**Lista de assuntos:**

**- Solicitação de boleto**

**- Negociação de dívida**

1. **Definir os Objetivos do Projeto**
   * Entender os requisitos específicos do negócio.
   * Estabelecer metas claras e mensuráveis.

Definições / Escopo / Exploratória

Assunto

Frontend

IA

Pergunta| Resposta

- Canvas

- Escopo

- Project Specification Document

- 5W2H

- What

- Why

- Where

- When

- Who

- Whom

- How

- How Much

- Métrica principal:

- Op1.: % de respostas com nota <= X

- Métricas secundárias. Exemplos:

- Op1.: % correção das respostas da IA, dado por:

e*quação de correção*

- % conversão grupo ação (agente com IA) x controle (agente sem IA)

- NPS ação x controle

- Etc.

- # Tokens / Custo

Quais informações do cliente?

- CPF

- Nome

- Idade

- Valor dívida

- Condições pré-aprovado

-

- Offline

-

Metas (Resultado esperado):

- 95% de boas respostas (Nota >= 3)

- No máximo 10% de respostas com nota 2

- No máximo 30% de respostas com nota 3

- Ao menos 50% de respostas com nota >= 4

- Nenhuma resposta com nota <=1

- Posso escalar:

-

50 dia

300 dia

> Tempo médio reposta

- Nota geral (NPS geral)

Cuidados:

- Viéses dos operadores ao fornecer a nota

- Os critérios podem ser subjetivos

- A régua de nota precisa estar bem clara, para evitar isso

- Podemos precisar de justificativas para a nota dada? Como faríamos isso?

- Há diferença entre as notas dadas por atendentes diferentes? Se sim, por quê?

- Fazer uma amostragem e calcular quanto cada operador está dando notas:

- “corretas”

- “incorretas” para mais ou para menos

Like, Deslike 🡪 Report

Notas:

0 – Extremamente crítico. Poderia gerar um impacto de imagem, processo judicial, etc.

1 – Crítico. Alto potencial de gerar reclamação e causar insatisfações no cliente

2 – Razoável. Não causaria danos severos ao banco, mas poderia ser mais empático/educado, etc.

3 – Boa resposta. Praticamente a mesma resposta que eu daria, porém com alguns ajustes...

4 – Ótima resposta. Praticamente a mesma resposta que eu daria.

5 – Excelente. Resposta ainda melhor do que o que eu faria.

- Fluxo AS IS x To Be

AS IS

1. Operador recebe a mensagem no WhatsApp (digitalk)

TO BE

1. Operador recebe a mensagem no WhatsApp (digitalk)
2. Copia e cola a mensagem na tela de IA
3. Avalia de 1 a 5 a resposta da IA
4. ...

- Análise exploratória

- Quantos atendentes de WhatsApp temos hoje? 4

- Qual horário de trabalho deles?

- Eles atendem apenas WhatsApp?

- Descrição do dia a dia de um atendente de WhatsApp

- Quantos clientes únicos são atendidos por:

- Dia, semana e mês

- Por assunto

- Por segmento de público

- Qual a # e % de conversão em negócio (se tiver)?

- Por operador

- É medido NPS desses clientes? Se sim, como e quais são os números?

- Geral

- Por operador

- Por assunto

- Por segmento

- Etc.

- Qual público é atendido pelo Whatsapp? Exemplo:

- PF Preventivos Clean até 30 dias em atraso

- Quais são os assuntos das conversas?

- Reclamação

- Negociação

- Solicitação de boleto

- Etc.

-

- Como é a dinâmica da conversa?

- É 100% receptivo?

- Qual o tempo de resposta do cliente?

- Qual o tempo de resposta do atendente?

- Qual o tempo entre o início e o fim da conversa?

- Exemplos de conversas via WhatsApp de diferentes cenários

11 mil procotolos - conversas whatsapp

- quantidade mensagens

- quantidade de caracteres

1. **Configuração do Ambiente de Desenvolvimento**
   * Escolher a linguagem de programação principal (Python, R, etc.).
   * Instalar IDEs e editores de código (VS Code, Jupyter Notebook).
   * Configurar um ambiente virtual (virtualenv, conda).

