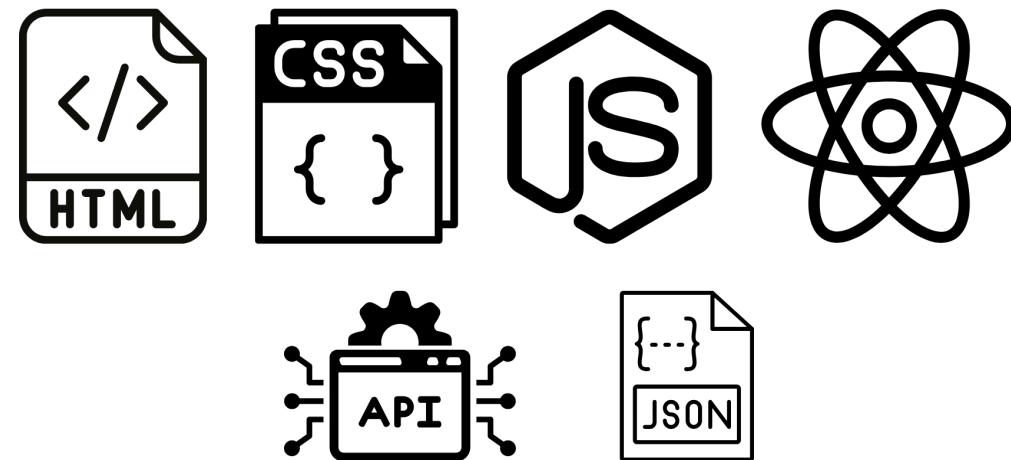
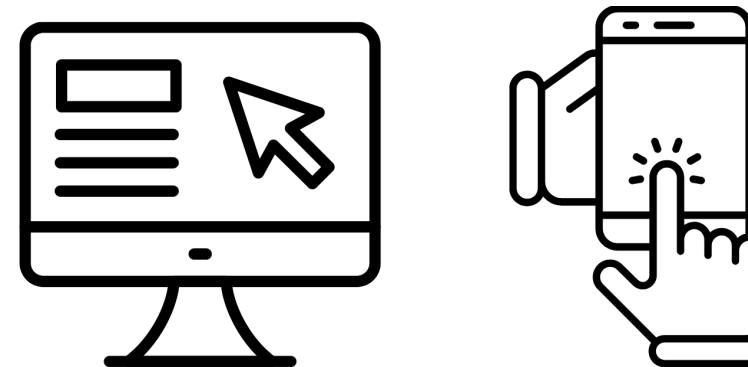


