

Analysis of Software Layout:

The initial proposal by the customer is to develop a solution to make the life of their customers much easier. It consists of capturing customer information and validating whether or not it is suitable for a loan. This should be done asynchronously and in the shortest possible time.

Steps to avoid the software crisis:

1.- A database modeling was carried out, initially with a relational structure, which is what is traditional in companies. However, due to the rise of nonrelational databases, and their speed at the time of making a query, I have declined this option. For this I have used MongoDB, which is very light and integrates very well with the customer’s request.

2.-Use cases were made to determine who would be the people who would participate in the software workflow (clients, analysts and Software administrators).

3.-Deployment viabilities were analyzed, for this I have implemented a virtual environment provided by Python on the back end side, this mainly to not generate version or package conflicts when deployed in production. It is advisable for these cases to implement Dockers images.

4.-On the Front side I used a JavaScript library called React, which allows me to test with a local Server, and provides a convenient modularization of the components, this avoids rewriting code and makes the application much more scalable.

5.-The main focus of React is to be able to consume resources directly from the browser which at the infrastructure level is excellent for any company, In addition, the way of rendering its components asynchronously allows that we do not have to reload the page to each query.

6.-Continuing with the traditional Controller View Model, I have implemented an api to consume it from the Front and provide quick queries, as well as any modification or deletion of logs.

7.-For the API tests, Postman was used, and Flask’s own packages were imported from the backend side, to make the connection with the database, as well as a package to organize the queries of the Json file generated in MongoDB.

8.-It was also necessary to install the CORS package, which according to Flask documentation, avoids conflicts with the server provided by React.

9.-For better working practices the backend of the fronted was separated and in this way, we can optimize our layout

10.-The last details were on the front side, a Bootstrap form was used, and some words generated from the official Lending Front website were included and an embedded video of the company was included.

11.-Finally, tests were carried out and the project was organized.