

PLANO DE ENSINO												
MÓDULO	Desenvolvimento Back-End com Python											
C/HORÁRIA	TEÓRICA	144H	PRÁTICA	56H	TOTAL	200H						
PROFESSOR(ES)												
EMENTA												
Lógica de Programação com Python. Noções de Banco de Dados. Python e Orientação a Objetos. Padrões de Desenvolvimento de Software. Introdução a WebServices. Django. Soft Skills.Git												
PLANO DE AULAS												
Módulo 1 - Lógica de Programação com Python (10 aulas)												
AULA	OBJETIVO/CONTEÚDOS/METODOLOGIA					PRÁTICA ORIENTADA						
1	<p>Introdução à Computação e Python</p> <p>Objetivo: Apresentar aos alunos os conceitos básicos de computação, a importância da programação e configurar o ambiente Python para iniciar os estudos.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • O que é computação e programação. • Porque aprender Python para Back-End. • Instalação do Python no sistema operacional. • Comandos básicos para iniciar o ambiente de programação. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>					BEP-P001						
2	<p>Variáveis, Tipos de Dados e Entrada de Dados</p> <p>Objetivo: Ensinar os conceitos fundamentais de variáveis, tipos de dados em Python e como receber dados do usuário.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Tipos de dados primitivos. • Diferença entre constantes e variáveis. • Declaração e uso de variáveis. • Entrada e saída de dados. • Casting de dados. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>					BEP -P002						

3	<p>Operadores Aritméticos, Relacionais e Lógicos</p> <p>Objetivo: Apresentar os operadores básicos para manipulação de dados e tomada de decisão em Python.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Operadores aritméticos. • Operadores relacionais. • Operadores lógicos. • Precedência e associação dos operadores. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P003
4	<p>Estruturas Condicionais (if, elif, else)</p> <p>Objetivo: Ensinar como implementar decisões e diferentes fluxos de execução baseados em condições.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Estrutura básica if. • Estrutura básica if - else. • Estrutura básica if – elif - else. • Boas práticas na escrita de condicionais. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P004
5	<p>Estruturas de Repetição (while, for)</p> <p>Objetivo: Apresentar as formas de repetição para executar blocos de código diversas vezes.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Laço while: sintaxe, uso e cuidados para evitar loops infinitos. • Laço for para iteração sobre intervalos e coleções. • Laços aninhados. • Função range(). • Comandos break e continue <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P005
6	Trabalhando com Listas	BEP -P006

	<p>Objetivo: Manipular coleções de dados em listas.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Criação e acesso a listas. • Métodos append(), insert(), remove(). • Fatiamento de listas.. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	
7	<p>Tuplas e Dicionários</p> <p>Objetivo: Conhecer outras estruturas de dados fundamentais.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Definição de tuplas. • Diferença entre tuplas e listas. • Estrutura e uso de dicionários. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P007
8	<p>Funções em Python</p> <p>Objetivo: Ensinar como organizar o código em blocos reutilizáveis (funções).</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Definição e chamada de funções. • Argumentos e parâmetros. • Retorno de valores. • Escopo de variáveis. <p>Metodologia: Aula expositiva, aplicação de exercícios</p>	BEP -P008
9	<p>Manipulação de Strings</p> <p>Objetivo: Trabalhar com textos de maneira eficiente.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Operações básicas com strings.. • Métodos úteis: lower(), upper(), strip(), split(), join(). • Formatação de strings. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P009

10	<p>Tratamento de Erros e Exceções</p> <p>Objetivo: Ensinar como identificar, capturar e tratar erros para tornar programas robustos.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • O que são exceções. • Bloco try-except. • Lançando exceções com raise. • Tratamento de exceções comuns. • Uso de finally. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P010
Módulo 2 – Noções de Banco de Dados (6 aulas)		
11	<p>Introdução a Banco de Dados Relacionais</p> <p>Objetivo: Apresentar os conceitos fundamentais de bancos de dados relacionais, sua importância e aplicações no desenvolvimento back-end.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • O que é um banco de dados. • Diferença entre banco de dados relacional e não relacional. • Conceitos básicos: tabela, linha, coluna, chave primária, chave estrangeira. • Modelo entidade-relacionamento (ER). • Introdução ao sistema gerenciador de banco de dados (SGBD). • Exemplos de SGBDs populares: MySQL, PostgreSQL, SQLite. • Exercícios de reconhecimento e interpretação de tabelas simples. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P011
12	<p>Introdução a SQL – Comandos Básicos (SELECT, INSERT)</p> <p>Objetivo: Ensinar a realizar consultas básicas e inserção de dados utilizando SQL, a linguagem padrão para bancos relacionais.</p> <p>Conteúdo:</p>	BEP -P012

