



**INSTITUTO BRASILEIRO DE ENSINO, DESENVOLVIMENTO E PESQUISA
CIÊNCIA DA COMPUTAÇÃO**

**BRUNA PEREIRA DA CUNHA, JOÃO MANOEL FREILI MELO NUNES, PEDRO
HENRIQUE NOBRE FORMIGA CHAVES**

HACKATHON

**BRASÍLIA – DF
2025**

BRUNA PEREIRA DA CUNHA, JOÃO MANOEL FREILI MELO NUNES, PEDRO
HENRIQUE NOBRE FORMIGA CHAVES

HACKATHON

**Trabalho apresentado ao curso Ciência da
Computação, como parte dos requisitos necessários à
obtenção do título de Hackathon.**

Professor: Lucas Martins

Disciplina: (...)

Brasília – DF

2025

Resumo

Este documento descreve tecnicamente um sistema desenvolvido em Python utilizando o microframework Flask. O sistema consiste em uma aplicação web que permite aos usuários submeterem informações sobre suas habilidades técnicas, aprendizados recentes e preferências de projeto, recebendo em troca uma sugestão de área profissional e uma justificativa baseada em um modelo de Machine Learning. Adicionalmente, o sistema inclui módulos para processamento e análise de dados de currículos, bem como para o treinamento do modelo de previsão.

SUMÁRIO

1 INTRODUÇÃO	5.
2 ARQUITETURA DO SISTEMA.....	6.
3 DESCRIÇÃO DOS MÓDULOS PRINCIPAIS	7.
3.1 Módulo de Inicialização e Execução (run.py, app/ __init__.py).....	7.
3.2 Módulo de Rotas e Interface com Usuário (app/routes.py).....	7.
3.3 Módulo de Lógica de Negócios e Machine Learning (app/models.py).....	7.
3.4. Módulos de Processamento de Dados de Currículos (dados.py, utils.py).....	8.
4 FLUXO DE DADOS E PROCESSAMENTO.....	9.
4.1 Treinamento do Modelo (via rota /treinar).....	9.
4.2 Previsão de Área para Usuário (via rota /).....	9.
4.3. Análise de Currículos (execução dos scripts dados.py/utils.py).....	9.
5 TECNOLOGIAS UTILIZADAS.....	10.
6 ARQUIVOS DE DADOS (CSV).....	11.
7 CONCLUSÃO.....	12.

1 INTRODUÇÃO

O sistema visa auxiliar na identificação de áreas de atuação potenciais para profissionais e estudantes da área de tecnologia, com base em suas competências e interesses. Ele é composto por uma interface web interativa e um backend que processa as informações do usuário e utiliza um modelo preditivo para fornecer recomendações. Scripts auxiliares permitem a análise de currículos e o treinamento do modelo.

2 ARQUITETURA DO SISTEMA

O sistema é modular e compreende os seguintes componentes principais:

- **Aplicação Web (Interface e Controle):** Desenvolvida com Flask, responsável por receber as entradas do usuário, interagir com o módulo de predição e apresentar os resultados.
- **Módulo de Modelagem e Predição:** Contém a lógica para treinar um modelo de classificação (Random Forest) e para realizar previsões de área profissional com base nos dados fornecidos.
- **Módulos de Processamento de Dados:** Scripts para pré-processamento e análise de currículos em formato CSV, extraindo habilidades e gerando relatórios.

3 DESCRIÇÃO DOS MÓDULOS PRINCIPAIS

3.1 Módulo de Inicialização e Execução (run.py, app/__init__.py)

- **run.py:** Ponto de entrada da aplicação. Responsável por criar a instância da aplicação Flask através da factory `create_app` e iniciar o servidor de desenvolvimento.
- **app/__init__.py:** Contém a factory `create_app()`. Inicializa a aplicação Flask, configura uma chave secreta (`SECRET_KEY`) e registra os blueprints (conjuntos de rotas) da aplicação, como o bp definido em `app/routes.py`.

