**Introdução**

Caro usuário, seja bem-vindo à documentação do nosso aplicativo dedicado à democratização do acesso aos microdados oficiais do INEP, com ênfase nas análises e predições de comportamento baseadas na Teoria de Resposta ao Item (TRI). Este manifesto inicial visa orientá-lo na compreensão e utilização eficiente dessa plataforma modular, fornecendo informações detalhadas para garantir que seu trabalho não apenas alcance seus objetivos imediatos, mas também permaneça valioso ao longo do tempo.

**Objetivo da Plataforma:** Nosso aplicativo foi desenvolvido com o propósito central de facilitar o acesso e a manipulação de microdados oficiais fornecidos pelo INEP. Reconhecemos a importância desses dados para estudantes, dataminers e pesquisadores em suas análises, especialmente quando se trata da TRI, uma abordagem fundamental para compreender o desempenho em avaliações educacionais.

**Por que o nome é EduStarAmericas, se publicamente é enemaster.app?** A designação "Edu" é derivada de edtech (ou Eduarda, haha) e "Star" faz alusão à subdivisão da SpaceX chamada Starlink – uma empresa dedicada a democratizar o acesso à internet globalmente. A inclusão de "Américas" é uma referência à América Latina, completando assim nossa visão abrangente e compromisso com a universalização do acesso à educação e além.

**Modularidade e Flexibilidade:** A arquitetura modular da nossa plataforma foi concebida para fornecer flexibilidade e adaptabilidade às diversas necessidades de pesquisa. Cada módulo foi desenvolvido para operar de maneira independente, permitindo que você selecione e integre funcionalidades de acordo com seus requisitos específicos.

**Compromisso com a Democracia de Dados:** Acreditamos firmemente na democratização do acesso a informações valiosas. Portanto, buscamos criar uma interface intuitiva e documentação clara, tornando os microdados e as análises baseadas na TRI acessíveis a uma variedade de usuários, independentemente do nível de expertise.

**Estrutura da Documentação:** Esta documentação está organizada de maneira a guiá-lo desde a instalação inicial até a execução avançada de análises. Cada seção aborda aspectos específicos da plataforma, fornecendo descrições detalhadas, exemplos práticos e dicas úteis para garantir uma experiência consistente e eficiente.

**Atualizações e Suporte:** Como o código está parcialmente disponível, atualizarei essa plataforma em tempo livre, sempre aceitando sugestões da comunidade.

**ATENÇÃO:** essa documentação não visa ser um tutorial de como fazer, ou algo parecido. É um guia para desenvolvedores entenderem o que eu quis fazer nas linhas de código. Busque sempre alguém com conhecimento técnico ou **dedique tempo para estudar (na ordem)**: *Python, pandas, fpdf, genanki, numpy, R, mirt e Pycaret.*

**Agradecimentos especiais à Mateus F. Prado** – pioneiro em democratizar o acesso a TRI para o público comum, Mateus inspirou fortemente esse projeto.

Estamos confiantes de que, ao explorar esta documentação, você estará mais bem preparado para aproveitar todo o potencial dos microdados do INEP e da TRI.

Niedson Emanoel, 25/11/2023.

**Sumário**

[Visão Geral do Projeto: 3](#_Toc155977391)

[1. Gerador: Funcionalidades Principais: 4](#_Toc155977392)

[1.1 Gerador: Helper (putHabOnItem): 6](#_Toc155977393)

[1.2 Gerador: Helper Relatório de Itens - relatorio\_itens.py: 7](#_Toc155977394)

[2. MCi (Módulo comum integrado): 8](#_Toc155977395)

[2.1 MCi Listas e Flashcards por Habilidades: 8](#_Toc155977396)

[2.2 MCi Listas de Treino e Revisão: 10](#_Toc155977397)

[2.2.1 Parâmetros específicos das Matérias (Padrão experimental): 11](#_Toc155977398)

[2.3 MCi Clusterização: 12](#_Toc155977399)

[3. TRI 13](#_Toc155977400)

[3.1 Criação de Simulado (com 30 habilidades) 13](#_Toc155977401)

[3.2 Correção do Simulado 15](#_Toc155977402)

[Descrição Geral do Fluxo de Execução 15](#_Toc155977403)

[3.3 Correção TRI das listas por Habilidade 16](#_Toc155977404)

[3.4 Curva CCI 17](#_Toc155977405)

[4. Análises de Grupo 18](#_Toc155977406)

[5. Notebooks .ipynb 19](#_Toc155977407)

[5.1 Links Notebooks: 19](#_Toc155977408)

[6. Roadmap 20](#_Toc155977409)

[7. Agradecimentos 21](#_Toc155977410)

[8. Anexos 22](#_Toc155977411)

[8.1 Orientação (INEP via LAI) Curva CCI 22](#_Toc155977412)

[8.2 Exemplo de gráfico de curva CCI 23](#_Toc155977413)

[8.3 Exemplo de questão gerada pelo MCi 24](#_Toc155977414)

[8.4 Exemplo de Distribuição da Relação Acerto x Nota 25](#_Toc155977415)

[8.5 Licença de Uso 26](#_Toc155977416)

[9. Referências 27](#_Toc155977417)

# Visão Geral do Projeto:

O **EduStarAmericas** (***ESA)*** está dividido em **nove** módulos independentes:

1. **Gerador:**
   * Responsável pela geração personalizada de conjuntos de dados a partir dos microdados do ENEM.
2. **Ciências Humanas (MCi[[1]](#footnote-1)):**
   * Focado na análise de desempenho em disciplinas como História, Geografia, Filosofia e Sociologia.
3. **Linguagens (MCi1):**
   * Concentra-se nas disciplinas de Língua Portuguesa e Língua Estrangeira.
4. **Matemática (MCi1):**
   * Direcionado à análise do desempenho em Matemática e suas tecnologias.
5. **Natureza (MCi1):**
   * Módulo dedicado à análise de Ciências da Natureza.
6. **TRI (Teoria de Resposta ao Item):**
   * Um módulo exclusivo para a aplicação da Teoria de Resposta ao Item, proporcionando análises avançadas de comportamento e padrões de resposta dos participantes.
7. **Itens\_BNI:**
   * Catálogo de questões em .png, .gif e .txt de itens educacionais do INEP
8. **Análises de Grupo:**
   * Análises de todos os participantes pelo arquivo dos microdados.
9. **Notebooks *.ipynb*:**
   * Compilado independente dos módulos para uso específico.

