

Modulo Laravel

Introduzione Internet

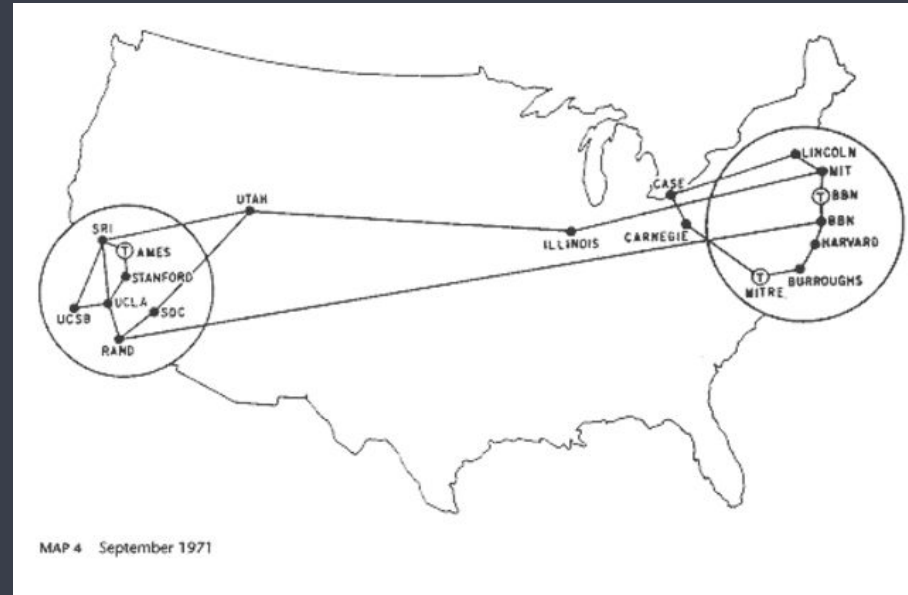
La Storia di Internet

1969 - A.R.P.A., un ente governativo americano per la difesa, sviluppa ARPANET (Una rete militare finalizzata allo scambio di informazioni, un sistema che doveva essere veloce e sicuro).

Utilizzando 4 poli universitari:

- UCLA (Università californica Los Angeles)
- SRI (Istituto di Ricerca)
- UCSB (Università californica Santa Barbara)
- UUSC (Università Utah ella School Computing)

1972 - Prima connessione nazionale in ARPANET.



La Storia di Internet

- Architettura Client/Server
- TCP/IP (Transmission Control Protocol/Internet protocol)
- Ampiezza di banda 50Kbs (<https://www.youtube.com/watch?v=OGwlwZs9PxE>)
- Era possibile mandare dei pacchetti testuali.
- Il primo messaggio su ARPANET fu inviato da uno studente di nome Charley Kline alle 10:30 del 29 ottobre 1969 con scritto: "Lo" Fonte: <https://www.icann.org/en/blogs/details/the-first-message-transmission-29-10-2019-en>

Internet Bandwidth Comparisons

Connection type	Speed		Transmit 1 GB		1 MB	10 K
	bps	B/sec	Days	hh : mm : ss	mm : ss	Seconds
28.8 Modem	28.8 K	3.5 K	3	10 : 51 : 01.62	4 : 51.27	2.8444
56K Modem	56 K	6.8 K	1	18 : 36 : 31.69	2 : 29.80	1.4629
DS0	64 K	8.0 K	1	12 : 24 : 32.00	2 : 08.00	1.2500
2 Channel ISDN	128 K	15.6 K		18 : 38 : 28.86	1 : 05.54	0.6400
Common DSL	512 K	62.5 K		4 : 39 : 37.22	: 16.38	0.1600
Cable	1 M	122.1 K		2 : 23 : 09.93	: 8.39	0.0819
T1	1.544 M	188.5 K		1 : 32 : 43.43	: 5.43	0.0531
T2	6.312 M	770.5 K		: 22 : 40.89	: 1.33	0.0100
Ethernet	10.000 M	1.220 M		: 14 : 18.99	: 0.84	0.0082
T3	44.736 M	5.461 M		: 3 : 12.01	: 0.19	0.0018
OC-1	51.840 M	6.328 M		: 2 : 45.70	: 0.16	0.0016
Fast Ethernet	100.00 M	12.207 M		: 1 : 25.90	: 0.08	0.0008
OC-3	155.52 M	18.984 M		: : 55.23	: 0.05	0.0005
OC-12	622.08 M	75.937 M		: : 13.81	: 0.01	0.0001

