

# Siti web dinamici

[marco.abbadini@unibg.it](mailto:marco.abbadini@unibg.it)

<https://cs.unibg.it/abbadini>

# Front-end, Back-end, database

- Il front-end generalmente tratta gli aspetti dell'applicazione che un utente può vedere
  - Html, CSS, Javascript, ...
- il back-end tratta invece tutti gli aspetti che riguardano il suo funzionamento
  - Php, Python, Java, NodeJs, ...
- Database
  - MySQL, SQLServer, PostgreSQL, SQLite
  - Database NoSql

# FRONT-END



Web App

Mobile App

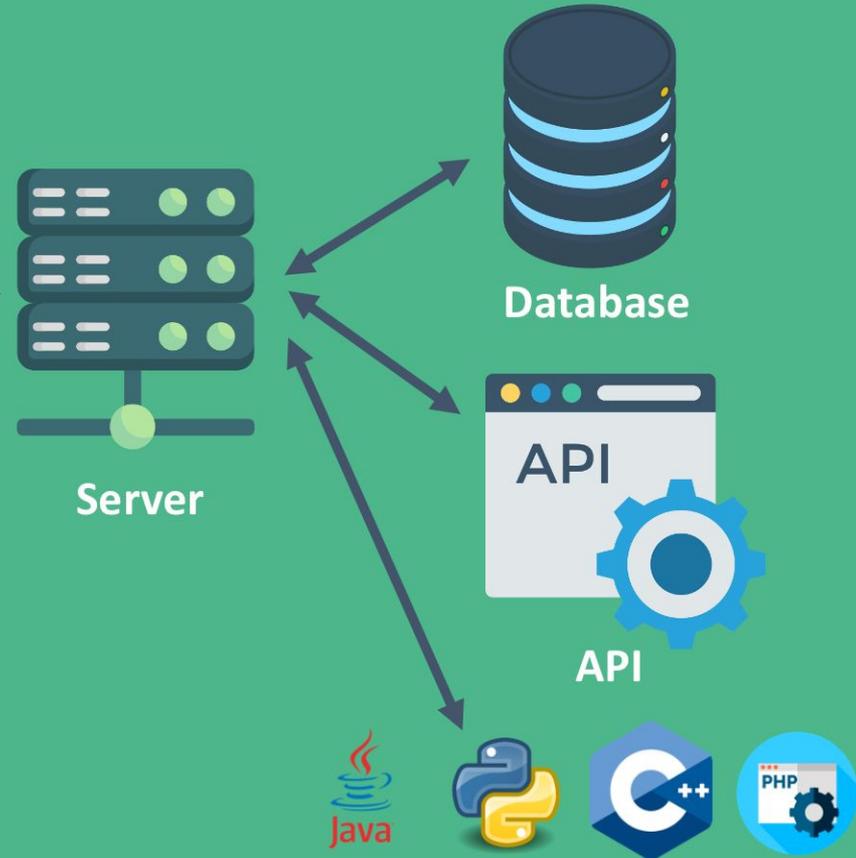
HTML

CSS

JS

JS

# BACK-END



Server

Database

API

API

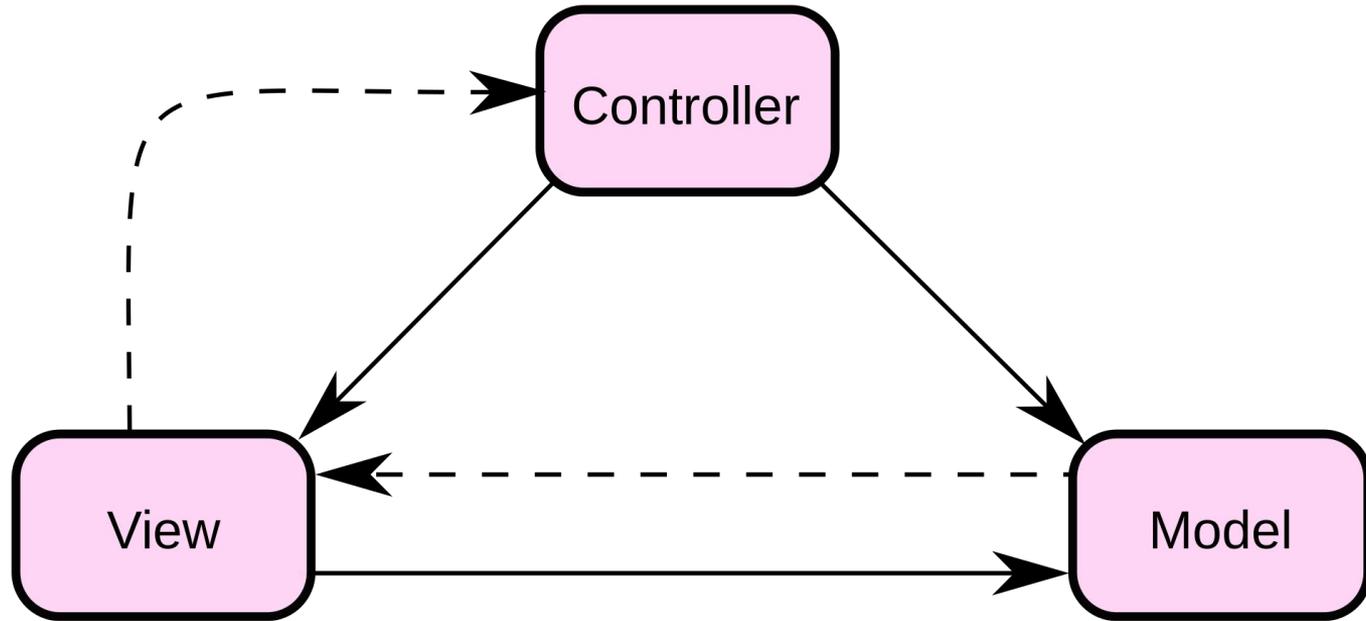
Java

Python

C++

PHP

# Model-View-Controller (1)



## Model-View-Controller (2)

- **Model:** il componente principale dell'applicazione, gestisce i dati, la logica e le regole dell'applicazione
- **View:** qualsiasi rappresentazione grafica dell'applicazione
- **Controller:** gestisce gli input e determina cosa devono fare il Model e la View

Generalmente:

1. Il Model gestisce i dati dell'applicazione e riceve l'input dell'utente da parte del Controller
2. La View gestisce la rappresentazione grafica del Model
3. Il Controller riceve gli input dell'utente e interagisce con il Model

# Backend

Ad oggi ci sono diverse alternative disponibili per sviluppare il lato backend di un'applicazione. Diversi linguaggi consentono di utilizzare tool differenti.

