interessadas do projeto

* + Cliente do projeto – Lanchonete Remy’s Burger
    - Departamento financeiro
    - Departamento administrativo
    - Departamento de Marketing
  + Gerente do Projeto: Eder Ferreira de Matos
  + Equipe de desenvolvimento do projeto:
    - Eder Ferreira de Matos
    - Brenda da Silva Lorençon

### 2.1.8 – Restrições

* + Orçamento limitado
  + Disponibilidade de conexão com a internet
  + Conta no aplicativo WhatsApp

### 2.1.9 – Premissas

* + Definir a linguagem de programação
  + Definir arquitetura do software

### 2.1.10 – Riscos

* + Falta de comunicação com a internet
  + Incompatibilidade de plataformas
  + Conflito de conexões com o WhatsApp

### 2.1.11 – Orçamento do projeto

* + Custos esperados: 17 mil reais
  + Prazo estimado: 2 meses

## 2.2 – Gerenciamento do escopo

### 2.2.1 – Abordagem de desenvolvimento

Será usada um desenvolvimento ágil, no ciclo incremental são definidos intervalos de 2 a 4 semanas, para planejar, definir, criar, testar e liberar as entregas. A partir de cada ciclo há uma validação de entrega, na qual é caracterizado o ciclo iterativo.

A cada entrega há revisões e retrospectivas que garantem aprimoramento contínuo, seguindo necessidades de mercado e as exigências do cliente.

### 2.2.2 – Levantamento de requisitos

Para melhor levantamento dos requisitos foi usado JAD (Joint Application Design) uma técnica que tem como ponto principal a cooperação de toda a equipe envolvida com a solução a ser criada. São feitas reuniões com os clientes na qual são definidos os requisitos tendo o ponto de vista de todos os envolvidos, desde o usuário final ou seu representante, quanto analistas, arquitetos, diretores etc.

**Requisitos Funcionais:**

* Validar opções de respostas
* Utilização de sockets para comunicação
* Salvar opções e mensagens em banco mongodb.
* Utilizar Java para processamento das mensagens

**Requisitos não funcionais:**

* Resposta em até 5 segundos para o usuário
* Opções enumeradas
* Responder usuários de qualquer plataforma
* Finalizar atendimentos após um tempo de inatividade
* Se o contato estiver salvo na agenda, deve utilizar o nome salvo na agenda, caso contrário deve utilizar o nome que o usuário colocou como na criação da conta no WhatsApp

### 2.2.4 – EAP

Aqui está a Estrutura Analítica do Projeto com a representação e organização em fases, apontando todo o ciclo de vida do projeto.

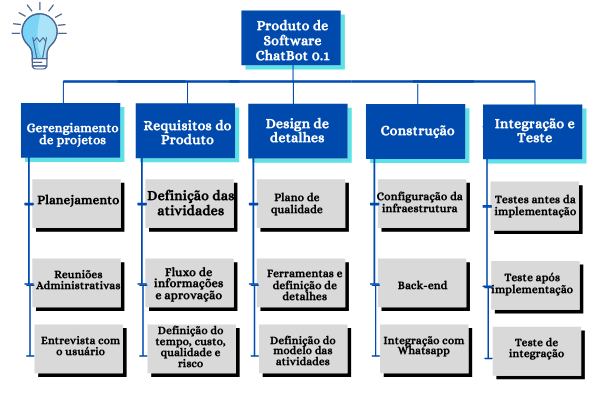


Figura 1 - EAP

## 2.3 – Gerenciamento do tempo

### 2.3.1 – Rede do projeto

Diagrama

Descrição gerada automaticamente

Figura 2 - Rede de Projeto

### 2.3.2 – Tabela de precedência

| **ID** | **Atividade** | **Duração** | **Atividades Precedentes** |
| --- | --- | --- | --- |
| 1 | Desenvolvimento da lógica do bot | 20 dias | - |
| 2 | Integração do algoritmo com o WhatsApp | 2 dias | 1 |
| 3 | Realização de testes de stress e monitoramento de performance | 4 dias | 2 |
| 4 | Criação da Infraestrutura da aplicação | 6 dias | - |
| 5 | Escrita de testes de software | 6 dias | - |

Figura 3 - Tabela de Precedência

### 2.3.3 – Caminhos

Os caminhos possíveis para esse projeto são:

* 1, 2, 3 – 26 dias
* 4 – 6 dias
* 5 – 6 dias

### 2.3.4 – Caminho crítico

O caminho crítico desse projeto é o 1, 2, 3, (Desenvolvimento da lógica da aplicação, Integração do algoritmo com o WhatsApp, Realização de testes de stress e monitoramento de performance, Finalização do projeto) totalizando 26 dias. Nesse caminho é realizado o desenvolvimento da regra de negócio da aplicação e realizado alguns testes para consolidação da performance da aplicação.

### 2.3.5 – Descrição das atividades

* Início do projeto
  + Inicialização do projeto e definição de quais caminhos vão ser seguidos, e em qual ordem
* Desenvolvimento da lógica do bot
  + Desenvolvimento de aplicação que responda mensagens recebidas de acordo no histórico de mensagens e no tempo, junto com informações armazenadas em um banco de dados
* Integração do algoritmo com o WhatsApp
  + Integração da aplicação desenvolvida com uma api de comunicação com o WhatsApp
* Realização de testes de stress e monitoramento de performance
  + Realização de testes para verificação da performance da aplicação.
* Documentação das atividades do bot
  + Documentação de atividades e códigos da aplicação
* Escrita de testes de software
  + Escrita de testes unitários e de integração para comprovação das atividades da aplicação, além de facilitar novas funcionalidades.
* Entrega do projeto
  + Entrega do projeto ao cliente para que o mesmo possa ser utilizado.

## 2.4 – Gerenciamento dos custos

### 2.4.1 - Planejar o gerenciamento dos custos

**Custo de equipe:** O projeto terá uma longa caminhada, com lima equipe bem específica para cada passo do projeto

**Mão de obra:** Será usados alguns softwares para desenvolvimento, são leves para evitar de ter que usar hardwares mais fortes e caros. Mas caso aconteça a quebra de um deles que são dois computadores, terá que ser feita a compra de um novo.

**Terceiros:** Não a necessidade de uma interação com terceiros.

**Equipamentos:** Alguns equipamentos podem quebrar durante o tempo, e a troca será necessária para mouse, teclados.

### 2.4.2 - Estimar os custos

Foi feita uma pesquisa de preços para troca de equipamentos que quebrem durante o processo. Também foi feito um levantamento para estimar os custos de acordo com toda a necessidade do projeto. E foi organizado em tópicos para cada tipo de levantamento:

* Mouse: Algumas marcas padrões chegam a ser 35,00.
* Teclado: Algumas marcas padrões chegam a ser 100,00.
* Infraestrutura AWS: 30,00 por mês
* Equipe de projeto: 3.800,00
* Reserva de emergência para riscos identificados: 500,00
* Reserva de emergência para riscos identificados: 300,00

### 2.4.3 - Determinar o orçamento

Com tudo isso foi organizada uma reunião para ver se os fatores e componentes seriam viáveis durante a trajetória do projeto, podendo definir toda a base necessária do máximo de gasto. Permanecendo o custo essencial para o desenvolvimento do chatbot.

Seguindo os preços do orçamento foi analisado um preço final de 5.000 reais.

### 2.4.4 – Controlar custos

Para um certo controle de custos, foi reforçado que seria somente para investimento de recurso máquina, e que seu limite maior será de 500 reais.

E em toda a trajetória haverá uma avaliação para analisar todos os componentes necessários.

## 2.5 – Gerenciamento da qualidade

### 2.5.1 - Diagramas de causa e efeito

#### 2.5.1.1 - Demora no atendimento

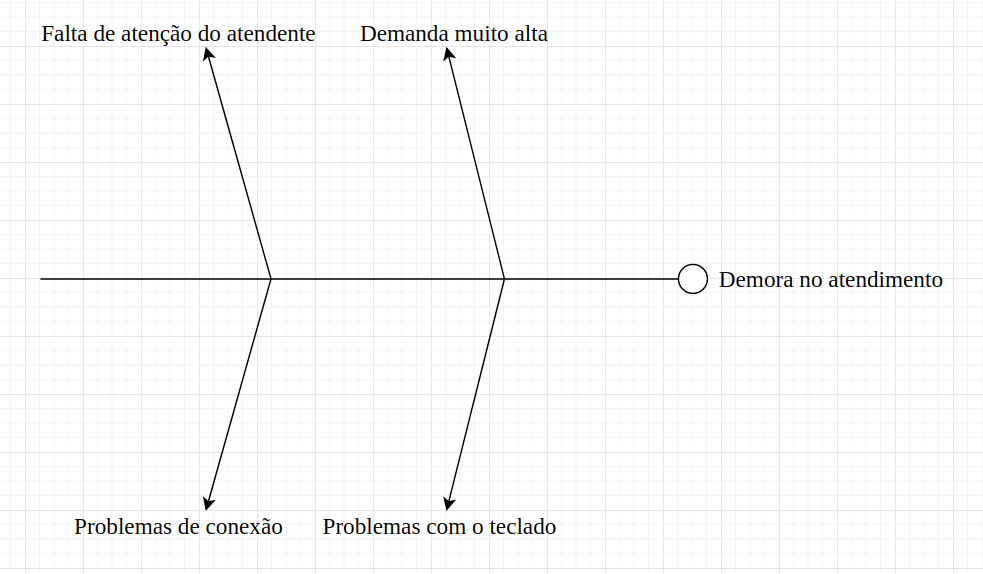


Figura 4 - Diagrama 01

A demora no atendimento é um dos problemas que podem fazer com que o estabelecimento tenha uma reputação ruim.