**Requisitos de Sistema**

Para contribuir para o desenvolvimento do ***ESA***, as seguintes dependências são necessárias:

* Python 3.x e RStudio
* Bibliotecas específicas para processamento de dados (detalhadas em cada módulo)

**Estrutura do Projeto**

O projeto está organizado de maneira modular para facilitar a manutenção e expansão futura. Cada módulo reside em um diretório específico, contendo arquivos essenciais para sua operação.

**Principais Tecnologias Utilizadas**

* **NumPy:** Fundamenta a manipulação eficiente de operações matemáticas.
* **Pandas:** Acesso aos Itens .csv e manipulação
* **FPDF:** Criação dos itens em PDF
* **GENANKI:** Manipulação de Arquivos anki.

# ****1**. **Gerador**:** Funcionalidades Principais:

1. **Função tri\_3pl\_enem(theta, a, b, c)**
   * Descrição: Implementa a fórmula de resposta ao item no modelo de 3 parâmetros (3PL) utilizado no ENEM.
   * Parâmetros:
     + theta: Habilidade do candidato.
     + a, b, c: Parâmetros específicos da questão.
2. **Função imageApi(code)**
   * Descrição: Gera a URL de uma imagem contendo itens de prova com base no código fornecido.
   * Parâmetros:
     + code: Código da questão.
3. **Função ocrImage(code)**
   * Descrição: Realiza OCR em uma imagem de uma questão de prova, retornando o texto extraído.
   * Parâmetros:
     + code: Código da questão.
4. **Função find\_theta(a, b, c, targ)**
   * Descrição: Encontra o valor de theta que resulta em um percentil específico de acerto (targ).
   * Parâmetros:
     + a, b, c: Parâmetros específicos da questão.
     + targ: Percentil alvo de acerto.
5. **Função find\_quantile(c)**
   * Descrição: Calcula o percentil correspondente a uma habilidade específica (c).
6. **Função get\_prova\_string(ano, co\_prova)**
   * Descrição: Retorna uma string identificando o tipo de prova com base no ano e código da prova.
7. **Função thetaToCsv(provas, dfItens)**
   * Descrição: Calcula os valores de theta para diferentes percentis e realiza operações relacionadas, adicionando os resultados ao DataFrame.
8. **Função Make()**
   * Descrição: Realiza a leitura dos dados de diferentes anos do ENEM, aplica a análise de theta às questões e salva os resultados em arquivos CSV e Excel.

Dependências:

* cv2: OpenCV, para manipulação de imagens.
* numpy: Para operações numéricas.
* requests: Para requisições HTTP.
* io.BytesIO: Para manipulação de dados binários em memória.
* pandas: Para manipulação de dados tabulares.
* matplotlib.pyplot: Para visualizações gráficas.
* pytesseract: Para a utilização do OCR.
* os: Para operações de sistema.

Execução:

* Certifique-se de ter as dependências instaladas (cv2, numpy, requests, io, pandas, matplotlib, pytesseract, os).
* O arquivo executável do Tesseract (Tesseract.exe) deve estar localizado na pasta ./Tesseract/.
* O código gera arquivos CSV e Excel contendo os resultados da análise.

Notas Adicionais:

* O script está configurado para analisar microdados dos anos de 2014 a 2022.
* A função Make() executa todo o processo, desde a leitura dos dados até a geração dos resultados. Certifique-se de ter os arquivos de dados CSV disponíveis no mesmo diretório e os Arquivos no Banco de Itens

## ****1.1 Gerador: Helper (putHabOnItem):****

Este documento apresenta uma visão técnica do helper putHabOnItem, destinado a adicionar informações de habilidades ausentes nos microdados da prova do ENEM de 2015 PPL (solicitação via Lei de Acesso à Informação). O script, escrito em Python, utiliza a biblioteca pandas para manipulação de dados.

Funcionalidades Principais:

1. **Leitura de Dados:**
   * O script lê os dados da prova do ENEM de 2015 PPL de dois arquivos CSV (E\_ITENS\_PROVA\_2015.csv e lai2015process.xlsx).
2. **Preenchimento de Habilidades Ausentes:**
   * Identifica as questões que não possuem informação de habilidade (CO\_HABILIDADE igual a 31).
   * Busca as informações de habilidade correspondentes aos códigos de questões na planilha lai2015process.xlsx.
   * Preenche as habilidades ausentes no DataFrame principal (dItens2015) com as informações obtidas.
3. **Geração de Novos Arquivos:**
   * Gera um novo arquivo CSV (ITENS\_PROVA\_2015.csv) contendo os dados atualizados.
   * Gera um arquivo Excel (ITENS\_PROVA\_2015.xlsx) para facilitar a visualização dos dados.

Execução:

* Certifique-se de ter a biblioteca pandas instalada (pip install pandas).
* Execute o script, garantindo que os arquivos E\_ITENS\_PROVA\_2015.csv e lai2015process.xlsx estejam no mesmo diretório.
* O código atualiza as habilidades no DataFrame principal (dItens2015) e gera novos arquivos CSV e Excel.

