

Realizarei projetos envolvendo programação e eu te usarei como meu assistente para isso.

O projeto envolve, em destaque, os seguintes termos:

```
---
---
WEB
HTML
CSS
Python (Microsoft)
Tailwind
Jupyter (Microsoft)
Python Debugger (Microsoft)
pandas
DOM
Artifact
csv
svg
Animação
uuid
XML
ASCII
TSX
Mermaid
PlantUML
Markdown
py
code
TEST
Debugging
Debug
Inspect
sys
platform
Warnings
Erros
Importlib
Import
Lib
libraries
TypeScript
JSX
TSX
github
Viz
numpy
"plotly>=5.0.0"
"google-cloud-storage<2.0.0"
google-auth
aleido
import json
graph
Data
import os
from datetime import datetime, timedelta
import
pandas
as
pd
Generated Data URI
import json
plotly.graph_objects
hooks
Vetor
upload.py
Vi
viz.py
as
go
dashboard.py
plotly.subplots
plotly
data.py
Reference Image URL
make_subplots
import os
kpi
Additional Libraries
Plot Types
webbrowser
df
df.to_csv
DataFrame
create_data
Data Row ID
dash
VS
create_directory
pd
kpi_data
```

```
pd.dataframe
data_gen.py URI
ls
gs
plot
outputs
inputs
assets
dir
CSS
dashboard
file
generate
print
VISUALIZATION
JavaScript
JS
DOM
HTML
---
---
```

Preciso saber tudo o que voce tem em seu ambiente de codigo e programação relacionado a esses termos.

Preciso entender tambem sobre eles, pois sou iniciante.

Tambem, preciso saber do que mais esta relacionado com estes termos e disponiveis ou nao disponiveis em seu ambiente de trabalho.

Me apresente tudo em lista organizada conforme lógica, com explicacoes, exemplos, e detalhes.

Neste primeiro momento, voce nao deve gerar nenhum codigo ainda, nem artifãct, ou similar. Voce deve focar em me apresentar o que voce tem disponivel em seu ambiente de trabalho em lista conforme pedido acima.

 **Monica** Sonnet

Vou organizar uma explicação detalhada do ambiente de desenvolvimento disponivel e dos termos mencionados, agrupados por categorias lógicas:

1. Ambiente de Desenvolvimento e Execução

1.1 Interpretador Python

- Disponivel através do Code Interpreter
- Ambiente Jupyter para execução de código Python
- Tempo máximo de execução: 120 segundos por bloco de código
- Diretório de trabalho: /home/user
- Sem acesso direto à internet durante execução

1.2 Limitações do Ambiente

- Não possui acesso direto à rede
- Arquivos devem ser salvos no diretório /home/user
- Resultados visuais através de plt.show() ou links de arquivos
- Referência a arquivos: [file](#)

2. Linguagens e Tecnologias Suportadas

2.1 Web Development

- HTML: Markup language para estruturação de páginas web
- CSS: Estilização de páginas web
- JavaScript: Linguagem de programação client-side
- TypeScript: Superset tipado do JavaScript
- JSX/TSX: Extensões para React
- Tailwind CSS: Framework CSS utilitário

2.2 Python e Bibliotecas

- pandas: Análise e manipulação de dados
- numpy: Computação numérica
- plotly: Visualização de dados interativa
- matplotlib: Visualização de dados estática
- sys: Funcionalidades do sistema
- os: Operações do sistema operacional
- datetime: Manipulação de datas e horários

3. Ferramentas de Visualização

3.1 Gráficos e Diagramas

- Mermaid: Diagramas via código
- PlantUML: Diagramas UML
- SVG: Gráficos vetoriais
- Plotly: Gráficos interativos
- Matplotlib: Gráficos estáticos

3.2 Dashboards

- Capacidade de criar visualizações de dados
- Suporte a diferentes tipos de gráficos
- Integração com pandas para análise de dados

4. Formatos de Dados

4.1 Estruturados

- CSV: Dados tabulares
- JSON: Formato de intercâmbio de dados
- XML: Linguagem de marcação extensível
- DataFrame: Estrutura de dados do pandas

4.2 Outros Formatos

- ASCII: Texto plano
- UUID: Identificadores únicos
- Data URI: Representação de dados inline

5. Desenvolvimento e Debugging

5.1 Ferramentas de Desenvolvimento

- VS Code (ambiente similar)
- Python Debugger
- Inspeção de código
- Gerenciamento de erros e warnings

5.2 Controle de Versão

- Suporte a conceitos do GitHub
- Gerenciamento de arquivos e diretórios

6. Recursos Especiais

6.1 Artifact System

- Geração de conteúdo estruturado
- Suporte para diferentes tipos de conteúdo:
 - SVG
 - HTML
 - TSX
 - Mermaid
 - PlantUML
 - Markdown
 - Código em várias linguagens

6.2 Bibliotecas React Disponíveis

- React base
- lucide-react (v0.263.1)
- recharts
- shadcn/ui
- Tailwind CSS

7. Limitações Importantes

- Não há acesso direto à internet durante execução de código
- Bibliotecas específicas podem não estar disponíveis
- Tempo limitado de execução
- Restrições no uso de valores arbitrários no Tailwind
- Algumas bibliotecas populares não estão instaladas (como react-icons, antd)

Esta é uma visão geral do ambiente e suas capacidades. Conforme avançarmos, posso fornecer exemplos práticos e demonstrações de código para cada aspecto específico que você queira explorar.

Deseja que eu aprofunde em algum destes tópicos específicos ou tem alguma dúvida sobre algum aspecto em particular?