	<ul style="list-style-type: none"> • O que é SQL. • Comando SELECT: seleção de dados em tabelas. • Filtros com FROM, WHERE, ORDER BY, LIMIT. • Inserção de dados com INSERT INTO. • Visualização prática com SQLite ou outro SGBD. • Exercícios simples para escrever comandos SELECT e INSERT. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	
13	<p>Comandos de Atualização e Remoção (UPDATE, DELETE)</p> <p>Objetivo: Capacitar os alunos a modificar e remover dados armazenados em tabelas de banco de dados.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Comando UPDATE: como atualizar registros existentes. • Cláusula WHERE para seleção de registros a atualizar. • Comando DELETE: remoção de dados. • Cuidados e boas práticas para evitar perda de dados. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P013
14	<p>Modelagem e Normalização de Dados</p> <p>Objetivo: Ensinar os alunos a estruturar dados de forma eficiente para evitar redundância e inconsistência.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Conceitos de modelagem de dados. • Normalização: 1^a, 2^a e 3^a formas normais. • Impacto da modelagem na performance e integridade dos dados. • Exercícios para identificar e normalizar tabelas simples. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P014
15	Joins e Consultas Avançadas	BEP -P015

	<p>Objetivo: Integrar dados de múltiplas tabelas.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • INNER JOIN, LEFT JOIN, RIGHT JOIN. • Subconsultas. • Ordenação e agrupamento de dados (ORDER BY, GROUP BY). <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	
16	<p>Conectando a Banco de Dados</p> <p>Objetivo: Introduzir os alunos à integração de linguagens de programação com bancos de dados</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Conceito de banco de dados embarcado. • Biblioteca nativa (como SQLite3): instalação e uso básico. • Criação de tabelas, inserção e consultas básicas via linguagem de programação. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P016
Módulo 3 – Programação e Orientação a Objetos (6 aulas)		
17	<p>Fundamentos de Programação Orientada a Objetos (POO)</p> <p>Objetivo: Introduzir os conceitos básicos de POO para alunos iniciantes e mostrar sua importância na organização de código.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • O que é programação orientada a objetos. • Comparação entre programação procedural e orientada a objetos. • Conceitos-chave: objeto, classe, atributo, método. • Exemplos práticos simples. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P017
18	<p>Criando e Instanciando Classes</p> <p>Objetivo: Implementar e utilizar classes.</p> <p>Conteúdo:</p>	BEP -P018

	<ul style="list-style-type: none"> • Definição de classes. • Métodos construtores (init). • Instanciação de objetos. • Exercícios para criação de classes e objetos básicos. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	
19	<p>Encapsulamento e Métodos de Instância</p> <p>Objetivo: Ensinar o conceito de encapsulamento e a criação de métodos de instância para manipulação de dados dentro de objetos.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Definição e importância do encapsulamento. • Modificadores de acesso: public, private, protected. • Criando métodos de instância para leitura e alteração de atributos. • Uso de getters e setters. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P019
20	<p>Herança e Polimorfismo</p> <p>Objetivo: Explicar os conceitos de herança e polimorfismo para reutilização e extensão de código.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Conceito de herança e suas vantagens. • Criando classes pai e subclasses. • Sobrescrita de métodos (override). • Polimorfismo: métodos que se comportam de maneira diferente conforme o objeto. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P020
21	<p>Composição e Associação entre Objetos</p> <p>Objetivo: Apresentar como criar relações entre objetos para modelar sistemas complexos.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Diferença entre composição e associação. • Exemplos práticos de associação entre classes. 	BEP -P021

	<ul style="list-style-type: none"> Exercícios para modelagem orientada a objetos com associação e composição. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	
22	<p>Tratamento de Exceções em Programação Orientada a Objetos</p> <p>Objetivo: Melhorar a segurança e robustez do código.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> Exceções customizadas. Tratamento com try-except dentro de classes. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P022
Módulo 4 – Padrões de Desenvolvimento de Software (6 aulas)		
23	<p>Introdução aos Padrões de Projeto (Design Patterns)</p> <p>Objetivo: Apresentar o conceito de padrões de projeto, sua importância para a organização do código e reutilização, e os tipos mais comuns.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> O que são padrões de projeto (design patterns). Benefícios do uso de padrões no desenvolvimento de software. Classificação dos padrões: criacionais, estruturais e comportamentais. Exemplos práticos de onde aplicar padrões em projetos reais. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P023
24	<p>Padrão Singleton</p> <p>Objetivo: Ensinar a implementar o padrão Singleton para garantir que uma classe tenha apenas uma instância.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> Conceito e finalidade do Singleton. Vantagens e possíveis armadilhas. Casos práticos de uso. 	BEP -P024