La Storia di Internet

1983: 213 macchine connesse ad Arpanet iniziano ad usare i protocolli TCP/IP, e l'Università del Wisconsin crea il **DNS (Domain Name System)** per facilitare la comunicazione a host conosciuti

Ma sempre nel 1983 ARPA esaurì il suo scopo: lo stato chiuse l'erogazione di fondi pubblici, la sezione militare si isolò, necessitando di segretezza assoluta a protezione delle proprie informazioni, e nacque perciò MILNET.

1990: Con il passare del tempo, l'esercito si disinteressò sempre più del progetto fino ad abbandonarlo e rimase sotto il pieno controllo delle università, diventando un utile strumento per scambiare le conoscenze scientifiche e per comunicare.

Arpanet, formata da 300 mila macchine, lascia il posto ad **Internet**. Cominciano a svilupparsi le **backbone** (dorsali ad alta velocità, vedi micelio funghi). Presso il CERN di Ginevra, è sviluppato il protocollo per lo scambio di immagini oltre che di dati (ipertesto), attraverso il linguaggio HTML.

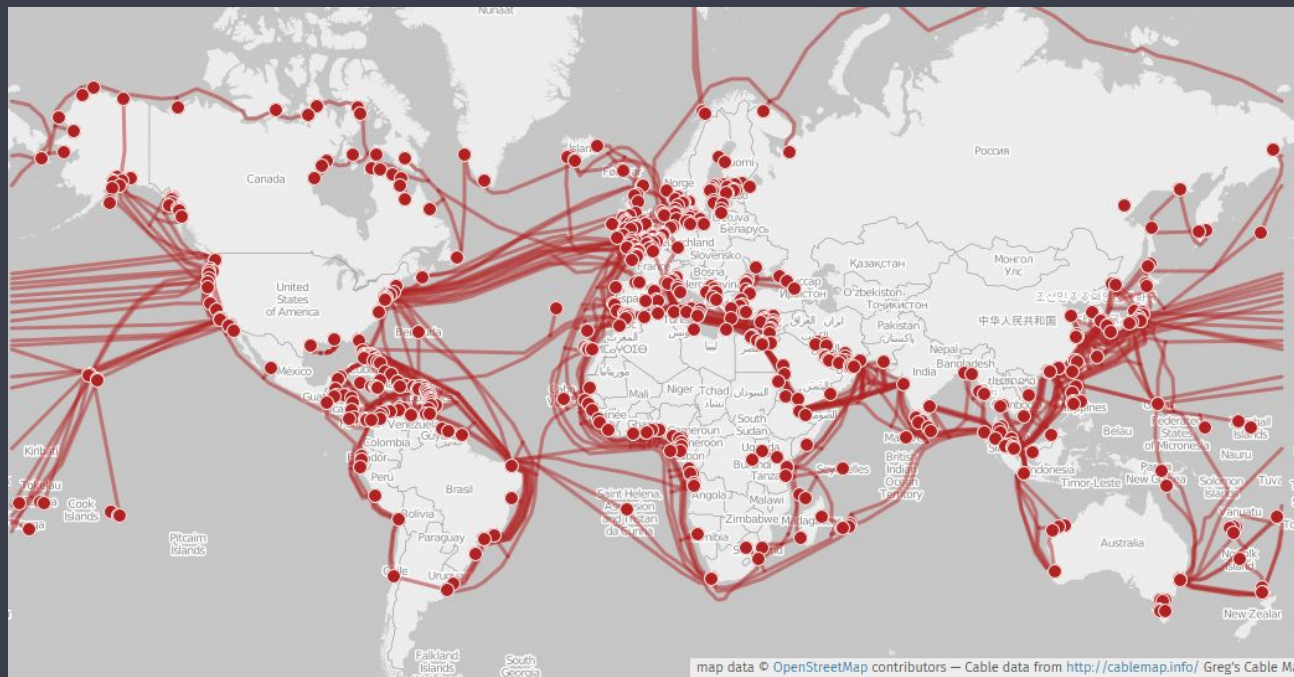
Backbone



Curiosità

- [Squali attaccano internet](#)
- [Video GeoPOP](#)

