Vagrant

Installare <https://www.virtualbox.org/wiki/Downloads>

Installare [vagrant.com](http://vagrant.com/)

Scaricare **laravel/homestead** => <https://app.vagrantup.com/boxes/search>

Apri Console/Git Bash/altro

**vagrant** #se mi dà l’elenco delle opzioni, è installato

**vagrant box add laravel/homestead** #seleziono virtualbox. NON LO FACCIO PERCHE’ PRENDE TEMPO

**git clone https://github.com/Swader/homestead\_improved.git LARAPROJECTS**

**cd LARAPROJECTS**

Apri **Homestead.yaml**

* Ho l’**IP** che mapperà la macchina virtuale, **provider**=virtualbox, **sites** mi dice i siti virtuali che creo dentro **LARAPROJECTS**, i **database** aggiunti,

**vagrant up** #se assente, scarica la VM, e poi la avvia (dà anche info sul forwarding delle porte)

vagrant status

**vagrant ssh** #da ora la console sarà quella di Ubuntu

**ls -al** #ora vedo la famosa cartella **/home/vagrant/Code/**

#se creo la cartella **test** nel mio PC, questa appare anche nella VM

Devo modificare anche il file hosts con 192.168.10.10

**vagrant reload --provision** #se ho modificato hosts devo lanciare questo x aggiornare

Installeremo putty per accedere con ssh alla macchina virtuale. Possiamo farlo  in windows anche con gitbash

composer create project laravel/laravel --prefer-dist

Composer

<https://packagist.org/>

{

"require": {

"vendor/pacchetto": "1.3.2",

"vendor/package2": "1.\*",

"vendor/package3": "^2.0.3"

}

}

Posso usare **PUTTY**, uso la sessione **homestead** da me creata (**user=pass=vagrant**)

Esco dalla cartella /Code e lancio

**composer create-project laravel/laravel –prefer-dist Code/lara\_1**

LARAVEL

INSTALLAZIONE : DALLA CARTELLA WWW LANCIO

**composer create-project laravel/laravel –prefer-dist lara\_1**

IMPORTAZIONE PROGETTO: SU WWW COPIO LA CARTELLA, CI VADO DENTRO. CREO IL FILE .env E LO COMPILO, CREO IL DB VUOTO SU PHPMYADMIN (CORRISPONDENTE A QUESTO PROGETTO) E LANCIO

**composer update** //crea la cartella vendor

**php artisan migrate** //crea le tabelle del DB

AGGIUNGI GLI HELPERS, ad esempio la funzione input() per prendere parametri dalla REQUEST\_URI. In realtà proprio quell’helper non lo riconosce, da verificare se serve per gli altri, altrimenti questo pacchetto si può mollare

**composer require laravel/helpers**

AGGIUNGI COMPONENTI: crea la classe **app/View/Components/Alert.php** e la vista **resoulces/views/components/alert.blade.php**

**php artisan make:component Alert**

**php artisan make:component Card –inline //**crea la Class, non la view, nel render della classe ci sarà l’HTML della view

INSTALLA BOOTSTRAP

**composer require laravel/ui**

**php artisan ui bootstrap --auth**

**npm install**

**//npm install bootstrap**

**npm run dev**

ESEGUIRE TUTTE LE MIGRATION DENTRO /migration

**php artisan migrate**

php artisan make:command –help

php artisan make:command --help

make:migration PER CREARE UNA MIGRATION

php artisan make:migration -h

Usage:

make:migration [options] [--] <name>

// <name> DEVE ESSERE UN NOME PARLANTE

Options:

--create[=CREATE] The table to be created

--table[=TABLE] The table to migrate

--ansi Force ANSI output

--no-ansi Disable ANSI output

-n, --no-interaction Do not ask any interactive question

--env[=ENV] The environment the command should run under

-v|vv|vvv, --verbose Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

---------------------------------------------------------

php artisan make:migration crea\_tabella\_albums --create albums //DEFINISCO TABELLA Albums

php artisan make:migration crea\_tabella\_photos --create=photos //DEFINISCO TABELLA Photos

php artisan migrate //ESEGUO LE MIGRATION: CREO LE TABELLE NEL DB

php artisan migrate:rollback //ANNULLO L'ULTIMA MODIFICA

php artisan migrate:fresh //SVUOTA E RICREA DA 0 LE TABELLE NEL DB

php artisan make:migration modifica\_albums-album\_name --table albums //MODIFICO TABELLA