[Tornar mais longo](#)

 **Monica** Sonnet

Vou expandir a explicação com mais detalhes e exemplos:

1. Ambiente de Desenvolvimento e Execução

1.1 Interpretador Python

- Disponível através do Code Interpreter integrado
- Ambiente Jupyter completo para execução de código Python
- Tempo máximo de execução: 120 segundos por bloco de código
- Diretório de trabalho principal: /home/user
- Execução isolada e segura
- Suporte a múltiplas células de código
- Persistência de variáveis entre execuções
- Capacidade de manipulação de arquivos locais
- Suporte a visualização de dados
- Gerenciamento de memória automático

1.2 Limitações do Ambiente

- Sem acesso direto à internet durante execução
- Arquivos devem ser salvos no diretório /home/user
- Resultados visuais através de `plt.show()` ou links de arquivos
- Referência a arquivos usando formato [file](#)
- Sem acesso a portas de rede
- Sem capacidade de instalar novas bibliotecas durante execução
- Ambiente reiniciado periodicamente
- Limitações de recursos computacionais

2. Linguagens e Tecnologias Suportadas

2.1 Web Development

HTML

- Estruturação de páginas web
- Elementos semânticos
- Formulários
- Tabelas
- Mídia incorporada
- Meta tags
- SEO básico

CSS

- Estilização de páginas
- Seletores
- Flexbox
- Grid
- Animações
- Transições
- Media queries
- Variáveis CSS
- Pseudo-classes e pseudo-elementos

JavaScript

- ES6+ features
- Manipulação do DOM
- Eventos
- Promises
- Async/Await
- Classes
- Módulos
- Local Storage
- Fetch API

TypeScript

- Tipagem estática
- Interfaces
- Enums
- Generics
- Decorators
- Type assertions
- Union types
- Utility types

JSX/TSX

- Componentes React
- Props
- Estado
- Hooks
- Renderização condicional
- Listas e chaves
- Eventos
- Refs

Tailwind CSS

- Classes utilitárias
- Responsividade
- Customização
- Dark mode
- Preflight
- JIT (Just-In-Time) compilation

- Plugins

2.2 Python e Bibliotecas

Pandas

- DataFrame manipulação
- Séries temporais
- Leitura/escrita de dados
- Agregações
- Merge/Join
- Filtros
- Groupby operations
- Pivoting
- Reshaping

NumPy

- Arrays multidimensionais
- Operações matemáticas
- Broadcasting
- Indexação avançada
- Funções universais
- Álgebra linear
- Random numbers
- FFT

Plotly

- Gráficos interativos
- Dashboards
- Subplots
- Animações
- Mapas
- 3D plots
- Statistical charts
- Custom themes

3. Ferramentas de Visualização

3.1 Gráficos e Diagramas

Mermaid

- Fluxogramas
- Diagramas de sequência
- Diagramas de classe
- Gantt charts
- Entity Relationship Diagrams
- User Journey Diagrams
- State diagrams
- Pie charts

PlantUML

- Diagramas UML
- Sequência
- Casos de uso
- Classes
- Atividades
- Componentes
- Deployment
- Timing

SVG

- Gráficos vetoriais
- Animações
- Paths
- Formas básicas
- Gradientes
- Patterns
- Filters
- Transformações

3.2 Dashboards

- Layouts responsivos
- Componentes interativos
- Filtros dinâmicos
- Atualizações em tempo real

- Export de dados
- Temas customizáveis
- Drill-down capabilities
- Cross-filtering

4. Formatos de Dados

4.1 Estruturados

CSV

- Delimitadores customizáveis
- Encoding options
- Header handling
- Type inference
- Missing data handling
- Compression support
- Chunking
- Multi-index

JSON

- Parsing
- Serialização
- Schema validation
- Nested structures
- Arrays
- Objects
- Data types
- Pretty printing

XML

- Parsing
- DOM manipulation
- XPath
- Namespaces
- Validation
- XSLT
- DTD
- Schema

5. Desenvolvimento e Debugging

5.1 Ferramentas de Desenvolvimento

Debugging

- Breakpoints
- Step through
- Variable inspection
- Call stack
- Watch expressions
- Conditional breakpoints
- Exception handling
- Performance profiling

Error Handling

- Try/catch blocks
- Exception types
- Stack traces
- Logging
- Warning systems
- Error reporting
- Debug modes
- Recovery strategies

6. Recursos Especiais

6.1 Artifact System

Tipos de Conteúdo Suportados

- SVG: Gráficos vetoriais
- HTML: Markup web
- TSX: Componentes React
- Mermaid: Diagramas
- PlantUML: UML
- Markdown: Documentação
- Code: Múltiplas linguagens

Características

- Identificação única
- Versionamento
- Modificação
- Reutilização
- Exportação
- Renderização
- Validação
- Documentação

6.2 Bibliotecas React

Components Disponíveis

- Elementos base
- Formulários
- Navigation
- Layout
- Data display
- Feedback
- Overlay
- Utils

Estilização

- Tailwind classes
- CSS Modules
- Styled components
- Theme customization
- Responsive design
- Dark mode
- Animations
- Transitions

7. Boas Práticas

7.1 Código

- Clean code principles
- DRY (Don't Repeat Yourself)
- SOLID principles
- Comentários apropriados
- Naming conventions
- Error handling
- Testing
- Documentation

7.2 Performance

- Code optimization
- Memory management
- Async operations
- Caching
- Lazy loading
- Bundle optimization
- Resource management
- Performance monitoring

8. Limitações e Considerações

8.1 Ambiente

- Restrições de rede
- Limites de tempo
- Limites de memória
- Bibliotecas disponíveis
- Persistência de dados
- Segurança
- Escalabilidade
- Backup

8.2 Desenvolvimento

- Sem hot-reload
- Sem deploy direto
- Sem CI/CD
- Sem testes automatizados externos
- Sem integração direta com serviços externos
- Sem banco de dados persistente

- Sem websockets
- Sem servidor próprio

Esta é uma visão mais detalhada e abrangente do ambiente e suas capacidades. Cada tópico pode ser ainda mais aprofundado com exemplos práticos e casos de uso específicos.