	<ul style="list-style-type: none"> • Exercícios para criar uma classe Singleton. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	
25	<p>Padrão Factory Method</p> <p>Objetivo: Explicar o padrão Factory Method para criação de objetos sem expor a lógica de criação ao cliente.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Conceito do Factory Method. • Diferenças entre Factory Method e Simple Factory. • Vantagens e exemplos práticos. • Exercícios para criar factories para diferentes objetos. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P025
26	<p>Padrão Observer</p> <p>Objetivo: Ensinar o padrão Observer para implementar comunicação entre objetos de forma desacoplada.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Conceito do padrão Observer. • Quando utilizar Observer. • Uso em eventos e notificações. • Exercícios práticos para criar observadores e sujeitos. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P026
27	<p>Padrão Decorator</p> <p>Objetivo: Apresentar o padrão Decorator para adicionar funcionalidades a objetos dinamicamente.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Definição e motivação do Decorator. • Diferença entre herança e composição via decorator. • Exemplos práticos e vantagens. • Exercícios para decorar objetos com funcionalidades adicionais. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P027

	Introdução ao MVC e Padrões em Rails Objetivo: Introduzir o padrão MVC (Model-View-Controller), arquitetura utilizada no framework Ruby on Rails. Conteúdo: <ul style="list-style-type: none">• O que é MVC e sua importância.• Componentes Model, View e Controller e suas responsabilidades.• Como padrões facilitam a organização e manutenção do código.• Exercícios para identificar e aplicar MVC em projetos pequenos. Metodologia: Aula expositiva, aplicação de exercícios.	
Módulo 5 – Introdução a WebServices (6 aulas)		
28	Conceitos Fundamentais de WebServices Objetivo: Apresentar o que são WebServices, sua importância na comunicação entre sistemas distribuídos, e os principais tipos existentes. Conteúdo: <ul style="list-style-type: none">• Definição e função de WebServices.• Tipos de WebServices: SOAP vs REST.• Protocolos usados (HTTP, HTTPS).• Formatos de dados: XML, JSON.• Exemplos práticos de uso em aplicações reais. Metodologia: Aula expositiva, aplicação de exercícios.	BEP -P028
29	RESTful Web Services – Fundamentos e Métodos HTTP Objetivo: Ensinar os princípios do REST e os métodos HTTP utilizados para manipular recursos em APIs. Conteúdo: <ul style="list-style-type: none">• Conceitos REST (Representational State Transfer).• Métodos HTTP: GET, POST, PUT, DELETE, PATCH.• Status HTTP comuns (200, 404, 500).• Estrutura de URLs RESTful.	BEP -P029
30	RESTful Web Services – Fundamentos e Métodos HTTP Objetivo: Ensinar os princípios do REST e os métodos HTTP utilizados para manipular recursos em APIs. Conteúdo: <ul style="list-style-type: none">• Conceitos REST (Representational State Transfer).• Métodos HTTP: GET, POST, PUT, DELETE, PATCH.• Status HTTP comuns (200, 404, 500).• Estrutura de URLs RESTful.	BEP -P030

	<ul style="list-style-type: none"> • Exercícios de modelagem de recursos REST. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	
31	<p>Criando uma API REST Simples</p> <p>Objetivo: Implementar uma API REST básica para entender a construção e funcionamento.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Setup do ambiente API. • Criando rotas simples. • Manipulação de requisições e respostas HTTP. • Enviando dados em JSON. • Testes básicos com ferramentas como Postman ou curl. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P031
32	<p>Autenticação e Autorização em WebServices</p> <p>Objetivo: Introduzir os conceitos de segurança em APIs, focando em autenticação e autorização.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Diferença entre autenticação e autorização. • Métodos comuns: Basic Auth, Token Auth, JWT. • Exemplos de implementação simples. • Importância da segurança para APIs. • Exercícios práticos de autenticação simples. <p>Bibliografia:</p> <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P032
33	<p>Manipulação de Erros e Logs em WebServices</p> <p>Objetivo: Ensinar a tratar erros adequadamente em APIs e registrar logs para monitoramento.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Tipos comuns de erros em WebServices. • Como retornar respostas de erro padronizadas. • Logs: o que registrar e como. • Ferramentas básicas para logging. 	BEP -P033

