



Ciência da Computação

## **INTRODUÇÃO À PROGRAMAÇÃO ESTRUTURADA**

### **RELATÓRIO DE AULAS PRÁTICAS**

Nome: \_\_\_\_\_

RA: \_\_\_\_\_

Polo de matrícula: \_\_\_\_\_

Local da realização da Aula Prática: \_\_\_\_\_

Ano da postagem \_\_\_\_\_

## **TÍTULO DA ATIVIDADE (ROTEIRO OU AULA): Ambiente de programação e saídas**

### **Orientações para o Relatório Final**

Cada aluno deve produzir um relatório curto (1 a 2 páginas) contendo:

- **Resumo teórico:** Explicação sobre o comando `print()` em Python, suas funcionalidades básicas (exibição de strings e variáveis), e as diferentes formas de formatação de saída (parâmetros `sep`, `end`, método `.format()` e f-strings).
- **Código-fonte comentado:** Apresentação do código-fonte completo desenvolvido nas atividades práticas, com comentários claros explicando o funcionamento de cada trecho de código implementado no projeto.
- **Resultados e Discussão:** Capturas de tela do programa em funcionamento e uma breve discussão sobre o que foi aprendido e as dificuldades superadas ao utilizar o comando `print()`.

### **Critérios de Avaliação**

<b>Critério</b>	<b>Peso</b>	<b>Descrição</b>
Qualidade do Resumo Teórico	2,0	Clareza, objetividade e demonstração de entendimento sobre o comando <code>print()</code> e suas funcionalidades.
Estrutura e Organização do Código	3,0	Uso adequado de variáveis, formatação e comentários. Código limpo, legível e bem estruturado.
Funcionamento da Solução	3,0	O programa deve executar corretamente todas as funcionalidades propostas (exibição de dados, formatação, interação com o usuário).
Criatividade e Aprimoramentos	2,0	Inclusão de melhorias que demonstrem domínio do conteúdo (ex.: uso de caracteres de escape, mensagens mais elaboradas).

**Nota Final:** Será a soma dos valores obtidos em cada critério. Alunos ou equipes que não cumprirem os requisitos mínimos de funcionamento do código ou não entregarem o relatório dentro do prazo terão sua nota diminuída proporcionalmente.

## **TÍTULO DA ATIVIDADE (ROTEIRO OU AULA): Variáveis tipo e operações**

### **Orientações para o Relatório Final**

Cada aluno deve produzir um relatório curto (1 a 2 páginas) contendo:

- **Resumo teórico:** Uma explicação clara e concisa sobre os conceitos de variáveis, tipos de dados (`int`, `float`, `str`, `bool`), entrada de dados (`input()`), saída de dados (`print()`) e conversão de tipos (type casting) em Python. Demonstre seu entendimento da teoria abordada na aula.
- **Código-fonte comentado:** O código-fonte completo das soluções desenvolvidas para o "Desafio Inicial" (Consumo Médio e Decomposição de Número) e a "Ampliação do Desafio" (Cálculo de IMC e Conversão Decimal para Binário). Cada trecho de código deve ser acompanhado de comentários explicativos detalhados sobre seu funcionamento e propósito.

### **Critérios de Avaliação**

<b>Critério</b>	<b>Peso</b>	<b>Descrição</b>
Qualidade do Resumo Teórico	2,0	Clareza, objetividade e demonstração de entendimento sobre a teoria abordada na aula (variáveis, tipos, input/output, type casting).
Estrutura e Organização do Código	3,0	Legibilidade, uso de comentários, nomes de variáveis significativos e organização lógica do código.
Funcionamento da Solução	3,0	O código do "Desafio Inicial" e da "Ampliação do Desafio" deve funcionar corretamente, produzindo os resultados esperados.
Criatividade e Aprimoramentos	2,0	Inclusão de melhorias que demonstrem domínio do conteúdo, como validação de entrada, mensagens de erro personalizadas, ou formatação avançada da saída.

