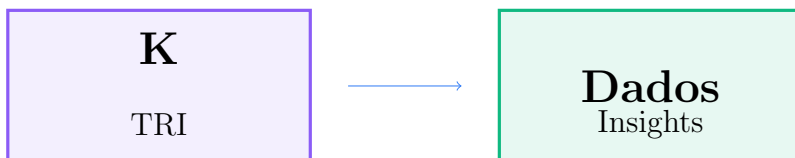


KAIROS

Sistema de Analise de Avaliaco'es com TRI

Manual Completo e Documentacao Tecnica



Desenvolvedor: Mauricio A. Ribeiro

Versao: 1.0.0

Data: 27 de dezembro de 2025

Licenca: MIT

Sistema de Analise Psicometrica para Educacao

Este documento fornece toda a documentacao necessaria para instalar, configurar e utilizar o sistema KAIROS. Inclui guias tecnicos, manuais de usuario e referencias de desenvolvimento.

Conteúdo

1	Introducao ao Sistema KAIROS	5
1.1	O que e o KAIROS?	5
1.2	Proposta de Valor	5
1.3	Publico-Alvo	5
1.4	Diferenciais Competitivos	5
1.5	Arquitetura Conceitual	6
1.6	Beneficios Esperados	6
2	Instalacao e Configuracao	7
2.1	Requisitos do Sistema	7
2.1.1	Requisitos Minimos	7
2.1.2	Requisitos Recomendados	7
2.2	Instalacao Passo a Passo	8
2.2.1	1. Preparacao do Ambiente	8
2.2.2	2. Criacao do Ambiente Virtual (Recomendado)	8
2.2.3	3. Instalacao das Dependencias	9
2.2.4	4. Verificacao da Instalacao	9
2.3	Configuracao do Ambiente	10
2.3.1	Variaveis de Ambiente (Opcional)	10
2.3.2	Configuracao do Streamlit	10
2.4	Primeira Execucao	11
2.4.1	Execucao Basica	11
2.4.2	Opcoes de Execucao Avancadas	11
2.4.3	Verificacao do Funcionamento	11
2.5	Solucao de Problemas de Instalacao	11
2.5.1	Problemas Comuns e Solucoes	11
2.5.2	Teste de Instalacao Completa	11
3	Manual do Usuario	14
3.1	Primeiros Passos	14
3.1.1	Tela Inicial	15
3.1.2	Estrutura da Interface	15
3.2	Configuracao do Gabarito	15
3.2.1	Importancia do Gabarito	15
3.2.2	Formatos Aceitos	16
3.2.3	Processo de Configuracao	16
3.3	Adicao de Alunos	16
3.3.1	Comparacao entre Modos de Entrada	16
3.3.2	Modo Manual: Guia Detalhado	17

3.3.3	Modo CSV: Preparacao do Arquivo	17
3.3.4	Script de Preparacao de CSV	18
3.4	Dashboard Principal	20
3.4.1	Abas e Funcionalidades	20
3.4.2	Metricas Principais	20
3.4.3	Panorama Geral da Turma	21
3.4.4	Analise Multidimensional das Questoes	21
3.5	Analise Individual	22
3.5.1	Selecao e Visualizacao	22
3.5.2	Metricas Individuais	22
3.5.3	Grafico de Desempenho por Questao	23
3.5.4	Recomendacoes Pedagogicas	23
3.6	Sistema de Tutores	23
3.6.1	Fundamentacao Teorica	23
3.6.2	criterios de Selecao	23
3.6.3	Calculo do Score de Tutor	24
3.6.4	Formacao de Grupos de Tutoria	25
3.6.5	Plano de Acao para Tutoria	25
3.7	Exportacao de Dados	26
3.7.1	Formatos Disponiveis	26
3.7.2	Estrutura dos Relatorios	26
3.7.3	Scripts de Processamento Pos-Exportacao	26
4	Teoria de Resposta ao Item (TRI)	28
4.1	Fundamentos da TRI	28
4.1.1	Diferencas em Relacao a TCT	29
4.1.2	Vantagens da TRI	29
4.2	Modelo Logístico de 2 Parâmetros (2PL)	29
4.2.1	Interpretacao Grafica	30
4.3	Parâmetros da TRI	30
4.3.1	Dificuldade (b)	30
4.3.2	Discriminacao (a)	30
4.3.3	Proficiencia ()	30
4.4	Estimacao dos Parâmetros	31
4.4.1	Metodos de Estimacao	31
4.4.2	Implementacao noCodigo	32
4.5	Indices de Qualidade	33
4.5.1	Confiabilidade do Teste	33
4.5.2	Indice de Discriminacao	33
4.5.3	Correlacao Bisserial Pontual	34
4.6	Aplicacoes Praticas da TRI	34

4.6.1	Identificacao de Itens Problematicos	34
4.6.2	Interpretacao Pedagogica	34
4.6.3	Formacao de Grupos de Nivel	34
4.7	Limitacoes e Consideracoes	35
4.7.1	Premissas da TRI	35
4.7.2	Limitacoes do Modelo Simplificado	35
5	Referencia Tecnica	36
5.1	Arquitetura do Sistema	36
5.1.1	Diagrama de Componentes	36
5.1.2	Fluxo de Dados	36
5.2	Estrutura doCodigo	37
5.2.1	Organizacao do Arquivo Principal	37
5.2.2	Gerenciamento de Estado	38
5.3	Otimizacao de Performance	38
5.3.1	Uso de Cache	38
5.3.2	Otimizacoes Especificas	38
5.4	Extensibilidade	39
5.4.1	Adicionando Novas Funcionalidades	39
5.4.2	Integracao com Outros Sistemas	39
5.5	Solucao de Problemas Tecnicos	39
5.5.1	Erros Comuns e Solucoes	39
5.5.2	Logs e Diagnostico	39
5.6	Seguranca e Privacidade	40
5.6.1	Consideracoes de Seguranca	40
5.6.2	Recomendacoes de Uso Seguro	40
A	Apendices	41
A.1	A: Glossario de Termos	41
A.2	B: Referencias Bibliograficas	41
A.3	C: Recursos Adicionais	42
A.3.1	Sites e Documentacao	42
A.3.2	Livros Recomendados	42
A.3.3	Cursos e Formacoes	42
A.4	D: Codigos de Exemplo	43
A.4.1	Script de Backup Automatico	43
A.4.2	Script de Monitoramento	44
A.5	E: Historico de Versoes	45
A.6	F: Contato e Suporte	46
A.6.1	Informacoes de Contato	46
A.6.2	Politica de Suporte	46

A.6.3 Como Reportar Problemas 46

A.6.4 Contribuindo com o Projeto 47

1 Introducao ao Sistema KAIROS

1.1 O que e o KAIROS?

O **KAIROS** e um sistema de analise psicometrica avancada desenvolvido especificamente para educadores, instituicoes de ensino e pesquisadores em educacao. O nome "KAIROS" vem do conceito grego que representa o "momento oportuno" ou "momento certo", refletindo a proposta do sistema: fornecer analises no momento certo para tomar decisoes pedagogicas informadas.

[colback=kairos-blue!5, colframe=kairos-blue!80, title=Origem do Nome, fonttitle=] **KAIROS () na mitologia grega representa o deus do "momento oportuno". No contexto educacional, significa fornecer analises precisas no momento certo para intervencoes pedagogicas efetivas.**

1.2 Proposta de Valor

O sistema KAIROS foi desenvolvido com os seguintes objetivos principais:

1. **Democratizar a analise psicometrica:** Tornar acessivel a professores e educadores ferramentas avancadas que antes estavam disponiveis apenas para especialistas.
2. **Fornecer insights acionaveis:** Ir alem de simples estatisticas e oferecer recomendacoes pedagogicas praticas e personalizadas.
3. **Integrar tecnologia e pedagogia:** Combinar metodos estatisticos avancados (TRI) com conhecimentos pedagogicos para criar uma ferramenta realmente util para educadores.
4. **Facilitar a tomada de decisoes:** Fornecer dados claros e interpretaveis que ajudem na tomada de decisoes sobre ensino, aprendizagem e avaliacao.

1.3 Publico-Alvo

O sistema KAIROS foi projetado para atender diferentes perfis de usuarios:

1.4 Diferenciais Competitivos

O KAIROS se diferencia de outras ferramentas disponiveis no mercado por:

1. **Interface amigavel:** Desenvolvida em Streamlit, oferece uma experiencia de usuario intuitiva e acessivel mesmo para nao-tecnicos.

Perfil	Necessidades Especificas
Professores	Analisar desempenho de turmas, identificar dificuldades individuais, formar grupos de estudo, personalizar ensino.
Coordenadores Pedagogicos	Monitorar desempenho institucional, avaliar qualidade de avaliacoes, planejar intervencoes, analisar dados agregados.
Pesquisadores em Educacao	Coletar dados para pesquisas, analisar propriedades psicometricas, validar instrumentos de avaliacao.
Instituicoes de Ensino	Avaliar eficacia de metodologias, monitorar qualidade do ensino, gerar relatorios institucionais.
Estudantes de Pos-Graduacao	Aprender sobre analise psicometrica, praticar com dados reais, desenvolver pesquisas academicas.