Python - Pyenv

Vs Code + Extensões

* **Python (Microsoft) OK**
* **Pylance OK**
* **Jupyter OK**
* **GitLens OK**
* **Docker OK**
* **WSL OK**
* **Markdown All in One OK**
* **GitHub Copilot OK**
* **Excel Viewer OK**
* **Pandas OK**
* **AREPL for Python OK**
* **Data Preview OK**
* **MagicPython OK**

*…or create a new repository on the command line*

echo "# chatbot-whatsapp-collections" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/caiquefilipini/chatbot-whatsapp-collections.git

git push -u origin main

*…or push an existing repository from the command line*

git remote add origin https://github.com/caiquefilipini/chatbot-whatsapp-collections.git

git branch -M main

git push -u origin main

Virtualenv

Editar arquivo de texto com vi

<https://pt.wikihow.com/Criar-e-Editar-um-Arquivo-de-Texto-no-Linux-Usando-o-Terminal#:~:text=Se%20fosse%20criar%20um%20arquivo%20chamado%20%E2%80%9CTrabalho%E2%80%9D%2C%20por%20exemplo%2C%20bastaria%20digitar%20vi%20Trabalho.txt>

documentação streamlit

<https://docs.streamlit.io/>

tutoriais:

<https://www.youtube.com/watch?v=sBhK-2K9bUc>

Sim, você precisa ter o Python instalado antes de criar um ambiente virtual. Vou detalhar o passo a passo para configurar o ambiente de desenvolvimento, incluindo a instalação do Python:

**Passo a Passo para Configuração do Ambiente de Desenvolvimento**

**Passo 1: Instalar o Python**

1. **Download e Instalação do Python:**
   * Acesse o site oficial do Python: [python.org](https://www.python.org/).
   * Vá para a seção de downloads e escolha a versão apropriada para o seu sistema operacional (Windows, macOS, Linux).
   * Baixe o instalador e siga as instruções de instalação. Certifique-se de marcar a opção "Add Python to PATH" durante a instalação no Windows.
2. **Verificar a Instalação:**
   * Abra o terminal (ou prompt de comando no Windows).
   * Digite o comando para verificar a versão do Python instalada:

bash

Copy code

python --version

ou

bash

Copy code

python3 --version

* + A saída deve mostrar a versão do Python instalada, por exemplo, **Python 3.9.1**.

**Passo 2: Criar a Pasta do Projeto**

1. **Criar a Pasta do Projeto:**
   * No terminal, navegue até o local onde você deseja criar a pasta do projeto.
   * Use o comando **mkdir** para criar uma nova pasta e navegue até ela:

bash

Copy code

mkdir nomeprojeto cd nomeprojeto

**Passo 3: Criar um Ambiente Virtual**

1. **Criar o Ambiente Virtual:**
   * No terminal, crie um ambiente virtual usando o comando **venv**:

bash

Copy code

python -m venv venv

ou

bash

Copy code

python3 -m venv venv

1. **Ativar o Ambiente Virtual:**
   * No Windows:

bash

Copy code

venv\Scripts\activate

* + No macOS/Linux:

bash

Copy code

source venv/bin/activate

1. **Verificar se o Ambiente Virtual Está Ativado:**
   * O prompt do terminal deve mudar para indicar que o ambiente virtual está ativo, mostrando **(venv)** no início da linha.

