

# Presentation of an Advanced Topic

---



**Course:** Programação Avançada para a Internet (Mestrado em Engenharia de Software)

---

## 1. Objective

The primary goal of this assignment is to allow you to perform a "deep dive" into a cutting-edge topic related to advanced web programming. It is also designed to enhance your technical writing and public speaking skills.

You are expected to become the "local expert" on your chosen topic and effectively teach it to your peers.

---

## 2. Task Description

In groups of two, you will select **one** of the topics listed below or **propose your own for approval**. Your group will then research this topic using academic papers, official documentation, technical articles, and case studies.

The result of this research will be delivered in two parts:

- **A concise, well-researched written paper** that provides a thorough technical overview of the topic.
- **A professional oral presentation** delivered to the class, summarizing your findings and highlighting the most important aspects.

### Part A: The Written Paper (50% of the grade of this assignment)

The paper should demonstrate an understanding of the selected subject.

- **Format:** The paper **should** be formatted using the Springer **Lecture Notes in Computer Science (LNCS)** template. The final submission must be a **PDF** file.
  - **Microsoft Word Template:** Download the template here: [Word Template \(splnproc1703.zip\)](#).
  - **LaTeX Template:** Download the package here: [LaTeX Template \(lncs2e.zip\)](#).
  - **Overleaf Option:** The LaTeX template is also available directly on Overleaf for ease of use: [Springer LNCS Template on Overleaf](#).
- **Length:** Recommended 8-12 pages, excluding references.
- **Structure:** The paper should include/focus on the following aspects:
  1. **Abstract:** A brief summary of the paper's content.
  2. **Introduction:** What is the problem or technology? Why is it relevant today?
  3. **Core Concepts:** Explain the fundamental principles, architecture, and key terminology.

4. **Practical Application / Case Study:** Provide an example of how the technology is used. This could include code snippets, architectural diagrams, or an analysis of a real-world implementation.
5. **Comparative Analysis:** Compare the topic to established or competing technologies. What are its strengths, weaknesses, and ideal use cases?
6. **Conclusion:** Summarize the key takeaways and discuss the future outlook for the technology.
7. **References:** All sources must be properly cited.

#### **Part B: The Oral Presentation (50% of the grade of this assignment)**

The presentation is your opportunity to share your expertise with the class in an engaging and informative way.

- **Duration:** 10-minute presentation + 5-minute Q&A session.
- **Format:** Slide-based presentation (e.g., PowerPoint, Google Slides, Marp). Slides should be clear and concise.
- **Content:** The presentation should:
  - Clearly introduce the topic and its importance.
  - Synthesize the most critical information from your paper.
  - Use diagrams, code examples, or a short demo to illustrate key points.
  - Conclude with a strong summary of the technology's impact and future.
  - All group members must participate actively in the presentation.

---

## **4. Topic Selection Process**

- Topics will be assigned on a **first-come, first-served basis**. Each topic may only be chosen by one group.
- To register your topic, **one member** of the group must complete the topic selection survey on Moodle, available at the following link: [Topic Selection Survey](#).
- In the survey, you will be required to provide the numbers and names of both group members, and to list your top **three** topic choices in order of preference.
- Topic allocation will follow the order of submission. If your most preferred topic has already been chosen, your second choice will be assigned, and so on. If all three of your choices are taken, I will ask your group to select from the remaining topics or to propose an alternative.
- **Deadline for Topic Selection:** October 23, 2025, 09:00.

---

## **6. Submission Deadlines and Presentation**

- **Written Paper Submission:** December 4, 2025, 09:00 via Moodle.
- **Presentation:** In class on December 4, 2025. You should submit your slides via Moodle before or immediately after the presentation.

---

## **7. List of Available Topics**

### **Category 1: Propose Your Own Topic**

- If you have a specific interest that is not on this list but is relevant to advanced web development, you may propose your own topic for approval.

## Category 2: Architectural Patterns & The Future Web

- **Micro-Frontends:** Architecting and implementing a web application with independently deployable frontend components.
- **Serverless & Edge Computing:** A comparative analysis of FaaS (e.g., AWS Lambda, Vercel Functions) and Edge Functions (e.g., Cloudflare Workers).
- **The JAMstack:** In-depth exploration of modern web architecture using JavaScript, APIs, and Markup.
- **WebAssembly (WASM):** The future of high-performance web applications; use cases and integration with JavaScript.

## Category 3: Advanced API & Data Layer

- **Beyond REST:** A deep dive into building efficient and flexible APIs with GraphQL.
- **Real-Time Communication:** A comparative analysis of WebSockets, Server-Sent Events (SSE), and WebRTC.
- **Modern Authentication:** A deep dive into OAuth 2.0, OpenID Connect, and JSON Web Tokens (JWTs).
- **Database Scaling Strategies:** A look into sharding, replication, and caching for NoSQL and relational databases.

## Category 4: Modern Frontend & User Experience

- **State Management in SPAs:** A comparative study of advanced state management libraries (e.g., Redux Toolkit, Zustand, Pinia).
- **Advanced Web Performance:** A practical guide to Core Web Vitals, code splitting, and modern optimization techniques.
- **Progressive Web Apps (PWAs):** Building native-like experiences with advanced offline capabilities and push notifications.
- **Web Accessibility in Depth:** Implementing WCAG 2.1 standards in complex, dynamic applications using ARIA and automated testing.
- **Modern CSS Architectures:** A comparative analysis of methodologies like CSS-in-JS, Utility-First (e.g., Tailwind CSS), and component-scoped CSS.

## Category 5: Tooling, Testing & Frameworks

- **Rendering Patterns:** A comparative analysis of SSR, SSG, and ISR using frameworks like Next.js or Astro.
- **The Rise of Meta-Frameworks:** An exploration of how frameworks like Next.js, Nuxt, and SvelteKit are shaping full-stack development.
- **Automated End-to-End Testing:** A deep dive into modern E2E testing frameworks like Cypress or Playwright.
- **AI in the Browser:** An introduction to running Machine Learning models on the client-side with TensorFlow.js.