	<ul style="list-style-type: none"> Exercícios de simulação e tratamento de erros. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	
34	<p>Introdução a WebSockets e Comunicação em Tempo Real</p> <p>Objetivo: Apresentar o conceito de WebSockets e seu uso para comunicação bidirecional em tempo real.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> Diferença entre HTTP e WebSockets. Casos de uso: chat, notificações, atualizações em tempo real. Implementação básica. Exercício prático: criar um canal de comunicação simples. <p>Bibliografia:</p> <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P034

Módulo 6 - Django (8 aulas)

AULA	OBJETIVO/CONTEÚDOS/METODOLOGIA	PRÁTICA ORIENTADA
35	<p>Introdução ao Django</p> <p>Objetivo: Compreender o framework Django e sua arquitetura.</p> <p>Conteúdos:</p> <ul style="list-style-type: none"> Conceito de MTV. Instalação de Django. Ambientação. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P035
36	<p>Modelos e Migrations</p> <p>Objetivo: Modelar dados com Django ORM.</p> <p>Conteúdos:</p> <ul style="list-style-type: none"> Definição de models. Criação e execução de migrations. 	BEP -P036

	<ul style="list-style-type: none"> • Admin automático. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	
37	<p>Views e Templates</p> <p>Objetivo: Criar páginas dinâmicas utilizando views e templates.</p> <p>Conteúdos:</p> <ul style="list-style-type: none"> • Funções de view. • Templates HTML com Django Template Language (DTL). • Contexto e renderização. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P037
38	<p>URLs e Rotas no Django</p> <p>Objetivo: Configurar rotas e URLs no Django.</p> <p>Conteúdos:</p> <ul style="list-style-type: none"> • Configuração de URLConf. • Namespaces e reverses. • Incluindo múltiplos apps. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P038
39	<p>Formulários e Validação de Dados</p> <p>Objetivo: Manipular formulários com segurança no Django.</p> <p>Conteúdos:</p> <ul style="list-style-type: none"> • Formulários via Django Forms. • Validação e limpeza de dados. • Mensagens de erro. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P039
40	<p>Autenticação e Autorização</p> <p>Objetivo: Implementar sistemas de autenticação de usuários.</p> <p>Conteúdos:</p>	BEP -P040

	<ul style="list-style-type: none"> • Sistema de autenticação embutido. • Login, logout e restrição de acesso. • Customização de usuários. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	
41	<p>Testes Automatizados</p> <p>Objetivo: Introduzir o conceito de testes e ensinar como criar testes automatizados básicos.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Importância dos testes. • Testes unitários, de integração e funcionais. • Frameworks de teste no Django (como Pytest). • Criar testes simples. • Executar testes e interpretar resultados. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P041
42	<p>Deploy de Aplicações Django</p> <p>Objetivo: Publicar aplicações Django em produção.</p> <p>Conteúdos:</p> <ul style="list-style-type: none"> • Preparação para produção. • Configuração de servidor (Gunicorn, Nginx). • Plataformas de deploy gratuito (Heroku, Render). <p>Metodologia: Aula expositiva, aplicação de exercícios</p>	BEP -P042
Módulo 7 - Soft Skills (3 aulas)		
AULA	OBJETIVO/CONTEÚDOS/METODOLOGIA	PRÁTICA ORIENTADA
43	<p>Comunicação Efetiva e Trabalho em Equipe</p> <p>Objetivo: Desenvolver habilidades de comunicação claras e eficazes para trabalhar em times de desenvolvimento.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Fundamentos da comunicação interpessoal. • Feedback construtivo e escuta ativa. • Trabalho colaborativo e resolução de conflitos. • Ferramentas para colaboração remota. 	BEP -P043

