EJ14

1) Abrimos nuestra consola favorita y escribimos el comando **“composer create-project laravel/laravel proyectoLogin”**

```
C:\Users\2daw3\proyectoLogin>composer create-project laravel/laravel proyectoLogin
```

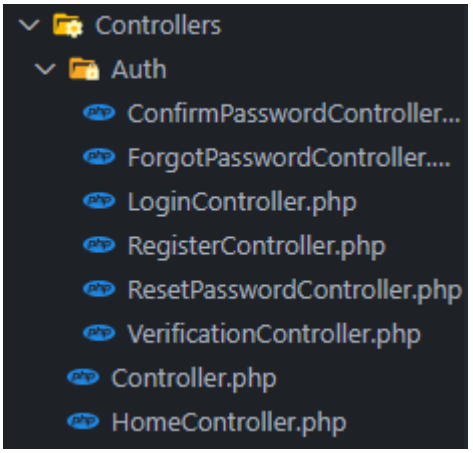
2) Despliega el sistema de autentificación de Laravel con el comando **“ composer require laravel/ui ” y “php artisan ui:auth”**(modificar ficher .env, hacer migración de tablas...).

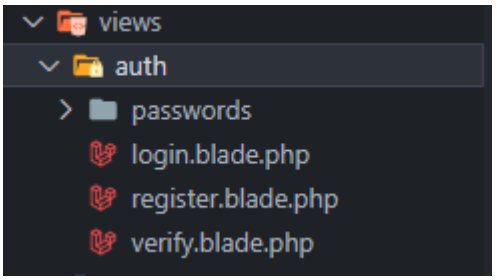
```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=
```

```
PS C:\Users\2daw3\proyectoLogin> php artisan ui:auth
Authentication scaffolding generated successfully.
```

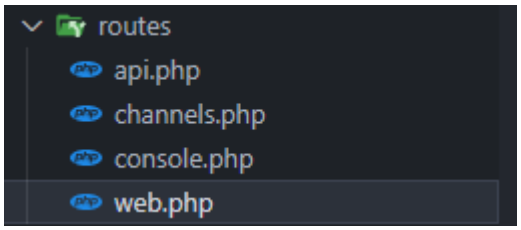
```
PS C:\Users\2daw3\proyectoLogin> composer require laravel/ui
Info from https://repo.packagist.org: #StandWithUkraine
Cannot use laravel/ui's latest version v4.1.1 as it requires php ^8.0 which is not satisfied by your platform.
Using version ^3.4 for laravel/ui
./composer.json has been updated
Running composer update laravel/ui
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
  - Locking laravel/ui (v3.4.6)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
  - Downloading laravel/ui (v3.4.6)
  - Installing laravel/ui (v3.4.6): Extracting archive
Package swiftmailer/swiftmailer is abandoned, you should avoid using it. Use symfony/mailer instead.
Generating optimized autoload files
```

3) **Verificamos** que se haya creado la carpetas correspondientes.





4) Verificamos en los diferentes apartados los cambios que nos a realizado estos comandos:
    a) Rutas: “**routes/web.php**”

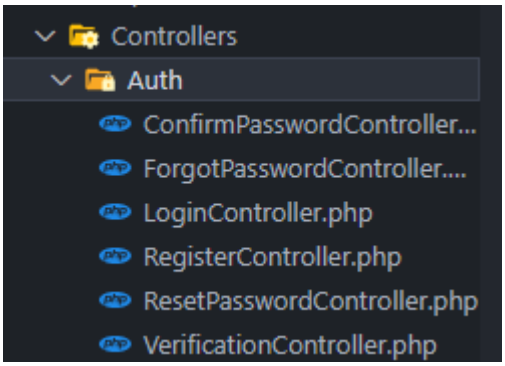



```php
Route::get('/', function () {
    return view('welcome');
});

Auth::routes();

Route::get('/home', [App\Http\Controllers\HomeController::class, 'index'])->name('home');
```

Podemos observar que se añade un apartado más con la ruta “/home” que utiliza el controlador “**`App\Http\Controllers\HomeController::class, 'index'`**”

```php
public function index()
{
    return view('home');
}
```

b) Controladores: podemos observar que se nos creo una carpeta “auth” y un controlador llamado “HomeController”