Praticamente todos os problemas identificados são causados por alguma interferência humana.

Com isso sabemos que o chatbot tem que ser rápido na tomada de decisão, e ser alimentado com informações precisas.

#### 2.5.1.2 - Transmissão de valores incorretos

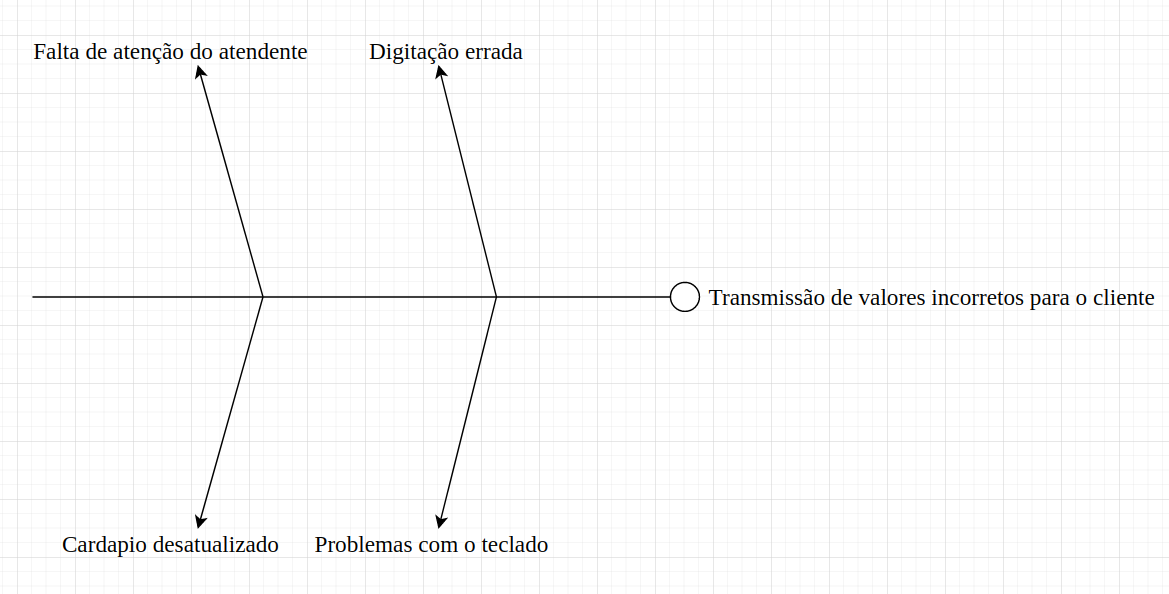


Figura 5 - Diagrama 02

Essa é uma falha que pode ocasionar frustração por parte do cliente, e até mesmo prejuízo para o estabelecimento. Novamente grande parte dos problemas são causados por falha humana.

Com isso identificamos que o chatbot deve sempre buscar as informações de valores de uma base de dados atualizada sempre que algum valor for alterado.

Por isso não é viável fixar valores no código do chatbot, pois os valores têm que ser dinâmicos.

#### 2.5.1.3 - Anotação de pedidos errados

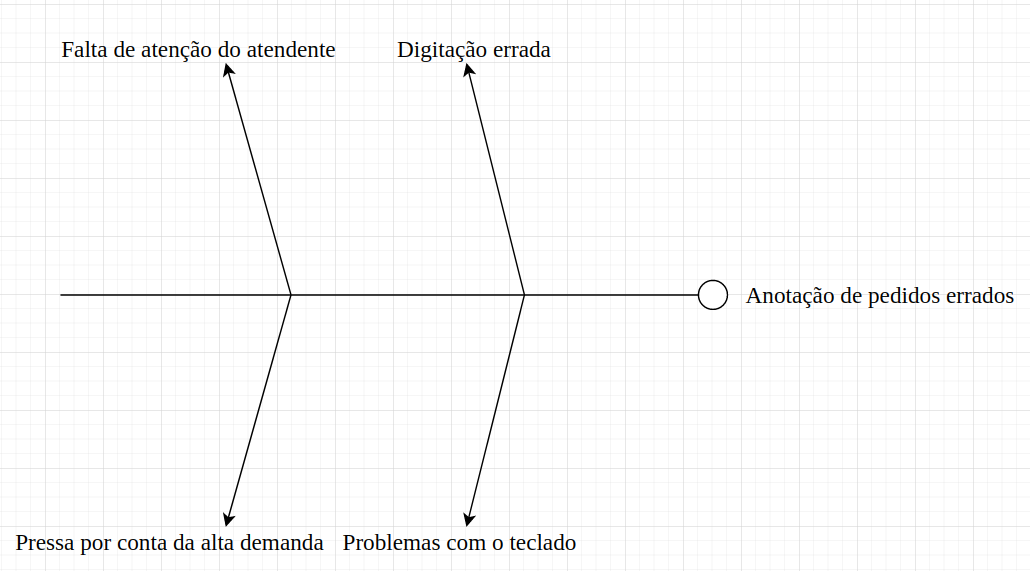


Figura 6 - Diagrama 03

A anotação de pedidos errados também é um grande problema que pode trazer muito prejuízo para o estabelecimento.

Com o chatbot só é anotado o que o cliente realmente pediu, se, possibilidades para meio termos.

#### 2.5.1.4 - Inversão de chats na resposta

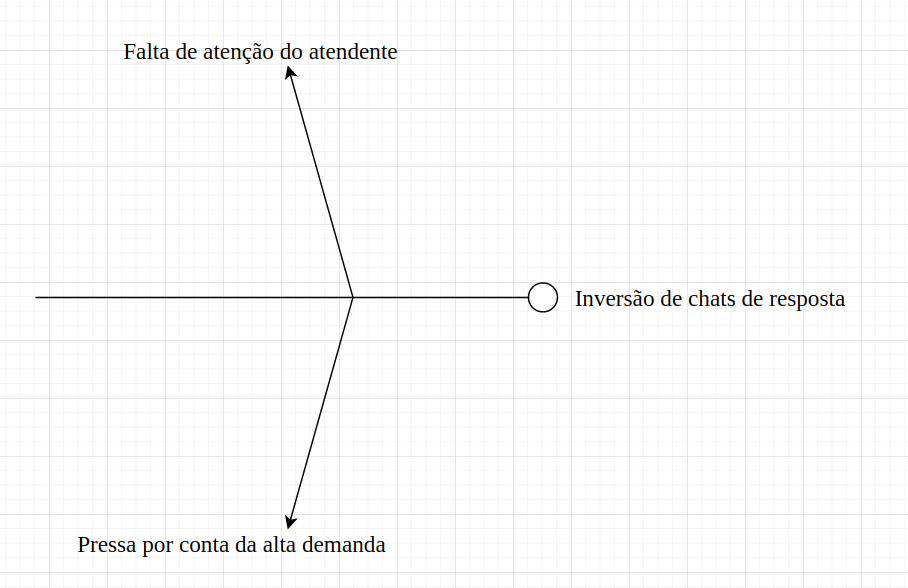


Figura 7 – Diagrama 04

### 2.5.2 - Ciclo PDCA

#### 2.5.2.1 – Planejamento

**Planejar**

* Qual é a ação planejada? (What?)

Realizar do projeto, ajustando as partes e fases do projeto, usando a metodologia e apresentação das informações necessárias de acordo com os requisitos do cliente.

* Por que é necessária esta ação? (Why?)

Para guiar o projeto e mostrar sua viabilidade após o planejamento de cada fase.

* Quem é o responsável? (Who?)

Os responsáveis são os líderes do projeto.

* Qual é o prazo? (When?)

O planejamento deve ser feito no máximo em 3 dias, pois o projeto é bem objetivo.

* Onde será desenvolvido? (Where?)

Será feito com ferramentas de organizações e em um escritório.

* Como será feito? (How?)

Será usado as ideias iniciais do projeto envolvendo cliente e desenvolvedores, após isso começará o desenvolvimento da documentação do planejamento. Os líderes do projeto se encarregarão de encaixar cada detalhe para apresentar depois ao cliente.

* Qual será o investimento? (How much?)

Algumas horas de trabalho diária, com prazo de 6 horas durante 3 dias.

**Executar**

Realizadas líderes do projeto com reuniões agendadas para organização do planejamento.

**Verificar**

Ter total verificação do planejamento para analisar se está de acordo com o que estava sendo pretendido pelos líderes do projeto e cliente

**Agir**

Durante o planejamento se surgir dúvidas, deverá ser conversado novamente entre os líderes para retirar as dúvidas e encaixar cada fase.

#### 2.5.2.2 - Reuniões Administrativas

**Planejar**

* Qual é a ação planejada? (What?)

As reuniões administrativas servem para manter os líderes em alta sintonia com o projeto.

* Por que é necessária esta ação? (Why?)

É necessária para não acontecer de os líderes se perderem com a comunicação e acabarem, desviando os objetivos do projeto.

