
UNIVERSIDADE PAULISTA – UNIP EaD Flex

RELATÓRIO DE AULA PRÁTICA

CURSO: _____.

DISCIPLINA:
_____.

ALUNO(A): _____.

R.A: _____. **POLO:** _____.

DATA: ____ / ____ / ____.

TÍTULO DA ATIVIDADE (ROTEIRO OU AULA): INTRODUÇÃO À LÓGICA DE PROGRAMAÇÃO E PYTHON

ORIENTAÇÕES:

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

Resumo Teórico:

- Explicar, com palavras próprias, o que é lógica de programação e por que ela é importante.
- Mencionar brevemente o que é pseudocódigo e fluxograma e como ajudam na organização de ideias.
- Citar as vantagens de usar Python para aprender programação.

Código-Fonte Comentado:

- Inserir o *código-fonte completo* da atividade proposta.
- Comentar as principais linhas, ressaltando o uso de print(), input(), variáveis etc.

REFERÊNCIAS: O aluno deverá colocar o nome dos livros e sites utilizados para a realização da atividade. As regras para fazer referência ao material utilizado deverão ser de acordo com a ABNT.

CRITÉRIOS PARA AVALIAÇÃO

Critério	Peso	Descrição
Qualidade do Resumo Teórico	2,0	Clareza, objetividade e demonstração de entendimento sobre lógica de programação, pseudocódigo, fluxograma e Python.
Estrutura e Organização do Código Funcionamento da Solução	3,0	O código deve estar indentado corretamente, usar nomes de variáveis adequados e conter comentários informativos (quando necessários).
	3,0	O programa deve rodar sem erros, solicitando informações ao usuário e apresentando as mensagens personalizadas.
Criatividade e Aprimoramentos	2,0	Adição de perguntas extras, uso de strings multilinha, personalização das mensagens e outras melhorias que demonstrem domínio do conteúdo.

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 DE CONTROLE EM PYTHON

ORIENTAÇÕES:

Cada aluno deve produzir um relatório sucinto (2 a 3 páginas) contendo:

Resumo Teórico

- Explicar, com suas palavras, como funcionam as estruturas de decisão (if, else, elif) e repetição (for, while) em Python.
- Mencionar a importância da lógica booleana, operadores e precedência.

Códigos-Fontes Comentados

- Inserir os *códigos completos* das atividades propostas (Desafio A e B, e opcional).
- Adicionar comentários que expliquem a lógica de cada parte relevante.

Breve Reflexão

- Descrever se houve dificuldades.
- Comentar possíveis aplicações práticas (por exemplo, jogos, validações de formulário etc.).

REFERÊNCIAS: O aluno deverá colocar o nome dos livros e sites utilizados para a realização da atividade. As regras para fazer referência ao material utilizado deverão ser de acordo com a ABNT.

CRITÉRIOS PARA AVALIAÇÃO

Critério	Peso	Descrição
Clareza do Resumo Teórico	2,0	Demonstração de entendimento sobre if, else, elif, for, while, lógica booleana e precedência de operadores.
Funcionalidade do Código	3,0	Códigos sem erros de execução, estruturas de Controle funcionando corretamente, tratamento mínimo de entradas inválidas etc.
Organização e Comentários do Código	2,0	Códigos bem indentados, variáveis com nomes adequados, comentários que facilitem a leitura e entendimento.
Criatividade e Completeness	3,0	Adição de funções extras (ex.: uso de break/continue), exemplos mais elaborados, melhorias na interação com o usuário etc.

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): TIPOS DE DADOS E VARIÁVEIS EM PYTHON

ORIENTAÇÕES:

Cada aluno deve produzir um relatório curto (2 a 3 páginas), contendo:

Resumo Teórico:

- Explicar em suas palavras os tipos de dados (int, float, complex, str) e por que variáveis são importantes.
- Descrever brevemente os operadores e suas finalidades.
- Comentar a convenção de nomes (PEP 8) em Python.

Códigos Desenvolvidos:

- Inserir o *código completo* das atividades propostas (Desafios A, B e C, se houver).
- Adicionar comentários essenciais que expliquem a lógica de cada trecho.

Conclusão:

- Descrever se a aula ajudou no entendimento de dados e variáveis.
- Possíveis aplicações práticas no contexto de viagens no tempo, conversão de dados ou manipulação textual.

REFERÊNCIAS: O aluno deverá colocar o nome dos livros e sites utilizados para a realização da atividade. As regras para fazer referência ao material utilizado deverão ser de acordo com a ABNT.