**Passo 4: Instalar Dependências Básicas**

1. **Instalar Bibliotecas Necessárias:**
   * Com o ambiente virtual ativado, instale as dependências básicas como **pandas**, **numpy**, **scikit-learn**, etc.:

bash

Copy code

pip install pandas numpy scikit-learn

1. **Gerar o Arquivo requirements.txt:**
   * Gere um arquivo **requirements.txt** para registrar as dependências do projeto:

bash

Copy code

pip freeze > requirements.txt

**Passo 5: Criar a Estrutura de Pastas**

1. **Criar Estrutura de Pastas:**
   * Crie uma estrutura básica de pastas dentro do projeto:

bash

Copy code

mkdir src tests data notebooks

* + **src:** Código-fonte do projeto.
  + **tests:** Testes unitários e de integração.
  + **data:** Datasets e arquivos de dados.
  + **notebooks:** Notebooks Jupyter para análise exploratória de dados.

**Passo 6: Configurar VS Code**

1. **Abrir o VS Code:**
   * Abra o VS Code na pasta do projeto:

bash

Copy code

code .

1. **Instalar Extensões:**
   * No VS Code, instale as seguintes extensões:
     + Python (Microsoft)
     + Pylance
     + Jupyter
     + GitLens
2. **Configurar o VS Code para Usar o Ambiente Virtual:**
   * Abra a paleta de comandos (**Ctrl+Shift+P** ou **Cmd+Shift+P**).
   * Digite e selecione **Python: Select Interpreter**.
   * Escolha o interpretador do ambiente virtual (**venv**).

**Passo 7: Configurar o Repositório do Projeto no GitHub**

1. **Inicializar o Repositório Git:**
   * No terminal, inicialize o repositório Git:

bash

Copy code

git init

1. **Criar o Arquivo .gitignore:**
   * Crie um arquivo **.gitignore** para ignorar arquivos desnecessários:

bash

Copy code

echo "venv/" > .gitignore echo "\*.pyc" >> .gitignore echo "\_\_pycache\_\_/" >> .gitignore

1. **Criar Repositório no GitHub:**
   * Crie um repositório no GitHub.
2. **Adicionar o Repositório Remoto:**
   * Adicione o repositório remoto ao seu projeto:

bash

Copy code

git remote add origin https://github.com/seuusuario/nomeprojeto.git

1. **Fazer o Primeiro Commit e Enviar para o GitHub:**
   * Adicione e faça commit das alterações:

bash

Copy code

git add . git commit -m "Initial commit" git push -u origin master

**Passo 8: Configurar CI/CD (GitHub Actions)**

1. **Criar a Pasta para Workflows do GitHub Actions:**

bash

Copy code

mkdir -p .github/workflows

1. **Criar o Arquivo ci.yml:**
   * Dentro da pasta **.github/workflows**, crie o arquivo **ci.yml** com o seguinte conteúdo:

yaml

Copy code

name: CI on: [push, pull\_request] jobs: build: runs-on: ubuntu-latest steps: - name: Checkout code uses: actions/checkout@v2 - name: Set up Python uses: actions/setup-python@v2 with: python-version: '3.x' - name: Install dependencies run: | python -m venv venv source venv/bin/activate pip install -r requirements.txt - name: Run tests run: | source venv/bin/activate pytest

1. **Criar o Arquivo requirements.txt:**
   * Se ainda não o fez, gere o arquivo **requirements.txt**:

bash

Copy code

pip freeze > requirements.txt

**Passo 9: Criar um Teste Simples**

1. **Criar um Arquivo de Teste:**
   * No diretório **tests**, crie um arquivo **test\_example.py** com um teste simples:

python

Copy code

def test\_example(): assert 1 + 1 == 2

1. **Executar os Testes Localmente:**
   * Verifique se tudo está funcionando executando os testes:

bash

Copy code

pytest

**Passo 10: Commit e Push das Alterações**

1. **Fazer Commit e Push das Novas Alterações:**

bash

Copy code

git add . git commit -m "Setup project structure and CI/CD" git push origin master

Agora você deve ter um ambiente de desenvolvimento totalmente configurado para iniciar o