* Quem é o responsável? (Who?)

Os responsáveis são os líderes do projeto.

* Qual é o prazo? (When?)

Essas reuniões devem ser feitas duas por semana até o final do projeto.

* Onde será desenvolvido? (Where?)

Será feito com ferramentas de organizações e em um escritório.

* Como será feito? (How?)

Será levantado os requisitos já passados antes pelo cliente, e então organizado pelos líderes usando a metodologia Scrum, por isso as reuniões em alta frequência.

* Qual será o investimento? (How much?)

Duas horas cada reunião para encaixe das fases e task.

**Executar**

Toda a execução será feita pelos líderes do projeto e com reuniões agendadas para organização do projeto.

**Verificar**

Ter total verificação das pautas de cada reunião durante as semanas.

**Agir**

Caso um dos líderes precise esclarecer pontos sobre ideias e ajustes deverá exigir uma nova reunião de urgência.

#### 2.5.2.3 - Entrevista com usuário

**Planejar**

* Qual é a ação planejada? (What?)

A ação planejada é ter algumas entrevistas com o cliente para poder estar corrigindo e definir corretamente os requisitos para a produção do projeto.

* Por que é necessária esta ação? (Why?)

É necessária para não acontecer de os líderes se perderem com o desenvolvimento do projeto sem atender os requisitos do cliente.

* Quem é o responsável? (Who?)

Os líderes do projeto e o cliente.

* Qual é o prazo? (When?)

Essas reuniões devem ser feitas durante o projeto pelo menos uma vez por semana.

* Onde será desenvolvido? (Where?)

Será feita no escritório com o cliente.

* Como será feito? (How?)

Será levantado os requisitos passados e atualizado pelo cliente, que ele ache necessário para ajustar no projeto durante o desenvolvimento e antes também.

* Qual será o investimento? (How much?)

Duas horas cada reunião para encaixe dos requisitos e ajustes.

**Executar**

Toda a execução será feita pelos líderes do projeto e com reuniões agendadas para organização do projeto.

**Verificar**

Ter total verificação das pautas de cada reunião durante as semanas.

**Agir**

Caso o cliente precise incluir algo no projeto deverá imediatamente entrar em contato com os líderes do projeto.

#### 2.5.2.4 - Definição das atividades

**Planejar**

* Qual é a ação planejada? (What?)

A ação planejada é para definir as atividades e passos do projeto e desta forma os líderes conseguem se ajustar para desenvolvimento do projeto.

* Por que é necessária esta ação? (Why?)

É necessária para não acontecer de os líderes se perderem com o desenvolvimento do projeto sem atender as atividades.

* Quem é o responsável? (Who?)

Os líderes do projeto.

* Qual é o prazo? (When?)

Essas atividades devem ser ajustadas em até duas semanas.

* Onde será desenvolvido? (Where?)

Usando algumas ferramentas de organização como trello e a metodologia scrum em nosso escritório.

* Como será feito? (How?)

Serão definidas as atividades separadas em processos e task para cada líder do projeto.

* Qual será o investimento? (How much?)

Terá como investimentos algumas horas diárias de acordo com cada atividade.

**Executar**

Toda a execução será feita pelos líderes do projeto e com comunicação a cada momento usando Discord.

**Verificar**

Ter total verificação das atividades e se elas estão organizadas corretamente.

**Agir**

Caso aconteça de alguma das atividades precisar de ajustes deverá ter uma comunicação diária entre os desenvolvedores.

#### 2.5.2.5 - Fluxo da Informações e aprovação

**Planejar**

* Qual é a ação planejada? (What?)

A ação planejada é ter esquemas (protótipos) com todas as informações estabelecidas para poder ser apresentado ao cliente e ter sua aprovação.

* Por que é necessária esta ação? (Why?)

É necessária para manter o cliente satisfeito e já informado sobre o que será desenvolvido para ele.

* Quem é o responsável? (Who?)

Os líderes do projeto.

* Qual é o prazo? (When?)

O prazo de uma semana após as atividades definidas.

* Onde será desenvolvido? (Where?)

Será feita no escritório com o cliente.

* Como será feito? (How?)

Será feita uma reunião para apresentar o esquema do projeto e assim aguardar aprovação do cliente durante a reunião.

* Qual será o investimento? (How much?)

Serão necessárias duas horas na reunião para aprovação e análise do cliente.

**Executar**

Toda a execução será feita pelos líderes do projeto após toda a criação do protótipo e com a reunião agendada já corretamente com o cliente.

**Verificar**

Analisar tudo o que será falado pelo cliente a verificar toda a aprovação do cliente levantando informações a mais necessárias ou ideias que poderão ser incluídas também.

**Agir**

Caso o cliente possuir questionamentos sobre o projeto, deverá ser ajustado de acordo e logo apresentar outro protótipo depois de 3 dias.

#### 2.5.2.6 - Definição do tempo, custo, qualidade e risco

**Planejar**

* Qual é a ação planejada? (What?)

A ação planejada é ter de definir o tempo total do projeto, custo, qualidade e riscos.

* Por que é necessária esta ação? (Why?)

É necessária para definir o que o projeto irá exigir e possuir.

* Quem é o responsável? (Who?)

Os líderes do projeto.

* Qual é o prazo? (When?)

O prazo de 4 semanas para toda a definição de tudo.

* Onde será desenvolvido? (Where?)

Será feita em uma ferramenta de texto e organização.

* Como será feito? (How?)

Será feita uma pesquisa para cada área de gerenciamento, assim definindo elas de acordo com cada parte exigida.

* Qual será o investimento? (How much?)

Será investido o tempo de 6 horas por dia para definição desses gerenciamentos.

**Executar**

Para a execução será necessária uma pesquisa antes para seguir um padrão correto para a documentação.

**Verificar**

Verificar se a parte de cada gerenciamento está correta e bem detalhada com seus sub tópicos.

**Agir**

Caso haja dúvidas durante a documentação e requisitos para cada uma deverá ser agendada uma reunião entre líderes.

#### 2.5.2.7 - Plano de qualidade

**Planejar**

* Qual é a ação planejada? (What?)

A ação planejada é identificar quais são as causas e efeitos podendo prevenir elas de acontecer principalmente agindo em cima da causa.

* Por que é necessária esta ação? (Why?)

É necessária para priorizar a qualidade do projeto desenvolvido.

* Quem é o responsável? (Who?)

Os líderes do projeto.

* Qual é o prazo? (When?)

O prazo de 1 semana para toda a definição de tudo.

* Onde será desenvolvido? (Where?)

Será feita em uma ferramenta de texto e organização.

* Como será feito? (How?)

Será feita uma pesquisa de causas e efeitos que podem acontecer e serem previnidas.

* Qual será o investimento? (How much?)

Será investido o tempo de 2 horas por dia durante a uma semana.

**Executar**

Para a execução será necessária uma pesquisa antes para seguir um padrão correto para o plano de qualidade.

**Verificar**

Verificar se realmente está correto as causas e efeitos do projeto.

**Agir**

Caso haja dúvidas durante a documentação deverá ser agendada uma reunião entre líderes.

#### 2.5.2.8 - Ferramentas e definição de detalhes

**Planejar**

* Qual é a ação planejada? (What?)

A ação planejada é identificar quais são as ferramentas necessárias para todo o desenvolvimento do projeto e a definição dos detalhes.

* Por que é necessária esta ação? (Why?)

É necessária para priorizar a organização da comunicação entre desenvolvedores e do projeto também.

* Quem é o responsável? (Who?)

Os líderes do projeto.

* Qual é o prazo? (When?)

O prazo é de 1 semana para organização e definição de ferramentas.

* Onde será desenvolvido? (Where?)

Será feita a reunião no escritório dos líderes.

* Como será feito? (How?)

Será feita uma pesquisa entre os desenvolvedores para melhor trabalharem durante o projeto.

* Qual será o investimento? (How much?)

Será investido o tempo de 2 horas por dia durante a uma semana.

**Executar**

Para a execução será necessária uma pesquisa pelos líderes para poderem definir quais as melhores ferramentas a ser usada.

**Verificar**

Verificar se as ferramentas atendem os requisitos para o desenvolvimento.

**Agir**

Caso haja problemas com alguma ferramenta deverá ser trocada imediatamente.

#### 2.5.2.9 - Modelo das atividades

**Planejar**

* Qual é a ação planejada? (What?)

A ação planejada é após a definição das ferramentas assim modelando as atividades totalmente de cada membro da equipe.

* Por que é necessária esta ação? (Why?)

É necessária para priorizar a organização do desenvolvimento do projeto.

* Quem é o responsável? (Who?)

Os líderes do projeto.

* Qual é o prazo? (When?)

O prazo é de 1 semana para a definição do modelo.

* Onde será desenvolvido? (Where?)

Será feita a reunião no escritório dos líderes.

* Como será feito? (How?)

Será feita de acordo com os requisitos do projeto assim definindo as funções de cada desenvolvedor e colocando em um modelo.

* Qual será o investimento? (How much?)

Será investido o tempo de uma hora de reunião durante uma semana.

**Executar**

Para a execução será necessária a definição das atividades para assim colocadas em um modelo.

**Verificar**

Verificar se o modelo das atividades está organizado corretamente.

