**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 4](#__RefHeading___Toc2427_525165005)

[​ 2 – Desenvolvimento teórico 5](#__RefHeading___Toc2429_525165005)

[​ 2.1 – Gerenciamento da integração 5](#__RefHeading___Toc2594_525165005)

[2.1.1 - Objetivos do projeto 5](#__RefHeading___Toc615_3418941532)

[2.1.2 - Situação atual e justificativa do projeto 5](#__RefHeading___Toc617_3418941532)

[2.1.3 - Objetivos e critérios de sucesso do projeto 5](#__RefHeading___Toc2596_525165005)

[2.1.4 - Estrutura analítica do projeto – Fases e principais entregas 6](#__RefHeading___Toc2598_525165005)

[2.1.5 - Principais requisitos das principais entregas 6](#__RefHeading___Toc2600_525165005)

[2.1.6 - Marcos 6](#__RefHeading___Toc625_3418941532)

[2.1.7 - Partes interessadas do projeto 7](#__RefHeading___Toc627_3418941532)

[2.1.8 – Restrições 7](#__RefHeading___Toc629_3418941532)

[2.1.9 – Premissas 7](#__RefHeading___Toc631_3418941532)

[2.1.10 – Riscos 7](#__RefHeading___Toc633_3418941532)

[2.1.11 – Orçamento do projeto 7](#__RefHeading___Toc635_3418941532)

[​ 2.2 – Gerenciamento do escopo 8](#__RefHeading___Toc4720_525165005)

[2.2.1 – Abordagem de desenvolvimento 8](#__RefHeading___Toc4722_525165005)

[2.2.2 – Levantamento de requisitos 8](#__RefHeading___Toc5991_3354941199)

[2.2.3 – Ciclo de vida do projeto 8](#__RefHeading___Toc5993_3354941199)

[2.2.4 – EAP 9](#__RefHeading___Toc5972_3354941199)

[​ 2.3 – Gerenciamento do tempo 10](#__RefHeading___Toc4724_525165005)

[2.3.1 – Rede do projeto 10](#__RefHeading___Toc4726_525165005)

[2.3.2 – Tabela de precedência 10](#__RefHeading___Toc4728_525165005)

[2.3.3 – Caminhos 11](#__RefHeading___Toc4730_525165005)

[2.3.4 – Caminho crítico 11](#__RefHeading___Toc4732_525165005)

[2.3.5 – Descrição das atividades 11](#__RefHeading___Toc4734_525165005)

[​ 2.4 – Gerenciamento dos custos 12](#__RefHeading___Toc4233_3354941199)

[2.4.1 - Planejar o gerenciamento dos custos 12](#__RefHeading___Toc4235_3354941199)

[2.4.2 - Estimar os custos 12](#__RefHeading___Toc4237_3354941199)

[2.4.3 - Determinar o orçamento 13](#__RefHeading___Toc4239_3354941199)

[2.4.4 – Controlar custos 13](#__RefHeading___Toc4241_3354941199)

[​ 2.5 – Gerenciamento da qualidade 13](#__RefHeading___Toc4243_3354941199)

[​ 2.6 – Gerenciamento dos recursos 13](#__RefHeading___Toc4245_3354941199)

[2.6.1 - Planejamento 13](#__RefHeading___Toc5974_3354941199)

[2.6.2 - Recursos e atividades 14](#__RefHeading___Toc5976_3354941199)

[2.6.3 - Adquirir Recursos 14](#__RefHeading___Toc5978_3354941199)

[2.6.4 - Desenvolvimento da equipe 14](#__RefHeading___Toc5980_3354941199)

[2.6.5 - Gerenciamento da Equipe 14](#__RefHeading___Toc5982_3354941199)

[2.6.6 - Controle de Recursos 14](#__RefHeading___Toc5984_3354941199)

[2.6.7 - PMO 14](#__RefHeading___Toc1308_2759728032)

[​ 2.7 – Gerenciamento das comunicações 15](#__RefHeading___Toc4247_3354941199)