- **JavaScript** può essere utilizzato per sviluppare il backend di un'applicazione utilizzando NodeJS
- Le Servlet sono oggetti **Java** che consentono di sviluppare applicazioni web gestendo l'interazione con il server
- **Php** è stato (ed è ancora oggi) utilizzato per implementare gran parte delle infrastrutture backend di moderne applicazioni (e.g., Facebook)

# Python per il backend

Noi vedremo un breve esempio di backend sviluppato in Python

Il framework Python più utilizzato è **Django** (e.g., Instagram)

- Velocità di sviluppo
- Scalabile
- Stabile
- Completo di funzionalità

Tuttavia inizialmente è complesso per chi si trova alle prime armi

# Flask

Noi utilizzeremo **Flask**

**Flask** è un micro-framework che consente di sviluppare applicazioni web in Python

Solitamente è utilizzato per gestire la fase di sviluppo di un progetto

Per noi, al momento, è più che sufficiente a mostrare come è possibile sviluppare una rapida applicazione web che interagisca con una base di dati

# Database

Per gestire l'interazione con i dati persistenti di un'applicazione è necessario collegare la nostra web app a una base di dati

Ci sono diverse soluzioni disponibili:

- MySQL
- PostgreSQL
- SQL Server

Noi utilizzeremo **SQLite** che consente di creare e gestire una base di dati in modo rapido e leggero



**LET'S DO THIS!**

**THANKS FOR LISTENING**



**ANY QUESTIONS?**