Tabela 1: Publico-alvo do sistema KAIROS

2. **Integracao completa:** Combina entrada de dados, analise, visualizacao e exportacao em uma unica plataforma.
3. **Sistema de tutores:** Funcionalidade unica que identifica automaticamente alunos com potencial para ajudar colegas.
4. **Multiplos formatos de exportacao:** Oferece flexibilidade para diferentes necessidades de uso dos dados.
5. **Gratuito e open-source:** Nao ha custos de licenca e o codigo esta disponivel para modificacoes e melhorias.

1.5 Arquitetura Conceitual

O sistema KAIROS opera com base em tres pilares fundamentais:

1.6 Beneficios Esperados

Ao utilizar o sistema KAIROS, os usuarios podem esperar os seguintes beneficios:

[colback=kairos-green!5, colframe=kairos-green!80, title=Beneficios Tangiveis, fonttitle=]

- **Economia de tempo:** Automatizacao de analises que manualmente levariam horas ou dias.
- **Decisoes mais informadas:** Baseadas em dados psicometricos solidos.

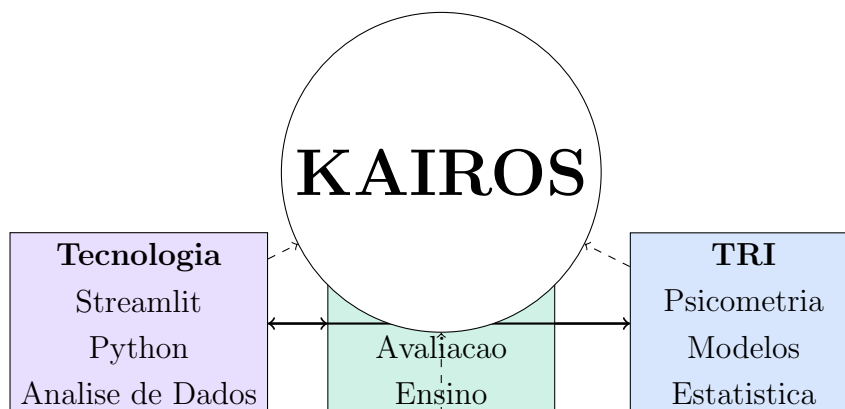


Figura 1: Arquitetura conceitual do sistema KAIROS

- **Personalizacao do ensino:** Identificacao precisa de necessidades individuais.
- **Melhoria da qualidade das avaliacoes:** Feedback sobre itens e questoes.
- **Documentacao completa:** Relatorios prontos para compartilhamento.

2 Instalacao e Configuracao

2.1 Requisitos do Sistema

Antes de iniciar a instalacao, verifique se seu sistema atende aos seguintes requisitos:

2.1.1 Requisitos Minimicos

Componente	Especificacao Minima
Sistema Operacional	Windows 10, macOS 10.14+, Ubuntu 18.04+
Processador	Dual-core 2.0 GHz
Memoria RAM	4 GB
Armazenamento	500 MB livres
Python	Versao 3.8 ou superior
Conexao de Internet	Para instalacao inicial (uso offline posterior)
Navegador Web	Chrome 80+, Firefox 75+, Safari 13+

Tabela 2: Requisitos minimicos do sistema

2.1.2 Requisitos Recomendados

[colback=red!5, colframe=red!80, title=Atencao com Versoes do Python, fonttitle=] **Al-
gumas bibliotecas podem ter incompatibilidades com versoes muito recentes
do Python. Recomenda-se utilizar Python 3.8 a 3.10 para maior estabilidade.**

Componente	Especificacao Recomendada
Sistema Operacional	Windows 11, macOS 12+, Ubuntu 20.04+
Processador	Quad-core 2.5 GHz
Memoria RAM	8 GB ou mais
Armazenamento	1 GB livres
Python	Versao 3.10 ou superior
Navegador Web	Chrome/Edge/Firefox atualizados

Tabela 3: Requisitos recomendados para melhor desempenho

2.2 Instalacao Passo a Passo

Siga estas instrucoes cuidadosamente para uma instalacao bem-sucedida:

2.2.1 1. Preparacao do Ambiente

1. Verifique se o Python esta instalado:

```
1 python --version
2 # ou
3 python3 --version
4
```

2. Se necessario, instale o Python:

- **Windows:** Baixe do site oficial python.org
- **Linux:** `sudo apt install python3 python3-pip`
- **macOS:** `brew install python`

3. Atualize o pip:

```
1 pip install --upgrade pip
2
```

2.2.2 2. Criacao do Ambiente Virtual (Recomendado)

Ambientes virtuais isolam as dependencias do projeto, evitando conflitos:

```
1 # Windows
2 python -m venv venv_kairos
3 venv_kairos\Scripts\activate
4
5 # Linux/macOS
6 python3 -m venv venv_kairos
7 source venv_kairos/bin/activate
```

[colback=kairos-blue!5, colframe=kairos-blue!80, title=Ambientes Virtuais, fonttitle=]
Usar ambientes virtuais e uma pratica recomendada em desenvolvimento Python. Isso mantem as dependencias do KAIROS separadas de outros projetos no seu sistema.

2.2.3 3. Instalacao das Dependencias

Instale todas as bibliotecas necessarias:

```
1 # Instalacao basica (minima)
2 pip install streamlit pandas numpy plotly
3
4 # Instalacao completa (recomendada)
5 pip install streamlit pandas numpy plotly scipy openpyxl
6
7 # Para desenvolvimento/adicionais
8 pip install pytest black flake8
```

2.2.4 4. Verificacao da Instalacao

Crie um arquivo de teste para verificar se tudo foi instalado corretamente:

```
1 import streamlit as st
2 import pandas as pd
3 import numpy as np
4 import plotly.express as px
5
6 print("Versoes instaladas:")
7 print(f"Streamlit: {st.__version__}")
8 print(f"Pandas: {pd.__version__}")
9 print(f"NumPy: {np.__version__}")
10 print(f"Plotly: {px.__version__}")
11
12 # Verifica dependencias opcionais
13 try:
14     import openpyxl
15     print(f"Openpyxl: {openpyxl.__version__} ")
16 except ImportError:
17     print("Openpyxl: Nao instalado (exportacao Excel limitada)")
18
19 print("\n Todas as dependencias estao instaladas corretamente!")
```

Listing 1: teste_instalacao.py

Execute o teste:

```
1 python teste_instalacao.py
```

2.3 Configuracao do Ambiente

2.3.1 Variaveis de Ambiente (Opcional)

Para configuracoes avancadas, crie um arquivo `.env`:

```
1 # Configuracoes do Streamlit
2 STREAMLIT_SERVER_PORT=8501
3 STREAMLIT_SERVER_ADDRESS=0.0.0.0
4 STREAMLIT_BROWSER_GATHER_USAGE_STATS=False
5
6 # Configuracoes do Sistema KAIROS
7 KAIROS_MAX_STUDENTS=1000
8 KAIROS_DEFAULT_THEME=dark
9 KAIROS_LANGUAGE=pt-br
10
11 # Configuracoes de Performance
12 KAIROS_CACHE_TTL=3600
13 KAIROS_MAX_CACHE_ENTRIES=100
```

Listing 2: `.env` - Variaveis de ambiente

2.3.2 Configuracao do Streamlit

Crie o arquivo `.streamlit/config.toml`:

```
1 [server]
2 port = 8501
3 address = "0.0.0.0"
4 enableCORS = false
5 enableXsrfProtection = true
6 maxUploadSize = 200
7 enableWebSocketCompression = true
8
9 [browser]
10 serverAddress = "localhost"
11 serverPort = 8501
12 gatherUsageStats = false
13
14 [theme]
15 primaryColor = "#8B5CF6"
16 backgroundColor = "#0F172A"
17 secondaryBackgroundColor = "#1E293B"
18 textColor = "#F1F5F9"
19 font = "sans serif"
20
21 [runner]
22 magicEnabled = false
23
24 [client]
```

```
25 showErrorDetails = true
```

Listing 3: config.toml

2.4 Primeira Execucao

2.4.1 Execucao Basica

```
1 # Navegue ate a pasta do projeto
2 cd caminho/para/kairos
3
4 # Execute o sistema
5 streamlit run 1_V01.py
```

2.4.2 Opcoes de Execucao Avancadas

```
1 # Executar em porta especifica
2 streamlit run 1_V01.py --server.port 8080
3
4 # Executar em rede local
5 streamlit run 1_V01.py --server.address 0.0.0.0 --server.port 8501
6
7 # Executar sem cache do navegador
8 streamlit run 1_V01.py --browser.serverAddress localhost --server.
   clearCache true
9
10 # Executar em modo de desenvolvimento
11 streamlit run 1_V01.py --logger.level debug
```

2.4.3 Verificacao do Funcionamento

Após executar o comando, seu navegador padrão deverá abrir automaticamente. Se não abrir:

1. Acesse manualmente: <http://localhost:8501>
2. Verifique se a porta não está sendo usada por outro processo
3. Confirme que não há firewalls bloqueando a conexão

2.5 Solucao de Problemas de Instalacao

2.5.1 Problemas Comuns e Solucoes

2.5.2 Teste de Instalacao Completa

Para verificar completamente a instalação, execute o script de diagnóstico:

Problema	Solucao
"ModuleNotFoundError" para alguma biblioteca	Execute: <code>pip install nome_da_biblioteca</code>
Porta 8501 ja em uso	Use outra porta: <code>--server.port 8502</code>
Erro de permissao no Linux	Use sudo ou ajuste permissoes da pasta
Streamlit nao inicia	Verifique se Python esta no PATH do sistema
Erros de compatibilidade	Use versoes especificas: <code>pip install streamlit==1.28.0</code>
Problemas com ambiente virtual	Recreate o ambiente: <code>rm -rf venv; python -m venv venv</code>

Tabela 4: Problemas comuns de instalacao

```

1 #!/usr/bin/env python3
2 """
3 Script de diagnostico para o sistema KAIROS
4 Verifica todos os componentes necessarios
5 """
6
7 import sys
8 import subprocess
9 import importlib.util
10
11 def check_python_version():
12     """Verifica versao do Python"""
13     version = sys.version_info
14     print(f"Python: {sys.version}")
15
16     if version.major == 3 and version.minor >= 8:
17         print("Versao do Python compativel")
18         return True
19     else:
20         print("Python 3.8 ou superior necessario")
21         return False
22
23 def check_package(package_name, min_version=None):
24     """Verifica se um pacote esta instalado"""
25     try:
26         spec = importlib.util.find_spec(package_name)
27         if spec is None:
28             print(f"{package_name}: Nao instalado")

```

```
29         return False
30
31     if min_version:
32         mod = importlib.import_module(package_name)
33         if hasattr(mod, '__version__'):
34             print(f"        {package_name}: {mod.__version__}")
35             return True
36         else:
37             print(f"        {package_name}: Instalado (versao
desconhecida)")
38             return True
39         else:
40             print(f"        {package_name}: Instalado")
41             return True
42     except ImportError:
43         print(f"        {package_name}: Nao instalado")
44         return False
45
46 def check_system_resources():
47     """Verifica recursos do sistema"""
48     import psutil
49
50     print(f"\nRecursos do Sistema:")
51     print(f"Memoria RAM: {psutil.virtual_memory().total / (1024**3):.1f}
GB")
52     print(f"CPU Cores: {psutil.cpu_count()}")
53     print(f"Espaco em disco: Livre")
54
55     return True
56
57 def main():
58     print("=== DIAGNOSTICO DO SISTEMA KAIROS ===\n")
59
60     # Verifica versao do Python
61     python_ok = check_python_version()
62
63     print("\n=== VERIFICACAO DE PACOTES ===")
64
65     # Pacotes obrigatorios
66     packages_required = [
67         ('streamlit', '1.28.0'),
68         ('pandas', '2.0.0'),
69         ('numpy', '1.24.0'),
70         ('plotly', '5.17.0'),
71         ('scipy', '1.11.0'),
72     ]
73
74     # Pacotes opcionais
```

```
75     packages_optional = [  
76         ('openpyxl', None),  
77     ]  
78  
79     all_ok = True  
80  
81     print("\n--- Pacotes Obrigatorios ---")  
82     for package, version in packages_required:  
83         if not check_package(package, version):  
84             all_ok = False  
85  
86     print("\n--- Pacotes Opcionais ---")  
87     for package, version in packages_optional:  
88         check_package(package, version)  
89  
90     # Verifica recursos do sistema  
91     try:  
92         import psutil  
93         check_system_resources()  
94     except ImportError:  
95         print("\n      psutil nao instalado - pulando verificacao de  
recursos")  
96  
97     print("\n=== RESUMO DO DIAGNOSTICO ===")  
98     if all_ok and python_ok:  
99         print("      Sistema pronto para executar o KAİROS!")  
100        print("\nPara iniciar:")  
101        print("1. Navegue ate a pasta do projeto")  
102        print("2. Execute: streamlit run 1_V01.py")  
103        print("3. Acesse: http://localhost:8501")  
104    else:  
105        print("      Alguns requisitos nao atendidos")  
106        print("\nExecute para instalar dependencias:")  
107        print("pip install streamlit pandas numpy plotly scipy openpyxl")  
108    )  
109  
110    return 0 if all_ok and python_ok else 1  
111  
112 if __name__ == "__main__":  
    sys.exit(main())
```

Listing 4: diagnostico_kairos.py

3 Manual do Usuario

3.1 Primeiros Passos

3.1.1 Tela Inicial

Ao acessar o sistema KAIROS pela primeira vez, voce vera a tela de boas-vindas com:

- **Titulo principal:** "KAIROS - Sistema de Analise de Avaliacoes"
- **Introducao:** Breve descricao das funcionalidades
- **Barra lateral:** Menu com configuracoes e guias
- **Area principal:** Local para interacao com o sistema

[colback=kairos-blue!5, colframe=kairos-blue!80, title=Explore a Interface, fonttitle=]
Antes de comecar a usar, explore todas as abas e menus para familiarizar-se com a interface. A barra lateral contem todas as funcionalidades principais organizadas logicamente.

3.1.2 Estrutura da Interface

A interface do KAIROS esta organizada em quatro areas principais:

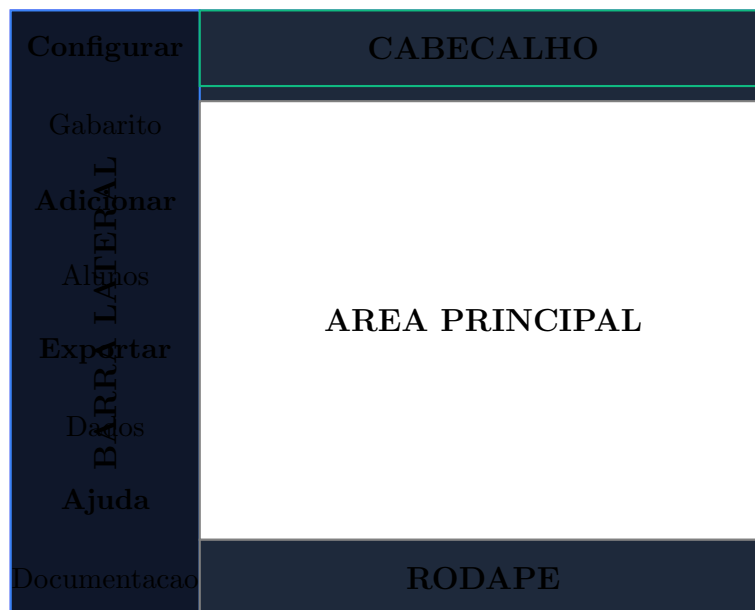


Figura 2: Estrutura da interface do KAIROS

3.2 Configuracao do Gabarito

3.2.1 Importancia do Gabarito

O gabarito e a base de toda a analise. Ele define as respostas corretas para cada questao e permite que o sistema:

1. Converta respostas em dados binarios (acerto/erro)

2. Calcule estatisticas por questao
3. Estime parametros psicometricos
4. Gere analises comparativas

3.2.2 Formatos Aceitos

Voce pode inserir o gabarito em varios formatos:

Formato	Exemplo
Com virgulas	A, B, C, D, A, B, C, D, E, A
Sem espacos	ABCDEABCDEABCDE
Com numeros	1, 2, 3, 4, 1, 2, 3, 4, 5, 1
Alternativas multiplas	A, B/C, D, A/B, C

Tabela 5: Formatos de entrada para gabarito

[colback=kairos-green!5, colframe=kairos-green!80, title=Dicas para o Gabarito, fonttitle=]

- Use letras maiusculas (A, B, C, D, E)
- Para questoes com multiplas respostas corretas, separe com barra (A/B)
- O sistema automaticamente conta o numero de questoes
- Verifique cuidadosamente cada resposta antes de prosseguir

3.2.3 Processo de Configuracao

Siga estes passos para configurar o gabarito:

1. Na barra lateral, localize a secao "Configurar Gabarito"
2. Digite as respostas corretas no campo de texto
3. Observe o preview que mostra as primeiras questoes
4. Confirme que o numero de questoes esta correto
5. Prossiga para a proxima etapa

3.3 Adicao de Alunos

3.3.1 Comparacao entre Modos de Entrada

Modo Manual	Modo CSV
Vantagens	<ul style="list-style-type: none"> • Ideal para pequenas turmas (ate 50 alunos) • Interface intuitiva • Correcao imediata de erros • Nenhum preparo previo necessario
Desvantagens	<ul style="list-style-type: none"> • Demorado para muitas questoes • Suscetivel a erros de digitacao • Nao pratico para turmas grandes
Quando Usar	<ul style="list-style-type: none"> • Testes rapidos • Turmas pequenas • Quando nao ha arquivo disponivel

Tabela 6: Comparacao entre modos de entrada

3.3.2 Modo Manual: Guia Detalhado

1. **Defina o numero de alunos:** Use o seletor numerico
2. **Preencha os dados para cada aluno:**
 - **Nome:** Identificador unico para cada aluno
 - **Respostas:** Selecione A, B, C, D, E ou "-" para omitida
3. **Organizacao das questoes:** As questoes sao organizadas automaticamente em colunas para facilitar a visualizacao
4. **Validacao:** O sistema verifica automaticamente:
 - Nomes duplicados
 - Numero de respostas por aluno
 - Formatos validos
5. **Processamento:** Clique em "Processar Respostas" para converter em dados analisaveis

