

## MODELO PARA A ENTREGA DAS ATIVIDADES

COMPONENTE CURRICULAR:	Projeto Aplicado II
NOME COMPLETO DO ALUNO:	Willians Carvalho da Silva -10416087
	João Pedro Santos Oliveira – 10423752
	Cesar Valentim Silva – 10416087
	Luciano Guimaraes Costas – 10289655

UNIVERSIDADE PRESBITERIANA MACKENZIE

CURSO DE TECNOLOGIA EM CIÊNCIA DE DADOS

PROJETO APLICADO II

PROPOSTA DE PROJETO

Título: Desenvolvimento de uma Aplicação de Processamento de Linguagem

Natural com Uso de Modelos da Hugging Fac

## Sumário

### 1. Introdução

- Apresentação do projeto e seus integrantes.
- Empresa escolhida: Hugging Face.
- Área de atuação: Processamento de Linguagem Natural (PLN).

### 2. Premissas do Projeto

- Dados utilizados: Datasets públicos compatíveis com modelos de PLN da Hugging Face.

### 3. Objetivos e Metas

- Objetivo Geral: Desenvolver uma aplicação para análise de textos utilizando modelos avançados de PLN.
- Metas Específicas:
  - Integração de modelos de PLN para classificação e sumarização de texto.
  - Desenvolvimento de uma interface de usuário interativa.
  - Avaliação da precisão das análises geradas.

### 4. Cronograma de Atividades

- Mês 1: Revisão de literatura e seleção de datasets.
- Mês 2: Integração de modelos de PLN e desenvolvimento da interface.
- Mês 3: Testes iniciais, coleta de feedback, ajustes e melhorias.
- Mês 4: Documentação, finalização e apresentação do projeto.

## 5. Definição do Método Analítico

Link do git: <https://github.com/joaosoliveira0907/projeto-aplicado2>

### 0. Linguagem de Programação Usada no Projeto

- O projeto será desenvolvido utilizando Python, por sua versatilidade em PLN e sua compatibilidade com a biblioteca Transformers da Hugging Face.
- Bibliotecas:
  - **Transformers:** Para modelos pré-treinados de PLN.
  - **Pandas e Numpy:** Manipulação e análise de dados.
  - **Scikit-learn:** Avaliação e métricas.
  - **Streamlit:** Interface de usuário.

### 1. Análise Exploratória da Base de Dados

- Dados explorados incluem:
  - **Distribuição das classes.**
  - **Comprimento dos textos.**
  - **Pré-processamento linguístico:** Ruídos, como caracteres especiais e palavras irrelevantes.

- Visualizações:
  - Histogramas de distribuição do tamanho dos textos.
  - Gráficos de barras para classes.

## 2. Tratamento da Base de Dados

- Limpeza e pré-processamento dos textos:
  - **Remoção de caracteres especiais e stop words.**
  - **Tokenização** com os modelos Hugging Face.
  - Divisão em **conjuntos de treinamento e teste.**
- **Fine-tuning** de modelos pré-treinados para adaptação aos dados do projeto.

## 3. Métodos Utilizados

- Base teórica baseada em **Modelos de Linguagem Baseados em Transformers** (BERT, GPT ou T5). Esses modelos utilizam a arquitetura **Transformer** para processar e gerar texto.
- Técnicas: **Tokenização** e **Fine-tuning** com Transfer Learning.

## 4. Definição e Cálculo de Acurácia

- A **acurácia** será calculada como a proporção de previsões corretas feitas pelo modelo.

- **Fórmula:**

$$\text{Acurácia} = (TP+TN+FP+FN)/(TP+TN)$$

**onde:**

TP (True Positives): número de instâncias corretamente classificadas como positivas.

TN (True Negatives): número de instâncias corretamente classificadas como negativas.

FP (False Positives): número de instâncias incorretamente classificadas como positivas.

FN (False Negatives): número de instâncias incorretamente classificadas como negativas.

- Outras métricas: **Precisão**, **Revocação** e **F1-score** para análise completa.

## 5. Base de Dados Escolhida

- Dataset público disponível na **Hugging Face**. Contém textos variados (notícias, resenhas) com rótulos para tarefas de classificação e sumarização.
- Nome do Dataset: [Especificar nome do dataset escolhido].
- Dados incluem um campo de texto e um rótulo para classificação.

## 6. Bibliotecas Python

## 7. Manipulação de dados:

- **Pandas e Numpy:** Estruturas de dados e operações numéricas.

## 8. Processamento de Linguagem Natural:

- **Transformers:** Para usar os modelos da Hugging Face.
- **spaCy e NLTK:** Pré-processamento textual (tokenização, remoção de stop words).

## 9. Treinamento e Avaliação:

- **Scikit-learn:** Algoritmos de aprendizado supervisionado e métricas.
- **TensorFlow ou PyTorch:** Para construção de redes neurais.

## 10. Visualização:

- **Matplotlib e Seaborn:** Gráficos e visualizações.