

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	<b>Introdução à Computação e Python</b>  <b>Objetivo:</b> Apresentar aos alunos os conceitos básicos de computação, a importância da programação e configurar o ambiente Python para iniciar os estudos. <b>Conteúdo:</b> <ul style="list-style-type: none"><li>O que é computação e programação.</li><li>Porque aprender Python para Back-End.</li><li>Instalação do Python no sistema operacional.</li><li>Comandos básicos para iniciar o ambiente de programação.</li></ul> <b>Metodologia:</b> Aula expositiva, aplicação de exercícios.					BEP-P001
2	<b>Variáveis, Tipos de Dados e Entrada de Dados</b>  <b>Objetivo:</b> Ensinar os conceitos fundamentais de variáveis, tipos de dados em Python e como receber dados do usuário. <b>Conteúdo:</b> <ul style="list-style-type: none"><li>Tipos de dados primitivos.</li><li>Diferença entre constantes e variáveis.</li><li>Declaração e uso de variáveis.</li><li>Entrada e saída de dados.</li><li>Casting de dados.</li></ul> <b>Metodologia:</b> Aula expositiva, aplicação de exercícios.					BEP -P002

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

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

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

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

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

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

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



25	<p><b>Padrão Factory Method</b></p> <p><b>Objetivo:</b> Explicar o padrão Factory Method para criação de objetos sem expor a lógica de criação ao cliente.</p> <p><b>Conteúdo:</b></p> <ul style="list-style-type: none"> <li>• Conceito do Factory Method.</li> <li>• Diferenças entre Factory Method e Simple Factory.</li> <li>• Vantagens e exemplos práticos.</li> <li>• Exercícios para criar factories para diferentes objetos.</li> </ul> <p><b>Metodologia:</b> Aula expositiva, aplicação de exercícios.</p>	BEP -P025
26	<p><b>Padrão Observer</b></p> <p><b>Objetivo:</b> Ensinar o padrão Observer para implementar comunicação entre objetos de forma desacoplada.</p> <p><b>Conteúdo:</b></p> <ul style="list-style-type: none"> <li>• Conceito do padrão Observer.</li> <li>• Quando utilizar Observer.</li> <li>• Uso em eventos e notificações.</li> <li>• Exercícios práticos para criar observadores e sujeitos.</li> </ul> <p><b>Metodologia:</b> Aula expositiva, aplicação de exercícios.</p>	BEP -P026
27	<p><b>Padrão Decorator</b></p> <p><b>Objetivo:</b> Apresentar o padrão Decorator para adicionar funcionalidades a objetos dinamicamente.</p> <p><b>Conteúdo:</b></p> <ul style="list-style-type: none"> <li>• Definição e motivação do Decorator.</li> <li>• Diferença entre herança e composição via decorator.</li> <li>• Exemplos práticos e vantagens.</li> <li>• Exercícios para decorar objetos com funcionalidades adicionais.</li> </ul> <p><b>Metodologia:</b> Aula expositiva, aplicação de exercícios.</p>	BEP -P027
28	<p><b>Introdução ao MVC e Padrões em Rails</b></p> <p><b>Objetivo:</b> Introduzir o padrão MVC (Model-View-Controller), arquitetura utilizada no framework Ruby on</p>	BEP -P028