# Programação Web II

## Apresentação da Disciplina



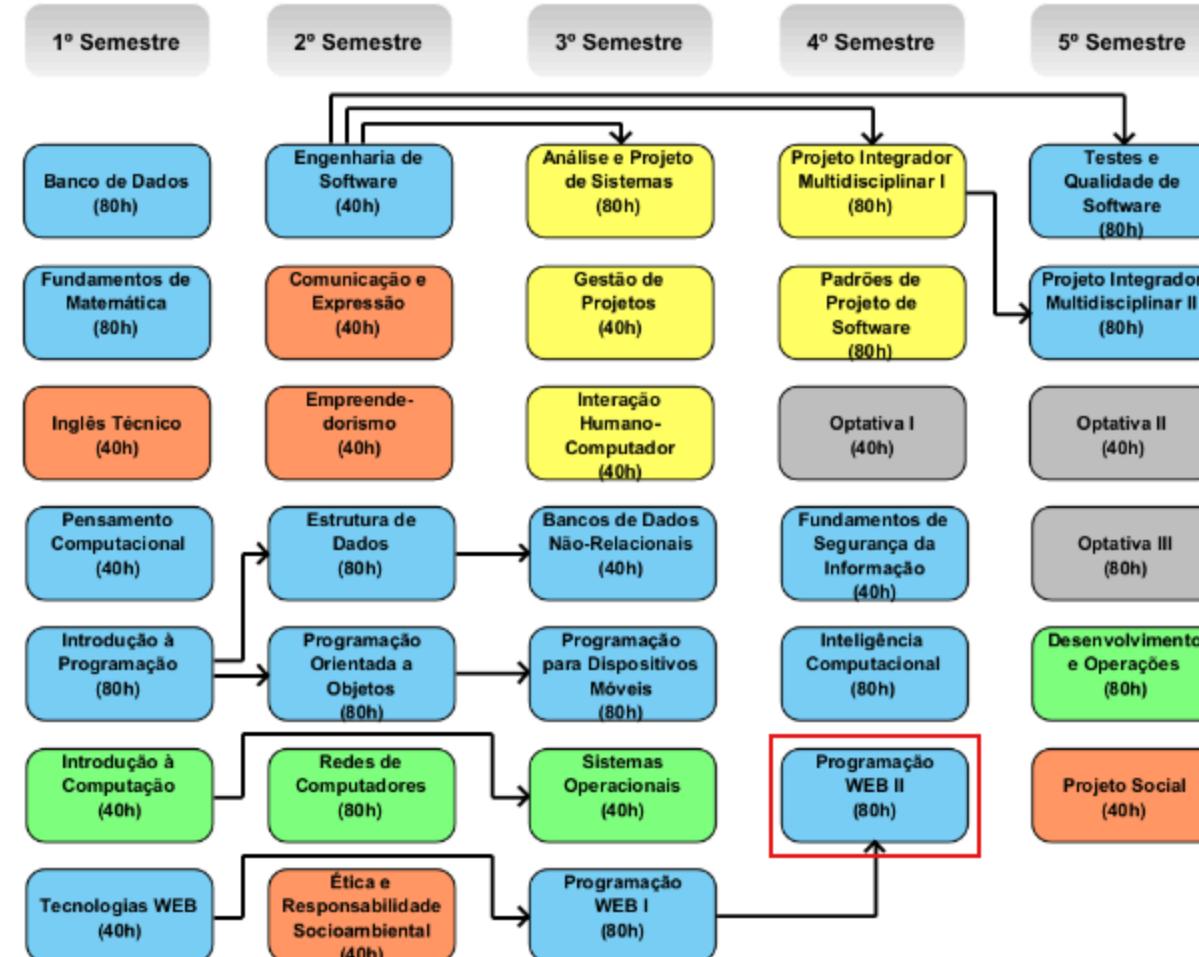
# Visão Geral

---

<b>Disciplina</b>	Programação Web II (PWEB II)
<b>Carga Horária</b>	80h (60h prática)
<b>Créditos</b>	4
<b>Pré-requisito</b>	PWEB I
<b>Semestre</b>	4



# Visão Geral



# Ementa da Disciplina

O que vamos aprender em Desenvolvimento Web II

Programação para front-end: páginas web dinâmicas. Desenvolvimento para front-end com frameworks. Tecnologias multiplataforma e full-stack. APIs e manipulação de dados.



# Programa da Disciplina

Como está organizado o conteúdo?

- **UNIDADE I - Programação para front-end:**
  - Introdução ao Javascript
  - Tipos primitivos, variáveis e operadores
  - Controle de fluxo: condicionais e estruturas de repetição
  - Funções e *callbacks*
  - Arrays e objetos
  - Manipulação do DOM
  - JavaScript moderno: recursos ES6+, promises, fetch API, módulos
  - Introdução ao TypeScript



# Programa da Disciplina

Como está organizado o conteúdo?

- **UNIDADE II - Desenvolvimento para front-end com frameworks:**
  - Ambiente Node.js: características, principais recursos e arquitetura
  - Proposição e características de frameworks front-end
  - Single Page Applications (SPA)
  - Proposição e visão geral de um projeto prático
  - Versionamento de código
  - Componentes, ciclo de vida e data binding
  - Comunicação entre componentes
  - Rotas de navegação e proteção de rotas
  - Testes unitários e cobertura de testes



# Programa da Disciplina

Como está organizado o conteúdo?

- **UNIDADE III - Tecnologias multiplataforma e full-stack:**
  - Progressive Web Application (PWA)
  - Propriedades do Manifest App
  - Services e Workers
  - Cache API e estratégias de cache
  - Web Push Notifications



# Programa da Disciplina

Como está organizado o conteúdo?

- **UNIDADE IV - APIs e manipulação de dados**
  - Serviços - Injeção de Dependência
  - Comunicação de componentes com serviços
  - Comunicação HTTP: get, post, put e delete
  - HTTP custom header
  - Autenticação e Autorização



# Metodologia

Como será o andamento da disciplina?