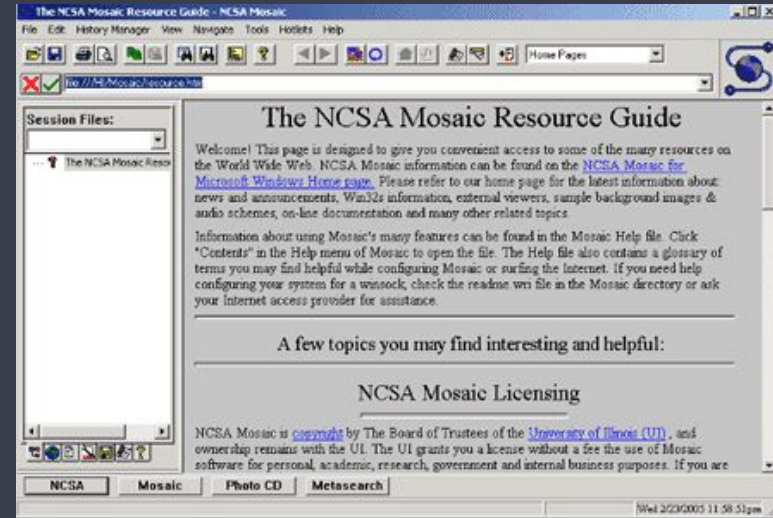
Mappa Backbone



II WWW

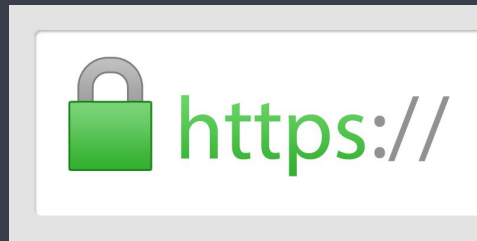
- World Wide Web o W3
- Nato nel 1991 al CERN di Ginevra
- Il Suo creatore è Tim berners-Lee
- Definì il protocollo HTTP (HyperText Transfer protocol)
- Definì il linguaggio HTML (HyperText Markup language)
- Nel 1993 fù scritto il primo Browser chiamato MOSAIC
- Primo sito web:

<http://info.cern.ch/hypertext/WWW/TheProject.html>



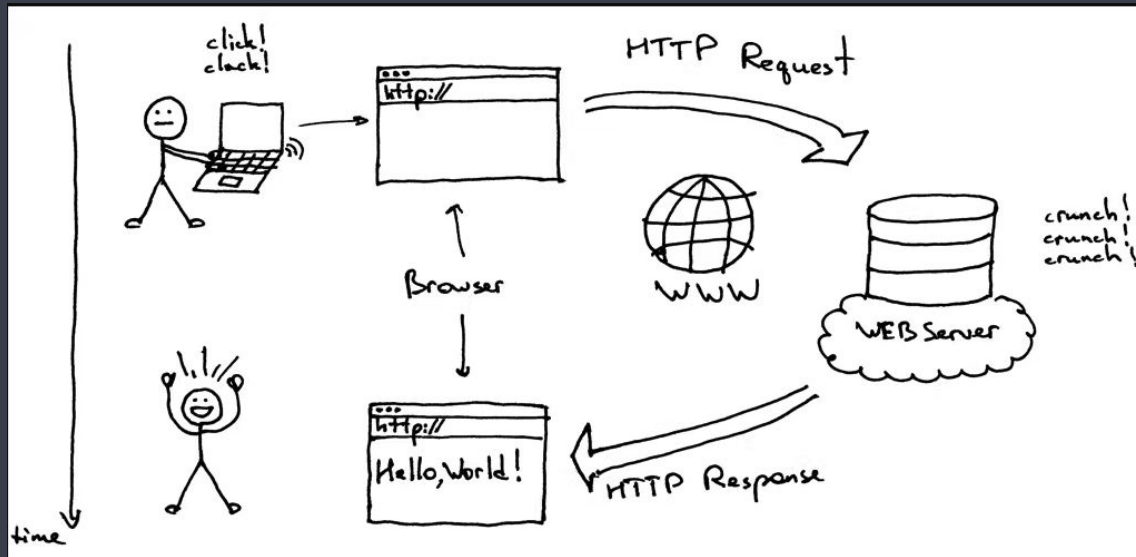
II WWW

- **HTTP (ora HTTPS):** Un server HTTP generalmente resta in ascolto delle richieste dei client sulla porta 80 usando il protocollo TCP a livello di trasporto.
- **HTML:** L'HTML è un linguaggio di pubblico dominio, la cui sintassi è stabilita dal World Wide Web Consortium (W3C).
<https://github.com/whatwg/html>
- **URL:** Uniform Resource Locator, è una sequenza di caratteri che identifica univocamente l'indirizzo di una risorsa su una rete di computer,



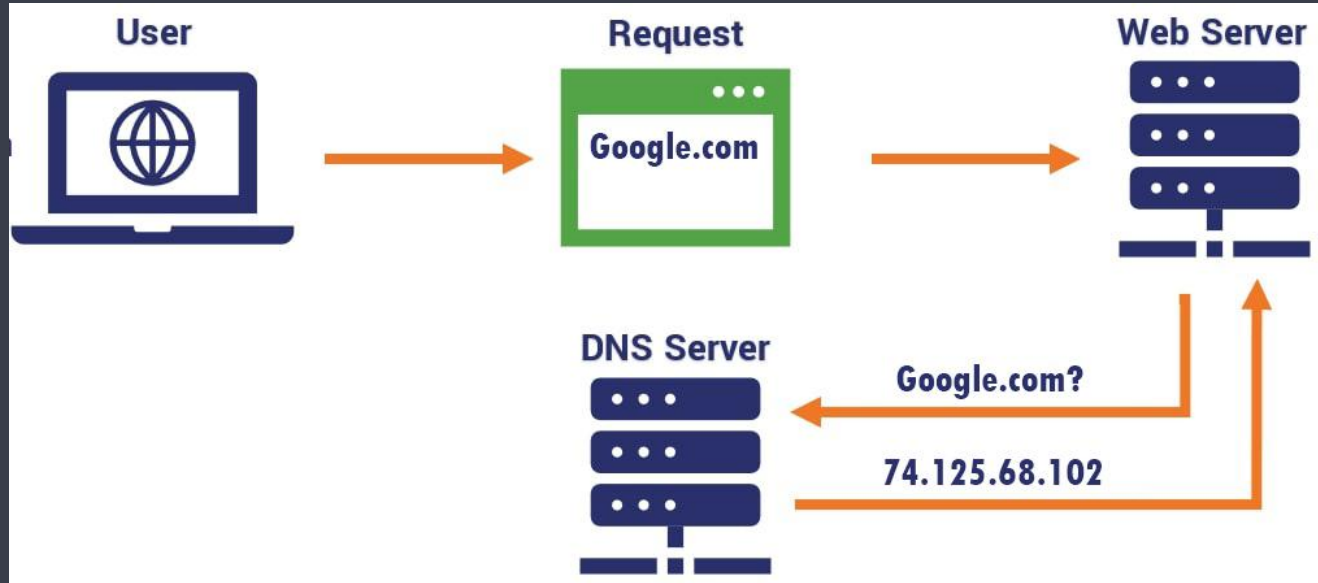
HTTP

- Modello Client/Server
- Client = Browser
- Server = Server Remoto
- Due tipi di messaggi:
 - Request = Richiesta
 - Response = Risposta



- HTTP è Stateless, ovvero il che significa che una volta termina la connessione tra il browser e il server viene persa. Si utilizzano i Cookie HTTP

HTTP - IP e DNS



Vediamo in Ispeziona il sito <https://www.google.com/>

HTTP - Metodi

- Metodi:
 - GET: Ottenere Una Risorsa, una pagina WEB. Ha sempre una Intestazione (HEADER). mai un Corpo (Body)
 - POST: Quando mandiamo una richiesta al server di creare una nuova risorsa, mandiamo un HEADER e un BODY con i dati sensibili da mandare.
 - PUT: Quando abbiamo già creato una risorsa e dobbiamo modificarla in totale, mandiamo un HEADER e un BODY;
 - PATCH: Quando dobbiamo modificare una parte della risorsa precedentemente creata, mandiamo un HEADER e un BODY;
 - DELETE: Quando dobbiamo eliminare una risorsa, mandiamo un HEADER e un BODY;