**Agir**

Caso haja alguma hierarquia das atividades incorreta, fazer uma reunião para reajuste do modelo.

#### 2.5.2.10 - Configuração da infraestrutura

**Planejar**

* Qual é a ação planejada? (What?)

A ação planejada é criação da infraestrutura na aws.

* Por que é necessária esta ação? (Why?)

É necessária para a hospedagem da aplicação.

* Quem é o responsável? (Who?)

Os líderes do projeto.

* Qual é o prazo? (When?)

O prazo é de uma semana.

* Onde será desenvolvido? (Where?)

Será desenvolvida na Amazon Web Services

* Como será feito? (How?)

Será feita uma criação de instancia no EC2, criação do banco de dados no RDS e configuração de logs no CloudWatch.

* Qual será o investimento? (How much?)

Será investido o tempo de duas horas por dia.

**Executar**

Para a execução será é necessário o começo da documentação parcial do projeto e o desenvolvimento inicial da aplicação.

**Verificar**

Verificar se a construção e configuração estão corretas.

**Agir**

Caso haja alguma dúvida com as configurações da infraestrutura deverá ser feita uma reunião.

#### 2.5.2.11 – Back-end

**Planejar**

* Qual é a ação planejada? (What?)

A ação planejada é fazer toda a programação do projeto.

* Por que é necessária esta ação? (Why?)

É necessária para o funcionamento da aplicação.

* Quem é o responsável? (Who?)

Os líderes do projeto.

* Qual é o prazo? (When?)

O prazo é de 4 semanas.

* Onde será desenvolvido? (Where?)

Será feito todo o back-end em home office, e assim sendo acompanhado pelos líderes do projeto com reuniões remotas.

* Como será feito? (How?)

Os desenvolvedores iniciarão a implementação da aplicação de acordo com a documentação do projeto.

* Qual será o investimento? (How much?)

Será 5 horas por dia durante as semanas.

**Executar**

Para executar será necessário que a documentação esteja totalmente pronta e com todas as funcionalidades do projeto já definidas.

**Verificar**

Verificar se a aplicação está de acordo com o documento.

**Agir**

Caso haja alguma dúvida durante a programação do back-end deverá ser feita uma reunião imediatamente.

#### 2.5.2.12 - Integração com o WhatsApp

**Planejar**

* Qual é a ação planejada? (What?)

A ação planejada é fazer a integração da aplicação com o WhatsApp.

* Por que é necessária esta ação? (Why?)

É necessária para o funcionamento da aplicação.

* Quem é o responsável? (Who?)

Os líderes do projeto.

* Qual é o prazo? (When?)

O prazo é de 1 semana.

* Onde será desenvolvido? (Where?)

Será desenvolvido após o back-end geral do projeto.

* Como será feito? (How?)

Os desenvolvedores iniciarão a implementação da integração com o WhatsApp de forma que ele funcione de acordo com o objetivo do projeto.

* Qual será o investimento? (How much?)

Será 2 horas por dia durante a semana.

**Executar**

Para executar será necessário que o back-end esteja completo.

**Verificar**

Verificar se a aplicação está de acordo com o documento.

**Agir**

Caso haja alguma dúvida durante a integração do WhatsApp fazer uma reunião imediatamente.

#### 2.5.2.13 - Teste em geral

**Planejar**

* Qual é a ação planejada? (What?)

A ação planejada é fazer testes com a aplicação.

* Por que é necessária esta ação? (Why?)

É necessária para avaliar todo o funcionamento da aplicação.

* Quem é o responsável? (Who?)

Os líderes do projeto.

* Qual é o prazo? (When?)

O prazo é calculado de uma semana para testes.

* Onde será desenvolvido? (Where?)

Os testes serão feitos em home office também, acompanhando remotamente a aplicação.

* Como será feito? (How?)

Serão baseados na documentação de testes do projeto.

* Qual será o investimento? (How much?)

Será 2 horas por dia durante a semana.

**Executar**

Para executar os testes deverão ter em mãos os requisitos de cada função da aplicação.

**Verificar**

Verificar se a aplicação está de acordo com o documento e 100% testada.

**Agir**

Caso haja alguma dúvida durante os testes deverá ser discutido entre os desenvolvedores.

## 2.6 – Gerenciamento dos recursos

### 2.6.1 - Planejamento

**Competências:** É necessário conhecimentos em toda a parte de desenvolvimento de Software para toda a equipe.

**Responsabilidades:** Os desenvolvedores deverão fazer um chatbot, em um período seguindo as metas exigidas no escopo do projeto.  
**Funções:** Cada desenvolvedor terá sua função assim como também separação de suas funções durante o desenvolvimento do projeto!

Desenvolvedor 1: Irá fazer todo o esquema de relação das ações do software.

Desenvolvedor 2: Irá programar todo o esquema desenvolvido.

### 2.6.2 - Recursos e atividades

Alguns programas serão necessários para a programação do projeto, também a programas de organização assim como o Trello, Discord, Git, Github.

As atividades foram separadas em etapas, ideias do chatbot, organização delas, depois início da programação e assim ajustes e testes.

### 2.6.3 - Adquirir Recursos

Estes recursos são de fácil acesso assim podendo estar disponível a toda equipe e são gratuitos.

### 2.6.4 - Desenvolvimento da equipe

Assim como foi citado nos recursos, foi usado ferramentas para melhor comunicação entre a equipe, pois é a chave de ouro para um bom desenvolvimento do projeto e desempenho. A ferramenta Discord também está sendo usada para comunicação.

### 2.6.5 - Gerenciamento da Equipe

A cada ação que o membro termina é relatado no Trello ou no chat feito somente para este projeto. Caso alguém da equipe precise de ajuda, relatará no chat e dando o feedback de suas ações.

### 2.6.6 - Controle de Recursos

Caso aconteça de uns dos membros não conseguir utilizar os recursos apresentados, deverá entrar em contato com sua equipe para discutirem uma melhor forma de eficiência!

### 2.6.7 – PMO

Logo a seguir teremos a apresentação do Escritório de Projeto, que mostrará a hierarquia de toda função da equipe e que terão atuarão em cargos específicos.

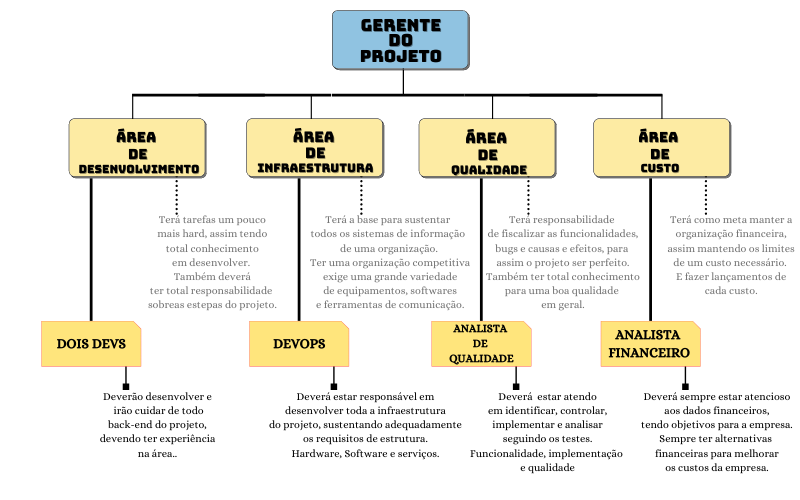


Figura 8 - PMO

A seguir está a tabela com salário de cada cargo, calculado o valor mensal por horas trabalhadas de cada membro.

|  |  |  |
| --- | --- | --- |
| **DEPARTAMENTO** | **CARGO** | **SALÁRIO** |
| Gerente do Projeto | Líder do Projeto | R$ 5.000.00 |
| Área de Desenvolvimento | Desenvolvedor | R$ 4.000.00 |
| Área de Infraestrutura | Devops | R$ 3.500.00 |
| Área de Qualidade | Analista de Qualidade | R$ 2.500.00 |
| Área de Custo | Analista Financeiro | R$ 2.000.00 |
| **Total Mensal** | **17.000.00** | |

Tabela 4 - Salário

## 2.7 – Gerenciamento das comunicações

### 2.7.1 – Matriz de comunicações

#### 2.7.1.1 - Registros das partes interessadas

* Qual informação? (What)
  + Registros das partes interessadas
* Qual propósito
  + Identificar partes interessadas e definir estratégias
* Quem é o responsável?
  + Gerente do projeto
* Quem precisa da informação? (Who)
  + Equipe de gerenciamento do projeto
* Quando? (When)
  + Início do projeto, com atualizações diárias
* Onde? (Where)
  + Discord e Trello
* Como?
  + Documento no formato pdf utilizando template pré-definido
* Procedimento?
  + Documento estratégico para a gestão do projeto e deve ser de conhecimento do gerente de projeto e pessoas de sua confiança. Deve ser reavaliado sempre, principalmente, quando aparecerem novos conflitos ou novos sinais de resistência do projeto.

#### 2.7.1.2 - Desenvolvimento da aplicação

* Qual informação? (What)
  + Desenvolvimento da aplicação
* Qual propósito
  + Determinar ordem das atividades
* Quem é o responsável?
  + Equipe do projeto
* Quem precisa da informação? (Who)
  + Equipe do projeto
* Quando? (When)
  + Início do projeto, com atualizações diárias
* Onde? (Where)
  + Discord e Trello, Github
* Como?
  + Tarefas no trello e mensagens informando atualizações.
* Procedimento?
  + No trello deve ser criada uma tarefa onde o título deve ser curto e direto ao ponto. Na descrição da tarefa deve ser levantado o porquê da tarefa, quem deverá realizar e uma data final de conclusão.

