

**Análise Textual - Obras Brasileiras**

**Projeto Aplicado 2**

Brasil, 03 de Março de 2023

**Resumo**

Esse projeto consiste em realizarmos análises de textos afim de obtermos análises mais completas e detalhadas através de inteligência artificial.

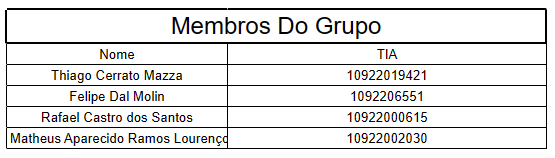
Baseado em bibliotecas como spacy e torch analisar obras brasileiras e identificarmos entidades nas palavras que mais se destacam, verbos, adjetivos, pronomes. E através da biblioteca wordcloud trazermos de uma maneira mais visual essa análise.

Utilizando a metodologia de Multi-Layer-Perceptron realizaremos as análises propostas chegando nos objetivos que queremos, onde com sua aplicação podemos realizar previsões em cima dos textos baseada em dados numéricos. Essa aplicação poderá gerar um chatbots, ou até mesmo sistemas de QA, seguindo os padrões da obra analisada.

# Apresentação Projeto Aplicado 2

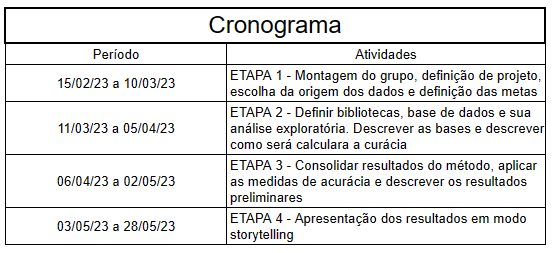
**Metodologia:** Através da biblioteca Spacy faremos as análises em cima do texto como separação de verbos, pronomes, adjetivos e com isso gerando um padrão de reconhecimento, destacando itens e palavras em comum pela biblioteca wordcloud. Além de através da inteligência artificial treinar uma IA para encima do texto proposto gerar um novo texto baseado no original.

**Membros do Grupo:**



**Contexto do estudo:** Manipulação de Texto - Obras Brasileiras através de reconhecimentos de entidades nomeadas. A obra que será utilizada como referência é a Trio em Lá Menor de Machado de Assis.

**Cronograma:**

****

**Link para o github:**

https://github.com/thiagomazza/Mackenzie---Projeto-Aplicado-2.git

# 

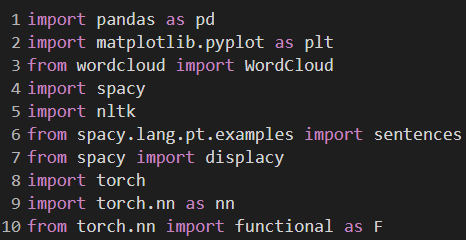
# Objetivos e Metas

Nosso objetivo é seguirmos com a análise do texto a fim de chegarmos aos seguintes objetivos/metas:

* Análise e separação gramatical
* Análise e separação de entidades (MER)
* Utilização dos algoritmos e treinamento para aperfeiçoamento
* Geração textual com verossimilhança com a base
* Reconhecer entidades nomeadas
* Avaliar por similaridade o texto gerado

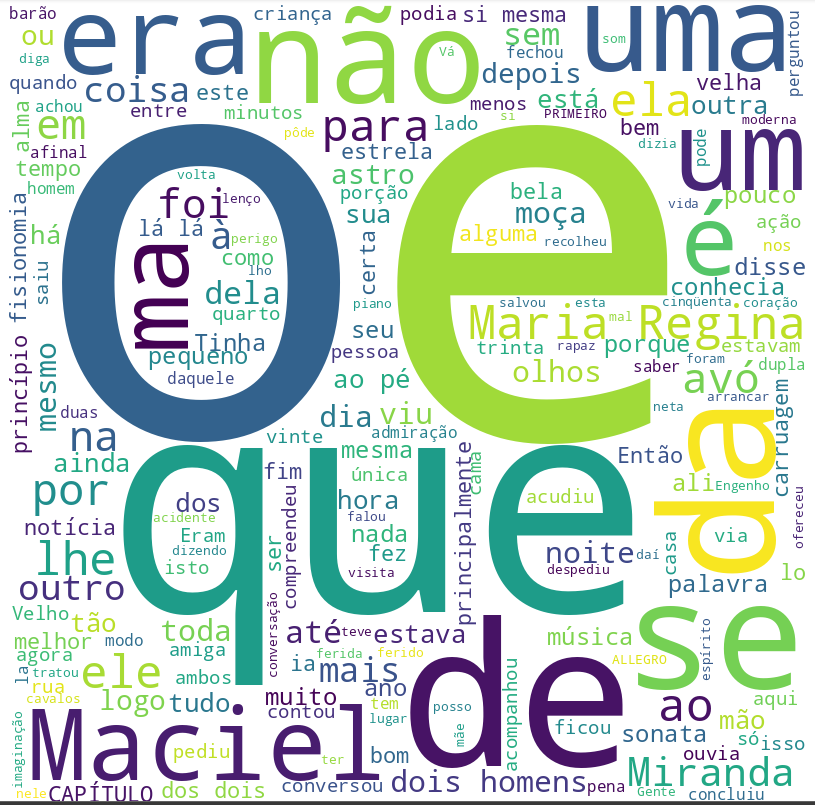
# Análise Exploratória da Base

Pacotes e linguagem: Iremos utilizar o Python como linguagem a ser utilizada e conforme print abaixo utilizaremos Pandas, matplotlib, wordcloud, spacy, nltk e torch:



Através da análise exploratória obtivemos algumas informações, destacamos todo o texto e fizemos dois tipos de análises: análise textual e análise NER.

