

Development Log - Kilometros de Vida

Project Evolution

Initial Delivery (EJS + MongoDB)

For the first delivery, we created a basic web application using EJS templates and connected forms to MongoDB Atlas to track donors and volunteers.

Capstone Upgrade (Full MERN Stack)

We transformed the project into a modern, professional-grade MERN stack application with advanced features and comprehensive documentation.

Team Contributions

Regina Beltrán López

Role: Frontend Design & UX Lead

Responsibilities: - Designed and implemented the visual identity and color scheme - Created responsive layouts for all pages using Tailwind CSS - Implemented animations using Framer Motion for enhanced user experience - Ensured accessibility standards (WCAG AA compliance) - Designed the donation map interface and marker popups

Key Achievements: - Transformed basic EJS templates into a modern React application - Created a cohesive design system with primary/secondary colors and typography - Implemented smooth scroll animations and counter effects on homepage - Designed mobile-responsive navigation with user profile dropdown

Personal Reflection: Working on this project taught me the importance of user-centered design in social impact applications. The biggest challenge was balancing aesthetic appeal with functional simplicity—our users range from restaurant owners to volunteers, so the interface needed to be intuitive for everyone. I'm particularly proud of the impact map visualization, which makes the project's reach immediately visible. This project reinforced my passion for using design to solve real-world problems.

Daniela Terán Martija

Role: Full-Stack Integration & Deployment Lead

Responsibilities: - Led the transformation from EJS to React architecture - Implemented MongoDB integration and data persistence - Set up Google OAuth 2.0 authentication flow - Configured Vite build system and development

environment - Created the Profile page with CRUD functionality - Implemented protected routes and authentication middleware

Key Achievements: - Successfully migrated from server-side rendering to SPA architecture - Integrated Google Identity Services for secure authentication - Implemented user context for global state management - Created comprehensive error handling across the application - Set up environment variable management for security

Personal Reflection: This capstone project was the culmination of everything we learned in the course. The most challenging aspect was implementing secure authentication without a traditional backend session system—I learned to leverage Google OAuth and client-side state management effectively. Debugging the proxy configuration between Vite and Express taught me valuable lessons about full-stack development. I'm proud that we built a production-ready application that could genuinely help reduce food waste in our community.

Álvaro Gallo Cruz

Role: Backend Architecture & Database Lead

Responsibilities: - Designed MongoDB database schema for users, givers, and drivers - Implemented RESTful API with Express.js - Created all backend endpoints with proper validation - Integrated Nominatim API for address geocoding - Implemented data validation and sanitization - Created seed script for demo data

Key Achievements: - Designed scalable database schema with proper relationships - Implemented full CRUD operations for donations - Created middleware for authentication and input sanitization - Integrated third-party geocoding API with rate limiting - Documented all API endpoints with examples

Personal Reflection: Building the backend for a real-world application was incredibly rewarding. The biggest technical challenge was handling asynchronous geocoding while respecting API rate limits—I implemented a queuing system with localStorage caching to solve this. I also learned the importance of proper error handling and validation; early in development, we had issues with malformed addresses breaking the map, which taught me to always validate and sanitize user input. This project showed me how backend architecture decisions directly impact user experience.

Technical Challenges & Solutions

Challenge 1: Authentication Without Complex Backend

Problem: Implementing user authentication without building a full session management system.

Solution: - Integrated Google OAuth 2.0 for secure, passwordless authentication
- Used React Context API to manage user state globally - Linked donations to users via email address - Created `ProtectedRoute` component to guard authenticated pages

Outcome: Secure authentication with minimal backend complexity, leveraging Google's infrastructure.

Challenge 2: Real-Time Map Visualization

Problem: Displaying donation locations on a map without a paid geocoding service.

Solution: - Integrated Nominatim (OpenStreetMap) free geocoding API - Implemented client-side geocoding to reduce server load - Added localStorage caching to minimize API calls - Respected 1-request-per-second rate limit with delays - Created loading indicator to show geocoding progress

Outcome: Functional map visualization with zero API costs and good performance.

Challenge 3: State Management Across Components

Problem: Sharing user authentication state between Navbar, Profile, and forms.