composer require doctrine/dbal //SERVE QUESTO PACKAGE DA COMPOSER

php artisan make:seeder UsersTableSeeder //CONVENZIONE NOME: nome\_table.'TableSeeder'

php artisan db:seed --class=UsersTableSeeder //LANCIA IL SEEDER

php artisan db:seed //LANCIA IL SEEDER DI DEFAULT. LO USERO' X CHIAMARE TUTTI I SEEDER

php artisan make:seed AlbumsTableSeeder //CREO UN SEEDER ANCHE PER albums

php artisan db:seed --class=AlbumsTableSeeder //LANCIA IL SEEDER

SELECT \* FROM albums AS a INNER JOIN users AS u ON a.user\_id = u.id;

php artisan make:model Models\\Photo -m //CREA MODEL E MIGRAZIONE: ATTENZIONE CHE NON CREI 2 MIGRAZIONI PER LA STESSA TABELLA

php artisan make:seed PhotosTableSeeder //CREA IL SEEDER

php artisan db:seed --class=PhotosTableSeeder //LANCIA IL SEEDER

php artisan db:seed //SEEDER DI DEFAULT. LO MODIFICO XKè CHIAMI TUTTI I SEEDER

php artisan migrate:refresh --seed //SVUOTA LE TABELLE E LANCIA I SEEDER: NON FUNGE

php artisan storage:link //LINK SIMBOLITO TRA \public\storage E \storage\app/public

**php artisan make:controller PhotosController --resource** //RESOURCE CONTROLLER, CON I METODI

php artisan route:list //ROTTE PREGENERATE PER IL RESOURCE CONTROLLER

php artisan vendor:publish //RECUPERA LE VIEW DEI VARI VENDOR

php artisan make:request AlbumRequest //REQUEST: PER VALIDARE I DATI SU UN FORM

nvm install latest //INSTALLARE nvm X WINDOWS

//nvm install 12.0.0 //INSTALLIAMO node.js

nvm list

nvm use 14.3.0

composer require laravel/ui //PRIMA ERA php artisan make:auth

npm install

npm run dev

php artisan ui bootstrap --auth

//php artisan ui vue --auth

php artisan make:policy AlbumPolicy --model=Album

php artisan make:policy PhotoPolicy --model=Photo

php artisan make:controller GalleryController

**php artisan make:model Models\Category -m** //CREA MODEL E ANCHE LA MIGRATION, TANTO DOVRO’ CREARE ANCHE LA TABELLA NEL DB

php artisan make:seed SeedAlbumCategoriesTable //SEED PER LA TABELLA categories

php artisan make:controller CategoryController --resource

php artisan route:list --name=album -v //VEDO SOLO LE ROTTE DI album

php artisan make:request CategoryRequest

php artisan make:migration add\_user\_id\_column\_categories --table=categories

php artisan migrate

php artisan make:middleware VerifyIsAdmin

php artisan make:controller Admin\AdminUsersController --resource // CREA ANCHE LA CARTELLA

composer require yajra/laravel-datatables-oracle //DATATABLES LATO SERVER

php artisan vendor:publish --provider="Yajra\DataTables\DataTablesServiceProvider"

php artisan make:request UserFormRequest //REQUEST: PER VALIDARE I DATI SU UN FORM

php artisan make:mail TestEmail //GLI DICO IL NOME DELLA CLASSE

php artisan logs:clear //NON CREDO FUNGA: PER PULIRE laravel.log

php artisan make:mail TestMd -m mails.testemailmd //EMAIL CON TEMPLATE MARKDOWN

php artisan vendor:publish --all //PER SCARICARE SOTTO vendor

php artisan event:generate //CREA Events E Listeners DA EventServiceProvider->listen

php artisan make:mail EmailNotifyAdminNewAlbum

php artisan make:factory CategoryFactory //NOVITA' 5.5: UNA FACTORY PER OGNI TABELLA

php artisan make:factory CategoryFactory -m \App\Models\Category

php artisan tinker //POSSO LANCIARE LA FACTORY DA TERMINAL

factory(App\Models\Category::class,5)->create();

factory(Category::class,5)->create();

php artisan tinker //POSSO CREARE RECORD NELLE MIE TABELLE

$category = new App\Models\Category();

$category->category\_name = ‘Categoria Uno’;

$category->user\_id = 1;

$category->save();