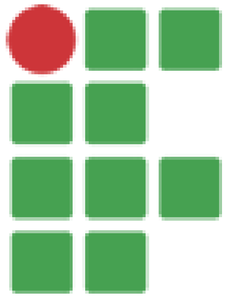


INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Disciplina

Modelagem de Domínio e Conceitos de Orientação a Objetos

PROF. FERNANDO TAMBERLINI ALVES

Como desenvolver um software?

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

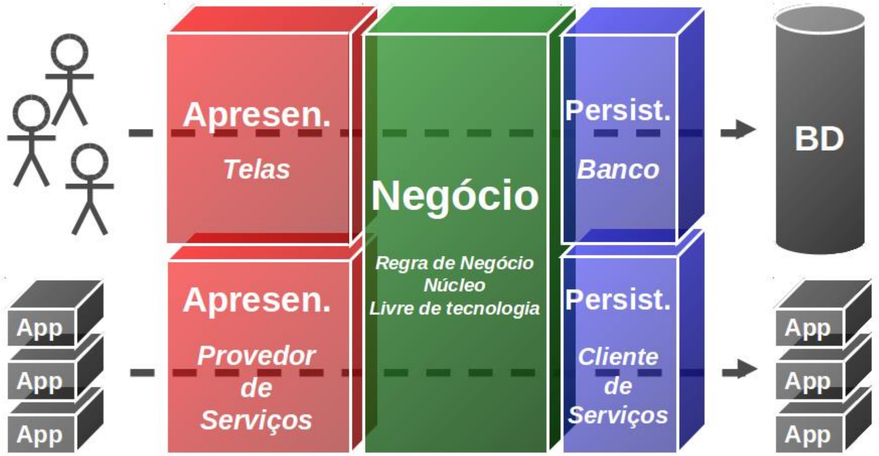
- Quais são as etapas, os métodos, as ferramentas e linguagem (arquitetura) necessária para desenvolver um software?

Eng. De Software

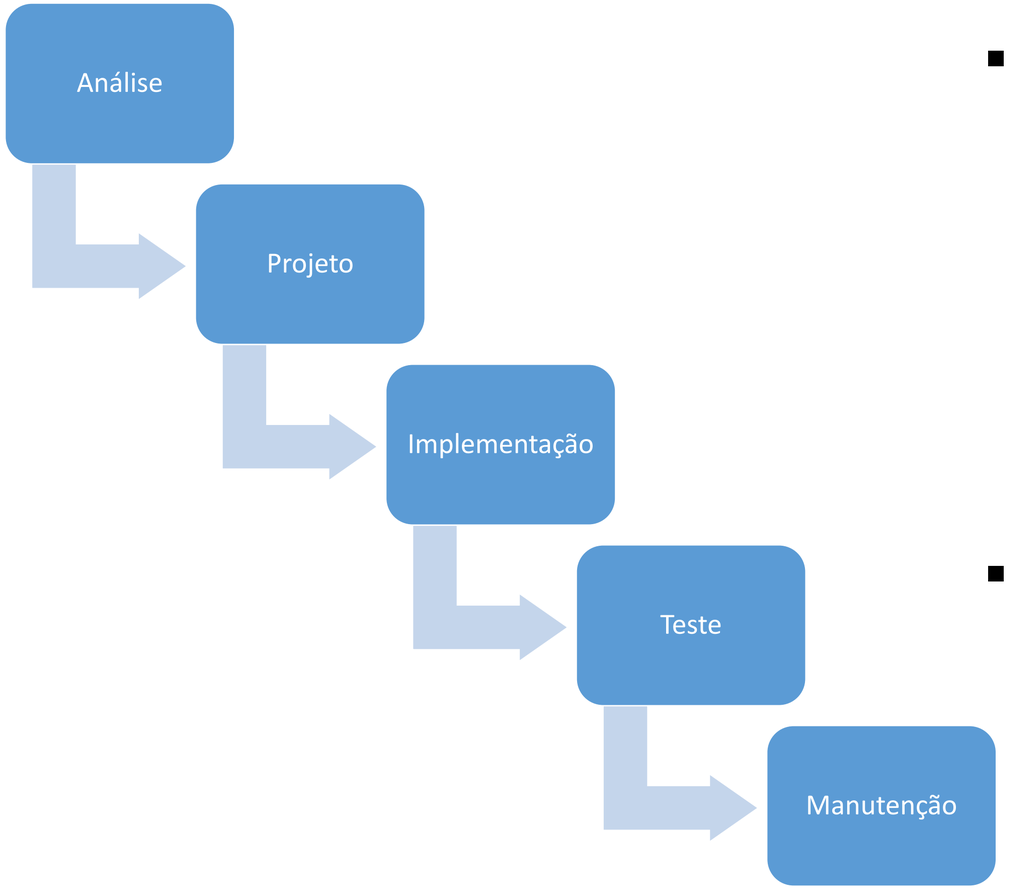
INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- A dificuldade de especificar a execução de tarefas simples
- A Engenharia de Software como agrupamento de técnicas, métodos e ferramentas de apoio ao processo de desenvolvimento de Sistema de Informação (SI)
- A Arquitetura de Software como a definição das características de como se desenvolverá um Sistema de Informação (SI)
- A descrição do problema a ser atacado pelo Sistema de Informação.
- O processo de levantamentos dos requisitos de um Sistema de Informação.