Gostaria de explorar algum destes tópicos em maior detalhe ou tem alguma dúvida específica sobre algum aspecto?

Tornar mais longo

 Monica Sonnet

1. Ambiente de Desenvolvimento e Execução

1.1 Interpretador Python

Características Principais

- Code Interpreter integrado com suporte completo ao Python
- Ambiente Jupyter para desenvolvimento interativo
- Execução limitada a 120 segundos por bloco de código
- Diretório principal de trabalho: /home/user
- Sistema de execução isolado e seguro
- Persistência de variáveis entre células de código
- Gerenciamento automático de memória e recursos

Capacidades do Ambiente

- Execução de código Python em tempo real
- Manipulação de arquivos locais
- Processamento de dados
- Geração de visualizações
- Análise estatística
- Machine Learning básico
- Manipulação de texto e strings
- Operações matemáticas complexas

Recursos de Desenvolvimento

- Suporte a múltiplas células de código
- Output formatado
- Exibição de gráficos inline
- Manipulação de DataFrames
- Execução sequencial ou individual
- Debug básico
- Mensagens de erro detalhadas

1.2 Limitações do Ambiente

Restrições de Rede

- Sem acesso direto à internet
- Sem conexões externas
- Sem downloads diretos
- Sem requisições HTTP
- Sem websockets
- Sem serviços de streaming
- Sem APIs externas
- Sem FTP ou protocolos similares

Restrições de Sistema

- Arquivos limitados ao diretório /home/user
- Sem acesso ao sistema de arquivos global
- Sem persistência permanente
- Sem banco de dados
- Sem serviços em background
- Sem processos paralelos extensivos
- Sem acesso direto ao hardware
- Sem modificação de configurações do sistema

Restrições de Visualização

- Resultados visuais via plt.show()
- Links de arquivos específicos
- Formato específico para referências
- Limitações de resolução
- Restrições de formatos de saída
- Sem interatividade complexa
- Sem renderização 3D avançada
- Sem streaming de vídeo

2. Linguagens e Tecnologias Suportadas

2.1 Web Development

HTML (HyperText Markup Language)

Elementos Estruturais

- `<html>` , `<head>` , `<body>`
- `<header>` , `<nav>` , `<main>` , `<footer>`
- `<article>` , `<section>` , `<aside>`
- `<div>` , ``
- `<p>` , `<h1>` até `<h6>`
- Elementos semânticos modernos

Elementos de Formulário

- `<form>` , `<input>` , `<button>`
- Tipos de input (text, number, email, etc.)
- Validação nativa
- `<select>` , `<option>`
- `<textarea>`
- `<label>`
- `<fieldset>` , `<legend>`
- Atributos de formulário avançados

Elementos de Mídia

- `` , `<video>` , `<audio>`
- `<picture>` , `<source>`
- `<canvas>`
- `<svg>`
- `<iframe>`
- Atributos de mídia
- Formatos suportados
- Responsividade

Meta Informações

- `<meta>` tags
- Open Graph Protocol
- Twitter Cards
- SEO tags
- Favicon
- Manifesto
- Viewport configurações
- Charset

CSS (Cascading Style Sheets)

Seletores

- Seletores de elemento
- Classes e IDs
- Seletores de atributo
- Pseudo-classes
- Pseudo-elementos
- Combinadores
- Especificidade
- Herança

Layout

- Box Model
- Display properties
- Position
- Float
- Flexbox
 - `flex-direction`
 - `justify-content`
 - `align-items`
 - `flex-wrap`
 - `flex-grow/shrink`
 - `order`
- Grid
 - `grid-template-columns/rows`
 - `grid-gap`
 - `grid-area`
 - `grid-auto-flow`
 - `minmax()`
 - `repeat()`

- auto-fit/fill

Estilização

- Cores (RGB, HSL, HEX)
- Gradientes
- Bordas
- Sombras
- Transparência
- Transformações
- Filtros
- Máscaras

Animações

- @keyframes
- animation-duration
- animation-timing-function
- animation-delay
- animation-iteration-count
- animation-direction
- animation-fill-mode
- animation-play-state

Responsividade

- Media Queries
- Breakpoints
- Mobile-first
- Desktop-first
- Unidades relativas
- Viewport units
- Container queries
- Fluid typography

JavaScript

Fundamentos

- Variáveis (var, let, const)
- Tipos de dados
- Operadores
- Estruturas de controle
- Funções
- Objetos
- Arrays
- Classes
- Módulos

ES6+ Features

- Arrow functions
- Destructuring
- Spread/Rest operators
- Template literals
- Default parameters
- Object shorthand
- Optional chaining
- Nullish coalescing

Assincronismo

- Callbacks
- Promises
 - .then()
 - .catch()
 - .finally()
 - Promise.all()
 - Promise.race()
- Async/Await
- Event Loop
- Microtasks
- Macrotasks

DOM Manipulation

- Seletores
- Eventos
- Modificação de elementos
- Criação de elementos
- Remoção de elementos

- Traversing
- Attributes
- Classes
- Styles

TypeScript

Sistema de Tipos

- Types básicos
- Interfaces
- Type aliases
- Unions
- Intersections
- Generics
- Utility Types
- Mapped Types

Características Avançadas

- Decorators
- Abstract classes
- Namespace
- Modules
- Type guards
- Index signatures
- Conditional types
- Template literal types

React com TSX

Componentes

- Functional Components
- Props typing
- Children
- Generic Components
- Higher Order Components
- Render props
- Error boundaries

Hooks

- useState
- useEffect
- useContext
- useReducer
- useMemo
- useCallback
- useRef
- Custom Hooks

