PROYECTO FINAL
DISEÑO DE APLICACIONES WEB

Daniel Buendia
Roberto Gómez

Índice de contenido

Planteamiento

Se pretende diseñar una base de datos que guarde toda la información relativa a una aplicación enfocada a reservas en distintas casas o apartamentos.

En dicha aplicación habrá clientes que serán “propietarios” de las casa y expondrán sus establecimientos para la reserva de ellos por parte de clientes que serán los “clientes finales”.

Para esto se gestionan usuarios y de ellos se desea guardar lo siguiente:

- Nombre, Apellidos, Correo electrónico, teléfono, etc... (datos comunes).
    - Si son propietarios se puede guardar su “rol” y además de eso, su dirección, cargo en el hotel, foto de perfil, etc...
    - Si son clientes finales se puede guardar su “rol” y además de eso, su dirección.
- También hay usuarios administradores.

Cada casa pertenece a una o varias categorías (ciudad, campo, playa, etc...).

Cada usuario puede ser propietario de uno o varias casas. De los casas se desea guardar lo siguiente:

- Nombre de la casas, propietario del misma, descripción corta, descripción larga, dirección o localización, ciudad, teléfonos de contacto, fotos, etc...
    - Una casa puede tener varios servicios y un servicio puede estar disponible en varias casas.
    - Las ciudades se deben guardar para poder realizar búsquedas por ciudad.
- Además de lo mencionado, es necesario guardar tarifas de la casa, disponibilidad, que servicios ofrece, historial de los clientes, comentarios que pueden hacer los clientes sobre la casa, etc...
    - Para el sistema de comentarios habrá que guardar quien lo ha hecho y a que hotel está dirigido, además de la fecha.
    - Una casa puede tener varios comentarios.

Paquetes instalados

Se muestra a continuación las librerías adicionales usadas para el desarrollo del proyecto.

- barryvdh/laravel-dompdf

Librería utilizada para la creación de PDF´S, dónde la utilizamos para descargas de facturas, manuales, etc...

- caouecs/laravel-lang

Esta librería se ha utilizado para crear un sitio multi-idioma donde en primera instancia dejamos disponible el Español y el Inglés para la página. También se queda preparado para añadir nuevos idiomas.

- irazasyed/telegram-bot-sdk

Este es un SDK no oficial de Telegram para el lenguaje PHP. Nos ha permitido desarrollar bots de telegram de manera sencilla para poder enviar mensajes de telegrama a los usuarios.

- paypal/rest-api-sdk-php

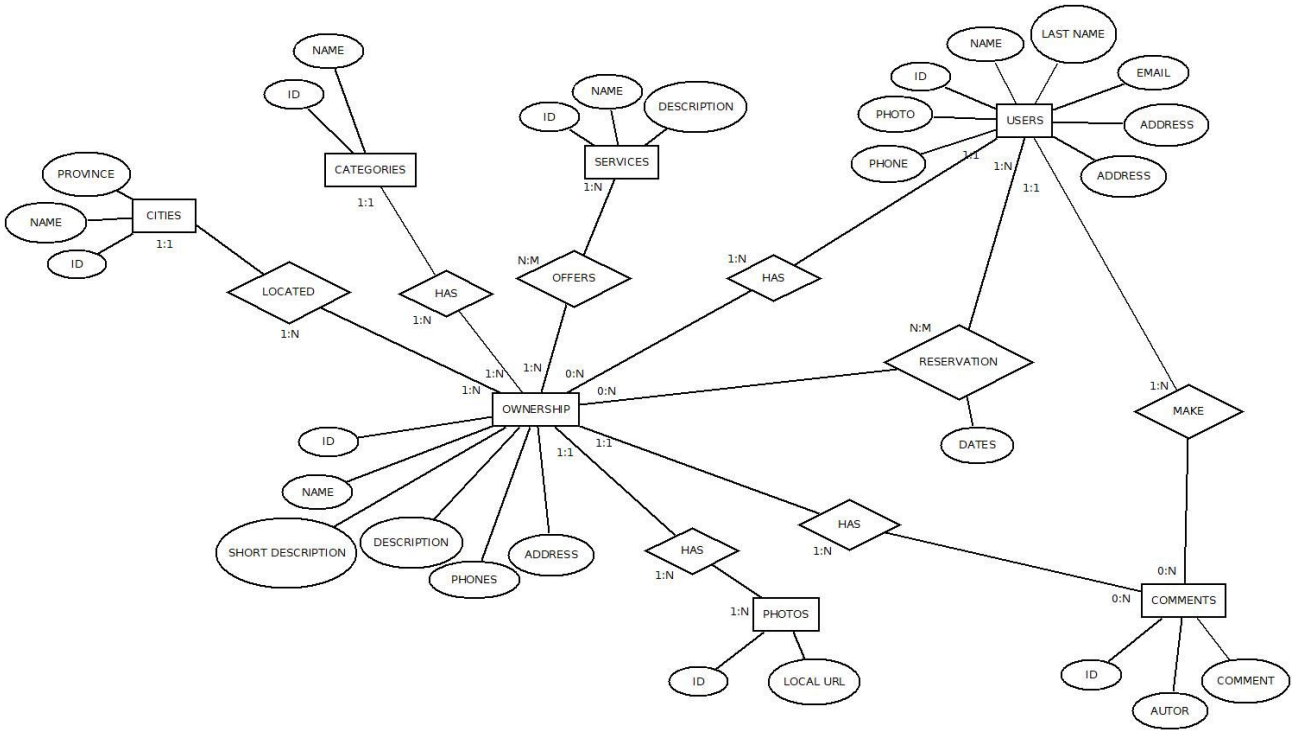
Librería que nos permite realizar pagos a través de la plataforma Paypal. Nos ha permitido que para los usuarios, de manera sencilla puedan realizar los pagos para las reservas con tan solo unos pocos click´s

- styde/html

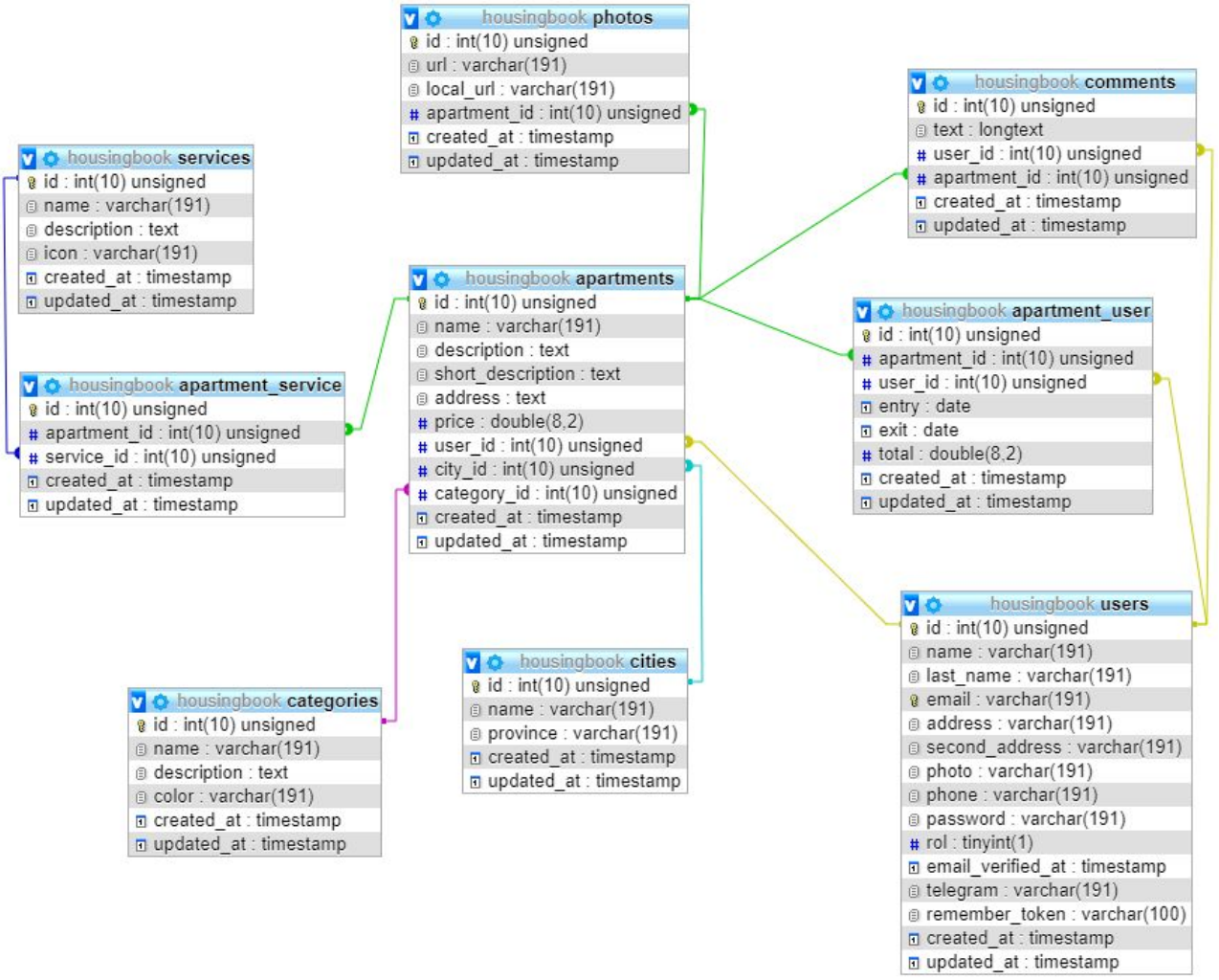
Esta librería nos ha permitido crear plantillas para los formularios agilizando un poco el trabajo.

Base de datos

Se adopta como primer pensamiento para la base de datos usando como base el planteamiento para la aplicación, modificando algunos aspectos.
Se crea un diagrama para empezar a trabajar a partir de aquí, pues no sabemos en principio con los problemas y necesidades que nos podemos encontrar:



Debido a las necesidades y problemas encontrados, durante el desarrollo de la aplicación se ha ido modificando la base de datos para adaptarla al cumplimiento de lo necesario, quedando hasta el momento de la siguiente manera:



- Migraciones

Se muestra a continuación las migraciones realizadas para las distintas tablas para la creación de la base de datos a partir de Laravel.

- Tabla usuarios.

Lo que podemos destacar en la migración para la tabla de la base de datos es la función "up" dónde añadimos los campos que nos han sido necesarios a parte de los que vienen por defecto.

- **Telegram:** Aquí guardamos el id de telegram del usuario para poder mandarle mensajes a través del bot.
- **email_verified_at:** Esta columna guardaremos si se ha confirmado el correo una vez registrado el usuario (si no se verifica habrá zonas a las que no podrá acceder).

```
public function up()
{
  Schema::create('users', function (Blueprint $table) {
     $table->increments('id');
     $table->string('name');
     $table->string('last_name');
     $table->string('email')->unique();
     $table->string('address');
     $table->string('second_address')->nullable();
     $table->string('photo')->nullable();
     $table->string('phone');
     $table->string('password');
     $table->boolean('rol')->default(0);
     $table->timestamp('email_verified_at')->nullable();
     $table->string('telegram')->nullable();
     $table->rememberToken();
     $table->timestamps();
  });
}
```

- Reinicio de contraseñas

Esta tabla se crea para que los usuarios puedan resetear su contraseña.

```
class CreatePasswordResetsTable extends Migration
{
  public function up()
  {
     Schema::create('password_resets', function (Blueprint $table) {
        $table->string('email')->index();
        $table->string('token');
        $table->timestamp('created_at')->nullable();
     });
  }

  public function down()
  {
     Schema::dropIfExists('password_resets');
  }
}
```

- Tabla categorías

Una tabla para guardar el nombre de las categorías.

- **color:** Hemos decidido guardar un color para cada categoría en hexadecimal para poder darle color cuando lo mostramos en las vistas en el momento que las mostramos.

```
public function up()
{
  Schema::create('categories', function (Blueprint $table) {
    $table->increments('id');
    $table->string('name');
    $table->text('description');
    $table->string('color');
    $table->timestamps();
  });
}
```

- Tabla servicios

En esta tabla guardaremos los servicios que ofrecen las casas.

- **icon:** Decidimos guardar el nombre del icono relacionado con el servicio para mostrar un icono cuando se vayan a mostrar los servicios que ofrece una casa.

```
public function up()
{
  Schema::create('services', function (Blueprint $table) {
    $table->increments('id');
    $table->string('name');
    $table->text('description');
    $table->string('icon');
    $table->timestamps();
  });
}
```

- Tabla ciudades

En esta tabla se guardan las ciudades para así poder hacer que una casa esté en una ciudad.

```
public function up()
{
  Schema::create('cities', function (Blueprint $table) {
    $table->increments('id');
    $table->string('name');
    $table->string('province');
    $table->timestamps();
  });
}
```

- Tabla apartamentos

Tabla apartamentos donde guardamos información general sobre los apartamentos y añadimos las claves foráneas para para las relaciones dónde un apartamento pertenece solo a un usuario, puede tener una categoría y está en una ciudad.

```
public function up()
{
  Schema::create('apartments', function (Blueprint $table) {
      $table->increments('id');
      $table->string('name');
      $table->text('description');
      $table->text('short_description');
      $table->text('address');
      $table->float('price');
      $table->unsignedInteger('user_id');
      $table->foreign('user_id')->references('id')->on('users')->onDelete('cascade')->onUpdate('cascade');
      $table->unsignedInteger('city_id');
      $table->foreign('city_id')->references('id')->on('cities')->onDelete('cascade')->onUpdate('cascade');
      $table->unsignedInteger('category_id');
      $table->foreign('category_id')->references('id')->on('categories')->onDelete('cascade')->onUpdate('cascade');
      $table->timestamps();
  });
}
```

- Tabla comentarios

En esta tabla se guardarán los comentarios hechos por los usuarios y a qué casa pertenece cada comentario.

```
public function up()
{
  Schema::create('comments', function (Blueprint $table) {
      $table->increments('id');
      $table->longText('text');
      $table->unsignedInteger('user_id');
      $table->foreign('user_id')->references('id')->on('users')->onDelete('cascade')->onUpdate('cascade');
      $table->unsignedInteger('apartment_id');
      $table->foreign('apartment_id')->references('id')->on('apartments')->onDelete('cascade')->onUpdate('cascade');
      $table->timestamps();
  });
}
```

- Tabla apartamento_usuario

Esta tabla, es una tabla pivote donde vamos a guardar los usuarios que alquilan una casa, que casa se alquila así como las fechas de entrada y salida y el precio total del alquiler:

```
public function up()
{
  Schema::create('apartment_user', function (Blueprint $table) {
    $table->increments('id');
    $table->unsignedInteger('apartment_id');
    $table->foreign('apartment_id')->references('id')->on('apartments')->onDelete('cascade')->onUpdate('cascade');
    $table->unsignedInteger('user_id');
    $table->foreign('user_id')->references('id')->on('users')->onDelete('cascade')->onUpdate('cascade');
    $table->date('entry');
    $table->date('exit');
    $table->float('total');
    $table->timestamps();
  });
}
```

- Tabla fotos

Se ha creado la siguiente tabla para guardar las rutas de las fotos de cada apartamento ya que vamos a guardar más de una foto.

```
public function up()
{
  Schema::create('photos', function (Blueprint $table) {
    $table->increments('id');
    $table->string('local_url');
    $table->unsignedInteger('apartment_id');
    $table->foreign('apartment_id')->references('id')->on('apartments')->onDelete('cascade')->onUpdate('cascade');
    $table->timestamps();
  });
}
```

- Tabla apartamento_servicio

Como las casas van a tener varios servicios, y los servicios pueden estar en más de una casa, se ha creado la siguiente tabla pivote para guardar lo necesario:

```
public function up()
{
  Schema::create('apartment_service', function (Blueprint $table) {
    $table->increments('id');
    $table->unsignedInteger('apartment_id');
    $table->foreign('apartment_id')->references('id')->on('apartments')->onDelete('cascade')->onUpdate('cascade');
    $table->unsignedInteger('service_id');
    $table->foreign('service_id')->references('id')->on('services')->onDelete('cascade')->onUpdate('cascade');
    $table->timestamps();
  });
}
```

- Seeders y factories

Para el desarrollo de la aplicación hemos trabajado con datos ficticios para así poder ver en funcionamiento la aplicación intentando simular los datos reales para realizar las distintas acciones que se pueden hacer en la página.

- Factories

Se han creado las siguientes factorías para indicar cómo van a ser los datos se van a crear cuando los seeder se ejecuten.

ApartmentFactory

```
$factory->define(App\Apartment::class, function (Faker $faker) {
  return [
     'name' => $faker->name,
     'description' => $faker->realText(),
     'short_description' => $faker->text,
     'address' => $faker->address,
     'price' => $faker->numberBetween(1, 500),
     'user_id' => \App\User::all()->random()->id,
     'city_id' => \App\City::all()->random()->id,
     'category_id' =>  \App\Category::all()->random()->id,
  ];
});
```

CategoryFactory

```
$factory->define(App\Category::class, function (Faker $faker) {
  return [
     'name' => $faker->unique()->word,
     'description' => $faker->realText(),
  ];
});
```

CityFactory

```
$factory->define(App\City::class, function (Faker $faker) {
  return [
     'name' => $faker->city,
     'province' => $faker->word,
  ];
});
```

CommentFactory

```
$factory->define(App\Comment::class, function (Faker $faker) {
  return [
    'text' => $faker->realText(),
    'user_id' => \App\User::all()->random()->id,
    'apartment_id' => \App\Apartment::all()->random()->id,
  ];
});
```

PhotoFactory

```
$factory->define(App\Photo::class, function (Faker $faker) {
  return [
    'url' => $faker->imageUrl(),
    'local_url' => \Faker\Provider\Image::image(storage_path() . '/app/public/photos', 600, 350, 'city', false),
    'apartment_id' => \App\Apartment::all()->random()->id,
  ];
});
```

ServiceFactory

```
$factory->define(App\Service::class, function (Faker $faker) {
  return [
    'name' => $faker->unique()->name,
    'description' => $faker->realText(),
  ];
});
```

UserFactory

```
$factory->define(App\User::class, function (Faker $faker) {
  return [
    'name' => $faker->name,
    'last_name' => $faker->unique()->userName,
    'email' => $faker->unique()->safeEmail,
    'address' => $faker->address,
    'second_address' => $faker->randomElement([$faker->address,null]),
    'photo' => $faker->imageUrl(),
    'phone' => $faker->phoneNumber,
    'email_verified_at' => now(),
    'password' => '$2y$10$TKh8H1.PfQx37YgCzwiKb.KjNyWgaHb9cbcoQgdIVFlYg7B77UdFm', // secret
    'remember_token' => str_random(10),
  ];
});
```

- Seeders

DatabaseSeeder

Aquí indicamos que seeders se van a ejecutar.

```
public function run()
{
   Storage::deleteDirectory('public/photos');
   Storage::makeDirectory('public/photos');
   $this->call(UserTableSeeder::class);
   $this->call(CategoryTableSeeder::class);
   $this->call(CityTableSeeder::class);
   $this->call(ServiceTableSeeder::class);
   $this->call(ApartmentTableSeeder::class);
   $this->call(PhotoTableSeeder::class);
   $this->call(CommentTableSeeder::class);
}
```

ApartmentTableSeeder

Este seeder se ha creado para rellenar la tabla apartamentos con 50 apartamentos. Se recalca a continuación lo que hace dicho seeder:

- Se crean 50 apartamentos.
- Por cada apartamento creado crea en la tabla “fotos” cinco fotos para cada apartamento además de lo siguiente:
    - Añade entre 1 y 6 servicios (de los que existen en la tabla “servicios”) en la tabla pivote “Apartment_services”,
    - En la tabla pivote “Apartment_user” añade la información necesaria para simular de que hay usuarios que han alquilado casas en un momento determinado.

```
public function run()
{
   factory(\App\Apartment::class,50)->create()->each(function (\App\Apartment $apartment){
      factory(\App\Photo::class, 5)->create(['apartment_id' => $apartment->id]);
      $services = \App\Service::all()->random(rand(1,6));
      $users = \App\User::all()->random(1);

      foreach ($services as $service)
      {
         $apartment->services()->attach($service->id);
      }

      foreach ($users as $user)
      {
         $entrada = \Carbon\Carbon::now()->subDays(rand(0,20));
         $salida = \Carbon\Carbon::now()->addDays(rand(2,5));
         $entradaDif = strtotime($entrada);
         $salidaDif = strtotime($salida);
         $datediff = $salidaDif - $entradaDif;
         $dif = ($datediff / (60 * 60 * 24));
         $apartment->users()->attach($user->id, ['total' => $dif * $apartment->price, 'entry' => $entrada,'exit' => $salida]);
      }
   });
}
```

CategoryTableSeeder

Este seeder rellena la tabla "categorías" con las categorías que hemos considerado y sus colores.

```
public function run()
{
  $names = [ 'Montaña', 'Ciudad', 'Campo', 'Playa' ];
  $colors = [ '#877b5a', '#ff7575', '#94dd89', '#70d8d3' ];

  foreach ($names as $key => $value){
    factory(\App\Category::class,1)->create([
      'name' => $value,
      'color' => $colors[$key],
    ]);
  }
}
```

CityTableSeeder

Este seeder creará varias ciudades con los datos especificado en el "factory".

```
public function run()
{
  factory(\App\City::class,20)->create();;
}
```

CommentTableSeeder

Este seeder creará 50 comentarios. En el factory se especifica a qué casa pertenece cada comentario.

```
public function run()
{
  factory(\App\Comment::class,50)->create();;
}
```

PhotoTableSeeder

Este seeder creará 20 fotos

```
public function run()
{
  //factory(\App\Photo::class,20)->create();; Comentado ya que la creación de las fotos se hará desde el ApartmentTableSeeder
}
```

ServiceTableSeeder

Este seeder creará los servicios que hemos pensado oportunos así como sus iconos en html

```
public function run()
{
  $icons = [ 'fa fa-car', 'fa fa-wifi', 'fa fa-swimming-pool', 'fas fa-smoking', 'fas fa-snowflake', 'fas fa-hamburger', ];
  $names = [ 'Parking',  'Wi-Fi', 'Piscina', 'Permitido Fumar', 'Aire acondicionado',  'Cocina' ];
  foreach($icons as $key => $value){
    factory(\App\Service::class,1)->create([
      'name' => $names[$key],
      'icon' => $value, ]);
 }
```

```
}
```

UserTableSeeder

Además de crear 10 usuarios creamos un usuario manualmente.

```
public function run()
{
  factory(\App\User::class,1)->create(['name' => 'housing',
     'last_name' => 'book',
     'email' => 'housingbook@gmail.com',
     'address' => 'C/ Esperanza',
     'phone' => '123456789',
     'rol' => 1,
     'telegram' => '12346789',
     'password' => bcrypt('123123Aa-')]);

  factory(\App\User::class,10)->create();
}
```

Rutas

Aquí se seccionará el archivo de rutas para poder explicarlo un poquito más claro sobre las rutas, grupos, middleware, etc.