1. **Infraestrutura e Ferramentas**
   * Configurar um repositório de código (GitHub, GitLab).
   * Definir o fluxo de trabalho Git (branching strategy, pull requests).
   * Selecionar e configurar um sistema de controle de versão.
   * Escolher ferramentas de integração contínua/delivery contínuo (CI/CD) (Jenkins, GitHub Actions).
2. **Coleta e Preparação de Dados**
   * Identificar fontes de dados relevantes (internas e externas).
   * Configurar acesso seguro às fontes de dados.
   * Estabelecer procedimentos de extração, transformação e carregamento (ETL).
   * Garantir a conformidade com leis de proteção de dados (LGPD, GDPR).
3. **Configuração de Banco de Dados**
   * Selecionar um sistema de gerenciamento de banco de dados (SQL, NoSQL).
   * Configurar bancos de dados de desenvolvimento, teste e produção.
   * Implementar backups regulares e planos de recuperação de desastres.
4. **Ferramentas de Análise de Dados**
   * Instalar bibliotecas e frameworks necessários (pandas, NumPy, scikit-learn, TensorFlow).
   * Configurar notebooks para exploração e análise de dados.
5. **Desenvolvimento e Testes de Modelos de Machine Learning**
   * Definir problemas de machine learning específicos para resolver.
   * Coletar e preparar datasets de treino e teste.
   * Desenvolver e treinar modelos de machine learning.
   * Implementar validação cruzada e tuning de hiperparâmetros.
   * Configurar pipelines de machine learning.
6. **Integração e APIs**
   * Desenvolver APIs para integração com sistemas externos.
   * Definir endpoints para funcionalidades principais do copilot.
7. **Segurança e Conformidade**
   * Implementar práticas de segurança para proteger dados sensíveis.
   * Garantir conformidade com regulamentos relevantes.
8. **Documentação e Treinamento**
   * Documentar o código e os procedimentos.
   * Criar tutoriais e guias para usuários e desenvolvedores.
   * Planejar sessões de treinamento para stakeholders.
9. **Monitoramento e Manutenção**
   * Configurar ferramentas de monitoramento de desempenho e logs (Prometheus, Grafana).
   * Estabelecer processos para manutenção contínua e atualizações de sistema.
10. **Gestão do Projeto**
    * Definir cronograma e marcos importantes.
    * Estabelecer uma equipe com papéis e responsabilidades claras.
    * Implementar ferramentas de gestão de projeto (Jira, Trello).

Project Specification Document

Chatbot Negociador de Dívidas

Fase 1: Copilot para atendente (POV)

Fase 2: Chatbot negociador

Fase 3: Chatbot negociador com áudio - TBD

- POV

Fase 4: Negociador virtual - TBD

-

Fase 1:

***1. Visão Geral do Projeto***

Nome do Projeto: Chatbot Negociador de Dívidas

Data de Início: 01/06/2024

Data de Término Prevista: 01/12/2024

Gerente do Projeto: Caíque Filipini

Equipe do Projeto:

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Empresa | Função | Responsabilidades |
| Caíque Filipini | Santander  (Recuperações) | Gerente de Projeto | * Coordenar todas as atividades do projeto. * Gerenciar cronogramas e prazos. * Facilitar a comunicação entre as partes interessadas. * Gerenciar riscos e resolver problemas. * Garantir que o projeto esteja alinhado com os objetivos de negócios. |
| Bruno Zan | Santander |  | Cientista de dados. |
| Victor Nolasco | Santander |  | Cientista de dados. |
|  |  |  |  |
|  |  |  |  |

**2. Objetivos do Projeto**

**Objetivo Principal:** Desenvolver uma ferramenta baseada em inteligência artificial para auxiliar negociadores de dívidas a otimizar suas interações com clientes e melhorar as taxas de recuperação de dívidas.

**Objetivos Específicos:**

1. Implementar um sistema de recomendação para propostas de negociação baseadas em perfis de clientes.
2. Automatizar a análise de dados históricos para identificar padrões de comportamento de pagamento.
3. Criar um chatbot inteligente que possa interagir com clientes e responder a perguntas frequentes.
4. Integrar a ferramenta com o CRM existente para facilitar a gestão dos dados dos clientes.
5. Garantir a conformidade com as regulamentações de proteção de dados, como a LGPD.

