

PLANO DE ENSINO

CURSO		MÓDULO	CÓDIGO
Tecnólogo em Análise e Desenvolvimento de Sistemas		Básico	
UNIDADE CURRICULAR	CARGA HORÁRIA PREVISTA	DOCENTE	TURMA
Linguagem de Programação	80	Luciano Santos - Michel Chaparro	
OBJETIVO DA UNIDADE CURRICULAR			
<ul style="list-style-type: none">Programar sistemas computacionais multicamadas, tendo em vista a aplicação em ambientes industriais.			

FUNDAMENTOS TÉCNICOS E CIENTÍFICOS ou CAPACIDADES TÉCNICAS

- Utilizar lógica de programação para estruturação de programas funcionais;
- Elaborar códigos de programação na composição de estruturas computacionais;
- Aplicar paradigmas de programação de acordo com as tecnologias de mercado;
- Utilizar as ferramentas de desenvolvimento (IDEs) para construção dos códigos de programação;
- Aplicar práticas de versionamento em repositórios remotos e locais, de acordo com as políticas corporativas estabelecidas;
- Criar aplicações computacionais para cliente/servidor considerando a necessidade do projeto;
- Testar o código com ferramentas automatizadas para redução de erros da aplicação computacional;
- Documentar os sistemas computacionais criados para viabilização da manutenção do código;
- Entregar o sistema conforme pipeline CI/CD, tendo em vista o atendimento as necessidades do projeto;

CAPACIDADES SOCIAIS, ORGANIZATIVAS E METODOLÓGICAS

- Demonstrar visão crítica.
- Demonstrar atenção a detalhes.
- Demonstrar capacidade de síntese.
- Demonstrar capacidade de solucionar problemas.
- Demonstrar capacidade de tomar decisão.
- Demonstrar organização.
- Demonstrar visão sistêmica.

CONHECIMENTOS

1 - Linguagem de programação

- Definição
- Características
- Ambientes programação (IDE)
- Compiladores e interpretadores
- Sintaxe e semântica
- Documentação

2 - Paradigmas de programação

- Declarativos
- Imperativos

3 - Variáveis e tipos de dados

- Caracteres
- Números inteiros e reais
- Vazio, nula e indefinidas
- Booleano

4 - Funções, sub-rotinas e exceções

5 - Estrutura de dados

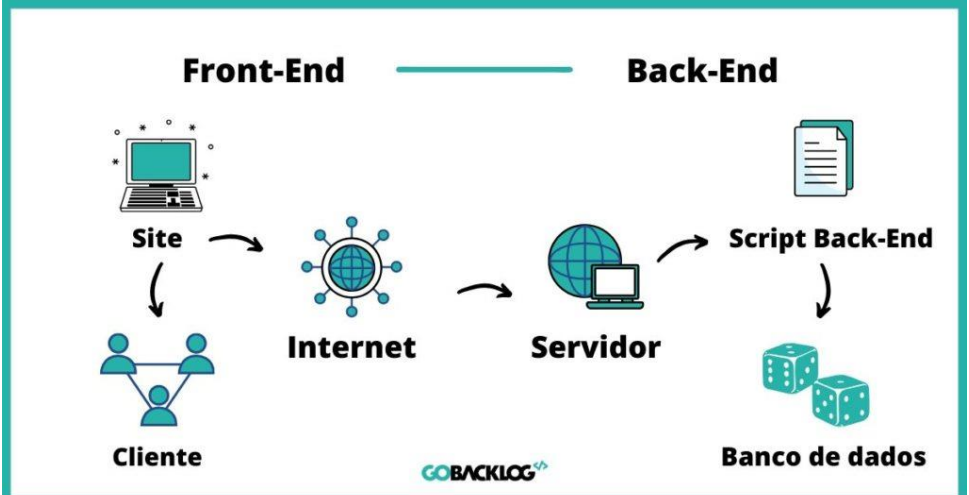

- Cadeias de caracteres (strings)
- Listas (arrays)
- Tuplas
- Dicionários
- Conjuntos