**Nota Final:** Será a soma dos valores obtidos em cada critério. Alunos ou equipes que não cumprirem os requisitos mínimos de funcionamento do código ou não entregarem o relatório dentro do prazo terão sua nota diminuída proporcionalmente.

## **TÍTULO DA ATIVIDADE (ROTEIRO OU AULA): Estruturas Condicionais**

### **Orientações para o Relatório Final**

Cada aluno deve produzir um relatório curto (1 a 2 páginas) contendo:

- **Resumo teórico:** Uma explicação clara e concisa sobre as estruturas condicionais (if, else, elif) em Python, seu propósito e como elas controlam o fluxo de execução de um programa.
- **Código-fonte comentado:** O código-fonte completo das soluções desenvolvidas para o "Desafio Inicial" (Verificação de Idade para CNH) e a "Ampliação do Desafio" (Classificação de Triângulos). Cada trecho de código deve ser acompanhado de comentários explicativos detalhados sobre seu funcionamento e propósito.

### **Critérios de Avaliação**

<b>Critério</b>	<b>Peso</b>	<b>Descrição</b>
Qualidade do Resumo Teórico	2,0	Clareza, objetividade e demonstração de entendimento sobre a teoria abordada na aula (estruturas condicionais).
Estrutura e Organização do Código	3,0	Legibilidade, uso de comentários, nomes de variáveis significativos e organização lógica do código.
Funcionamento da Solução	3,0	O código do "Desafio Inicial" e da "Ampliação do Desafio" deve funcionar corretamente, produzindo os resultados esperados.
Criatividade e Aprimoramentos	2,0	Inclusão de melhorias que demonstrem domínio do conteúdo, como validação de entrada robusta, tratamento de casos especiais, ou apresentação mais elaborada dos resultados.

**Nota Final:** Será a soma dos valores obtidos em cada critério. Alunos ou equipes que não cumprirem os requisitos mínimos de funcionamento do código ou não entregarem o relatório dentro do prazo terão sua nota diminuída proporcionalmente.

## **TÍTULO DA ATIVIDADE (ROTEIRO OU AULA): Laços não contados**

### **Orientações para o Relatório Final**

Cada aluno deve produzir um relatório curto (1 a 2 páginas) contendo:

- **Resumo teórico:** Uma explicação clara e concisa sobre o laço de repetição `while` em Python, seu propósito, sintaxe e como ele controla o fluxo de execução de um programa. Inclua também a explicação sobre a estrutura `do-while` (presente em outras linguagens) e as formas de simulá-la em Python.
- **Código-fonte comentado:** O código-fonte completo das soluções desenvolvidas para o "Desafio Inicial" (Imprimindo os Cinco Primeiros Números Pares) e a "Ampliação do Desafio" (Login com Tentativas Limitadas). Cada trecho de código deve ser acompanhado de comentários explicativos detalhados sobre seu funcionamento e propósito.

### **Critérios de Avaliação**

<b>Critério</b>	<b>Peso</b>	<b>Descrição</b>
Qualidade do Resumo Teórico	2,0	Clareza, objetividade e demonstração de entendimento sobre a teoria abordada na aula (laço <code>while</code> e simulação de <code>do-while</code> ).
Estrutura e Organização do Código	3,0	Legibilidade, uso de comentários, nomes de variáveis significativos e organização lógica do código.
Funcionamento da Solução	3,0	O código do "Desafio Inicial" e da "Ampliação do Desafio" deve funcionar corretamente, produzindo os resultados esperados.
Criatividade e Aprimoramentos	2,0	Inclusão de melhorias que demonstrem domínio do conteúdo, como validação de entrada robusta, tratamento de casos especiais, ou apresentação mais elaborada dos resultados.

**Nota Final:** Será a soma dos valores obtidos em cada critério. Alunos ou equipes que não cumprirem os requisitos mínimos de funcionamento do código ou não entregarem o relatório dentro do prazo terão sua nota diminuída proporcionalmente.