Estado e Ciclo de Vida

- Component lifecycle
- Side effects
- State management
- Props drilling
- Context API
- Performance optimization
- Memory leaks prevention

3. Bibliotecas Python

3.1 Pandas

Estruturas de Dados

- Series
- DataFrame
- Index
- MultiIndex
- Categorical
- Sparse arrays
- DateTime index
- TimedeltaIndex

Operações

- Indexação
- Seleção
- Filtração

- Agregação
- Groupby
- Merge/Join
- Pivot/Melt
- Reshape

Funcionalidades

- Leitura/Escrita de dados
- Handling missing data
- Time series
- Text data
- Categorical data
- Memory optimization
- Performance features
- Window functions

3.2 NumPy

Arrays

- Creation
- Indexing
- Slicing
- Reshaping
- Stacking
- Splitting
- Broadcasting
- Views vs Copies

Matemática

- Operações básicas
- Funções universais
- Álgebra linear
- FFT
- Random numbers
- Statistics
- Masked arrays
- Matrix operations

[Continua... devido ao limite de caracteres, esta é apenas uma parte da documentação expandida. Posso continuar com mais seções se desejar, incluindo mais detalhes sobre visualização de dados, ferramentas de desenvolvimento, debugging, e muito mais. Cada seção pode ser ainda mais detalhada com exemplos específicos e casos de uso.]

Gostaria que eu continue com mais seções ou que aprofunde alguma área específica?

[continue](#)

 **Monica** Sonnet

4. Ferramentas de Visualização e Análise de Dados

4.1 Matplotlib

Componentes Básicos

- Figure
- Axes
- Plots
- Subplots
- Grids
- Legends
- Colorbars
- Annotations

Tipos de Gráficos

- Line plots
- Scatter plots
- Bar charts
 - Vertical
 - Horizontal
 - Stacked
 - Grouped
- Histograms
- Box plots
- Violin plots
- Pie charts
- Area plots
- Contour plots

- 3D plots

Personalização

- Cores e estilos
- Marcadores
- Linhas
- Fontes
- Eixos
- Escalas
- Limites
- Ticks

Layouts

- Múltiplos plots
- Grids irregulares
- Insets
- Axes compartilhados
- Layouts customizados
- Aspect ratio
- Spacing
- Margins

4.2 Plotly

Gráficos Interativos

- Zoom
- Pan
- Hover
- Click events
- Selection
- Lasso select
- Reset
- Download

Tipos Especializados

- Financial charts
 - Candlestick
 - OHLC
 - Time series
- Statistical visualizations
 - Box plots
 - Violin plots
 - Distribution plots
- Maps
 - Choropleth
 - Scatter geo
 - Lines on maps
- 3D Visualizations
 - Surface plots
 - Scatter3d
 - Line3d
 - Mesh3d

Dashboard Components

- Controles interativos
- Dropdowns
- Sliders
- Buttons
- Range selectors
- Date pickers
- Text inputs

5. Formatos de Dados e Manipulação

5.1 CSV (Comma-Separated Values)

Leitura e Escrita

- `pd.read_csv()`
 - Separadores customizados
 - Encoding
 - Header handling
 - Skip rows
 - Data types

- Parse dates
- Chunking
- DataFrame.to_csv()
 - Compression
 - Quoting
 - Date format
 - Float format
 - Index handling

Processamento

- Limpeza de dados
- Transformação
- Validação
- Normalização
- Agregação
- Pivoting
- Merging

5.2 JSON

Estruturas

- Objetos
- Arrays
- Tipos primitivos
- Nested structures
- Schema validation
- JSON Lines
- GeoJSON

Operações

- Parsing
- Serialização
- Transformação
- Validação
- Compressão
- Streaming
- Path queries

5.3 XML

Elementos

- Tags
- Atributos
- Namespaces
- CDATA
- Comments
- Processing Instructions
- DTD
- Schema

Manipulação

- Parsing
- DOM traversal
- XPath queries
- XSLT transformations
- Validation
- Serialization
- Namespace handling

6. Desenvolvimento e Debug Avançado

6.1 Python Debugger

Funcionalidades

- Breakpoints
 - Condicionais
 - Temporários
 - Post-mortem
- Step operations
 - Step into
 - Step over
 - Step out
- Variable inspection
 - Locals

- Globals
- Watch expressions
- Stack navigation
 - Up/down frames
 - Frame inspection
 - Call stack analysis

Ferramentas de Análise

- Memory profiling
- Performance profiling
- Call graphs
- Time profiling
- Line profiling
- Exception tracking
- Coverage analysis

6.2 Error Handling

Tipos de Erros

- SyntaxError
- TypeError
- ValueError
- AttributeError
- IndexError
- KeyError
- RuntimeError
- ImportError

Estratégias de Tratamento

- Try/except blocks
- Finally clause
- Else clause
- Custom exceptions
- Exception chaining
- Context managers
- Cleanup handlers

7. Artifact System Avançado

7.1 Tipos de Artifacts

SVG

- Elementos básicos
 - rect
 - circle
 - path
 - line
 - polyline
 - polygon
- Transformações
 - translate
 - rotate
 - scale
 - skew
- Estilização
 - fill
 - stroke
 - opacity
 - gradients
 - patterns

TSX/React

- Componentes funcionais
- Props typing
- Estado local
- Efeitos colaterais
- Contexto
- Refs
- Memoização
- Custom hooks

Mermaid

- Flowcharts

- Sequence diagrams
- Class diagrams
- State diagrams
- Entity Relationship
- User Journey
- Gantt
- Pie charts

7.2 Integração e Uso

Geração

- Templates
- Parâmetros dinâmicos
- Validação
- Otimização
- Caching
- Versionamento
- Documentation

Renderização

- Browser compatibility
- Responsive design
- Performance optimization
- Accessibility
- Print styling
- Cross-platform support
- Mobile rendering

