

Foodio

Web-app di delivery

Funzionalità

Una web-app per la consegna a domicilio di beni, che mette in contatto diretto clienti, esercenti e rider.

Un utente può effettuare ordini presso i negozi nelle vicinanze e avere aggiornamenti sullo stato dell'ordine.

Esercenti e rider hanno la possibilità di ricevere ordini in real-time, visualizzando in anticipo il compenso per la consegna proposta

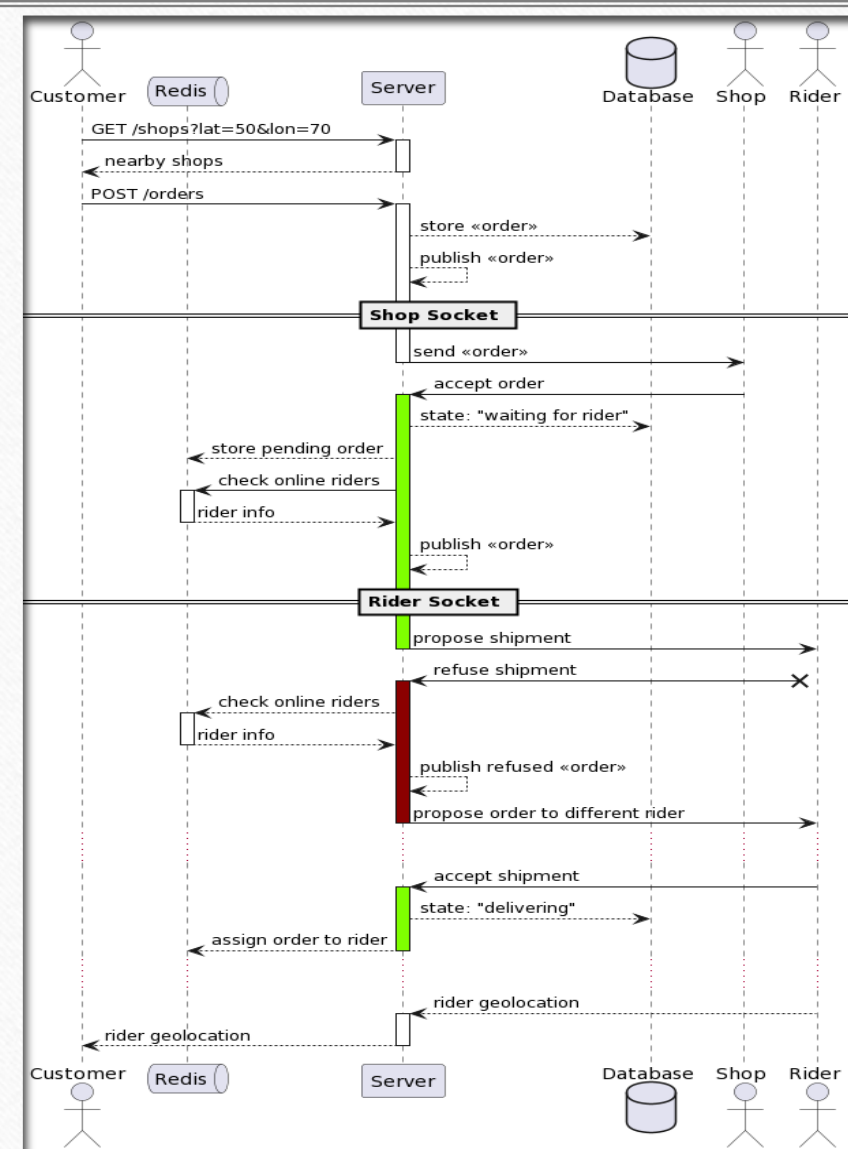
Architettura

Gli approcci architetturali adottati sono l'architettura client-server, il modello *ReST* (*Representational State Transfer*), con particolare riferimento allo sviluppo delle REST API, e il paradigma *publish/subscribe*, impiegato come middleware fra componenti indipendenti del sistema.

Entities

Interaction

Tipico scenario di un ordine inoltrato con successo



Data Entities

E User
_id : text «generated»
role : enum username : text password : text credit : number geolocation : Object
working : boolean restaurant name : text menu : Object

E Order
_id : text «generated»
restaurantUsername : text stateOfOrder : text bill : number order : Object customerPosition : Object restaurantPosition : Object time : Date

Tecnologie Abilitanti

Stack *MEAN*:

- *Angular*
- *Node.js + Express*
- *Mongo DB*

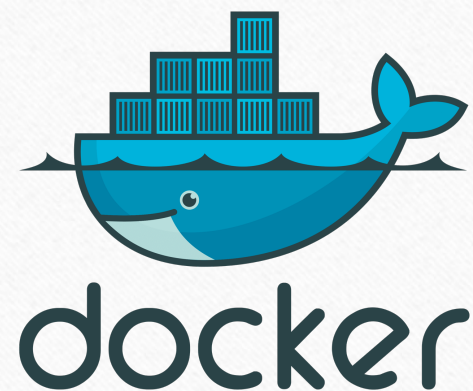


Tecnologie Abilitanti

Redis: in-memory database open-source, ideale per implementare caching in memoria ad elevata disponibilità. Mette a disposizione anche una libreria *pub/sub*

Socket.io: libreria che abilita una comunicazione bidirezionale a bassa latenza e event-based fra client e server