6 - Estruturas de controle

- De seleção
- De repetição
- Contadores e acumuladores

7 - Testes

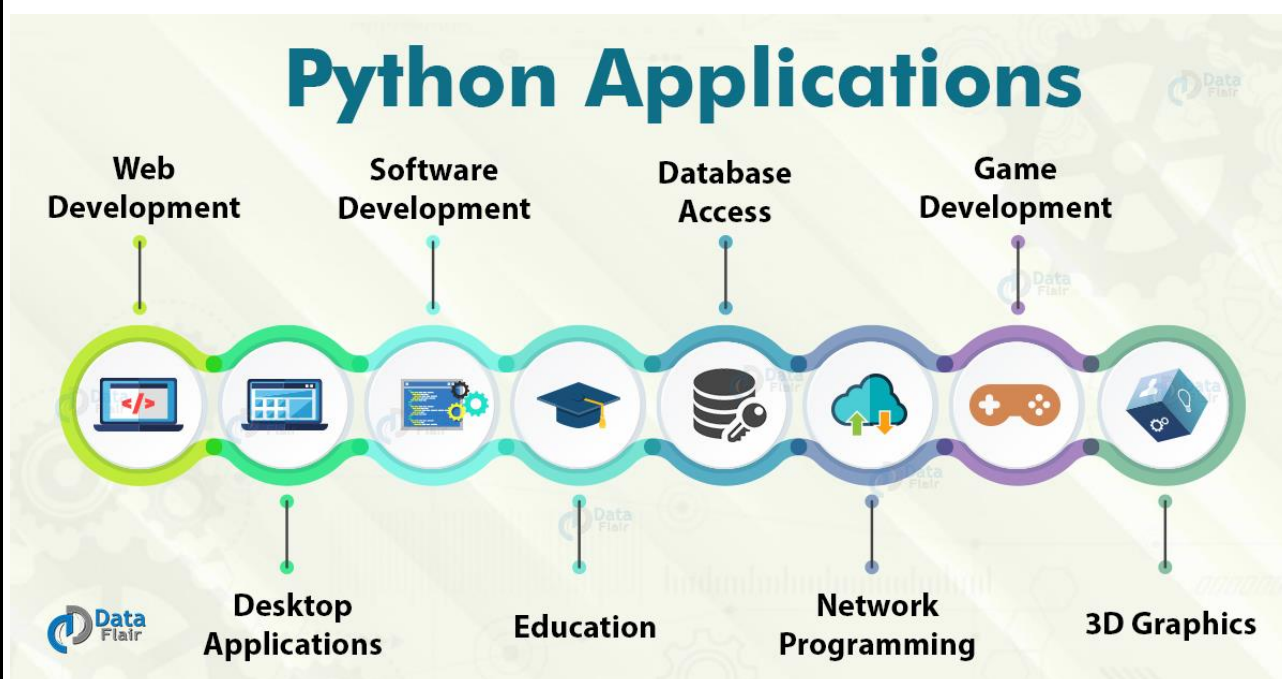
- Unitários
- Integrados

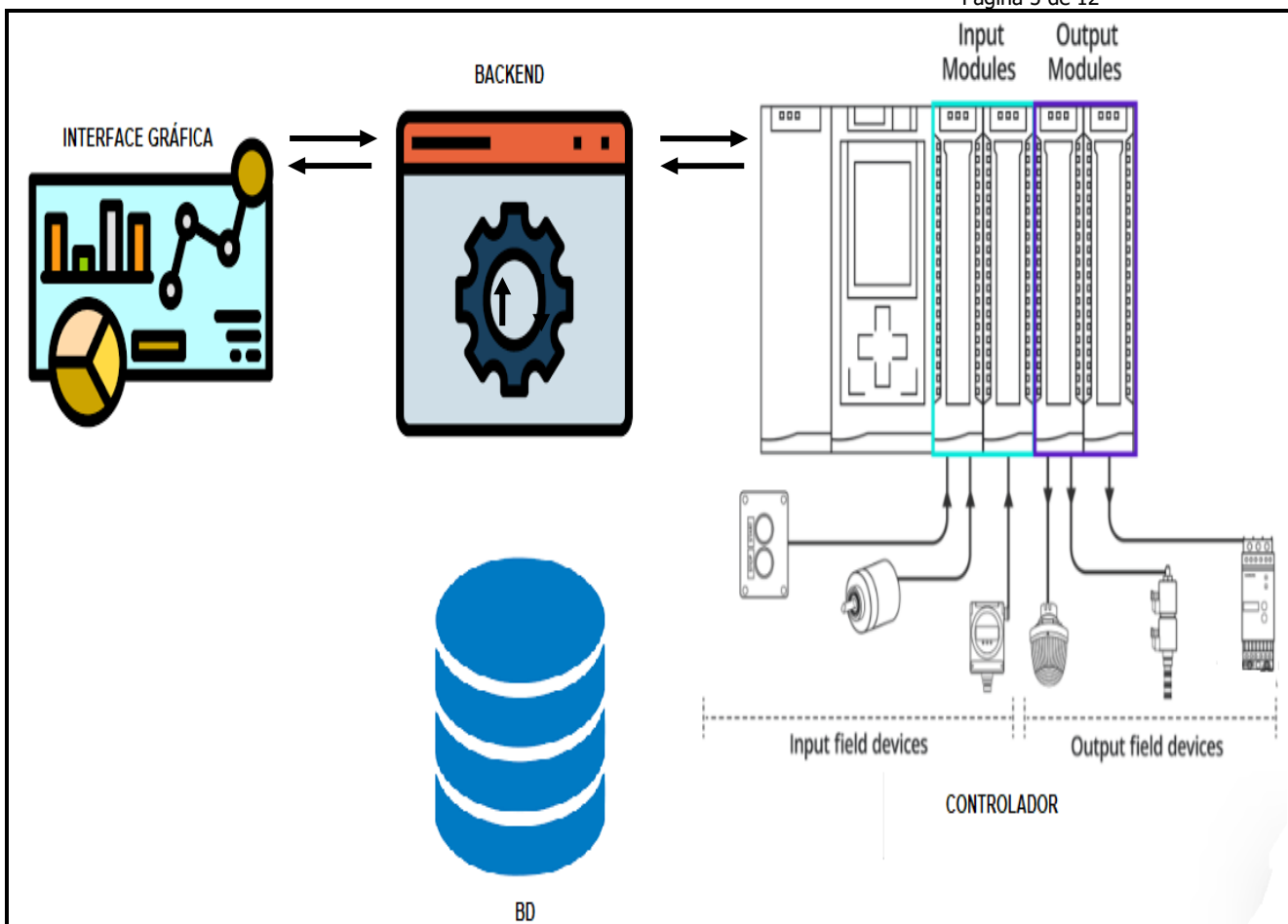
ATIVIDADE	SITUAÇÃO PROBLEMA
CONTEXTO	
<p>Uma pequena industria de processos térmicos pretende automatizar alguns passos de seu processo, com o interesse de aumentar a produção, melhorar a qualidade dos produtos, diminuir retrabalhos e ter a possibilidade de rastrear lotes,.</p>	
<p>A sua empresa foi contratada para realizar a programação de Frontend e Backend do processo de implementação dessas melhorias.</p>	
	
<p>A parte inicial do processo consiste no back-end da aplicação através de uma API. Atualmente, existem inúmeras linguagens de programação server-side, isto é, que atuam no servidor na camada Back-end:</p>	
	

É incorreto afirmar que existe alguma melhor, pois, isto depende de cada situação e aplicação, e portanto, é de crucial importância que se saiba quais são as principais diferenças de cada linguagem e qual a ideal para o escopo do projeto desejado, como por exemplo:

- Agilidade no desenvolvimento;
- Velocidade no processamento;
- Atual conhecimento da linguagem;
- Capacidade da linguagem em realizar os requisitos do projeto;
- Arquitetura do servidor que hospedará a linguagem;
- Ciclo de vida da linguagem;

Contudo a linguagem de programação PYTHON tem apresentado uma crescente e massiva utilização nos últimos tempos, é mais usada no backend, mas também pode ser aplicada em frontend. Além disso é uma das linguagens de programação com uma curva de aprendizagem mais acentuada e a existência de inúmeras bibliotecas de auxílio.