Arquitetura... Mais uma analogia



Modelo de Cascata



- O modelo de cascata é um dos modelos de desenvolvimento de software mais antigos e tradicionais. Ele segue uma **abordagem linear e sequencial**, dividindo o processo de desenvolvimento em fases distintas, como análise, projeto, implementação, teste e manutenção.
- Cada fase deve ser concluída antes de passar para a próxima, e as mudanças nos requisitos são difíceis de acomodar após o início da fase de implementação

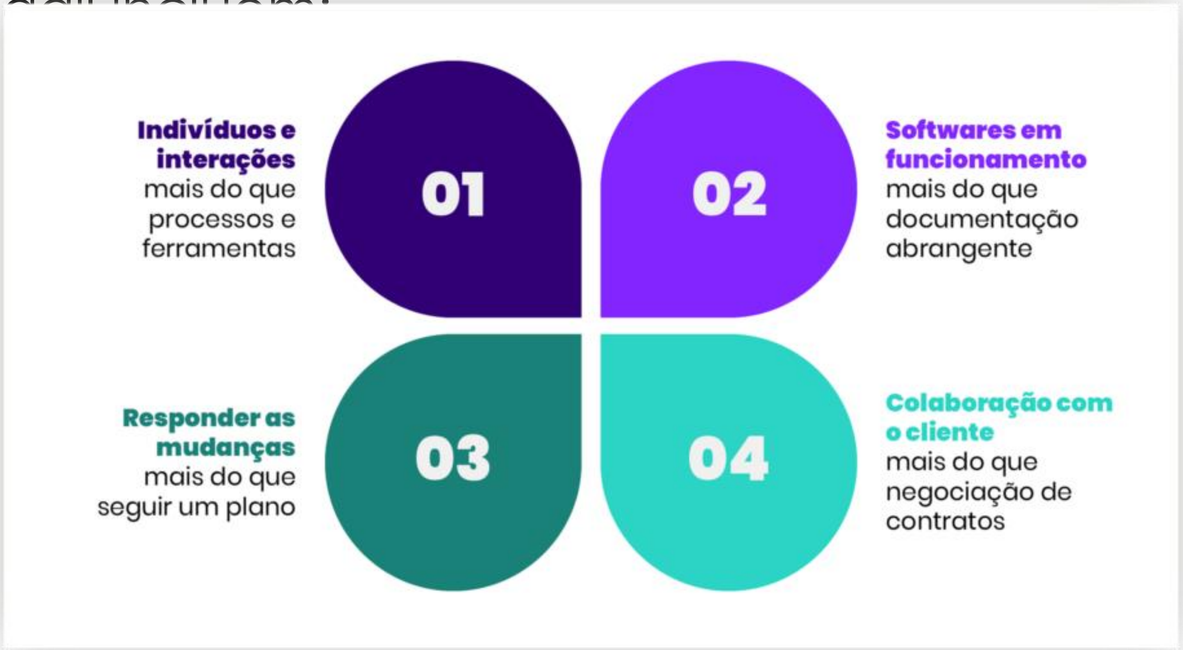
Modelo de Desenvolvimento Ágil

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- O modelo ágil no desenvolvimento de software é uma abordagem moderna e flexível que se concentra na entrega contínua de software funcional, **de modo interativo e incremental**, e na colaboração próxima com os clientes. Algumas das metodologias ágeis mais conhecidas incluem Scrum, Kanban e Extreme Programming (XP).

- As principais características do modelo ágil incluem:

  - Iterações e Incrementos
  - Colaboração
  - Flexibilidade
  - Entrega Contínua
  - Auto-organização
  - Teste Contínuo
  - Transparência.



