**SPRINT**

**FIAP**

**INTELIGÊNCIA ARTIFICIAL E COMPUTACIONAL**

**- Bruno Cheng Yu Jin | RM: 93931**

**- Caio Manoel Bezerra de Araujo | 94778**

**- Daniel Kramer de Holleben Santos| RM: 94277**

**- Luana Domingos Dias | RM: 93191**

**- Lucas Ferraz Martins | RM: 94821**

**- Rafaela Hitomi Osugi | RM: 95062**

**Introdução**

Nós do grupo da Cheese Bread estamos desenvolvendo um primeiro api rest que contêm uma IA do ChatGPT o qual auxilie os empregados da SANOFI a gerarem arquivos pré prontos apenas para preencherem, portanto não precisando fazer a mesma tarefa repetidamente de fazer um novo tipo de arquivo para cada tipo de situação. Essa API rest que estamos desenvolvendo também contará com a função de atualizar os status de entrega dos produtos da SANOFI, assim será dois projetos em um.

Para esse projeto estamos investindo todo o nosso tempo para projetar a parte do back end e entender como vamos aplicar a regra de negócio. Apenas quando toda essa parte estiver finalizada que nós iremos focar na parte de interface de usuário. A ideia da parte front end é que ela seja o mais simples e intuitiva possível, para que não consuma muito tempo dos funcionários da SANOFI, consequentemente possibilitando uma produção mais eficiente.

**Diferenças de um projeto para outro**

A princípio o nosso projeto era apenas desenvolver uma IA que auxiliasse na criação de arquivos pré prontos, economizando o tempo dos funcionários da SANOFI. Então inicialmente o objetivo era desenvolver uma API rest que utilizasse a API da OpenAI e que gerasse os arquivos.

Entretanto houve mudanças que acabaram acrescentando mais trabalhos a se fazer. Primeiramente, apenas criando nosso próprio ChatGPT que será possível utilizar a chave de requisição. Outro fator que muda é que a OpenAI deu certa preferência ao python em questão de desenvolvimento de API, dificultando o desenvolvimento da API rest principal que seria feito em Java. A última mudança foi ser necessário incluir mais um projeto da SANOFI, que seria um sistema o qual pudesse atualizar os status de entrega dos produtos vendidos pela empresa.

Em suma, para realizar o nosso projeto é necessário criar uma API rest própria em python apenas para lidar com a parte de IA, e assim implementá-la para a nossa API principal que será feita em java. Dessa forma, é mais fácil incluir a IA e ela fica totalmente separada da API principal, facilitando a parte de manutenção. Na API principal será incluído todo um sistema contendo a lógica e as páginas para a parte de atualização das entregas dos produtos da SANOFI.

**Detalhes do projeto**

Uma parte do grupo está focado em apenas desenvolver uma API rest de python para fazer a IA, outra parte está fazendo a API principal, e a outra focada no banco de dados. O framework que está sendo utilizado para a API de IA é o flask, justamente por pegar as respostas que o ChatGPT irá nos fornecer e poder transformá-la em JSON, e então na API principal iremos consumir essa API de IA e trabalhar tudo em cima dela. Para a API de IA, as bibliotecas usadas atualmente são: a de pandas, que serve para consumir a resposta do GPT e criar tabelas baseadas nela; a Document que está sendo usada para gerar os arquivos word e excel no final quando for solicitado; e por fim a biblioteca da OpenAI, para poder utilizar a API key do ChatGPT. Essas são as bibliotecas utilizadas atualmente, lembrando que a parte da API da IA está em desenvolvimento, uma vez que está sendo analisado o que é necessário para poder fazer o projeto sem problemas. Já na parte da API principal estamos usando java com o Spring Boot para criar a API rest e poder desenvolver toda a aplicação do banco de dados, que será feita com o MySQL. E agora temos o desafio de incluir toda a parte de status de entrega que será aplicado nessa API principal. Por fim, quando todo o back end estiver pronto, será usado o flutter para poder gerar toda a parte visual. Ainda não temos muito detalhes de como fazer essa parte, por justamente não ter sido pensado melhor, pois agora não é o foco.

**Explicação breve do funcionamento**

**Seções:**

1. **AI API (Python Flask)**
   * **Framework**: Flask
   * **Bibliotecas**:
     + pandas: Consome as respostas do ChatGPT e cria tabelas.
     + Document: Gera arquivos Word e Excel.
     + OpenAI: Utiliza a API key do ChatGPT.
   * **Função**:
     + Recebe as respostas do ChatGPT.
     + Transforma as respostas em JSON.
     + Cria tabelas e gera documentos quando solicitado.
   * **Conexão**: Envia dados em formato JSON para a API Principal.
2. **Main API (Java Spring Boot)**
   * **Framework**: Spring Boot
   * **Banco de Dados**: MySQL
   * **Função**:
     + Consome os dados JSON da AI API.
     + Processa os dados e gerencia o status de entrega.
   * **Conexão**: Fornece backend para o Frontend.
3. **Frontend (Flutter)**
   * **Framework**: Flutter
   * **Função**:
     + Gera a interface visual da aplicação.
   * **Conexão**: Consome dados da Main API.

**Forma que a IA esta sendo implementada nesse projeto**

**Modelo ChatGPT**: Utiliza IA para gerar respostas em linguagem natural.

**Manipulação de Dados**: Usa pandas para organizar e preparar dados.

**Automação**: Gera documentos automaticamente com Document.

**Integração de Sistemas**: Consome e processa dados entre APIs diferentes.

**Gestão de Status**: Monitora e gerencia processos de entrega com base em dados da IA.

**Interface de Usuário**: Exibe dados processados de forma interativa e amigável.