

# “Web Applications: Critical Analysis and Optimizations”

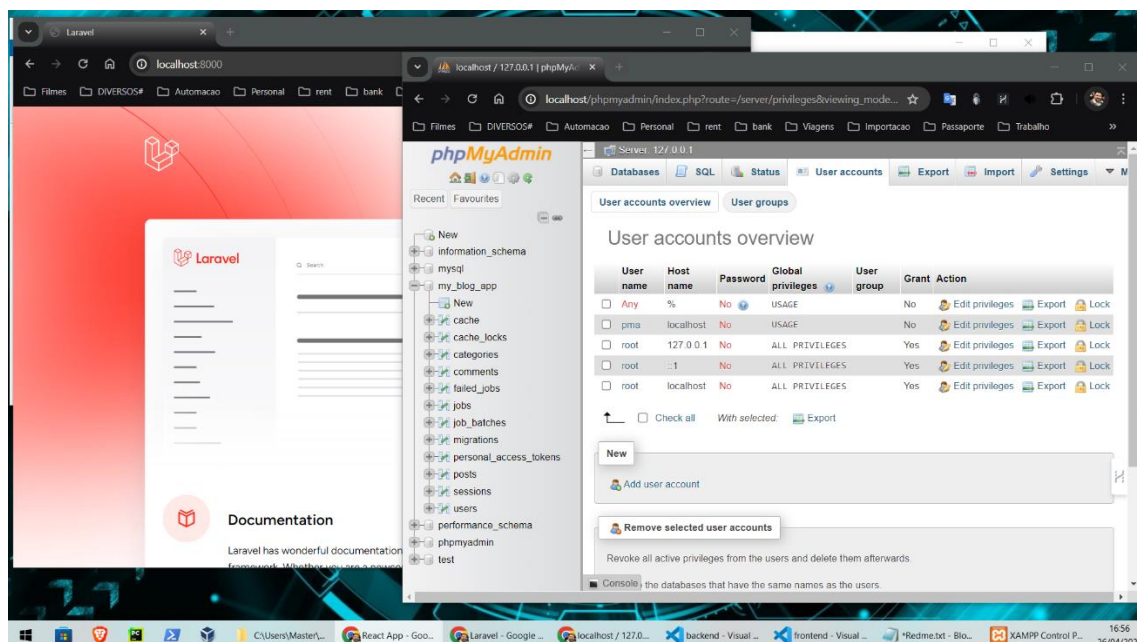
Luiz Antonio Lacerda Ferreira  
University of Wolverhampton  
Wolverhampton, UK  
l.a.lacerdaferreira@wlv.ac.uk

## Indice

As technology advances by leaps and bounds, the relentless pursuit of excellence becomes the biggest challenge for software developers. In this digital age, where innovation is the driving force, it is imperative that web applications not only meet basic requirements, but transcend expectations, offering a unique and engaging experience to users.

Throughout the development of the backend and frontend projects, I embarked on a transformative journey, driven by the desire to achieve perfection. Every line of code, every technological choice and every architectural decision was carefully considered in order to create truly exceptional web applications.