- Aulas expositivas e dialogadas;
- Aulas no estilo *live coding*;
- Exercícios práticos em laboratório;
- Atividades práticas e mini projetos (individual ou em grupo);
- Desafios com pontuação extra;
- Projeto Final seguindo ritos scrum.

Vamos usar o Git e o GitHub em toda a disciplina.



# Alinhando as Expectativas

💡 O que esperar e o que **NÃO esperar** da disciplina de Programação Web II?

## 👎 O que não esperar de PWEB II

- ✗ Curso aprofundado em um framework específico do início ao fim.
- ✗ Somente “telinhas bonitas” sem arquitetura, testes e boas práticas de UX e acessibilidade.
- ✗ Foco em back-end, bancos de dados ou DevOps (não é o escopo da disciplina).
- ✗ Pular direto para bibliotecas e ignorar fundamentos de JS/TS, HTTP.
- ✗ Cobrir todo o ecossistema de UI libs, design systems e cada detalhe de styling.
- ✗ Disciplina “fácil”: exige prática contínua, leitura de docs e entrega de projeto.

## 👍 O que esperar de PWEB II

- ✅ Fundamentos sólidos para construir aplicações front-end modernas, como SPAs e PWAs.
- ✅ JavaScript/TypeScript moderno, componentes, roteamento, formulários e estado.
- ✅ Consumo de APIs REST/HTTP, tratamento de erros, headers e autenticação básica.
- ✅ PWA: manifest, service workers e estratégias de cache.
- ✅ Acessibilidade e UX essenciais aplicadas ao fluxo principal da aplicação.
- ✅ Projeto guiado em sala como *case base* para o seu projeto final.
- ✅ Autonomia para se aprofundar na documentação oficial de um framework específico quando necessário.

# Notas

Como minhas notas serão atribuídas?

- A nota de cada etapa será composta da seguinte forma:

- **N1:**

- **NI** - Nota da avaliação teórico/prática (individual);
    - **NP** - Nota das atividades práticas realizadas na etapa.
    - **NPF** - Nota relativa ao projeto final (definição e especificação de requisitos).

$$N1 = (2 \times NI + NP + NPF) / 4$$

- **N2:**

- **NI** - Nota da avaliação teórico/prática (individual);
    - **NPF** - Nota relativa ao projeto final (entregas parciais e entrega final).

$$N2 = (2 \times NI + 3 \times NPF) / 5$$



# Notas

---

Como minhas notas serão atribuídas?

- Pontuação das atividades práticas:
  - $NP = (scores\_obtidos / total\_scores) * 10$
- Pontuação do Projeto Final
  - N1 - Uma nota relativa à entrega inicial, com a definição do problema, análise e modelagem inicial.
  - N2 - Serão solicitadas entregas parciais, onde cada uma receberá uma nota de 0 a 10, com um peso associado. A nota final do projeto será uma média ponderada das notas das entregas parciais.

# Ferramentas de Apoio

Vamos usar quais ferramentas?

- **Google Classroom:** Compartilhamento de material, postagem de atividades, entrega de atividades, controle de notas das atividades.
- **Git e GitHub:** Versionamento dos mini projetos e projeto final; Entrega de atividades práticas.
- **Editor de Código / IDE:** VS Code.
- **Testes de APIs e integração:** Postman.
- **Linguagem de Programação e Frameworks (não exclusivos):** JavaScript, TypeScript, Tailwind CSS e React/Next.js