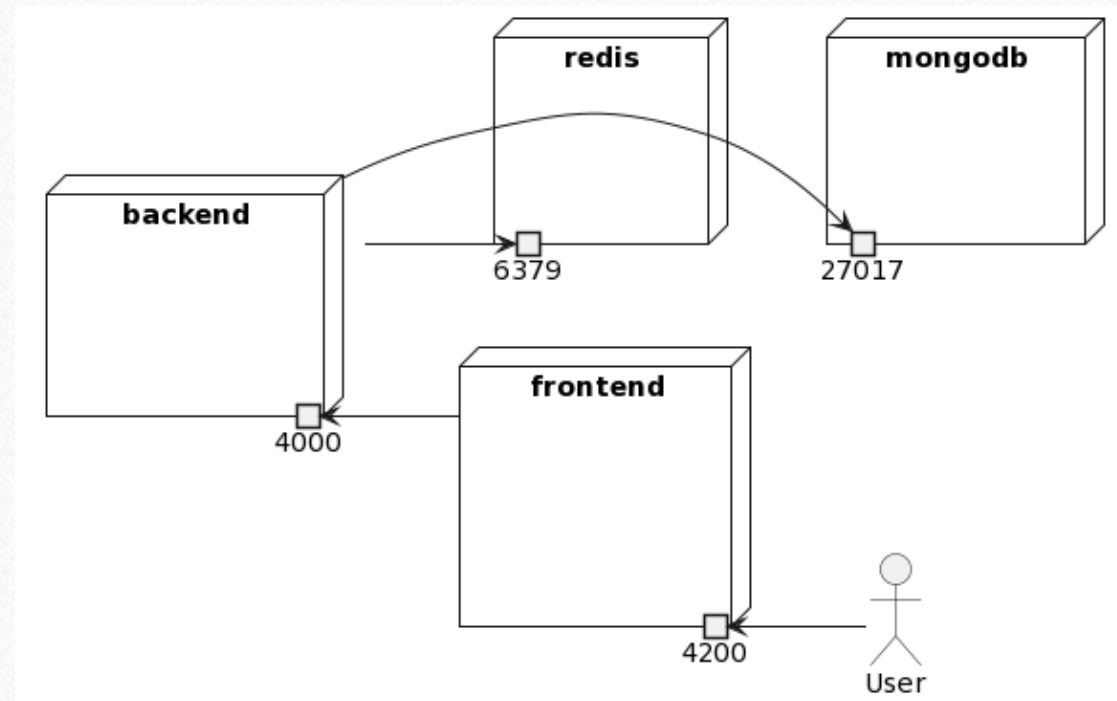
Docker: soluzione software che permette di deployare un'applicazione tramite *container*: pacchetti software vergini, minimali e con tutte le dipendenze necessarie.



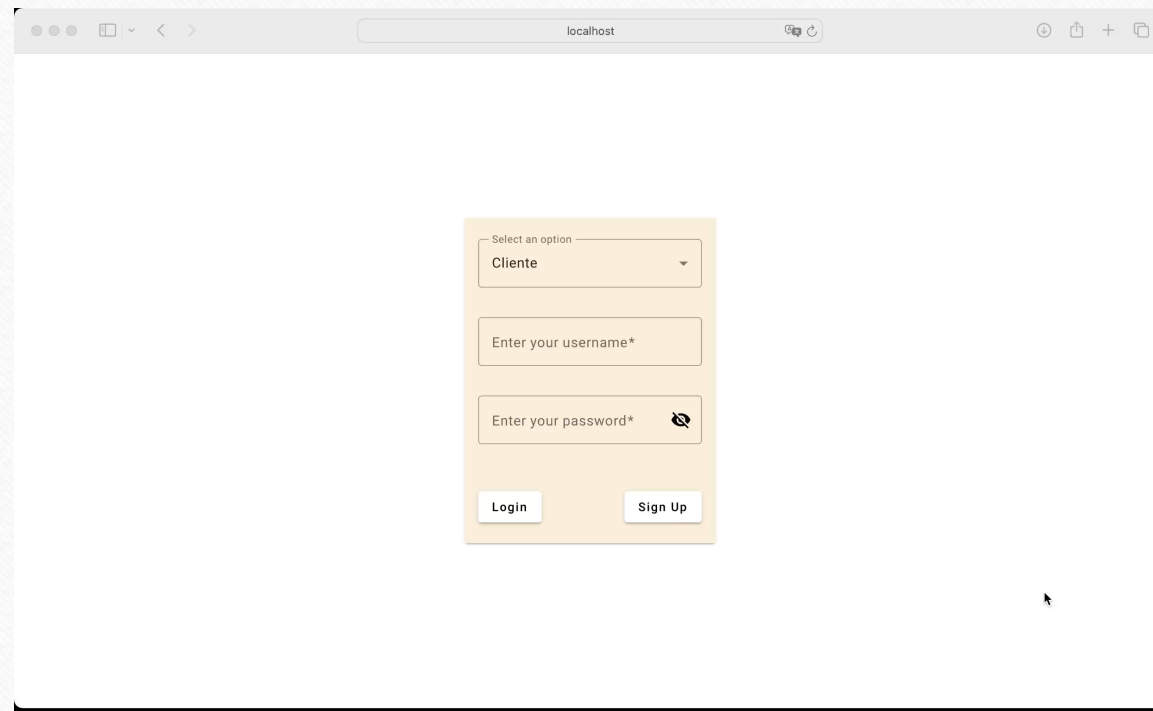
Deployment Structure

Usufruento della funzionalità *docker compose*, sono stati istanziati 4 container che costituiscono lo scheletro dell'applicazione:

- **Redis** e **MongoDB**, utilizzando le immagini disponibili sul Docker Hub;
- Buildando 2 immagini locali per il codice di **backend** e **frontend**



Demo



A web browser window is shown with the address bar displaying "localhost". The browser window contains a login form with the following elements:

- A dropdown menu labeled "Select an option" with "Cliente" selected.
- A text input field labeled "Enter your username*".
- A text input field labeled "Enter your password*" with a toggle icon.
- Two buttons: "Login" and "Sign Up".