	Metodologia: Aula expositiva, aplicação de exercícios.	
44	<p>Gestão do Tempo e Produtividade</p> <p>Objetivo: Ensinar técnicas práticas para organizar o tempo e aumentar a produtividade pessoal e em equipe.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> Importância da gestão do tempo no desenvolvimento de software. Técnicas: Pomodoro, matriz de Eisenhower, GTD (Getting Things Done). Priorização de tarefas e definição de metas SMART. Ferramentas digitais para organização (Trello, Todoist). Evitando procrastinação e distrações. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P044
45	<p>Pensamento Crítico e Resolução de Problemas</p> <p>Objetivo: Desenvolver a habilidade de análise crítica e técnicas para resolver problemas complexos em desenvolvimento.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> O que é pensamento crítico. Técnicas para análise e decomposição de problemas. Brainstorming e mind mapping. Tomada de decisão baseada em dados. Exercícios práticos com problemas típicos de programação. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P045
Módulo 8 - Git (3 aulas)		
AULA	OBJETIVO/CONTEÚDOS/METODOLOGIA	PRÁTICA ORIENTADA
46	<p>Introdução ao Git e Controle de Versão</p> <p>Objetivo: Apresentar os conceitos básicos de controle de versão e iniciar o uso do Git para controle de código.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> O que é controle de versão e sua importância. Conceitos: repositório, commit, branch, merge. 	BEP -P046

	<ul style="list-style-type: none"> • Instalação e configuração inicial do Git. • Criando o primeiro repositório local. • Fluxo básico de trabalho com commits. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	
47	<p>Fluxo de Trabalho com Branches e GitHub</p> <p>Objetivo: Ensinar a usar branches para organizar o desenvolvimento e como colaborar com GitHub.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Criação e troca de branches. • Merge e resolução de conflitos. • Introdução ao GitHub e criação de repositórios remotos. • Push, pull e pull requests. • Boas práticas para colaboração em equipe. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P047
48	<p>Fluxos de Trabalho com Git</p> <p>Objetivo: Desenvolver fluxos de trabalho eficientes com Git, incluindo boas práticas para times de desenvolvimento.</p> <p>Conteúdo:</p> <ul style="list-style-type: none"> • Fluxo Git básico: feature branch, merge, pull request. • Git Flow: desenvolvimento com múltiplos branches. • Brainstorming e mind mapping. • Resolução de conflitos de merge. • Estratégias para commits claros e históricos limpos. <p>Metodologia: Aula expositiva, aplicação de exercícios.</p>	BEP -P048
AVALIAÇÃO		
<ul style="list-style-type: none"> - 30% participação ativa nas aulas presenciais, participação nos fóruns de discussão - 30% entregas realizadas (cumprimento de prazos, adequação das entregas realizadas) - 40% avaliação final 		
REFERÊNCIAS BIBLIOGRÁFICAS		
<ul style="list-style-type: none"> • Python.org - Getting Started (https://www.python.org/about/gettingstarted/) • W3Schools - Python Variables (https://www.w3schools.com/python/python_variables.asp) 		

- Programiz - Python Input, Output (<https://www.programiz.com/python-programming/input-output>)
- Python Docs - Expressions (<https://docs.python.org/3/reference/expressions.html>)
- Real Python - Conditional Statements (<https://realpython.com/python-conditional-statements/>)
- W3Schools - Python Loops (https://www.w3schools.com/python/python_for_loops.asp)
- Programiz - Python Lists (<https://www.programiz.com/python-programming/list>)
- W3Schools - Python Dictionaries (https://www.w3schools.com/python/python_dictionaries.asp)
- Real Python - Defining Functions (<https://realpython.com/defining-your-own-python-function/>)
- Python Docs - Text Sequence Type str (<https://docs.python.org/3/library/stdtypes.html#text-sequence-type-str>)
- Tutorial Banco de Dados Relacional (vídeo e texto): <https://www.freecodecamp.org/news/relational-databases-for-beginners/>
- Curso básico de banco de dados - SQL (em português): <https://www.sqlitetutorial.net/pt/>
- SQL Tutorial para iniciantes (W3Schools): https://www.w3schools.com/sql/sql_select.asp
- SQL Introduction - Mode Analytics: <https://mode.com/sql-tutorial/introduction-to-sql/>
- Guia UPDATE e DELETE (W3Schools): https://www.w3schools.com/sql/sql_update.asp
- SQL UPDATE Tutorial (TutorialsPoint): <https://www.tutorialspoint.com/sql/sql-update-query.htm>
- Guia de Normalização de Banco de Dados (em português): <https://www.devmedia.com.br/normalizacao-de-banco-de-dados/28469>
- Data Normalization Tutorial: <https://www.studytonight.com/dbms/database-normalization.php>
- Python Docs - sqlite3 module (<https://docs.python.org/3/library/sqlite3.html>)
- Guia ActiveRecord para iniciantes: https://guides.rubyonrails.org/active_record_basics.html
- Programiz - Python SQLite (<https://www.programiz.com/python-programming/sqlite>)
- SQLZoo - Joins (https://sqlzoo.net/wiki/SQL_Joins)
- SQLite Tutorial - Joins (<https://www.sqlitetutorial.net/sqlite-inner-join/>)
- Programiz - Python OOP (<https://www.programiz.com/python-programming/object-oriented-programming>)
- W3Schools - Python Classes/Objects (https://www.w3schools.com/python/python_classes.asp)
- Real Python - OOP in Python (<https://realpython.com/python3-object-oriented-programming/>)
- Python Docs - Data Model (<https://docs.python.org/3/reference/datamodel.html>)
- W3Schools - Python Inheritance (https://www.w3schools.com/python/python_inheritance.asp)
- Python Docs - abc module (<https://docs.python.org/3/library/abc.html>)
- Programiz - Python OOP Examples (<https://www.programiz.com/python-programming/object-oriented-programming>)