## Código da Turma no Classroom

---

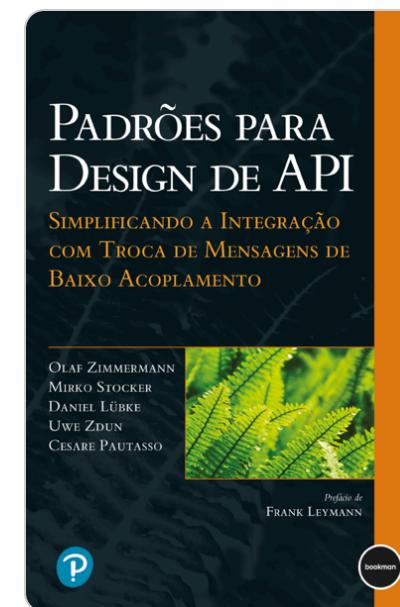
ifzy2zwa



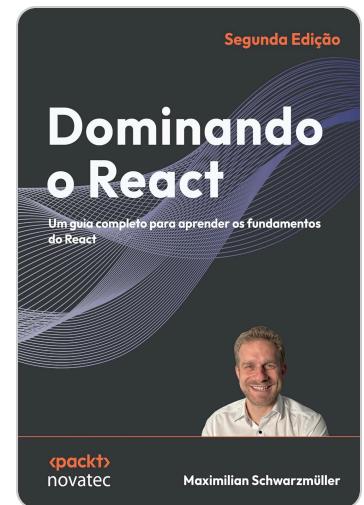
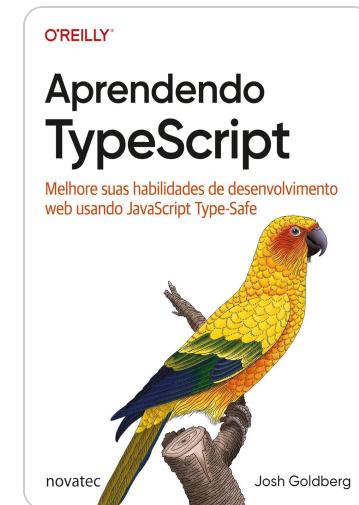
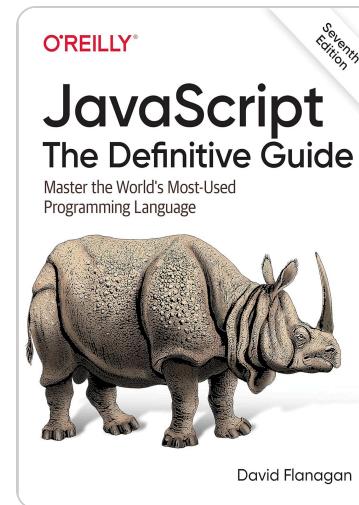
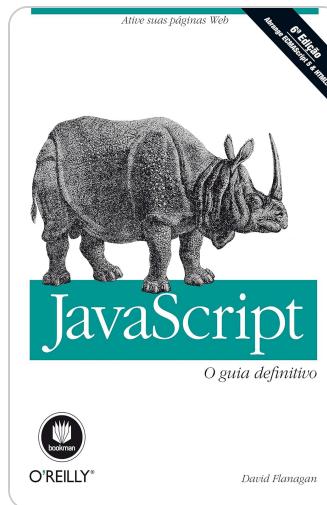
# Bibliografia Básica

[Ver no Sophia](#)[Ver na BVU](#)[Ver na BVU](#)

# Bibliografia Complementar

[Ver na BVU](#)[Ver na BVU](#)[Ver na BVU](#)[Ver na BVU](#)[Ver na BVU](#)

# Bibliografia Alternativa

[Ver na Amazon](#)[Ver na Amazon](#)[Ver na Amazon](#)[Ver na Amazon](#)[Ver na Amazon](#)[Ver na Amazon](#)

# Referências Técnicas

Plataformas de ensino, blogs técnicos e documentações oficiais

## Plataformas de Estudo

- **MDN Web Docs:** <https://developer.mozilla.org/pt-BR/>
- **W3Schools:** <https://www.w3schools.com/>
- **GeeksforGeeks:** <https://www.geeksforgeeks.org/>
- **Google Web.dev:** <https://web.dev/>
- **freeCodeCamp:** <https://www.freecodecamp.org/>
- **Roadmap.sh Frontend:** <https://roadmap.sh/frontend>

## Documentações Oficiais

- **React:** <https://react.dev/>
- **Next.js:** <https://nextjs.org/docs>
- **Vue.js:** <https://vuejs.org/guide/introduction.html>
- **Angular:** <https://angular.dev/overview>
- **TypeScript:** <https://www.typescriptlang.org/pt/docs/>
- **Tailwind CSS:** <https://tailwindcss.com/docs/>
- **Shadcn:** <https://ui.shadcn.com/docs>
- **Ant Design React:** <https://ant.design/docs/react/introduce>

# Avaliação Diagnóstica

| Testando os conhecimentos prévios e alinhando as expectativas.

<https://forms.gle/7zmAN1PfwKjNEvNSA>