3.3.3 Modo CSV: Preparacao do Arquivo

Prepare seu arquivo CSV seguindo estas especificacoes:

1 Aluno , Q1 , Q2 , Q3 , Q4 , Q5 , Q6 , Q7 , Q8 , Q9 , Q10
--

```

2 Joao Silva,A,B,C,D,A,B,C,D,E,A
3 Maria Santos,B,A,D,C,B,A,D,C,B,A
4 Pedro Costa,C,D,A,B,C,D,A,B,C,D
5 Ana Oliveira,D,C,B,A,D,C,B,A,D,C
6 Lucas Pereira,A,A,A,A,B,B,B,B,C,C
7 Julia Fernandes,B,B,B,B,C,C,C,C,D,D

```

Listing 5: Exemplo de arquivo CSV formatado corretamente

Regras importantes para o CSV:

- A primeira coluna DEVE se chamar "Aluno"
- Demais colunas devem seguir o padrao Q1, Q2, Q3, etc.
- Use apenas letras A, B, C, D, E para respostas
- Nao use caracteres especiais ou acentos nos nomes das colunas
- Salve o arquivo como UTF-8 para evitar problemas com caracteres especiais

[colback=red!5, colframe=red!80, title=Problemas Comuns com CSV, fonttitle=]

- Separador errado: Use virgula, nao ponto-e-virgula
- Encoding incorreto: Salve como UTF-8
- Cabecalho faltante: A primeira linha deve conter os nomes das colunas
- Numero de colunas: Deve corresponder ao numero de questoes do gabarito

3.3.4 Script de Preparacao de CSV

Para ajudar na preparacao de arquivos CSV, use este script:

```

1 import pandas as pd
2 import sys
3
4 def preparar_csv(input_file, output_file, num_questoes):
5     """
6     Prepara arquivo CSV para uso no KAIROS
7     """
8     try:
9         # Ler arquivo de entrada
10        df = pd.read_excel(input_file) if input_file.endswith('.xlsx')
11        else pd.read_csv(input_file)

```

```
12     # Renomear primeira coluna para 'Aluno'
13     if df.columns[0] != 'Aluno':
14         df = df.rename(columns={df.columns[0]: 'Aluno'})
15
16     # Verificar numero de questoes
17     colunas_respostas = [col for col in df.columns if col != 'Aluno'
18 ]
19
20     if len(colunas_respostas) != num_questoes:
21         print(f"Atencao: Arquivo tem {len(colunas_respostas)}
22 questoes, mas gabarito tem {num_questoes}")
23         print("Ser o mantidas apenas as primeiras colunas
24 correspondentes")
25         colunas_respostas = colunas_respostas[:num_questoes]
26
27     # Converter respostas para maiusculas
28     for col in colunas_respostas:
29         df[col] = df[col].astype(str).str.upper()
30
31     # Selecionar apenas colunas necessarias
32     colunas_finais = ['Aluno'] + colunas_respostas
33     df_final = df[colunas_finais].copy()
34
35     # Salvar CSV formatado
36     df_final.to_csv(output_file, index=False, encoding='utf-8')
37
38     print(f"    Arquivo preparado: {output_file}")
39     print(f"    Alunos: {len(df_final)}")
40     print(f"    Questoes: {len(colunas_respostas)}")
41     print(f"    Primeiros alunos:")
42     print(df_final.head().to_string())
43
44 except Exception as e:
45     print(f"    Erro: {str(e)}")
46     return False
47
48 return True
49
50 if __name__ == "__main__":
51     # Exemplo de uso
52     input_file = "dados_brutos.xlsx"
53     output_file = "dados_kairos.csv"
54     num_questoes = 20 # Ajuste conforme seu gabarito
55
56     preparar_csv(input_file, output_file, num_questoes)
```

Listing 6: preparar_csv.py

3.4 Dashboard Principal

3.4.1 Abas e Funcionalidades

O dashboard do KAIROS esta organizado em quatro abas principais:

Aba	Funcionalidades e Utilidade
Dashboard	<ul style="list-style-type: none"> • Visao geral da turma • Metricas principais • Graficos consolidados • Ranking de alunos • Identificacao de problemas
Analise Individual	<ul style="list-style-type: none"> • Desempenho detalhado por aluno • Graficos especificos • Recomendacoes pedagogicas • Comparacao com a turma • Evolucao por questao
Tutores de Colegas	<ul style="list-style-type: none"> • Identificacao automatica de tutores • Formacao de grupos • Score de tutoria • Sugestoes de intervencao
Exportar Dados	<ul style="list-style-type: none"> • Geracao de relatorios • Exportacao multiplos formatos • Dados brutos processados • Personalizacao de saidas

Tabela 7: Funcionalidades das abas principais

3.4.2 Metricas Principais

O dashboard apresenta quatro metricas-chave em tempo real:

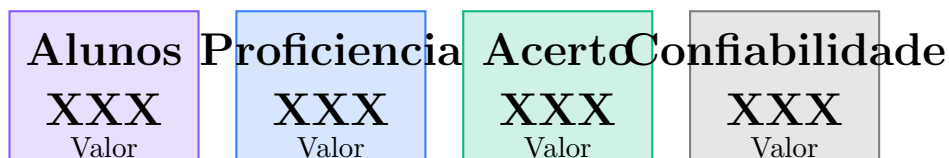


Figura 3: Metricas principais do dashboard

Interpretacao das metricas:

- **Alunos:** Numero total de alunos analisados

- **Proficiencia Media:** Habilidade media da turma na escala TRI
- **Taxa de Acerto:** Porcentagem media de acertos
- **Confiabilidade:** Indice de consistencia interna do teste

3.4.3 Panorama Geral da Turma

O grafico de panorama geral mostra quatro visoes integradas:

1. **Distribuicao da Proficiencia:** Histograma mostrando como os alunos estao distribuidos
2. **Dificuldade por Questao:** Grafico de barras mostrando a dificuldade de cada questao
3. **Taxa de Acerto da Turma:** Linha mostrando o desempenho medio por questao
4. **Top 5 Melhores Desempenhos:** Ranking dos alunos com maior proficiencia

[colback=kairos-green!5, colframe=kairos-green!80, title=Interpretacao do Panorama, font-title=]

- **Distribuicao normal:** Ideal, mostra variabilidade adequada
- **Distribuicao assimetrica:** Pode indicar problemas no teste
- **Dificuldade balanceada:** Questoes com dificuldades variadas
- **Taxa de acerto equilibrada:** Entre 30% e 70% e ideal

3.4.4 Analise Multidimensional das Questoes

Este grafico de dispersao mostra tres dimensoes simultaneamente:

- **Eixo X:** Dificuldade da questao (b)
- **Eixo Y:** Discriminacao da questao (a)
- **Tamanho do ponto:** Taxa de acerto da questao
- **Cor do ponto:** Correlacao bisserial

Zonas de interpretacao:

- **Canto superior esquerdo:** Questoes faceis com boa discriminacao (otimas)

- **Canto superior direito:** Questoes dificeis com boa discriminacao (desafiadoras)
- **Canto inferior:** Questoes com baixa discriminacao (problematicas)
- **Pontos pequenos:** Questoes com baixa taxa de acerto

3.5 Analise Individual

3.5.1 Selecao e Visualizacao

Para analisar um aluno especifico:

1. Selecione a aba " Analise Individual"
2. Escolha o aluno no menu dropdown
3. Observe as quatro metricas principais
4. Analise o grafico de desempenho por questao
5. Leia as recomendacoes pedagogicas

3.5.2 Metricas Individuais

Cada aluno tem quatro metricas especificas:

Metrica	Interpretacao
Proficiencia ()	<ul style="list-style-type: none"> • Valor numerico na escala TRI • Comparavel entre diferentes testes • Indica nivel de habilidade
Pontuacao	<ul style="list-style-type: none"> • Numero de acertos sobre total • Facil de entender • Nao comparavel entre testes diferentes
% Acerto	<ul style="list-style-type: none"> • Porcentagem de questoes acertadas • Familiar para professores • Limitações psicométricas
Posicao	<ul style="list-style-type: none"> • Ranking na turma • Comparacao relativa • Motivacional

Tabela 8: Metricas individuais e suas interpretacoes

3.5.3 Grafico de Desempenho por Questao

Este grafico mostra:

- **Barras coloridas:** Acerto (verde) ou erro (vermelho) em cada questao
- **Linha roxa:** Dificuldade de cada questao (eixo direito)
- **Hover informacao:** Detalhes ao passar o mouse

Padroes a observar:

- **Sequencias de acertos:** Areas de dominio
- **Sequencias de erros:** Dificuldades especificas
- **Correlacao com dificuldade:** Erros em questoes facis sao preocupantes
- **Padrao aleatorio:** Pode indicar chute

3.5.4 Recomendacoes Pedagogicas

Baseado na proficiencia (), o sistema gera recomendacoes especificas:

3.6 Sistema de Tutores

3.6.1 Fundamentacao Teorica

O sistema de identificacao de tutores baseia-se em:

1. **Aprendizagem entre pares:** Alunos aprendem efetivamente com colegas
2. **Zona de Desenvolvimento Proximal:** Tutores atuam como mediadores
3. **Autoeficacia:** Tutores desenvolvem confianca e competencias
4. **Responsabilidade compartilhada:** Toda a turma se beneficia

3.6.2 Criterios de Selecao

Para ser identificado como tutor potencial, o aluno deve atender a:

- **Proficiencia alta:** ≥ 1.0 (acima da media significativamente)
- **Desempenho consistente:** Taxa de acerto $\geq 70\%$
- **Score combinado:** Balanceamento entre proficiencia e consistencia

Faixa de	Classificacao	Recomendacoes
$j - 1.5$	Muito baixa	<ul style="list-style-type: none"> • Intervencao imediata • Atendimento individualizado • Acompanhamento constante • Reavaliacao diagnostica
$-1.5 \quad j - 0.5$	Baixa	<ul style="list-style-type: none"> • Reforco escolar • Grupos de estudo • Pratica adicional • Monitoramento
$-0.5 \quad j \quad 0.5$	Media	<ul style="list-style-type: none"> • Manter ritmo • Desafios moderados • Aprimoramento • Acompanhamento regular
$0.5 \quad j \quad 1.5$	Alta	<ul style="list-style-type: none"> • Desafios adicionais • Tutoria de colegas • Aprofundamento • Lideranca
1.5	Muito alta	<ul style="list-style-type: none"> • Tutor principal • Atividades avancadas • Mentoria • Desenvolvimento especial

Tabela 9: Recomendacoes pedagogicas baseadas na proficiencia

[colback=kairos-blue!5, colframe=kairos-blue!80, title=Ajuste de Critérios, fonttitle=]
Em turmas com nível geral baixo, voce pode ajustar os criterios:

- Reduza o limite de para 0.5
- Considere alunos com bom desempenho relativo
- Use o top 10% da turma como referencia

3.6.3 Calculo do Score de Tutor

O score de tutor e calculado combinando:

$$\text{Score}_{\text{tutor}} = 0.6 \times \text{Norm}(\theta) + 0.4 \times \text{Norm}(\% \text{Acerto})$$

Onde:

- Norm(θ): Proficiencia normalizada entre 0 e 1
- Norm(%Acerto): Taxa de acerto normalizada entre 0 e 1

3.6.4 Formacao de Grupos de Tutoria

Sugestoes praticas para formar grupos:

Estrategia	Implementacao
Grupos Heterogeneos	1 tutor + 3-4 alunos com dificuldades similares
Tutoria Especifica	Tutor especialista em topico especifico
Rotacao de Tutores	Tutores diferentes para diferentes temas
Tutoria em Duplas	1 tutor + 1 aluno (para casos especiais)
Grupos Colaborativos	Multiplos tutores trabalhando juntos

Tabela 10: Estrategias de formacao de grupos

3.6.5 Plano de Acao para Tutoria

1. Organizacao inicial:

- Identificar tutores (sistema KAIROS)
- Formar grupos equilibrados
- Definir horarios fixos
- Estabelecer objetivos claros

2. Implementacao:

- Sessoes semanais de 45-60 minutos
- Foco em topicos especificos
- Material de apoio preparado
- Registro de atividades

3. Monitoramento:

- Avaliacoes quinzenais

- Feedback dos tutores
- Ajuste de grupos
- Reconhecimento de progresso

4. Avaliacao:

- Medicao de impacto
- Satisfacao dos participantes
- Melhorias continuas
- Expansao do programa

3.7 Exportacao de Dados

3.7.1 Formatos Disponiveis

O KAIROS oferece multiplos formatos de exportacao:

3.7.2 Estrutura dos Relatorios

Todos os relatorios incluem as seguintes secoes:

1. **Metadados:** Informacoes basicas sobre a analise
2. **Explicacao TRI:** Contexto para interpretacao
3. **Panorama da Turma:** Estatisticas gerais
4. **Top Tutores:** Lista de alunos com potencial
5. **Ranking Completo:** Todos os alunos ordenados
6. **Analise de Questoes:** Parametros de cada item
7. **Questoes Problematicas:** Itens que precisam revisao
8. **Recomendacoes:** Sugestoes pedagogicas

3.7.3 Scripts de Processamento Pos-Exportacao

Para trabalhar com os dados exportados, use estes scripts:

```
1 import pandas as pd
2 import json
```

Formato	Melhor Para	Caracteristicas
TXT	<ul style="list-style-type: none"> • Relatorios impressos • Compartilhamento simples • Leitura humana 	<ul style="list-style-type: none"> • Formato texto puro • Inclui explicacoes • Facil de ler • Sem estrutura complexa
CSV	<ul style="list-style-type: none"> • Planilhas • Analise adicional • Integracao 	<ul style="list-style-type: none"> • Dados estruturados • Multiplas secoes • Compativel • Editavel
Excel	<ul style="list-style-type: none"> • Analise avancada • Graficos • Relatorios formais 	<ul style="list-style-type: none"> • Multiplas abas • Formatacao • Graficos • Profissional
JSON	<ul style="list-style-type: none"> • Integracao sistemas • APIs • Programacao 	<ul style="list-style-type: none"> • Estruturado • Hierarquico • Programavel • Moderno
ZIP	<ul style="list-style-type: none"> • Backup • Arquivos multiplos • Organizacao 	<ul style="list-style-type: none"> • Compactado • Completo • Conveniente • Versatil

Tabela 11: Formatos de exportacao disponiveis

```

3 from datetime import datetime
4
5 def processar_relatorio_txt(arquivo_txt):
6     """Processa relatorio TXT para extracao de insights"""
7     with open(arquivo_txt, 'r', encoding='utf-8') as f:
8         conteudo = f.read()
9
10    insights = {
11        'data_processamento': datetime.now().isoformat(),
12        'arquivo': arquivo_txt,
13        'tamanho': len(conteudo),
14        'linhas': conteudo.count('\n') + 1,
15        'secoes_encontradas': []
16    }
17
18    # Identifica secoes

```

```
19     secoes = [  
20         'RELATORIO DE ANALISE',  
21         'PANORAMA GERAL DA TURMA',  
22         'TOP 10 TUTORES',  
23         'ANALISE DAS QUESTOES',  
24         'RECOMENDACOES PEDAGOGICAS'  
25     ]  
26  
27     for secao in secoes:  
28         if secao in conteudo:  
29             insights['secoes_encontradas'].append(secao)  
30  
31     return insights  
32  
33 def converter_csv_para_graficos(arquivo_csv):  
34     """Converte CSV do KAİROS para visualizacoes personalizadas"""  
35     df = pd.read_csv(arquivo_csv)  
36  
37     # Identifica tipo de arquivo  
38     if 'Aluno' in df.columns and 'Proficiencia' in df.columns:  
39         tipo = 'resultados_alunos'  
40     elif 'Questao' in df.columns and 'Dificuldade' in df.columns:  
41         tipo = 'analise_questoes'  
42     else:  
43         tipo = 'desconhecido'  
44  
45     return {  
46         'tipo': tipo,  
47         'colunas': list(df.columns),  
48         'registros': len(df),  
49         'amostra': df.head().to_dict('records')  
50     }  
51  
52 if __name__ == "__main__":  
53     # Exemplo de uso  
54     arquivo = "Relatorio_Turma_TRI_20240115.txt"  
55     insights = processar_relatorio_txt(arquivo)  
56     print(f"Insights do relatorio: {insights}")
```

Listing 7: processar_exportacao.py

4 Teoria de Resposta ao Item (TRI)

4.1 Fundamentos da TRI

A Teoria de Resposta ao Item (TRI) é um conjunto de modelos matemáticos que descrevem a relação entre a habilidade latente de um indivíduo e sua probabilidade de responder

corretamente a um item de teste.

4.1.1 Diferencas em Relacao a TCT

Teoria Classica dos Testes (TCT)	Teoria de Resposta ao Item (TRI)
<ul style="list-style-type: none"> • Foco no teste como um todo • Escores dependentes do teste • Amostra-dependente • Suposicoes menos rigidas 	<ul style="list-style-type: none"> • Foco nos itens individualmente • Medidas independentes do teste • Amostra-independente • Suposicoes matematicas fortes

Tabela 12: Comparacao entre TCT e TRI

4.1.2 Vantagens da TRI

1. **Independencia da amostra:** Os parametros dos itens nao dependem da amostra de respondentes
2. **Independencia do teste:** As medidas de habilidade nao dependem do conjunto especifico de itens
3. **Erro de medida variável:** O erro de medida varia ao longo da escala de habilidade
4. **Comparabilidade:** Permite comparar individuos que responderam a testes diferentes
5. **Testes adaptativos:** Base para testes computadorizados adaptativos

4.2 Modelo Logístico de 2 Parâmetros (2PL)

O sistema KAIROS implementa o modelo logístico de 2 parâmetros (2PL), que e representado pela equacao:

$$P_i(\theta) = \frac{1}{1 + e^{-a_i(\theta - b_i)}}$$

Onde:

- $P_i(\theta)$: Probabilidade de um individuo com habilidade θ responder corretamente ao item i
- a_i : Parâmetro de discriminacao do item i

- b_i : Parâmetro de dificuldade do item i
- θ : Habilidade latente do respondente
- e : Constante matematica de Euler (aproximadamente 2.71828)

4.2.1 Interpretacao Grafica

A curva caracteristica do item (CCI) mostra visualmente a relacao:

$$\theta P(\theta) a = 1.5, b = 0 a = 0.3, b = 0 a = 1.0, b = 1.5 a = 1.0,$$

Figura 4: Curvas Caracteristicas de Item (CCI) para diferentes parametros

4.3 Parâmetros da TRI

4.3.1 Dificuldade (b)

O parâmetro de dificuldade (b) indica o ponto na escala de habilidade onde a probabilidade de acerto é 0.5 (50%).