## **TÍTULO DA ATIVIDADE (ROTEIRO OU AULA): Laços contados**

### **Orientações para o Relatório Final**

Cada aluno deve produzir um relatório curto (1 a 2 páginas) contendo:

- **Resumo teórico:** Uma explicação clara e concisa sobre o laço for e a função range() em Python, seu propósito, sintaxe e como eles controlam o fluxo de execução de um programa.
- **Código-fonte comentado:** O código-fonte completo das soluções desenvolvidas para o "Desafio Inicial" (Soma dos N Primeiros Números Inteiros) e a "Ampliação do Desafio" (Verificação de Número Primo). Cada trecho de código deve ser acompanhado de comentários explicativos detalhados sobre seu funcionamento e propósito.

### **Critérios de Avaliação**

<b>Critério</b>	<b>Peso</b>	<b>Descrição</b>
Qualidade do Resumo Teórico	2,0	Clareza, objetividade e demonstração de entendimento sobre a teoria abordada na aula (laço for e função range()).
Estrutura e Organização do Código	3,0	Legibilidade, uso de comentários, nomes de variáveis significativos e organização lógica do código.
Funcionamento da Solução	3,0	O código do "Desafio Inicial" e da "Ampliação do Desafio" deve funcionar corretamente, produzindo os resultados esperados.
Criatividade e Aprimoramentos	2,0	Inclusão de melhorias que demonstrem domínio do conteúdo, como validação de entrada robusta, tratamento de casos especiais, ou apresentação mais elaborada dos resultados.

**Nota Final:** Será a soma dos valores obtidos em cada critério. Alunos ou equipes que não cumprirem os requisitos mínimos de funcionamento do código ou não entregarem o relatório dentro do prazo terão sua nota diminuída proporcionalmente.

## **TÍTULO DA ATIVIDADE (ROTEIRO OU AULA): Funções**

### **Orientações para o Relatório Final**

Cada aluno deve produzir um relatório curto (1 a 2 páginas) contendo:

- **Resumo teórico:** Uma explicação clara e concisa sobre o que são funções em Python, por que são importantes (modularidade, reusabilidade, abstração), como são definidas (sintaxe com def, parâmetros) e como o comando return funciona. Inclua a definição de fatorial e como funções podem ser usadas para calculá-lo.
- **Código-fonte comentado:** O código-fonte completo das soluções desenvolvidas para o "Desafio Inicial" (Cálculo de Fatorial) e a "Ampliação do Desafio" (Cálculo de Combinação). Cada trecho de código deve ser acompanhado de comentários explicativos detalhados sobre seu funcionamento, o propósito de cada linha ou bloco, e como as funções foram utilizadas e reutilizadas.

### **Critérios de Avaliação**

<b>Critério</b>	<b>Peso</b>	<b>Descrição</b>
Qualidade do Resumo Teórico	2,0	Clareza, objetividade e demonstração de entendimento sobre a teoria abordada na aula (funções, parâmetros, retorno, fatorial).
Estrutura e Organização do Código	3,0	Legibilidade, uso de comentários explicativos, nomes de funções e variáveis significativos e organização lógica do código.
Funcionamento da Solução	3,0	O código do "Desafio Inicial" e da "Ampliação do Desafio" deve funcionar corretamente, produzindo os resultados esperados para diferentes entradas.
Criatividade e Aprimoramentos	2,0	Inclusão de melhorias que demonstrem domínio do conteúdo, como validação de entrada robusta, tratamento de casos especiais, ou apresentação mais elaborada dos resultados.

**Nota Final:** Será a soma dos valores obtidos em cada critério. Alunos ou equipes que não cumprirem os requisitos mínimos de funcionamento do código ou não entregarem o relatório dentro do prazo terão sua nota diminuída proporcionalmente.