CRITÉRIOS PARA AVALIAÇÃO

Critério	Peso	Descrição
Clareza do Resumo Teórico	2,0	Demonstração de entendimento sobre if, else, elif, for, while, lógica booleana e precedência de operadores.
Funcionalidade do Código	3,0	Códigos sem erros de execução, estruturas de controle funcionando corretamente, tratamento mínimo de entradas inválidas etc.
Organização e Comentários do Código	2,0	Códigos bem indentados, variáveis com nomes adequados, comentários que facilitem a leitura e entendimento.
Criatividade e Completeness	3,0	Adição de funções extras (ex.: uso de break/continue), exemplos mais elaborados, melhorias na interação com o usuário etc.

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 EM PYTHON

ORIENTAÇÕES:

Cada aluno deve produzir um relatório sintético (2 a 3 páginas) contendo:

Resumo Teórico:

- Conceituação de função, parâmetros, retorno.
- Diferença entre variáveis de escopo local e global.

Códigos Desenvolvidos:

- Inserir os *códigos-fonte completos* das atividades (Desafios A e B, e opcionalmente C).
- Comentar as partes relevantes das funções, mostrando claramente a entrada e saída (retorno).

Reflexão:

- Benefícios de usar funções (organização, reuso, clareza).
- Dificuldades encontradas e soluções.

REFERÊNCIAS: O aluno deverá colocar o nome dos livros e sites utilizados para a realização da atividade. As regras para fazer referência ao material utilizado deverão ser de acordo com a ABNT.

CRITÉRIOS PARA AVALIAÇÃO

Critério	Peso	Descrição
Clareza do Resumo Teórico	2,0	Mostra entendimento de como criar e chamar funções, passagem de parâmetros, retorno de valores e escopo de variáveis.
Implementação Correta do Código	3,0	Códigos funcionais, uso adequado de def ..., return, parâmetros, teste de casos básicos (cenários de entrada e saída).
Boas Práticas e Organização	3,0	Legibilidade, indentação, nome de funções e variáveis coerentes, modularidade (funções pequenas e objetivas).
Originalidade / Abrangência	2,0	Abordagem criativa nos desafios, exemplos adicionais, integração com temas anteriores (ex. paradoxos, cálculos de data, etc.).

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 LISTAS E DICIONÁRIOS

ORIENTAÇÕES:

Cada aluno deve produzir um relatório (2 a 3 páginas) contendo:

Resumo Teórico:

- Explicar em suas palavras:
- O que são listas, como funcionam índices, métodos e fatiamento.
- O que são dicionários, como funcionam chaves e valores, principais métodos.

Códigos Desenvolvidos:

- Inserir *códigos completos* dos Desafios A, B (C se houver).
- Comentar partes importantes, destacando como ocorrem inserções, remoções, buscas etc.

Conclusão:

- Experiência prática: o que foi mais fácil, o que foi desafiador, aplicação no contexto de viagens no tempo.

REFERÊNCIAS: O aluno deverá colocar o nome dos livros e sites utilizados para a realização da atividade. As regras para fazer referência ao material utilizado deverão ser de acordo com a ABNT.

CRITÉRIOS PARA AVALIAÇÃO

Critério	Peso	Descrição
Clareza no Resumo Teórico	2,0	Demonstração de entendimento sobre listas (mutabilidade, índices, métodos) e dicionários (mapeamento chave-valor, métodos, uso).
Implementação Correta do Código	3,0	Códigos funcionais, manipulação adequada de listas e dicionários, tratamento básico de exceções ou casos especiais (ex.: remoções).
Organização e Boas Práticas	3,0	Código limpo, indentação correta, nomes de variáveis claríssimos. Fatiamento adequado, uso de métodos e iterações onde necessário.
Criatividade / Completeness	2,0	Soluções originais nos desafios, possibilidade de extensão (ex.: dicionários aninhados), funções extras de consulta ou manipulação.

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): ENTRADA E SAÍDA DE DADOS EM PYTHON

ORIENTAÇÕES:

Cada aluno deve produzir um relatório (2 a 3 páginas) contendo:

Resumo Teórico:

- Conceituar “entrada de dados” (teclado, input, validações) e “saída de dados” (print, formatações).
- Explicar como funciona “leitura/escrita de arquivos” em Python e possíveis modos (r, w, a, json).

Códigos Desenvolvidos:

- Inserir *códigos completos* dos desafios (A, B e opcional C).
- Comentar partes-chave (abertura de arquivo, manipulação de strings, conversões de tipo etc.).

Conclusão:

- Relatar se foi simples ou desafiador manipular E/S.
- Possíveis aplicações práticas (logs, configuração persistente, relatórios etc.).