Interpretacao dos valores:

- $b < -2$: Questao muito facil (acertada por quase todos)
- $-2 \leq b < -1$: Questao facil
- $-1 \leq b < 1$: Questao de dificuldade media
- $1 \leq b < 2$: Questao dificil
- $b \geq 2$: Questao muito dificil (acertada por poucos)

[colback=kairos-green!5, colframe=kairos-green!80, title=Escala de Dificuldade, fonttitle=] **Na pratica, valores entre -3 e +3 sao comuns. Valores fora dessa faixa podem indicar problemas no item ou na amostra.**

4.3.2 Discriminacao (a)

O parâmetro de discriminacao (a) indica quao bem o item diferencia individuos com diferentes niveis de habilidade.

Interpretacao dos valores:

4.3.3 Proficiencia (θ)

A proficiencia (θ) representa o nivel de habilidade do respondente na escala TRI.

Valor	Classificacao	Interpretacao e Acao
$a < 0$	Negativa	<ul style="list-style-type: none"> • Item funciona inversamente • Revisao imediata necessaria • Possivel erro no gabarito
$0 \leq a < 0.3$	Baixa	<ul style="list-style-type: none"> • Discriminacao insuficiente • Considerar revisao • Pode nao medir bem a habilidade
$0.3 \leq a < 0.6$	Moderada	<ul style="list-style-type: none"> • Discriminacao adequada • Item aceitavel • Bom para uso geral
$0.6 \leq a < 1.0$	Alta	<ul style="list-style-type: none"> • Boa discriminacao • Item de qualidade • Ideal para testes
$a \geq 1.0$	Muito alta	<ul style="list-style-type: none"> • Excelente discriminacao • Item muito bom • Manter em bancos de itens

Tabela 13: Interpretacao do parâmetro de discriminacao

Interpretacao dos valores:

- $\theta < -2$: Proficiencia muito baixa
- $-2 \leq \theta < -1$: Proficiencia baixa
- $-1 \leq \theta < 1$: Proficiencia media
- $1 \leq \theta < 2$: Proficiencia alta
- $\theta \geq 2$: Proficiencia muito alta

[colback=kairos-blue!5, colframe=kairos-blue!80, title=Escala Padronizada, fonttitle=]
Na TRI, a escala e padronizada com media 0 e desvio padrao 1. Isso permite comparacoes entre diferentes testes e populacoes.

4.4 Estimacao dos Parâmetros**4.4.1 Metodos de Estimacao**

O sistema KAIROS utiliza uma abordagem simplificada para estimacao dos parâmetros:

1. **Dificuldade (b)**: Estimada a partir da proporção de acertos (p-value)

$$b_i = -\ln\left(\frac{p_i}{1-p_i}\right)$$

2. **Discriminação (a)**: Estimada a partir da correlação item-total

$$a_i = 2.5 \times r_{it}$$

onde r_{it} é a correlação ponto-bisserial entre o item e o escore total

3. **Proficiência (θ)**: Estimada usando escores padronizados

$$\theta_j = \Phi^{-1}\left(\frac{X_j + 0.5}{n + 1}\right)$$

onde Φ^{-1} é a função quantil da distribuição normal padrão

4.4.2 Implementação no Código

```

1 def fit_model(self, response_matrix):
2     """Ajusta modelo TRI 2PL aos dados"""
3     n_students, n_items = response_matrix.shape
4
5     # 1. Estimacao da dificuldade (b) a partir de p-values
6     p_values = response_matrix.mean(axis=0)
7     p_values = np.clip(p_values, 0.001, 0.999) # Evitar divisao por
8     zero
9     difficulty = -np.log(p_values / (1 - p_values))
10
11     # 2. Estimacao da discriminacao (a) a partir de correlacao
12     discrimination = []
13     total_scores = response_matrix.sum(axis=1)
14
15     for i in range(n_items):
16         try:
17             # Correlacao ponto-bisserial
18             corr = pearsonr(response_matrix[:, i], total_scores)[0]
19             # Converter para escala de discriminacao
20             discrimination.append(2.5 * corr if not np.isnan(corr) else
21 0.5)
22         except:
23             discrimination.append(0.5) # Valor padrao
24
25     # 3. Estimacao da proficiencia (theta)
26     student_p = (response_matrix.sum(axis=1) + 0.5) / (n_items + 1)
27     student_p = np.clip(student_p, 0.001, 0.999)
28     ability = norm.ppf(student_p) # Transformacao inversa normal

```

```

28     return {
29         'difficulty': difficulty,
30         'discrimination': np.array(discrimination),
31         'ability': ability
32     }

```

Listing 8: Estimacao de parametros no KAIROS

4.5 Indices de Qualidade

4.5.1 Confiabilidade do Teste

A confiabilidade no contexto TRI pode ser estimada por:

$$\alpha = \frac{n \times \bar{r}}{1 + (n - 1) \times \bar{r}}$$

Onde:

- n : Numero de itens
- \bar{r} : Correlacao media item-total

Interpretacao de α (Alpha de Cronbach):

- $\alpha < 0.6$: Confiabilidade baixa
- $0.6 \leq \alpha < 0.8$: Confiabilidade moderada
- $0.8 \leq \alpha < 0.9$: Confiabilidade boa
- $\alpha \geq 0.9$: Confiabilidade excelente

4.5.2 Indice de Discriminacao

Calculado como a diferenca entre a proporcao de acertos no grupo superior e no grupo inferior:

$$D_i = P_{\text{superior}} - P_{\text{inferior}}$$

Interpretacao:

- $D_i < 0.2$: Discriminacao baixa
- $0.2 \leq D_i < 0.3$: Discriminacao moderada

- $0.3 \leq D_i < 0.4$: Discriminacao boa
- $D_i \geq 0.4$: Discriminacao excelente

4.5.3 Correlacao Bisserial Pontual

Mede a associacao entre a resposta ao item e a habilidade total:

$$r_{pb} = \frac{M_1 - M_0}{s} \times \sqrt{\frac{p}{q}}$$

Interpretacao:

- $r_{pb} < 0.1$: Correlacao muito baixa
- $0.1 \leq r_{pb} < 0.2$: Correlacao baixa
- $0.2 \leq r_{pb} < 0.3$: Correlacao moderada
- $r_{pb} \geq 0.3$: Correlacao alta

4.6 Aplicacoes Praticas da TRI

4.6.1 Identificacao de Itens Problematicos

Use os seguintes criterios para identificar itens que precisam de revisao:

1. **Discriminacao baixa** ($a < 0.3$): Item nao diferencia bem alunos
2. **Dificuldade extrema** ($|b| > 2$): Muito facil ou muito dificil
3. **Correlacao negativa**: Item funciona inversamente
4. **Padrao de respostas atipico**: Analise de distratores

4.6.2 Interpretacao Pedagogica

Como usar os resultados da TRI para acoes pedagogicas:

4.6.3 Formacao de Grupos de Nivel

Use as proficiencias estimadas para formar grupos pedagogicos:

1. **Grupo de intervencao**: $\theta < -1$ (necessitam apoio intensivo)
2. **Grupo de desenvolvimento**: $-1 \leq \theta < 0$ (beneficiam-se de praticas)

Resultado	Ação Pedagógica Recomendada
Item muito fácil ($b < -2$)	<ul style="list-style-type: none"> • Considerar remover do teste • Usar apenas para motivação inicial • Verificar se mede objetivo educacional
Item muito difícil ($b > 2$)	<ul style="list-style-type: none"> • Revisar conteúdo ensinado • Considerar como item bonus • Avaliar adequação ao nível
Baixa discriminação ($a < 0.3$)	<ul style="list-style-type: none"> • Revisar redação do item • Analisar distratores • Verificar ambiguidades
Correlação negativa	<ul style="list-style-type: none"> • Verificar gabarito • Revisar completamente • Possivelmente eliminar

Tabela 14: Ações pedagógicas baseadas em análise de itens

3. **Grupo de consolidação:** $0 \leq \theta < 1$ (prontos para desafios)
4. **Grupo de aprofundamento:** $\theta \geq 1$ (preparados para extensões)

4.7 Limitações e Considerações

4.7.1 Premissas da TRI

O modelo 2PL assume:

1. **Unidimensionalidade:** O teste mede apenas uma habilidade latente
2. **Independência local:** As respostas aos itens são independentes dada a habilidade
3. **Monotonicidade:** A probabilidade de acerto aumenta com a habilidade
4. **Forma funcional específica:** A relação é logística (ou normal ogiva)