Notas Adicionais:

* Devolva o arquivo corrigido para a pasta principal do gerador.
* Certifique-se de ter permissões de leitura e escrita no diretório onde os arquivos são salvos.

## **1.2 **Gerador: Helper**** Relatório de Itens - relatorio\_itens.py:

Este documento fornece uma visão técnica do script Python relatorio\_itens.py, que é responsável por gerar relatórios de itens sem correção comentada no banco de itens. O script utiliza diversas bibliotecas, como pandas, plotly, genanki, fpdf, requests, e outras, para realizar tarefas específicas relacionadas à análise de itens do ENEM.

Funcionalidades Principais:

1. **Geração de Relatórios em PDF:**
   * O script gera relatórios em PDF para questões sem correção comentada, agrupadas por área de conhecimento (Ciências Humanas, Linguagens, Ciências da Natureza e Matemática).
   * Inclui informações relevantes, como número da questão, ano, habilidade, proficiência estimada e link para a resolução no YouTube (quando disponível).
2. **Tratamento de Dados:**
   * Manipula dados provenientes de arquivos CSV para obter informações sobre questões sem correção comentada.
   * Organiza e processa os dados de forma a criar relatórios estruturados e visualmente atraentes.
3. **Geração de Imagens e Códigos de Barras:**
   * Baixa imagens das questões do ENEM a partir de URLs e as integra nos relatórios.
   * Gera códigos de barras para identificação das questões.
4. **Multithreading:**
   * Utiliza multithreading para processar áreas de conhecimento simultaneamente, otimizando o desempenho.

Execução:

* O script requer as bibliotecas pandas, plotly, genanki, fpdf, requests, Pillow, string, os, seaborn, matplotlib, wordcloud, barcode e concurrent.futures.
* Certifique-se de ter os diretórios corretos (como "images") e as imagens de fundo necessárias ("fundo.png" e "fundo2.png") no mesmo diretório do script.
* Execute o script utilizando Python 3.

Saída:

* O script gera arquivos PDF com relatórios para cada área de conhecimento sem correção comentada.
* Os arquivos de saída são nomeados como "Sem\_Correcao\_{AreaDeConhecimento}.pdf" (ex: Sem\_Correcao\_Humanas.pdf).

Notas Adicionais:

* O script lida com questões que não possuem correção comentada e organiza as informações de maneira clara e visualmente atraente nos relatórios.
* Certifique-se de ter permissões de leitura e escrita nos diretórios onde os arquivos são salvos.

# 2. MCi (Módulo comum integrado):

Este módulo automatiza a criação de flashcards Anki e um PDF contendo questões do Enem por Habilidades, incluindo imagens, respostas e links para resoluções no YouTube. Também tem como funcionalidade gerar listas de treino e de revisão para alunos. A documentação abordará as principais funções e processos envolvidos.

## 2.1 MCi Listas e Flashcards por Habilidades:

Estrutura do Código

Funções Principais:

**generate\_random\_number()**

* **Descrição:** Gera um número aleatório para identificar o baralho Anki.
* **Retorno:** Um número inteiro aleatório.

**remover\_caracteres\_invalidos(texto)**

* **Descrição:** Remove caracteres inválidos de uma string.
* **Entrada:** Uma string.
* **Retorno:** A string sem os caracteres inválidos.

**toYoutube(textPrompt)**

* **Descrição:** Cria um link de pesquisa no YouTube com base em um prompt de texto.
* **Entrada:** Uma string de texto.
* **Retorno:** Uma URL de pesquisa no YouTube.

**Capa(dItens)**

* **Descrição:** Gera uma nuvem de palavras a partir das palavras-chave associadas às questões.
* **Entrada:** Um DataFrame (dItens) contendo informações sobre as questões.
* **Saída:** Uma imagem PNG com a nuvem de palavras.

**questHab(dItens)**

* **Descrição:** Cria flashcards Anki, PDF com questões e um arquivo PDF com o gabarito.
* **Entrada:** Um DataFrame (dItens) contendo informações sobre as questões.
* **Saída:** O nome do arquivo PDF gerado.

Detalhes sobre a Geração de Flashcards

1. **Geração do Baralho e Modelo Anki:**
   * Um baralho e modelo Anki são criados usando a biblioteca genanki.
   * O modelo inclui campos como 'Questão', 'Resposta' e 'Imagem'.
2. **Download de Imagens:**
   * As imagens das questões e códigos de barras são baixadas da web.
3. **Criação dos Flashcards:**
   * Cada flashcard é um objeto genanki.Note com campos preenchidos com dados da questão.
4. **Criação do Pacote Anki:**
   * Um pacote Anki é gerado com os flashcards e as imagens.

Detalhes sobre a Geração do PDF

1. **Criação do PDF:**
   * Um PDF é criado usando a biblioteca fpdf.
2. **Adição de Páginas e Imagens:**
   * A primeira página inclui a nuvem de palavras.
   * Páginas subsequentes contêm questões e imagens.
   * A última página contém o gabarito em forma de tabela.
3. **Adição de Links:**
   * Links para resoluções no YouTube são incorporados ao PDF.

Observações

* Certifique-se de ter todas as bibliotecas necessárias instaladas antes de executar o código.
* O código utiliza o serviço de hospedagem niedsonemanoel.com.br para obter imagens; certifique-se de que o serviço está acessível.
* Pode ser necessário ajustar o código para execução em ambientes diferentes.

**Aviso:** Este módulo foi projetado para ser executado em ambientes que suportam as bibliotecas mencionadas. A execução em outros ambientes pode exigir modificações.