[2.7.1 – Matriz de comunicações 15](#__RefHeading___Toc6019_3354941199)

[​ 2.8 – Gerenciamento dos riscos 21](#__RefHeading___Toc4249_3354941199)

[​ 2.8.1 - Riscos do projeto 21](#__RefHeading___Toc4259_3354941199)

[2.8.2 – Matriz de riscos 23](#__RefHeading___Toc4261_3354941199)

[​ 2.9 – Gerenciamento das aquisições e contratações 23](#__RefHeading___Toc4251_3354941199)

[​ 3 – Desenvolvimento prático 24](#__RefHeading___Toc2431_525165005)

[3.1 – Metodologia 24](#__RefHeading___Toc655_3418941532)

[3.1.1 – Planejamento da Sprint 24](#__RefHeading___Toc6163_3354941199)

[3.1.2 – Reuniões diárias - Dailys 24](#__RefHeading___Toc6165_3354941199)

[3.2 – Linguagem de programação 25](#__RefHeading___Toc6153_3354941199)

[3.2.1 – Puppeteer 25](#__RefHeading___Toc1310_2759728032)

[3.2.2 – Socket 25](#__RefHeading___Toc1312_2759728032)

[3.2.3 – WAPI 26](#__RefHeading___Toc1314_2759728032)

[3.3 – Licensas 26](#__RefHeading___Toc659_3418941532)

[3.4 - Códigos 26](#__RefHeading___Toc1316_2759728032)

[3.5 - Funcionalidades 26](#__RefHeading___Toc661_3418941532)

[3.6 - Telas 26](#__RefHeading___Toc663_3418941532)

[​ 4 – Considerações finais 27](#__RefHeading___Toc2435_525165005)

[​ Referências 28](#__RefHeading___Toc2437_525165005)

# 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 |

**Controle de versões**

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

### 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 o mesmo 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 |

### 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: 3 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

**Requisitos não funcionais:**

• Resposta em até 5 segundos para o usuário

• Opções enumeradas

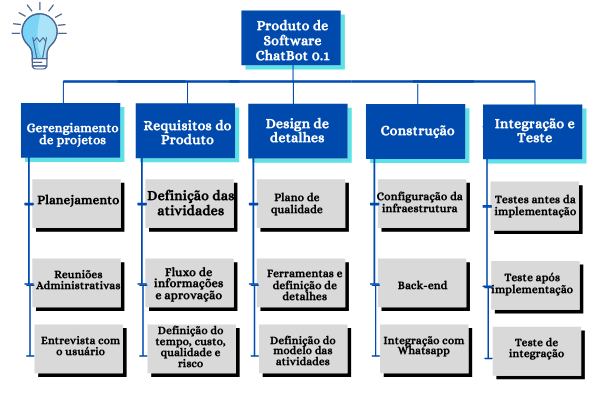
• Responder usuários de qualquer plataforma

### 2.2.3 – **Ciclo de vida do projeto**

* + Definição das atividades do projeto
  + Determinar a duração de cada atividade a ser realizada pelos desenvolvedores.
  + Identificação da sequência lógica de cada atividade
  + Determinar o tipo e a quantidade de recursos necessários
  + Determinar o custo
  + Montar a rede do projeto
  + Estruturar o diagrama de decisões do projeto
  + Criar modelos de entidades do banco de dados
  + Criar um fluxo de pedidos
  + Criar fluxo de informações do estabelecimento
  + Criar fluxo de comentários
  + Criar toda finalização do atendimento
  + Escrever testes
  + Finalização do projeto