- Design Patterns Overview: <https://refactoring.guru/design-patterns/what-is-pattern>
- Refactoring Guru - Singleton (<https://refactoring.guru/pt-br/design-patterns/singleton>)
- Refactoring Guru - Factory Method (<https://refactoring.guru/pt-br/design-patterns/factory-method>)
- Refactoring Guru - Observer (<https://refactoring.guru/pt-br/design-patterns/observer>)
- MVC Pattern Explained: <https://www.geeksforgeeks.org/mvc-design-pattern/>
- Web Services Basics: <https://www.tutorialspoint.com/webservices/index.htm>
- RESTful API Introduction: <https://restfulapi.net/>
- REST API Tutorial: <https://restfulapi.net/rest-architecture/>
- HTTP Methods: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>
- Building REST API with Ruby: <https://dev.to/diablopixels/building-a-rest-api-with-ruby-part-1-3h1f>
- JSON Handling in Ruby: <https://ruby-doc.org/stdlib-2.7.1/libdoc/json/rdoc/JSON.html>
- API Authentication Guide: <https://developer.okta.com/blog/2018/10/31/simple-api-authentication-with-ruby>
- JWT Basics: <https://jwt.io/introduction/>
- API Error Handling: <https://www.restapitutorial.com/httpstatuscodes.html>
- WebSocket Basics: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API
- Devise GitHub: <https://github.com/heartcombo/devise>
- Devise Tutorial: <https://www.digitalocean.com/community/tutorials/how-to-add-authentication-to-your-app-with-devise>
- Postman Docs (<https://learning.postman.com/docs/getting-started/introduction/>)
- Swagger Docs (<https://swagger.io/docs/>)
- Django Docs - Getting Started (<https://docs.djangoproject.com/en/stable/intro/tutorial01/>)
- Django Project - Overview (<https://docs.djangoproject.com/en/stable/misc/design-philosophies/>)
- Django Models Docs (<https://docs.djangoproject.com/en/stable/topics/db/models/>)
- Django Templates (<https://docs.djangoproject.com/en/stable/ref/templates/>)
- Django Forms (<https://docs.djangoproject.com/en/stable/topics/forms/>)
- Django Auth Docs (<https://docs.djangoproject.com/en/stable/topics/auth/>)
- Django Deployment Checklist (<https://docs.djangoproject.com/en/stable/howto/deployment/checklist/>)
- Guia de Comunicação Eficaz: <https://www.mindtools.com/CommSkill/CommunicationIntro.htm>
- Trabalho em Equipe: <https://www.coursera.org/learn/teamwork-skills-effective-communication>
- Gestão do Tempo: https://www.mindtools.com/pages/article/newHTE_00.htm
- Técnicas de Produtividade: <https://todoist.com/productivity-methods>
- Pensamento Crítico: <https://www.skillsyouneed.com/learn/critical-thinking.html>

Conecta e Capacita

- Resolução de Problemas:
https://www.mindtools.com/pages/main/newMN_TMC.htm
- Pro Git Book: <https://git-scm.com/book/en/v2>
- Git Tutorial: <https://www.atlassian.com/git/tutorials/what-is-version-control>
- Git Branching: <https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell>
- GitHub Guide: <https://guides.github.com/activities/hello-world/>
- Git Flow: <https://nvie.com/posts/a-successful-git-branching-model/>
- Atlassian Git Workflow: <https://www.atlassian.com/git/tutorials/comparing-workflows>