#### 2.7.1.3 - Registros das partes interessadas

* Qual informação? (What)
  + Registros das partes interessadas
* Qual propósito
  + Identificar partes interessadas e definir estratégias
* Quem é o responsável?
  + Gerente do projeto
* Quem precisa da informação? (Who)
  + Equipe de gerenciamento do projeto
* Quando? (When)
  + Início do projeto, com atualizações diárias
* Onde? (Where)
  + Discord e Trello
* Como?
  + Documento no formato pdf utilizando template pré-definido
* Procedimento?
  + Documento estratégico para a gestão do projeto e deve ser de conhecimento do gerente de projeto e pessoas de sua confiança. Deve ser reavaliado sempre, principalmente, quando aparecerem novos conflitos ou novos sinais de resistência do projeto.

#### 2.7.1.4 – Cronograma

* Qual informação? (What)
  + Cronograma
* Qual propósito
  + Definir datas limites e monitorar atividades
* Quem é o responsável?
  + Gerente do projeto
* Quem precisa da informação? (Who)
  + Equipe do projeto
* Quando? (When)
  + Início do projeto, com atualizações diárias
* Onde? (Where)
  + Discord e Trello
* Como?
  + Documento no formato pdf utilizando template pré-definido
* Procedimento?
  + O gerente de projeto deve criar o cronograma decompondo cada pacote de trabalho criado na EAP gerada na declaração de escopo do projeto e distribuir para toda equipe do projeto. O cronograma deve ser aprovado pelas principais partes interessadas e os responsáveis pelas atividades devem concordar com a duração das mesmas e estarem comprometidos.

#### 2.7.1.5 - Documentação de riscos

* Qual informação? (What)
  + Documentação de riscos
* Qual propósito
  + Identificar riscos e ações
* Quem é o responsável?
  + Gerente do projeto
* Quem precisa da informação? (Who)
  + Equipe de gerenciamento do projeto
* Quando? (When)
  + Início do projeto, com atualizações diárias
* Onde? (Where)
  + Discord
* Como?
  + Documento no formato pdf utilizando template pré-definido
* Procedimento?
  + O gerente de projeto deve identificar os riscos e desenvolver opções e ações para aumentar as oportunidades e reduzir as ameaças aos objetivos do projeto. Durante a execução do projeto, o GP deve executar o plano de resposta aos riscos como previsto, identificar novos riscos conforme mudanças no ambiente, além de apresentar um sumário dos riscos nas reuniões de status.

#### 2.7.1.6 - Plano do projeto

* Qual informação? (What)
  + Plano do projeto
* Qual propósito
  + Conduzir equipe do projeto em como agir
* Quem é o responsável?
  + Gerente do projeto
* Quem precisa da informação? (Who)
  + Equipe do projeto
* Quando? (When)
  + Início do projeto, com atualizações diárias
* Onde? (Where)
  + Discord,Trello
* Como?
  + Documento no formato pdf utilizando template pré-definido
* Procedimento?
  + O gerente de projeto deve definir, integrar e coordenar todos as seções ou planos auxiliares em um plano de projeto. O plano de projeto é a principal fonte de informações de como o projeto será planejado, executado, controlado e encerrado. Ele deve ser aprovado pelas principais partes interessadas do projeto e distribuído conforme definido no próprio plano. As linhas de base de prazo, custo e escopo devem ser salvas após a aprovação do plano do projeto.

#### 2.7.1.7 - Registro de bugs

* Qual informação? (What)
  + Registro de bugs
* Qual propósito
  + Registrar problemas encontrados e possíveis soluções
* Quem é o responsável?
  + Gerente do projeto
* Quem precisa da informação? (Who)
  + Equipe do projeto
* Quando? (When)
  + Início do projeto, com atualizações diárias
* Onde? (Where)
  + Discord, Trello
* Como?
  + Quadro à parte no trello.
* Procedimento?
  + O gerente de projeto receberá informações e organizará no novo, enviando para os envolvidos

2.7.1.8 - Finalização do projeto

* Qual informação? (What)
  + Finalização do projeto
* Qual propósito
  + Registrar os principais problemas e as soluções para que não aja demora num evento futuro
* Quem é o responsável?
  + Gerente do projeto
* Quem precisa da informação? (Who)
  + Equipe do projeto
* Quando? (When)
  + Final do projeto
* Onde? (Where)
  + Discord, Trello, Github
* Como?
  + Documento com lições aprendidas
* Procedimento?
  + O gerente de projeto usará o Issues Log para documentar principais problemas e soluções encontradas para evitar erros em projetos futuros. Enviará aos envolvidos e armazenará na pasta de projetos e na base histórica de lições aprendidas.

## 2.8 – Gerenciamento dos riscos

### 2.8.1 - Riscos do projeto

RISCO 01: ATRASO NA ENTREGA DO PROJETO

* + Probabilidade: 10%
  + Impacto: Muito alto
  + O atraso na entrega do projeto é um risco que qualquer projeto corre, e afeta diretamente o usuário final.
  + Podendo ter consequências tanto financeiras quanto judiciais.

RISCO 02: FALTA DE MÃO DE OBRA

* + Probabilidade: 10%
  + Impacto: Alto
  + A falta de mão de obra também é um risco que afetaria a entrega do projeto.
  + Por isso, a gestão do tempo é importante, levando em consideração quanto de mão de obra estará disponível

RISCO 03: FALTA DE INFRAESTRUTURA

* + Probabilidade: 10%
  + Impacto: Alto
  + A falta de infraestrutura também é um risco que iria afetar o funcionamento do sistema.
  + Porém como o sistema deve ser leve, até mesmo em equipamentos mais simples rodaria normalmente

RISCO 04: FALTA DE CONHECIMENTO PELO USUÁRIO

* + Probabilidade: 10%
  + Impacto: Médio
  + A falta de conhecimento por parte do usuário também é um risco que afetaria a experiencia do cliente que estaria realizando o pedido.
  + Porém como nesse software a única interação seria por meio de manutenção com uma interface amigável, a probabilidade pode ser minimizada.

RISCO 05: FALTA DE ATUALIZAÇÃO DE DADOS

* + Probabilidade: 50%
  + Impacto: Muito alto
  + A falta de atualização de dados é um risco que teria um impacto muito grande, principalmente financeiro, e pode ocorrer com maior facilidade por conta de necessitar de uma ação humana.

RISCO 06: FALHA DE COMUNICAÇÃO COM A INTERNET

* + Probabilidade: 30%
  + Impacto: Muito alto
  + Como todo o sistema necessita de comunicação com a internet, a falta dela ocasionaria um prejuízo enorme.

RISCO 07: ABERTURA DO WHATSAPP WEB EM OUTRO NAVEGADOR

* + Probabilidade: 50%
  + Impacto: Muito alto
  + De acordo com a arquitetura escolhida para criação do sistema, uma limitação é que o WhatsApp web não poderá ser aberto em nenhum outro navegador. por conta do chatbot utilizar o mesmo.
  + O impacto da abertura do WhatsApp web em outro navegador seria a pausa do funcionamento do chatbot. Sendo assim, haveria grandes prejuízos.

### 2.8.2 – Matriz de riscos



Figura 9 - Matriz de Risco

## 2.9 – Gerenciamento das aquisições e contratações

Essa área do conhecimento não se aplica a esse trabalho.

# 3 – Desenvolvimento prático

## 3.1 – Metodologia

Para o desenvolvimento do chatbot foi utilizado algumas cerimônias da metodologia ágil Scrum. Todas as cerimônias foram realizadas em chamadas e grupos organizados no Discord.