3.2 Módulo de Rotas e Interface com Usuário (app/routes.py)

Define as rotas da aplicação e as funções controladoras associadas:

- **Rota / (Métodos GET e POST):**
 - **GET:** Apresenta o formulário principal (renderizado a partir de um template HTML, e.g., `index.html`) onde o usuário pode inserir suas tecnologias conhecidas, aprendizado recente e tipo de projeto de interesse. Popula os campos do formulário com opções predefinidas (e.g., `TECNOLOGIAS_POSSIVEIS`).
 - **POST:** Coleta os dados submetidos pelo usuário. Invoca a função `prever_area` do módulo `app.models` para obter a sugestão de área e sua justificativa. Exibe o resultado ou mensagens de erro na interface.
- **Rota /treinar (Método GET):**
 - Dispara o processo de treinamento do modelo de Machine Learning, utilizando o arquivo `alunos.csv`. Invoca a função `treinar_modelo` de `app.models`. Fornece feedback ao usuário sobre o sucesso ou falha do treinamento.

3.3 Módulo de Lógica de Negócios e Machine Learning (app/models.py)

Contém a inteligência central do sistema:

- **Constantes:** Define listas de tecnologias (`TECNOLOGIAS_POSSIVEIS`), categorias de aprendizado (`CATEGORIAS_APRENDIZADO_POSSIVEIS`) e tipos de projeto (`CATEGORIAS_PROJETO_POSSIVEIS`), além de mapeamentos detalhados de áreas.

- **treinar_modelo(caminho_csv):** Lê os dados de treinamento do arquivo alunos.csv. Realiza o pré-processamento dos dados utilizando MultiLabelBinarizer (para tecnologias) e OneHotEncoder (para aprendizado e tipo de projeto). Treina um classificador RandomForestClassifier e persiste o modelo treinado e o transformador de dados (pipeline) em um arquivo .pkl usando joblib.
- **carregar_modelo_e_transformer():** Carrega o modelo e o transformador persistidos.
- **prever_area(candidato_dict):** Recebe um dicionário com os dados do usuário. Utiliza o transformador carregado para preparar os dados e o modelo treinado para prever a área profissional. Gera uma justificativa para a previsão através da função justificar_previsao.
- **justificar_previsao(...):** Cria um texto explicativo para a área sugerida, detalhando habilidades correspondentes, potencial de desenvolvimento e conhecimentos complementares.

3.4. Módulos de Processamento de Dados de Currículos (dados.py, utils.py)

Estes scripts (com funcionalidades sobrepostas ou representando diferentes estágios de desenvolvimento) são destinados à análise em lote de arquivos de currículos:

- **preprocessar_dados(input_file, output_file):** Carrega dados de um arquivo CSV (e.g., CV.csv), remove duplicatas com base no conteúdo do currículo e salva o resultado em um novo CSV.
- **analisar_candidatos(df, requisitos):** Processa um DataFrame de currículos. Gera um ID único para cada candidato. Para uma lista de habilidades (requisitos) definida, verifica a presença de cada habilidade nos currículos (utilizando expressões regulares). Calcula um score e percentual de compatibilidade, e lista as habilidades encontradas e faltantes para cada candidato. Os resultados são usualmente salvos em um arquivo CSV (e.g., candidatos_web_design_detalhado.csv).

4 FLUXO DE DADOS E PROCESSAMENTO

4.1 Treinamento do Modelo (via rota /treinar)

1. O administrador acessa a rota /treinar.
2. O sistema lê o arquivo alunos.csv.
3. Os dados são pré-processados (binarização e one-hot encoding).
4. Um modelo RandomForestClassifier é treinado com esses dados.
5. O modelo treinado e o transformador são salvos em disco (modelo_predicao_area.pkl).