## **TÍTULO DA ATIVIDADE (ROTEIRO OU AULA): Manipulação de Strings**

### **Orientações para o Relatório Final**

Cada aluno deve produzir um relatório curto (1 a 2 páginas) contendo:

- **Resumo teórico:** Uma explicação clara e concisa sobre o que são strings em Python, suas características (imutabilidade, indexação), as operações básicas (+, \*, len()) e os principais métodos de manipulação (replace(), count(), find(), split(), upper(), lower()). Inclua a explicação de ord() e chr() e a importância da tabela ASCII.
- **Código-fonte comentado:** O código-fonte completo das soluções desenvolvidas para o "Desafio Inicial" (Verificador de Palíndromos) e a "Ampliação do Desafio" (Contador de Ocorrências). Cada trecho de código deve ser acompanhado de comentários explicativos detalhados sobre seu funcionamento, o propósito de cada linha ou bloco, e como os conceitos e métodos de string foram aplicados.

### **Critérios de Avaliação**

<b>Critério</b>	<b>Peso</b>	<b>Descrição</b>
Qualidade do Resumo Teórico	2,0	Clareza, objetividade e demonstração de entendimento sobre a teoria abordada na aula (strings, operações, métodos, ASCII).
Estrutura e Organização do Código	3,0	Legibilidade, uso de comentários explicativos, nomes de variáveis significativos e organização lógica do código.
Funcionamento da Solução	3,0	O código do "Desafio Inicial" e da "Ampliação do Desafio" deve funcionar corretamente, produzindo os resultados esperados para diferentes entradas.
Criatividade e Aprimoramentos	2,0	Inclusão de melhorias que demonstrem domínio do conteúdo, como validação de entrada robusta, tratamento de casos especiais, ou apresentação mais elaborada dos resultados.

**Nota Final:** Será a soma dos valores obtidos em cada critério. Alunos ou equipes que não cumprirem os requisitos mínimos de funcionamento do código ou não entregarem o relatório dentro do prazo terão sua nota diminuída proporcionalmente.



## **TÍTULO DA ATIVIDADE (ROTEIRO OU AULA): Coleções**

### **Orientações para o Relatório Final**

Cada aluno deve produzir um relatório curto (1 a 2 páginas) contendo:

- **Resumo teórico:** Uma explicação clara e concisa sobre o que são sequências em Python, com foco em listas e tuplas. Inclua suas características (mutabilidade/imutabilidade, indexação, fatiamento) e os principais métodos de manipulação de listas (adicionar, remover, modificar, buscar, ordenar).
- **Código-fonte comentado:** O código-fonte completo das soluções desenvolvidas para o "Desafio Inicial" (Gerenciador de Tarefas) e a "Ampliação do Desafio" (Análise de Dados de Vendas). Cada trecho de código deve ser acompanhado de comentários explicativos detalhados sobre seu funcionamento, o propósito de cada linha ou bloco, e como os conceitos e métodos de listas foram aplicados.

### **Critérios de Avaliação**

<b>Critério</b>	<b>Peso</b>	<b>Descrição</b>
Qualidade do Resumo Teórico	2,0	Clareza, objetividade e demonstração de entendimento sobre a teoria abordada na aula (sequências, listas, tuplas, operações, métodos).
Estrutura e Organização do Código	3,0	Legibilidade, uso de comentários explicativos, nomes de variáveis significativos e organização lógica do código.
Funcionamento da Solução	3,0	O código do "Desafio Inicial" e da "Ampliação do Desafio" deve funcionar corretamente, produzindo os resultados esperados para diferentes entradas.
Criatividade e Aprimoramentos	2,0	Inclusão de melhorias que demonstrem domínio do conteúdo, como validação de entrada robusta, tratamento de casos especiais, ou apresentação mais elaborada dos resultados.

**Nota Final:** Será a soma dos valores obtidos em cada critério. Alunos ou equipes que não cumprirem os requisitos mínimos de funcionamento do código ou não entregarem o relatório dentro do prazo terão sua nota diminuída proporcionalmente.