- Las rutas que requieran de estar autenticado (middlewares se especifican en los controladores) se le ha indicado que el usuario que acceda deber de tener el correo verificado.

```
Auth::routes(['verify' => true]);
```

- Las siguientes rutas son las rutas a las que se pueden acceder sin estar autenticado, digamos la zona pública.

```
Route::get('/home', 'HomeController@index')->name('home');
Route::get('/apartments','ApartmentController@search')->name('search');
Route::get('/apartments/category/{Category}','ApartmentController@searchCategory')->name('searchCategory');
Route::post('/apartments/filter','ApartmentController@filter')->name('filterApartments');

Route::prefix('public')->group(function(){
  Route::get('apartments','PublicController@index')->name('apartments.public');
  Route::get('apartments/{apartment}','PublicController@show')->name('apartments.show');
});
```

- Ruta para acceder a los perfiles de los usuarios

```
Route::prefix('profile')->group(function () {
  Route::get('{name}','ProfileController@index')->name('profile.index');
});
```

- Las siguientes rutas se especifican para ver una casa en concreto y una vista con todas las casas.

```
Route::prefix('public')->group(function(){
  Route::get('apartments','PublicController@index')->name('apartments.public');
  Route::get('apartments/{apartment}','PublicController@show')->name('apartments.show');
});
```

- Digamos que los usuarios disponen de un “back-office” o “dashboard” dónde pueden modificar sus perfiles, crear apartamentos, modificarlos, borrarlos, etc..

```
Route::prefix('dashboard')->group(function(){
  Route::get('', 'DashboardController@index')->name('dashboard');
  Route::get('index', 'DashboardController@indexAjax')->name('dashboard.index');
  Route::get('user/show', 'dashboard\UserController@show')->name('user.show');
  Route::get('user/edit', 'dashboard\UserController@edit')->name('user.edit');
  Route::put('{user}/update', 'dashboard\UserController@update')->name('user.update');
  Route::get('user/telegram','dashboard\UserController@telegram')->name('user.telegram');
  Route::get('user/telegram/tutorial','dashboard\UserController@tutorial')->name('user.tutorial');
  Route::get('user/password','dashboard\UserController@password')->name('user.password');
  Route::put('{user}/telegram/update','dashboard\UserController@telegramUpdate')->name('user.telegramupdate');
  Route::put('{user}/password/update','dashboard\UserController@passwordUpdate')->name('user.passwordupdate');

  Route::prefix('apartment')->group(function() {
     Route::get('show/{apartment}','dashboard\ApartmentController@show')->name('apartment.showapartment');
     Route::get('index', 'dashboard\ApartmentController@index')->name('apartment.index');
     Route::get('create','dashboard\ApartmentController@create')->name('apartment.createapartment');
     Route::post('store','dashboard\ApartmentController@store')->name('apartment.storeapartment');
     Route::delete('delete/{apartment}','dashboard\ApartmentController@destroy')->name('apartment.deleteapartment');
     Route::get('{apartment}/edit','dashboard\ApartmentController@edit')->name('apartment.editapartment');
     Route::put('{apartment}/update','dashboard\ApartmentController@update')->name('apartment.updateapartment');
  });
```

- Se han creado también, rutas para poder mostrar facturas, descargarlas, etc.

```
Route::prefix('invoices')->group(function(){
  Route::get('index', 'dashboard\InvoiceController@index')->name('invoice.index');
  Route::get('invoices', 'dashboard\InvoiceController@invoices')->name('invoice.invoices');
  Route::post('download', 'dashboard\InvoiceController@download')->name('invoice.download');
});
```

- Ruta para interactuar con el controlador de telgram y así poder envíar mensajes a los usuarios.

```
Route::get('/activity', 'TelegramController@updatedActivity');
```

- Ruta para poder crear el sitio multi-idioma

```
Route::get('set_language/{lang}', 'Controller@setLanguage')->name('set_language');
```

- Las siguientes rutas son las necesarias para realizar el proceso de alquilar una casa. Se comprueba que está libre y nos lleva para realizar el pago del alquiler.

```
Route::prefix('rent')->group(function(){
  Route::get('store/{apartment}','RentController@store')->name('rent.apartment');
  Route::post('confirm/{apartment}','RentController@validation')->name('apartment.validate');
})
Route::post('paypal/confirm', 'RentController@confirm')->name('paypal-confirm');
Route::post('paypal/pay', 'PaymentController@payPaypal')->name('paypal-pay');
Route::get('paypal/status', 'PaymentController@getStatus')->name('paypal-status');
```

Modelos

Se muestra a continuación los modelos creados (hemos creado un modelo para cada tabla pero mostramos los más significativos) para poder realizar toda la lógica necesaria para poder obtener información de la base de datos de las relaciones, reglas de validación, guardado en masa, etc.

- Apartment

Lo primero que podemos destacar en este modelo es la asignación de datos en masa

```
protected $fillable = ['name','description','address','short_description','city_id','user_id', 'price'];
```

Las siguientes funciones nos sirven para obtener los datos de las relaciones de las tablas. Con las siguientes opciones podemos obtener:

- La ciudad a la que pertenece cada casa.
- A qué usuario pertenece.
- Las fotos de cada casa.
- Los servicio que ofrece cada casa.
- La categoría a la que pertenece.
- Los comentarios de cada casa.
-

```php
public function city()
{
  return $this->belongsTo(City::class);
}

public function user()
{
  return $this->belongsTo(User::class);
}

public function photos()
{
  return $this->hasMany(Photo::class);
}

public function services()
{
  return $this->belongsToMany(Service::class);
}

public function category()
{
  return $this->belongsTo(Category::class);
}

public function users()
{
  return $this->belongsToMany(User::class);
}

public function getPhotoImageAttribute()
{
  return 'storage/photos/' . $this->photos()->first()->local_url;
}

public function comments()
{
  return $this->hasMany(Comment::class);
}
```

Las siguientes funciones nos sirven también para obtener los datos de las relaciones de las tablas pero en este caso, se hace el uso de las tablas pivote utilizando la función proporcionada por laravel "withPivot" y especificando las columnas que queremos. Con las siguientes funciones podemos obtener lo siguiente:

- La disponibilidad de una casa a partir de una fecha de entrada y salida.
- Fechas en las que una casa está reservada
- Ganancias que tiene un usuario con una casa

```
public function checkDisponibility($entrada,$salida)
{
  return $this->belongsToMany(User::class)->withPivot('entry',
'exit')->whereBetween('entry',[$entrada,$salida])->whereBetween('exit',[$entrada,$salida]);
}

public function getDatesRented($id)
{
  return $this->belongsToMany(User::class)->withPivot('entry',
'exit')->wherePivot('user_id',$id)->latest('apartment_user.created_at')->first();
}

public function noAvailableDates()
{
  return $this->belongsToMany(User::class)->withPivot('entry','exit')->get(['entry','exit']);
}

public function getReservedDates()
{
  return $this->belongsToMany(User::class)->withPivot('entry','exit')->get(['entry','exit','name']);
}

public function getMoneyGained()
{
  return $this->belongsToMany(User::class)->withPivot('total')->get(['total']);
}
```

- Category

Con la función añadida podemos ver las casas de cada apartamento.

```
public function apartments()
{
  return $this->hasMany(Apartment::class);
}
```

- Comment

Modelo donde especificamos la función necesaria para ver a qué usuario pertenece un usuario.

```
public function user()
{
  return $this->belongsTo(User::class);
}
```

- User

Especificamos la asignación de datos en masa.

```
protected $fillable = [
  'name','last_name', 'email', 'password','address','second_address','photo','phone','telegram'
];
```

Las siguientes funciones dónde podemos ver los apartamentos de cada usuario y los comentarios que ha realizado.

```
public function apartments()
{
  return $this->hasMany(Apartment::class);
}

public function comments()
{
  return $this->hasMany(Comment::class);
}
```

En las siguientes funciones se hace el uso de la función “withPivot” para poder acceder a la información de las tablas pivote de la base de datos y así obtener lo siguiente:

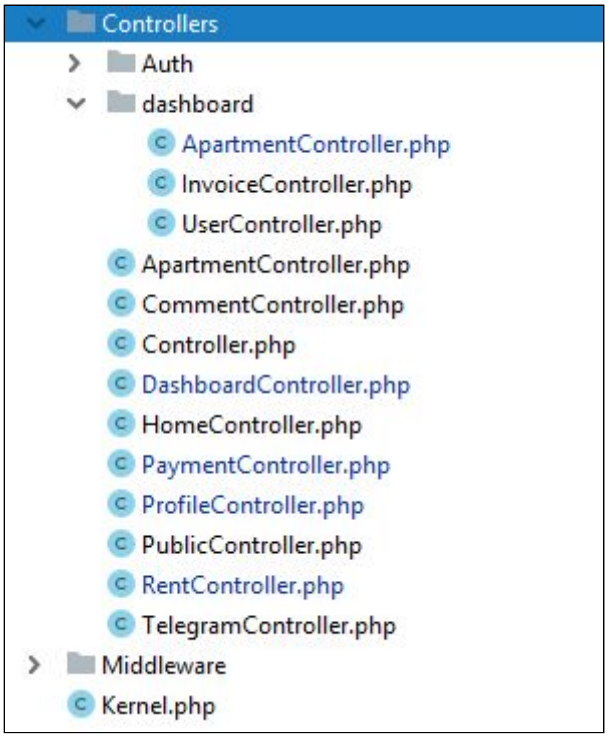
- En qué casas ha estado el usuario.
- Las facturas de de un usuario.
- Una factura en concreto de un usuario

```
public function apartmentsUser()
{
  return $this->belongsToMany(Apartment::class)->withPivot('apartment_id','user_id','exit');
}
public function invoices()
{
  return $this->belongsToMany(Apartment::class)->withPivot('id', 'entry', 'exit', 'total');
}
public function invoice($id, $user_id)
{
  return $this->belongsToMany(Apartment::class)->withPivot('id', 'entry', 'exit', 'total', 'created_at')->wherePivot('id',
$id)->wherePivot('user_id', $user_id);
}
```

Controladores

A continuación en la captura se puede apreciar la estructura de controladores que mostraremos después.

Se ha creado una carpeta “dashboard” para guardar los controladores que se sitúan en el “back-office” donde los usuarios pueden crear, modificar, borrar sus casas y modificar sus perfiles. Los demás controladores son los encargados de mostrar un poco la parte más pública de la página, realizar los alquileres, ver las casas, etc.



Se mostrarán a continuación los controlades que hemos creado, y controladores que vienen por defecto y han sido modificados.

- Controladores en Auth

- RegistrerController

En este controlador hemos guardado las validaciones para el registro de los usuarios.

```
class RegisterController extends Controller
{
  use RegistersUsers;
  protected $redirectTo = '/';

  public function __construct()
  {
    $this->middleware('guest');
  }

  protected function validator(array $data)
  {
    return Validator::make($data, [
      'name' => ['required', 'string', 'max:255', 'regex:/^[a-zA-Z ]*$/'],
      'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
      'password' => ['required', 'string', 'min:8', 'confirmed'],
      'lastname' => ['required','string', 'regex:/^[a-zA-Z ]*$/'],
      'address' => ['required','string'],
      'phone' => ['required','string', 'regex:/^[9|6|7|8][0-9]{8}$/'],
      'conditions' => ['required']
    ]);
  }

  protected function create(array $data)
  {
    return User::create([
      'name' => $data['name'],
      'email' => $data['email'],
      'password' => Hash::make($data['password']),
      'last_name' => $data['lastname'],
      'address' => $data['address'],
      'phone' => $data['phone'],
      'photo' => '/img/profile-foto.jpg'
    ]);
  }
}
```

- VerificationController

Controlador creado para poder enviar correos a los usuarios cuando se registran y así poder verificarlos.

```
use App\Http\Controllers\Controller;
use Illuminate\Foundation\Auth\VerifiesEmails;

class VerificationController extends Controller
{
  use VerifiesEmails;
  protected $redirectTo = '/';

  public function __construct()
  {
    $this->middleware('auth');
    $this->middleware('signed')->only('verify');
    $this->middleware('throttle:6,1')->only('verify', 'resend');
  }
}
```

- Controladores en dashboard

- ApartmentController

En este controlador es dónde se dispone de un CRUD para los apartamentos.

Este controlador requiere del uso de las siguientes clases y librerías.

```
use App\Apartment;
use App\Category;
use App\City;
use App\Helpers\Helper;
use App\Photo;
use App\Service;
use Illuminate\Contracts\Validation\Rule;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Validator;
```

En el constructor indicamos que middleware se interpone en el controlador.

```
public function __construct()
{
  return $this->middleware(['auth','verified']);
}
```

La siguiente función es la que nos va a mostrar todas las casas (se utiliza ajax).

```
public function index(Request $request)
{
  if($request->ajax()){
     $apartments = Apartment::where('user_id',auth()->user()->id)->paginate(6);
     $view = view('dashboard.apartment.index',compact('apartments'))->render();
     return response()->json(['html'=>$view]);
  } else {
     return back();
  }
}
```

La función que nos muestra una casa en concreto(se utiliza ajax).

```
public function show(Request $request,Apartment $apartment)
{
  if($request->ajax()){

     $apartment->load('city','user','photos','services');
     $view = view('dashboard.apartment.show',compact('apartment'))->render();

     return response()->json(['html'=>$view]);

  } else {
     return back();
  }
}
```

La siguiente función es la que nos carga la vista con el formulario para crear una casa.

```
public function create(Request $request)
{
  $cities = City::all();
  $services = Service::all();
  $categories = Category::all();
  return view('dashboard.apartment.create',compact('cities','services','categories'));
}
```

Esta función es la que valida si los campos de formulario para crear son válidos, y en el caso de que lo sea creará una nueva casa.

```
public function store(ApartmentRequest $request)
{
  if($request->file('photos') != null){
     if(count($request->file('photos')) != 4){
        session()->flash('error', 'apartments.photos_bad');
        return back();
     }else{
        foreach($request->file('photos') as $photo){
           if(Helper::validateFile($photo) == false){
              session()->flash('error', 'apartments.photos_extension');
              return back();
           }
        }
     }

  }else{
     session()->flash('error', 'apartments.photos_bad');
     return back();
  }

  $apartment = new Apartment();

  $request->merge(['city_id' => $request->city]);
  $apartment->fill($request->all());
  $apartment->category_id = $request->category;
  $apartment->user_id = auth()->user()->id;
  $apartment->save();

  foreach($request->file('photos') as $photo){
     $picture = Helper::uploadFile($photo);
     $photo = new Photo();
     $photo->local_url = $picture;
     $photo->apartment_id = $apartment->id;
     $photo->save();
  }

  Helper::servicesTableFill($apartment,$request->services);
  return back()->with('success', __('apartments.create'));
}
```

La siguiente función es la encargada de devolvernos el formulario para editar una casa.

```
public function edit(Request $request,Apartment $apartment)
{
   if($apartment->user->id == auth()->user()->id) {

       $cities = City::all();
       $services = Service::all();
       $categories = Category::all();

       $apartmentServices = $apartment->services()->get();

       return view('dashboard.apartment.edit', compact('apartment', 'cities', 'services', 'categories','apartmentServices'));

   } else {
       return back();
   }
}
```

Esta es la función encargada de validar lo recibido del formulario de actualización y en el caso de que sea todo correcto actualizará la casa.

```
public function update(ApartmentRequest $request, Apartment $apartment)
{
  if($apartment->user->id == auth()->user()->id){

    if($request->file('photos') != null){
      if(count($request->file('photos')) != 4){
        session()->flash('error', 'apartments.photos_bad');
        return back();
      }else{
        foreach($request->file('photos') as $photo){
          if(Helper::validateFile($photo) == false){
            session()->flash('error', 'apartments.photos_extension');
            return back();
          }
        }
      }
    }

    $request->merge(['city_id' => $request->city]);
    $apartment->fill($request->all());
    $apartment->user_id = auth()->user()->id;
    $apartment->save();

    if($request->file('photos') != null){
      foreach($request->file('photos') as $photo){
        $picture = Helper::uploadFile($photo);

        $photo = new Photo();
        $photo->local_url = $picture;
        $photo->apartment_id = $apartment->id;
        $photo->save();
      }
    }

    Helper::servicesTableFill($apartment, $request->services);
    return back()->with('success', __('apartments.create'));
  } else {
    return back();
  }
}
```

La siguiente función sirve para borrar las casas que indique el usuario.

```
public function destroy(ApartmentRequest $request)
{
  $aparmentId = Apartment::find($request->idHolder);

  if($aparmentId->user->id == auth()->user()->id) {

    foreach ($aparmentId->photos as $photo){
      if (\File::exists(storage_path('app/public/photos/' . $photo->local_url))) {
        \File::delete(storage_path('app/public/photos/' . $photo->local_url));
      }
    }

    $aparmentId->delete();

    return redirect(route('dashboard'))->with('success', __('profile.deletesuccess'));
  } else {
    return back();
  }
}
```

Con el siguiente método el usuario también puede borrar solo las fotos de una casa.

```
public function photodestroy(ApartmentRequest $request)
{
  $aparmentId = Apartment::find($request->idHolder);

  if($aparmentId->user->id == auth()->user()->id) {

    foreach ($aparmentId->photos as $photo){
      if (\File::exists(storage_path('app/public/photos/' . $photo->local_url))) {
        \File::delete(storage_path('app/public/photos/' . $photo->local_url));
      }
      $photo->delete();
    }

    return back()->with('success', __('profile.deletephotosuccess'));
  } else {
    return back()->with('error', __('profile.deleteapartment'));
  }
}
```

- InvoiceController

Controlador creado para poder mostrar las facturas de los alquileres de los usuarios, descargarlas, etc.

Este controlador requiere del uso de las siguientes clases y librerías.

```
use Barryvdh\DomPDF\Facade;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\App;
```

En el constructor indicamos que middleware se interpone en el controlador.

```
public function __construct()
{
  return $this->middleware(['auth','verified']);
}
```

En la siguiente función se obtienen las facturas del usuario autenticado (para ello se utiliza la función “invoices” declarada en el modelo de “user”)

```
public function index(Request $request)
{
  if($request->ajax()){

    $apartments = Auth()->user()->invoices;
    $view =  view('dashboard.invoices.index', compact('apartments'))->render();
    return response()->json(['html'=>$view]);

  } else {
    return back();
  }
}
```

- UserController

Controlador creado para que los usuarios en el “back-office” puedan modificar sus perfiles, cambiar contraseñas, etc.

Este controlador requiere del uso de las siguientes clases y librerías.

```
use App\Helpers\Helper;
use App\Http\Controllers\Controller;
use App\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\App;
use Illuminate\Validation\Rule;
use Barryvdh\DomPDF\Facade;
```

En el constructor indicamos que middleware se interpone en el controlador.

```
public function __construct()
{
   return $this->middleware(['auth','verified']);
}
```

La siguiente función nos carga una vista dónde se mostrará toda la información del usuario autenticado (utiliza petición ajax).

```
public function show(Request $request,User $user)
{
   if($request->ajax()){
      $view = view('dashboard.user.show', compact('user'))->render();

      return response()->json(['html'=>$view]);

   } else {
      return back();
   }
}
```

La función “edit” nos cargará la vista con el formulario para que un usuario autenticado pueda editar su perfil.

```
public function edit(Request $request)
{
   return view('dashboard.user.edit')->render();
}
```

Esta función actualiza un usuario en el caso de que todas las validaciones sean correctas.

```
public function update(User $user, Request $request)
{
  if($user->id == auth()->user()->id) {

    if($request->file('photo'))
    {
      $photo = Helper::uploadFile($request->file('photo'));
      $user->photo = $photo;
    }

    $user->fill($request->all());
    $user->save();

    return back()->with('success',__('profile.profileupdatedocorrectly'));
  } else {
    return redirect('/dashboard/'. auth()->user()->id . '/edit');
  }
}
```

La siguiente función carga una vista dónde se muestra el “id” de telegram del usuario autenticado en el caso de que lo tenga y si no tiene un formulario para actualizar su “id” de telegram.

```
public function telegram(Request $request)
{
  return view('dashboard.user.telegram');
}
```

Esta función actualiza el “id” de telegram que usuario introduzca en el formulario mencionado.

```
public function telegramUpdate(User $user, Request $request)
{
  if($user->id == auth()->user()->id)
  {
    $user->fill($request->all());
    $user->save();

    return back()->with('success',__('profile.telegramupdatedcorrectly'));
  } else {
    return redirect('/dashboard/'. auth()->user()->id . '/telegram');
  }
}
```

Las dos funciones siguientes:

- La primera carga la vista con un formulario para actualizar la contraseña-
- La segunda Actualiza la contraseña en el caso de que las validaciones sean correctas.

```
public function password(Request $request)
{
   if($request->ajax()){
      $view = view('dashboard.user.password')->render();

      return response()->json(['html'=>$view]);

   } else {
      return back();
   }
}


public function passwordUpdate(User $user, Request $request)
{
   if($user->id == auth()->user()->id)
   {
      $user->password = bcrypt($request->password);
      $user->save();

      return back()->with('success',__('profile.passwordupdatedcorrectly'));
   } else {
      return redirect('/dashboard/'. auth()->user()->id . '/password');
   }
}
```

La siguiente función hace el uso de “dompdf” para cargar un pdf online donde se muestra un tutorial.

```
public function tutorial()
{
   $pdf = App::make('dompdf.wrapper');
   $pdf = Facade::loadView('pdf.tutorial');
   return $pdf->stream();
}
```

- Controladores generales

- ApartmentController

Este controlador requiere del uso de las siguientes clases y librerías.

```
use App\Apartment;
use App\Category;
use App\City;
use App\Service;
use Illuminate\Http\Request;
```

El middleware que se interponer en el controlador es el siguiente.

```
public function __construct()
{
  return $this->middleware(['auth','verified'])->except(['search','searchCategory','filter']);
}
```

La siguiente función nos cargará los apartamentos que pertenezcan a una ciudad introducida en un formulario

```
public function search(Request $request)
{
  $categories = Category::all();
  $max = Apartment::max('price');
  $min = Apartment::min('price');
  $apartments = [];
  $cities = City::where('name', 'LIKE', '%'.$request->search . '%')->get();
  $latest_apartment = Apartment::orderBy('id', 'DESC')->take(3)->get();
  $services = Service::all();

  $ids = [];

  if(count($cities))
  {
    foreach($cities as $city)
    {
      $ids[] = $city->id;
    }
  }
    $apartments = Apartment::with('city','user','photos','services')->whereIn('city_id',$ids)->get();

  return view('guest.index',compact('apartments','latest_apartment', 'categories', 'min', 'max','services'));
}
```

Esta función es similar a la anterior pero nos buscará las casas a partir de una categoría seleccionada.

```
public function searchCategory($id)
{
  $categories = Category::all();
  $max = Apartment::max('price');
  $min = Apartment::min('price');
  $latest_apartment = Apartment::orderBy('id', 'DESC')->take(3)->get();
  $services = Service::all();

  $category = Category::find($id);

  $apartments = $category->apartments()->get();
  return view('guest.index',compact('apartments','latest_apartment', 'categories', 'min', 'max','services'));
}
```

Con la siguiente función atendemos a la solicitud de un formularios donde se eligen unos filtros de búsqueda. Nos devuelven los apartamentos en función de lo seleccionado en el formulario.

```
public function filter(Request $request)
{
  $categories = Category::all();
  $max = Apartment::max('price');
  $min = Apartment::min('price');
  $latest_apartment = Apartment::orderBy('id', 'DESC')->take(3)->get();
  $services = Service::all();
  $request->validate([
     "service"    => "array",
     "service.*"  => "numeric | exists:services,id",
     "category"    => "array",
     "category.*"  => "numeric | exists:categories,id",
     'range' => "required"
  ],["service.*" => __('form.servicenotvalid'), "category.*" => __('form.categorynotvalid'),'range.required' => __('form.range')]);

  $ids = [];
  $ids2 = [];
  $apartments = [];

  if($request->category != null && $request->service !=null)
  {
     foreach($request->category as $item)
     {
        $ids[] = $item;
     }

     foreach($request->service as $item2)
     {
        $ids2[] = $item2;
     }

     $apartments = Apartment::where('price','<',$request->range)->whereIn('category_id',$ids)->get();
     $apartmentsServices = [];
     foreach($apartments as $apartment)
     {

        $servicesApartment = $apartment->services()->get();
        $contador = 0;
```

```
foreach($servicesApartment as $item)
        {
            if(in_array($item->id,$ids2))
                $contador++;
        }

        if($contador == count($ids2))
            $apartmentsServices[] = $apartment;
    }

    $apartments = $apartmentsServices;
  }

  if($request->category != null && $request->service == null)
  {
    foreach($request->category as $item)
    {
        $ids[] = $item;
    }
    $apartments = Apartment::where('price','<',$request->range)->whereIn('category_id',$ids)->get();
  }

  if($request->category == null && $request->service != null)
  {
    foreach($request->service as $item)
    {
        $ids[] = $item;
    }
    $apartments = Apartment::where('price','<',$request->range)->get();

    $apartmentsServices = [];

    foreach($apartments as $apartment)
    {
        /*if($apartment->services()->wherePivotIn('service_id',$ids2)->exists())
            $apartmentsServices[] = $apartment;*/
        $servicesApartment = $apartment->services()->get();

        $contador = 0;

        foreach($servicesApartment as $item)
        {
            if(in_array($item->id,$ids))
                $contador++;
        }

        if($contador == count($ids))
            $apartmentsServices[] = $apartment;
    }

    $apartments = $apartmentsServices;
  }

  if($request->category == null && $request->service == null)
  {
    $apartments = Apartment::where('price','<',$request->range)->get();
  }

  return view('guest.index',compact('apartments','latest_apartment', 'categories', 'min', 'max','services'));
}
```

- CommentController

Este controlador nos sirve para almacenar los comentarios que realiza un usuario.