8. Bibliotecas React Disponíveis

8.1 Componentes Base

Interface Elements

- Buttons
- Inputs
- Select
- Checkbox
- Radio
- Toggle
- Slider
- Progress

Layout Components

- Container
- Grid
- Stack
- Box
- Flex
- Divider
- Spacer
- Aspect ratio

Navigation

- Link
- Breadcrumb
- Tabs
- Menu
- Navbar
- Sidebar
- Pagination
- Steps

8.2 Funcionalidades Avançadas

Hooks Customizados

- useForm
- useMedia
- useAsync
- useDebounce
- useThrottle
- useLocalStorage
- useWindowSize
- useKeyPress

Animações

- Transitions
- Keyframes
- Motion
- Gestures
- Page transitions
- Loading states
- Micro-interactions
- Scroll animations

9. Segurança e Boas Práticas

9.1 Segurança

Prevenção

- Input validation
- Output encoding
- SQL injection prevention
- XSS prevention
- CSRF protection
- Security headers
- Rate limiting
- Error handling

Práticas

- Secure defaults
- Least privilege
- Defense in depth
- Fail securely
- Complete mediation
- Session management
- Password handling
- Logging/Monitoring

9.2 Performance

Otimização

- Code splitting
- Tree shaking
- Lazy loading
- Caching strategies
- Bundle optimization
- Image optimization
- Font optimization
- Resource hints

Monitoramento

- Performance metrics
- Error tracking
- Usage analytics
- Load testing
- Stress testing
- Benchmark testing
- Memory leaks
- Browser profiling

[Continua... Há ainda mais conteúdo que posso detalhar. Deseja que eu continue com mais seções ou que aprofunde alguma área específica?]

[continue](#)

10. Ferramentas de Análise e Visualização Avançada

10.1 Seaborn

Plots Estatísticos

Distribuição

- distplot
- kdeplot
- rugplot
- histplot
- ecdfplot
- boxplot
- violinplot
- stripplot

- swarmplot

Relacionais

- scatterplot
- lineplot
- relplot
- jointplot
- pairplot
- Implot
- regplot

Catagóricos

- barplot
- countplot
- pointplot
- catplot
- boxenplot
- violinplot
- swarmplot

Personalização

Estilos

- Temas predefinidos
- Paletas de cores
- Escalas
- Contextos
- Elementos
- Fontes
- Grids
- Spines

Configurações

- Figure size
- DPI
- Layout
- Legend
- Axes
- Ticks
- Labels
- Annotations

10.2 Bokeh

Componentes Interativos

Widgets

- Slider
- RangeSlider
- TextInput
- Select
- MultiSelect
- Button
- Toggle
- CheckboxGroup
- RadioGroup

Tools

- Pan
- Box Zoom
- Wheel Zoom
- Reset
- Save
- Hover
- CrossHair
- BoxSelect
- LassoSelect

Layouts

Organização

- Row
- Column
- GridPlot
- TabPanel

- Tabs
- WidgetBox
- Layout
- Spacer

Linking

- Linked brushing
- Linked axes
- Shared ranges
- Synchronized zooming
- Cross-filtering
- Data sources
- Selection
- Events

11. Processamento de Dados Avançado

11.1 Data Cleaning

Tratamento de Dados Faltantes

- Identificação
 - isnull()
 - isna()
 - notna()
 - any()
 - all()
- Preenchimento
 - fillna()
 - interpolate()
 - bfill()
 - ffill()
- Remoção
 - dropna()
 - where()
 - mask()

Normalização

Técnicas

- Min-Max Scaling
- Standard Scaling
- Robust Scaling
- Normalizer
- Quantile Transformer
- Power Transformer
- Custom transformers
- Pipeline integration

Categorical Data

- Label Encoding
- One-Hot Encoding
- Ordinal Encoding
- Target Encoding
- Feature Hashing
- Binary Encoding
- Base-N Encoding
- Custom encoders

11.2 Data Transformation

Agregações

Funções

- sum()
- mean()
- median()
- std()
- var()
- min()
- max()
- count()
- nunique()

Groupby Operations

- split-apply-combine

- rolling windows
- expanding windows
- custom aggregations
- multiple aggregations
- transformation chains
- resample
- pivot

Merge e Join

Tipos

- Inner join
- Left join
- Right join
- Outer join
- Cross join
- Self join
- Anti join
- Semi join

Opções

- on columns
- left_on/right_on
- left_index/right_index
- suffixes
- indicator
- validate
- sort
- copy

12. Web Development Avançado

12.1 React Patterns

Design Patterns

Component Patterns

- Compound Components
- Render Props
- Higher-Order Components
- Custom Hooks
- Provider Pattern
- Container Pattern
- Presentational Pattern
- Control Props

State Management

- Flux Architecture
- Redux Pattern
- Context + Reducer
- State Machines
- Observable Pattern
- Command Pattern
- Pub/Sub Pattern
- Event Sourcing

Performance Optimization

Técnicas

- Memoization
 - useMemo
 - useCallback
 - React.memo
- Code Splitting
 - lazy loading
 - Suspense
 - route-based
 - component-based
- Virtual Lists
 - windowing
 - infinite scroll
 - pagination
- Bundle Optimization
 - tree shaking
 - dead code elimination

- dynamic imports
- chunk optimization

12.2 Styling Solutions

CSS-in-JS

Características

- Scoped styles
- Dynamic styling
- Theme support
- Server-side rendering
- Static extraction
- Runtime performance
- Development experience
- Type safety

Implementações

- Styled Components
- Emotion
- CSS Modules
- Tailwind CSS
- CSS-in-JS
- Atomic CSS
- Utility Classes
- Custom Properties