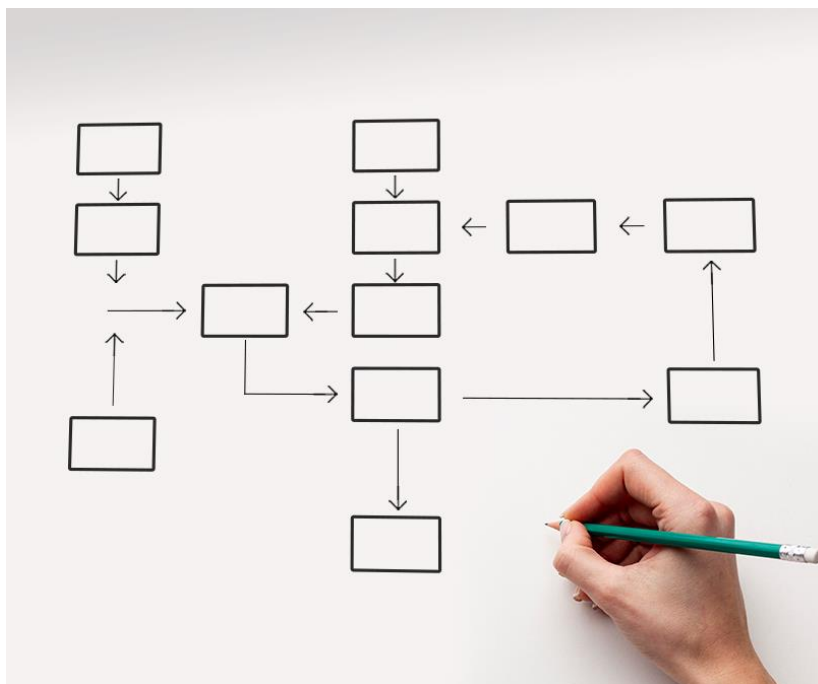




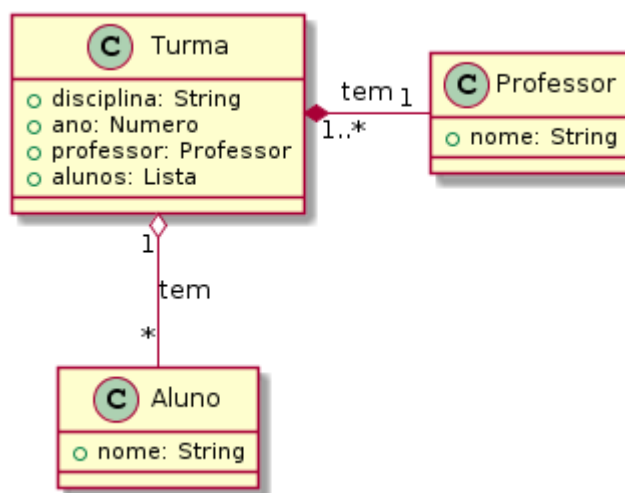
A sua função é criar um server-side, típico processo de um serviço backend, onde a partir do recebimento de um código e quantidade de peças a serem fabricadas, seu script em linguagem de programação Python começa a automatizar os processos, conforme:

- Com o código da peça a ser fabricado e sua quantidade o serviço procura um banco de dados local para fornecer o tempo e a temperatura que a peça deve permanecer em processo de têmpera.
- O serviço então passa para a interface de controle do sistema (PLCs ou microcontroladores) o tempo e temperatura do processo;
- A partir do término do processo o sistema de controle envia a data de fabricação e o número do lote que deve ser armazenado no banco de dados pelo server-side;
- O nosso backend também deve prover uma dashboard para os dados de produção do lote, como a data de processamento, quantidade produzida, tipo de peça na forma de gráficos que melhor se associem ao processo;

- Criar também um fluxograma do processo



- Criar também um diagrama UML ou diagramas de classes para ajudar na programação de classes e objetos se existirem na sua programação;



INSTRUMENTO DE REGISTRO DE AVALIAÇÃO

Natureza dos Critérios	Fundamentos Técnicos e Científicos ou Capacidades Técnicas	Critérios de avaliação	Alunos							
Competências Técnicas	Utilizar lógica de programação para de estruturação de programas funcionais;	Consegue criar métodos ou funções para particionar e separar suas tarefas e ações no programa?	F							
			S							
		Consegue criar funções ou métodos com parâmetros ou argumentos?	F							
			S							
		Criou métodos ou funções com retorno?	F							
			S							
	Elaborar códigos de na de programação composição estruturas computacionais;	Substituiu linhas repetidas por estruturas de repetição?	F							
			S							
		Aplicou controle de continuidade ou saída das estruturas de repetição?	F							
			S							
		Escolheu as estruturas de repetição para ler ou escrever vetores?	F							
			S							
		Combinou estruturas de decisão dentro das de repetição para separar grupos de dados?	F							
			S							
	Aplicar paradigmas de programação de acordo com as tecnologias de mercado;	Reconhece os paradigmas de programação imperativo e funcional nos seus programas?	F							
			S							
		Em orientação a objetos	F							