	<p>Rails.</p> <p><b>Conteúdo:</b></p> <ul style="list-style-type: none"> <li>• O que é MVC e sua importância.</li> <li>• Componentes Model, View e Controller e suas responsabilidades.</li> <li>• Como padrões facilitam a organização e manutenção do código.</li> <li>• Exercícios para identificar e aplicar MVC em projetos pequenos.</li> </ul> <p><b>Metodologia:</b> Aula expositiva, aplicação de exercícios.</p>	
<b>Módulo 5 – Introdução a WebServices (6 aulas)</b>		
29	<p><b>Conceitos Fundamentais de WebServices</b></p> <p><b>Objetivo:</b> Apresentar o que são WebServices, sua importância na comunicação entre sistemas distribuídos, e os principais tipos existentes.</p> <p><b>Conteúdo:</b></p> <ul style="list-style-type: none"> <li>• Definição e função de WebServices.</li> <li>• Tipos de WebServices: SOAP vs REST.</li> <li>• Protocolos usados (HTTP, HTTPS).</li> <li>• Formatos de dados: XML, JSON.</li> <li>• Exemplos práticos de uso em aplicações reais.</li> </ul> <p><b>Metodologia:</b> Aula expositiva, aplicação de exercícios.</p>	BEP -P029
30	<p><b>RESTful Web Services – Fundamentos e Métodos HTTP</b></p> <p><b>Objetivo:</b> Ensinar os princípios do REST e os métodos HTTP utilizados para manipular recursos em APIs.</p> <p><b>Conteúdo:</b></p> <ul style="list-style-type: none"> <li>• Conceitos REST (Representational State Transfer).</li> <li>• Métodos HTTP: GET, POST, PUT, DELETE, PATCH.</li> <li>• Status HTTP comuns (200, 404, 500).</li> <li>• Estrutura de URLs RESTful.</li> <li>• Exercícios de modelagem de recursos REST.</li> </ul> <p><b>Metodologia:</b> Aula expositiva, aplicação de exercícios.</p>	BEP -P030

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

34	<b>Introdução a WebSockets e Comunicação em Tempo Real</b>  <b>Objetivo:</b> Apresentar o conceito de WebSockets e seu uso para comunicação bidirecional em tempo real. <b>Conteúdo:</b> <ul style="list-style-type: none"> <li>Diferença entre HTTP e WebSockets.</li> <li>Casos de uso: chat, notificações, atualizações em tempo real.</li> <li>Implementação básica.</li> <li>Exercício prático: criar um canal de comunicação simples.</li> </ul> <b>Bibliografia:</b>  <b>Metodologia:</b> Aula expositiva, aplicação de exercícios.	BEP -P034
<b>Módulo 6 - Django (8 aulas)</b>		
<b>AULA</b>	<b>OBJETIVO/CONTEÚDOS/METODOLOGIA</b>	<b>PRÁTICA ORIENTADA</b>
35	<b>Introdução ao Django</b>  <b>Objetivo:</b> Compreender o framework Django e sua arquitetura. <b>Conteúdos:</b> <ul style="list-style-type: none"> <li>Conceito de MTV.</li> <li>Instalação de Django.</li> <li>Ambientação.</li> </ul> <b>Metodologia:</b> Aula expositiva, aplicação de exercícios.	BEP -P035
36	<b>Modelos e Migrations</b>  <b>Objetivo:</b> Modelar dados com Django ORM. <b>Conteúdos:</b> <ul style="list-style-type: none"> <li>Definição de models.</li> <li>Criação e execução de migrations.</li> <li>Admin automático.</li> </ul> <b>Metodologia:</b> Aula expositiva, aplicação de exercícios.	BEP -P036
37	<b>Views e Templates</b>	BEP -P037

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

41	<b>Testes Automatizados</b>  <b>Objetivo:</b> Introduzir o conceito de testes e ensinar como criar testes automatizados básicos. <b>Conteúdo:</b> <ul style="list-style-type: none"> <li>• Importância dos testes.</li> <li>• Testes unitários, de integração e funcionais.</li> <li>• Frameworks de teste no Django (como Pytest).</li> <li>• Criar testes simples.</li> <li>• Executar testes e interpretar resultados.</li> </ul> <b>Metodologia:</b> Aula expositiva, aplicação de exercícios.	BEP -P041
42	<b>Deploy de Aplicações Django</b>  <b>Objetivo:</b> Publicar aplicações Django em produção. <b>Conteúdos:</b> <ul style="list-style-type: none"> <li>• Preparação para produção.</li> <li>• Configuração de servidor (Gunicorn, Nginx).</li> <li>• Plataformas de deploy gratuito (Heroku, Render).</li> </ul> <b>Metodologia:</b> Aula expositiva, aplicação de exercícios	BEP -P042
<b>Módulo 7 - Soft Skills (3 aulas)</b>		
<b>AULA</b>	<b>OBJETIVO/CONTEÚDOS/METODOLOGIA</b>	<b>PRÁTICA ORIENTADA</b>
43	<b>Comunicação Efetiva e Trabalho em Equipe</b>  <b>Objetivo:</b> Desenvolver habilidades de comunicação claras e eficazes para trabalhar em times de desenvolvimento. <b>Conteúdo:</b> <ul style="list-style-type: none"> <li>• Fundamentos da comunicação interpessoal.</li> <li>• Feedback construtivo e escuta ativa.</li> <li>• Trabalho colaborativo e resolução de conflitos.</li> <li>• Ferramentas para colaboração remota.</li> </ul> <b>Metodologia:</b> Aula expositiva, aplicação de exercícios.	BEP -P043
44	<b>Gestão do Tempo e Produtividade</b>  <b>Objetivo:</b> Ensinar técnicas práticas para organizar o tempo e	BEP -P044