Escopo do Projeto

**Incluído:**

* Desenvolvimento de algoritmos de machine learning para análise de dados.
* Criação de um sistema de recomendação personalizado.
* Desenvolvimento de um chatbot baseado em IA.
* Integração com sistemas de CRM.
* Treinamento e suporte para usuários finais.

**Excluído:**

* Desenvolvimento de um novo CRM.
* Gestão direta de dados sensíveis sem consentimento.

Requisitos Funcionais

* **RF01:** O sistema deve ser capaz de analisar dados históricos de pagamentos e negociações.
* **RF02:** O sistema deve recomendar propostas de negociação personalizadas para cada cliente.
* **RF03:** O chatbot deve responder a perguntas frequentes e coletar informações básicas dos clientes.
* **RF04:** O sistema deve permitir a integração com CRM existente.
* **RF05:** O sistema deve gerar relatórios sobre a eficácia das negociações.

**5. Requisitos Não Funcionais**

* **RNF01:** O sistema deve garantir a privacidade e segurança dos dados dos clientes.
* **RNF02:** O sistema deve ser escalável para atender a um grande número de usuários simultaneamente.
* **RNF03:** O sistema deve ter alta disponibilidade, com uptime mínimo de 99.9%.

**7. Riscos do Projeto**

**Riscos Identificados:**

1. **Risco:** Dados insuficientes para treinar os modelos de IA.
   * **Mitigação:** Realizar coleta e análise de dados preliminares.
2. **Risco:** Integração com o CRM existente pode ser mais complexa do que o previsto.
   * **Mitigação:** Envolver especialistas em integração desde o início do projeto.
3. **Risco:** Resistência dos negociadores em adotar a nova ferramenta.
   * **Mitigação:** Implementar um programa de treinamento abrangente e sessões de feedback.

**8. Orçamento do Projeto**

**Recursos Humanos:**

* Desenvolvedores: $100,000
* Especialistas em IA: $50,000
* Gerente de Projeto: $30,000

**Infraestrutura:**

* Servidores e Cloud: $20,000
* Licenças de Software: $10,000

**Outros:**

* Treinamentos: $5,000
* Contingência: $10,000

**Total Estimado:** $225,000

**9. Aprovação do Projeto**

**Aprovado por:**

* **Nome:** João Silva  
  **Título:** Gerente de Projeto  
  **Data:** 31/05/2024

**Aprovado por:**

* **Nome:** Maria Oliveira  
  **Título:** Diretora de TI  
  **Data:** 31/05/2024

**Cerimônias**

**- Daily 10 min / objetivo / quem participa**

**- Weekly 1h / objetivo / quem participa**

**- Como fugir/contornar impeditivos**

**Cronograma Detalhado do Projeto**

**- Julho rodar primeira versão front**

**Fase 1: Planejamento (01/06/2024 - 15/06/2024)**

**Objetivo:** Definir os detalhes do projeto, escopo, requisitos, e montar a equipe.

| **Atividade** | **Data de Início** | **Data de Término** | **Responsável** |
| --- | --- | --- | --- |
| Reunião de Kickoff | 01/06/2024 | 01/06/2024 | Gerente de Projeto |
| Definição do Escopo | 02/06/2024 | 03/06/2024 | Gerente de Projeto |
| Levantamento de Requisitos | 04/06/2024 | 06/06/2024 | Gerente de Projeto, Analista de Dados |
| Análise de Riscos | 07/06/2024 | 08/06/2024 | Gerente de Projeto |
| Montagem da Equipe | 09/06/2024 | 10/06/2024 | Gerente de Projeto |
| Planejamento do Cronograma | 11/06/2024 | 12/06/2024 | Gerente de Projeto |
| Aprovação do Planejamento | 13/06/2024 | 15/06/2024 | Gerente de Projeto |