		Página 6 de 12									
		consegue reconhecer objetos e classes?	S								
	Utilizar as ferramentas de desenvolvimento (IDEs) para construção dos códigos de programação;	Instalou IDE usando atalhos e endentação?	F								
			S								
		Definiu a melhor IDE com base em recurso?	F								
			S								
	Aplicar práticas de versionamento em repositórios remotos e locais, de acordo com as políticas corporativas estabelecidas;	Criou conta no GI	F								
			S								
		Instalou o GIT BASH e iniciou o diretório remoto	F								
			S								
		Criou diretórios locais e remotos	F								
			S								
		Realizou commits e PUSH de versões do sistema em andamento	F								
		S									
Competências Técnicas	Criar aplicações computacionais para cliente/servidor considerando a necessidade do projeto;	Consegue escrever e ler do banco de dados locais	F								
			S								
		A API consegue ler e escrever os dados da interface de comunicação	F								
			S								
		Consegue fazer a comunicação com o bloco frontend da API.	F								
			S								
	Testar o código com ferramentas automatizadas para redução de erros da aplicação computacional;	Usou ferramentas de verificação e de análise de código, refazendo o código e fazendo seu correto versionamento	F								
			S								
	Documentar os sistemas computacionais criados para viabilização da manutenção do código;	Usou as técnicas de documentação de software para fazer a documentação da API do projeto	F								
			S								
	Entregar o sistema conforme pipeline CI/CD, tendo em vista o atendimento as	Aplicou as práticas de pipeline para compilação, entrega, lançamento e implantação do	F								
			S								

PLANO DE AULA (Estratégia de ensino e aprendizagem)		
CONHECIMENTOS	ESTRATÉGIAS	MEDIAÇÃO
Apresentação da Unidade Curricular; Apresentação da Situação Problema Integradora	Exposição Dialogada, demonstração de código;	Quais as diferenças entre linguagens estruturada e POO? Qual a melhor abordagem entre as duas?
Entendendo mais a fundo a Situação Problema Integradora	Exposição Dialogada, apresentação de exemplos;	Quais os problemas existentes na situação problema que o paradigma OO pode solucionar?
Software de codificação;	Exposição Dialogada com Projetor, resolução de exercícios, demonstração e simulação de código, Hands-On	Em sua opinião, para criar uma página web somente é possível com uma IDE? Quais as IDE's que você conhece relacionadas à programação?
Obtenção de dados.	Exposição Dialogada com Projetor, resolução de exercícios, demonstração e criação de BOTS, Hands-On	Posso fazer raspagem em todas a páginas web? Como saber o que é permitido ou não?
Linguagem de Programação Orientada objeto	Exposição Dialogada, resolução de exercícios, Hands-On, Demonstração e codificação;	Como sei se estou programando em Estruturada ou POO? Como idênticas candidatas a métodos? Como sei se uma classe pode ser genérica ou especializada? Quando devo criar heranças de métodos ou implementá-los? Por que devo usar interface para conectar funcionalidade de classes?
Versionamento	Exposição Dialogada, resolução de exercícios, Hands-On, criação de repositório de código fonte no GIT;	O que é o GITHUB Por que devo manter meu código em um repositório online? O que é compartilhar o código com uma equipe? O acontece se eu perder meu código local? Quais os benefícios de um repositório desde perfil?

ATIVIDADE	SITUAÇÃO DE APRENDIZAGEM SOMATIVA: Linguagem de Programação
CONTEXTO	
<p>Após sua última entrega parcial, conforme o escopo acordado, você recebeu importantes feedbacks positivos, demonstrando que realmente conseguiu finalizar seu aplicativo, contudo, o projeto ainda não está concluído, você deve atender todos os requisitos do projeto:</p>	
<ul style="list-style-type: none">- Para isso elabore uma rotina para cumprir os passos constantes da pirâmide de teste e validação do software conforme especificado na figura anterior.- Elaborar toda a documentação exigida para a validação;- Executar ferramenta para análise e qualidade de código e fazer as modificações reportadas, inclusive o versionamento local e remoto;- Executar o programa de teste;	

NÍVEIS DE DESEMPENHO	NÍVEIS	NOTA
Atendeu todos os critérios críticos (21) e todos os desejáveis (7)	10	100
Atendeu todos os critérios críticos (21) e 6 desejáveis	9	90
Atendeu todos os critérios críticos (21) e 5 desejáveis	8	80
Atendeu todos os critérios críticos (21) e entre 3 e 4 desejáveis	7	70
Atendeu todos os critérios críticos (21) e entre 1 e 2 desejáveis	6	60
Atendeu todos os critérios críticos (21) e nenhum desejável	5	50
Atendeu entre 20 e 13 critérios críticos e qualquer quantidade de desejáveis	4	40
Atendeu entre 9 e 12 critérios críticos e qualquer quantidade de desejáveis	3	30
Atendeu entre 5 e 8 critérios críticos e qualquer quantidade de desejáveis	2	20
Abaixo de 5 critérios críticos	1	10

NÍVEL MÍNIMO DE DESEMPENHO ESPERADO	5
--	----------

ELABORAÇÃO	DATA	APROVAÇÃO	DATA
Luciano Santos / Michel Chaparro	12 / 12 / 2022		/ /