Em uma análise inicial simples, apenas para identificarmos as palavras mais descritas no texto tempo a nuvem de palavras abaixo (biblioteca wordcloud):



Análise textual, através das bibliotecas spacy, token fizemos a separação gramatical das palavras dentro do texto, como:

* Substantivos
* Verbos
* Adjetivos
* Advérbios
* Outros

Na imagem abaixo podemos destacar através da biblioteca wordcloud uma nuvem das palavras com maiores relevâncias dos itens destacados na relação acima:



Análise NER, através da biblioteca spacy utilizamos o método de reconhecimento textual de entidade nomeada. Entre elas destacamos palavras como:

* Pessoas
* Locais
* Entidades / Organizações
* Diversos (demais tipos)

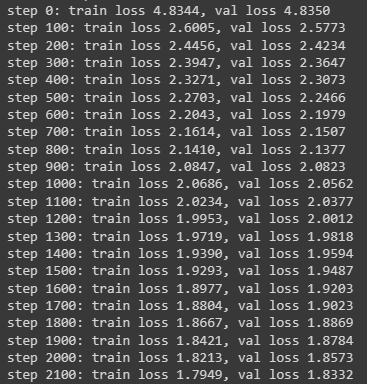
N imagem abaixo destacamos através da biblioteca wordcloud as nuvens de palavras:



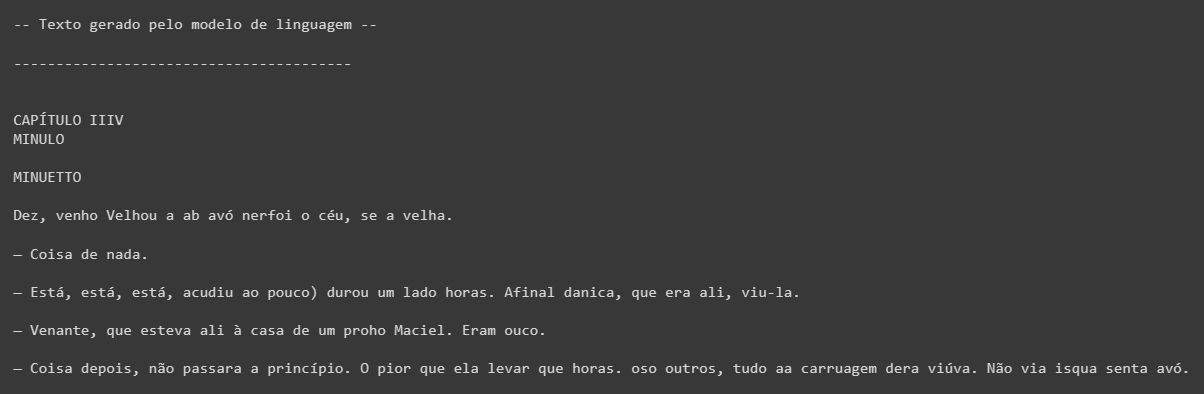
Através desta análise é possível verificar que o algoritmo não consegue ser tão assertivo quanto à análise textual anterior.

Após a análise textual fizemos o treinamento da IA encima do texto proposto a fim de identificar sua saída. nosso principal objetivo era com que o texto de saída fosse o mais próximo possível do texto original.

Na imagem abaixo conseguimos identificar os passos do treinamento, onde são indicados as quantidades de perdas e tentativas para se chegar ao mais próximo do texto original:



O início do texto de saída ficou da seguinte maneira:



Veja que ele se perde em algumas palavras, acaba concatenando outras e gerando novas palavras.

Iremos utilizar o modelo de linguagem PyTorch, onde a partir do texto fornecido ele realiza uma série de etapas que envolvem o pré-processamento do texto, a criação de um modelo de linguagem, realiza o treinamento e por fim gera um novo texto.

Abaixo uma breve explicação sobre cada etapa:

* Pré-processamento do texto: Antes de alimentar o texto para o modelo, é necessário realizar o pré-processamento, que pode incluir tokenização, normalização, remoção de stopwords, entre outros. Nessa etapa, você também pode dividir o texto em sequências de tamanho fixo para alimentar o modelo durante o treinamento.
* Criação do modelo de linguagem: Para criar um modelo de linguagem em PyTorch, é possível usar a classe nn.Module. A arquitetura do modelo pode variar de acordo com a escolha do usuário, mas geralmente é composta por camadas de embeddings, camadas de convolução e/ou camadas de atenção.
* Treinamento do modelo: O treinamento do modelo envolve a definição da função de perda, a escolha do algoritmo de otimização, a definição da taxa de aprendizado e o treinamento em si, que envolve a alimentação das sequências de texto no modelo e o cálculo do gradiente para atualização dos pesos.
* Geração de um novo texto: Após o treinamento, é possível gerar um novo texto com base no modelo treinado. Para isso, basta alimentar uma sequência inicial no modelo e obter as previsões de saída para cada token seguinte até atingir um comprimento máximo ou um token de finalização.

Após isso, fazemos algumas tratativas para executarmos a similaridade dos textos, fazemos a Tokenização, removemos stop-words e lematizamos as palavras (limpeza dos textos). Criamos os vetores de cada texto e em seguida fazemos o cálculo de similaridade.

Veja na imagem abaixo o resultado, veja que o valor é bom (0,8185) conforme a imagem abaixo:

