# LEARNING JOURNEY

**3.1** Our team learned how to handle the unforeseen challenges that appeared along the way, how to delegate tasks, and the true meaning of teamwork. We learned to utilize new resources and manage our time effectively; we learned from one another, and through this collaboration, we continuously improved.

**3.2** Our information was verified piece by piece to ensure the application functioned on every mobile device. We conducted consecutive tests with the support of a feedback team and consulted specialized websites to ensure that our security protocols meet all necessary standards to prevent errors.

**3.3 Bibliography**


- **CLOUDFLARE.** *Cloudflare Workers Documentation*. Official documentation for the cloud execution platform used in the app's backend.
- **CLOUDFLARE.** *Cloudflare Pages Documentation*. Official documentation regarding hosting and publishing web applications.
- **LEAFLET.** *Leaflet Documentation*. JavaScript library used to display interactive maps in the browser.
- **MDN WEB DOCS.** *Web App Manifest*. Reference material on web application manifests.
- **MDN WEB DOCS.** *Service Worker API*. Supporting content on offline functionality, caching, and the PWA experience.
- **MDN WEB DOCS.** *Window: localStorage property*. Reference for local storage within the browser.
- **OPENSTREETMAP CONTRIBUTORS.** *OpenStreetMap*. Open cartographic base used for maps and geographic data.
- **OPENSTREETMAP FOUNDATION.** *Nominatim Documentation*. Geocoding service used to transform place names into coordinates.
- **OVERPASS API.** *Overpass API Documentation*. Tool used to query OpenStreetMap data.
- **PWABUILDER.** *PWABuilder*. Platform used to prepare the web app for packaging as an installable application.
- **TURSO.** *Turso Documentation*. Documentation for the cloud database utilized in the project.
- **W3C.** *Manifest App Information* and materials on installable web applications. Conceptual reference for PWA structure.


*Translated into English by Gemini*.

**Methodology**

In the production of **RADAR ESCOLAR**, a "**vibe coding**" logic was applied—meaning a more creative, fast, and experimental way of programming. We tested ideas, adjusted interfaces, and improved functions bit by bit until the app became fully functional.

We also utilized **remixed content**, as the project brought together technical documentation references, interface examples, open libraries, public maps, and existing web standards to create a new solution with its own identity.

In other words, the app didn't appear out of thin air: it was built by mixing learning, adaptation, testing, and customization, as is common in many modern technology projects.

---

**Production Credits**

**Development Credits:**

Application developed by the **CyberPrincess** group for the **TechnoGirls 2026** project, based on open web technologies, collaborative maps, and PWA architecture.

**Methodological Credits:**

Development involved practical experimentation, rapid prototyping, vibe coding, successive testing, and the remixing of technical and visual references to transform a school project idea into a functional application.

**Technical App Credits:**

Frontend in HTML, CSS, and JavaScript; interactive maps with Leaflet and OpenStreetMap; cloud publishing; PWA manifest and service workers for app installation and packaging.

The **RADAR ESCOLAR** application was developed using open web technologies and a creation methodology based on experimentation, prototyping, and the remixing of technical content. During production, references from documentation on PWAs, digital maps, local storage, cloud hosting, and databases were utilized. The process also aligns with the concept of **vibe coding**, understood as a more creative, iterative, and rapid-test-oriented development practice, without losing focus on final functionality. The result is a digital school system with a web version and the capability to be packaged as an installable app.

**Team Cyber Princess**