HTTP - Metodi

Create → **POST**

Read → **GET**

Uppdate → **PUT**

Delete → **DELETE**

HTTP
Methods

HTTP - Status Code

- 1xx—Informational: Il server non ha completato completamente la richiesta, sta ancora pensando ed è in una fase di transizione.
- 2xx—Successful: Il server ha completato con successo la richiesta
- 3xx—Redirects: Questo blocco è per i reindirizzamenti, significa che hai richiesto un indirizzo ma sei stato inviato altrove
- 4xx—Client Errors: C'è qualche errore dalla tua parte
- 5xx—Server Errors: C'è qualche errore dalla parte del server

Tutti gli stati: https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

Cos'è Laravel

- Laravel è un Framework PHP con architettura MVC (Model View Controller)
- Il creatore si chiama Taylor Otwell (<https://twitter.com/taylorotwell/status/1182663227525734404>).
- Il progetto è completamente Open Source* da giugno 2011 ed è disponibile al seguente link:
<https://github.com/laravel/laravel>
- Attualmente conta più di 800 Contributor che dedicano gratuitamente il loro tempo libero all'evoluzione del progetto
- Ovviamente esistono anche Contributor stipendiati (Laravel Core Team) che si occupano della manutenzione del codice, bug fixing e falle di sicurezza che necessitano di interventi repentini.



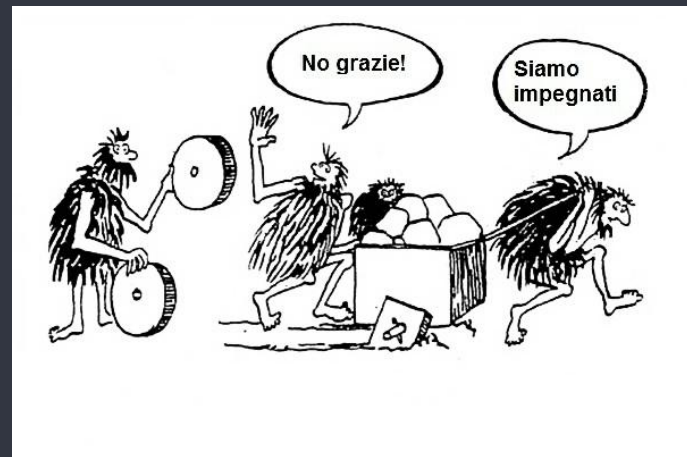
*Open-source - [Approfondimento TedTalk](#)

Cosa significa Framework

Un framework non è un linguaggio di programmazione, ma è un'architettura logica di supporto sulla quale un software può essere progettato e realizzato.

Bootstrap è un Framework CSS, non è un linguaggio di programmazione ma una serie di classi pre-costruite che ti permettono uno sviluppo rapido:

Non bisogna reinventare la ruota ad ogni progetto.