Este controlador requiere del uso de las siguientes clases y librerías.

```
use App\Apartment;
use App\Comment;
use Illuminate\Http\Request;
```

En la siguiente función “call” se especifica el middleware que se interpone en el controlador.

```
public function __call($method, $parameters)
{
   return $this->middleware(['auth','verified']);
}
```

La siguiente función del controlador es la que almacena los comentarios que hace un usuario (almacenará a que casa pertenece el comentario y qué usuario lo hace).

```
public function store(Request $request,Apartment $apartment)
{
   $comment = new Comment();

   $comment->text = $request->comment;
   $comment->apartment_id = $apartment->id;
   $comment->user_id = auth()->user()->id;

   $comment->save();

   return back()->with('success','Comentario creado con exito');
}
```

- DashboardController

Controlador creado para que cargue la vista del “dashboard” o “back-office” de los usuarios.

El middleware que se interponer en el controlador es el siguiente.

```
public function __construct()
{
   return $this->middleware(['auth','verified']);
}
```

La siguiente función es la que nos carga la vista “index” del “back-office”.

```
public function indexAjax()
{
  $totalEarnings = 0;

  $apartments = auth()->user()->apartments;

  foreach($apartments as $apartment)
  {
    $rents = $apartment->getMoneyGained();

    foreach ($rents as $rent){
      $totalEarnings += $rent->total;
    }
  }

  $user = auth()->user();

  $allDates = [];

  foreach($user->apartments as $apartment)
  {
    $dates = $apartment->getReservedDates();

    foreach ($dates as $date)
    {
      $date->last_name = $apartment->name;
      $allDates[] = $date;
    }
  }

  $view = view("dashboard.indexAjax",compact('totalEarnings','allDates'))->render();

  return response()->json(['html'=>$view]);
}
```

- PaymentController

Este controlador se ha creado para poder implementar la realización de pagos mediante la plataforma PayPal (En el proceso de realizar un pago también se utiliza el controlador RentController). Podemos recalcar dos partes en este controlador, la función que envía la solicitud para realizar el pago y la función que atiende la solicitud devuelta al hacer el pago.

Este controlador requiere del uso de las siguientes clases y librerías.

```
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Input;
use PayPal\Rest\ApiContext;
use PayPal\Auth\OAuthTokenCredential;
use PayPal\Api\Amount;
use PayPal\Api\Details;
use PayPal\Api\Item;
use PayPal\Api\ItemList;
use PayPal\Api\Payer;
use PayPal\Api\Payment;
use PayPal\Api\RedirectUrls;
use PayPal\Api\PaymentExecution;
use PayPal\Api\Transaction;
use PayPal\Exception\PayPalConnectionException;
use Illuminate\Support\Facades\Session;
```

En el constructor de la clase se establece la configuración necesaria para poder conectar con la API de PayPal.

```
private $_api_context;
public function __construct()
{
   $payal_conf = \Config::get('paypal');
   $this->_api_context = new ApiContext(new OAuthTokenCredential(
      $payal_conf['client_id'],
      $payal_conf['secret']
   ));
   $this->_api_context->setConfig($payal_conf['settings']);
}
```

El siguiente controlador es el que realiza la solicitud a la API para realizar el pago. En primera instancia recibía los datos de un formulario (precio, cantidad de días, etc.), pero se ha optado por recibir esa información de variables de sesión declaradas en el primer paso del proceso de alquilar una casa (se ha validado la información y llegamos aquí).

```
public function payPaypal(Request $request)
{
  $payer = new Payer();
  $payer->setPaymentMethod("paypal"); //Se selecciona el método de pago
  $apartment = session('apartmentReservado'); //Almacenamos el apartamento que viene en una variable de sesión

  $item1 = new Item(); // Se crea un nuevo ítem. Como es el alquiler de una casa solo creamos un apartamento
  $item1->setName($apartment->name)
    ->setCurrency('EUR') //Se establece la moneda de pago
    ->setQuantity(1) // La cantidad
    ->setPrice($apartment->price * session('days')); // Se establece el precio. El precio por la cantidad de días almacenado en sesión

  $itemList = new ItemList(); //Se crea una lista con el total
  $itemList->setItems(array($item1));
  $amount = new Amount();
  $amount->setCurrency("EUR") ->setTotal($apartment->price * session('days'));

  $transaction = new Transaction();
  $transaction->setAmount($amount)
    ->setItemList($itemList)
    ->setDescription("Payment description");

  $redirect_urls = new RedirectUrls(); // Se establecen las Url de redirección
  $redirect_urls->setReturnUrl(url('paypal/status'))->setCancelUrl(url('paypal-status'));

  $payment = new Payment(); //Se establece los datos para realizar el pago
  $payment->setIntent('Sale')
    ->setPayer($payer)
    ->setRedirectUrls($redirect_urls)
    ->setTransactions(array($transaction));

  try { //Aquí es
    $payment->create($this->_api_context);
  } catch (\PayPal\Exception\PPConnectionException $ex) {
    if (\Config::get('app.debug')) {
      session()->put('error', 'Connection timeout');
      return redirect('public/apartments');
    } else {
      session()->put('error', 'Some error occur, sorry for inconvenient');
      return redirect('public/apartments');
    }
  }
  foreach ($payment->getLinks() as $link) {
    if ($link->getRel() == 'approval_url') {
      $redirect_url = $link->getHref();
      break;
    }
  }
  session()->put('paypal_payment_id', $payment->getId());
  if (isset($redirect_url)) {
    return redirect($redirect_url); // Aquí ya nos redirige a la página de PayPal para realizar el pago
  }
  session()->put('error', 'Unknown error occurred');
  return redirect('public/apartments');
}
```

La siguiente función atiende la solicitud que se recibe de la página de PayPal después de realizar el pago correctamente o no.

```
public function getStatus(Request $request)
{
  $apartment = session('apartmentReservado');

  $pay_id = session('paypal_payment_id');

  if (empty($request->PayerID) || empty($request->token)) {
     session()->put('error', 'Payment failed');
     return redirect('public/apartments');
  }

  $payment = Payment::get($pay_id, $this->_api_context);

  $execution = new PaymentExecution();
  $execution->setPayerId($request->PayerID);

  $result = $payment->execute($execution, $this->_api_context);

  session()->forget('paypal_payment_id');
  if ($result->getState() == 'approved') {
     return redirect('rent/store/' . $apartment->id);
  }else{
     session()->put('error', 'Payment failed');
     return redirect('public/apartments');
  }

}
```

- ProfileController

Este controlador se ha creado para mostrar el perfil de un usuario y los apartamentos que puede tener el mismo.

Se muestra la función que carga la vista del usuario con sus apartamentos y el middleware que se interpone en el controlador.

```
public function __construct()
{
  return $this->middleware(['auth','verified']);
}

public function index(Request $request)
{
  $user = User::where('name', '=', $request->name)->firstOrFail();
  $nums = 1;
  $nums2 = 0;
  return view('profile.index', compact('user', 'nums', 'nums2'));
}
```

- PublicController

Este controlador nos muestra la parte pública de la página (parte en la que no es necesario estar autenticado) donde se muestran las casas con toda su información.

En este controlador se han usado las siguientes clases.

```
use App\Apartment;
use App\Category;
use App\Service;
```

La siguiente función nos carga una vista con todas las casas disponibles además de las últimas añadidas.

```
public function index()
{
  $apartments = Apartment::with('city','user','photos')->paginate(6);
  $latest_apartment = Apartment::orderBy('id', 'DESC')->take(3)->get();
  $categories = Category::all();
  $max = Apartment::max('price');
  $min = Apartment::min('price');
  $services = Service::all();

  return view('guest.index',compact('apartments', 'latest_apartment', 'categories', 'min', 'max','services'));
}
```

Esta función nos muestra una casa en concreto. Cabe destacar que se realizan consultas para obtener las fechas en los que la casa está reservada para no permitir marcar esas fechas en el calendario. Se muestra una fecha u otra en función del idioma seleccionado en la página

```
public function show(Apartment $apartment)
{
  $apartment->load('city','user','photos','services');
  $randoms_apartments = Apartment::where('id','!=',$apartment->id)->where('city_id',$apartment->city_id)->paginate(4);

  $dates = $apartment->noAvailableDates();

  $allDates = [];

  foreach ($dates as $date)
  {
    $date_from = strtotime($date->entry);
    $date_to = strtotime($date->exit);

    for ($i=$date_from; $i<=$date_to; $i+=86400) {
      $allDates[] = config('app.locale') == 'es' ? date("d-m-Y", $i) : date("Y-m-d", $i);
    }
  }


  return view('guest.show',compact('apartment','randoms_apartments','allDates'));
}
```

La siguiente función del controlador nos carga una página de bienvenida.

```
public function welcome()
{
   $categories = Category::all();
   return view('welcome', compact('categories'));
}
```

- RentController

Este controlador es el primero que interfiere en el proceso de pago, valida si las fechas y los datos provenientes de una casa son correcto y nos carga la vista para realizar el siguiente paso en el proceso de pago (primer método de "PaymentController" comentado anteriormente).

Para este controlador se usan las siguientes clases.

```
use App\Apartment;
use App\Mail\NewRent;
use App\Mail\SuccessfullyRent;
use App\User;
use Carbon\Carbon;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Mail;
```

En el constructor del controlador se especifica el middleware que interfiere en el controlador.

```
public function __construct()
{
   return $this->middleware(['auth','verified']);
}
```

La función “validation” recibe los datos de un formulario, también recibe la casa que se va a alquilar.

- Se cambia el formato de las fechas para poder realizar las consultas.
- Se calcula la diferencia entre una fecha y otra.
- Se comprueba la disponibilidad con la función “checkDisponibility” creada en el modelo “Apartment”.
- En el caso de que la consulta devuelve resultados quiere decir que no está disponible en las fechas solicitadas.
- En el caso de que las validaciones sean correctas, se crean las variables de sesión para iniciar el proceso de pago (se carga una vista para confirmar el pago).

```
public function validation(Request $request, Apartment $apartment)
{
  $request->entrada = str_replace('/','-',$request->entrada);
  $request->salida = str_replace('/','-',$request->salida);

  $date = new \DateTime($request->entrada);
  $date2 = new \DateTime($request->salida);

  $days = $date->diff($date2)->format('%a');

  $entrada = $date->format('Y-m-d');
  $salida = $date2->format('Y-m-d');

  $entradaCheck = Carbon::parse($entrada);
  $salidaCheck = Carbon::parse($salida);

  if($entradaCheck > $salidaCheck || $days == 0){
     return back()->with('error', __('apartments.rent_error'));
  }

  $checkDisponibility = $apartment->checkDisponibility($entrada,$salida);

  if($checkDisponibility->count()){
     return back()->with('error', __('apartments.rent_error_disponibility'));
  }else{
     session([
        'entradaEstancia' => $entrada,
        'salidaEstancia' => $salida,
        'apartmentReservado' => $apartment,
        'UserReservado' => auth()->user()->id,
        'days' => $days,
        ]);

     return view('paypal.demo', compact('apartment', 'entrada', 'salida', 'days'));
  }

}
```

En el caso de que el proceso de pago se realice correctamente se llama a la ruta que apunta hacia este método, que almacena en "apartment_user" el usuario que alquila, apartamento alquilado, fechas, precio total, etc.

- Podemos destacar que aquí, Realizamos envío de correos (al usuario que alquila y al dueño de la casa).
- También se envía un mensaje de telegram al usuario que alquila y al dueño de la casa en el caso de que tenga añadida una "id" de telegram.

```php
public function store(Apartment $apartment)
{
  $apartmentSession = session('apartmentReservado');
  $entrada = session('entradaEstancia');
  $salida = session('salidaEstancia');
  $price = session('days') * $apartment->price;

  if($apartment->id != $apartmentSession->id)
  {
     session(['error' => 'Unknown error occurred']);
     return redirect('public/apartments');

  }else{

     $apartment->users()->attach(auth()->user()->id, ['entry' => $entrada, 'exit' => $salida, 'total' => $price], "asdada");

     //Email para dueño
     $owner = User::where('id', $apartment->user_id)->first();

     Mail::to($owner)->send(new NewRent(auth()->user(), $apartment));

     //Email para persona que alquila
     $user = auth()->user();
     Mail::to($user)->send(new SuccessfullyRent(auth()->user(), $apartment));

     //Telegram para dueño y persona que alquila
     TelegramController::sendTelegrams(auth()->user(),$owner->telegram,$apartment);

     session()->forget('apartmentReservado');

     return redirect('public/apartments')->with('success', __('apartments.rent_success'));

  }
}
```

- TelegramController

Este controlador es el encargado de enviar mensajes de telegram a los usuarios. Se hace la llamada la función de este controlador desde “RentController”.

Se hace uso de las siguientes clases.

```
use App\Apartment;
use App\User;
use Auth;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Config;
use Telegram\Bot\Laravel\Facades\Telegram;
use Telegram\Bot\Methods\Chat;
```

La siguiente función es la que envía los mensajes de telegram.

- Se le da formato a las fechas en función del idioma seleccionado en la página.
- En el caso de que el usuario que alquila y el dueño de la casa tengan guardada la “id” de telegram se les enviará un mensaje.

```
public static function sendTelegrams($user,$ownerTelegram,Apartment $apartment)
{
  try {

    $dates = $apartment->getDatesRented($user->id);
    $entrada = Config::get('app.locale') == 'es' ? date("d/m/Y", strtotime($dates->pivot->entry)) : $dates->pivot->entry;
    $salida = Config::get('app.locale') == 'es' ? date("d/m/Y", strtotime($dates->pivot->exit)) : $dates->pivot->exit;

    if($ownerTelegram != null){
       Telegram::sendMessage([
          'chat_id' => $ownerTelegram,
          'text' => __('apartments.your') . $apartment->name . __('apartments.rented') . $entrada . __('apartments.to') . $salida
       ]);
    }

    if($user->telegram != null)
    {
       Telegram::sendMessage([
          'chat_id' => $user->telegram,
          'text' => __('apartments.useryour') . $apartment->name . __('apartments.userrented') . $entrada . __('apartments.userto')
. $salida
       ]);
    }
  } catch (\Exception $exception){

  }
  return;
}
```

Sitio multi-idioma

Se ha configurado el proyecto para crear un sitio multi-idioma, en inglés y en español.

Para ello se requiere la instalación de la librería “caouecs” mencionada en los paquetes instalados.
Con la librería instalada para configurarlo podemos hacer lo siguiente.

Se añade la función “setLanguage” en “Controller”.

```
public function setLanguage(string $lang)
{
  if(array_key_exists($lang, config('language'))){
     session()->put('applocale', $lang);
  }
  return back();
}
```

En el archivo “Kernel.php” se añade la siguiente línea en “middlewaregroups”.

```
\App\Http\Middleware\Language::class,
```

Se crea el middleware “language” para atender las solicitudes.

```
use Carbon\Carbon;
use Closure;
class Language
{
  public function handle($request, Closure $next)
  {
     if (session('applocale')) {
        $configLang = config('language')[session('applocale')];
        setlocale(LC_TIME, $configLang[1] . '.utf8');
        Carbon::setLocale(session('applocale'));
        \App::setLocale(session('applocale'));
     } else {
        session()->put('applocale', config('app.fallback_locale'));
        setlocale(LC_TIME, 'es_ES.utf8');
        Carbon::setLocale(session('applocale'));
        \App::setLocale(config('app.fallback_locale'));
     }
     return $next($request);
  }
}
```

hemos creado el archivo “language.php” en el directorio “config” con lo siguiente.

```php
<?php
return [
   'en' => ['English', 'en_US'],
   'es' => ['Espanish', 'es_ES'],
];
```

Se crea la siguiente ruta para que el usuario cambie el idioma.

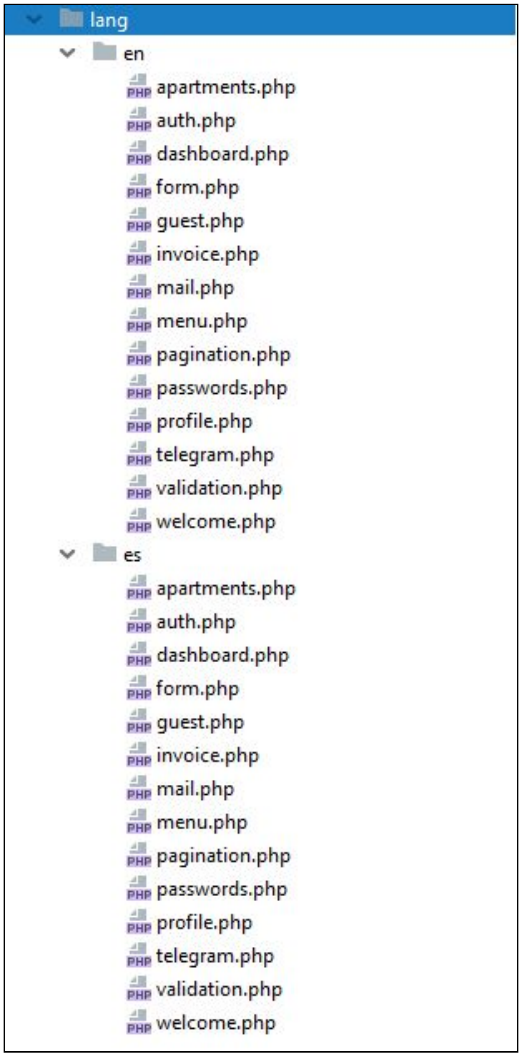
```php
Route::get('set_language/{lang}', 'Controller@setLanguage')->name('set_language');
```

Con esto solo queda añadir los textos para que se traduzcan en función del idioma seleccionado haciendo referencia a ellos en las vistas html.

```
<h2 class="text-center title">{{__(apartment.latest')}}
```

```
'latest' => 'Last apartments added',
```

En “resources\lang” se guardan los textos para las traducciones.



Uso de Styde

Se hace uso de la librería Styde para la creación de formularios (login, registro, creación de casas, etc.).

Para ello se ha creado un “Provider” llamado “FormServiceProvider” donde especificamos todos los campos que vamos a utilizar en los formularios.

```
public function boot()
{
   Form::component('text','components.form.text', ['name', 'value' => null, 'attributes' => []]);
   Form::component('textArea','components.form.textarea', ['name', 'value' => null, 'attributes' => []]);
   Form::component('number','components.form.number', ['name', 'value' => null, 'attributes' => []]);
   Form::component('date','components.form.date', ['name', 'value' => null, 'attributes' => []]);
   Form::component('email','components.form.email', ['name', 'value' => null, 'attributes' => []]);
   Form::component('password','components.form.password', ['name', 'attributes' => []]);
   Form::component('radio','components.form.radio', ['name', 'value' => null, 'attributes' => []]);
   Form::component('checkbox','components.form.checkbox', ['name', 'value' => null, 'attributes' => []]);
   Form::component('file', 'components.form.file', ['name', 'attributes' => []]);
   Form::component('hidden', 'components.form.hidden', ['name', 'value'=>null, 'attributes' => []]);
   Form::component('submit', 'components.form.submit', ['value' => 'Submit', 'attributes' => []]);
   Form::component('reset', 'components.form.reset', ['value' => 'Reset', 'attributes' => []]);
}
```

Para usar cada componente creado en el “Provider” se ha creado una plantilla por cada uno.

- **checkbox:**

```
<div class="form-group">
   {{ Form::label($name, null, ['class' => 'control-label']) }}
   {{ Form::checkbox($name, $value, array_merge(['class' => 'form-control'], $attributes)) }}
</div>
```

- **date:**

```
<div class="form-group">
   {{ Form::label($name, null, ['class' => 'control-label']) }}
   {{ Form::date($name, $value, array_merge(['class' => 'form-control'], $attributes)) }}
</div>
```

- **email:**

```
<div class="form-group">
   {{ Form::label($name, null, ['class' => 'control-label']) }}
   {{ Form::email($name, $value, array_merge(['class' => 'form-control'], $attributes)) }}
</div>
```

- **file:**

```
<div>
   {{ Form::file($name, array_merge(['class' => 'form-control'], $attributes) }}
</div>
```

- **hidden:**

```
{{ Form::hidden($name, $value, $attributes) }}
```

- **number:**

```
<div class="form-group">
  {{ Form::label($name, null, ['class' => 'control-label']) }}
  {{ Form::number($name, $value, array_merge(['class' => 'form-control'], $attributes)) }}
</div>
```

- **password:**

```
<div class="form-group">
  {{ Form::label($name, null, ['class' => 'control-label']) }}
  {{ Form::password($name, array_merge(['class' => 'form-control'], $attributes)) }}
</div>
```

- **radio:**

```
<div class="form-group">
  {{ Form::label($name, null, ['class' => 'control-label']) }}
  {{ Form::radio($name, $value, array_merge(['class' => 'form-control'], $attributes)) }}
</div>
```

- **reset:**

```
<div>
  {{ Form::reset($value, $attributes) }}
</div>
```

- **submit:**

```
<div>
  {{ Form::submit($value, $attributes) }}
</div>
```

- **text:**

```
<div class="form-group">
  {{ Form::label($name, null, ['class' => 'control-label']) }}
  {{ Form::text($name, $value, array_merge(['class' => 'form-control'], $attributes)) }}
</div>
```

- **textarea:**

```
<div class="form-group">
  {{ Form::label($name, null, ['class' => 'control-label']) }}
  {{ Form::textarea($name, $value, array_merge(['class' => 'form-control'], $attributes)) }}
</div>
```

Helpers

Se ha creado un “Helper”para trasladar parte de lógica de los controladores de manera que pueda quedar código un poco más claro.

- Helper

Aquí definimos las siguientes funciones.

“UploadFile” nos sirve para procesar los ficheros recibidos en formularios y subirlos a un directorio que tenemos definido.

```
public static function uploadFile($photo)
 {
    $filenameWithExt = $photo->getClientOriginalName();
    $filename = pathinfo($filenameWithExt, PATHINFO_FILENAME);
    $extension = $photo->getClientOriginalExtension();
    $filenameToStore = $filename.'_'.time().'.'.$extension;
    $photo->storeAs('public/photos/', $filenameToStore);

    return $filenameToStore;
 }
```

La siguiente función nos ayuda a almacenar en la tabla pivot de categorias:

```
public static function categoriesTableFill($apartment, $objects)
{
  foreach($objects as $object)
  {
     $apartment->categories()->attach($object);
  }
}
```

La función “servicesTableFill” nos sirve para almacenar los servicios que tendrá una casa en la tabla pivot.

```
public static function servicesTableFill($apartment, $objects)
{
  foreach($objects as $object)
  {
     $apartment->services()->attach($object);
  }
}
```

La siguiente función nos ayuda a validar los ficheros que sube el usuario ya que hemos experimentado algunos problemas en las validaciones proporcionadas por laravel.

```
public static function validateFile($photo)
{
   $extension = $photo->getClientOriginalExtension();

   if($extension == 'png'){
      return true;
   }else if($extension == 'jpg'){
      return true;
   }else if($extension == 'jpeg'){
      return true;
   }else{
      return false;
   }
}
```

Request

Se han creado “request” para llevarnos las validaciones de los controladores y los modelos hacía estas clases.

- ApartmentRequest

En la siguiente función se declaran las reglas de validación para las casas.

```
public function rules()
{
  switch($this->method()) {
     case 'GET':
     case 'DELETE':
        return [
           'idHolder' => 'numeric'
        ];
     case 'POST':
        return [
           'name' => 'required | min:3 | unique:apartments,name',
           'description' => 'required | min:3 | max:300',
           'address' => 'required | min:10 | max:100',
           'short_description' => 'required | min:3 | max: 100',
           'city' => 'required | exists:cities,id',
           'services' => 'required',
           'category' => 'required',
           'price' => 'required',
        ];
     case 'PUT':
        return [
           'name' => ['required','min:3',Rule::unique('apartments','name')->ignore($this->apartment->id)],
           'description' => 'required | min:3 | max:300',
           'address' => 'required | min:10 | max:100',
           'short_description' => 'required | min:3 | max: 100',
           'city' => 'required | exists:cities,id',
           'services' => 'required',
           'category' => 'required',
           'price' => 'required'
        ];
     case 'PATCH':
     default:break;
  }
}
```

- PasswordRequest

En la siguiente función se declaran las reglas de validación para las contraseñas.

```
public function rules()
{
  switch($this->method()) {
     case 'GET':
     case 'DELETE':
     case 'POST':
     case 'PUT':
        return ['password' => 'required | string | min:8 | confirmed'];
     case 'PATCH':
     default:break;
  }
}
```

- TelegramRequest

En la siguiente función se declaran las reglas de validación para el telegram.

```
public function rules()
{
  switch($this->method()) {
     case 'GET':
     case 'DELETE':
     case 'POST':
     case 'PUT':
        return [telegram => 'required | numeric'];
     case 'PATCH':
     default:break;
  }
}
```

- UserRequest

En la siguiente función se declaran las reglas de validación para los usuarios.

```
public function rules()
{
  switch($this->method()) {
     case 'GET':
     case 'DELETE':
     case 'POST':
     case 'PUT':
        return [ 'name' => 'required | min:3 | regex:/^[a-zA-Z]*$/',
           'last_name' => 'required | min:3 | max:300 | regex:/^[a-zA-Záéíóú ]*$',
           'email' => ['required','min:10','max:100',Rule::unique('users','email')->ignore($this->user->id)],
           'address' => 'required',
           'phone' => ['required' , 'regex:/^[9|6|7|8][0-9]{8}$/']
         ];
     case 'PATCH':
     default:break;
  }
}
```

Vistas

A continuación se muestran y comentan las vistas creadas y utilizadas para el proyecto. Se debe mencionar que para el proyecto se han utilizado dos plantillas de creative-tim.