HomeController

```php
class HomeController extends Controller
{
    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('auth');
    }
```

podemos ver el uso de middlewares “auth” y una función redirigiendo a la vista home

```php
/**
 * Show the application dashboard.
 *
 * @return \Illuminate\Contracts\Support\Renderable
 */
public function index()
{
    return view('home');
}
```

Podemos Observar en el archivo Http/Kernel.php las diferentes rutas de los middlewares

```
 */
protected $routeMiddleware = [
    'auth' => \App\Http\Middleware\Authenticate::class,
    'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
    'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
    'can' => \Illuminate\Auth\Middleware\Authorize::class,
    'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
    'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,
    'signed' => \Illuminate\Routing\Middleware\ValidateSignature::class,
    'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
    'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
];
```

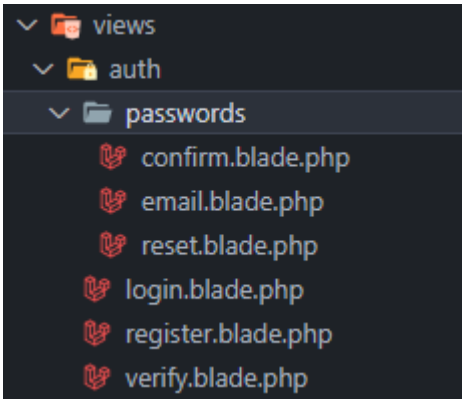
c) Middlewares:

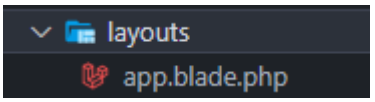
Los middleware proporcionan un mecanismo conveniente para filtrar solicitudes HTTP entrantes a tu aplicación

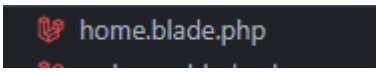
Laravel incluye un middleware que verifica si el usuario de tu aplicación está autenticado. Si el usuario no está autenticado, el middleware redireccionará al usuario a la pantalla de inicio de sesión

```
class Authenticate extends Middleware
{
    /**
     * Get the path the user should be redirected to when they are not authenticated.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return string|null
     */
    protected function redirectTo($request)
    {
        if (! $request->expectsJson()) {
            return route('login');
        }
    }
}
```

d) vistas: podemos observar la creación de diferentes archivos:







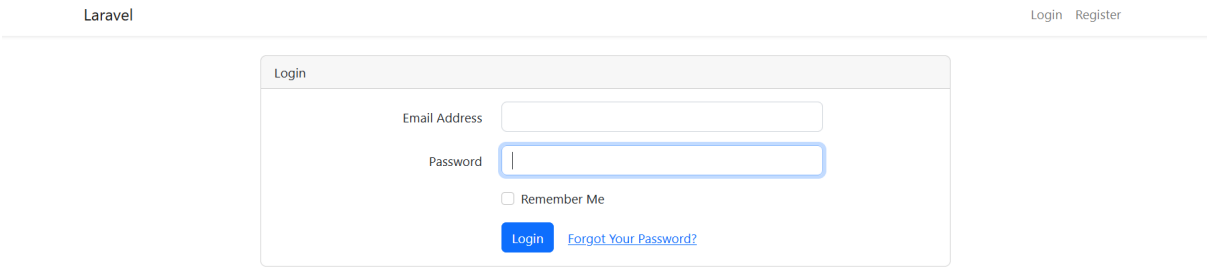
layouts/app.blade.php

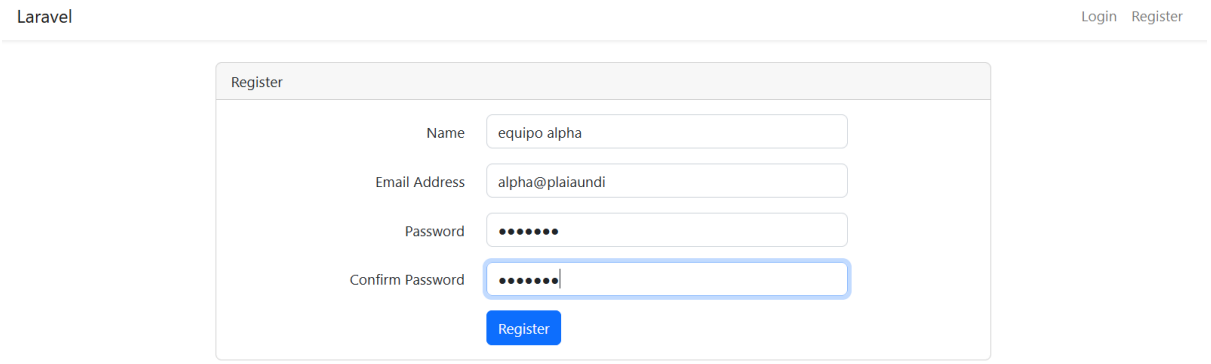
```
<!doctype html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- CSRF Token -->
    <meta name="csrf-token" content="{{ csrf_token() }}">

    <title>{{ config('app.name', 'Laravel') }}</title>

    <!-- Scripts -->
    <script src="{{ asset('js/app.js') }}" defer></script>

    <!-- Fonts -->
    <link rel="dns-prefetch" href="//fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css?family=Nunito" rel="stylesheet">

    <!-- Styles -->
    <link href="{{ asset('css/app.css') }}" rel="stylesheet">
</head>
<body>
    <div id="app">
        <nav class="navbar navbar-expand-md navbar-light bg-white shadow-sm">
            <div class="container">
                <a class="navbar-brand" href="{{ url('/') }}">
                    {{ config('app.name', 'Laravel') }}
                </a>
                <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded
                    <span class="navbar-toggler-icon"></span>
                </button>

                <div class="collapse navbar-collapse" id="navbarSupportedContent">
                    <!-- Left Side Of Navbar -->
                    <ul class="navbar-nav me-auto">

                    </ul>

                    <!-- Right Side Of Navbar -->
                    <ul class="navbar-nav ms-auto">
                        <!-- Authentication Links -->
                        @guest
                            @if (Route::has('login'))
                                <li class="nav-item">
                                    <a class="nav-link" href="{{ route('login') }}">{{ __('Login') }}</a>
                                </li>
                            @endif

                            @if (Route::has('register'))
                                <li class="nav-item">
                                    <a class="nav-link" href="{{ route('register') }}">{{ __('Register') }}</a>
```

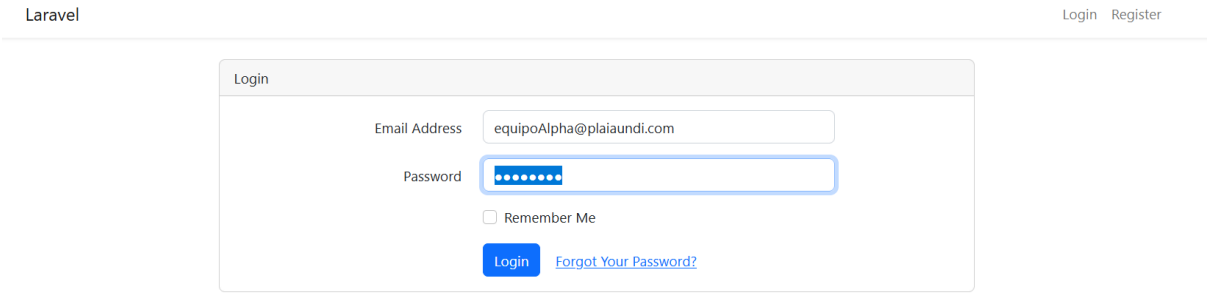
home.blade.php

```blade
@extends('layouts.app')

@section('content')
<div class="container">
    <div class="row justify-content-center">
        <div class="col-md-8">
            <div class="card">
                <div class="card-header">{{ __('Dashboard') }}</div>

                <div class="card-body">
                    @if (session('status'))
                        <div class="alert alert-success" role="alert">
                            {{ session('status') }}
                        </div>
                    @endif

                    {{ __('You are logged in!') }}
                </div>
            </div>
        </div>
    </div>
</div>
@endsection
```

5) verificamos el inicio de sesión y registro




Laravel
Login
Register
Register
Name
equipo alpha
Email Address
alpha@plaiaundi
Password
Confirm Password
Register


Laravel
Login
Register
Login
Email Address
equipoAlpha@plaiaundi.com
Password
Remember Me
Login
Forgot Your Password?


Laravel
jairo
Dashboard
You are logged in!