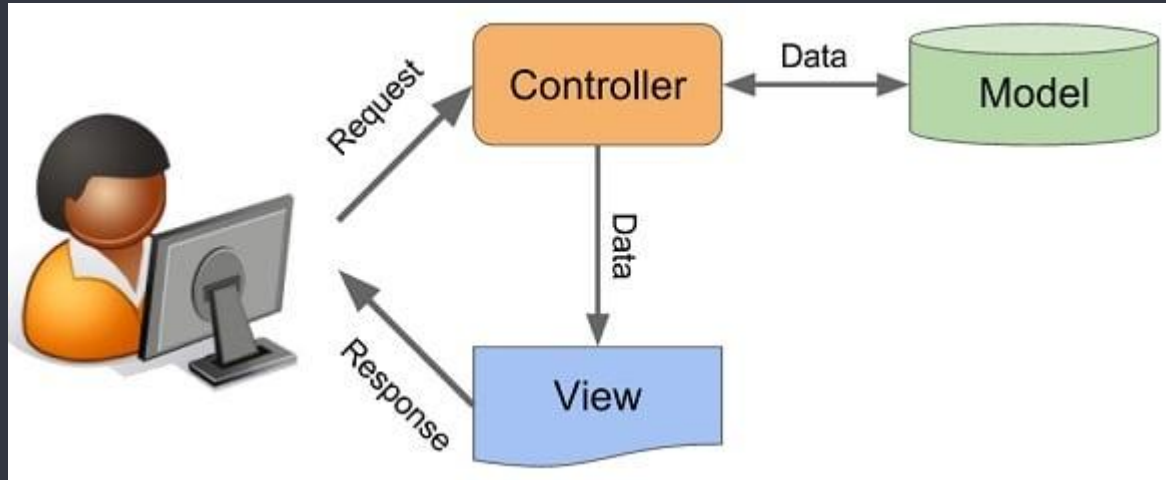


Cosa significa MVC

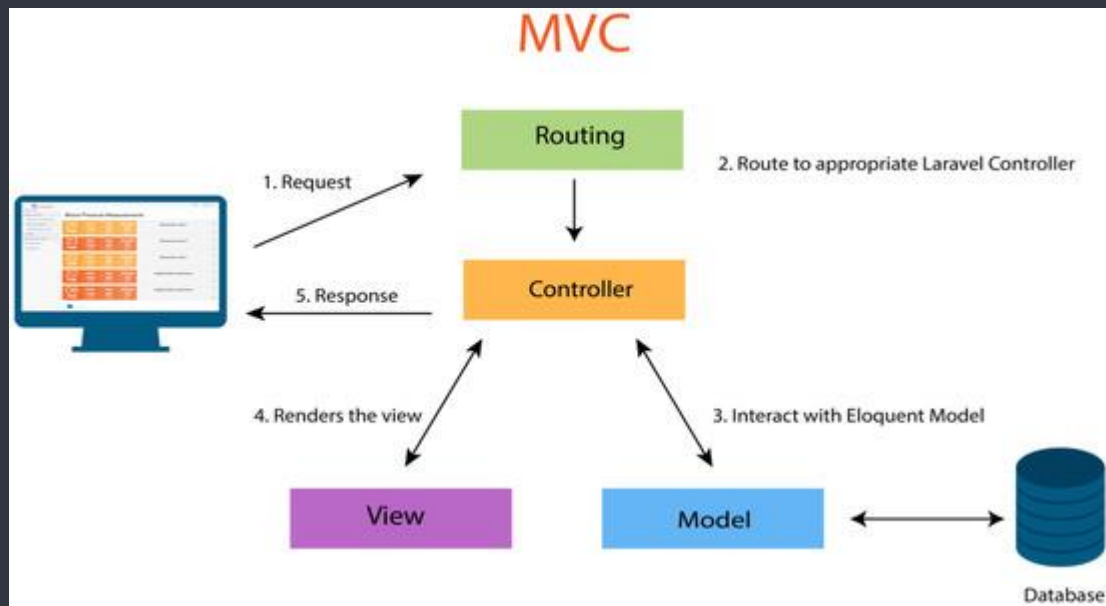
MVC — Model-View-Controller — è uno schema architetturale molto diffuso nello sviluppo di software, visto che si occupa della suddivisione del flusso logico in tre blocchi interconnessi tra loro.

- **Funzionalità di Business [Model]:** Il blocco che gestisce direttamente i dati, i principi logici e le regole dell'applicazione. Questo significa che lo stesso dato può avere un output diverso in base all'interfaccia grafica e al tipo di richiesta, il tutto senza modificare l'informazione sorgente.
- **Logica di Controllo [Controller]:** Il nucleo del sistema (Core) è il blocco che contiene materialmente il codice con tutte le istruzioni scritte dallo sviluppatore. Intercetta gli input dell'utente, li elabora, interroga i dati e genera un output non interpretato.
- **Logica di Presentazione [View]:** Interfaccia grafica (GUI) o Vista (View), è il blocco che si interfaccia direttamente con l'utente finale. Qui vengono visualizzati ed interpretati tutti gli output generati dal sistema.