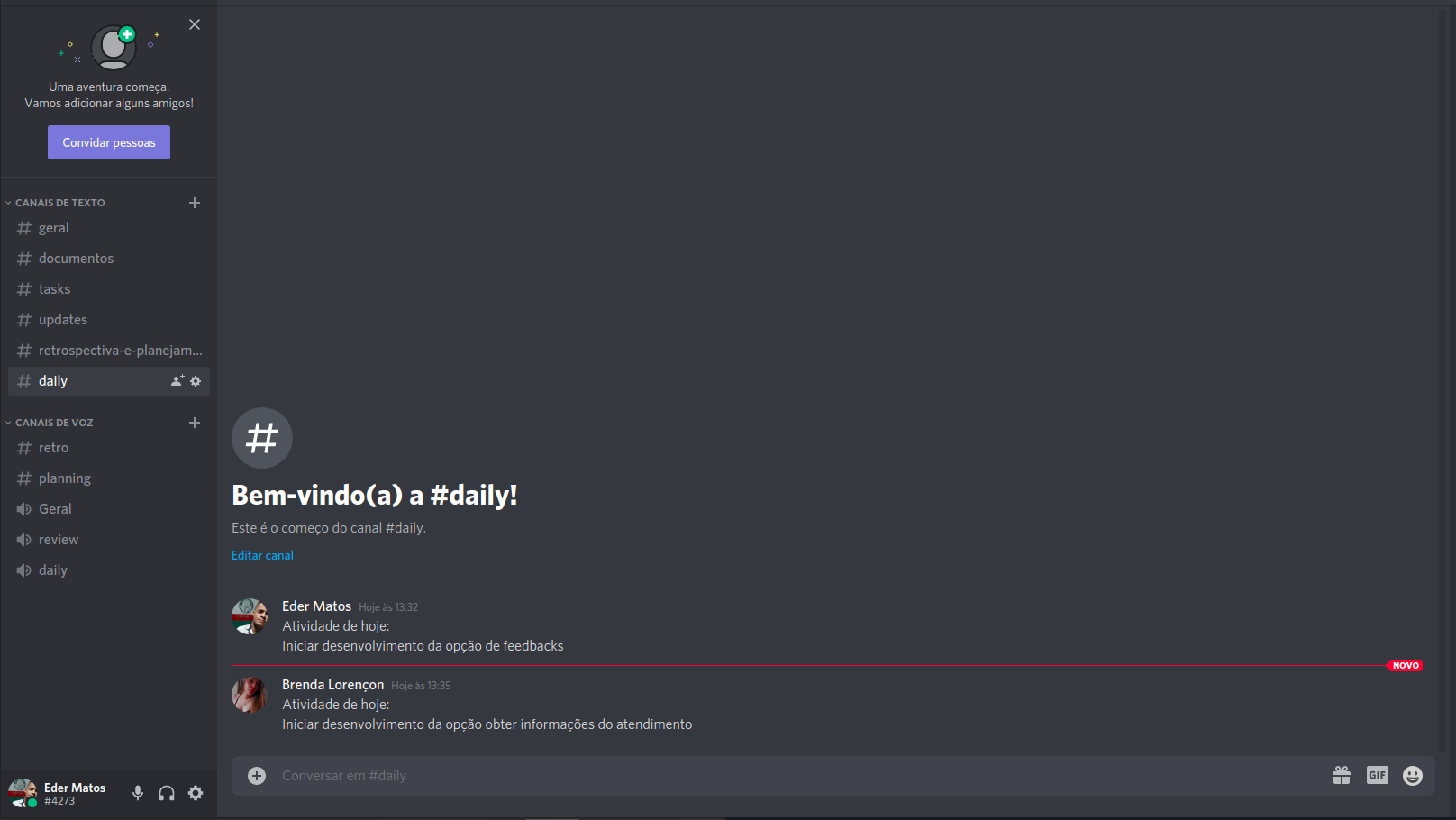


Figura 10 - Metodologia

### 3.1.1 – Planejamento da Sprint

Timebox: 1 hora

Integrantes: P.O, o Scrum Master e desenvolvedores

Tema: O que deve ser feito, e quebra das atividades e entendimento tecnicamente sobre o que deve ser feito

### 3.1.2 – Reuniões diárias - Dailys

Timebox: 15 minutos

Integrantes: P.O, o Scrum Master e desenvolvedores

Temas:

* O que fez no dia anterior
* O que está programado no dia atual
* Se tem algum impedimento

## 3.2 – Linguagem de programação

Para o desenvolvimento dessa aplicação, foi utilizado as linguagens NodeJS, e Java. Além disso, foram utilizado Socket para comunicação e uma biblioteca criada em NodeJS chamada Puppeter. A seguir temos mais detalhes.

### 3.2.1 – Puppeteer

Puppeteer é uma biblioteca node que fornece uma API de alto nível para controlar o Chrome ou Chromium, com ela é possível fazer a iniciação de um browser e o controle do mesmo via comandos escritos em JavaScript.

Essa biblioteca é bastante conhecida também para criação e automatização de testes de interface. Com ela é possível tirar prints de tela, exportar no formato pdf, o que auxilia na manutenção de sistemas.

### 3.2.2 – Socket

Um socket é um mecanismo de comunicação (dois sentidos) entre dois programas a funcionar (normalmente) numa rede.

Uma aplicação do servidor é executada numa determinada máquina e tem um socket ligado a uma porta específica dessa máquina. O servidor espera que um cliente faça um pedido de ligação através desse socket. O servidor ao aceitar a ligação cria um novo socket para uma porta diferente (e assim permite novas ligações). No lado do cliente um socket é criado e é usado para comunicar com o servidor (numa porta disponível na máquina cliente).

### 3.2.3 – WAPI

Wapi é um pacote utilizado para fornecer uma interface python para interagir com o WhatsAPP Web para enviar e receber mensagens do Whatsapp. Ele é baseado no aplicativo oficial do Whatsapp Web Browser e usa a automação do navegador Selenium para se comunicar com o Whatsapp Web. Ele fornece uma série de funções escritas em JavaScript que podemos executar no navegador.

Uma dessas funções é a “waitNewMessages” que é uma função que é chamada assim que o usuário recebe uma nova mensagem no WhatsApp, e na execução dessa função, temos acesso à mensagem, assim como o seu remetente. Uma outra função importante é a “sendMessage”, que possibilita enviar uma mensagem de texto para um usuário específico.

## 3.3 – Licenças

Nos termos da Lei n. 9.609/98, a utilização de softwares no Brasil está sujeita à prévia autorização dos proprietários, mediante a celebração de um contrato de licença. Esse licenciamento é, basicamente, o instrumento jurídico que garante aos licenciados o direito de uso/modificação e aos desenvolvedores e empresas a tutela autoral.

Geralmente, as licenças perduram por prazos pré-fixados pelos licenciantes e, por esse motivo, após a expiração, a renovação é obrigatória para o uso regular desses programas — caso contrário, a conduta é enquadrada como prática criminosa (popularmente conhecida como pirataria). (Conheça os tipos de licença de software e sua importância)

A biblioteca “puppeter” utiliza a licença “*Apache License 2.0*”.

Esta licença oferece bem mais flexibilidade aos usuários. Isso porque:

* O código-fonte não precisa ser necessariamente público quando a distribuição do software é feita.
* Modificações podem ser lançadas sob qualquer licença.
* Mudanças feitas no código-fonte precisam ser documentadas.
* Oferece a mesma proteção de uso de patente da GPLv3 que vimos acima.
* Proíbe explicitamente o uso de nomes de marcas registradas encontrados no projeto.

*(**Tudo o que você precisa saber sobre as licenças de projetos open source)*

O pacote “WAPI” utiliza a licença “*MIT License*”

Essa licença oferece pouquíssima proteção aos autores do software. Vamos dar uma olhada nas principais cláusulas:

* O código-fonte não precisa ser necessariamente público quando a distribuição do software é feita.
* Modificações podem ser lançadas sob qualquer licença.
* Mudanças feitas no código-fonte não precisam ser documentadas.
* Não tem nenhuma posição em relação ao uso de patentes.

*(**Tudo o que você precisa saber sobre as licenças de projetos open source)*

## 3.4 – Funcionalidades

Aqui temos as funcionalidades atendidas pelo chatbot. Elas podem ser acionadas por parte do usuário, mas também existe 1 funcionalidade que é de total autonomia do chatbot.

### 3.4.1 – Realizar pedido

A principal funcionalidade dessa aplicação é a realização de pedidos. Um pedido obrigatoriamente deve ter itens, que podem ser lanches, bebidas ou porções. Então nesse caso é possível fazer o pedido de lanches bebidas e porções, e após finalizar para saber qual o valor total do pedido.

Então no momento em que o cliente escolher realizar um pedido, deve ser listado para ele quais os tipos de pedido que ele pode fazer, no caso Lanches, Bebidas ou Porções.

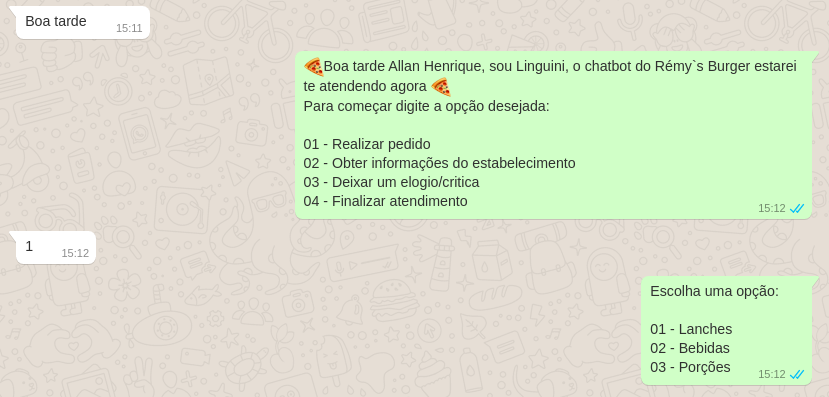


Figura 11 - Realizar Pedido

Nesse momento o cliente deve informar qual tipo de pedido que ele gostaria de fazer.

#### 3.4.1.1 – Pedindo um lanche

Se o cliente optar por pedir um lanche, o chatbot deve primeiramente listar quais as opções de lanche que estão disponíveis, seguido do valor do lanche.