6) para verificar el usuario con el nick, agregamos en la tabla user un nuevo campo llamado “nick”

```php
$table->string('nick')->unique();
```

7) escribimos en la consola el comando “php artisan migrate:refresh” para que nos agregue los cambios en nuestra base de datos.

```
PS C:\Users\2daw3\proyectoLogin> php artisan migrate:refresh
Rolling back: 2019_12_14_000001_create_personal_access_tokens_table
Rolled back:  2019_12_14_000001_create_personal_access_tokens_table (31.57ms)
Rolling back: 2019_08_19_000000_create_failed_jobs_table
Rolled back:  2019_08_19_000000_create_failed_jobs_table (20.68ms)
Rolling back: 2014_10_12_100000_create_password_resets_table
Rolled back:  2014_10_12_100000_create_password_resets_table (25.21ms)
Rolling back: 2014_10_12_000000_create_users_table
Rolled back:  2014_10_12_000000_create_users_table (80.59ms)
Migrating: 2014_10_12_000000_create_users_table
Migrated:  2014_10_12_000000_create_users_table (80.26ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated:  2014_10_12_100000_create_password_resets_table (74.94ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated:  2019_08_19_000000_create_failed_jobs_table (79.78ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated:  2019_12_14_000001_create_personal_access_tokens_table (89.49ms)
```

8) nos dirigimos a models/user.php y agregamos el campo nick en $fillable

```php
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'email',
        'nick',
        'password',
    ];
```

9) Nos dirigimos a “app\Http\Controllers\Auth\LoginController.php” y agregamos la función username()

```php
    }
    public function username()
    {
        return 'nick';
    }
}
```

Este método devuelve el nombre de usuario de inicio de sesión que utilizará el controlador. Básicamente, indica: “Este es el campo de la base de datos que debe buscar como nombre de usuario cuando autentica un usuario”.

10) nos dirigimos a “app\Http\Controllers\Auth\RegisterController.php” y agregamos el campo “nick” en los apartados validator y create

```php
protected function validator(array $data)
{
    return Validator::make($data, [
        'name' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
        'nick' => ['required', 'string', 'unique:users'],
        'password' => ['required', 'string', 'min:8', 'confirmed'],
    ]);
}
```

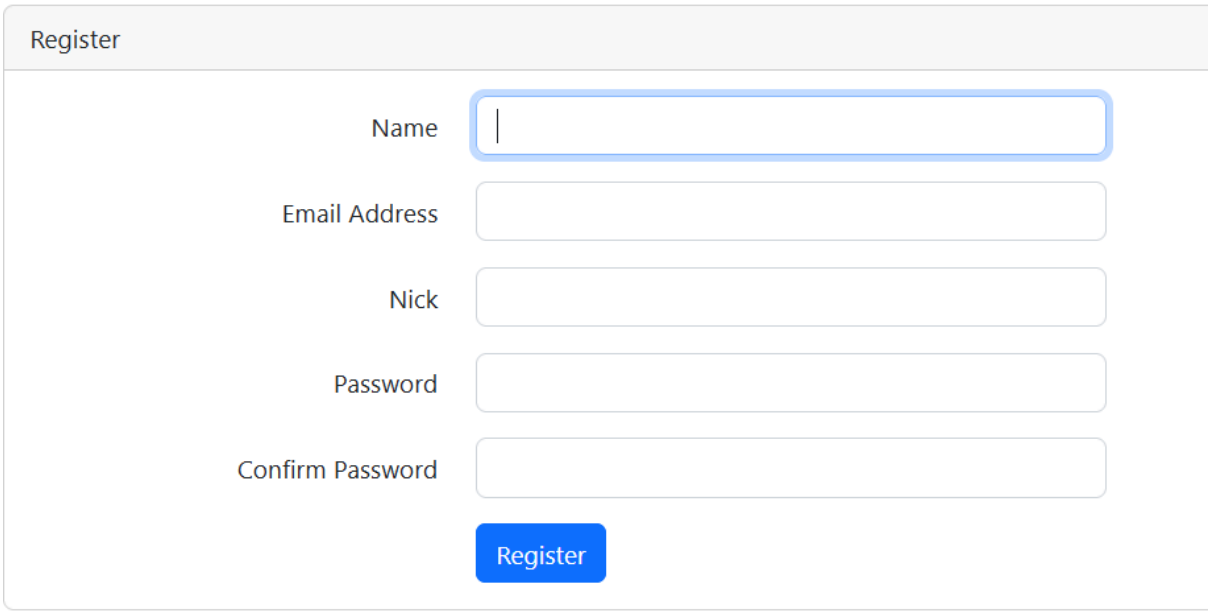
```php
protected function create(array $data)
{
    return User::create([
        'name' => $data['name'],
        'email' => $data['email'],
        'nick' => $data['nick'],
        'password' => Hash::make($data['password']),
    ]);
}
```