## **TÍTULO DA ATIVIDADE (ROTEIRO OU AULA): Dicionário**

### **Orientações para o Relatório Final**

Cada aluno deve produzir um relatório curto (1 a 2 páginas) contendo:

- **Resumo teórico:** Explicação sobre o conceito de dicionários em Python, suas características (mutáveis, ordenados, chave-valor) e sua importância na programação estruturada para organização de dados.
- **Código-fonte comentado:** Apresentação do código-fonte completo do sistema de agenda desenvolvido, com comentários claros explicando o funcionamento de cada função e trecho de código implementado no projeto.
- **Resultados e Discussão:** Capturas de tela do programa em funcionamento (pelo menos uma para cada funcionalidade principal: adicionar, mostrar, remover), e uma breve discussão sobre o que foi aprendido e as dificuldades superadas.

### **Critérios de Avaliação**

<b>Critério</b>	<b>Peso</b>	<b>Descrição</b>
Qualidade do Resumo Teórico	2,0	Clareza, objetividade e demonstração de entendimento sobre o conceito de dicionários e sua aplicação.
Estrutura e Organização do Código	3,0	Uso adequado de funções, variáveis e comentários. Código limpo, legível e bem estruturado.
Funcionamento da Solução	3,0	O sistema de agenda deve executar corretamente todas as funcionalidades propostas (adicionar, mostrar, remover).
Criatividade e Aprimoramentos	2,0	Inclusão de melhorias que demonstrem domínio do conteúdo (ex.: validação de entrada, persistência de dados simples, interface mais amigável).

**Nota Final:** Será a soma dos valores obtidos em cada critério. Alunos ou equipes que não cumprirem os requisitos mínimos de funcionamento do código ou não entregarem o relatório dentro do prazo terão sua nota diminuída proporcionalmente.

## **TÍTULO DA ATIVIDADE (ROTEIRO OU AULA): Arquivo sequencial**

### **Orientações para o Relatório Final**

Cada aluno deve produzir um relatório curto (1 a 2 páginas) contendo:

- **Resumo teórico:** Uma explicação clara e concisa sobre a manipulação de arquivos .txt em Python. Inclua os conceitos de abertura (open()), modos ("r", "w", "a"), fechamento (close()), leitura (readlines()) e escrita (write()), e a importância da codificação (encoding="utf-8").
- **Código-fonte comentado:** O código-fonte completo das soluções desenvolvidas para o "Desafio Inicial" (Sistema Simples de Gerenciamento de Arquivos) e a "Ampliação do Desafio" (Jogo da Forca). Cada trecho de código deve ser acompanhado de comentários explicativos detalhados sobre seu funcionamento, o propósito de cada linha ou bloco, e como os conceitos de manipulação de arquivos foram aplicados.

### **Critérios de Avaliação**

<b>Critério</b>	<b>Peso</b>	<b>Descrição</b>
Qualidade do Resumo Teórico	2,0	Clareza, objetividade e demonstração de entendimento sobre a teoria abordada na aula (manipulação de arquivos, modos, métodos).
Estrutura e Organização do Código	3,0	Legibilidade, uso de comentários explicativos, nomes de funções e variáveis significativos e organização lógica do código.
Funcionamento da Solução	3,0	O código do "Desafio Inicial" e da "Ampliação do Desafio" deve funcionar corretamente, produzindo os resultados esperados para diferentes entradas.
Criatividade e Aprimoramentos	2,0	Inclusão de melhorias que demonstrem domínio do conteúdo, como validação de entrada robusta, tratamento de casos especiais, ou apresentação mais elaborada dos resultados.

**Nota Final:** Será a soma dos valores obtidos em cada critério. Alunos ou equipes que não cumprirem os requisitos mínimos de funcionamento do código ou não entregarem o relatório dentro do prazo terão sua nota diminuída proporcionalmente.