Cosa significa MVC

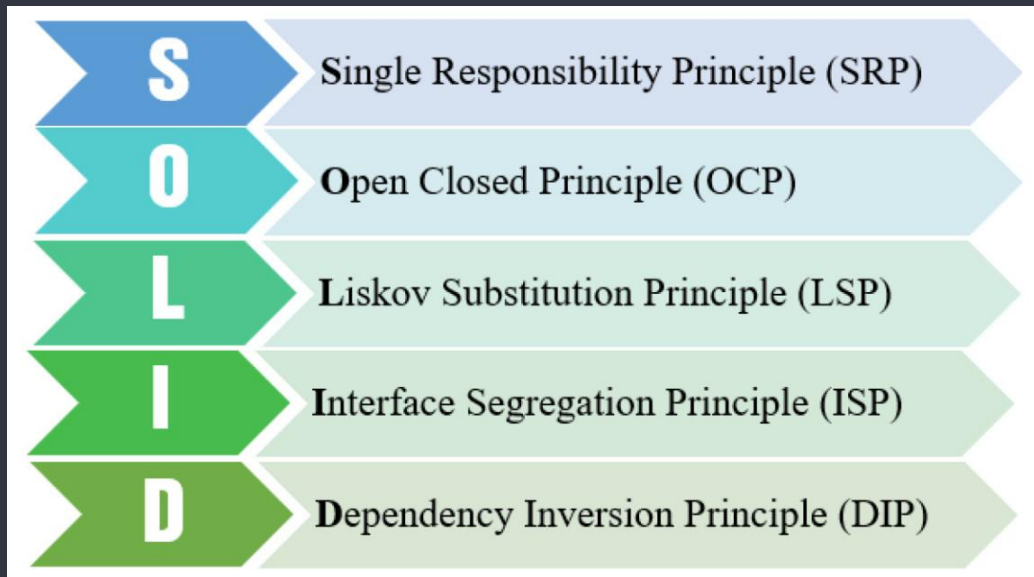


Cosa significa MVC in Laravel?



Principi S.O.L.I.D.

Di seguito i principi SOLID della programmazione ad oggetti.



Principi S.O.L.I.D.

[S] Single responsibility principle (o principio di singola responsabilità): Ogni classe dovrebbe avere una ed una sola responsabilità. Se stai realizzando una logica complessa, separa tutto in piccole funzioni che restituiscono un risultato.

Ne gioverai sia per quanto riguarda la robustezza del codice, in caso di problemi potrai isolare meglio la parte incriminata e correggerla sia per il riuso del codice.

Fonte: <https://accesto.com/blog/solid-php-solid-principles-in-php/>

Principi S.O.L.I.D.

[O] Open/closed principle (o principio aperto/chiuso): Ragiona sempre secondo il principio di scalabilità. Ovvero un'entità software dovrebbe essere aperta alle estensioni, ma chiusa alle modifiche.

Esempio: Hai creato una funzione per leggere un .csv contenente dei dati. E se il fornitore dovesse modificarlo con un .json? Estendere questa nuova funzionalità è possibile o dovrai mettere mano al codice sorgente e modificarlo?

Fonte: <https://accesto.com/blog/solid-php-solid-principles-in-php/>

Principi S.O.L.I.D.

[L] Liskov substitution principle (o principio di sostituzione di Liskov): Questo principio ha lo scopo di guidare il processo di progettazione delle classi all'interno delle gerarchie con l'obiettivo principale di non creare classi figlie che hanno sempre meno in comune con il genitore, ma che richiedono solo alcuni metodi della classe genitore.

Ovvero, se estendiamo una classe dobbiamo assicurarci che questa estensione mantenga lo scopo e l'integrità della classe principale. In caso contrario dobbiamo creare una nuova classe.

Fonte: <https://accesto.com/blog/solid-php-solid-principles-in-php/>

Principi S.O.L.I.D.

[/] Interface segregation principle (o principio di segregazione delle interfacce): Sorvoliamo

Fonte: <https://accesto.com/blog/solid-php-solid-principles-in-php/>

Principi S.O.L.I.D.

[D] Dependency inversion principle (o principio di inversione delle dipendenze): Sviluppare processi logici troppo elaborati fuori dalla classe primaria non è mai buona prassi, in quanto quello stesso sviluppo potrebbe essere utile ad un'altra sottoclasse.