Modelagem de Domínio

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- A modelagem de domínio no desenvolvimento de software é uma prática crucial que envolve a criação de representações abstratas e estruturadas dos conceitos, entidades, regras de negócios e relacionamentos dentro de um determinado domínio de negócios ou área de aplicação.

- Essa técnica ajuda a equipe de desenvolvimento a compreender profundamente o contexto em que o software será usado, o que, por sua vez, facilita a criação de soluções eficazes e alinhadas com as necessidades dos usuários e das partes interessadas.

Modelagem de Domínio

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Aqui estão alguns aspectos importantes da modelagem de domínio:

- Compreensão do Domínio: A modelagem de domínio ajuda a equipe de desenvolvimento a adquirir um conhecimento profundo sobre o domínio de negócios em que o software será aplicado. Isso envolve a identificação e a definição de conceitos-chave, objetos, entidades e suas interações.

- Abstração e Simplificação: A modelagem de domínio simplifica a complexidade do mundo real, representando-o de forma mais estruturada e abstrata. Isso torna mais fácil para os desenvolvedores e stakeholders compreenderem e comunicarem as nuances do domínio.

- Documentação: A modelagem de domínio serve como uma documentação valiosa que pode ser usada para garantir a compreensão comum entre membros da equipe, além de facilitar a comunicação com clientes e partes interessadas.

Modelagem de Domínio

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Alinhamento com a Realidade do Negócio: Ao mapear os conceitos do domínio, a modelagem de domínio ajuda a garantir que o software seja alinhado com os processos de negócios, requisitos e metas do cliente.

- Base para o Design de Software: A modelagem de domínio fornece uma base sólida para o design do software, orientando a criação de classes, objetos e relacionamentos que refletem com precisão o domínio.

- Identificação de Padrões e Antipadrões: Ela pode ajudar a identificar padrões e antipadrões no domínio, permitindo que a equipe de desenvolvimento tome decisões informadas sobre como projetar o software.

- Evolução e Manutenção: A modelagem de domínio também facilita a manutenção e a evolução contínua do software, pois ajuda a entender como as mudanças no domínio afetam o sistema.

Modelagem de Domínio

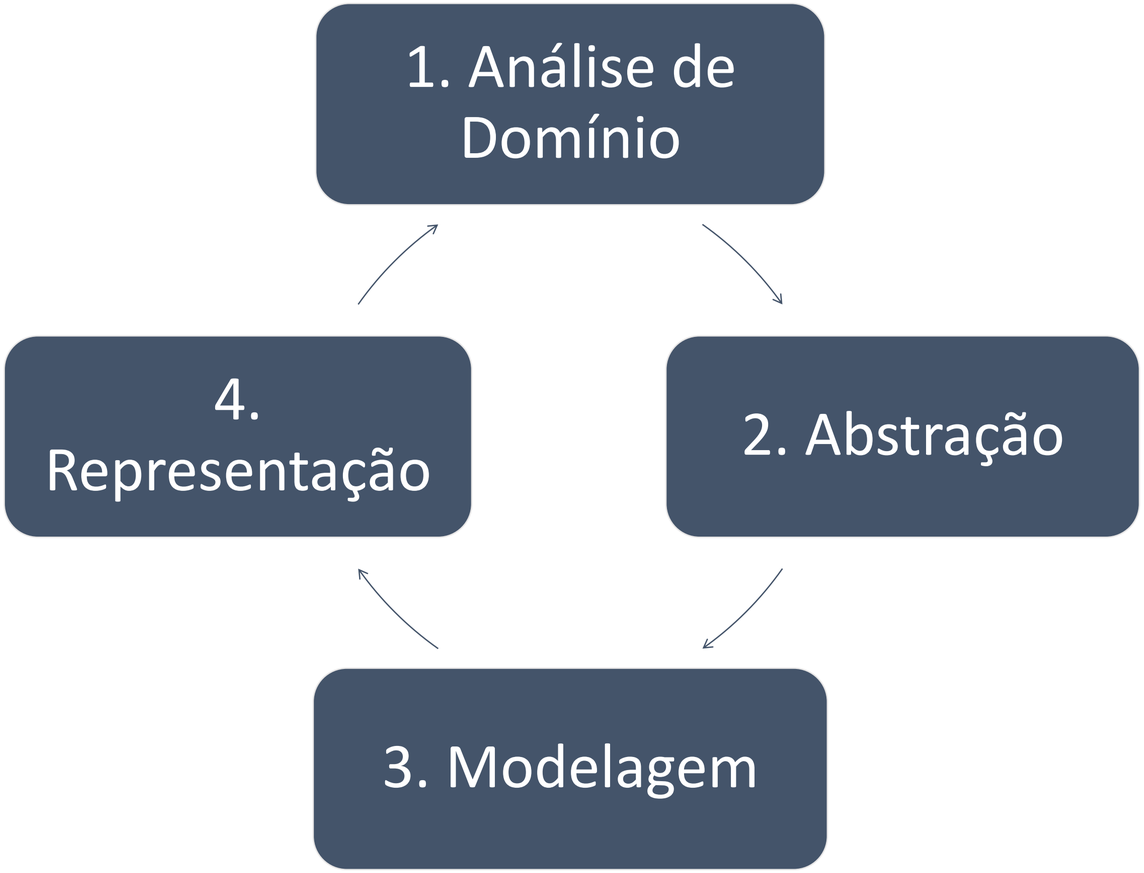
INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