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

47	<p><b>Fluxo de Trabalho com Branches e GitHub</b></p> <p><b>Objetivo:</b> Ensinar a usar branches para organizar o desenvolvimento e como colaborar com GitHub.</p> <p><b>Conteúdo:</b></p> <ul style="list-style-type: none"> <li>• Criação e troca de branches.</li> <li>• Merge e resolução de conflitos.</li> <li>• Introdução ao GitHub e criação de repositórios remotos.</li> <li>• Push, pull e pull requests.</li> <li>• Boas práticas para colaboração em equipe.</li> </ul> <p><b>Metodologia:</b> Aula expositiva, aplicação de exercícios.</p>	BEP -P047
48	<p><b>Fluxos de Trabalho com Git</b></p> <p><b>Objetivo:</b> Desenvolver fluxos de trabalho eficientes com Git, incluindo boas práticas para times de desenvolvimento.</p> <p><b>Conteúdo:</b></p> <ul style="list-style-type: none"> <li>• Fluxo Git básico: feature branch, merge, pull request.</li> <li>• Git Flow: desenvolvimento com múltiplos branches.</li> <li>• Brainstorming e mind mapping.</li> <li>• Resolução de conflitos de merge.</li> <li>• Estratégias para commits claros e históricos limpos.</li> </ul> <p><b>Metodologia:</b> Aula expositiva, aplicação de exercícios.</p>	BEP -P048
<b>AVALIAÇÃO</b>		
<p>- 30% participação ativa nas aulas presenciais, participação nos fóruns de discussão</p> <p>- 30% entregas realizadas (cumprimento de prazos, adequação das entregas realizadas)</p> <p>- 40% avaliação final</p>		
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>		
<ul style="list-style-type: none"> <li>• Python.org - Getting Started (<a href="https://www.python.org/about/gettingstarted/">https://www.python.org/about/gettingstarted/</a>)</li> <li>• W3Schools - Python Variables (<a href="https://www.w3schools.com/python/python_variables.asp">https://www.w3schools.com/python/python_variables.asp</a>)</li> <li>• Programiz - Python Input, Output (<a href="https://www.programiz.com/python-programming/input-output">https://www.programiz.com/python-programming/input-output</a>)</li> <li>• Python Docs - Expressions (<a href="https://docs.python.org/3/reference/expressions.html">https://docs.python.org/3/reference/expressions.html</a>)</li> <li>• Real Python - Conditional Statements (<a href="https://realpython.com/python-conditional-statements/">https://realpython.com/python-conditional-statements/</a>)</li> <li>• W3Schools - Python Loops</li> </ul>		



- ([https://www.w3schools.com/python/python\\_for\\_loops.asp](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](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](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](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](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](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](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](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>)

- [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](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](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>
- Resolução de Problemas: [https://www.mindtools.com/pages/main/newMN\\_TMC.htm](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/>



## Conecta e Capacita

- Git Flow: <https://nvie.com/posts/a-successful-git-branching-model/>
- Atlassian Git Workflow: <https://www.atlassian.com/git/tutorials/comparing-workflows>