**Fase 2: Análise (16/06/2024 - 30/06/2024)**

**Objetivo:** Coletar e analisar dados, definir a arquitetura inicial e identificar necessidades técnicas.

| **Atividade** | **Data de Início** | **Data de Término** | **Responsável** |
| --- | --- | --- | --- |
| Coleta de Dados | 16/06/2024 | 20/06/2024 | Analista de Dados |
| Análise Exploratória de Dados | 21/06/2024 | 25/06/2024 | Analista de Dados |
| Definição da Arquitetura | 26/06/2024 | 30/06/2024 | Desenvolvedor Chefe |

**Fase 3: Design (01/07/2024 - 31/07/2024)**

**Objetivo:** Criar wireframes, protótipos, e especificações técnicas detalhadas.

| **Atividade** | **Data de Início** | **Data de Término** | **Responsável** |
| --- | --- | --- | --- |
| Design de UX/UI | 01/07/2024 | 10/07/2024 | Designer UX/UI |
| Criação de Protótipos | 11/07/2024 | 20/07/2024 | Designer UX/UI |
| Revisão de Protótipos | 21/07/2024 | 25/07/2024 | Gerente de Projeto, Stakeholders |
| Definição de Especificações Técnicas | 26/07/2024 | 31/07/2024 | Desenvolvedor Chefe, Especialista em IA, Engenheiro DevOps |

**Fase 4: Desenvolvimento (01/08/2024 - 30/10/2024)**

**Objetivo:** Implementar funcionalidades principais, desenvolver modelos de machine learning, e configurar integrações.

| **Atividade** | **Data de Início** | **Data de Término** | **Responsável** |
| --- | --- | --- | --- |
| Configuração do Ambiente | 01/08/2024 | 05/08/2024 | Engenheiro DevOps |
| Desenvolvimento de Funcionalidades Básicas | 06/08/2024 | 20/08/2024 | Desenvolvedores |
| Desenvolvimento de Modelos de ML | 21/08/2024 | 10/09/2024 | Especialistas em IA |
| Integração com CRM | 11/09/2024 | 30/09/2024 | Especialista em Integração |
| Desenvolvimento de Chatbot | 01/10/2024 | 15/10/2024 | Desenvolvedores, Especialistas em IA |
| Testes Unitários e de Integração | 16/10/2024 | 30/10/2024 | Desenvolvedores, Especialistas em IA |

**Fase 5: Testes (01/11/2024 - 15/11/2024)**

**Objetivo:** Validar a funcionalidade, desempenho e segurança do sistema.

| **Atividade** | **Data de Início** | **Data de Término** | **Responsável** |
| --- | --- | --- | --- |
| Testes de Funcionalidade | 01/11/2024 | 05/11/2024 | Desenvolvedores, QA |
| Testes de Desempenho | 06/11/2024 | 10/11/2024 | Engenheiro DevOps |
| Testes de Segurança | 11/11/2024 | 15/11/2024 | Especialista em Segurança |

**Fase 6: Implementação (16/11/2024 - 30/11/2024)**

**Objetivo:** Desplegar o sistema em produção e realizar treinamentos para os usuários finais.

| **Atividade** | **Data de Início** | **Data de Término** | **Responsável** |
| --- | --- | --- | --- |
| Deploy em Produção | 16/11/2024 | 20/11/2024 | Engenheiro DevOps |
| Treinamento de Usuários | 21/11/2024 | 25/11/2024 | Gerente de Projeto, Designer UX/UI |
| Sessões de Feedback e Ajustes Finais | 26/11/2024 | 30/11/2024 | Todos os membros da equipe |

**Fase 7: Avaliação (01/12/2024)**

**Objetivo:** Avaliar o sucesso do projeto, documentar lições aprendidas e planejar futuras melhorias.

| **Atividade** | **Data de Início** | **Data de Término** | **Responsável** |
| --- | --- | --- | --- |
| Reunião de Avaliação | 01/12/2024 | 01/12/2024 | Gerente de Projeto, Stakeholders |