## 2.2 MCi Listas de Treino e Revisão:

Este módulo automatiza a criação de flashcards Anki e um PDF contendo questões de exames, incluindo imagens, respostas e links para resoluções no YouTube. O código é projetado para ser versátil, suportando diferentes áreas de conhecimento (Matemática, Humanas, Natureza e Linguagens).

Variáveis de Entrada

* **nome (str):** Nome do usuário ou identificação para o arquivo de saída.
* **nota (float):** Nota do usuário na área específica (Matemática, Humanas, Natureza ou Linguagens).

Ajustes de Sensibilidade do Algoritmo

Esses ajustes são utilizados para equilibrar as questões nos flashcards gerados.

* **aj165 (int):** Ajuste para a sensibilidade do algoritmo na probabilidade de ((c+1)/2)%.
* **aj265 (int):** Ajuste para a sensibilidade do algoritmo na probabilidade de ((c+1)/2)%.
* **aj399 (int):** Ajuste para a sensibilidade do algoritmo na probabilidade de 99%.
* **aj499 (int):** Ajuste para a sensibilidade do algoritmo na probabilidade de 99%.

Funções Principais

questionBalance\_65(name, nota\_CN, dfResult)

Gera um conjunto de flashcards Anki e um arquivo .pdf contendo questões equilibradas para a probabilidade de ((c+1)/2)%.

* **Parâmetros:**
  + **name (str):** Nome do arquivo de saída.
  + **nota\_CN (float):** Nota do usuário na área específica.
  + **dfResult (DataFrame):** DataFrame contendo os dados das questões.
* **Retorno:**
  + **str:** Nome do arquivo gerado.

questionBalance\_99(name, nota\_CN, dfResult)

Gera um conjunto de flashcards Anki e um arquivo .pdf contendo questões equilibradas para a probabilidade de 99%.

* **Parâmetros:**
  + **name (str):** Nome do arquivo de saída.
  + **nota\_CN (float):** Nota do usuário na área específica.
  + **dfResult (DataFrame):** DataFrame contendo os dados das questões.
* **Retorno:**
  + **str:** Nome do arquivo gerado.

Observações

* O código suporta diferentes áreas de conhecimento, representadas pelas siglas 'MT' (Matemática), 'CH' (Ciências Humanas), 'CN' (Ciências da Natureza) e 'LC' (Linguagens e Códigos). A área específica é definida pela variável prova.
* Certifique-se de adaptar as configurações específicas do Anki e do ambiente de execução, pois algumas funcionalidades podem variar dependendo do ambiente.

Explicação Parâmetros:

* aj165: Até que proficiência a lista de treino vai no máximo
  + Ex: Nota aluno 600, aj165: 170, máximo proficiência = 770 (600+170)
* aj265: Até que proficiência a lista de treino vai no mínimo
  + Ex: Nota aluno 600, aj265: 50, mínimo proficiência = 550 (600-50)
* **Range lista de Treino neste exemplo: questões de 550 até 770**
* aj399: Até que proficiência a lista de revisão vai no máximo
  + Ex: Nota aluno 600, aj399: 170, máximo proficiência = 770 (600+170)
* aj499: Até que proficiência a lista de treino vai no mínimo
  + Ex: Nota aluno 600, aj499: 40, máximo proficiência = 560 (600-40)
* **Range lista de Revisão neste exemplo: questões de theta\_99 de 560 até 770**

### 2.2.1 Parâmetros específicos das Matérias (Padrão experimental):

Linguagens

* **aj165 (int):** **170**
* **aj265 (int):** **30**
* **aj399 (int):** **120**
* **aj499 (int):** **20**

Humanas

* **aj165 (int):** **180**
* **aj265 (int):** **40**
* **aj399 (int):** **130**
* **aj499 (int):** **20**

Natureza

* **aj165 (int):** **150**
* **aj265 (int):** **25**
* **aj399 (int):** **110**
* **aj499 (int):** **10**

Matemática

* **aj165 (int):** **200**
* **aj265 (int):** **30**
* **aj399 (int):** **130**
* **aj499 (int):** **10**

**\*Obtido experimentalmente.**

## 2.3 MCi Clusterização:

Este módulo automatiza o processo de clusterização de questões de exames, utilizando técnicas de aprendizado não supervisionado (clustering) para agrupar questões semelhantes com base em uma característica específica. O código é projetado para ser versátil, suportando diferentes áreas de conhecimento (Matemática, Humanas, Natureza e Linguagens).

**Variáveis de Entrada**

* **Disciplina (str):** Área de conhecimento da disciplina a ser analisada (Matemática, Humanas, Natureza ou Linguagens).
* **colunaCluster (str):** Característica específica utilizada para a clusterização (e.g., 'theta\_065', 'theta\_080', 'theta\_099', etc.).

Funções Principais

cluster\_questions(Disciplina, colunaCluster)

Realiza a clusterização de questões e exporta os resultados para arquivos CSV.

* **Parâmetros:**
  + **Disciplina (str):** Área de conhecimento da disciplina a ser analisada.
  + **colunaCluster (str):** Característica específica utilizada para a clusterização.
* **Saída:**
  + Arquivos CSV exportados para cada cluster.
  + Arquivo CSV consolidado com as atribuições de cluster.

Observações

* A variável colunaCluster define a característica utilizada para a clusterização, podendo ser a proficiência em diferentes percentis ou outras características relevantes.
* O código utiliza a biblioteca PyCaret para simplificar o processo de clustering e avaliação de modelos.
* A clusterização é útil para identificar padrões e semelhanças entre questões, possibilitando uma análise mais aprofundada do conjunto de questões disponíveis.
* O código exporta arquivos CSV separados para cada cluster, permitindo uma análise individual de cada grupo de questões.
* Certifique-se de adaptar as configurações específicas do ambiente de execução, como o caminho de exportação de arquivos (os.chdir), ao utilizar o código fora do ambiente Colab.