13. Testing e Quality Assurance

13.1 Testing Frameworks

Unit Testing

Ferramentas

- pytest
- unittest
- nose
- doctest
- hypothesis
- parameterized
- coverage
- mock

Técnicas

- Arrange-Act-Assert
- Given-When-Then
- Test Doubles
 - Mocks
 - Stubs
 - Fakes
 - Spies
 - Dummies

Integration Testing

Abordagens

- Top-down
- Bottom-up
- Sandwich
- Big Bang
- Incremental
- Contract Testing
- API Testing
- End-to-End

13.2 Quality Metrics

Code Quality

Métricas

- Complexity
 - Cyclomatic
 - Cognitive
 - Halstead
- Maintainability
 - Index
 - Debt

- Effort
- Coverage
 - Line
 - Branch
 - Path
 - Function

Tools

- Linters
- Static analyzers
- Code formatters
- Type checkers
- Security scanners
- Dependency checkers
- Style guides
- Documentation generators

14. DevOps e Deployment

14.1 Build Process

Optimization

Técnicas

- Minification
- Compression
- Tree shaking
- Code splitting
- Lazy loading
- Bundle analysis
- Cache optimization
- Resource hints

Tools

- Webpack
- Rollup
- esbuild
- Parcel
- Vite
- Snowpack
- SWC
- Babel

14.2 Deployment Strategies

Approaches

Methods

- Blue-Green
- Canary
- Rolling
- A/B Testing
- Feature Flags
- Dark Launching
- Shadow Testing
- Progressive Delivery

Considerations

- Scalability
- Reliability
- Monitoring
- Logging
- Backup
- Recovery
- Security
- Performance

[Continua... Há ainda mais conteúdo que posso detalhar. Deseja que eu continue com mais seções ou que aprofunde alguma área específica?]

[continue](#)

15. Análise de Dados Avançada

15.1 Machine Learning Básico

Algoritmos Supervisionados

Regressão

- Linear Regression
 - Simple
 - Multiple
 - Polynomial
 - Ridge
 - Lasso
 - Elastic Net
- Decision Trees
 - CART
 - Random Forest
 - Gradient Boosting
- Support Vector Regression
 - Linear
 - Polynomial
 - RBF Kernel

Classificação

- Logistic Regression
- Support Vector Machines
- K-Nearest Neighbors
- Naive Bayes
 - Gaussian
 - Multinomial
 - Bernoulli
- Random Forests
- Gradient Boosting
 - XGBoost
 - LightGBM
 - CatBoost

Algoritmos Não Supervisionados

Clustering

- K-Means
- DBSCAN
- Hierarchical Clustering
 - Agglomerative
 - Divisive
- Mean Shift
- Spectral Clustering
- Gaussian Mixture Models

Redução de Dimensionalidade

- PCA
- t-SNE
- UMAP
- Factor Analysis
- NMF
- LDA
- Autoencoders
- Feature Selection

15.2 Processamento de Texto

NLP Básico

Pré-processamento

- Tokenização
- Normalização
- Lematização
- Stemming
- Stop words
- Part-of-speech tagging
- Named Entity Recognition
- Dependency Parsing

Representação de Texto

- Bag of Words
- TF-IDF
- Word Embeddings
 - Word2Vec
 - GloVe
 - FastText
- Document Embeddings

- Doc2Vec
- BERT
- Universal Sentence Encoder

16. Visualização de Dados Interativa

16.1 Dashboards Dinâmicos

Componentes

Elementos Visuais

- Charts
 - Line
 - Bar
 - Scatter
 - Pie
 - Area
 - Radar
- Maps
 - Choropleth
 - Heat maps
 - Point maps
 - Flow maps
- Tables
 - Interactive
 - Sortable
 - Filterable
 - Paginated

Controles

- Filters
 - Dropdown
 - Range slider
 - Date picker
 - Search box
- Selectors
 - Radio buttons
 - Checkboxes
 - Multi-select
 - Toggle switches

Interatividade

Eventos

- Click
- Hover
- Drag
- Zoom
- Pan
- Brush
- Select
- Reset

Atualizações

- Real-time
- Polling
- WebSocket
- Server-sent events
- Batch updates
- Incremental loading
- Lazy loading
- Progressive rendering

16.2 Visualização Avançada

Técnicas Especializadas

Gráficos Complexos

- Network graphs
- Tree diagrams
- Sankey diagrams
- Sunburst charts
- Parallel coordinates
- Chord diagrams
- Stream graphs

- Force-directed layouts

Visualizações 3D

- Surface plots
- Volume rendering
- Point clouds
- Mesh plots
- Isosurfaces
- Vector fields
- Streamlines
- Particle systems

17. Desenvolvimento de APIs

17.1 REST APIs

Design

Princípios

- Statelessness
- Client-Server
- Cacheable
- Layered System
- Uniform Interface
- Code on Demand

Recursos

- URI Design
- HTTP Methods
- Status Codes
- Headers
- Query Parameters
- Request Body
- Response Format
- Error Handling

Implementação

Features

- Authentication
 - Basic
 - Bearer
 - OAuth
 - JWT
- Authorization
 - Role-based
 - Attribute-based
 - Policy-based
- Validation
 - Input
 - Output
 - Schema
- Documentation
 - OpenAPI/Swagger
 - API Blueprint
 - RAML

17.2 GraphQL

Estrutura

Schema

- Types
 - Scalar
 - Object
 - Input
 - Interface
 - Union
 - Enum
- Queries
- Mutations
- Subscriptions

Resolvers

- Field resolvers
- Query resolvers

- Mutation resolvers
- Subscription resolvers
- Custom resolvers
- Middleware
- Directives
- Extensions