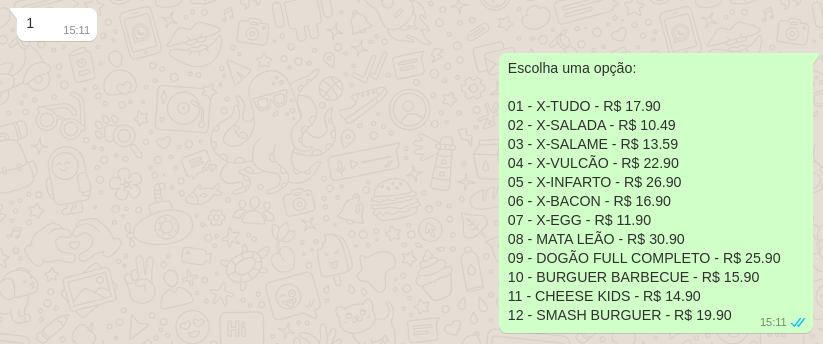


Figura 12 - Pedido de um lanche

Após isso será pedido que o usuário informe qual a quantidade que ele deseja daquele lanche.

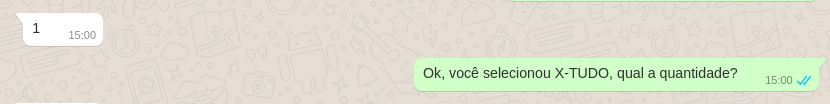


Figura 13 - Quantidade de Lanche

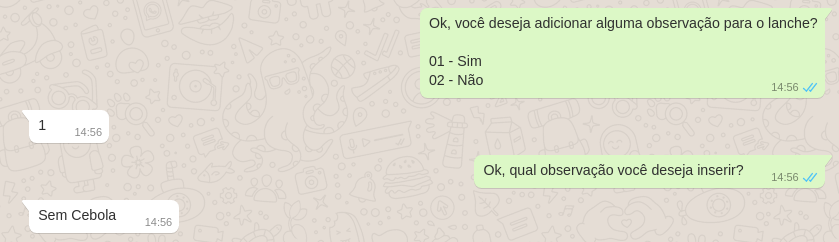
Após informar a quantidade desejada, é possível informar uma observação livre, onde o cliente pode pedir para remover algum ingrediente por exemplo.

Figura 14 - Quantidade

Em seguida é possível pedir mais um lanche, uma bebida, uma porção, ou finalizar o atendimento.



Figura 15 - Outros

#### 3.4.1.2 – Pedindo uma bebida

Se o cliente optar por pedir um lanche, o chatbot deve primeiramente listar quais as opções de lanche que estão disponíveis, seguido do valor do lanche.

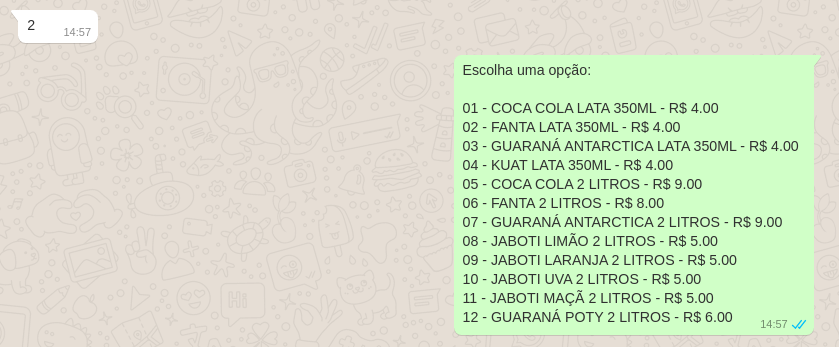


Figura 16 – Bebidas

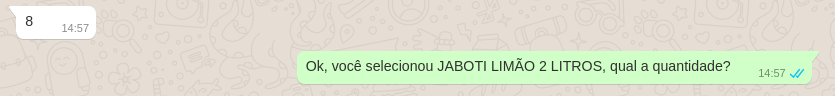
Após escolher o a bebida, é pedido para que o cliente informe qual a quantidade desejada.

Figura 17 - Quantidade de Bebida

Em seguida é possível pedir mais uma bebida, um lanche, uma porção, ou finalizar o atendimento.



Figura 18 - Outros

#### 3.4.1.3 – Pedindo uma porção

A realização de pedido de uma porção segue a mesma estrutura do pedido de uma bebida. Ou seja, é listado as porções, após o cliente escolher qual a porção deseja deve ser informada a quantidade e em seguida se o cliente deseja pedir outra porção, um lanche, uma bebida ou finalizar o atendimento.

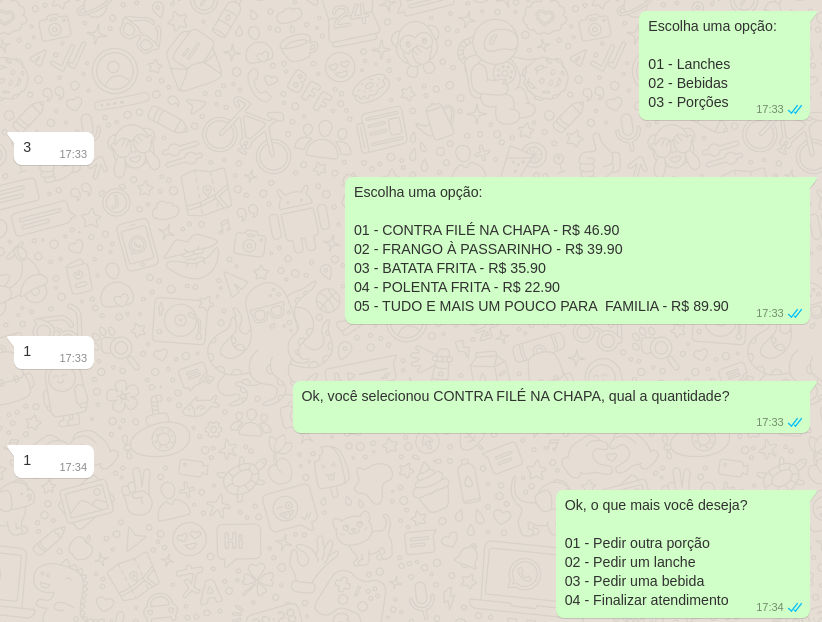


Figura 19 - Pedido de Porções

### 3.4.2 – Obter informações sobre o estabelecimento

Caso o cliente deseje obter informações do estabelecimento, deve ser mostrado à ele o endereço do estabelecimento, assim como os horários em que o estabelecimento oferece atendimento e números de telefone.

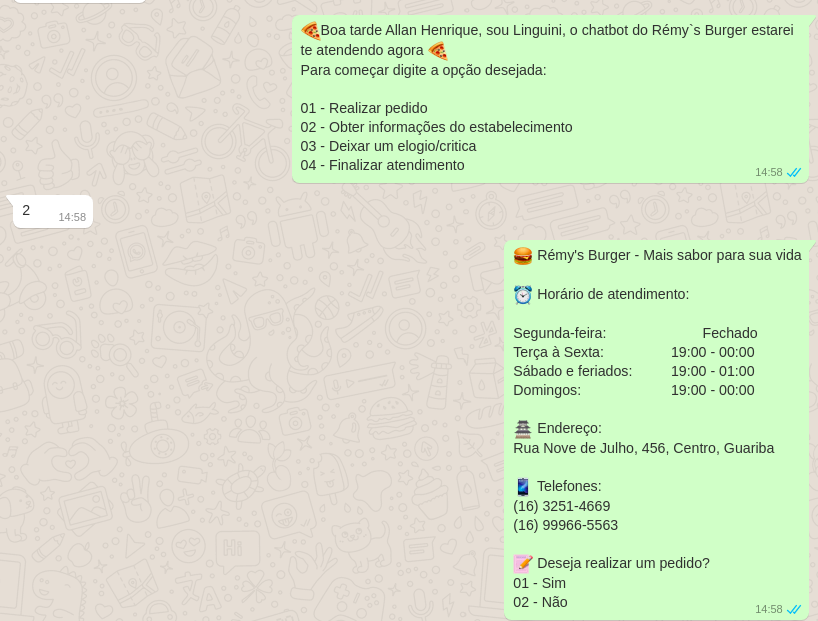


Figura 20 - Informações do Estabelecimento

Após isso é permitido que o cliente realize um pedido, ou finalize o atendimento.

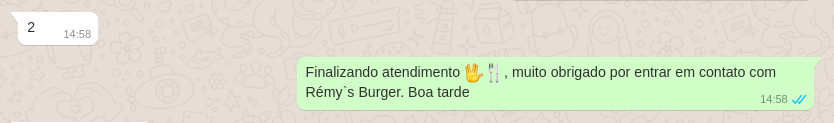


Figura 21 - Finalizando pedido

### 3.4.3 – Deixar elogio/critica

Uma funcionalidade importante para a aplicação é obter um feedback do usuário, seja ele positivo ou negativo.

Para deixar um feedback, o cliente deve selecionar a 3ª opção da primeira listagem de opções.