# 3. TRI

## 3.1 Criação de Simulado (com 30 habilidades)

Este módulo tem como objetivo automatizar a geração de simulados para diferentes disciplinas do ENEM (Matemática, Humanas, Natureza ou Linguagens). O código extrai questões de acordo com critérios específicos, cria um arquivo CSV com os dados do simulado e um arquivo PDF com as questões e o cartão resposta.

Descrição Geral do Fluxo de Execução

1. **Carregamento dos Itens:**
   * Leitura de um arquivo CSV contendo itens previamente ordenados por triagem, disponibilizado em um repositório online.
2. **Filtragem de Itens:**
   * Seleção de itens específicos para a disciplina escolhida.
   * Filtragem de itens de habilidades específicas (entre 1 e 30).
   * Exclusão de itens abandonados.
3. **Seleção de Questões Únicas e Repetidas:**
   * Escolha de uma questão única para cada habilidade (1 a 30).
   * Seleção de 12 questões adicionais permitindo repetições, com no máximo 3 repetições por habilidade.
4. **Verificação de Todas as Habilidades:**
   * Verificação se todas as 30 habilidades estão presentes no conjunto de questões.
   * Se faltarem habilidades, seleção de itens adicionais para as habilidades ausentes.
5. **Análises Estatísticas do Simulado:**
   * Cálculo da proficiência da questão mais difícil e mais fácil.
   * Geração de gráficos que mostram a distribuição de questões por prova, ano e gabarito.
6. **Exportação de Dados:**
   * Criação de um arquivo CSV contendo os dados do simulado.
   * Exportação do simulado em formato CSV.
7. **Geração do Cartão Resposta em PDF:**
   * Criação de um arquivo PDF contendo as questões do simulado e um cartão resposta para o aluno.

Detalhes Técnicos

Dependências

* O código faz uso das seguintes bibliotecas Python:
  + numpy, pandas: Manipulação e análise de dados.
  + matplotlib, seaborn: Visualização de dados e criação de gráficos.
  + fpdf: Geração de arquivos PDF.
  + requests: Requisições HTTP para download de imagens.
  + PIL: Processamento de imagens.
  + os: Manipulação de sistema de arquivos.

Configurações

* Certifique-se de que as bibliotecas mencionadas acima estão instaladas no ambiente de execução.
* O código contém URLs para imagens e dados externos, e a execução pode depender da conectividade com a internet.

Utilização

* Para gerar um simulado, basta chamar a função gera\_simulado(Disciplina), onde Disciplina é uma string que pode ser "MT", "CH", "CN" ou "LC".
* A função retornará:
  + Um arquivo CSV contendo os dados do simulado.
  + Um arquivo PDF com as questões do simulado e um cartão resposta.

## 3.2 Correção do Simulado

Este módulo realiza a correção automática de um simulado TRI (Teoria de Resposta ao Item) gerado anteriormente. O código calcula a nota TRI para um candidato com base nas respostas fornecidas e nos parâmetros dos itens.

## Descrição Geral do Fluxo de Execução

1. **Carregamento dos Dados do Simulado:**
   * Leitura do arquivo CSV contendo os dados do simulado gerado anteriormente.
   * Exibição de instruções para o usuário fazer o upload do arquivo CSV gerado.
2. **Definição de Funções Matemáticas:**
   * calcular\_probabilidade(theta, a, b, c): Calcula a probabilidade de uma resposta correta dada a proficiência do candidato e os parâmetros do item.
   * calcular\_verossimilhanca(theta, a, b, c, x): Calcula a verossimilhança da resposta do candidato para um conjunto de itens.
   * encontrar\_theta\_max(a, b, c, x): Encontra o valor de proficiência (theta) que maximiza a verossimilhança para o candidato.
3. **Obtenção das Respostas do Candidato:**
   * O usuário fornece as respostas para cada questão do simulado.
   * As respostas são validadas para garantir que são A, B, C, D, E ou X (questão anulada).
4. **Cálculo da Nota TRI e Estatísticas:**
   * O código calcula a nota TRI para o candidato usando os parâmetros dos itens e as respostas fornecidas.
   * Exibição da pontuação TRI do candidato.
   * Exibição do número de acertos.
5. **Exibição de Parâmetros dos Itens e Respostas do Candidato:**
   * Exibição dos parâmetros dos itens (NU\_PARAM\_A, NU\_PARAM\_B, NU\_PARAM\_C).
   * Exibição da matriz de respostas do candidato.

Detalhes Técnicos

Dependências

* O código faz uso das seguintes bibliotecas Python:
  + numpy, pandas: Manipulação e análise de dados.
  + scipy: Cálculos científicos, incluindo otimização.

Observações

* Certifique-se de ter as permissões necessárias para realizar o upload de arquivos no ambiente de execução.
* O código utiliza otimização para encontrar o valor de proficiência (theta) que maximiza a verossimilhança do candidato.

## 3.3 Correção TRI das listas por Habilidade

Este módulo realiza a análise de itens específicos relacionados a uma habilidade escolhida em uma disciplina do ENEM. O código utiliza a Teoria de Resposta ao Item (TRI) para calcular a nota TRI de um candidato com base nas respostas fornecidas e nos parâmetros dos itens associados a uma habilidade específica.