- Plantillas

Se han utilizado las siguientes plantillas:

- Now UI Kit PRO
- Paper Dashboard 2

- Layouts

Podemos mostrar primeramente los layouts de las vistas para poder tener más o menos una idea de cómo se distribuyen las vistas ya que utilizamos dos plantillas diferentes. Una plantilla se utiliza para para la parte principal donde podemos ver los apartamentos, hacer reservas, etc.. y la otra plantilla es como un panel de control donde los usuarios podrán crear casas, modificarse los perfiles, etc.

- Página principal

Para la página principal se han creado dos layouts, podemos decir que uno utiliza una barra de navegación y el otro no.

app.blade.php

De esta vista a continuación indicamos la parte html.

```
<meta content='width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0, shrink-to-fit=no' name='viewport'/>
  <!--     Fonts and icons     -->
  <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700,200|Open+Sans+Condensed:700" rel="stylesheet">
  <link href="https://use.fontawesome.com/releases/v5.0.6/css/all.css" rel="stylesheet">
  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.0/css/all.css"
integrity="sha384-lZN37f5QGtY3VHgisS14W3ExzMWZxybE1SJSEsQp9S+oqd12jhcu+A56Ebc1zFSJ" crossorigin="anonymous">
  <!-- CSS Files -->
  <link href="{{asset('css/bootstrap.min.css')}}" rel="stylesheet" />
  <link href="{{asset('css/now-ui-kit.css?v=1.2.2')}}" rel="stylesheet" />
  <!-- CSS Just for demo purpose, don't include it in your project -->
  <link href="{{asset('demo/demo.css')}}" rel="stylesheet" />
  <link rel="stylesheet" href="{{asset('dat/bootstrap-datepicker3.min.css')}}">
</head>
<body class="presentation-page">
<noscript>
  <iframe src="https://www.googletagmanager.com/ns.html?id=GTM-NKDMSK6" height="0" width="0"
style="display:none;visibility:hidden"></iframe>
</noscript>
<main >
  @yield('content')
</main>
</body>
</html>
```

Para este layout se han utilizado los siguientes script.

```
<script type="text/javascript">
  $(document).ready(function() {
    if ($(window).width() >= 991) {

      setTimeout(function() {
        var rellax = new Rellax('.rellax', {
          center: true
        });
      }, 5000);

      var rellaxHeader = new Rellax('.rellax-header');
      var rellaxText = new Rellax('.rellax-text');
    }

  });

</script>
```

```
<script>
  (function(w, d, s, l, i) {
    w[l] = w[l] || [];
    w[l].push({
      'gtm.start': new Date().getTime(),
      event: 'gtm.js'
    });
    var f = d.getElementsByTagName(s)[0],
      j = d.createElement(s),
      dl = l != 'dataLayer' ? '&l=' + l : '';
    j.async = true;
    j.src =
      'https://www.googletagmanager.com/gtm.js?id=' + i + dl;
    f.parentNode.insertBefore(j, f);
  })(window, document, 'script', 'dataLayer', 'GTM-NKDMSK6');
</script>
```

Se utilizan también los siguientes scripts.

```
<!--   Core JS Files   -->
<script src="{{ asset('js/core/jquery.min.js')}}" type="text/javascript"></script>
<script src="{{ asset('js/core/popper.min.js')}}" type="text/javascript"></script>
<script src="{{ asset('js/core/bootstrap.min.js')}}" type="text/javascript"></script>
<!--  Plugin for Switches, full documentation here: http://www.jque.re/plugins/version3/bootstrap.switch/ -->
<script src="{{ asset('js/plugins/bootstrap-switch.js')}}"></script>
<!--  Plugin for the Sliders, full documentation here: http://refreshless.com/nouislider/ -->
<script src="{{ asset('js/plugins/nouislider.min.js')}}" type="text/javascript"></script>
<!--  Plugin for the DatePicker, full documentation here: https://github.com/uxsolutions/bootstrap-datepicker -->
<script src="{{ asset('js/plugins/moment.min.js')}}"></script>
<!--   Plugin for Tags, full documentation here: https://github.com/bootstrap-tagsinput/bootstrap-tagsinputs  -->
<script src="{{ asset('js/plugins/bootstrap-tagsinput.js')}}"></script>
<!--   Plugin for Select, full documentation here: http://silviomoreto.github.io/bootstrap-select -->
<script src="{{ asset('js/plugins/bootstrap-selectpicker.js')}}" type="text/javascript"></script>
<!--  Plugin for the DateTimePicker, full documentation here: https://eonasdan.github.io/bootstrap-datetimepicker/ -->
<script src="{{ asset('js/plugins/bootstrap-datetimepicker.js')}}" type="text/javascript"></script>
<!--  Google Maps Plugin    -->
<script src="https://maps.googleapis.com/maps/api/js?key=YOUR_KEY_HERE"></script>
<!-- Control Center for Now Ui Kit: parallax effects, scripts for the example pages etc -->
<script src="{{ asset('js/now-ui-kit.js?v=1.2.2')}}" type="text/javascript"></script>
<!-- Library for parallax, used just in Presentation Page -->
<script src="{{ asset('js/plugins/presentation-page/rellax.min.js')}}" type="text/javascript"></script>
```

app-nav.blade.php

Esta vista es idéntica a la anterior pero en esta se incluye un “navbar”.

```
@include('partials.navbar')
```

El navbar incluido es el siguiente.

```
<nav class="navbar navbar-expand-lg bg-primary fixed-top navbar-transparent" color-on-scroll="500">
  <div class="container">
    <div class="navbar-translate">
      <a class="navbar-brand" href="{{url('./')}}">
        HousingBook
      </a>
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navigation"
aria-controls="navigation-index" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-bar top-bar"></span>
        <span class="navbar-toggler-bar middle-bar"></span>
        <span class="navbar-toggler-bar bottom-bar"></span>
      </button>
    </div>
    <div class="collapse navbar-collapse" data-nav-image="./assets/img//blurred-image-1.jpg" data-color="orange">
      <ul class="navbar-nav ml-auto">

        <li class="nav-item">
          <a href="{{route('apartments.public')}}" class="nav-link">
            <i class="now-ui-icons design_image" aria-hidden="true"></i>
            <p>{{__('menu.apartment')}}</p>
          </a>
        </li>

        @guest
          <li class="nav-item">
            <a class="nav-link btn btn-primary" href="{{ route('login') }}">{{ __('menu.login') }}</a>
          </li>
          @if (Route::has('register'))
            <li class="nav-item">
              <a class="nav-link btn btn-primary" href="{{ route('register') }}">{{ __('menu.register') }}</a>
            </li>
          @endif
        @else
          <li class="nav-item">
            <a href="{{route('profile.reservations')}}" class="nav-link">
              <i class="now-ui-icons education_agenda-bookmark" aria-hidden="true"></i>
              <p>{{__('menu.reservations')}}</p>
            </a>
          </li>
          <li class="nav-item dropdown">

            <a id="navbarDropdown" class="nav-link dropdown-toggle" href="#" role="button" data-toggle="dropdown"
aria-haspopup="true" aria-expanded="false" v-pre>
              {{ Auth::user()->name }} <span class="caret"></span>
            </a>

            <div class="dropdown-menu dropdown-menu-right" aria-labelledby="navbarDropdown">


              <a href="{{route('dashboard')}}" class="dropdown-item">
                <i class="now-ui-icons design_bullet-list-67" aria-hidden="true"></i>
                <p>{{__('menu.dashboard')}}</p>
              </a>


              <a href="{{url('profile/'. Auth::user()->name)}}" class="dropdown-item">
                <i class="now-ui-icons users_circle-08" aria-hidden="true"></i>
                <p>{{__('menu.profile')}}</p>
              </a>
```

```
            <a href="" data-toggle="modal" data-target="#apartment" class="dropdown-item">
                    <i class="now-ui-icons arrows-1_share-66" aria-hidden="true"></i>
                    <p>{{ __('menu.Logout') }}</p>
                </a>
            </div>
          </li>


        @endguest
        <li class="nav-item dropdown">
          <a id="navbarDropdown" class="nav-link dropdown-toggle" href="#" role="button" data-toggle="dropdown"
aria-haspopup="true" aria-expanded="false" v-pre>
            <span>{{__('menu.language')}}</span>
          </a>
          <ul class="dropdown-menu dropdown-menu-right"
            aria-labelledby="navbarDropdown" role="menu">
            <li>
              <a href="{{route('set_language', ['es'])}}" class="dropdown-item">
                {{__('menu.spain')}}
              </a>
            </li>
            <li>
              <a href="{{route('set_language', ['en'])}}" class="dropdown-item">
                {{__('menu.english')}}
              </a>
            </li>
          </ul>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

Se incluye también en esta vista “navbar” un modal.

```
<!-- Modal -->
<div id="apartment" class="modal fade" role="dialog">
  <div class="modal-dialog">
    <!-- Modal content-->
    <div class="modal-content bg-light">
      <div class="modal-header" style="height:20px;">
        <button type="button" class="close" data-dismiss="modal">&times;</button>
      </div>
      <div class="modal-body">
        <p class="justify-content-center text-center">{{__('auth.session')}}</p>
      </div>
      <div class="modal-footer" style="margin-right: 10px">
        <form id="logout-form" action="{{ route('logout') }}" method="POST">
          @csrf
          <input type="submit" class="btn btn-danger" value="{{__('apartments.accept')}}"
name="{{__('apartments.accept')}}">
        </form>
        <button type="button" class="btn btn-primary" data-dismiss="modal">{{__('apartments.cancel')}}</button>
      </div>
    </div>
  </div>
</div>
```

app-verify.php

Esta vista se ha creado para mostrar un formulario de verificación y es similar a alas anteriores pudiendo diferenciar lo siguiente.

```
<body class="login-page">

<noscript>
  <iframe src="https://www.googletagmanager.com/ns.html?id=GTM-NKDMSK6" height="0" width="0"
style="display:none;visibility:hidden"></iframe>
</noscript>

@yield('content')

</body>
```

Esta vista tiene el siguiente script.

```
<script>
  $(function() {
    $('#alert').click(function(){
      $('#box-two').css("transform","translate(250px,0)");
    });
  });
</script>
```

- Dashboard

Para la parte del dashboard se han creado dos layouts para poder utilizar en algunos sitios peticiones mediante ajax ya que han surgido algunos problemas y se ha decidido crear dos, una para peticiones normales y otra para mostrarla mediante ajax.

app-dash.php

Para la parte donde realizamos peticiones por ajax se ha creado este laoyout, el cual lo vamos a seccionar.

Empezamos primeramente hasta la apertura de la etiqueta “body”.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="csrf-token" content="{{ csrf_token() }}">
  <link rel="apple-touch-icon" sizes="76x76" href="{{asset('assets/img/apple-icon.png')}}">
  <link rel="icon" type="image/png" href="{{asset('assets/img/favicon.png')}}">
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
  <title>
    HousingBook
  </title>
<meta content='width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0, shrink-to-fit=no' name='viewport' />

  <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700,200" rel="stylesheet" />
  <link href="https://maxcdn.bootstrapcdn.com/font-awesome/latest/css/font-awesome.min.css" rel="stylesheet">
  <link href="https://use.fontawesome.com/releases/v5.0.6/css/all.css" rel="stylesheet">
  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.0/css/all.css"
integrity="sha384-lZN37f5QGtY3VHgisS14W3ExzMWZxybE1SJSEsQp9S+oqd12jhcu+A56Ebc1zFSJ"
crossorigin="anonymous">

  <link href="{{asset('assets/css/bootstrap.min.css')}}" rel="stylesheet" />
  <link href="{{asset('assets/css/paper-dashboard.css?v=2.0.1')}}" rel="stylesheet" />

  <link href="{{asset('assets/demo/demo.css')}}" rel="stylesheet" />
  <style>
    .red-tooltip + .tooltip > .tooltip-inner {background-color: #f00;}
    .btns-calendar button{
      background-color: #f48942;
    }
    .btns-calendar button:hover{
      background-color: #c46e35;
    }
    .fondo{
      background-color: #e2e0dc;
    }
  </style>
</head>

<body class="">
```

Esta siguiente sección muestra una barra lateral en la parte izquierda.

```
<div class="wrapper ">
  <div class="sidebar" data-color="brown" data-active-color="danger">      <div class="sidebar-wrapper">
        <div class="user">
          <div class="photo">
            <img src="{{url('/storage/photos/'.auth()->user()->photo)}}"  />
          </div>
          <div class="info">
            <a data-toggle="collapse" href="#collapseExample" class="collapsed">
         <span>
          {{Auth::user()->name}}
            <b class="caret"></b>
         </span>
            </a>
            <div class="clearfix"></div>
            <div class="collapse" id="collapseExample">
              <ul class="nav">
                <li>
                  <a href="#" id="profile">
                    <span class="sidebar-mini-icon">MP</span>
                    <span class="sidebar-normal">{{__('dashboard.profile')}}</span>
                  </a>
                </li>
                <li>
                  <a href="{{route('user.edit')}}" id="profileedit">
                    <span class="sidebar-mini-icon">EP</span>
                    <span class="sidebar-normal">{{__('dashboard.edit')}}</span>
                  </a>
                </li>
                <li>
                  <a href="#" id="profilepassword">
                    <span class="sidebar-mini-icon">PC</span>
                    <span class="sidebar-normal">{{__('dashboard.password')}}</span>
                  </a>
                </li>
                <li>
                  <a href="{{route('user.telegram')}}" id="profiletelegram">
                    <span class="sidebar-mini-icon">TI</span>
                    <span class="sidebar-normal">{{__('dashboard.telegram')}}</span>
                  </a>
                </li>

              </ul>
            </div>
          </div>
        </div>
        <ul class="nav">
          <li id="activo1">
            <a href="#" id="dashboard">
              <i class="nc-icon nc-bank nc-bullet-list-67"></i>
              <p>{{__('menu.dashboard')}}</p>
            </a>
          </li>
          <li id="activo2">
            <a data-toggle="collapse" href="#mapsExamples">
              <i class="nc-icon nc-pin-3"></i>
              <p>
                {{__('menu.apartment')}}
                <b class="caret"></b>
              </p>
            </a>
```

```
            <div class="collapse" id="mapsExamples">
                <ul class="nav">
                    <li>
                        <a href="{{ url('/dashboard/apartment/create') }}">
                            <span class="sidebar-mini-icon">AA</span>
                            <span class="sidebar-normal">{{__('profile.createapartment')}}</span>
                        </a>
                    </li>
                    <li>
                        <a href="#" id="apartmentmanage">
                            <span class="sidebar-mini-icon">EA</span>
                            <span class="sidebar-normal">{{__('profile.manage')}}</span>
                        </a>
                    </li>
                </ul>
            </div>
        </li>
        <li id="activo3">
            <a data-toggle="collapse" href="#History">
                <i class="fa fa-history"></i>
                <p>
                    {{__('menu.history')}}
                    <b class="caret"></b>
                </p>
            </a>
            <div class="collapse " id="History">
                <ul class="nav">
                    <li>
                        <a href="#" id="history">
                            <span class="sidebar-mini-icon">H</span>
                            <span class="sidebar-normal">{{__('menu.history')}}</span>
                        </a>
                    </li>
                    <li>
                        <a href="#" id="invoices">
                            <span class="sidebar-mini-icon">I</span>
                            <span class="sidebar-normal">{{__('menu.invoices')}}</span>
                        </a>
                    </li>
                </ul>
            </div>
        </li>

    </ul>
  </div>
</div>
```

Esta sección muestra una barra horizontal en la parte superior.

```
<div class="main-panel">
  <!-- Navbar -->
  <nav class="navbar navbar-expand-lg navbar-absolute fixed-top navbar-transparent">
    <div class="container-fluid">
      <div class="navbar-wrapper">
        <div class="navbar-minimize">
          <button id="minimizeSidebar" class="btn btn-icon btn-round">
            <i class="nc-icon nc-minimal-right text-center visible-on-sidebar-mini"></i>
            <i class="nc-icon nc-minimal-left text-center visible-on-sidebar-regular"></i>
          </button>
        </div>
        <div class="navbar-toggle">
          <button type="button" class="navbar-toggler">
            <span class="navbar-toggler-bar bar1"></span>
            <span class="navbar-toggler-bar bar2"></span>
            <span class="navbar-toggler-bar bar3"></span>
          </button>
        </div>
        <a class="navbar-brand" href="/">HousingBook</a>
      </div>
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navigation"
aria-controls="navigation-index" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-bar navbar-kebab"></span>
        <span class="navbar-toggler-bar navbar-kebab"></span>
        <span class="navbar-toggler-bar navbar-kebab"></span>
      </button>
      <div class="collapse navbar-collapse justify-content-end" id="navigation">

        <ul class="navbar-nav">
          <li class="nav-item">

            <a class="nav-link btn-magnify" href="{{route('apartments.public')}}">
              <i class="nc-icon nc-minimal-left"></i>
              <p>
                <span class="d-lg-none d-md-block"></span>
              </p>{{__('menu.return')}}
            </a>
          </li>
          <li class="nav-item btn-rotate dropdown">
            <a class="nav-link dropdown-toggle" href="http://example.com" id="navbarDropdownMenuLink"
data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
              {{ Auth::user()->name . ' - ' . Auth::user()->email }}
              <p>
                <span class="d-lg-none d-md-block"></span>
              </p>
            </a>
            <div class="dropdown-menu dropdown-menu-right" aria-labelledby="navbarDropdownMenuLink">

              <form id="logout-form" action="{{ route('logout') }}" method="POST" style="display: none;">
                @csrf
              </form>
              <a href="" data-toggle="modal" data-target="#apartment" class="dropdown-item">
                <i class="now-ui-icons arrows-1_share-66" aria-hidden="true"></i>
                <p>{{ __('menu.Logout') }}</p>
              </a>
            </div>
          </li>
          <li class="nav-item dropdown">
            <a id="navbarDropdown" class="nav-link dropdown-toggle" href="#" role="button" data-toggle="dropdown"
aria-haspopup="true" aria-expanded="false" v-pre>
```

```
            <span>{{__('menu.language')}}</span>
                </a>
                <ul class="dropdown-menu dropdown-menu-right"
                  aria-labelledby="navbarDropdown" role="menu">
                  <li>
                    <a href="{{route('set_language', ['es'])}}" class="dropdown-item">
                      {{__('menu.spain')}}
                    </a>
                  </li>
                  <li>
                    <a href="{{route('set_language', ['en'])}}" class="dropdown-item">
                      {{__('menu.english')}}
                    </a>
                  </li>
                </ul>
              </li>

            </ul>
          </div>
        </div>
      </nav>
```

En la siguiente sección es donde se van a mostrar errore, mensajes “success” en el caso de que los haya y es dónde va el contenido en dónde se introducirá el html solicitado mediante ajax.

```
    <br><br><br>
    <div class="mt-5 row justify-content-center">
      @if(count($errors) > 0)
        <div class="col-md-5">
          @foreach($errors->all() as $error)
            <div class="callout alert alert-danger">
              {{$error}}
            </div>
          @endforeach
        </div>
      @endif
      @if(session('success'))
        <div class="col-md-5">
          <div class="alert alert-info">
            {{session('success')}}
          </div>
        </div>
      @endif

      @if(session('error'))
        <div class="col-md-5">
          <div class="alert alert-info">
            {{session('error')}}
          </div>
        </div>
      @endif
    </div>
    <div id="ajaxviews"></div>

</div>
```

Se hace importación de los siguientes scripts.

```
<script src="{{asset('assets/js/core/jquery.min.js')}}"></script>
<script src="{{asset('assets/js/core/popper.min.js')}}"></script>
<script src="{{asset('assets/js/core/bootstrap.min.js')}}"></script>
<script src="{{asset('assets/js/plugins/perfect-scrollbar.jquery.min.js')}}"></script>
<script src="{{asset('assets/js/plugins/moment.min.js')}}"></script>
<!--  Plugin for Switches, full documentation here: http://www.jque.re/plugins/version3/bootstrap.switch/ -->
<script src="{{asset('assets/js/plugins/bootstrap-switch.js')}}"></script>
<!--  Plugin for Sweet Alert -->
<script src="{{asset('assets/js/plugins/sweetalert2.min.js')}}"></script>
<!-- Forms Validations Plugin -->
<script src="{{asset('assets/js/plugins/jquery.validate.min.js')}}"></script>
<!--  Plugin for the Wizard, full documentation here: https://github.com/VinceG/twitter-bootstrap-wizard -->
<script src="{{asset('assets/js/plugins/jquery.bootstrap-wizard.js')}}"></script>
<!--   Plugin for Select, full documentation here: http://silviomoreto.github.io/bootstrap-select -->
<script src="{{asset('assets/js/plugins/bootstrap-selectpicker.js')}}"></script>
<!--  Plugin for the DateTimePicker, full documentation here: https://eonasdan.github.io/bootstrap-datetimepicker/ -->
<script src="{{asset('assets/js/plugins/bootstrap-datetimepicker.js')}}"></script>
<!--  DataTables.net Plugin, full documentation here: https://datatables.net/    -->
<script src="{{asset('assets/js/plugins/jquery.dataTables.min.js')}}"></script>
<!--   Plugin for Tags, full documentation here: https://github.com/bootstrap-tagsinput/bootstrap-tagsinputs  -->
<script src="{{asset('assets/js/plugins/bootstrap-tagsinput.js')}}"></script>
<!-- Plugin for Fileupload, full documentation here: http://www.jasny.net/bootstrap/javascript/#fileinput -->
<script src="{{asset('assets/js/plugins/jasny-bootstrap.min.js')}}"></script>
<!--  Full Calendar Plugin, full documentation here: https://github.com/fullcalendar/fullcalendar    -->
<script src="{{asset('assets/js/plugins/fullcalendar.min.js')}}"></script>
<!-- Vector Map plugin, full documentation here: http://jvectormap.com/documentation/ -->
<script src="{{asset('assets/js/plugins/jquery-jvectormap.js')}}"></script>
<!--  Plugin for the Bootstrap Table -->
<script src="{{asset('assets/js/plugins/nouislider.min.js')}}"></script>
<!--  Google Maps Plugin    -->
<script src="https://maps.googleapis.com/maps/api/js?key=YOUR_KEY_HERE"></script>
<!-- Chart JS -->
<script src="{{asset('assets/js/plugins/chartjs.min.js')}}"></script>
<!--  Notifications Plugin    -->
<script src="{{asset('assets/js/plugins/bootstrap-notify.js')}}"></script>
<!-- Control Center for Now Ui Dashboard: parallax effects, scripts for the example pages etc -->
<script src="{{asset('assets/js/paper-dashboard.min.js?v=2.0.1')}}" type="text/javascript"></script>
<!-- Paper Dashboard DEMO methods, don't include it in your project! -->
<script src="{{asset('assets/demo/demo.js')}}"></script>
```

El script que se encarga de realizar las peticiones ajax es el siguiente. Se han identificado diferentes botones y se les ha añadido un evento para realizar la petición ajax.

```
<script>
  $(document).ready(function() {

    $.ajax(
      {
        url: "/dashboard/index",
        type: 'GET',
      }).done(

      function(data)
      {
        $('#ajaxviews').html(data.html);
      }
    );

    $("#dashboard").click(function(event){
      event.preventDefault();
      location.reload();
    });

    $("#history").click(function(event){
      event.preventDefault();
      $.ajax(
        {
          url: "/dashboard/invoices/index",
          type: 'GET',
        }).done(

        function(data)
        {
          $('#ajaxviews').html(data.html);
        }
      );
    });

    $("#invoices").click(function(event){
      event.preventDefault();
      $.ajax(
        {
          url: "/dashboard/invoices/invoices",
          type: 'GET',
        }).done(

        function(data)
        {
          $('#ajaxviews').html(data.html);
        }
      );
    });
```

```
$("#profile").click(function(event){
        event.preventDefault();
        $.ajax(
           {
              url: "/dashboard/user/show",
              type: 'GET',
           }).done(

           function(data)
           {
              $('#ajaxviews').html(data.html);
           }
        );
     });

     $("#profilepassword").click(function(event){
        event.preventDefault();
        $.ajax(
           {
              url: "/dashboard/user/password",
              type: 'GET',
           }).done(

           function(data)
           {
              $('#ajaxviews').html(data.html);
           }
        );
     });

     $("#apartmentmanage").click(function(event){
        event.preventDefault();
        $.ajax(
           {
              url: "/dashboard/apartment/index",
              type: 'GET',
           }).done(

           function(data)
           {
              $('#ajaxviews').html(data.html);
           }
        );
     });
  });
</script>
```

app-dash.php

Esta vista es similar a la anterior solo que como no hace uso de ajax se ha prescindido de los script que realizan la petición y del div donde se introduce la información tras las peticiones ajax `<div id="ajaxviews"></div>` . En su lugar se introduce lo siguiente.

```
@yield('content')
```