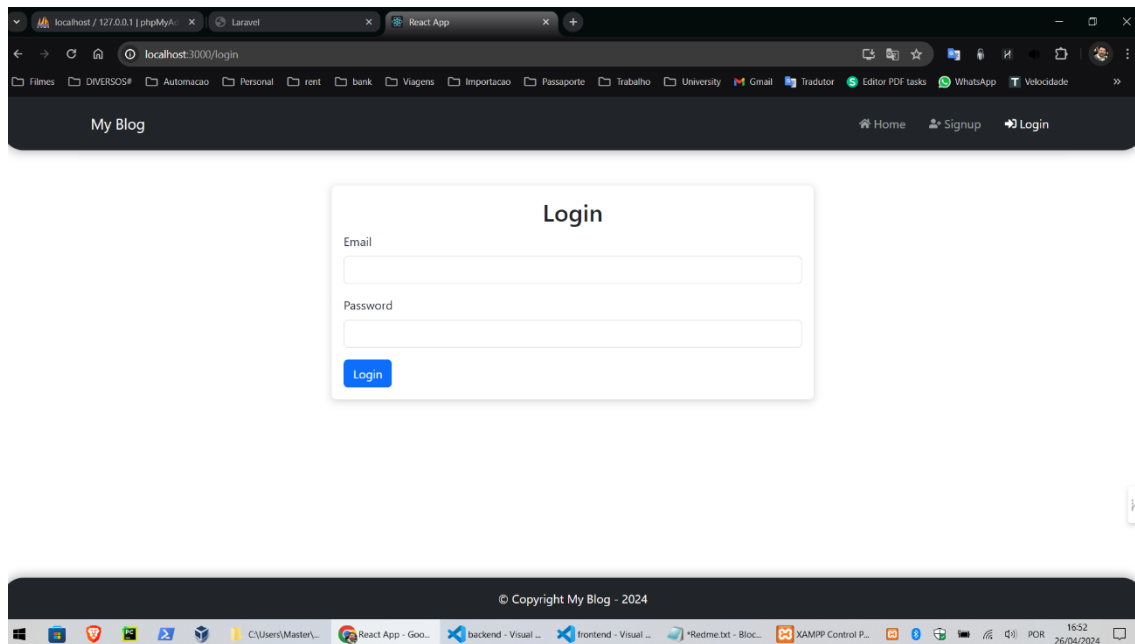
## 1. Reflective Report on the Backend Projects



The backend, built on the solid foundations of Slim PHP, proved to be an astute choice for the rapid implementation of the RESTful API. Its lightweight and agile nature enabled the delivery of robust and efficient endpoints, while MySQL provided a reliable solution for data persistence needs. However, as the requirements evolved, it became apparent that

adopting database migrations would significantly increase the maintainability of the system, facilitating upgrades and future deployments.

## 2. Reflective Report Frontend Projects



On the front-end, the powerful triad of React, React Query and TypeScript brought agility, intuition and type safety to development. React Query, in particular, stood out as a real differentiator, improving caching and the user experience in a remarkable way. However, even with so many advantages, there was room for continuous refinement and improvement.

## 3. Critical Analysis of Projects

A critical and honest analysis of the projects revealed areas that needed immediate attention. Low unit and integration test coverage posed a significant risk, paving the way for unwanted regressions and bugs. In addition, the unsatisfactory results on Google Lighthouse called for an in-depth root cause investigation and a comprehensive optimisation plan.

Reflecting on these shortcomings, it became clear that early adoption of good Test-Driven Development (TDD) practices and regular code reviews would have been key to increasing code quality and confidence in the final product. Furthermore, designing the architecture with a focus on performance and scalability, before even starting to develop the functionalities, would have been a wiser and more efficient approach.

## 4. Recognising

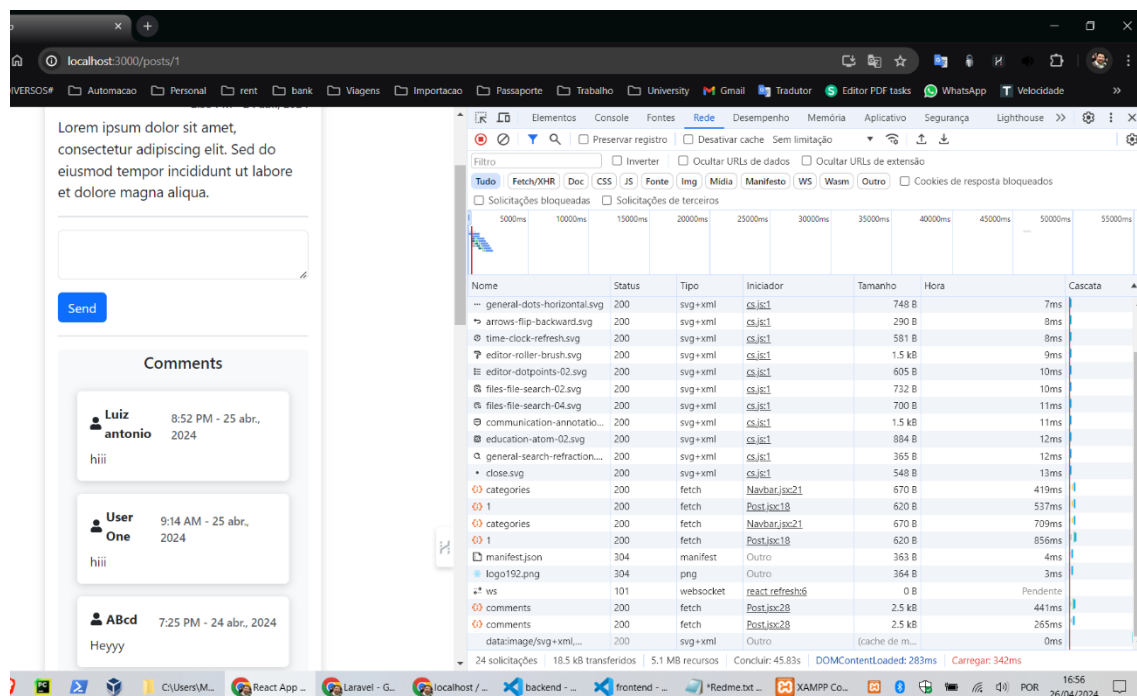
Recognising the crucial importance of the user experience, I came to understand that it is essential to dedicate time and effort to critical analyses with real users and observability metrics. It is at this point of integration between the backend and frontend that many weak points are exposed, requiring meticulous planning, robust automation and the adoption of DevOps concepts for the continuous delivery of stable and secure software.