1. Identificar o contexto do Sistema de Informação (Dados, Processo e Atores Envolvidos);

2. Idealizar como funcionará o Sistema de Informação pretendido;

3. Modelar a solução idealizada;

4. Representar o modelo idealizado numa linguagem clara e legível.



Conceitos Importantes

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Domínio – está relacionado com os conceitos, informações e tipo de dados pertencentes ao sistema a ser desenvolvido;
- Abstração – é ação intelectual por meio da qual identificamos mentalmente os conceitos e propriedades inerentes ao sistema a ser desenvolvido;
- Modelo – é a imagem criada após a abstração sobre os conceitos e informações imaginadas no processo de criação do sistema;
- Representação – é o modo como será representado o modelo mental criado;

A representação de um modelo de domínio pode ser feita através do que chamamos de "Diagrama". Uma das formas mais comuns de criarmos diagramas que representam modelos de domínio é através do Diagrama de Classes da UML.

Javascript - Tipo de Dados

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

JavaScript é uma linguagem de programação que é fracamente e dinamicamente tipada, o que significa que o tipo de dados de uma variável é determinado em tempo de execução e sem a necessidade de declarar o seu tipo.

Em JavaScript, os tipos de dados podem ser divididos em duas categorias principais: primitivos e não primitivos (também chamados de objetos).

Primitive Data Types	Reference Data Types
• Boolean • Number • BigInt • String • Undefined • Null • Symbol	• Array • Object • RegEx • Date • ....

Javascript - Tipo de Dados

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Tipos Primitivos:

- **Boolean (Booleano):**
  Representa um valor lógico verdadeiro ou falso.
  Exemplo: let diaEnsolarado = true;

- **Number (Número):**
  Representa tanto números inteiros quanto de ponto flutuante.
  Exemplo: let idade = 25;

- **BigInt:**
  Um tipo especial de número para representar inteiros grandes.
  Exemplo: const valorGrande = 1234567890123456789012345678901234567890n;

- **String (Texto):**
  Sequência de caracteres, normalmente usada para representar texto.
  Exemplo: let nome = "Maria";

Javascript - Tipo de Dados

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Tipos Primitivos:

- **Null (Nulo):**
  Representa a ausência intencional de algum valor.
  Exemplo: let valor = null;

- **Undefined (Indefinido):**
  Representa uma variável que foi declarada, mas ainda não recebeu nenhum valor.
  Exemplo: let sobrenome; // undefined

- **Symbol:**
  Introduzido no ECMAScript 6 (ES6), representa um identificador único.
  Exemplo: const id = Symbol('id');

Javascript – Fracamente Tipada

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

JavaScript é uma linguagem fracamente tipada. Isso significa que você não precisa declarar o tipo de uma variável ao criá-la. O tipo da variável é determinado automaticamente quando o programa é executado, e pode mudar durante a execução do programa.

Em linguagens fortemente tipadas, como Java ou C#, você precisa declarar o tipo de uma variável quando a cria, e essa variável só pode armazenar valores desse tipo específico. Por exemplo, se você declarar uma variável como um número inteiro em Java, ela só pode armazenar inteiros. Em **JavaScript**, por outro lado, você pode ter uma variável que armazena um número em um momento e, em seguida, uma string em outro momento, sem problemas.

```
1   let x = 10;    // x é um número
2   x = "Olá";     // agora x é uma string
3   console.log(typeof(x))
4   x = true;      // agora x é um booleano
5   console.log(typeof(x));
```

Javascript - Declaração de variáveis

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Em JavaScript, temos três palavras-chave para declarar variáveis: **var, let, e const**. Cada uma delas tem um escopo diferente e comportamentos específicos. Vamos explorar as diferenças entre elas:

“var”:

- ✓ var era a única forma de declarar variáveis em versões antigas do JavaScript.
- ✓ Tem um escopo de função ou global. Isso significa que, se uma variável é declarada dentro de uma função usando var, ela é visível em toda a função.
- ✓ Se declarada fora de uma função, torna-se uma variável global.
- ✓ Pode ser redeclarada e atualizada em qualquer lugar do código.
- ✓ Não respeita o escopo de bloco, o que pode levar a bugs difíceis de rastrear.