En estas dos vistas se introduce un modal.

```
<div id="apartment" class="modal fade" role="dialog">
  <div class="modal-dialog">

    <!-- Modal content-->
    <div class="modal-content bg-light">
      <div class="modal-header" style="height:20px;">
        <button type="button" class="close" data-dismiss="modal">&times;</button>
      </div>
      <div class="modal-body">
        <p class="justify-content-center text-center">{{__('auth.session')}}</p>
      </div>
      <div class="modal-footer" style="margin-right: 10px">
        <form id="logout-form" action="{{ route('logout') }}" method="POST">
          @csrf
          <input type="submit" class="btn btn-danger" value="{{__('apartments.accept')}}"
name="{{__('apartments.accept')}}">
        </form>
        <button type="button" class="btn btn-primary" data-dismiss="modal">{{__('apartments.cancel')}}</button>
      </div>
    </div>

  </div>
</div>
```

- Errores

Se han personalizado los errores en el proyecto. Para ello se ha creado un layout llamado error y se han creado diferentes vistas para los error y se le ha dado estilo quedando de la siguiente manera.

```
<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>@yield('title')</title>
  <link rel="dns-prefetch" href="//fonts.gstatic.com">
  <link href="https://fonts.googleapis.com/css?family=Nunito" rel="stylesheet" type="text/css">
  <style>
    html, body {
      background-color: #fff;
      color: #636b6f;
      font-family: 'Nunito', sans-serif;
      font-weight: 100;
      height: 100vh;
      margin: 0;
    }

    .full-height {
      height: 100vh;
    }
    .flex-center {
      align-items: center;
      display: flex;
      justify-content: center;
    }
    .position-ref {
      position: relative;
    }
    .code {
      color: #f48942;
      border-right: 4px solid;
      font-size: 60px;
      padding: 0 15px 0 15px;
      text-align: center;
    }
    .message {
      font-size: 18px;
      text-align: center;
    }
    .titulo{
      font-size: 40px;
      background: linear-gradient(to top,  #f48942 -5%, #ba6832 100%);
      -webkit-background-clip: text;
      -webkit-text-fill-color: transparent;
      margin-right: 50px;
    }
  </style>
```

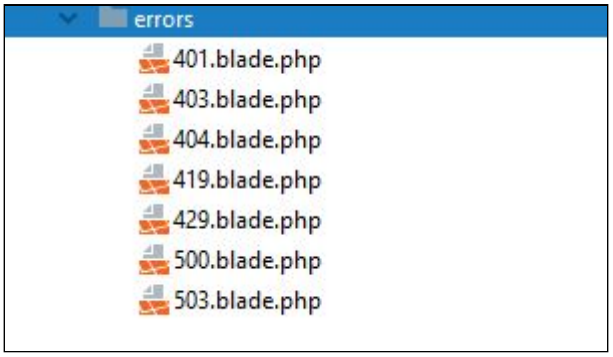
```
</head>
<body>
<div class="flex-center position-ref full-height">
  <h1 class="titulo">HousingBook </h1>
  <div class="code">
    <strong>@yield('code')</strong>
  </div>

  <div class="message" style="padding: 10px;">
    @yield('message_esp')
  </div>
  <div class="message" style="padding: 10px;">
    @yield('message_en')
  </div>
</div>
</body>
</html>
```

Para cada vista de cada error se ha creado una vista similar a esta pero diferenciando el tipo de error y especificando el mensaje según el error.

```
@extends('layouts.error_illustrated')

@section('title', __('Unauthorized'))
@section('code', '401')
@section('message_en', 'Unauthorized')
@section('message_esp', 'No autorizado')
```



- Vista welcome

Se ha creado una vista principal para la página dónde se introduce un buscador para buscar por ciudades y varias tarjetas para buscar por categorías al hacer click sobre ellas.

Se incluye uno de los layouts mencionados y un header.

```
@extends('layouts.app-nav')
@include('partials.header')
```

El header que se muestra en esta vista welcome queda de la siguiente manera.

```
<div class="page-header clear-filter">
  <div class="rellax-header rellax-header-sky" data-rellax-speed="-4">
    <div class="page-header-image" style="background-image: url('../img/presentation-page/nuk-pro-back-sky.jpg')">
    </div>
  </div>
  <div class="rellax-header rellax-header-buildings" data-rellax-speed="0">
                                        <div class="page-header-image page-header-city" style="background-image:
url('../img/presentation-page/nuk-pro-buildings.png')">
    </div>
  </div>
  <div class="rellax-text-container rellax-text" style="font-size: 10px">
    <h1 class="h1-seo rellax-text" data-rellax-speed="-1">HousingBook</h1>

  </div>
  <h3 class="h3-description rellax-text" data-rellax-speed="-1">{{__('welcome.head')}}</h3>
</div>
```

El html perteneciente a la vista welcome es el siguiente.

```
@section('content')
  <div class="section section-components" data-background-color="dark-red">
    <div class="container">
      <div class="row mt-1" >
        <div class="col-md-8 ml-auto mr-auto">
          <h2 class="text-center title">{{__('welcome.search')}} <br><br>
            <small class="description">{{__('welcome.anywhere')}}</small>
          </h2>
          {{Form::open(['method' => 'GET','action' => 'ApartmentController@search'])}}
          <div class="row justify-content-center">
            <div class="col-md-6">
              <div class="form-group">
                    {{Form::text('search','',['placeholder' => __('welcome.input'), 'class' => 'form-control','autocomplete' =>
'family-name', 'required'])}}
              </div>
            </div>
          </div>
          <div class="row justify-content-center">
            <div class="col-md-4">
              <button type="submit" class="btn btn-primary btn-round btn-block">{{__('welcome.searchs')}}</button>
            </div>
          </div>
          {{Form::close()}}
          <h2 class="text-center title"> <small class="description">{{__('welcome.prefeer')}}</small> </h2>
        </div>
      </div>
      <div class="row">
        <div class="col-md-3">
          <div class="card-container first-card">
            <div class="card-component">
              <a href="{{route('searchCategory', 1)}}">
                <div class="front">
                  <img src="{{ asset('img/mountain.jpg')}}">
                </div>
              </a>
            </div>
          </div>
        </div>
        <div class="col-md-3">
          <div class="card-container second-card">
            <div class="card-component">
              <a href="{{route('searchCategory', 2)}}">
                <div class="front">
                  <img src="{{ asset('img/city.jpg')}}">
                </div>
              </a>
            </div>
          </div>
        </div>
        <div class="col-md-3">
          <div class="card-container third-card">
            <div class="card-component">
              <a href="{{route('searchCategory', 3)}}">
                <div class="front">
                  <img src="{{ asset('img/campo.jpg')}}">
                </div>
              </a>
```

```
                </div>

            </div>
          </div>
          <div class="col-md-3">
            <div class="card-container fourth-card">
              <div class="card-component">
                <a href="{{route('searchCategory',  4)}}">
                  <div class="front">
                    <img src="{{ asset('img/beach.jpg')}}">
                  </div>
                </a>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
@endsection
```

- Página principal

Para la parte principal de la página se han creado varias vistas que se van a mostrar a continuación.

- index.blade.php de perfil de usuario

En esta vista se muestra información del usuario con todos sus apartamentos.

```
<div class="wrapper">
  <div class="page-header page-header-small header-filter" filter-color="orange">
    <div class="page-header-image" data-parallax="true" style="background-image:url('../img/bg5.jpg');">
    </div>
    <div class="container">
      <div class="photo-container">
        <img src="{{url('/storage/photos/'.$user->photo)}}" alt="{{$user->name}}" style="width: 100px; height: 100px;">
      </div>
      <h3 class="title">{{$user->name}}</h3>
      <p class="category">{{$user->last_name}}</p>
      <div class="content">
        <div class="social-description">
          <div class="social-description">
            <h2>{{count($user->apartments)}}</h2>
            <p>{{__('menu.apartment')}}</p>
          </div>
        </div>

      </div>
    </div>
  </div>
```

```
<div class="container">
  <div class="row justify-content-center">
    @foreach($user->apartments as $apartment)
      <div class="col-md-8 mb-5 shadow rounded p-3">
        <div class="row">
          <div class="col-lg-3 col-md-6">

            <ul class="nav nav-pills nav-pills-primary nav-pills-icons flex-column" role="tablist">
              <li class="nav-item">
                <a class="nav-link active" data-toggle="tab" href="#link{{$nums}}" role="tablist">
                  <i class="now-ui-icons text_align-left"></i> {{__('apartments.details')}}
                </a>
              </li>
              <li class="nav-item">
                <a class="nav-link" data-toggle="tab" href="#link{{$nums + 1}}" role="tablist">
                  <i class="now-ui-icons media-1_camera-compact"></i> {{__('apartments.photos')}}
                </a>
              </li>
            </ul>
          </div>
          <div class="col-md-8">
            <div class="tab-content">
              <div class="tab-pane active" id="link{{$nums}}">
                <h6 class="text-primary">
                  {{$apartment->name}}
                </h6>
                {{$apartment->short_description}}
              </div>
              <div class="tab-pane ml-5" id="link{{$nums + 1}}">
                <div id="productCarousel{{$nums + 1}}" class="carousel slide" data-ride="carousel" data-interval="8000"
style="width: 80%; height: 80%;">
                  <ol class="carousel-indicators">
                    <li data-target="#productCarousel" data-slide-to="0" class="active"></li>
                    <li data-target="#productCarousel" data-slide-to="1"></li>
                    <li data-target="#productCarousel" data-slide-to="2"></li>
                    <li data-target="#productCarousel" data-slide-to="3"></li>
                  </ol>
                  <div class="carousel-inner" role="listbox">
                    @if(count($apartment->photos))
                      <div class="carousel-item active">
                        <img class="d-block img-raised" src="{{url('/storage/photos/'.$apartment->photos[0]->local_url)}}"
alt="{{$apartment->name}}">
                      </div>
                      <div class="carousel-item">
                        <img class="d-block img-raised" src="{{url('/storage/photos/'.$apartment->photos[1]->local_url)}}"
alt="{{$apartment->name}}">
                      </div>
                      <div class="carousel-item">
                        <img class="d-block img-raised" src="{{url('/storage/photos/'.$apartment->photos[2]->local_url)}}"
alt="{{$apartment->name}}">
                      </div>
                      <div class="carousel-item">
                        <img class="d-block img-raised" src="{{url('/storage/photos/'.$apartment->photos[3]->local_url)}}"
alt="{{$apartment->name}}">
                      </div>
                    @else
                      <div class="carousel-item active">
                        <img class="d-block img-raised" src="{{ asset('img/notfound.png')}}" width="350" height="200"
alt="{{$apartment->name}}">
                      </div>
```

```
                            <div class="carousel-item">
                              <img class="d-block img-raised" src="{{ asset('img/notfound.png')}}" width="350" height="200"
alt="{{$apartment->name}}">
                            </div>
                            <div class="carousel-item">
                              <img class="d-block img-raised" src="{{ asset('img/notfound.png')}}" width="350" height="200"
alt="{{$apartment->name}}">
                            </div>
                            <div class="carousel-item">
                              <img class="d-block img-raised" src="{{ asset('img/notfound.png')}}" width="350" height="200"
alt="{{$apartment->name}}">
                            </div>
                          @endif
                        </div>
                        <a class="carousel-control-prev" href="#productCarousel{{$nums + 1}}" role="button" data-slide="prev">
                          <button type="button" class="btn btn-primary btn-icon btn-round btn-sm" name="button">
                            <i class="now-ui-icons arrows-1_minimal-left"></i>
                          </button>
                        </a>
                        <a class="carousel-control-next" href="#productCarousel{{$nums + 1}}" role="button" data-slide="next">
                          <button type="button" class="btn btn-primary btn-icon btn-round btn-sm" name="button">
                            <i class="now-ui-icons arrows-1_minimal-right"></i>
                          </button>
                        </a>
                      </div>
                    </div>
                  </div>
                </div>
              </div>
            </div>
            <?php $nums = $nums + 2;?>
        @endforeach
    </div>
</div>
```

- reservations.blade.php

Se ha creado una vista para los usuarios autenticados donde se muestra si tiene reservas pendientes o si tiene reservas pasadas.

```
@extends('layouts.app-nav')
@section('content')
  <div class="blogs-5" style="background-image: url('{{asset('img/header.jpg')}}');">
    <div class="container">
      <h1 class="mt-5 text-light mb-5">{{__('profile.reservations')}}</h1>
    </div>
  </div>
  <div class="container mt-5">
    <h3 class="text-center">{{__('profile.pendingreservations')}}</h3>
    <hr>
    <div class="row justify-content-center">
      <div class="col-md-8"></div>
      <div class="col-md-2">
        <h5 class="text-center">{{__('profile.entry')}}</h5>
      </div>
      <div class="col-md-2">
        <h5 class="text-center">{{__('profile.exit')}}</h5>
      </div>
    </div>
    @forelse($pending as $item)
      <div class="row justify-content-center">
        <div class="col-md-8">
          <p class="text-center p-1 text-light rounded" style="background-color:
{{$item->category->color}};">{{$item->name}}</p>
        </div>
        @if(config('app.locale') == 'es')
          <div class="col-md-2">
            <p class="text-center p-1 text-light rounded bg-info ">{{date("d/m/Y", strtotime($item->pivot->entry))}}</p>
          </div>
          <div class="col-md-2">
            <p class="text-center p-1 text-light rounded bg-info">{{date("d/m/Y", strtotime($item->pivot->exit))}}</p>
          </div>
        @else
          <div class="col-md-2">
            <p class="text-center p-1 text-light rounded bg-info ">{{$item->pivot->entry}}</p>
          </div>
          <div class="col-md-2">
            <p class="text-center p-1 text-light rounded bg-info">{{$item->pivot->exit}}</p>
          </div>
        @endif
      </div>
    @empty
      <h3 class="text-center text-primary">{{__('profile.noreservations')}}</h3>
    @endforelse
    <h3 class="mt-5 text-center">{{__('profile.pastreservations')}}</h3>
    <hr>
    <div class="row justify-content-center">
      <div class="col-md-8"></div>
      <div class="col-md-2">
        <h5 class="text-center">{{__('profile.entry')}}</h5>
      </div>
```

```
<div class="col-md-2">
            <h5 class="text-center">{{__('profile.exit')}}</h5>
        </div>
    </div>
    @forelse($past as $item)
        <div class="row justify-content-center">
            <div class="col-md-8">
                <p class="text-center p-1 text-light rounded" style="background-color:
{{$item->category->color}};">{{$item->name}}</p>
            </div>
            @if(config('app.locale') == 'es')
                <div class="col-md-2">
                    <p class="text-center p-1 text-light rounded bg-info ">{{date("d/m/Y", strtotime($item->pivot->entry))}}</p>
                </div>
                <div class="col-md-2">
                    <p class="text-center p-1 text-light rounded bg-info">{{date("d/m/Y", strtotime($item->pivot->exit))}}</p>
                </div>
            @else
                <div class="col-md-2">
                    <p class="text-center p-1 text-light rounded bg-info ">{{$item->pivot->entry}}</p>
                </div>
                <div class="col-md-2">
                    <p class="text-center p-1 text-light rounded bg-info">{{$item->pivot->exit}}</p>
                </div>
            @endif
        </div>
    @empty
        <h3 class="text-center text-primary">{{__('profile.noreservationspass')}}</h3>
    @endforelse
  </div>
@endsection
```

- index.blade.php de apartamentos

Para las casas (donde se muestran todas) se ha creado una vista que vamos a seccionar en varias partes para entenderla mejor.

Esta vista extiende del layout con “navbar”.

```
@extends('layouts.app-nav')
```

La sección `@section('content')` contiene lo siguiente, una primera sección con una imagen de fondo donde se van a mostrar los errores y los mensajes de “success”.

```
<div class="blogs-5" style="background-image: url('{{asset('/img/aparts.jpg')}}');">
    <div class="container">
      <div class="row justify-content-center">
        <div class="col-md-7 text-center">
          @if(session('success'))
            <div class="alert alert-info alert-dismissible fade show">
              <button type="button" aria-hidden="true" class="close" data-dismiss="alert" aria-label="Close">
                <i class="nc-icon nc-simple-remove"></i>
              </button>
              <span>{{session('success')}}</span>
            </div>
          @endif
          @if(session('error'))

            <div class="alert alert-danger alert-dismissible fade show">
              <button type="button" aria-hidden="true" class="close" data-dismiss="alert" aria-label="Close">
                <i class="nc-icon nc-simple-remove"></i>
              </button>
              <span><h6>{{session('error')}}</h6></span>
            </div>
          @endif
        </div>
      </div>
```

Esta siguiente parte es donde se muestra una foto con los últimas tres casas añadidas.

```
<div class="row">
        <div class="col-md-10 ml-auto mr-auto">
          <h2 class="title text-light">{{__('apartments.latest')}}</h2>
          <div class="row">

            @foreach($latest_apartment as $apartment)
              <div class="col-md-4">
                <div class="card card-blog">
                  <div class="card-image">
                    <a href="{{route('apartments.show',$apartment->id)}}">
                      @if(count($apartment->photos))
                        <img class="img rounded" src="{{url('/storage/photos/'.$apartment->photos[0]->local_url)}}">
                      @else
                        <img class="img rounded" src="{{ asset('img/notfound.png')}}" height="170" width="350"/>
                      @endif
                    </a>
                  </div>
                  <div class="card-body">
                    <h6 class="category text-primary">{{$apartment->name}}</h6>
                    <h5 class="card-title">
                      {{$apartment->city->name}}
                    </h5>
                    <div class="card-footer">
                      <div class="author">
                        <img src="{{url('/storage/photos/'.$apartment->user->photo)}}" alt="..." class="avatar img-raised">
                        <span>{{$apartment->user->name}}</span>
                      </div>

                    </div>
                  </div>
                </div>
              </div>
            @endforeach


          </div>
        </div>
      </div>
    </div>
```

En la parte izquierda de la página se muestra un formulario para filtrar las casas por precio, categoría, etc.

```
<div class="section">
  <div class="row">
    <div class="col-md-3">
      <div class="container p-2">
        <h3>{{__('form.filter')}}</h3>
        <hr>
        <div class="container border shadow">
          {{Form::open(['method' => 'POST', 'action'=>'ApartmentController@filter'])}}
          <h4 class="card-title">
            <button class="btn btn-default btn-icon btn-neutral pull-right" rel="tooltip" title="Reset Filter" type="reset">
              <i class="arrows-1_refresh-69 now-ui-icons"></i>
            </button>
          </h4>
          <h4 class="text-primary">{{__('form.categories')}}</h4>
          <hr>
          @foreach($categories as $category)
            <div class="form-check">

              <label class="form-check-label">
                <input class="form-check-input" type="checkbox" name="category[]" value="{{$category->id}}">
                <span class="form-check-sign bg-primary text-light"></span>
                {{__('form.' . $category->name)}}
              </label>
            </div>
          @endforeach
          <h4 class="text-primary">{{__('form.services')}}</h4>
          <hr>
          @foreach($services as $service)
            <div class="form-check">

              <label class="form-check-label">
                <input class="form-check-input" type="checkbox" name="service[]" value="{{$service->id}}">
                <span class="form-check-sign bg-primary text-light"></span>
                {{__('form.' . $service->name)}}
              </label>
            </div>
          @endforeach
          <h4 class="text-primary">{{__('form.price')}}</h4>
          <hr>
          <p>
            <span id="price-left" class="price-left pull-left" data-currency="&euro;">{{$min}} &euro;</span>
            <span class="rangePrice text-center justify-content-center" style="margin-left: 160px">{{$max/2}}</span>
            <span id="price-right" class="price-right pull-right" data-currency="&euro;">{{$max}} &euro;</span>
          </p>
          <br>
          <p><input type="range" name="range" class="form-control custom-range text-primary" max="{{$max}}"
min="{{$min}}" id="slider"></p>
          <button type="submit" class="btn btn-info" style="width: 100%">{{__('form.filtersend')}}</button>
          {{Form::close()}}
        </div>
      </div>
    </div>
```

Por último, esta parte muestra todas las casas, de manera paginada.

```
    <div class="col-md-7">
      <div class="flash-message">
        @if(count($errors) > 0)
          <div class="col-md-5">
            @foreach($errors->all() as $error)
              <div class="callout alert alert-danger">
                {{$error}}
              </div>
            @endforeach
          </div>
        @endif
      </div>
      <h2 class="section-title">{{__('guest.apartments')}}</h2>
      <hr>
      <div class="row">
        @forelse($apartments as $apartment)
          <div class="col-md-4 ">
            <div class="card card-blog ">
              <div class="card-image">
                <a href="{{route('apartments.show',$apartment->id)}}">
                  @if(count($apartment->photos))
                    <img class="img rounded" src="{{url('/storage/photos/'.$apartment->photos[0]->local_url)}}">
                  @else
                    <img class="img rounded" src="{{ asset('img/notfound.png')}}" height="200" width="350">
                  @endif
                </a>
              </div>
              <div class="card-body">
                <h6 class="category text-primary">{{$apartment->name}} </h6>
                <h5 class="card-title">{{$apartment->city->name}} </h5>
                <p>{{$apartment->short_description}}</p>
                <div class="card-footer text-center">
                  <div class="author">
                    <img src="{{url('/storage/photos/'.$apartment->user->photo)}}" alt="..." class="avatar img-raised">
                    <span>{{$apartment->user->name}}</span>
                  </div>
                </div>
                <div class="mt-2">
                  <div class="text-light rounded" style="background-color: {{$apartment->category->color}};">
                    <p  class="text-center">{{$apartment->category->name}}</p>
                  </div>
                </div>
              </div>
            </div>
          </div>
        @empty
          <h3>{{__('guest.noapartments')}}</h3>
        @endforelse
        <div class="mx-auto d-block">
          @if($apartments instanceof \Illuminate\Pagination\LengthAwarePaginator )
            {{$apartments->links()}}
          @endif
        </div>
      </div>
    </div>
  </div>
</div>
```

En esta vista se añade un script para que cuando, se cambie el valor de un rango se muestre el valor seleccionado.

```
<script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.4.0.min.js"></script>
<script>
  let rango = $(".rangePrice");
  $('#slider').change(function(event){
     rango.text(event.currentTarget.value + "€");
  });
</script>
```

- index.blade.php de apartamentos

Para mostrar la información de cada apartamento se ha creado la siguiente vista.
Esta vista extiende del layout con “navbar”.

```
@extends('layouts.app-nav')
```

En la primera parte de esta vista se muestra el titulo de la casa con una imagen de fondo.

```
<div class="blogs-5" data-parallax="true" style="background-image: url('{{asset('/img/bg32.jpg')}}');">
  <div class="container text-light mb-5">
    <div class="content-center mb-5">
      <div class="row mb-5">
        <div class="col-md-8 ml-auto mr-auto mb-5">
          <h1 class="title">{{$apartment->name}}</h1>
          <h5>{{$apartment->short_description}}</h5>
          <br><br><br>
        </div>
      </div>
    </div>
  </div>
</div>
```

A continuación entre la foto y la descripción de la casa es dónde se mostrarán los mensajes.

```
<div class="wrapper">
  <div class="section">
    <div class="container">
      <div class="row justify-content-center">
        <div class="col-md-7 text-center">
          @if(session('success'))
            <div class="alert alert-info alert-dismissible fade show">
              <button type="button" aria-hidden="true" class="close" data-dismiss="alert" aria-label="Close">
                <i class="nc-icon nc-simple-remove"></i>
              </button>
              <span>{{session('success')}}</span>
            </div>
          @endif
          @if(session('error'))

            <div class="alert alert-danger alert-dismissible fade show">
              <button type="button" aria-hidden="true" class="close" data-dismiss="alert" aria-label="Close">
                <i class="nc-icon nc-simple-remove"></i>
              </button>
              <span><h6>{{session('error')}}</h6></span>
            </div>
          @endif
        </div>
      </div>
```