Fonte: <https://accesto.com/blog/solid-php-solid-principles-in-php/>

Cenni: Eloquent ORM

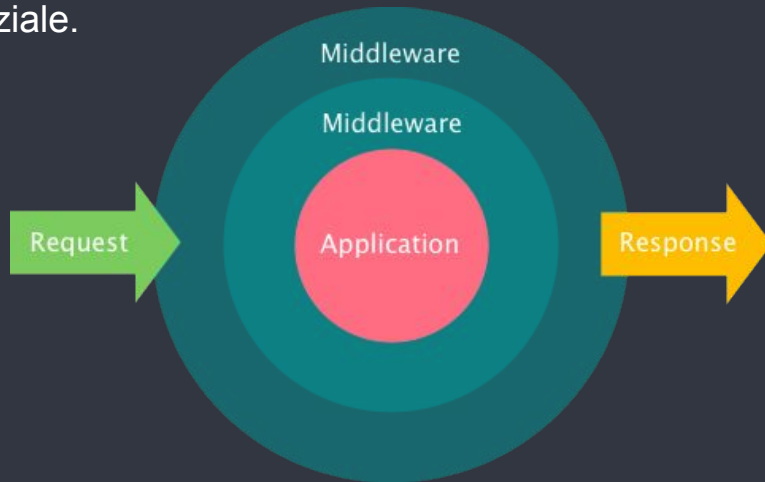
ORM (Object-Relational Mapping) è un'interfaccia che fornisce, tramite programmazione a oggetti, tutti i servizi inerenti alla persistenza dei dati astruendo le caratteristiche del DBMS usato. In parole povere **ce ne fregiamo del tipo di DB!**

Possiamo gestire le entità e le relazioni di un DBMS direttamente tramite Laravel con Eloquent in modo agevole scrivendo poche righe di codice e soprattutto chiare, interfacciarsi con un linguaggio piuttosto che un altro non ci interessa.

Cenni: Middleware

Una applicazione web, in fin dei conti, non è altro che un'interfaccia grafica in grado di lanciare una serie di chiamate HTTP al server contenenti un input e, di tutta risposta, il server stesso si prodigherà nel comunicare l'output al browser/client.

Il Middleware (o più middleware) si pone in mezzo a questo scambio di informazioni, una specie di filtro tra il Client e il Server, in modo tale da evitare sovraccarichi inutili al server per chiamate non previste dal flow iniziale.



Cenni: Facades

Le facciate (traduzione letteraria), non sono altro che un'interfaccia semplificata di un sottosistema più complesso.

Tramite l'assegnazione di un Aliases, sarà più facile richiamare e ricordare la classe prescelta.

Il concetto di Facade è forse l'argomento più amato e odiato al tempo stesso. Il codice è subito leggibile e facilmente interpretabile grazie alle facade. Ma la continua staticizzazione dei metodi per molti è un limite in termini di performance.



```
Test::print('ciao');
```

Il primo Progetto Laravel

Ci serviranno:

- PHP
- COMPOSER
- LARAVEL INSTALLER

Il primo Progetto Laravel 2



```
francescomansi@192: ~  
laravel new primo-progetto-laravel
```

A terminal window with a dark background. The title bar shows 'francescomansi@192: ~'. The command 'laravel new primo-progetto-laravel' is entered and executed, indicated by a green checkmark in the top right corner.



```
francescomansi@192: ~/desktop/primo-progetto-laravel  
INFO Application ready in [primo-progetto-laravel]. Build something amazing.  
~/desktop cd primo-progetto-laravel  
~/desktop/primo-progetto-laravel _
```

A terminal window with a dark background. The title bar shows 'francescomansi@192: ~/desktop/primo-progetto-laravel'. The first line is an informational message: 'INFO Application ready in [primo-progetto-laravel]. Build something amazing.' The second line shows the command 'cd primo-progetto-laravel' being executed from the desktop directory. The third line shows the current directory as '~/desktop/primo-progetto-laravel'. A yellow progress bar in the top right corner indicates a duration of '1m 11s'.

Comandi Artisan

A terminal snippet showing the output of a git status command, which is 'git master', followed by the command 'php artisan'. The text is displayed in a light blue monospace font on a dark background.

```
git master php artisan
```

<https://laravel.com/docs/11.x/artisan#main-content>

Lancia il server locale



PROBLEMI TERMINALE PORTE

~/Desktop/primo-progetto-laravel

```
php artisan serve
```

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server

Comando Artisan