Solution: - Created `UserContext` with React Context API - Implemented `login()` and `logout()` functions - Wrapped entire app in `UserProvider` - Components access user data via `useUser()` hook

Outcome: Clean, centralized state management without external libraries like Redux.

Challenge 4: MongoDB Connection Issues

Problem: Initial deployment had authentication failures with MongoDB Atlas.

Solution: - Created `.env.template` with example configuration - Documented setup process in `TODO.md` - Implemented proper error handling for connection failures - Added clear error messages to guide users

Outcome: Clear documentation for future setup and deployment.

Challenge 5: Form Validation & Error Handling

Problem: Inconsistent error messages and poor user feedback on form submissions.

Solution: - Implemented server-side validation with descriptive error messages
- Added client-side validation for immediate feedback - Created consistent error/success message components - Used Framer Motion for smooth error message animations

Outcome: Professional user experience with clear, helpful feedback.

Architectural Decisions

Why React over EJS?

We migrated from EJS to React for: - **Better UX:** Single-page application with smooth transitions - **State Management:** Easier to manage complex user interactions - **Component Reusability:** DRY principle for forms and UI elements - **Modern Ecosystem:** Access to libraries like Framer Motion and React Leaflet

Why Client-Side Geocoding?

We chose client-side geocoding because: - **Reduced Server Load:** Offload processing to client browsers - **Caching:** localStorage persists geocoded addresses - **Simplicity:** No need for backend geocoding infrastructure - **Cost:** Free tier of Nominatim sufficient for our needs

Why Google OAuth?

We chose Google OAuth because: - **Security:** No password storage or management - **Trust:** Users already trust Google authentication - **Simplicity:** Single sign-on reduces friction - **User Data:** Access to verified email and profile picture

Real-World Application

Current Impact

This application demonstrates a viable solution to food waste logistics. With proper deployment and community adoption, it could: - Connect 100+ restaurants and markets with volunteer drivers - Reduce food waste by facilitating

surplus redistribution - Feed thousands of families through efficient coordination
- Provide transparency through the impact map

Scalability Considerations

To scale this solution city-wide, we would need:

1. **Notification System:** Alert volunteers when donations are nearby
2. **Route Optimization:** Calculate efficient pickup routes
3. **Admin Dashboard:** Manage users and monitor system health
4. **Mobile App:** Native iOS/Android apps for drivers
5. **Analytics:** Track impact metrics (meals saved, CO2 reduced)

Lessons for Production

If deploying this to production, we would:

1. Implement JWT-based authentication with refresh tokens
2. Add comprehensive automated testing (Jest, Cypress)
3. Set up CI/CD pipeline for automated deployment
4. Implement monitoring and logging (Sentry, LogRocket)
5. Add database backups and disaster recovery
6. Implement rate limiting and DDoS protection
7. Conduct security audit and penetration testing

Final Reflection

What We Learned

This capstone project taught us:

- **Full-Stack Integration:** How frontend and backend work together
- **Real-World Constraints:** Balancing features with time and resources
- **User-Centered Design:** Building for actual users, not just requirements
- **Documentation:** The importance of clear, comprehensive documentation
- **Teamwork:** Effective collaboration and task division

What We're Proud Of

- **Professional Quality:** Production-ready code and design
- **Social Impact:** Solving a real problem in our community
- **Technical Depth:** Implementing advanced features (OAuth, maps, CRUD)
- **Documentation:** Comprehensive guides and diagrams
- **User Experience:** Smooth, intuitive interface

What We Would Do Differently

- **Earlier Testing:** Should have tested with real users sooner
- **Mobile-First:** Should have designed for mobile from the start
- **Automated Tests:** Would add unit and integration tests
- **Performance:** Could optimize bundle size and loading times
- **Accessibility:** Would conduct formal accessibility audit

Conclusion

Kilometros de Vida represents the culmination of our learning in this course. We successfully built a full-stack MERN application that addresses a real social problem with professional-grade code, design, and documentation. The project demonstrates our mastery of modern web development while creating something that could genuinely make a difference in our community.

The journey from basic EJS templates to a sophisticated React application taught us invaluable lessons about software engineering, user experience, and social entrepreneurship. We're proud of what we've built and excited about the potential impact it could have.