La siguiente parte es la que muestra las fotos del apartamento.

```
<div class="row">
  <div class="col-md-5">
    <div id="productCarousel" class="carousel slide" data-ride="carousel" data-interval="8000">
      <ol class="carousel-indicators">
        <li data-target="#productCarousel" data-slide-to="0" class="active"></li>
        <li data-target="#productCarousel" data-slide-to="1"></li>
        <li data-target="#productCarousel" data-slide-to="2"></li>
        <li data-target="#productCarousel" data-slide-to="3"></li>
      </ol>
      <div class="carousel-inner" role="listbox">
        @if(count($apartment->photos))
          <div class="carousel-item active">
            <img class="d-block img-raised" src="{{url('/storage/photos/'.$apartment->photos[0]->local_url)}}"
alt="{{$apartment->name}}">
          </div>
          <div class="carousel-item">
            <img class="d-block img-raised" src="{{url('/storage/photos/'.$apartment->photos[1]->local_url)}}"
alt="{{$apartment->name}}">
          </div>
          <div class="carousel-item">
            <img class="d-block img-raised" src="{{url('/storage/photos/'.$apartment->photos[2]->local_url)}}"
alt="{{$apartment->name}}">
          </div>
          <div class="carousel-item">
            <img class="d-block img-raised" src="{{url('/storage/photos/'.$apartment->photos[3]->local_url)}}"
alt="{{$apartment->name}}">
          </div>
        @else
          <div class="carousel-item active">
            <img class="d-block img-raised" src="{{ asset('img/notfound.png')}}" width="600" height="350"
alt="{{$apartment->name}}">
          </div>
          <div class="carousel-item">
            <img class="d-block img-raised" src="{{ asset('img/notfound.png')}}" width="600" height="350"
alt="{{$apartment->name}}">
          </div>
          <div class="carousel-item">
            <img class="d-block img-raised" src="{{ asset('img/notfound.png')}}" width="600" height="350"
alt="{{$apartment->name}}">
          </div>
          <div class="carousel-item">
            <img class="d-block img-raised" src="{{ asset('img/notfound.png')}}" width="600" height="350"
alt="{{$apartment->name}}">
          </div>
        @endif
      </div>
      <a class="carousel-control-prev" href="#productCarousel" role="button" data-slide="prev">
        <button type="button" class="btn btn-primary btn-icon btn-round btn-sm" name="button">
          <i class="now-ui-icons arrows-1_minimal-left"></i>
        </button>
      </a>
      <a class="carousel-control-next" href="#productCarousel" role="button" data-slide="next">
        <button type="button" class="btn btn-primary btn-icon btn-round btn-sm" name="button">
          <i class="now-ui-icons arrows-1_minimal-right"></i>
        </button>
      </a>
    </div>
```

La siguiente sección de esta vista muestra información sobre la casa.

```
<br>
  <p class="blockquote blockquote-primary rounded">
    {{$apartment->short_description}}
    <br>
    <br>
    <small>{{$apartment->user->name}}</small>
  </p>
</div>
<div class="col-md-6 ml-auto mr-auto">
  <h2 class="title"> {{$apartment->name}} </h2>
  <h5 class="category">{{$apartment->address}}</h5>
  <h5 class="main-price">{{$apartment->city->name}}</h5>
  <h6>{{__('guest.priceper')}}: <span class="text-warning">{{$apartment->price}} &euro;</span></h6>
  <div id="accordion" role="tablist" aria-multiselectable="true" class="card-collapse">
    <div class="card card-plain">
      <div class="card-header" role="tab" id="headingOne">
        <a data-toggle="collapse" data-parent="#accordion" href="#collapseOne" aria-expanded="true"
aria-controls="collapseOne">
          {{__('guest.description')}}
          <i class="now-ui-icons arrows-1_minimal-down"></i>
        </a>
      </div>
      <div id="collapseOne" class="collapse show" role="tabpanel" aria-labelledby="headingOne">
        <div class="card-body">
          <p>{{$apartment->description}}</p>
        </div>
      </div>
    </div>
    <div class="card card-plain">
      <div class="card-header" role="tab" id="headingTwo">
        <a class="collapsed" data-toggle="collapse" data-parent="#accordion" href="#collapseTwo" aria-expanded="false"
aria-controls="collapseTwo">
          {{__('guest.owner')}}
          <i class="now-ui-icons arrows-1_minimal-down"></i>
        </a>
      </div>
      <div id="collapseTwo" class="collapse" role="tabpanel" aria-labelledby="headingTwo">
        <div class="card-body">
          <p><span class="text-left">{{$apartment->user->name}} </span><span class="text-right"><a
href="{{route('profile.index',$apartment->user->name)}}" class="btn btn-sm btn-primary text-light"
target="_blank">{{__('apartments.profile')}}</a></span></p>
        </div>
      </div>
    </div>

  </div>

</div>
```

A continuación se añade un formulario para realizar la reserva de la casa. En este formulario se escogen las fechas de entrada y de salida. Este formulario se ayuda de un script para deshabilitar los días en los que la casa no está disponible.

```
<div class="container mb-5 mt-5 bg-primary text-light rounded pb-2 pt-4">
  {{Form::open(['method' => 'POST', 'action' => ['RentController@validation', $apartment->id]])}}
  <div class="row align-middle">
    <div class="col-md-1 mt-3">
      <label>{{__('guest.checkin')}}</label>
    </div>
    <div class="col-md-4 mt-2">


      <input type="text" name="entrada" id="datepicker" class="form-control bg-light text-black" autocomplete="off"
required>
    </div>
    <div class="col-md-1 mt-3">
      <label>{{__('guest.checkout')}}</label>
    </div>
    <div class="col-md-4 mt-2">

      <input type="text" name="salida" id="datepicker2" class="form-control bg-light text-black" autocomplete="off"
required>
    </div>
    @can('rent', $apartment)
      <div class="col-md-2">
        <button class="btn btn-primary mr-3">{{__('guest.rent')}} <i class="now-ui-icons
shopping_cart-simple"></i></button>
      </div>
    @endcan
    @guest
      <div class="col-md-2">
        <button class="btn btn-primary mr-3">{{__('guest.rent')}} <i class="now-ui-icons
shopping_cart-simple"></i></button>
      </div>
    @endguest
  </div>
  {{Form::close()}}
</div>
```

Este es el script introducido en la misma vista que hace que los días en los que una casa está reservada no se puedan marcar.

```
<link href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-datepicker/1.5.0/css/bootstrap-datepicker.css" rel="stylesheet">
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.js"></script>
<script src="jquery-ui.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-datepicker/1.5.0/js/bootstrap-datepicker.js"></script>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-datepicker/1.6.4/css/bootstrap-datepicker3.css"/>
<script type="text/javascript">
  let array = {!! json_encode($allDates) !!};
  let language = "{!! config('app.locale'); !!}";
  let formato;
  formato = language == "es" ? "dd/mm/yyyy" : "yyyy-mm-dd";
  (function($){
     $.fn.datepicker.dates['es'] = {
        days: ["Domingo", "Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado"],
        daysShort: ["Dom", "Lun", "Mar", "Mié", "Jue", "Vie", "Sáb"],
        daysMin: ["Do", "Lu", "Ma", "Mi", "Ju", "Vi", "Sa"],
        months: ["Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto", "Septiembre", "Octubre",
"Noviembre", "Diciembre"],
        monthsShort: ["Ene", "Feb", "Mar", "Abr", "May", "Jun", "Jul", "Ago", "Sep", "Oct", "Nov", "Dic"],
        today: "Hoy",
        monthsTitle: "Meses",
        clear: "Borrar",
        weekStart: 1,
     };

     $('#datepicker').datepicker({
        format: formato,
        language: language,
        todayBtn: "linked",
        clearBtn: true,
        todayHighlight: true,
        autoclose: true,
        datesDisabled: array
     });

     $('#datepicker2').datepicker({
        format: formato,
        language: language,
        clearBtn: true,
        todayBtn: "linked",
        todayHighlight: true,
        autoclose: true,
        datesDisabled: array
     });
  }(jQuery));
</script>
```

La siguiente sección muestra si una casa tiene servicios o no. Si los tuviera se mostrarían en forma de icono.

```
<div class="container mb-5">
    <div class="card card-plain">
      <div class="card-header" role="tab" id="headingThree">
        <a class="collapsed text-black" data-toggle="collapse" data-parent="#accordion" href="#collapseThree"
aria-expanded="false" aria-controls="collapseThree">
          {{__('guest.services')}}
          <i class="now-ui-icons arrows-1_minimal-down"></i>
        </a>
        <hr>
      </div>
      <div id="collapseThree" class="collapse" role="tabpanel" aria-labelledby="headingThree">
        <div class="card-body text-center">

          @forelse($apartment->services as $service)
            <div class="d-inline-block p-5">
              <div class="">
                <p>{{__('form.' . $service->name)}}</p>
                <i class="fas {{$service->icon}} border border-info rounded-circle p-4 text-info" style="font-size:30px;"></i>
              </div>
            </div>
          @empty
            <p>{{__('guest.noservices')}}</p>
          @endforelse

        </div>
      </div>
    </div>

  </div>
</div>
```

La siguiente sección muestra los comentarios de una casa.

```
<div class="row">
    <div class="col-md-8 ml-auto mr-auto">
      <div class="media-area">
        <h3 class="title text-center">
          <small>{{count($apartment->comments)}} {{__('guest.comments')}}</small>
        </h3>
        @foreach($apartment->comments as $comment)
          @auth
            @if($comment->user->id == auth()->user()->id)
              <div class="col-md-2">
                {{Form::open(['method' => 'DELETE','action' =>
['CommentController@destroy',$comment->id,$apartment->id]])}}
                <button type="submit" class="btn btn-primary mr-3">{{__('guest.removecomment')}} <i
class="now-ui-icons ui-1_simple-remove"></i></button>
                {{Form::close()}}
              </div>
            @endif
          @endauth
          <div class="media">
            <a class="pull-left" href="#pablo">
              <div class="avatar">
                <img class="media-object img-raised" src="{{url('/storage/photos/'.$comment->user->photo)}}" alt="..." />
              </div>
            </a>
            <div class="media-body">
              <h5 class="media-heading">{{$comment->user->name}}
                <small class="text-muted">&middot; {{Config::get('app.locale') == 'es' ? date("d/m/Y H:i:s",
strtotime($comment->created_at)) : $comment->created_at}}</small>
              </h5>
              <p>{{$comment->text}}</p>

            </div>
          </div>
        @endforeach
      </div>

    </div>
  </div>
</div>
```

Por último añadimos en la página una sección para mostrar casas relacionadas con la que se está viendo en ese momento (en este caso, casas que están en la misma ciudad).

```
    <div class="section related-products" data-background-color="black">
      <div class="container">
        <h3 class="title text-center">{{__('guest.related')}}</h3>
        <div class="row">
          @forelse($randoms_apartments as $random_apartment)
              <div class="col-sm-6 col-md-3">
               <div class="card card-product">
                 <div class="card-image justify-content-center">
                   <a href="{{route('apartments.show',$random_apartment->id)}}">
                     @if(count($random_apartment->photos))
        <img class="img rounded" src="{{url('/storage/photos/'.$random_apartment->photos[0]->local_url)}}"/>
                     @else
                          <img class="img rounded" src="{{ asset('img/notfound.png')}}" height="130" width="350"/>
                     @endif
                    </a>
                 </div>
                 <div class="card-body">
                   <h5 class="card-title">
                     <span class="card-link">{{$random_apartment->name}}</span>
                   </h5>
                   <div class="card-description">
                     {{str_limit($random_apartment->short_description,50)}}
                   </div>
                   <div class="">
                     <div class="text-center justify-content-center">
                       <span class="text-black">{{$random_apartment->city->name}}</span>
                     </div>
                   </div>
                 </div>
               </div>
              </div>
          @empty
              <div class="text-center justify-content-center mx-auto d-block">
               <h3>{{__('guest.norelated')}}</h3>
              </div>
          @endforelse
        </div>
      </div>
    </div>
  </div>
</div>
```

- login.blade.php

Esta vista hereda de @extends('layouts.app-verify') y es la vista donde el usuario introducirá sus datos para iniciar sesión.

```
@section('content')
<div class="page-header header-filter" filter-color="orange">
  <div class="page-header-image" style="background-image:url('./img/login.jpg')"></div>
  <div class="content">
    <div class="container">
      <div class="row justify-content-center">
        @if ($errors->any())
              <div class="col-md-6" id="alert">
                 <div class="alert alert-danger rounded"  role="alert">
                    <div class="container">
                       <button type="button" class="close" data-dismiss="alert" aria-label="Close">
            <span aria-hidden="true">
             <i class="now-ui-icons ui-1_simple-remove"></i>
            </span>
                       </button>
                       <p><strong>{{__('profile.ups')}}</strong></p>
                       @foreach ($errors->all() as $error)
                          <p>{{ $error }}</p>
                       @endforeach
                    </div>
                 </div>
              </div>
        @endif
      </div>
      <div class="col-md-5 ml-auto mr-auto">
        <div class="card card-login card-plain">
          {{Form::open(['method' => 'POST', 'url'=>'login'])}}
            @csrf
            <div class="card-header text-center">
              <div class="logo-container">
                 <img src="../img/now-logo.png" alt="">
              </div>
            </div>
            <div class="card-body">
              <div class="input-group no-border input-lg">
                 <div class="input-group-prepend">
                    <span class="input-group-text"><i class="users_circle-08"></i></span>
                 </div>
                 {{ Form::email('email', old('email'), ['placeholder' => __('profile.email'), 'class' => 'form-control']) }}
              </div>
              <div class="input-group no-border input-lg">
                 <div class="input-group-prepend">
                    <span class="input-group-text"><i class="text_caps-small"></i></span>
                 </div>
                 {{ Form::password('password', ['placeholder' => __('profile.password'), 'class' => 'form-control']) }}
              </div>
            </div>
            <div class="card-footer text-center">
              <button type="submit" class="btn btn-primary btn-round btn-lg btn-block">{{__('auth.started')}}</button>
            </div>
            <div class="pull-left">
              <h6><a href="{{route('register')}}" class="link footer-link">{{__('auth.account')}}</a></h6>
            </div>
```

```
  <div class="pull-right">
                    <h6>
                        <a href="{{url('password/reset')}}" class="link footer-link">{{__('auth.help')}}</a>
                    </h6>
                </div>
            {{Form::close()}}
          </div>
        </div>
      </div>
  </div>
  <footer class="footer">
  </footer>
</div>

@endsection
```

- register.blade.php

Esta vista se ha creado para que un usuario pueda registrarse en la página.

Esta vista hereda de **@extends('layouts.app-verify')** .

La primera parte de esta vista es donde se muestran los mensajes.

```
<div class="page-header header-filter" filter-color="black">
  <div class="page-header-image" style="background-image:url(./img/bg18.jpg)"></div>
  <div class="content">
    <div class="container">
      <div class="row">
        <div class="col-md-5 ml-auto mr-auto">

            @if ($errors->any())
                <div class="alert alert-danger rounded"  id="alert" role="alert">
                  <div class="container">
                    <button type="button" class="close" data-dismiss="alert" aria-label="Close">
            <span aria-hidden="true">
              <i class="now-ui-icons ui-1_simple-remove"></i>
            </span>
                    </button>
                    <p><strong>Ups!</strong> {{__('auth.error')}}</p>
                    @foreach ($errors->all() as $error)
                        <p>{{ $error }}</p>
                    @endforeach
                  </div>
                </div>
            @endif
        </div>
```

A continuación se puede ver el formulario de registro creado para esta vista.

```
        <div class="col-md-4 mr-auto">
          <div class="card card-signup">
            <div class="card-body">
              <h4 class="card-title text-center">{{__('auth.register')}}</h4>
              {!! Form::open(['route'=>'register' , 'method'=>'post']) !!}
                @csrf
                <div class="input-group">
                  <div class="input-group-prepend">
                    <span class="input-group-text"><i class="users_circle-08"></i></span>
                  </div>
                  {{ Form::text('name', old('name'), ['placeholder' => __('profile.name'), 'class' => 'form-control']) }}
                </div>
                <div class="input-group">
                  <div class="input-group-prepend">
                    <span class="input-group-text"><i class="text_caps-small"></i></span>
                  </div>
                  {{ Form::text('lastname', old('lastname'), ['placeholder' => __('profile.last_name'), 'class' => 'form-control']) }}
                </div>
                <div class="input-group">
                  <div class="input-group-prepend">
                    <span class="input-group-text"><i class="text_caps-small"></i></span>
                  </div>
                  {{ Form::text('address', old('address'), ['placeholder' => __('profile.address'), 'class' => 'form-control']) }}
                </div>
                <div class="input-group">
                  <div class="input-group-prepend">
                    <span class="input-group-text"><i class="text_caps-small"></i></span>
                  </div>
                  {{ Form::text('phone', old('phone'), ['placeholder' => __('profile.phone'), 'class' => 'form-control']) }}
                </div>
                <div class="input-group">
                  <div class="input-group-prepend">
                    <span class="input-group-text"><i class="ui-1_email-85"></i></span>
                  </div>
                  {{ Form::email('email', old('email'), ['placeholder' => __('profile.email'), 'class' => 'form-control']) }}
                </div>
                <div class="input-group">
                  <div class="input-group-prepend">
                    <span class="input-group-text"><i class="ui-1_email-85"></i></span>
                  </div>
                  {{ Form::password('password', ['placeholder' => __('profile.password'), 'class' => 'form-control']) }}
                </div>
                <div class="input-group">
                  <div class="input-group-prepend">
                    <span class="input-group-text"><i class="ui-1_email-85"></i></span>
                  </div>
                  {{ Form::password('password_confirmation', ['placeholder' => __('profile.password_confirm'), 'class' =>
'form-control']) }}
                </div>
```

```
                    <div class="form-check">
                      <label class="form-check-label">
                        <input class="form-check-input" type="checkbox" name="conditions">

                        <span class="form-check-sign"></span>
                        {{__('auth.conditions')}}
                        <a type="button" class="" data-toggle="modal"
data-target="#exampleModal">{{__('auth.conditions2')}}</a>.
                      </label>
                    </div>
                    <div class="card-footer text-center">
                      <button class="btn btn-primary btn-round btn-lg">{{__('auth.started2')}}</button>
                    </div>

                  {!! Form::close() !!}
                </div>
              </div>
            </div>
          </div>
        </div>
```

Para esta vista se añade también un modal para mostrar los términos y condiciones.

```
<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel"
aria-hidden="true">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title text-dark" id="exampleModalLabel">{{__('auth.conditions3')}}</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        <p class="text-dark">Lorem ipsum dolor sit amet, consectetur adipisicing elit. A aliquam, cum cupiditate distinctio dolore
eaque facere, fugit illum impedit nemo officia, quam ratione soluta totam ullam vero voluptatibus? Ea, minus!</p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
      </div>
    </div>
  </div>
</div>
```

- register.blade.php

Esta vista se ha creado para que en el caso de que un usuario que no tenga verificado su correo se muestre esta vista indicando que debe de verificar su correo.

```
@extends('layouts.app-verify')

@section('content')
  <noscript>
                        <iframe    src="https://www.googletagmanager.com/ns.html?id=GTM-NKDMSK6"    height="0"    width="0"
style="display:none;visibility:hidden"></iframe>
  </noscript>

  <div class="page-header header-filter" filter-color="orange">
    <div class="page-header-image" style="background-image:url('{{asset('img/bg11.jpg')}}')"></div>
    <div class="container">
      <div class="row justify-content-center">
        <div class="col-md-4">
          <div class="info info-hover">
            <div class="icon icon-info icon-circle">
              <i class="now-ui-icons ui-1_email-85"></i>
            </div>
            <h4 class="info-title">{{__('auth.verify')}}</h4>
          </div>
        </div>
      </div>
      <div class="row justify-content-center">
        <div class="col-md-8">
          <div class="card p-1 mt-2">
            <div class="card-header text-black p-1"><h4>{{ __('auth.verify_title') }}</h4></div>

            <div class="card-body text-black">
              @if (session('resent'))
                <div class="alert alert-success" role="alert">
                  <p>{{ __('auth.mail') }}</p>
                </div>
              @endif
                    {{ __('auth.continue') }} <a class="text-info" href="{{ route('verification.resend') }}">{{ __('auth.continue_2')
}}</a>.
            </div>
          </div>
        </div>
      </div>
    </div>

  </div>
@endsection
```

- cities.blade.php

Se ha creado esta vista para incluir un mapa de Google donde se puede seleccionar una ciudad para mostrar las casas que están en dicha ciudad.

```
@extends('layouts.app-nav')
@section('content')
  <div class="blogs-5" style="background-image: url('{{asset('/img/aparts.jpg')}}');">
    <div class="container">
      <div class="row">
        <div class="col-md-10 ml-auto mr-auto">
          <div class="row justify-content-center">
            <h2 class="title text-light">{{__('apartments.cities')}}</h2>
          </div>

        </div>
      </div>
    </div>
  </div>
  <div class="container">
    <div class="row justify-content-center mt-5">
      <div class="col-md-10">
        <iframe class="borde shadow-lg"
src="https://www.google.com/maps/d/u/1/embed?mid=1PTfcZSQ0RahxXZR2__f9eHAD_CHHDcHr" width="900"
height="700"></iframe>
      </div>
    </div>
  </div>
  @include('partials.footer')
@endsection
```

- footer.blade.php

En algunas vistas de la parte principal de la página se incluye un footer sencillo.

```
<footer class="footer bg-primary mb-0 mt-5 text-light">

  <div class="row justify-content-center">
    <div class="col-md-10">
      <h5 class="text-center">Housingbook</h5>
      <hr>
    </div>

  </div>
  <p class="text-center">Trabajo fin de curso</p>
  <p class="text-center">Daniel Buendia - Roberto Gómez</p>
</footer>
```

- Dashboard

Para el dashboard o panel de control de usuarios se ha creado una serie de vistas que se comentan a continuación.

- index.blade.php

Para el index de esta parte se ha creado una vista que hereda de **@extends('layouts.app-dash')**.

