Finally I chose a very basic folder structure, one of the most classic and simple.
The project follows a microservice architecture, where the API works as an Orquestator, receiving urls to verify, then creating them in the database and delegating the check operation to celery workers.
The UrlsCheck model stores every url information.
The available endpoints implement an async pattern where initially the response returns accepted, later the user can query the progress of the generated batch.

This is made using Celery with Redis has broker and Cache, once the urls are received, each one generates a celery task which processes it in a parallel process.
Every worker has a retry system in case of unexpected error. The successful cases are stored in cache.

All the application is handled in Docker, so it can be upgraded by adding more workers and makes it easy to deploy or make local tests.