Descrição Geral do Fluxo de Execução

1. **Carregamento dos Dados dos Itens:**
   * Leitura dos dados dos itens a partir de um arquivo CSV hospedado online.
   * Filtro dos itens com base na disciplina e habilidade escolhidas.
2. **Definição de Funções Matemáticas:**
   * calcular\_probabilidade(theta, a, b, c): Calcula a probabilidade de uma resposta correta dada a proficiência do candidato e os parâmetros do item.
   * calcular\_verossimilhanca(theta, a, b, c, x): Calcula a verossimilhança da resposta do candidato para um conjunto de itens.
   * encontrar\_theta\_max(a, b, c, x): Encontra o valor de proficiência (theta) que maximiza a verossimilhança para o candidato.
3. **Obtenção das Respostas do Candidato:**
   * O usuário fornece as respostas (1 para correta, 0 para incorreta) para cada questão associada à habilidade escolhida.
4. **Cálculo da Nota TRI e Exibição:**
   * O código calcula a nota TRI para o candidato usando os parâmetros dos itens e as respostas fornecidas.
   * Exibição da pontuação TRI do candidato para a habilidade escolhida.

Detalhes Técnicos

Dependências

* O código faz uso das seguintes bibliotecas Python:
  + numpy, pandas: Manipulação e análise de dados.
  + scipy: Cálculos científicos, incluindo otimização.

Utilização

1. Execute o código no ambiente Colab.
2. Escolha a disciplina e a habilidade desejadas.
3. Forneça as respostas do candidato para as questões associadas à habilidade escolhida.

## 3.4 Curva CCI

Este módulo gera a Curva Característica do Item (CCI) com base nos parâmetros da Teoria de Resposta ao Item (TRI). A CCI representa a relação entre a probabilidade de um candidato acertar uma questão e o nível de proficiência do candidato.

Descrição Geral do Fluxo de Execução

1. **Definição dos Parâmetros da TRI:**
   * O usuário insere os parâmetros a, b e c, que representam, respectivamente, os parâmetros de discriminação, dificuldade e acerto ao acaso.
2. **Cálculo da Probabilidade de Acerto:**
   * Utilizando os parâmetros fornecidos, calcula a probabilidade de um candidato acertar uma questão para diferentes níveis de proficiência (theta).
3. **Geração do Gráfico CCI:**
   * Gera o gráfico da CCI, representando a probabilidade de acerto em função do nível de proficiência.
   * Destaca a proficiência que resulta em uma probabilidade pré-determinada (usualmente 50%).

Detalhes Técnicos

Dependências

* O código faz uso das seguintes bibliotecas Python:
  + numpy, pandas: Manipulação e análise de dados.
  + matplotlib.pyplot: Geração de gráficos.
  + scipy: Cálculos científicos, incluindo otimização.

Utilização

1. Execute o código no ambiente Colab.
2. Insira os valores dos parâmetros a, b e c.
3. O gráfico da CCI será gerado, mostrando a relação entre a probabilidade de acerto e o nível de proficiência.

Observações

* O código utiliza uma abordagem de busca binária para encontrar o nível de proficiência que resulta na probabilidade desejada.
* A CCI é uma ferramenta importante na análise de itens em avaliações educacionais, fornecendo insights sobre a sensibilidade do item à habilidade do candidato.

Para informações mais detalhadas sobre a implementação de funções específicas, consulte o código-fonte disponível no link original do Colab.

# 4. Análises de Grupo

Este módulo automatiza a análise de dados do Exame Nacional do Ensino Médio (ENEM) para uma disciplina específica (Matemática, Humanas, Natureza ou Linguagens) e um ano escolhido. O código extrai os microdados do ENEM do INEP, realiza análises estatísticas, e exporta os resultados para um arquivo Excel e um arquivo de texto.

Variáveis de Entrada

* **ANO (str):** Ano do ENEM a ser analisado (2014 a 2023).
* **Disciplina (str):** Disciplina específica do ENEM a ser analisada (Matemática, Humanas, Natureza ou Linguagens).
* **MICRODADOS\_2014 a MICRODADOS\_2023 (str):** Links para os arquivos ZIP contendo os microdados do ENEM para cada ano.

Funções Principais

analise\_enem(ANO, Disciplina, MICRODADOS)

Realiza a análise dos microdados do ENEM, gerando estatísticas e exportando os resultados para arquivos Excel e texto.

* **Parâmetros:**
  + **ANO (str):** Ano do ENEM a ser analisado.
  + **Disciplina (str):** Disciplina específica do ENEM a ser analisada.
  + **MICRODADOS (str):** Link para o arquivo ZIP contendo os microdados do ENEM para o ano especificado.
* **Saída:**
  + Arquivo Excel contendo estatísticas sobre a relação entre acertos e notas.
  + Arquivo de texto contendo a distribuição percentual de notas em intervalos de 100 pontos.

Observações

* O código utiliza os microdados do ENEM disponibilizados pelo INEP para o ano escolhido.
* Realiza a filtragem dos dados para considerar apenas participantes presentes e com notas não nulas na disciplina específica.
* Calcula a quantidade de alunos que tiraram determinada nota ou mais na disciplina, em percentual total de participantes e em percentual considerando apenas alunos válidos.
* Gera estatísticas sobre a relação entre a quantidade de acertos e as notas, exportando essas informações para um arquivo Excel.
* Exporta a distribuição percentual de notas em intervalos de 100 pontos para um arquivo de texto.

# 5. Notebooks .ipynb

Confidencial até a liberação do algoritmo publicamente em 2025.

Porque confidencial? Os notebooks são a expressão modular deste código, qualquer pessoa até mesmo partindo de um celular consegue utiliza-los sem precisar de muito conhecimento técnico. A liberação se dará quando o algoritmo for liberado para pesquisadores e professores em momento oportuno.