En la primera parte de la vista se muestran unas cartas con información sobre el usuario autenticado.

```
@section('content')
  <div class="content">
    <div class="row">
      <div class="col-md-12 ml-auto mr-auto shadow p-2">
        <div class="row">
          <div class="col-lg-6 col-md-6 col-sm-6">
            <div class="card card-stats">
              <div class="card-body ">
                <div class="row">
                  <div class="col-5 col-md-4">
                    <div class="icon-big text-center icon-warning">
                      <i class="fa fa-home text-warning"></i>
                    </div>
                  </div>
                  <div class="col-7 col-md-8">
                    <div class="numbers">
                      <p class="card-category">{{__('dashboard.apartment')}}</p>
                      <p class="card-title">{{count(Auth()->user()->apartments)}}
                      <p>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
          <div class="col-lg-6 col-md-6 col-sm-6">
            <div class="card card-stats">
              <div class="card-body ">
                <div class="row">
                  <div class="col-5 col-md-4">
                    <div class="icon-big text-center icon-warning">
                      <i class="nc-icon nc-money-coins text-success"></i>
                    </div>
                  </div>
                  <div class="col-7 col-md-8">
                    <div class="numbers">
                      <p class="card-category">{{__('dashboard.totals')}}</p>
                      <p class="card-title">{{$totalEarnings}} &euro;
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
```

En esta vista se muestra un calendario mediante javascript donde se muestran las reservas de las casas del usuario autenticado. El html necesario para el calendario es el siguiente.

```
<h3 class="text-center text-primary justify-content-center text-center mx-auto d-block">{{__('dashboard.resume')}}</h3>

<div class="row">
  <div class="col-md-7 ml-auto mr-auto">
    <div class="card card-calendar">
      <div class="card-body ">
        <div id="fullCalendar" class="btns-calendar"></div>
      </div>
    </div>
  </div>
</div>
```

El script necesario para que se muestre el calendario y que funcione correctamente es el siguiente.

```
<script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.4.0.min.js"></script>
<script>
  $(document).onload(function() {

    let language = "{!! config('app.locale'); !!}";

    let colours = ['#093145','#107896','#829356','#BCA136','#C2571A','#9A2617'];

    let array = {!! json_encode($allDates) !!};

    let months, abrev, days, today;

    if(language == "es"){
      months = ['Enero','Febrero','Marzo','Abril','Mayo','Junio','Julio','Agosto','Septiembre','Octubre','Noviembre','Diciembre'];
      abrev = ['Ene','Feb','Mar','Abr','May','Jun','Jul','Ago','Sep','Oct','Nov','Dic'];
      days = ['Lun','Mar','Mie','Jue','Vie','Sáb','Dom'];
      today = "Hoy";
    } else {
      months = ['January','February','March','April','May','June','July','August','September','October','November','December'];
      abrev = ['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec'];
      days = ['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat'];
      today = "Today";
    }

    let getEvent = [];
    for(let i = 0, j = 0;i < array.length;i++,j++)
    {
      let title = "{!! __('dashboard.rentedTo') !!}: " + array[i].name.toString() + ", {!! __('dashboard.rented') !!}: " +
array[i].last_name.toString();
      let start = array[i].entry;
      let end = array[i].exit;
      let color;

      if(j == 6)
      {
        j=0;
      }
```

```
color = colours[j];

        let insertEvents = {};
        insertEvents =
          {
            title: title,
            start: start,
            end: end,
            color: color,
            textColor: 'white',
          },
        getEvent.push(insertEvents);
      }

      $('#fullCalendar').fullCalendar({
        editable: false,
        weekMode: 'liquid',
        weekends: true,
        events: getEvent,
        monthNames: months,
        monthNamesShort: abrev,
        dayNamesShort: days,
        buttonText: {
          today: today
        },
        eventRender: function(event, element) {
          let title = event.title.split(",");
          $(element).tooltip({
            title: title[0] + "<br>" + title[1],
            content: title[1],
            trigger: 'hover',
            html: true,
            placement: 'top',
            });
        },
      });
  });
</script>
```

- Vistas de las casas en dashboard

Se han creado una serie de vistas para las casas donde se pueden ver los detalles de una casa, formulario de creación de casas, edición, etc.

create.blade.php

Esta vista hereda **@extends('layouts.app-dash-apartment')** y se muestra el formulario de creación de una casa.

```
<div class="content p-5"> <h3 class="">{{__('profile.createapartment')}}</h3>
  <div class="row">
    <div class="col-md-6">
          {{Form::open(['method' => 'POST', 'id' => 'upload_form', 'action' => 'dashboard\ApartmentController@store','files' =>
true])}}
       <div class="card ">
          <div class="card-body ">
            <h4 class="card-title">{{__('form.general')}}</h4>
            <div class="row">
              <label class="col-sm-2 col-form-label">{{__('form.nameapartment')}}</label>
              <div class="col-sm-10">
                <div class="form-group">
                                        {{Form::text('name', old('name'),['placeholder' => __('form.placeholdername'), 'class' =>
'form-control','required'])}}
                </div>
              </div>
            </div>
            <div class="row">
              <label class="col-sm-2 col-form-label">{{__('form.address')}}</label>
              <div class="col-sm-10">
                <div class="form-group">
                                {{Form::text('address', old('address'),['placeholder' => __('form.placeholderaddress'), 'class' =>
'form-control','required'])}}
                </div>
              </div>
            </div>
            <div class="row">
              <label class="col-sm-2 col-form-label">{{__('form.shortapartment')}}</label>
              <div class="col-sm-10">
                <div class="form-group">
                    {{Form::text('short_description', old('short_description'),['placeholder' => __('form.placeholdershort'), 'class'
=> 'form-control','required'])}}
                </div>
              </div>
            </div>
            <div class="row">
              <label class="col-sm-2 col-form-label">{{__('form.descriptionapartment')}}</label>
              <div class="col-sm-10">
                <div class="form-group">
                    {{Form::textarea('description', old('description'),['placeholder' => __('form.placeholderdescription'), 'class' =>
'form-control','required'])}}
                </div>
              </div>
            </div>
```

```
            <div class="row">
              <label class="col-sm-2 col-form-label">{{__('form.price')}}</label>
              <div class="col-sm-10 checkbox-radios">
                <div class="form-group">
                  <label class="form-check-label">
                    <input type="number" name="price" class="form-control bg-warning" min="1">
                  </label>
                </div>
              </div>
            </div>
<div class="row">
  <label class="col-sm-2 col-form-label">{{__('form.services')}}</label>
  <div class="col-sm-10 checkbox-radios">
    @foreach($services as $service)
      <div class="form-check">
        <label class="form-check-label">
          <input class="form-check-input" name="services[]" type="checkbox" value="{{$service->id}}">
          <span class="form-check-sign bg-success"></span>
          {{__('form.' . $service->name)}} <i class="{{$service->icon}} bg-primary rounded-circle text-light p-2"></i>
        </label>
      </div>
    @endforeach
  </div>
</div>
<div class="row justify-content-center">
  <div class="col-md-4">
    <h4 class="card-title">{{__('form.city')}}</h4>
    <select class="form-control" data-size="7" name="city" data-style="btn btn-primary btn-round" required>
      <option disabled>{{__('form.city')}}</option>
      @foreach($cities as $city)
        <option value="{{$city->id}}">{{$city->name}}</option>
      @endforeach
    </select>
  </div>
  <div class="col-md-4">
    <h4 class="card-title">{{__('form.categories')}}</h4>
    <select class="form-control" name="category" data-style="btn btn-info btn-round" data-size="7" required>
      <option disabled>{{__('form.categories')}}</option>
      @foreach($categories as $category)
        <option value="{{$category->id}}">{{__('form.' . $category->name)}}</option>
      @endforeach
    </select>
  </div>

</div>
```

```
<br>
        <div class="row justify-content-center">

          <div class="col-md-3 col-sm-4">

            <div class="fileinput fileinput-new text-center" data-provides="fileinput">
              <div class="fileinput-new thumbnail img-circle">
                <img src="{{asset('assets/img/placeholder.jpg')}}" alt="...">
              </div>
              <div class="fileinput-preview fileinput-exists thumbnail img-circle"></div>
              <div>
        <span class="btn btn-round btn-rose btn-file">
          <span class="fileinput-new">{{__('apartments.addphoto')}}</span>
          <span class="fileinput-exists">{{__('apartments.changephoto')}}</span>
          {{Form::file('photos[]', ['id' => 'photos', 'multiple' => 'multiple'])}}
        </span>
                  <br />
                                <a href="#" class="btn btn-danger btn-round fileinput-exists" data-dismiss="fileinput"><i class="fa
fa-times"></i> Remove</a>
              </div>
            </div>
          </div>
        </div>
        <div class="row">
          <i class="mx-auto d-block justify-content-center">{{__('apartments.4photos')}}</i>
        </div>
      </div>
      <div class="card-footer ">
        <button type="submit" class="btn btn-info" style="width: 100%">{{__('form.submit')}}</button>
      </div>
    </div>
    {{Form::close()}}
</div>
```

La siguiente parte de la vista es donde se muestran los mensajes.

```
    <div class="col-md-6">
      @if(session('success'))
        <button type="button" aria-hidden="true" class="close" data-dismiss="alert" aria-label="Close">
          <i class="nc-icon nc-simple-remove"></i>
        </button>
        <div class="alert alert-info alert-dismissible fade show">
            <span>
            {{__(session('success'))}}
        </span>
        </div>
      @endif
      @if(session('error'))
        <div class="alert alert-danger alert-dismissible fade show">
          <button type="button" aria-hidden="true" class="close" data-dismiss="alert" aria-label="Close">
            <i class="nc-icon nc-simple-remove"></i>
          </button>
          <span>{{__('form.apartmentcreatefail')}}</span>
        </div>
        <div class="alert alert-danger alert-dismissible fade show">
          <button type="button" aria-hidden="true" class="close" data-dismiss="alert" aria-label="Close">
            <i class="nc-icon nc-simple-remove"></i>
          </button>
          <span> {{__(session('error'))}}</span>
        </div>
      @endif
      @if(count($errors) > 0)
        <div class="alert alert-danger alert-dismissible fade show">
          <button type="button" aria-hidden="true" class="close" data-dismiss="alert" aria-label="Close">
            <i class="nc-icon nc-simple-remove"></i>
          </button>
          <span>{{__('form.apartmentcreatefail')}}</span>
        </div>
        @foreach($errors->all() as $error)
          <div class="alert alert-danger alert-dismissible fade show">
            <button type="button" aria-hidden="true" class="close" data-dismiss="alert" aria-label="Close">
              <i class="nc-icon nc-simple-remove"></i>
            </button>
            <span>{{$error}}</span>
          </div>
        @endforeach
      @endif
    </div>
  </div>
</div>
```

edit.blade.php

Esta vista es similar a la anterior, se muestra un formulario con los mismos campos, los mensajes se muestran en la misma posición, pero cambia la declaración de los campos de los formularios.

```
{{Form::open(['id' => 'form_update',
            'action' => ['dashboard\ApartmentController@update',$apartment->id],
            'method' => 'PUT','files' => true])}}

 {{Form::text('name', old('name',$apartment->name),
            ['placeholder' => __('form.placeholdername'),
            'class' => 'form-control','required'])}}

{{Form::text('address', old('address',$apartment->address),
            ['placeholder' => __('form.placeholderaddress'),
             'class' => 'form-control','required'])}}

{{Form::text('short_description', old('short_description',$apartment->short_description),
            ['placeholder' => __('form.placeholdershort'),
             'class' => 'form-control','required'])}}

 {{Form::textarea('description', old('description',$apartment->description),
            ['placeholder' => __('form.placeholderdescription'),
            'class' => 'form-control','required'])}}
```

Se muestran los servicios que tenía la casa marcados.

```
        @foreach($services as $service)
          <div class="form-check">
            <label class="form-check-label">
              @foreach($apartmentServices as $apartmentService)
                <input class="form-check-input" name="services[]" type="checkbox" value="{{$service->id}}">
                @if($apartmentService->name == $service->name)
                  <span class="form-check-sign bg-success"></span>
                  <input class="form-check-input" name="services[]" type="checkbox" value="{{$service->id}}" checked>
                  @break
                @else
                  <input class="form-check-input" name="services[]" type="checkbox" value="{{$service->id}}">
                @endif
              @endforeach                                    <span class="form-check-sign bg-success"></span>
              {{__('form.' . $service->name)}} <i class="{{$service->icon}} bg-primary rounded-circle text-light p-2"></i>
            </label>
          </div>
        @endforeach
```

Se muestra la ciudad y categoría a la que pertenece la casa.

```
<div class="row justify-content-center">
        <div class="col-md-4">
            <h4 class="card-title">{{__('form.city')}}</h4>
            <select class="form-control" data-size="7" name="city" data-style="btn btn-primary btn-round" required>
                <option disabled>{{__('form.city')}}</option>
                @foreach($cities as $city)
                    @if($city->id == $apartment->city_id)
                        <option value="{{$city->id}}" selected>{{$city->name}}</option>
                    @else
                        <option value="{{$city->id}}">{{$city->name}}</option>
                    @endif
                @endforeach
            </select>
        </div>
        <div class="col-md-4">
            <h4 class="card-title">{{__('form.categories')}}</h4>
            <select class="form-control" name="category" data-style="btn btn-info btn-round" data-size="7" required>
                <option disabled>{{__('form.categories')}}</option>
                @foreach($categories as $category)
                    @if($category->id == $apartment->category_id)
                        <option value="{{$category->id}}" selected>{{__('form.' . $category->name)}}</option>
                    @else
                        <option value="{{$category->id}}">{{__('form.' . $category->name)}}</option>
                    @endif
                @endforeach
            </select>
        </div>

    </div>
```

index.blade.php

Esta vista muestra una lista de todas las casas de un usuario con los botones necesarios par realizar diferentes acciones, editar, ver, borrar, etc.
De esta vista se destaca una tabla donde se muestran las casas.

```
<div class="content p-5">
  <div class="row">
    <div class="col-md-12">
      <div class="card">
        <div class="card-header">
          <h2 class="card-title">{{__('profile.apartmenthello')}}</h2>
        </div>
        <div class="card-body">
          <div class="table-responsive">
            <table class="table">
              <thead class="text-primary">
              <th>{{__('profile.apartmentname')}}</th>
              <th>{{__('profile.apartmentcity')}}</th>
              <th>{{__('form.price')}}</th>
              <th>{{__('profile.apartmentaddress')}}</th>

              <th class="text-center">{{__('form.actions')}}</th>
              </thead>
              <tbody>
              @forelse($apartments as $apartment)
                <tr>
                  <td>{{$apartment->name}}</td>
                  <td>{{$apartment->city->name}}</td>
                  <td>{{$apartment->price}}€</td>
                  <td>{{$apartment->address}}</td>
                  <td class="text-center text-light">
                    <a href="{{$apartment->id}}"
                      data-toggle="tooltip" title="{{__('profile.show')}}"
                      class="btn btn-info btn-icon btn-sm show">
                       <i class="fa fa-user"></i>
                    </a>
                    <a href="{{url('/dashboard/apartment/' . $apartment->id .'/edit')}}"
                      class="btn btn-success btn-icon btn-sm edit"
                      data-toggle="tooltip" title="{{__('profile.edit')}}">
                       <i class="fa fa-edit"></i>
                    </a>
                    <button data-toggle="modal" data-target="#photo" data-id="{{$apartment->id}}" class="photoApartment
btn btn-warning btn-icon btn-sm"><i class="fa fa-photo"></i></button>
                    <button data-toggle="modal" data-target="#borrar" data-id="{{$apartment->id}}" class="Apartment btn
btn-danger btn-icon btn-sm"><i class="fa fa-times"></i></button>
                  </td>
                </tr>
              @empty
                <h3>{{__('profile.apartmentempty')}}</h3>
              @endforelse
              </tbody>
            </table>
          </div>
```

Esta vista dispone de dos modales. Uno de los modales se muestra cuando se va a borrar las fotos de un apartamento y el otro se muestra cuando se va a borrar un apartamento.

```
<div id="photo" class="modal fade" role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header" style="height:20px;">
        <button type="button" class="close" data-dismiss="modal">&times;</button>
      </div>
      <div class="modal-body">
        <p>{{__('apartments.photodelete')}}</p>
      </div>
      <div class="modal-footer" style="margin-right: 10px">
        {{Form::open(['action' => ['dashboard\ApartmentController@photodestroy'], 'method' => 'DELETE', 'class' =>
['d-inline-block']])}}
        <input type="hidden" name="idHolder" id="idHolder">
        <input type="submit" class="btn btn-danger" value="{{__('apartments.accept')}}" name="{{__('apartments.accept')}}">
        {{Form::close()}}
        <button type="button" class="btn btn-primary" data-dismiss="modal">{{__('apartments.cancel')}}</button>
      </div>
    </div>

  </div>
</div>
```

```
<div id="borrar" class="modal fade" role="dialog">
  <div class="modal-dialog">

    <!-- Modal content-->
    <div class="modal-content">
      <div class="modal-header" style="height:20px;">
        <button type="button" class="close" data-dismiss="modal">&times;</button>
      </div>
      <div class="modal-body">
        <p>{{__('apartments.apartmentdelete')}}</p>
      </div>
      <div class="modal-footer" style="margin-right: 10px">
        {{Form::open(['action' => ['dashboard\ApartmentController@destroy'], 'method' => 'DELETE', 'class' =>
['d-inline-block']])}}
        <input type="submit" class="btn btn-danger" value="{{__('apartments.accept')}}" name="{{__('apartments.accept')}}">
        <input type="hidden" name="idHolder" id="idHolder2">
        {{Form::close()}}
        <button type="button" class="btn btn-primary" data-dismiss="modal">{{__('apartments.cancel')}}</button>
      </div>
    </div>

  </div>
</div>
```

Para los botones que se muestran en esta vista se añaden eventos para realizar peticiones ajax.

```
<script>

  $(".show").each(function() {
     $(this).click(function (event) {

        if(event.currentTarget.href !== undefined)
        {
           event.preventDefault();

           let indice = event.currentTarget.href.split("/");

           $.ajax(
              {
                 url: "/dashboard/apartment/show/" + indice[3],
                 type: 'GET',
              }).done(

              function(data)
              {
                 $('#ajaxviews').html(data.html);
              }
           );
        }

     });
  });

  $(document).on("click", ".photoApartment", function () {
     var eventId = $(this).data('id');
     $('#idHolder').val( eventId );
  });

  $(document).on("click", ".Apartment", function () {
     var eventId = $(this).data('id');
     $('#idHolder2').val( eventId );
  });

</script>
```

show.blade.php

Para mostrar un apartamento en concreto con toda su información se ha creado esta vista. **Esta vista mediante ajax**.

```
<div class="wrapper">
  <div class="section">
    <div class="container">
      <div class="row">
        <div class="col-md-5">
          <div id="productCarousel" class="carousel slide" data-ride="carousel" data-interval="8000">
            <ol class="carousel-indicators">
              <li data-target="#productCarousel" data-slide-to="0" class="active"></li>
              <li data-target="#productCarousel" data-slide-to="1"></li>
              <li data-target="#productCarousel" data-slide-to="2"></li>
              <li data-target="#productCarousel" data-slide-to="3"></li>
            </ol>
            <div class="carousel-inner" role="listbox">
              @if(count($apartment->photos))
                <div class="carousel-item active">
                  <img class="d-block img-raised" src="{{url('/storage/photos/'.$apartment->photos[0]->local_url)}}"
alt="{{$apartment->name}}">
                </div>
                <div class="carousel-item">
                  <img class="d-block img-raised" src="{{url('/storage/photos/'.$apartment->photos[1]->local_url)}}"
alt="{{$apartment->name}}">
                </div>
                <div class="carousel-item">
                  <img class="d-block img-raised" src="{{url('/storage/photos/'.$apartment->photos[2]->local_url)}}"
alt="{{$apartment->name}}">
                </div>
                <div class="carousel-item">
                  <img class="d-block img-raised" src="{{url('/storage/photos/'.$apartment->photos[3]->local_url)}}"
alt="{{$apartment->name}}">
                </div>
              @else
                <div class="carousel-item active">
                  <img class="d-block img-raised" src="{{ asset('img/notfound.png')}}" width="600" height="350"
alt="{{$apartment->name}}">
                </div>
                <div class="carousel-item">
                  <img class="d-block img-raised" src="{{ asset('img/notfound.png')}}" width="600" height="350"
alt="{{$apartment->name}}">
                </div>
                <div class="carousel-item">
                  <img class="d-block img-raised" src="{{ asset('img/notfound.png')}}" width="600" height="350"
alt="{{$apartment->name}}">
                </div>
                <div class="carousel-item">
                  <img class="d-block img-raised" src="{{ asset('img/notfound.png')}}" width="600" height="350"
alt="{{$apartment->name}}">
                </div>
              @endif
            </div>
```

```
                <a class="carousel-control-prev" href="#productCarousel" role="button" data-slide="prev">
                  <button type="button" class="btn btn-primary btn-icon btn-round btn-sm" name="button">
                    <i class="now-ui-icons arrows-1_minimal-left"></i>
                  </button>
                </a>
                <a class="carousel-control-next" href="#productCarousel" role="button" data-slide="next">
                  <button type="button" class="btn btn-primary btn-icon btn-round btn-sm" name="button">
                    <i class="now-ui-icons arrows-1_minimal-right"></i>
                  </button>
                </a>
              </div>
              <p class="blockquote blockquote-primary">
                {{$apartment->short_description}}
                <br>
                <br>
                <small>{{$apartment->user->name}}</small>
              </p>
            </div>
            <div class="col-md-6 ml-auto mr-auto">
              <h2 class="title"> {{$apartment->name}} </h2>
              <h5 class="category">{{$apartment->address}}</h5>
              <h5 class="main-price">{{$apartment->city->name}}</h5>
              <div id="accordion" role="tablist" aria-multiselectable="true" class="card-collapse">
                <div class="card card-plain">
                  <div class="card-header" role="tab" id="headingOne">
                    <a data-toggle="collapse" data-parent="#accordion" href="#collapseOne" aria-expanded="true"
aria-controls="collapseOne">
                      {{__('guest.description')}}
                      <i class="now-ui-icons arrows-1_minimal-down"></i>
                    </a>
                  </div>
                  <div id="collapseOne" class="collapse show" role="tabpanel" aria-labelledby="headingOne">
                    <div class="card-body">
                      <p>{{$apartment->description}}</p>
                    </div>
                  </div>
                </div>
```

```
                <div class="card card-plain">
                  <div class="card-header" role="tab" id="headingTwo">
                    <a class="collapsed" data-toggle="collapse" data-parent="#accordion" href="#collapseTwo"
aria-expanded="false" aria-controls="collapseTwo">
                      {{__('guest.owner')}}
                      <i class="now-ui-icons arrows-1_minimal-down"></i>
                    </a>
                  </div>
                  <div id="collapseTwo" class="collapse" role="tabpanel" aria-labelledby="headingTwo">
                    <div class="card-body">
                      <p>{{$apartment->user->name}}</p>
                    </div>
                  </div>
                </div>
                <div class="card card-plain">
                  <div class="card-header" role="tab" id="headingThree">
                    <a class="collapsed" data-toggle="collapse" data-parent="#accordion" href="#collapseThree"
aria-expanded="false" aria-controls="collapseThree">
                      {{__('guest.services')}}
                      <i class="now-ui-icons arrows-1_minimal-down"></i>
                    </a>
                  </div>
                  <div id="collapseThree" class="collapse" role="tabpanel" aria-labelledby="headingThree">
                    <div class="card-body">
                      <ul>
                        @forelse($apartment->services as $service)
                          <li>{{__('form.' . $service->name)}}</li>
                        @empty
                          <p>{{__('guest.noservices')}}</p>
                        @endforelse
                      </ul>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

- Vistas de usuarios en dashboard

A continuación se muestran las vistas relacionadas con los usuarios en la páginas (vistas donde los usuarios pueden editar su información).

edit.blade.php

Esta vista es donde un usuario puede editar su información de perfil.
Dicha vista hereda de **@extends('layouts.app-dash-apartment')** y su contenido es el siguiente.

En primera parte se muestran los mensajes.

```
<div class="row justify-content-center">
  @if(session('success'))
    <div class="col-md-6">
      <div class="alert alert-info alert-dismissible fade show">
        <button type="button" aria-hidden="true" class="close" data-dismiss="alert" aria-label="Close">
          <i class="nc-icon nc-simple-remove"></i>
        </button>
        <span>
          {{__('profile.profileupdatedocorrectly')}}
      </span>
      </div>
    </div>
  @endif
  @if(session('error'))
    <div class="col-md-6">
      <div class="alert alert-danger alert-dismissible fade show">
        <button type="button" aria-hidden="true" class="close" data-dismiss="alert" aria-label="Close">
          <i class="nc-icon nc-simple-remove"></i>
        </button>
        <span>
          {{__(session('error'))}}
      </span>
      </div>
    </div>
  @endif
  @if(count($errors) > 0)
    <div class="col-md-6">
      <div class="alert alert-danger alert-dismissible fade show">
        <button type="button" aria-hidden="true" class="close" data-dismiss="alert" aria-label="Close">
          <i class="nc-icon nc-simple-remove"></i>
        </button>
        <span>
          {{__('profile.profileupdatedfailed')}}
      </span>
      </div>
      @foreach($errors->all() as $error)
        <div class="callout alert alert-danger">
          {{$error}}
        </div>
      @endforeach
    </div>
  @endif
</div>
```

La segunda parte de esta vista y más importante es el formulario para editar toda su información.