Javascript - Declaração de variáveis

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

“let:”

- ✓ Introduzido no ECMAScript 6 (ES6), tem um escopo de bloco.
- ✓ Variáveis declaradas com let só são acessíveis dentro do bloco em que são definidas (por exemplo, dentro de loops for, if, while, etc.).
- ✓ Não pode ser redeclarada no mesmo escopo.
- ✓ Pode ser atualizada.

“const:”

- ✓ Também introduzido no ECMAScript 6 (ES6), const cria uma variável cujo valor é fixo (constante).
- ✓ Como let, const tem escopo de bloco.
- ✓ Não pode ser reatribuída. Isso significa que uma vez que uma constante recebe um valor, esse valor não pode ser alterado ou redefinido.
- ✓ No entanto, para objetos e arrays, o valor dos elementos do objeto ou array pode ser modificado. A referência ao objeto ou array é constante, mas não os valores dentro dele.

Javascript - Declaração de variáveis

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Escolhendo entre var, let e const:

- ✓ Use const por padrão para declarar variáveis que não precisam ser reatribuídas.
- ✓ Use let para variáveis que precisam ser reatribuídas.
- ✓ Evite var na maioria dos casos, devido ao seu escopo de função global que pode levar a bugs difíceis de depurar.

Javascript - Declaração de variáveis

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Exemplos:

```
1   function exemplo() {
2       if (true) {
3         var x = 10;
4       }
5       console.log(x); // 10
6     }
7
8     console.log(x); // undefined
9
10  function exemplo() {
11      if (true) {
12        let y = 20;
13        console.log(y); // 20
14      }
15      console.log(y); // Uncaught ReferenceError: y is not defined
16  }
17
18    let y = 30;
19    console.log(y); // 30
20
21  const z = 40;
22  z = 50; // TypeError: Assignment to constant variable.
```

Programação Orientada a Objetos

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

A Orientação a Objetos (OO) é um paradigma de programação que se baseia na ideia de organizar o código em torno de objetos, que são instâncias de classes. É uma abordagem poderosa e amplamente utilizada no desenvolvimento de software, pois ajuda a criar sistemas **mais organizados, modulares e reutilizáveis**.

A Orientação a Objetos promove uma abordagem mais natural e organizada para modelar o mundo real em código, tornando-o mais legível, manutenível e flexível. É amplamente utilizada em linguagens de programação como Java, C++, Python, C#, entre outras, e é aplicável a uma variedade de domínios de desenvolvimento de software, desde aplicações desktop até sistemas distribuídos e aplicativos web.

Programação Orientada a Objetos

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Aqui estão os principais conceitos do paradigma de Orientação a Objetos:

Objeto: Um objeto é uma instância concreta de uma classe. Ele representa uma entidade do mundo real e possui atributos (propriedades) que descrevem seu estado e métodos que definem seu comportamento. Por exemplo, um objeto "Carro" pode ter atributos como "cor" e "velocidade" e métodos como "ligar" e "desligar".

Classe: Uma classe é um modelo ou uma definição abstrata que descreve as características comuns a um conjunto de objetos. Ela serve como um plano para a criação de objetos. A classe define os atributos e métodos que seus objetos terão em comum. Por exemplo, uma classe "Carro" define os atributos e métodos que todos os carros terão