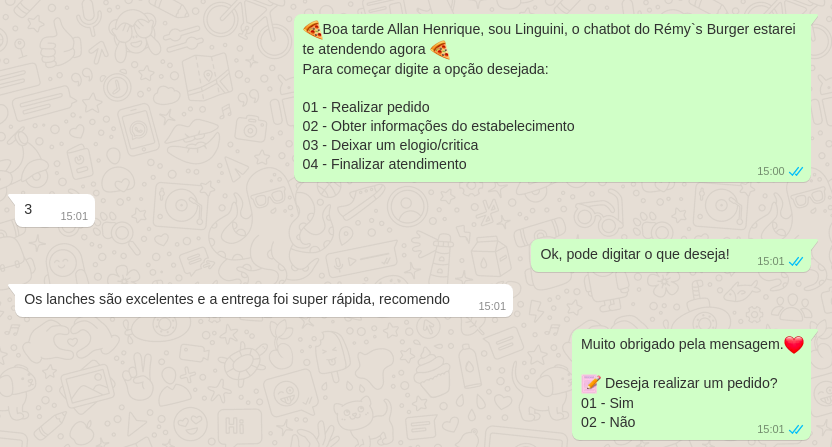


Figura 22 - Elogio/Crítica

Após isso é possível realizar um pedido ou finalizar o atendimento.

### 3.4.4 – Finalizar atendimento

Existem 3 formas diferentes de finalizar um atendimento. A primeira é realizando um pedido, a segunda é não realizando nenhum pedido, e a terceira uma finalização por falta de interação.

#### 3.4.4.1 – Finalização de atendimento de um pedido

Após selecionar todos os itens de um pedido, é apresentado para o cliente uma série de informações sobre o pedido, e por fim finalizado o atendimento.

Nessas informações temos:

* Horário de início do atendimento
* Horário de término do atendimento
* Itens do pedido
  + Nome do item
  + Quantidade
  + Valor unitário
  + Valor total
* Valor total do pedido
* Tempo de espera para a conclusão do pedido

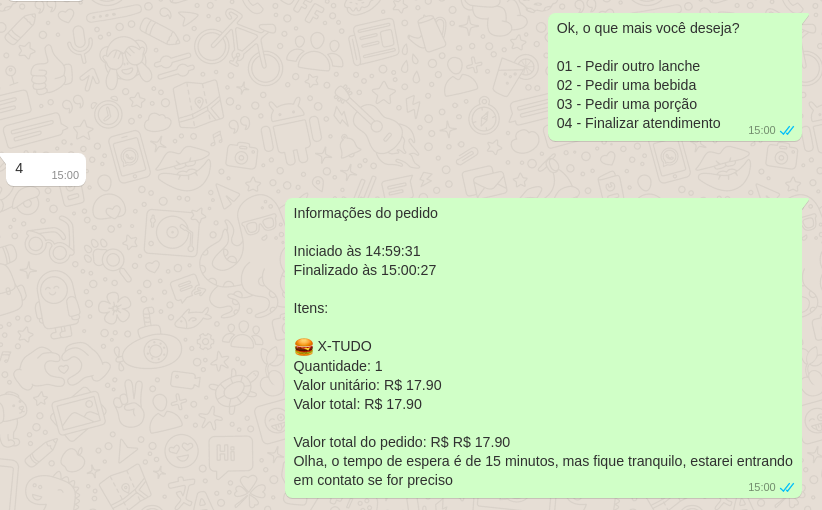


Figura 23 - Finalizando Pedido

#### 3.4.4.2 – Finalização de atendimento sem pedido

Também é possível finalizar um atendimento sem realizar um pedido, basta escolher a opção “Finalizar Atendimento” na primeira listagem de opções, ou escolher não fazer um pedido ao dar um feedback ou na opção de “Obter informações do estabelecimento”.

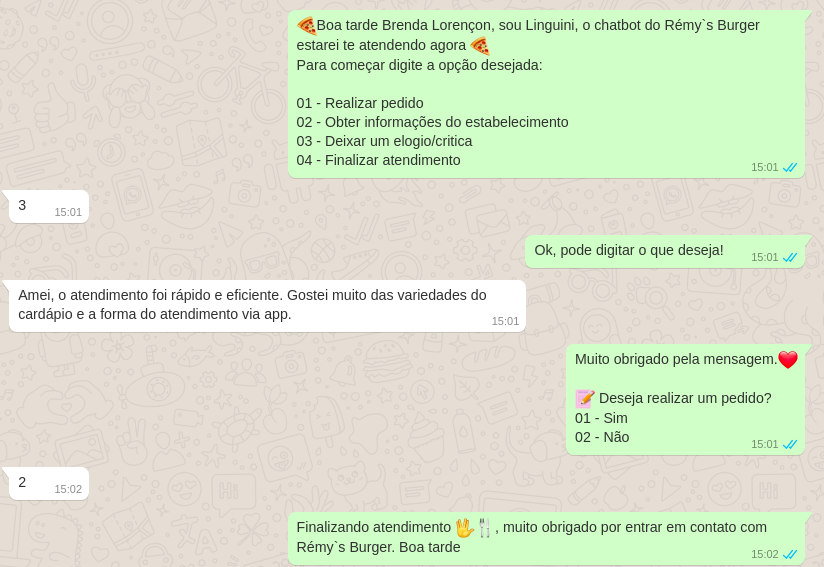


Figura 24 - Finalizando 01

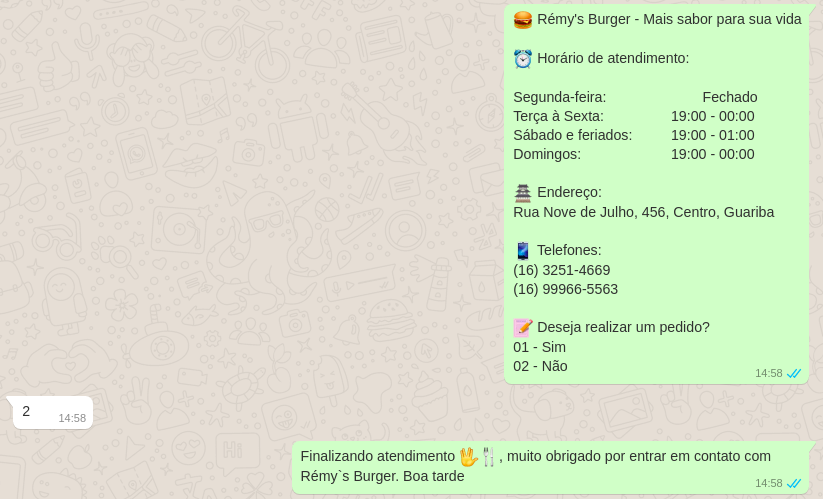


Figura 25 - Finalizando 02

#### 3.4.4.3 – Finalização de atendimento por falta de interação

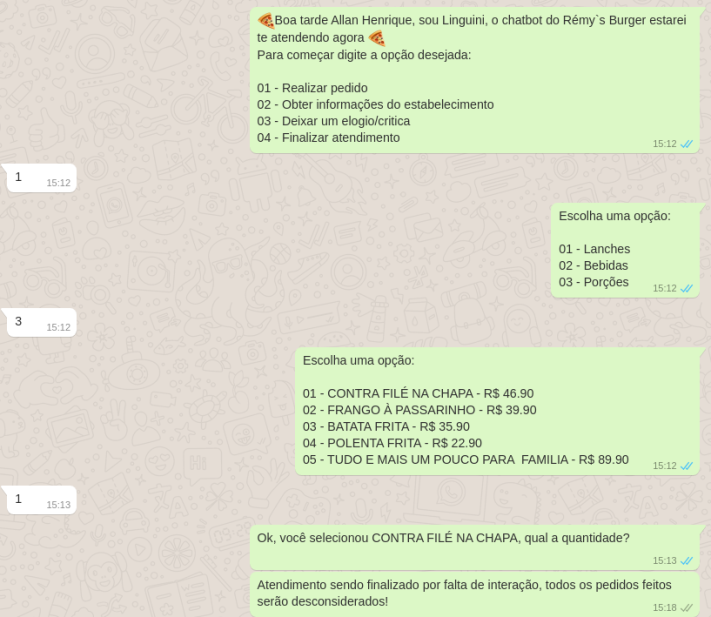


Figura 26 - Falta de interação

Essa funcionalidade tem como objetivo finalizar atendimentos que tiveram um grande espaço de tempo de inatividade. Supondo que o cliente mande uma mensagem, o chatbot responde, porém o cliente não retorno, ou seja não finaliza o atendimento. Nesse caso após 5 minutos de inatividade o chatbot irá finalizar o atendimento, e comunicar ao cliente que o atendimento foi finalizado por falta de interação.

# 4 – Considerações finais

O desenvolvimento desse projeto trouxe como finalidade uma comunicação mais rápida entre um cliente e um estabelecimento comercial do ramo alimentício, e o produto desenvolvido se mostrou rápido, de fácil entendimento e muito útil para diversos estabelecimentos. É um produto importante, que pode trazer muitas vantagens para quem utilizar.

Todas as áreas abordadas nesse documento foram de extrema importância para que o projeto fosse desenvolvido e permitido evoluir de forma que beneficie os interessados.

As tecnologias utilizadas permitem fácil escalonamento, permitindo uma melhor experiência do usuário e menor custos.

# Referências

PMI. A Guide to the Project Management Body of Knowledge. 5°  Edição, 2013.

https://github.com/puppeteer/puppeteer

https://github.com/mukulhase/WebWhatsapp-Wrapper

https://agasus.com.br/gestao-de-ti/conheca-os-tipos-de-licenca-de-software-e-sua-importancia/

https://medium.com/trainingcenter/tudo-o-que-voc%C3%AA-precisa-saber-sobre-as-licen%C3%A7as-de-projetos-open-source-aaccbe23e50d