## **TÍTULO DA ATIVIDADE (ROTEIRO OU AULA): Módulos, sys e from...import**

### **Orientações para o Relatório Final**

Cada aluno deve produzir um relatório curto (1 a 2 páginas) contendo:

- **Resumo teórico:** Uma explicação clara e concisa sobre o conceito de módulos em Python, a importância do pip para a instalação de pacotes, a função do módulo sys para interagir com o ambiente, e as diferenças e casos de uso das declarações import e from...import. Inclua uma breve menção aos arquivos .pyc e sua finalidade.
- **Código-fonte comentado:** O código-fonte completo das soluções desenvolvidas para o "Desafio Inicial" (Faker) e a "Ampliação do Desafio" (sys.executable e sys.path). Cada trecho de código deve ser acompanhado de comentários explicativos detalhados sobre seu funcionamento e propósito.

### **Critérios de Avaliação**

<b>Critério</b>	<b>Peso</b>	<b>Descrição</b>
Qualidade do Resumo Teórico	2,0	Clareza, objetividade e demonstração de entendimento sobre a teoria abordada na aula (módulos, pip, sys, importações, pyc).
Estrutura e Organização do Código	3,0	Legibilidade, uso de comentários, nomes de variáveis significativos e organização lógica do código.
Funcionamento da Solução	3,0	O código do "Desafio Inicial" e da "Ampliação do Desafio" deve funcionar corretamente, produzindo os resultados esperados.
Criatividade e Aprimoramentos	2,0	Inclusão de melhorias que demonstrem domínio do conteúdo, como validação de entrada robusta, tratamento de casos especiais, ou apresentação mais elaborada dos resultados.

**Nota Final:** Será a soma dos valores obtidos em cada critério. Alunos ou equipes que não cumprirem os requisitos mínimos de funcionamento do código ou não entregarem o relatório dentro do prazo terão sua nota diminuída proporcionalmente.

## **TÍTULO DA ATIVIDADE (ROTEIRO OU AULA): Conexão com Banco de Dados**

### **Orientações para o Relatório Final**

Cada aluno deve produzir um relatório curto (1 a 2 páginas) contendo:

- **Resumo teórico:** Uma explicação clara e concisa sobre o que é um banco de dados SQLite, suas características (embarcado, sem servidor, arquivo único), a importância do módulo `sqlite3` em Python, e os conceitos das operações SQL básicas (`SELECT`, `INSERT`, `UPDATE`, `DELETE`) aplicadas ao SQLite.
- **Código-fonte comentado:** O código-fonte completo das soluções desenvolvidas para o "Desafio Inicial" (Listagem de Alunos) e a "Ampliação do Desafio" (Sistema de Gerenciamento de Alunos). Cada trecho de código deve ser acompanhado de comentários explicativos detalhados sobre seu funcionamento, o propósito de cada linha ou bloco, e como a interação com o banco de dados foi implementada.

### **Critérios de Avaliação**

<b>Critério</b>	<b>Peso</b>	<b>Descrição</b>
Qualidade do Resumo Teórico	2,0	Clareza, objetividade e demonstração de entendimento sobre a teoria abordada na aula (SQLite, módulo <code>sqlite3</code> , operações SQL).
Estrutura e Organização do Código	3,0	Legibilidade, uso de comentários, nomes de variáveis e funções significativos e organização lógica do código.
Funcionamento da Solução	3,0	O código do "Desafio Inicial" e da "Ampliação do Desafio" deve funcionar corretamente, realizando as operações esperadas no banco de dados SQLite Banco_Dados.db e na tabela Aluno.
Criatividade e Aprimoramentos	2,0	Inclusão de melhorias que demonstrem domínio do conteúdo, como validação de entrada robusta, tratamento de casos especiais, ou apresentação mais elaborada dos resultados.

**Nota Final:** Será a soma dos valores obtidos em cada critério. Alunos ou equipes que não cumprirem os requisitos mínimos de funcionamento do código ou não entregarem o relatório dentro do prazo terão sua nota diminuída proporcionalmente.