PROGETTO WEB APP

ARCHITETTURA DI UNA WEB APP

**FRONT END / BACK END**

Una web app è strutturata da una parte di front end e una parte di back end che denotano, rispettivamente, la parte visibile all'utente di un programma e con cui egli può interagire e la parte che permette l'effettivo funzionamento di queste interazioni. Il *front end*, nella sua accezione più generale, è responsabile dell'acquisizione dei dati di ingresso e della loro elaborazione con modalità conformi a specifiche predefinite e invarianti, tali da renderli utilizzabili dal *back end*. Il collegamento del *front end* al *back end* è un caso particolare di interfaccia.

**FRAMEWORK**

Per lo sviluppo di una web app è necessario un Framework ossia un'architettura logica di supporto sulla quale un software può essere progettato e realizzato, spesso facilitandone lo sviluppo da parte del programmatore. Il framework non è un programma autonomo, ma appare piuttosto come un tipo particolare di libreria standard composta da una collezione di classi e funzioni.

Vediamo alcuni esempi di Framework Backend:

1. **Python**: Django, Flask
2. **Java**: Spring e Hibernate

Vediamo ora alcuni esempi di Framework Frontend:

1. **Javascript**: Angular, React, VueJS
2. **CSS:** Bootstrap, Foundation

**DATABASE**

Tutte le web app dispongono di un database per la raccolta e la memorizzazione dei dati. I database possono essere suddivisi in due diversi tipi:

I database **RDMS** (**Relational Database Management System**) sono un tipo di [database](https://www.oracle.com/it/database/what-is-database.html) di archiviazione che fornisce accesso a data points correlati tra loro. I database relazionali sono basati sul modello relazionale, un modo intuitivo e diretto di rappresentare i dati nelle tabelle. In un database relazionale ogni riga della tabella è un record con un ID univoco chiamato chiave. Le colonne della tabella contengono gli attributi dei dati e ogni record di solito ha un valore per ogni attributo, rendendo facile stabilire le relazioni tra i data points.

Ecco alcuni esempi di DBMS più famosi sul mercato: MySQL, SQLite, PostgreSQL, Microsoft SQL.

   

I database **NoSQL** vengono definiti "non relazionali", "DB NoSQL" o "non SQL" per evidenziare il fatto che sono in grado di gestire volumi elevati di dati non strutturati in rapida evoluzione in modi diversi rispetto a un database relazionale (SQL) con righe e tabelle.

Ecco alcuni esempi di database NoSQL più famosi sul mercato: Elasticsearch, MongoDB, Firebase.

  