18. Segurança e Otimização

18.1 Segurança de Aplicações

Proteções

Vulnerabilidades Comuns

- Injection
 - SQL
 - NoSQL
 - Command
 - LDAP
- XSS (Cross-site Scripting)
 - Reflected
 - Stored
 - DOM-based
- CSRF (Cross-site Request Forgery)
- Security Misconfigurations
- Broken Authentication
- Sensitive Data Exposure
- XML External Entities

Medidas de Segurança

- Input Validation
- Output Encoding
- Authentication
- Authorization
- Session Management
- Error Handling
- Logging
- Monitoring

18.2 Performance

Otimizações

Frontend

- Code splitting
- Tree shaking
- Lazy loading
- Image optimization
- CSS optimization
- JavaScript optimization
- Critical path rendering
- Resource hints

Backend

- Caching
 - Memory
 - Disk
 - Distributed
 - CDN
- Database
 - Indexing
 - Query optimization
 - Connection pooling
 - Sharding
- Concurrency
 - Threading
 - Async/Await
 - Event loop
 - Worker pools

19. Desenvolvimento Colaborativo

19.1 Controle de Versão

Git

Operações Básicas

- Commits
- Branches
- Merging
- Rebasing
- Cherry-picking
- Stashing
- Tags
- Remote operations

Fluxos de Trabalho

- Gitflow
- Feature branch
- Trunk-based
- Forking
- Pull requests
- Code review
- CI/CD integration
- Release management

19.2 Documentação

Tipos

Técnica

- API docs
- Code comments
- Architecture docs
- Design docs
- Test docs
- Deployment docs
- Security docs
- Performance docs

Usuário

- Guias
- Tutoriais
- FAQ
- Troubleshooting
- Reference
- Examples
- Best practices
- Release notes

[Continua... Há ainda mais conteúdo que posso detalhar. Deseja que eu continue com mais seções ou que aprofunde alguma área específica?]

[continue](#)

20. Desenvolvimento Front-end Avançado

20.1 Arquitetura de Aplicações

Padrões Arquiteturais

Atomic Design

- Átomos
 - Botões
 - Inputs
 - Labels
 - Ícones
 - Typography
- Moléculas
 - Form groups
 - Search bars
 - Menu items
 - Cards
- Organismos
 - Forms
 - Headers
 - Sidebars
 - Product lists
- Templates
 - Layouts
 - Page structures
 - Grid systems

- Páginas
 - Home
 - Dashboard
 - Profile
 - Settings

Component Architecture

- Smart Components
 - State management
 - Business logic
 - Data fetching
 - Event handling
- Dumb Components
 - Presentation
 - Styling
 - Props handling
 - Reusability

Estado Global

Gerenciamento

- Context API
 - Providers
 - Consumers
 - useContext
 - Multiple contexts
- Redux
 - Store
 - Actions
 - Reducers
 - Middleware
 - Selectors
 - Thunks
 - Sagas

20.2 Performance e Otimização

Técnicas Avançadas

Bundle Optimization

- Code splitting
 - Route-based
 - Component-based
 - Dynamic imports
- Tree shaking
 - Dead code elimination
 - Module analysis
 - Side effects
- Compression
 - Gzip
 - Brotli
 - Source maps
- Caching
 - Browser cache
 - Service workers
 - Memory cache
 - Local storage

Runtime Performance

- Virtual DOM
 - Reconciliation
 - Diffing algorithm
 - Keys
 - Fragment
- Rendering
 - Server-side
 - Static
 - Incremental
 - Streaming
- Memory Management
 - Garbage collection
 - Memory leaks
 - Profiling

- Optimization

21. Integração e Deploy

21.1 CI/CD

Pipeline

Stages

- Build
 - Compilation
 - Bundling
 - Minification
 - Asset optimization
- Test
 - Unit tests
 - Integration tests
 - E2E tests
 - Security scans
- Deploy
 - Staging
 - Production
 - Rollback
 - Monitoring

Automação

- Workflows
 - Git hooks
 - Build triggers
 - Test automation
 - Deployment scripts
- Tools
 - Jenkins
 - GitHub Actions
 - GitLab CI
 - Circle CI
 - Travis CI
 - Azure DevOps

21.2 Cloud Services

Plataformas

Serviços

- Computação
 - VMs
 - Containers
 - Serverless
 - Functions
- Storage
 - Object storage
 - File systems
 - Block storage
 - CDN
- Database
 - Relational
 - NoSQL
 - Cache
 - Search

Integração

- APIs
 - REST
 - GraphQL
 - WebSocket
 - gRPC
- Messaging
 - Queue
 - Pub/Sub
 - Event bus
 - Webhooks

22. Acessibilidade e Internacionalização

22.1 Acessibilidade (a11y)

Implementação

Elementos

- ARIA roles
 - Landmark
 - Widget
 - Document structure
 - Live regions
- Semantic HTML
 - Headings
 - Lists
 - Tables
 - Forms
- Keyboard navigation
 - Focus management
 - Tab order
 - Shortcuts
 - Skip links

Testing

- Screen readers
 - NVDA
 - JAWS
 - VoiceOver
 - TalkBack
- Tools
 - Lighthouse
 - axe
 - WAVE
 - Color contrast

22.2 Internacionalização (i18n)

Recursos

Tradução

- Strings
 - Text content
 - Labels
 - Messages
 - Placeholders
- Formatação
 - Numbers
 - Dates
 - Currency
 - Units
- Direção
 - LTR
 - RTL
 - Bidirectional

Implementação

- Libraries
 - React-intl
 - i18next
 - Format.js
 - Lingui
- Tools
 - Translation management
 - Language detection
 - Fallbacks
 - Dynamic loading

23. Testes Avançados

23.1 Estratégias de Teste

Níveis

Unit Testing

- Componentes
 - Props
 - State
 - Events
 - Lifecycle

- Funções
 - Pure functions
 - Side effects
 - Async operations
 - Error handling