```
<div class="row justify-content-center p-5">

  <div class="col-md-6">
    <div class="card card-user">
      <div class="card-header">
        <h5 class="title text-center">{{__('profile.edit_profile')}}</h5>
      </div>
      <hr>
      <div class="card-body ">
        {{Form::open(['method' => 'PUT','action' => ['dashboard\UserController@update',auth()->user()->id],'files' => true])}}
        <div class="row">
          <label class="col-md-3 col-form-label">{{__('profile.name')}}</label>

          <div class="col-md-8">
            <div class="form-group">
              {{Form::text('name',old('name',Auth()->user()->name), ['id' => 'name','class' => 'form-control'])}}
            </div>
          </div>
        </div>
        <div class="row">
          <label class="col-md-3 col-form-label">{{__('profile.last_name')}}</label>
          <div class="col-md-8">
            <div class="form-group">
              {{Form::text('last_name',old('last_name',Auth()->user()->last_name), ['id' => 'last_name','class' => 'form-control'])}}
            </div>
          </div>
        </div>
        <div class="row">
          <label class="col-md-3 col-form-label">{{__('profile.email')}}</label>
          <div class="col-md-8">
            <div class="form-group">
              {{Form::text('email',old('email', Auth()->user()->email), ['id' => 'email','class' => 'form-control'])}}
            </div>
          </div>
        </div>
        <div class="row">
          <label class="col-md-3 col-form-label">{{__('profile.apartmentaddress')}}</label>
          <div class="col-md-8">
            <div class="form-group">
              {{Form::text('address',old('address',Auth()->user()->address), ['id' => 'address','class' => 'form-control'])}}
            </div>
          </div>
        </div>
        <div class="row">
          <label class="col-md-3 col-form-label">{{__('profile.secondaddress')}}</label>
          <div class="col-md-8">
            <div class="form-group">
              {{Form::text('second_address',old('second_address',Auth()->user()->second_address), ['id' =>
'second_address','class' => 'form-control'])}}
            </div>
          </div>
        </div>
```

```
            <div class="row">
              <label class="col-md-3 col-form-label">{{__('profile.phone')}}</label>
              <div class="col-md-8">
                <div class="form-group">
                  <input type="tel" class="form-control" value="{{ old('phone') ?: Auth()->user()->phone }}" name="phone"
id="phone">
                </div>
              </div>
            </div>
            <div class="row justify-content-center">

              <div class="col-md-3 col-sm-4">

                <div class="fileinput fileinput-new text-center" data-provides="fileinput">
                  <div class="fileinput-new thumbnail img-circle">
                    <img src="{{asset('assets/img/placeholder.jpg')}}" alt="...">
                  </div>
                  <div class="fileinput-preview fileinput-exists thumbnail img-circle"></div>
                  <div>
            <span class="btn btn-round btn-rose btn-file">
              <span class="fileinput-new">Add Photo</span>
              <span class="fileinput-exists">Change</span>
                {{Form::file('photo')}}
            </span>
                    <br />
                    <a href="#" class="btn btn-danger btn-round fileinput-exists" data-dismiss="fileinput"><i class="fa
fa-times"></i> Remove</a>
                  </div>
                </div>
              </div>
            </div>
            <hr>
            <div class="row justify-content-center">
              <div class="col-md-3 col-sm-4">
                {{Form::submit(__('profile.update'),['class' => 'btn btn-primary'])}}
              </div>
            </div>
            {{Form::close()}}
          </div>
        </div>
      </div>
    </div>
```

password.blade.php

La siguiente vista se utiliza para que un usuario pueda actualizar su contraseña.

```
<div class="content">
  <div class="row p-5 justify-content-center">
    <div class="col-md-1">

    </div>
    <div class="col-md-6">
      <div class="card card-user">
        <div class="card-header">
          <h5 class="title text-center">{{__('profile.edit_password')}}</h5>
        </div>
        <hr>
        <div class="card-body ">
          {{Form::open(['method' => 'PUT','action' => ['dashboard\UserController@passwordUpdate',auth()->user()->id]])}}
          <div class="row">
            <label class="col-md-3 col-form-label">{{__('profile.password')}}</label>

            <div class="col-md-8">
              <div class="form-group">
                {{Form::password('password', ['id' => 'password','class' => 'form-control'])}}
              </div>
            </div>
          </div>
          <div class="row">
            <label class="col-md-3 col-form-label">{{__('profile.password_confirm')}}</label>
            <div class="col-md-8">
              <div class="form-group">
                {{Form::password('password_confirmation',['id' => 'password_confirmation','class' => 'form-control'])}}
              </div>
            </div>
          </div>
          <div class="row justify-content-center">
            <div class="col-md-3 col-sm-4">
              {{Form::submit(__('profile.update'),['class' => 'btn btn-primary'])}}
            </div>
          </div>
          {{Form::close()}}
        </div>
      </div>
    </div>
```

En esta vista los mensajes de error se muestran en la parte inferior del formulario.

```
    @if(session('success'))
      <div class="col-md-5">
        <div class="alert alert-info alert-dismissible fade show">
          <button type="button" aria-hidden="true" class="close" data-dismiss="alert" aria-label="Close">
            <i class="nc-icon nc-simple-remove"></i>
          </button>
          <span>
            {{__('profile.passwordupdatedcorrectly')}}
        </span>
        </div>
      </div>
    @endif
    @if(count($errors) > 0)
      <div class="col-md-5">
        <div class="alert alert-danger alert-dismissible fade show">
          <button type="button" aria-hidden="true" class="close" data-dismiss="alert" aria-label="Close">
            <i class="nc-icon nc-simple-remove"></i>
          </button>
          <span>
            {{__('profile.passwordupdatedfailed')}}
        </span>
        </div>
        @foreach($errors->all() as $error)
          <div class="callout alert alert-danger">
            {{$error}}
          </div>
        @endforeach
      </div>

    @endif
  </div>
</div>
```

show.blade.php

Esta vista simplemente muestra toda la información del usuario autenticado. **Esta vista se carga mediante ajax.**

```
<div class="col-md-4">
    <div class="card card-user">
        <div class="image">
            <img src="{{asset('img/bg19.jpg')}}" alt="...">
        </div>
        <div class="card-body">
            <div class="author">
                <img class="avatar border-gray" src="{{url('/storage/photos/'.auth()->user()->photo)}}"
alt="{{auth()->user()->name}}">
                <h5 class="title">{{Auth()->user()->name}}</h5>
                <p class="description">
                    {{Auth()->user()->last_name}}
                </p>
            </div>

        </div>
        <div class="card-footer">
            <hr>
            <div class="button-container">
                <div class="row">
                    <div class="col-lg-3 col-md-6 col-6 ml-auto">

                    </div>
                    <div class="col-lg-4 col-md-6 col-6 ml-auto mr-auto">
                        <h5>{{count(Auth()->user()->apartments)}}
                            <br>
                            <small>{{__('menu.apartment')}}</small>
                        </h5>
                    </div>
                    <div class="col-lg-3 mr-auto">

                    </div>
                </div>
            </div>
        </div>
    </div>

</div>
```

```
<div class="col-md-8">
  <div class="card">
    <div class="card-header">
      <h5 class="title">{{__('profile.profile')}}</h5>
    </div>
    <div class="card-body">
      <form>
        <div class="row">
          <div class="col-md-5 pr-1">
            <div class="form-group">
              <label>{{__('profile.email')}}</label>
              <input type="text" class="form-control" disabled="" value="{{Auth()->user()->email}}">
            </div>
          </div>
          <div class="col-md-3 px-1">
            <div class="form-group">
              <label>{{__('profile.name')}}</label>
              <input type="text" class="form-control" disabled=""  value="{{Auth()->user()->name}}">
            </div>
          </div>
          <div class="col-md-4 pl-1">
            <div class="form-group">
              <label for="exampleInputEmail1">{{__('profile.last_name')}}</label>
              <input type="email" class="form-control" disabled="" value="{{Auth()->user()->last_name}}">
            </div>
          </div>
        </div>

        <div class="row">
          <div class="col-md-12">
            <div class="form-group">
              <label>{{__('profile.apartmentaddress')}}</label>
              <input type="text" class="form-control" disabled="" value="{{Auth()->user()->address}}">
            </div>
          </div>
        </div>
        <div class="row">
          <div class="col-md-12">
            <div class="form-group">
              <label>{{__('profile.secondaddress')}}</label>
              <input type="text" class="form-control" disabled="" value="{{Auth()->user()->second_address}}">
            </div>
          </div>
        </div>
        <div class="row">
          <div class="col-md-3">
            <div class="form-group">
              <label>{{__('profile.phone')}}</label>
              <input type="text" class="form-control" disabled=""  value="{{Auth()->user()->phone}}">
            </div>
          </div>
          <div class="col-md-7"></div>
        </div>

      </form>
    </div>
  </div>
</div>
```

telegram.blade.php

Esta vista se ha creado para que un usuario pueda ver su id de telegram y también actualizarla en el caso de que quisiera hacerlo.
Esta vista hereda de **@extends('layouts.app-dash-apartment')** .

Como primera parte de esta vista mostramos los mensajes.

```
<div class="row justify-content-center">

  @if(session('success'))
    <div class="col-md-8">
      <div class="alert alert-info alert-dismissible fade show">
        <button type="button" aria-hidden="true" class="close" data-dismiss="alert" aria-label="Close">
          <i class="nc-icon nc-simple-remove"></i>
        </button>
        <span>
          {{__('profile.telegramupdatedcorrectly')}}
      </span>
      </div>
    </div>
  @endif
  @if(count($errors) > 0)
    <div class="col-md-8">
      <div class="alert alert-danger alert-dismissible fade show">
        <button type="button" aria-hidden="true" class="close" data-dismiss="alert" aria-label="Close">
          <i class="nc-icon nc-simple-remove"></i>
        </button>
        <span>
          {{__('profile.telegramupdatedfailed')}}
      </span>
      </div>
      @foreach($errors->all() as $error)
        <div class="callout alert alert-danger">
          {{$error}}
        </div>
      @endforeach
    </div>
  @endif
</div>
```

```
<div class="row justify-content-center">
  <div class="col-md-8">
    <div class="card card-user">
      <div class="card-header">
        <h5 class="title text-center">{{__('profile.edit_telegram')}}</h5>
        <div class="bg-warning rounded">
          <p class="text-center text-white p-2">
            <strong>{{__('telegram.note')}}</strong>{{__('telegram.one')}} <strong>{{__('telegram.two')}}<span class="bg-light
text-warning p-1 rounded"><a
                        href="https://web.telegram.org/#/im?p=@housingbook_bot" target="_blank">@housing_bot </a></span>
{{__('telegram.three')}}</strong>."
          </p>
        </div>
        <div>
          <p class="text-center">
            <strong>
              <a href="{{url('dashboard/user/telegram/tutorial')}}" class="mx-auto d-block"
target="_blank">{{__('profile.obtain_telegram')}}</a>
            </strong>
          </p>
        </div>
        <div class="row justify-content-center">
        </div>
        <hr>
        <div class="card-body ">
          {{Form::open(['method' => 'PUT','action' => ['dashboard\UserController@telegramUpdate', auth()->user()->id]])}}
          <div class="row">
            <label class="col-md-3 col-form-label">{{__('profile.telegram')}}</label>

            <div class="col-md-8">
              <div class="form-group">
                {{Form::text('telegram',old('name',Auth()->user()->telegram), ['id' => 'name','class' => 'form-control'])}}
              </div>
            </div>
          </div>
          <div class="row justify-content-center">
            <div class="col-md-3 col-sm-4">
              {{Form::submit(__('profile.update'),['class' => 'btn btn-primary'])}}
            </div>
          </div>
          {{Form::close()}}
        </div>
      </div>
    </div>
  </div>
</div>
```

- Vistas para emails

Se han creado dos vistas para el envío de email cuando se completa el proceso de reserva de una casa. Este proceso culmina con envío de dos correos, uno para el dueño de la casa y otro para el usuario que alquila.

- new_rent.blade.php

Esta vista es la para el envío de correo al dueño de la casa que se alquila.
Se añade los siguientes estilos en línea.

```
<style media="screen">
  *{
     font-family: arial;
     color: #474747;
  }
  body{
     background-color: #e2e2e2;
  }
  ul{
     list-style-type: none;
  }

  .card{
     background-color: white;
     padding: 5px;
     box-shadow: -1px 2px 7px 3px #878787;
     border-radius: 5px;
     position:absolute;
     left:25%;
     width: 50%;
  }

  .card-header{
     padding: 5px;

  }
  .line-header{
     border-bottom: 1px solid #e2e2e2;
     widh: 100px;
     text-align: center;

  }
  h3{
     text-align: center;
  }
  .card-body{
     padding-left: 10px;
  }
```

```
  .total{
     border: 1px solid #b2b2b2;
     border-radius: 3px;
     padding: 5px;
     padding-left: 10px;
     margin-bottom: 10px;
     box-shadow: -1px 1px 3px 1px #b2b2b2;
  }
  span{
     background-color: #d84949;
     padding: 5px;
     border-radius: 5px;
     color: white;
  }

  a{
     color: #0194a8;
     text-decoration: none;
  }
  .centrado{
     text-align: center;
  }
  img{
     width: 100%;
     heidht: 100%;
  }
</style>
```

El html introducido en el **<body>** es el siguiente.

```
<div class="card">
  <div class="card-header">
    <h3>{{__('mail.rentsuccess')}}</h3>
    <div class="line-header"></div>
    <img src="www.housingbook.es/img/mail-template.jpg"></img>
  </div>
  <div class="card-body">
    <h4>{{__('mail.customerdata')}}</h4>
    <ul>
      <li><strong>{{__('mail.customername')}}:</strong> {{$user->name}}</li>
      <li><strong>{{__('mail.customerlast_name')}}:</strong> {{$user->last_name}}</li>
      <li><strong>{{__('mail.customeraddress')}}:</strong> {{$user->address}}</li>
      <li></li>
      <li><strong>{{__('mail.customeremail')}}:</strong> {{$user->email}}</li>
      <li><strong>{{__('mail.customerphone')}}:</strong> {{$user->phone}}</li>
    </ul>
    <h4>{{__('mail.ownerdata')}}:</h4>
    <ul>
      <li><strong>{{__('mail.ownername')}}:</strong> {{$apartment->user->name}}</li>
      <li><strong>{{__('mail.ownerlast_name')}}:</strong> {{$apartment->user->last_name}}</li>
      <li></li>
      <li><strong>{{__('mail.owneremail')}}:</strong> {{$apartment->user->email}}</li>
      <li><strong>{{__('mail.ownerphone')}}:</strong> {{$apartment->user->phone}}</li>
    </ul>
    <div class="line-header"></div>
    <h4>{{__('mail.rentdata')}}</h4>
    <ul>
      <li><strong>{{__('mail.apartmentname')}}:</strong> {{$apartment->name}}</li>
      <li><strong>{{__('mail.apartmentaddress')}}:</strong> {{$apartment->address}}</li>
      <li><strong>{{__('mail.apartmentphone')}}:</strong> {{$apartment->phone}}</li>

    </ul>
    <div class="total">
      <p><strong>{{__('mail.apartmentduration')}}:</strong> {{session('entradaEstancia') . " -- " . session('salidaEstancia')}}</p>
      <p><strong>{{__('mail.apartmentprice')}}:</strong><span> {{$apartment->price * session('days')}} &euro;</span></p>
    </div>
    <div class="line-header"></div>
    <p class="centrado"> {{__('mail.generaldoubts')}}:</p>
    <p class="centrado"><a href="#">admin@housingbook.es</a></p>
  </div>
</div>
```

- successfully_rent.blade.php

Esta vista es la para el envío de correo al dueño de la casa que se alquila.
Se añade los siguientes estilos en línea.

```
<style media="screen">
  *{
     font-family: arial;
     color: #474747;
  }
  body{
     background-color: #e2e2e2;
  }
  ul{
     list-style-type: none;
  }

  .card{
     background-color: white;
     padding: 5px;
     box-shadow: -1px 2px 7px 3px #878787;
     border-radius: 5px;
     position:absolute;
     left:25%;
     width: 50%;
  }

  .card-header{
     padding: 5px;

  }
  .line-header{
     border-bottom: 1px solid #e2e2e2;
     widh: 100px;
     text-align: center;

  }
  h3{
     text-align: center;
  }
  .card-body{
     padding-left: 10px;
  }
```

```
  .total{
     border: 1px solid #b2b2b2;
     border-radius: 3px;
     padding: 5px;
     padding-left: 10px;
     margin-bottom: 10px;
     box-shadow: -1px 1px 3px 1px #b2b2b2;
  }
  span{
     background-color: #d84949;
     padding: 5px;
     border-radius: 5px;
     color: white;
  }

  a{
     color: #0194a8;
     text-decoration: none;
  }
  .centrado{
     text-align: center;
  }
  img{
     width: 100%;
     heidht: 100%;
  }
</style>
```

El html introducido en el **<body>** es el siguiente.

```
<div class="card">
  <div class="card-header">
    <h3>{{__('mail.rentsuccess')}}</h3>
    <div class="line-header"></div>
    <img src="www.housingbook.es/img/mail-template.jpg"></img>
  </div>
  <div class="card-body">
    <h4>{{__('mail.customerdata')}}</h4>
    <ul>
      <li><strong>{{__('mail.customername')}}:</strong> {{$user->name}}</li>
      <li><strong>{{__('mail.customerlast_name')}}:</strong> {{$user->last_name}}</li>
      <li><strong>{{__('mail.customeraddress')}}:</strong> {{$user->address}}</li>
      <li></li>
      <li><strong>{{__('mail.customeremail')}}:</strong> {{$user->email}}</li>
      <li><strong>{{__('mail.customerphone')}}:</strong> {{$user->phone}}</li>
    </ul>
    <h4>{{__('mail.ownerdata')}}:</h4>
    <ul>
      <li><strong>{{__('mail.ownername')}}:</strong> {{$apartment->user->name}}</li>
      <li><strong>{{__('mail.ownerlast_name')}}:</strong> {{$apartment->user->last_name}}</li>
      <li></li>
      <li><strong>{{__('mail.owneremail')}}:</strong> {{$apartment->user->email}}</li>
      <li><strong>{{__('mail.ownerphone')}}:</strong> {{$apartment->user->phone}}</li>
    </ul>
    <div class="line-header"></div>
    <h4>{{__('mail.rentdata')}}</h4>
    <ul>
      <li><strong>{{__('mail.apartmentname')}}:</strong> {{$apartment->name}}</li>
      <li><strong>{{__('mail.apartmentaddress')}}:</strong> {{$apartment->address}}</li>
      <li><strong>{{__('mail.apartmentphone')}}:</strong> {{$apartment->phone}}</li>

    </ul>
    <div class="total">
      <p><strong>{{__('mail.apartmentduration')}}:</strong> {{session('entradaEstancia') . " -- " . session('salidaEstancia')}}</p>
      <p><strong>{{__('mail.apartmentprice')}}:</strong><span> {{$apartment->price * session('days')}} &euro;</span></p>
    </div>
    <div class="line-header"></div>
    <p class="centrado"> {{__('mail.generaldoubts')}}:</p>
    <p class="centrado"><a href="#">admin@housingbook.es</a></p>
  </div>
</div>
```

- email.blade.php

Por defecto, para la verificación del email se crea esta vista que nos la proporciona laravel.

```
@component('mail::message')
{{-- Greeting --}}
@if (! empty($greeting))
# {{ $greeting }}
@else
@if ($level === 'error')
# @lang('Whoops!')
@else
# @lang('Hello!')
@endif
@endif
{{-- Intro Lines --}}
@foreach ($introLines as $line)
{{ $line }}
@endforeach
{{-- Action Button --}}
@isset($actionText)
<?php
  switch ($level) {
      case 'success':
      case 'error':
          $color = $level;
          break;
      default:
          $color = 'primary';
  }
?>
@component('mail::button', ['url' => str_replace('http://localhost',config('app.url'),$actionUrl), 'color' => $color])
{{ $actionText }}
@endcomponent
@endisset
@foreach ($outroLines as $line)
{{ $line }}
@endforeach
@if (! empty($salutation))
{{ $salutation }}
@else
@lang('Regards'),<br>{{ config('app.name') }}
@endif
@isset($actionText)
@slot('subcopy')
@lang(
  "If you're having trouble clicking the \":actionText\" button, copy and paste the URL below\n".
  'into your web browser: [:actionURL](:actionURL)',
  [
      'actionText' => $actionText,
      'actionURL' => str_replace('http://localhost',config('app.url'),$actionUrl),
  ]
)
@endslot
@endisset
@endcomponent
```

- Generación de pdf

Se han creado dos vistas para la generación de pdf´s. En ellas se especifica el estilo en línea.
La generación de pd´f se realiza de dos maneras, una de ellas es la descarga de un fichero “.pdf” y la otras se genera un pdf en línea que se abre en el navegador.

Esto es un ejemplo de como se descarga un pdf.

```
$pdf = App::make('dompdf.wrapper');
$pdf = Facade::loadView('pdf.invoice', compact('invoice', 'days'));
return $pdf->download('invoice-'. $invoice->pivot->entry .''. $invoice->pivot->exit .'.pdf');
```

Y este es otro ejemplo de generar un pdf en línea.

```
$pdf = App::make('dompdf.wrapper');
  $pdf = Facade::loadView('pdf.tutorial');
  return $pdf->stream();
```

- invoice.blade.php

Esta vista genera una factura a partir de los datos facilitados por el controlador. Se crean estilos en línea.

```
<style>
  *{    color: #6b6b6b;   }
  .derecha{
     color: #f48942;
     font-size: 40px;
     opacity: 0.5;
  }
  .tabla{      width: 100%;   }
  .datos{      font-size: 12px;   }
  .cabecera{     background-color: #f48942;   }
  .footer {
     position: fixed;
     left: 0;
     bottom: 5%;
     width: 100%;
     background-color: #f48942;
     color: white;
     text-align: center;
     border-radius: 5px;
     box-shadow: 10px 10px 8px -6px rgba(0,0,0,0.75);
  }
  .titulo{
     color: #f48942;
     font-size: 60px;
  }
</style>
```

En el **<body>**, donde se imprime toda la información que llega desde el controlador es el siguiente.

```
<table class="tabla">
  <tr>
    <td><h3 class="titulo">HousingBook</h3></td>
    <td><h3 class="text-right derecha">{{__('invoice.invoice')}}</h3></td>
  </tr>
</table>
<h4><strong>{{auth()->user()->name}}</strong></h4>
<h6><strong>{{__('invoice.anywhere')}}</strong></h6>
<br>
<table class="tabla">
  <tr>
    <td>{{auth()->user()->address}}</td>
    <td><strong>{{__('invoice.date')}}</strong> {{Config::get('app.locale') == 'es' ? date("d/m/Y", strtotime($invoice->pivot->entry))
: $invoice->pivot->entry}} </td>
  </tr>
  <tr>
    <td>{{auth()->user()->email}}</td>
    <td><strong>{{__('invoice.invoice')}}</strong> {{rand(10000,1000000)}}</td>
  </tr>
  <tr>
    <td><strong></strong> {{auth()->user()->phone}}</td>
  </tr>
</table>
<br>
<table class="tabla">
  <tr><td><strong>{{__('invoice.to')}}</strong></td></tr>
  <tr>
    <td>{{Auth()->user()->last_name}}, {{Auth()->user()->name}}</td>
  </tr>
  <tr>
    <td>{{Auth()->user()->address}}</td>
  </tr>
</table>
<br>
<table class="table datos p-1">
  <thead>
  <tr class="cabecera text-light">
    <th>{{__('invoice.description')}}</th>
    <th class="text-right">{{__('invoice.price')}}</th>
    <th class="text-right">{{__('invoice.days')}}</th>
    <th class="text-right">Total</th>
  </tr>
  </thead>
  <tbody class="tbody">
  <tr>
    @if(Config::get('app.locale') == 'es')
      <td>{{$invoice->name}} - {{date("d/m/Y", strtotime($invoice->pivot->entry))}}  /  {{date("d/m/Y",
strtotime($invoice->pivot->exit))}}</td>
    @else
      <td>{{$invoice->name}} - {{$invoice->pivot->entry}}  /  {{$invoice->pivot->exit}}</td>
    @endif
    <td class="text-right">{{$invoice->price}} &euro;</td>
    <td class="text-right">{{$days}}</td>
    <td class="text-right">{{$invoice->pivot->total}} &euro;</td>
  </tr>
```

```
<tr>
      <td colspan="3" class="text-right"><strong>Total:</strong></td>
      <td class="text-right">{{$invoice->pivot->total}} &euro;</td>
  </tr>

  </tbody>


</table>

<hr>
<footer class="footer text-center text-light p-2">
  <h3>HousingBook</h3>
  <p>{{__('invoice.issue')}}</p>
  <br><br>
</footer>
```

- tutorial.blade.php

Esta vista muestra un tutorial de como obtener la id de telegram desde el móvil.
Se cargan estilos en línea para la vista.

```
<style>
  *{
     color: #6b6b6b;
  }
  .fotos{
     text-align: center;
  }
  img{
     border: 1px solid #6b6b6b;
     border-radius: 5px;

  }
  h3{
     color: white;
     background-color: #1cc5db;
     padding: 10px;
     border-radius: 5px;
  }
</style>
```

En el **<body>** se introduce lo siguiente.

```
<h3 class="fotos">{{__('telegram.how')}}</h3>
<hr>
<p>{{__('telegram.identi')}}</p>
<p>{{__('telegram.util')}}</p>
<hr>
<p>{{__('telegram.obtain')}}</p>
<br>
<p class="fotos"><img src="{{asset('img/telegram-1.jpg')}}"  width="300" height="300" alt="">
  <img src="{{asset('img/telegram-2.jpg')}}" width="300" height="300" alt=""></p>
<p>{{__('telegram.id')}}</p>
<p class="fotos"><img src="{{asset('img/telegram-3.jpg')}}" width="300" height="300" alt=""></p>
```