Ambiente Javascript

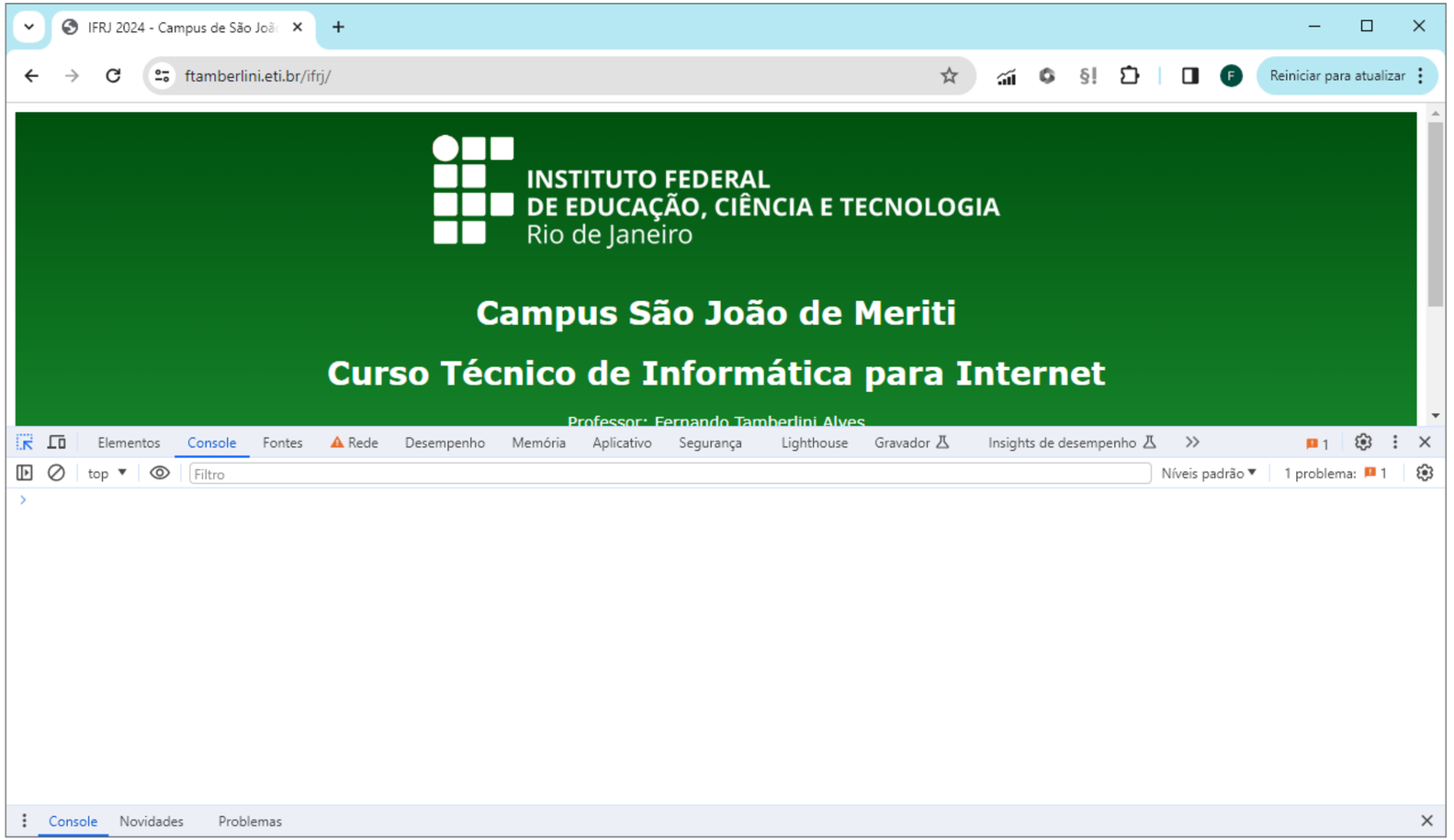
INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Ambiente de Desenvolvimento Integrado (IDE)
  - Editor (VSCode)
  - Navegador (Interpretador de Javascript)
- Forma de Execução
  - Tag <script> no HTML
  - Arquivo .js separado de HTML
  - Console do Navegador
- Forma de Saída
  - Alert()
  - Console.log()
  - Tag HTML