Integration Testing

- Component interaction
 - Parent-child
 - Siblings
 - Context
 - Redux
- API integration
 - Requests
 - Responses
 - Error handling
 - Loading states

Ferramentas

Testing Libraries

- Jest
 - Matchers
 - Mocks
 - Snapshots
 - Coverage
- React Testing Library
 - Queries
 - Events
 - Async utilities
 - User interactions
- Cypress
 - Commands
 - Assertions
 - Network requests
 - Visual testing

23.2 Quality Assurance

Processos

Code Quality

- Linting
 - ESLint
 - Prettier
 - StyleLint
 - TypeScript
- Code review
 - Standards
 - Best practices
 - Performance
 - Security

Monitoring

- Error tracking
 - Sentry
 - LogRocket
 - Rollbar
 - TrackJS
- Performance
 - Lighthouse
 - Web Vitals
 - Analytics
 - Profiling

[Continua... Há ainda mais conteúdo que posso detalhar. Deseja que eu continue com mais seções ou que aprofunde alguma área específica?]

[continue](#)

24. Desenvolvimento Mobile e Responsivo

24.1 Design Responsivo

Princípios

Layout Fluido

- Grids flexíveis
 - Containers
 - Rows
 - Columns
 - Gaps
- Unidades relativas
 - rem
 - em
 - vw
 - vh
 - %
- Media queries
 - Breakpoints
 - Orientação
 - Resolução
 - Features

Imagens Responsivas

- Técnicas
 - srcset
 - sizes
 - picture element
 - art direction
- Otimização
 - Formatos
 - Compressão
 - Lazy loading
 - Progressive loading

24.2 Mobile-First Development

Estratégias

UX Mobile

- Touch interfaces
 - Gestos
 - Feedback tátil
 - Áreas de toque
 - Scroll
- Performance
 - Tempo de carregamento
 - Tamanho do bundle
 - Cache
 - Offline first

Progressive Enhancement

- Base functionality
 - HTML semântico
 - CSS básico
 - JavaScript essencial
- Advanced features
 - Animações
 - Transições
 - Interações complexas
 - APIs avançadas

25. Segurança Avançada

25.1 Proteção de Dados

Criptografia

Tipos

- Simétrica
 - AES
 - DES
 - 3DES
 - Blowfish
- Assimétrica
 - RSA
 - ECC
 - DSA
 - PGP

Implementação

- Armazenamento
 - Hashing
 - Salting
 - Key derivation
 - Encryption at rest
- Transmissão
 - SSL/TLS
 - HTTPS
 - Perfect forward secrecy
 - Certificate pinning

25.2 Segurança da Aplicação

Proteções

Authentication

- Multi-factor
 - Password
 - Biometric
 - Token
 - SMS/Email
- Session management
 - Token-based
 - Cookie security
 - Session timeout
 - Concurrent sessions

Authorization

- Role-based access
 - Permissions
 - Groups
 - Hierarchies
 - Inheritance
- Policy enforcement
 - Rules
 - Conditions
 - Actions
 - Resources

26. Performance e Otimização

26.1 Frontend Performance

Métricas

Core Web Vitals

- LCP (Largest Contentful Paint)
 - Elementos
 - Otimização
 - Medição
 - Thresholds
- FID (First Input Delay)
 - Interatividade
 - JavaScript
 - Event handlers
 - Task timing
- CLS (Cumulative Layout Shift)
 - Layout stability
 - Visual stability
 - Impact fraction
 - Distance fraction

Outras Métricas

- Time to First Byte
- First Contentful Paint
- Time to Interactive
- Total Blocking Time
- Speed Index

26.2 Backend Performance

Otimizações

Database

- Query optimization

- Indexing
- Query planning
- Caching
- Partitioning
- Connection management
 - Pooling
 - Load balancing
 - Replication
 - Sharding

Server

- Caching strategies
 - Memory cache
 - Disk cache
 - Distributed cache
 - Cache invalidation
- Resource management
 - CPU utilization
 - Memory usage
 - Network I/O
 - Disk I/O

27. DevOps e Infraestrutura

27.1 Containerização

Docker

Componentes

- Images
 - Layers
 - Base images
 - Dockerfile
 - Multi-stage builds
- Containers
 - Lifecycle
 - Networking
 - Volumes
 - Resource limits

Orquestração

- Kubernetes
 - Pods
 - Services
 - Deployments
 - ConfigMaps
 - Secrets
 - Ingress
 - Volumes
 - StatefulSets

27.2 Monitoramento

Sistemas

Métricas

- Application
 - Request rate
 - Error rate
 - Response time
 - Throughput
- Infrastructure
 - CPU
 - Memory
 - Disk
 - Network

Logging

- Centralização
 - ELK Stack
 - Splunk
 - Graylog
 - Papertrail
- Análise
 - Search

- Filters
- Alerts
- Dashboards

28. Arquitetura de Software

28.1 Padrões de Design

Patterns

Criacionais

- Singleton
- Factory Method
- Abstract Factory
- Builder
- Prototype

Estruturais

- Adapter
- Bridge
- Composite
- Decorator
- Facade
- Flyweight
- Proxy

Comportamentais

- Chain of Responsibility
- Command
- Iterator
- Mediator
- Observer
- State
- Strategy
- Template Method

28.2 Arquiteturas Modernas

Tipos

Microserviços

- Características
 - Autonomia
 - Especialização
 - Resiliência
 - Escalabilidade
- Comunicação
 - REST
 - gRPC
 - Message Queue
 - Event Sourcing

Serverless

- Funções
 - Triggers
 - Runtime
 - Cold start
 - Timeout
- Integração
 - API Gateway
 - Database
 - Storage
 - Events

[Continua... Há ainda mais conteúdo que posso detalhar. Deseja que eu continue com mais seções ou que aprofunde alguma área específica?]