## 5.1 Links Notebooks:

* [Cluster / Questions](https://colab.research.google.com/drive/10Bxwcm--fQch8mWGAJi2BrJOvNz1YQRH?usp=sharing)
* [Gráfico CCI/TRI](https://colab.research.google.com/drive/1BTx3FMLYG1OOH-idh4-vsD3cmTcy9mKs?usp=sharing)
* [Gerador](https://colab.research.google.com/drive/1ESEkdRC3HvmOa1_OUENmdy2bWqbA7Gy-?usp=sharing)
* [CriarSimulado](https://colab.research.google.com/drive/1f7lwuE2xHh4IzEwPxEiz7yy2aM7n49WQ?usp=sharing)
* [Hab\_Anki/Pdf](https://colab.research.google.com/drive/1gSjRGEjdQ6pV3Gb7qK_Og_JtJnDIbbY7?usp=sharing)
* [Microdados Acerto x Nota](https://colab.research.google.com/drive/1iIEQRwcPj9e62pCQNE38MuihhET3me8N?usp=sharing)
* [TRI/Habilidades](https://colab.research.google.com/drive/1kf8xAUPShXwxJZ8xtIPM7857MKxYAhUy?usp=sharing)
* [Corretor TRI/Simulados](https://colab.research.google.com/drive/1p2ELduE5PzsSsrz8E-leAgdwEewjX9Om?usp=sharing)
* [Github](https://github.com/NiedsonEmanoel/NiedsonEmanoel/tree/main/enem)

\*Se você recebeu o arquivo com os Links, pode utilizar em suas análises (Ctrl+enter), mas não divulgue ainda.

# 6. Roadmap

Aqui descrevemos nossas metas durante a existência do ESA e marcamos em OK (verde as concluídas, em ABAN (vermelho) as abandonadas e em DOING as que estão sendo realizadas

* **2023**

o   **Gerar gráfico CCI**

o   **Perfil de Nota x Renda:  demanda alto processamento e dados são pouco úteis**

o   **Estimar nota TRI por Regressão Linear: não há demanda.**

o   **Indicar proficiência do Item ((c+1)/2)**

o   **Gerar PDF de Habilidades ordenas pela proficiência**

o   **Algoritmo gamificado de realização de simulados/mini simulados**

o   **Calcular a nota TRI em Python.**

o   **Indicar questões pela probabilidade de 99% e ((c+1)/2) de acerto**

o   **Leitura óptica (OCR dos Itens)**

o   **Criar simulado autoral e corrigir via máxima verossimilhança: funciona, precisa de ajustes**

o   **Clusterizar alunos, questões e desempenhos**

o   **Buscar resolução pela OCR dos itens no Youtube**

o   **Resolução comentada dos Itens**

o   **Mínimo de 5 anos de prova no banco**

o   **Interfaces visuais e BetaTest público**

o   **Contar acertos e tri nos microdados de qualquer ano**

**Saldo de 4096 questões no Banco de Itens. (3 injeções)**

**I1: 2021, 2020 e 2019 (fev)**

**I2: 2022, 2018 e 2017 (ago)**

**I3: 2016, 2015 e 2014 (dez)\***

·        **2024**

o   **Publicização dos dados via página @enemaster.app**

o   **Inserir dados ENCCEJA e 2023 no banco de itens**

o   **Estimar parâmetros TRI para questões de vestibulares e utiliza-las de maneira equivalente aos do ENEM.**

* **2025**
  + Liberação do código fonte do EAS para pesquisadores, professores e interessados sob a licença MIT.

# 7. Agradecimentos

Quero dedicar uma expressão sincera de gratidão aos professores e alunos extraordinários que foram peças-chave na concretização deste projeto.

**Professores:**

* Mateus F. Prado
* Nathália Lopes
* Hassã
* Thales Silva
* Diego Rubinakes
* Selma Calgoroto

**Alunos:**

* Henrique Angelim
* Daniel Rubens
* Eduardo Santos
* Maria Leite
* Wenderson Santos
* Luiggi Crivelenti
* Isaque Araújo
* Maria Eduarda Bittencourt
* João Pedro Lopes
* Gabriel Carvalhal
* Érik Augusto

Cada um de vocês trouxe uma perspectiva única, expertise e entusiasmo que elevaram este projeto a novos patamares. Agradeço sinceramente por sua incrível colaboração e pelo compromisso.

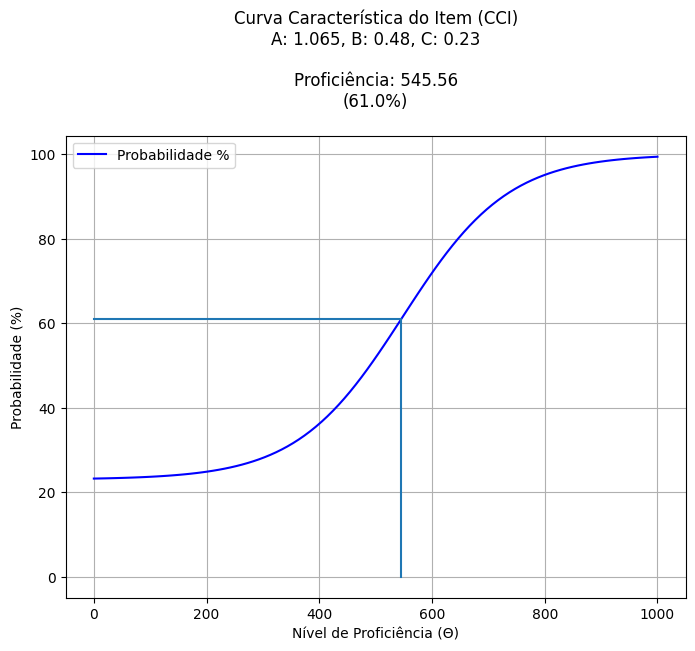
Seu impacto positivo permanecerá como parte integral da jornada deste projeto. Obrigado por serem uma fonte de inspiração e por compartilharem seus conhecimentos e talentos.