DevTools - Chrome

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

- Console do Chrome
  - F12
  - Ctrl + Shift + i
  - Botão Direito - Inspecionar



Ambiente de Desenvolvimento

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

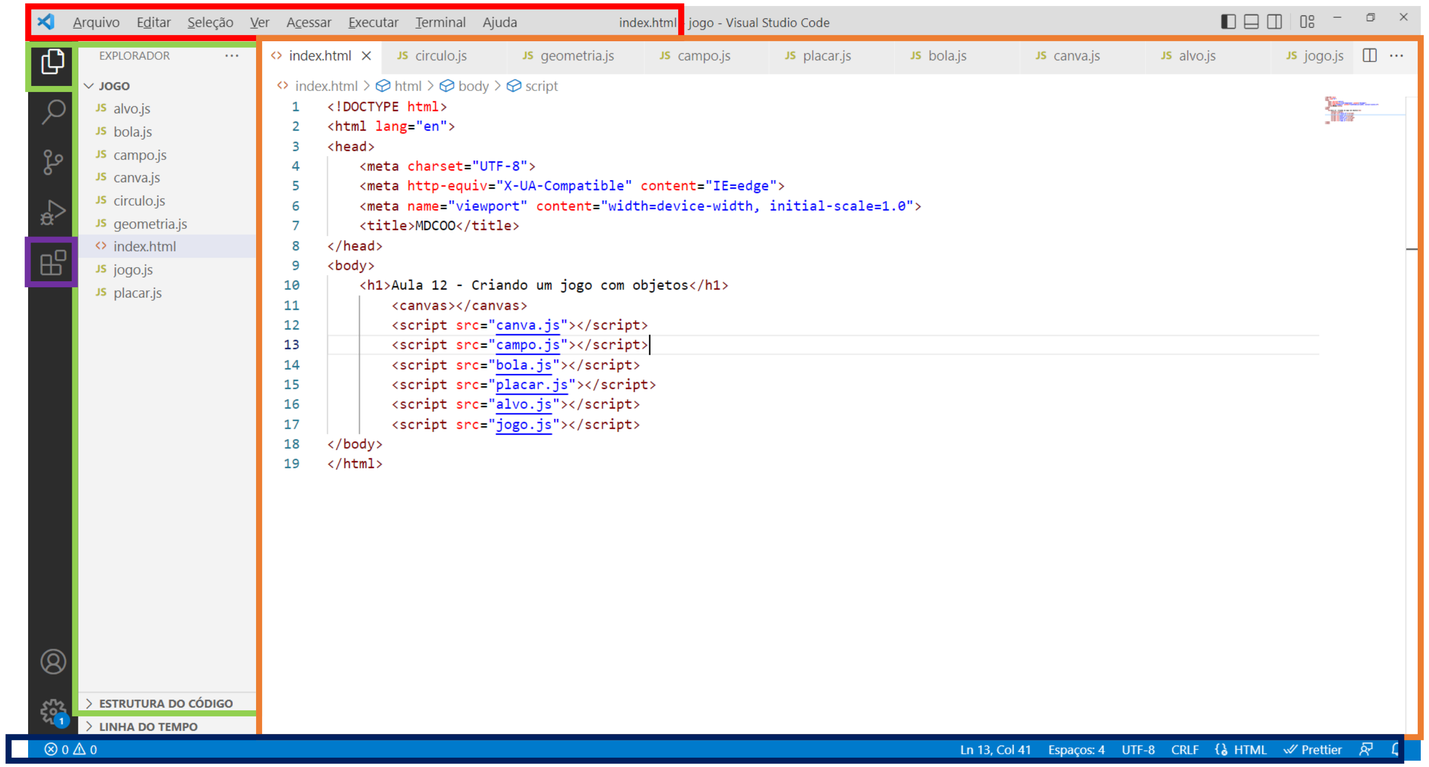
Menu Superior

Explorador

Editor

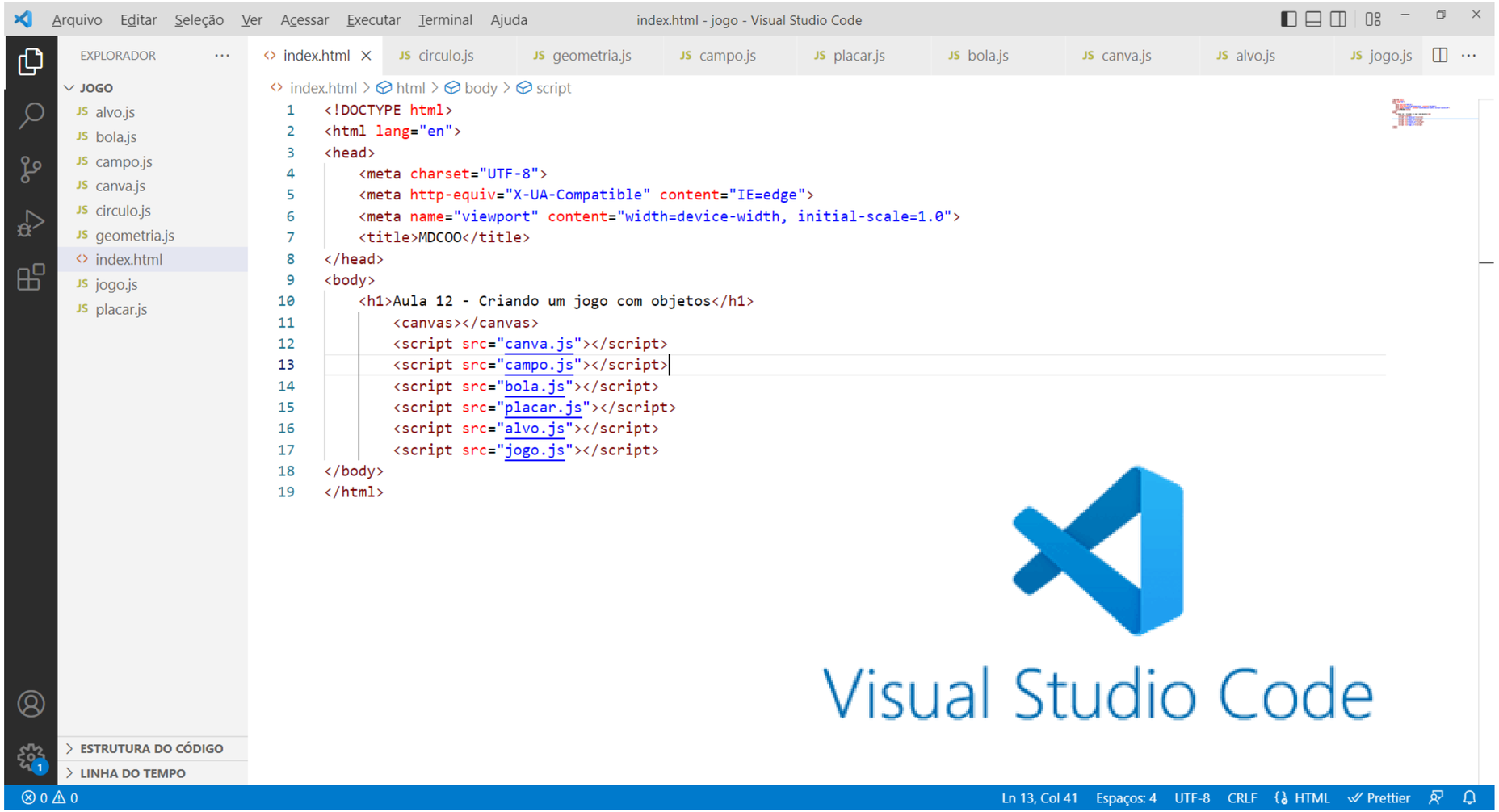
Extensões

Barra de Status



Ambiente de Desenvolvimento

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro



Ambiente de Desenvolvimento

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Teclas de Atalho

Comandos Menu Arquivo

- Abrir Arquivo — Ctrl + O
- Abrir Pasta — Ctrl + K Ctrl + O
- Novo Arquivo Texto — Ctrl + N
- Novo Arquivo — Ctrl + Alt + Windows + N
- Fechar Pasta — Ctrl + K F
- Salvar — Ctrl + S
- Salvar Como — Ctrl + Shift + S

Comandos Menu Editar

- Desfazer — Ctrl + Z
- Refazer — Ctrl + Y
- Comentário Linha — Ctrl + ; ou Ctrl + /
- Comentário Bloco — Shift + Alt + A

Comandos Menu Seleção

- Copiar Linha Cima — Shift + Alt + Seta Para Cima
- Copiar Linha Abaixo — Shift + Alt + Seta Para Baixo
- Mover Linha Cima — Alt + Seta Para Cima
- Mover Linha Abaixo — Alt + Seta Para Baixo
- Cursor Acima — Ctrl + Alt + Seta Para Cima
- Cursor Abaixo — Ctrl + Alt + Seta Para Baixo

- Adicionar Próxima — Ctrl + D
- Adicionar Todas — Ctrl + Shift + L

- Modo Seleção Coluna – Ir no Menu Superior

Atividade Prática

1. Abrir o editor de texto VSCODE;
2. Digitar a conteúdo do arquivo “aula01.html” exposto no projetor;
3. Salvar o arquivo “aula01.html” na área de trabalho e abri-lo em algum navegador;
4. Explorar o formulário do arquivo criado e apontar os seus erros.

Atividade Prática

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

1. Testar a aplicação informando o peso e altura e calculando o IMC;
2. Tente fazer um teste mais amplo possível;
3. Tente descobrir porque o “Estilo” da página está diferente e como a página HTML chama a função JavaScript;
4. Tente alterar a página HTML de forma a evitar alguns tipos de erro;
5. Pense como poderia representar esta aplicação para que um programador codifique uma aplicação similar.

Dúvidas?

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Rio de Janeiro

Dúvidas?

fernando.alves@ifrj.edu.br

ftamberlini.dev.br