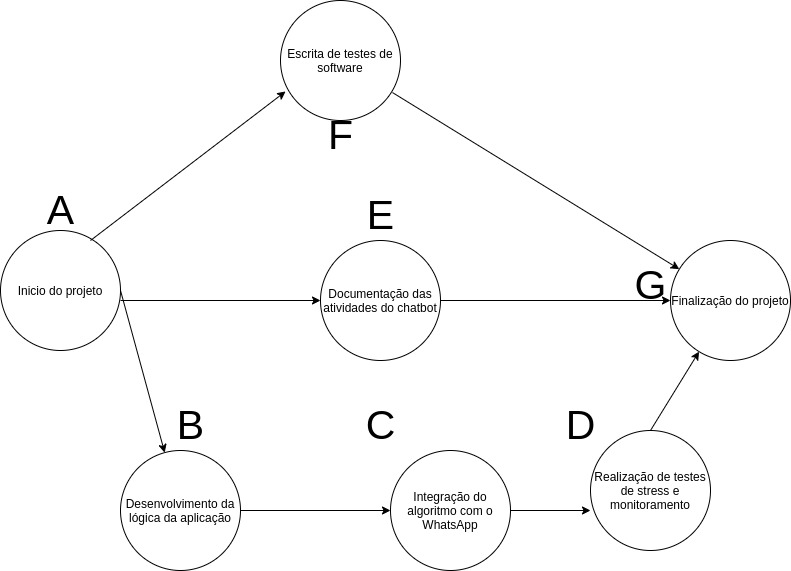
### 2.2.4 – **EAP**

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



## 2.3 – Gerenciamento do **tempo**

### 2.3.1 – **Rede do projeto**



### 2.3.2 – Tabela de precedência

| **ID** | **Atividade** | **Duração** | **Atividades Precedentes** |
| --- | --- | --- | --- |
| A | Inicio do projeto | - | - |
| B | Desenvolvimento da logica do bot | 20 dias | A |
| C | Integração do algoritmo com o Whatsapp | 2 dias | B |
| D | Realização de testes de stress e monitoramento de performance | 4 dias | C |
| E | Documentação das atividades do bot | 6 dias | A |
| F | Escrita de testes de software | 6 dias | A |
| G | Entrega do projeto | - | B, C, E |

### 2.3.3 – Caminhos

Os caminhos possíveis para esse projeto são:

* A, B, C, D, G – 26 dias
* A, E, G – 6 duas
* A, F, G – 6 dias

### 2.3.4 – Caminho crítico

O caminho crítico desse projeto é o A, B, C, D, G, (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

* Inicio do projeto
  + Inicialização do projeto e definição de quais caminhos vão ser seguidos, e em qual ordem
* Desenvolvimento da logica 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

## 2.6 – 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 das mesmas, 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 também 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**

## 2.7 – Gerenciamento das comunicações

### 2.7.1 – Matriz de comunicações

**Registros das partes interessadas**

* Qual informação? (What)
  + Registros das partes interessadas
* Qual propósito
  + Identificar partes interessadas e definir estrategias
* Quem é o responsável?
  + Gerente do projeto
* Quem precisa da informação?(Who)
  + Equipe de gerenciamento do projeto
* Quando? (When)
  + Inicio 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.

**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)
  + Inicio 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 titulo deve ser curto e direto ao ponto. Na descrição da tarefa deve ser levantado o porque da tarefa, quem deverá realizar e uma data final de conclusão.

**Registros das partes interessadas**

* Qual informação? (What)
  + Registros das partes interessadas
* Qual propósito
  + Identificar partes interessadas e definir estrategias
* Quem é o responsável?
  + Gerente do projeto
* Quem precisa da informação?(Who)
  + Equipe de gerenciamento do projeto
* Quando? (When)
  + Inicio 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.

**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)
  + Inicio 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.

**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)
  + Inicio 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.

**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)
  + Inicio 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.