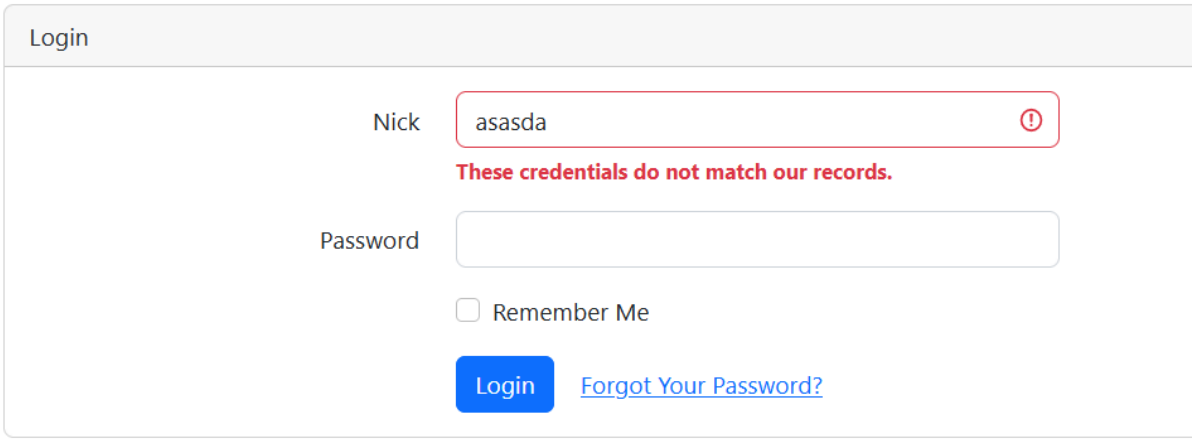
11) Modificamos nuestras vistas de autenticación agregando el campo nick en las rutas “resources\views\auth\login.blade.php” y “resources\views\auth\register.blade.php”

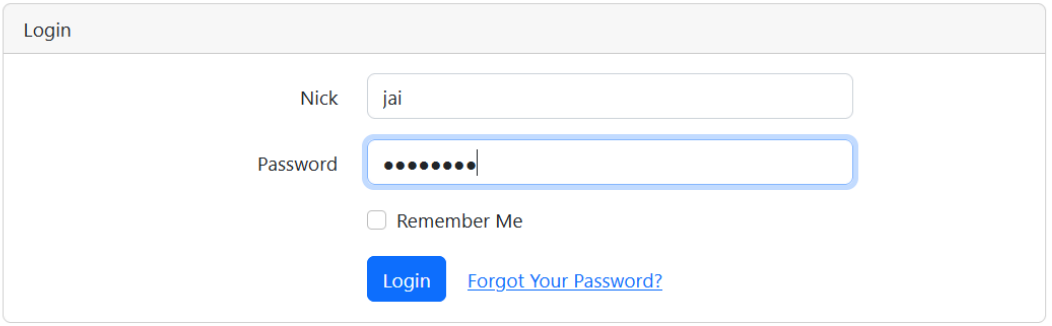
```
<label for="nick" class="col-md-4 col-form-label text-md-end">{{ __('Nick') }}</label>

<div class="col-md-6">
    <input id="nick" type="nick"
        class="form-control @error('nick') is-invalid @enderror" name="nick"
        value="{{ old('nick') }}" required autocomplete="nick">

    @error('nick')
        <span class="invalid-feedback" role="alert">
            <strong>{{ $message }}</strong>
        </span>
    @enderror
</div>
```

```
<label for="nick"
    class="col-md-4 col-form-label text-md-end">{{ __('Nick') }}</label>

<div class="col-md-6">
    <input id="nick" type="nick"
        class="form-control @error('nick') is-invalid @enderror" name="nick"
        value="{{ old('nick') }}" required autocomplete="nick">

    @error('nick')
        <span class="invalid-feedback" role="alert">
            <strong>{{ $message }}</strong>
        </span>
    @enderror
</div>
```

12) verificamos




Login
Nick
asasda
These credentials do not match our records.
Password
Remember Me
Login
Forgot Your Password?


Login
Nick
jai
Password
Remember Me
Login
Forgot Your Password?


Dashboard
You are logged in!