[colback=red!5, colframe=red!80, title=Validação das Premissas, fonttitle=] **Em contextos práticos, essas premissas são aproximadamente verdadeiras. Grandes violações podem afetar a validade das estimativas.**

4.7.2 Limitações do Modelo Simplificado

A implementação do KAIROS utiliza uma versão simplificada que:

- Nao estima parâmetro de acerto ao acaso (modelo 3PL)
- Usa metodos de estimacao aproximados
- Assume normalidade das habilidades
- Pode nao ser ideal para testes muito curtos

[colback=kairos-blue!5, colframe=kairos-blue!80, title=Quando Usar com Cautela, font-title=]

- Testes com menos de 10 itens
- Amostras muito pequenas (menos de 30 alunos)
- Itens com alto efeito de acerto ao acaso
- Quando precisa-se de precisao maxima

5 Referencia Tecnica

5.1 Arquitetura do Sistema

5.1.1 Diagrama de Componentes

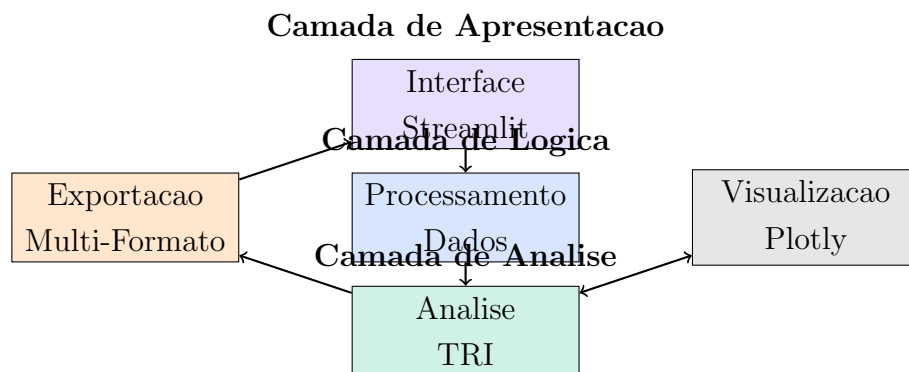


Figura 5: Arquitetura em camadas do sistema KAIROS

5.1.2 Fluxo de Dados

O sistema processa os dados seguindo este fluxo:

1. **Entrada:** Respostas dos alunos (manual ou CSV)
2. **Validacao:** Verificacao de formatos e consistencia
3. **Conversao:** Respostas para formato binario (0/1)

4. **Analise TRI:** Calculo de parametros psicometricos
5. **Processamento:** Geracao de metricas e estatisticas
6. **Visualizacao:** Criacao de graficos e dashboards
7. **Exportacao:** Geracao de relatorios e dados processados

5.2 Estrutura doCodigo

5.2.1 Organizacao do Arquivo Principal

O arquivo 1_V01.py esta organizado em secoes logicas:

```
1 # 1. IMPORTACOES
2 import streamlit as st
3 import pandas as pd
4 # ... outras importacoes
5
6 # 2. CONFIGURACOES
7 st.set_page_config(...)
8 # ... configuracoes CSS
9
10 # 3. CLASSES E FUNCOES
11 class TRI_Simulator:
12     # Implementacao da TRI
13
14 # 4. FUNCOES DE ANALISE
15 @st.cache_data
16 def run_advanced_tri_analysis(df):
17     # Analise principal
18
19 # 5. FUNCOES DE VISUALIZACAO
20 def plot_theta_distribution_dark(...):
21     # Graficos
22
23 # 6. FUNCOES DE EXPORTACAO
24 def create_text_report(...):
25     # Relatorios
26
27 # 7. INTERFACE PRINCIPAL
28 # Sidebar
29 # Area principal
30 # Abas e funcionalidades
```

Listing 9: Estrutura do arquivo principal

5.2.2 Gerenciamento de Estado

O sistema usa `st.session_state` para gerenciar estado:

```
1 # Inicializacao do estado
2 if 'df_manual' not in st.session_state:
3     st.session_state.df_manual = None
4
5 if 'df_binary' not in st.session_state:
6     st.session_state.df_binary = None
7
8 # Uso do estado
9 if st.session_state.df_binary is not None:
10     # Processar dados existentes
11     process_analysis(st.session_state.df_binary)
12
13 # Atualizacao do estado
14 st.session_state.df_binary = processed_data
```

5.3 Otimizacao de Performance

5.3.1 Uso de Cache

O sistema implementa cache em varios niveis:

```
1 # Cache de dados (Streamlit)
2 @st.cache_data(
3     ttl=3600, # Time to live: 1 hora
4     max_entries=10, # Maximo de entradas em cache
5     show_spinner=True # Mostrar spinner durante processamento
6 )
7 def process_data(df):
8     # Processamento pesado
9     return result
10
11 # Cache de recursos
12 @st.cache_resource
13 def load_model():
14     # Carregar modelo pesado
15     return model
```

5.3.2 Otimizacoes Especificas

1. **Processamento em lotes:** Para grandes volumes de dados
2. **Selecao de dados:** Carregar apenas o necessario
3. **Indexacao eficiente:** Usar indices do pandas

4. **Tipos de dados apropriados:** Usar tipos menores quando possivel

5.4 Extensibilidade

5.4.1 Adicionando Novas Funcionalidades

Para estender o sistema:

```
1 # 1. Nova funcao de analise
2 def minha_nova_analise(df):
3     """Implemente sua nova analise aqui"""
4     resultado = df.groupby('categoria').mean()
5     return resultado
6
7 # 2. Nova visualizacao
8 def plot_novo_grafico(data):
9     """Crie sua nova visualizacao"""
10    fig = px.scatter(data, x='var1', y='var2')
11    return fig
12
13 # 3. Integracao na interface
14 if st.sidebar.checkbox("Nova Funcionalidade"):
15     resultado = minha_nova_analise(df)
16     st.plotly_chart(plot_novo_grafico(resultado))
```

5.4.2 Integracao com Outros Sistemas

O sistema pode ser integrado com:

- **Sistemas de gestao escolar:** Via exportacao CSV/JSON
- **APIs web:** Atraves de scripts Python
- **Bancos de dados:** Usando bibliotecas como SQLAlchemy
- **Ferramentas de BI:** Power BI, Tableau via exportacao

5.5 Solucao de Problemas Tecnicos

5.5.1 Erros Comuns e Solucoes

5.5.2 Logs e Diagnostico

Para diagnosticar problemas:

```
1 # Executar com logs detalhados
2 streamlit run 1_V01.py --logger.level debug
3
4 # Verificar logs
```

Erro/Problema	Solucao
"MemoryError" ao processar muitos dados	<ul style="list-style-type: none"> • Aumente memoria RAM • Processe em lotes menores • Use tipos de dados eficientes
Graficos lentos ou travando	<ul style="list-style-type: none"> • Reduza numero de pontos nos graficos • Use agregacao de dados • Desative animacoes
Interface nao atualiza	<ul style="list-style-type: none"> • Limpe cache do navegador • Reinicie o Streamlit • Verifique conexao
Exportacao falha	<ul style="list-style-type: none"> • Verifique permissoes de escrita • Confira espaco em disco • Use formato alternativo

Tabela 15: Problemas tecnicos comuns e solucoes

```

5 tail -f ~/.streamlit/logs/error.log
6
7 # Testar componentes individualmente
8 python -c "import pandas as pd; print(pd.__version__)"

```

5.6 Seguranca e Privacidade

5.6.1 Consideracoes de Seguranca

1. **Dados sensiveis:** Nao inclua informacoes pessoais identificaveis
2. **Local storage:** Os dados sao processados localmente
3. **Acesso:** Restrito ao usuario local (por padrao)
4. **Backup:** Mantenha copias dos dados importantes

5.6.2 Recomendacoes de Uso Seguro

- Use o sistema apenas em redes confiaveis
- Mantenha o Python e as bibliotecas atualizados
- Revise codigos de terceiros antes de executar
- Faça backup regular dos dados

- Use ambientes virtuais para isolamento

A Apendices

A.1 A: Glossario de Termos

Termo	Definicao
TRI	Teoria de Resposta ao Item. Conjunto de modelos matematicos que relacionam habilidade latente com probabilidade de resposta.
Proficiencia ()	Medida de habilidade na escala TRI. Media 0, desvio padrao 1.
Dificuldade (b)	Parâmetro que indica o nivel de habilidade necessario para 50% de chance de acerto.
Discriminacao (a)	Capacidade do item de diferenciar individuos com diferentes niveis de habilidade.
CCI	Curva Caracteristica do Item. Grafico que mostra a relacao entre habilidade e probabilidade de acerto.
Confiabilidade	Consistencia interna do teste. Medida pela correlacao entre itens.
Alpha de Cronbach	Medida comum de confiabilidade baseada na consistencia interna.
Correlacao Bisserial	Correlacao entre uma variavel dicotomica (item) e uma continua (habilidade).
p-value	Proporcao de respondentes que acertaram o item.
Item-total correla-tion	Correlacao entre o item e o escore total do teste.
Unidimensionalidade	Premissa de que o teste mede apenas uma habilidade latente.
Independencia Local	Premissa de que as respostas aos itens sao independentes dada a habilidade.