REFERÊNCIAS: O aluno deverá colocar o nome dos livros e sites utilizados para a realização da atividade. As regras para fazer referência ao material utilizado deverão ser de acordo com a ABNT.

CRITÉRIOS PARA AVALIAÇÃO

Critério	Peso	Descrição
Entendimento do Conceito de E/S	2,0	Clareza ao explicar input, print, leitura e escrita em arquivos, JSON.
Implementação Correta do Código	3,0	Códigos funcionais, uso de try/except (quando necessário), abertura e fechamento de arquivos adequadamente.
Uso de Boas Práticas	3,0	Escopo bem definido, funções para separar lógica, formatações de saída (f-strings), manipulação correta de exceções.
Criatividade / Completeness	2,0	Extensão dos exemplos (ex.: filtrar dados, contagem regressiva, JSON com múltiplas chaves etc.).

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): DEPURAÇÃO E TESTE DE ALGORITMOS

ORIENTAÇÕES:

Cada aluno deve produzir um relatório (2 a 3 páginas) contendo:

Resumo Teórico:

- Diferenciar “depuração” e “teste automatizado”; por que ambos são necessários.
- Ferramentas em Python (pdb, unittest, pytest).

Procedimento Prático:

- Explicar como depurou (ex.: prints, uso de breakpoint() ou debugger IDE).
- Descrever o conjunto de testes criados (testes básicos, cenários, entradas, saídas esperadas).

Resultados:

- Mostrar as falhas (bugs) encontradas, o que foi corrigido.
- Exibir o resultado final dos testes (todos passaram, algum falhou etc.).

Conclusão:

- Reflexão sobre a experiência de localizar e corrigir erros.
- Benefícios de manter testes para garantir estabilidade futura do código.

REFERÊNCIAS: O aluno deverá colocar o nome dos livros e sites utilizados para a realização da atividade. As regras para fazer referência ao material utilizado deverão ser de acordo com a ABNT.

CRITÉRIOS PARA AVALIAÇÃO

Critério	Peso	Descrição
Entendimento de Depuração	2,0	Demonstra uso correto de print, pdb ou debugger IDE para isolar bugs; clareza ao explicar processo de identificar falhas.
Implementação de Testes	3,0	Testes básicos (assert, unittest, pytest) cobrindo as funções principais; coerência nos cenários de teste e checagens adequadas.
Organização e Clareza do Relatório	3,0	Código legível, relatório bem estruturado, descrição clara de bugs, testes, outputs etc.
Correção e Completude das Soluções	2,0	Bugs devidamente corrigidos; testes rodam sem falhas inesperadas; respostas/pontos identificados são coerentes com o problema proposto

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): INTRODUÇÃO À PROGRAMAÇÃO ORIENTADA A OBJETOS (POO) EM PYTHON

ORIENTAÇÕES:

Cada aluno deve produzir um relatório (2 a 3 páginas) contendo:

Resumo Teórico:

- Diferenciar “depuração” e “teste automatizado”; por que ambos são necessários.
- Ferramentas em Python (pdb, unittest, pytest).

Procedimento Prático:

- Explicar como depurou (ex.: prints, uso de breakpoint() ou debugger IDE).
- Descrever o conjunto de testes criados (testes básicos, cenários, entradas, saídas esperadas).

Resultados:

- Mostrar as falhas (bugs) encontradas, o que foi corrigido.
- Exibir o resultado final dos testes (todos passaram, algum falhou etc.).

Conclusão:

- Reflexão sobre a experiência de localizar e corrigir erros.
- Benefícios de manter testes para garantir estabilidade futura do código.

REFERÊNCIAS: O aluno deverá colocar o nome dos livros e sites utilizados para a realização da atividade. As regras para fazer referência ao material utilizado deverão ser de acordo com a ABNT.

CRITÉRIOS PARA AVALIAÇÃO

Entendimento de Depuração	2,0	Demonstra uso correto de print, pdb ou debugger IDE para isolar bugs; clareza ao explicar processo de identificar falhas.
Implementação de Testes	3,0	Testes básicos (assert, unittest, pytest) cobrindo as funções principais; coerência nos cenários de teste e checagens adequadas.
Organização e Clareza do Relatório	3,0	Código legível, relatório bem estruturado, descrição clara de bugs, testes, outputs, etc.
Correção e Completude das Soluções	2,0	Bugs devidamente corrigidos; testes rodam sem falhas inesperadas; respostas/pontos identificados são coerentes com o problema proposto.

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.