# 8. Anexos

## 8.1 Orientação (INEP via LAI) Curva CCI

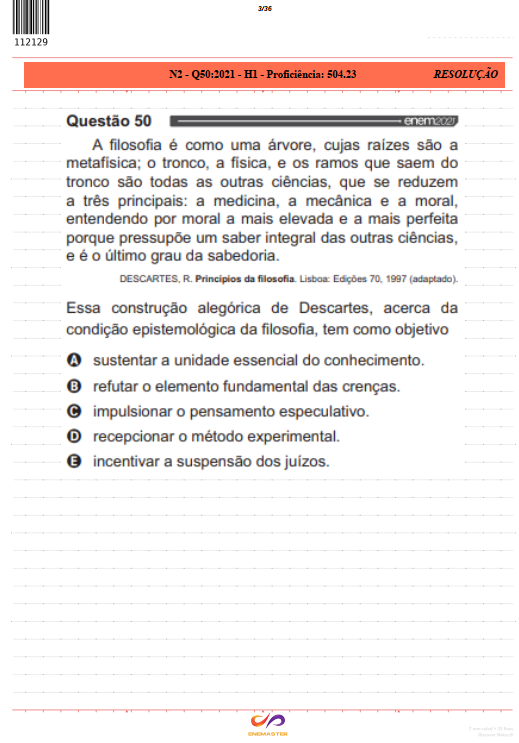


## 8.2 Exemplo de gráfico de curva CCI



Fonte: O autor, gerado pela ESA no módulo notebooks/cci.py

## 8.3 Exemplo de questão gerada pelo MCi



Fonte: O autor, gerada pelo algoritmo MCi – anki\_hab.py, para Humanas (CH) e H1

## 8.4 Exemplo de Distribuição da Relação Acerto x Nota



\*A saída é referente ao arquivo excel.

## 8.5 Licença de Uso

Copyright (c) 2024 N.E.A, Brito.

Permission is hereby granted, partially free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to handle the Software, with the rights to use without selling copies of the Software, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES, OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT, OR OTHERWISE, ARISING FROM, OUT OF, OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

RECIPIENTS MUST, WHENEVER USING THIS ALGORITHM, WHETHER GRATUITOUSLY OR FOR A FEE, MAINTAIN SCHOLARSHIP DISTRIBUTION PROGRAMS OR EQUIVALENT FOR LOW-INCOME STUDENTS. THEY MUST ALWAYS CITE (N.E.A, BRITO) AND USE THE TERMINOLOGY "ENEMASTER.APP," AND ARE PROHIBITED FROM USING DEVELOPMENT NAMES (ESA) EDUSTARAMERICAS.

Direitos autorais (c) 2024 N.E.A, Brito.

É concedida permissão, de forma parcialmente gratuita, a qualquer pessoa que obtenha uma cópia deste software e dos arquivos de documentação associados (o "Software"), para lidar com o Software, com os direitos de usar SEM vender cópias do Software, sujeitas às seguintes condições:

O aviso de direitos autorais acima e este aviso de permissão devem ser incluídos em todas as cópias ou partes substanciais do Software.

O SOFTWARE É FORNECIDO "COMO ESTÁ", SEM GARANTIA DE QUALQUER TIPO, EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS NÃO SE LIMITANDO ÀS GARANTIAS DE COMERCIALIZAÇÃO, ADEQUAÇÃO A UM DETERMINADO PROPÓSITO E NÃO VIOLAÇÃO. EM NENHUM CASO OS AUTORES OU TITULARES DE DIREITOS AUTORAIS SERÃO RESPONSÁVEIS POR QUALQUER RECLAMAÇÃO, DANOS OU OUTRA RESPONSABILIDADE, QUER EM UMA AÇÃO DE CONTRATO, DELITO OU OUTRA FORMA, DECORRENTE DE, OU EM CONEXÃO COM O SOFTWARE OU O USO OU OUTRAS NEGOCIAÇÕES NO SOFTWARE.

OS REDISTRIBUIDORES DEVEM SEMPRE QUE UTILIZAR ESSE ALGORITMO, DE FORMA GRATÚITA OU PAGA MANTER PROGRAMAS DE DISTRIBUIÇÃO DE BOLSAS DE ESTUDO OU EQUIVALENTE PARA ALUNOS DE BAIXA RENDA.

SEMPRE DEVERÃO CITAR (N.E.A, BRITO) E UTILIZAR A TERMINOLOGIA ENEMASTER.APP, SENDO VEDADO OS NOMES DE DESENVOLVIMENTO (ESA) EDUSTARAMERICAS

**Software em processo de registro no INPI.**

# 9. Referências

Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (Inep). (2011). *Nota técnica: procedimento de cálculo das notas do Enem.* Acesso em 2023`, disponível em GOV.BR: https://download.inep.gov.br/educacao\_basica/enem/nota\_tecnica/2011/nota\_tecnica\_procedimento\_de\_calculo\_das\_notas\_enem\_2.pdf

INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA. (12 de 01 de 2024). *Microdados do Enem (2014 a 2022).* Fonte: Microdados Enem: https://www.gov.br/inep/pt-br/acesso-a-informacao/dados-abertos/microdados/enem

1. **MCi**: ***Módulo comum integrado:*** Áreas que compartilham o mesmo código, porém com mínimas alterações de parâmetros para separar os estágios de desenvolvimento. [↑](#footnote-ref-1)