A.2 B: Referencias Bibliograficas

Referências

- [1] Baker, F. B. (2001). *The basics of item response theory*. ERIC Clearinghouse on Assessment and Evaluation.

- [2] Embretson, S. E., & Reise, S. P. (2000). *Item response theory for psychologists*. Lawrence Erlbaum Associates.
- [3] De Ayala, R. J. (2013). *The theory and practice of item response theory*. Guilford Publications.
- [4] Hambleton, R. K., Swaminathan, H., & Rogers, H. J. (1991). *Fundamentals of item response theory*. Sage.
- [5] Streamlit Documentation. (2024). *Official Streamlit Documentation*. Disponível em: <https://docs.streamlit.io>
- [6] McKinney, W., & pandas development team. (2024). *pandas: powerful data structures for data analysis*. Disponível em: <https://pandas.pydata.org>
- [7] Plotly Technologies Inc. (2024). *Plotly Python Open Source Graphing Library*. Disponível em: <https://plotly.com/python>
- [8] Virtanen, P., et al. (2024). *SciPy: fundamental algorithms for scientific computing in Python*. Nature Methods, 17(3), 261-272.

A.3 C: Recursos Adicionais

A.3.1 Sites e Documentacao

- Documentacao oficial do Streamlit: <https://docs.streamlit.io>
- Repositorio do projeto: <https://github.com/seu-usuario/kairos>
- Forum de discussao: <https://discuss.streamlit.io>
- Tutoriais de TRI: <https://www.ime.usp.br/~abe/lista/pdfm9cJKUmFz.pdf>

A.3.2 Livros Recomendados

- "Item Response Theory- Frank B. Baker
- "Psychometrics: An Introduction- R. Michael Furr
- "Streamlit for Data Science- Tyler Richards
- "Python for Data Analysis- Wes McKinney

A.3.3 Cursos e Formacoes

- Coursera: "Introduction to Psychometrics"

- **edX**: "Data Analysis for Social Scientists"
- **Udemy**: "Python for Data Science and Machine Learning"
- **Pluralsight**: "Data Visualization with Python"

A.4 D: Codigos de Exemplo

A.4.1 Script de Backup Automatico

```
1 import shutil
2 import os
3 from datetime import datetime
4 import json
5
6 def fazer_backup(pasta_origem, pasta_backup):
7     """Faz backup automatico dos dados do KAIROS"""
8
9     if not os.path.exists(pasta_backup):
10         os.makedirs(pasta_backup)
11
12     data_hora = datetime.now().strftime("%Y%m%d_%H%M%S")
13     nome_backup = f"kairos_backup_{data_hora}"
14     caminho_backup = os.path.join(pasta_backup, nome_backup)
15
16     # Copiar arquivos
17     if os.path.exists(pasta_origem):
18         shutil.copytree(pasta_origem, caminho_backup)
19
20     # Registrar backup
21     registro = {
22         'data': data_hora,
23         'pasta': caminho_backup,
24         'tamanho': sum(
25             os.path.getsize(os.path.join(dirpath, filename))
26             for dirpath, dirnames, filenames in os.walk(
27                 caminho_backup)
28             for filename in filenames
29         ) / (1024*1024), # Em MB
30         'arquivos': len([
31             f for _, _, files in os.walk(caminho_backup)
32             for f in files
33         ])
34     }
35
36     # Salvar registro
37     with open(os.path.join(pasta_backup, 'backups.json'), 'a') as f:
38         json.dump(registro, f)
39         f.write('\n')
```

```
39
40     print(f"      Backup criado: {caminho_backup}")
41     print(f"      Tamanho: {registro['tamanho']:.2f} MB")
42     print(f"      Arquivos: {registro['arquivos']}")
43
44     return True
45
46     return False
47
48 if __name__ == "__main__":
49     # Configuracoes
50     PASTA_DADOS = "./data" # Ajuste conforme seu projeto
51     PASTA_BACKUP = "./backups"
52
53     fazer_backup(PASTA_DADOS , PASTA_BACKUP)
```

Listing 10: backup_automatico.py

A.4.2 Script de Monitoramento

```
1 import psutil
2 import time
3 import json
4 from datetime import datetime
5
6 def monitorar_recursos(intervalo=60, duracao=3600):
7     """Monitora uso de recursos durante execucao"""
8
9     dados_monitoramento = []
10    inicio = time.time()
11
12    print(f"      Iniciando monitoramento por {duracao/60:.0f} minutos"
13    )
14    print("Pressione Ctrl+C para interromper\n")
15
16    try:
17        while time.time() - inicio < duracao:
18            # Coletar metricas
19            timestamp = datetime.now().isoformat()
20
21            metricas = {
22                'timestamp': timestamp,
23                'cpu_percent': psutil.cpu_percent(interval=1),
24                'memory_percent': psutil.virtual_memory().percent,
25                'memory_used_mb': psutil.virtual_memory().used /
26                (1024*1024),
27                'disk_usage': psutil.disk_usage('/').percent,
28                'process_count': len(psutil.pids())
29            }
```

```
28         dados_monitoramento.append(metricas)
29
30
31         # Exibir resumo
32         print(f"{timestamp[:19]} | CPU: {metricas['cpu_percent']:5.1f}% | "
33               f"RAM: {metricas['memory_percent']:5.1f}% | "
34               f"Processos: {metricas['process_count']:4d}")
35
36         time.sleep(intervalo - 1) # Ajustar para intervalo
37
38     except KeyboardInterrupt:
39         print("\n\n Monitoramento interrompido pelo usuario")
40
41     # Salvar dados
42     nome_arquivo = f"monitoramento_{datetime.now().strftime('%Y%m%d_%H%M%S')}.json"
43     with open(nome_arquivo, 'w') as f:
44         json.dump(dados_monitoramento, f, indent=2)
45
46     print(f"\n\n Dados salvos em: {nome_arquivo}")
47
48     # Resumo final
49     if dados_monitoramento:
50         cpu_media = sum(m['cpu_percent'] for m in dados_monitoramento) /
51         len(dados_monitoramento)
52         ram_media = sum(m['memory_percent'] for m in dados_monitoramento
53         ) / len(dados_monitoramento)
54
55         print(f"\n\n RESUMO FINAL:")
56         print(f"    Tempo monitorado: {len(dados_monitoramento)}
57         intervalos")
58         print(f"    CPU media: {cpu_media:.1f}%")
59         print(f"    RAM media: {ram_media:.1f}%")
60         print(f"    Consumo maximo de RAM: {max(m['memory_used_mb'] for m
61         in dados_monitoramento):.0f} MB")
62
63     return dados_monitoramento
64
65 if __name__ == "__main__":
66     # Monitorar por 5 minutos com intervalo de 10 segundos
67     monitorar_recursos(intervalo=10, duracao=300)
```

Listing 11: monitoramento.py

A.5 E: Historico de Versoes

Versao	Data	Status	Mudancas Principais
1.0.0	2024-01-15	Estavel	Versao inicial completa com todas as funcionalidades basicas
0.9.0	2024-01-10	Beta	Sistema de tutores implementado, melhorias na interface
0.8.0	2024-01-05	Beta	Exportacao multi-formato, otimizacoes de performance
0.7.0	2024-01-01	Alpha	Dashboard interativo completo, graficos com Plotly
0.6.0	2023-12-20	Alpha	Implementacao basica da TRI, processamento de dados
0.5.0	2023-12-15	Prototipo	Interface Streamlit basica, entrada manual de dados
0.1.0	2023-12-01	Conceito	Prova de conceito, estrutura inicial do projeto

A.6 F: Contato e Suporte

A.6.1 Informacoes de Contato

- **Desenvolvedor:** Mauricio A. Ribeiro
- **E-mail:** suporte@kairos-analise.com
- **GitHub:** <https://github.com/seu-usuario/kairos>
- **Documentacao Online:** <https://kairos-analise.readthedocs.io>
- **Forum de Discussao:** <https://github.com/seu-usuario/kairos/discussions>

A.6.2 Politica de Suporte

- **Suporte Basico:** Gratuito via GitHub Issues
- **Resposta:** Em ate 72 horas uteis
- **Escopo:** Bugs, problemas de instalacao, duvidas basicas
- **Fora do Escopo:** Customizacoes extensivas, integracoes complexas

A.6.3 Como Reportar Problemas

Ao reportar um problema, inclua:

1. **Descricao clara:** O que aconteceu vs. o que era esperado
2. **Passos para reproduzir:** Sequencia exata para reproduzir o problema
3. **Contexto:** Sistema operacional, versao do Python, tamanho dos dados
4. **Logs de erro:** Mensagens de erro completas

5. Screenshots: Quando aplicavel

A.6.4 Contribuindo com o Projeto

O projeto e open-source e aceita contribuicoes:

- **Reporte bugs:** Crie uma Issue no GitHub
- **Sugira melhorias:** Use a aba Discussions
- **Envie codigo:** Pull Requests sao bem-vindos
- **Melhore documentacao:** Corrija erros ou adicione exemplos
- **Traducoes:** Ajude a traduzir para outros idiomas