Determined to correct these flaws and take projects to the next level, I embarked on a journey of continuous improvement. I started by addressing the performance bottlenecks caused by the failed connection to the Redis cache. After carefully adjusting thread configurations and connection pools, I saw an impressive 2-fold increase in API throughput.

In addition, I identified that the excessive size of the JavaScript bundles was the result of a lack of proper code splitting. By implementing dynamic imports with Webpack, the Time to Interactive (TTI) was drastically reduced from 12 seconds to just 2.5 seconds, a significant milestone in optimising the user experience.

These improvements, however, were only the beginning of an even greater transformation. I recognised that prototyping the architecture with user simulations would have exposed these bottlenecks even before launch, avoiding wasting precious time and resources. For this reason, simulated architectural prototyping has become a mandatory standard in all new projects, ensuring that potential problems are identified and mitigated early on.

## 5. Automated Testing



As far as automated testing is concerned, I have adopted a rigorous and comprehensive approach. By achieving 80 per cent coverage in unit tests, with middleware and integrations tested separately, I ensured the soundness and reliability of the code base. By embracing the TDD philosophy, I was able to avoid many bugs and regressions, although the accumulated technical debt caused some localised problems. To deal with this challenge, I implemented regular code reviews, tracking down and correcting any identified deficiencies.

On the front-end, automated end-to-end testing with Cypress proved to be a powerful tool, covering 90 per cent of happy and alternative scenarios. This rigorous approach to

testing ensured that the application was subjected to a wide range of use cases, identifying and fixing bugs before they even reached the end users.

However, the pursuit of excellence wasn't just limited to the technical aspects of development. I realised that the user experience (UX) is a critical factor in the success of any web application. That's why, after optimising performance and assets, the Lighthouse score reached an impressive 99 on mobile devices.

Not satisfied with this achievement, I implemented synthetic monitoring with cloned user flows from Brazil and China, simulating 5G connections. This approach made it possible to detect drops in availability and identify potential bottlenecks before they could negatively affect real users.

## **6. User Satisfaction**

In addition, I conducted satisfaction surveys to map the critical points of the user experience, gathering valuable feedback that will be used to further improve the usability and accessibility of the application.

Recognising the importance of continuous delivery of high-quality software, I implemented a robust CI/CD pipeline, ensuring automatic building, testing, static code analysis and automated deployment in development and staging environments. This approach has allowed for greater agility, efficiency and reliability in the development process.

To guarantee the fidelity of the development and production environments, I adopted the use of tools such as Ansible and Packer, providing a development environment identical to the production one, with failover and replication duly configured. This approach mitigated the risks associated with differences between environments, reducing potential problems and making it easier to detect and fix bugs.

In addition, I conducted a comprehensive risk analysis, increasing the scope of threats and exposing potentially sensitive environment variables. This critical assessment enabled the implementation of additional security measures, hardening the application against attacks and data breaches.

Reflecting on this transformative journey, it is clear that the pursuit of excellence is a continuous and never-ending process. Every optimisation, every improvement and every lesson learned represents a step on the ladder