**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)
  + Inicio 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

**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



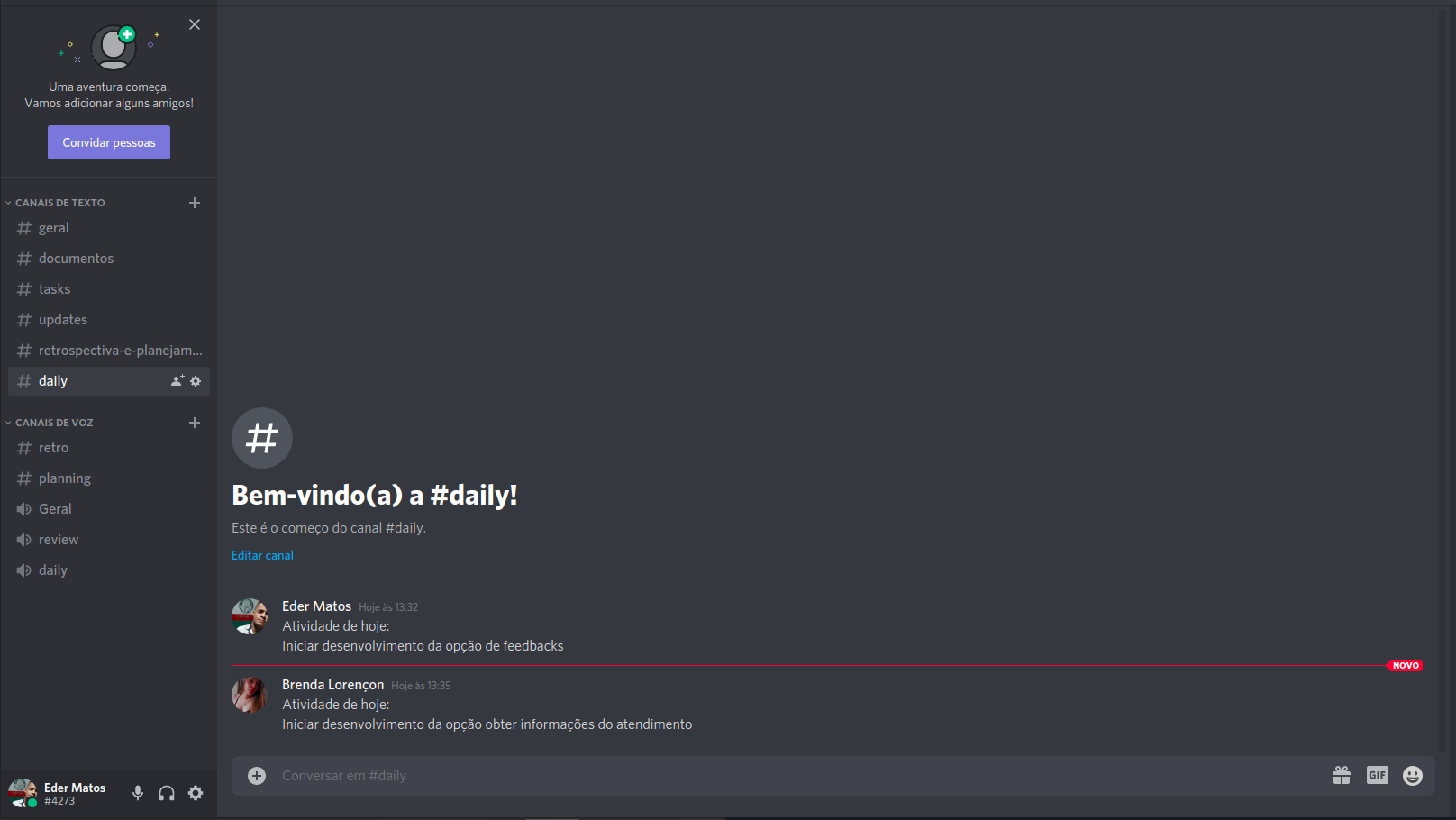
## 2.9 – Gerenciamento das aquisições e contratações

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.



### 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 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 usado 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 serie de funções escritas em JavaScript que podemos executar no navegador.

Uma desses 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 especifico.

## 3.3 – **Licensas**

## **3.4 - Códigos**

## 3.5 - Funcionalidades

## 3.6 - Telas

# 4 – Considerações finais

# Referências

https://github.com/puppeteer/puppeteer

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