4.2 Previsão de Área para Usuário (via rota /)

1. Usuário acessa a interface web (formulário).
2. Usuário preenche e submete o formulário com suas tecnologias, aprendizado e tipo de projeto.
3. Os dados são enviados ao backend (Flask).
4. A função prever_area é chamada: a. O modelo e transformador previamente treinados são carregados. b. Os dados do usuário são transformados para o formato esperado pelo modelo. c. O modelo prediz a área. d. Uma justificativa para a predição é gerada.
5. A área prevista e a justificativa são exibidas na interface do usuário.

4.3. Análise de Currículos (execução dos scripts dados.py/utils.py)

1. Um arquivo CSV de entrada (e.g., CV.csv) é fornecido ao script.
2. O script pré-processa os dados (limpeza, remoção de duplicatas).
3. Para cada currículo, o script analisa a presença de habilidades predefinidas.
4. Um relatório detalhado, geralmente em formato CSV (e.g., candidatos_web_design_detalhado.csv), é gerado com a pontuação, habilidades encontradas e faltantes por candidato.

5 TECNOLOGIAS UTILIZADAS

- Linguagem de Programação: Python
- Framework Web: Flask
- Bibliotecas de Machine Learning e Manipulação de Dados:
 - Scikit-learn (RandomForestClassifier, MultiLabelBinarizer, OneHotEncoder, ColumnTransformer, Pipeline)
 - Pandas (manipulação de DataFrames)
 - NumPy (operações numéricas)
 - Joblib (persistência de modelos)
- **Outras Bibliotecas:** os (interação com o sistema operacional), re (expressões regulares), uuid (geração de identificadores únicos).

6 ARQUIVOS DE DADOS (CSV)

- **alunos.csv:** Utilizado para treinar o modelo de previsão de área. Contém colunas como 'Tecnologias_Conhecidas', 'Aprendizado_Recente', 'Tipo_Projeto_Preferido' e a 'Area_Prevista_Correta' (target).
- **CV.csv / Curriculum Vitae.csv:** Arquivos de entrada para os scripts de análise de currículos. Contêm dados textuais de currículos.
- **candidatos_web_design_detalhado.csv:** Arquivo de saída gerado pelos scripts de análise, detalhando a compatibilidade de candidatos com requisitos específicos.
- (Outros arquivos CSV mencionados podem ser intermediários ou variações).

7 CONCLUSÃO

O sistema de análise de perfil profissional e sugestão de área, documentado neste relatório, demonstra uma arquitetura modular e funcional para atender a dois propósitos principais: (1) fornecer aos usuários uma orientação de carreira personalizada através de uma interface web interativa, utilizando um modelo de Machine Learning para sugestões baseadas em dados; e (2) oferecer ferramentas para processamento e análise em lote de currículos, auxiliando na identificação de talentos com base em habilidades específicas.

A utilização do microframework Flask para a aplicação web, combinada com bibliotecas robustas como Scikit-learn e Pandas para a modelagem de dados e Machine Learning, resulta em uma solução eficiente e escalável. Os scripts auxiliares para tratamento de dados de currículos complementam a funcionalidade principal, permitindo uma análise de dados mais ampla e a preparação de insumos para o treinamento do modelo preditivo.

O sistema apresenta-se como uma ferramenta valiosa tanto para indivíduos em busca de direcionamento profissional na área de tecnologia quanto para recrutadores ou gestores que necessitam analisar perfis de candidatos de forma sistemática. Futuras melhorias podem incluir a expansão do conjunto de dados de treinamento para aprimorar a acurácia do modelo, a incorporação de novas tecnologias e áreas do conhecimento, e o desenvolvimento de uma interface de administração mais elaborada para gerenciamento do sistema e dos dados.

Em suma, o projeto integra com sucesso técnicas de desenvolvimento web, processamento de dados e aprendizado de máquina para criar uma solução prática e informativa no domínio da orientação profissional e análise de competências.