

Generar un CRUD
completo

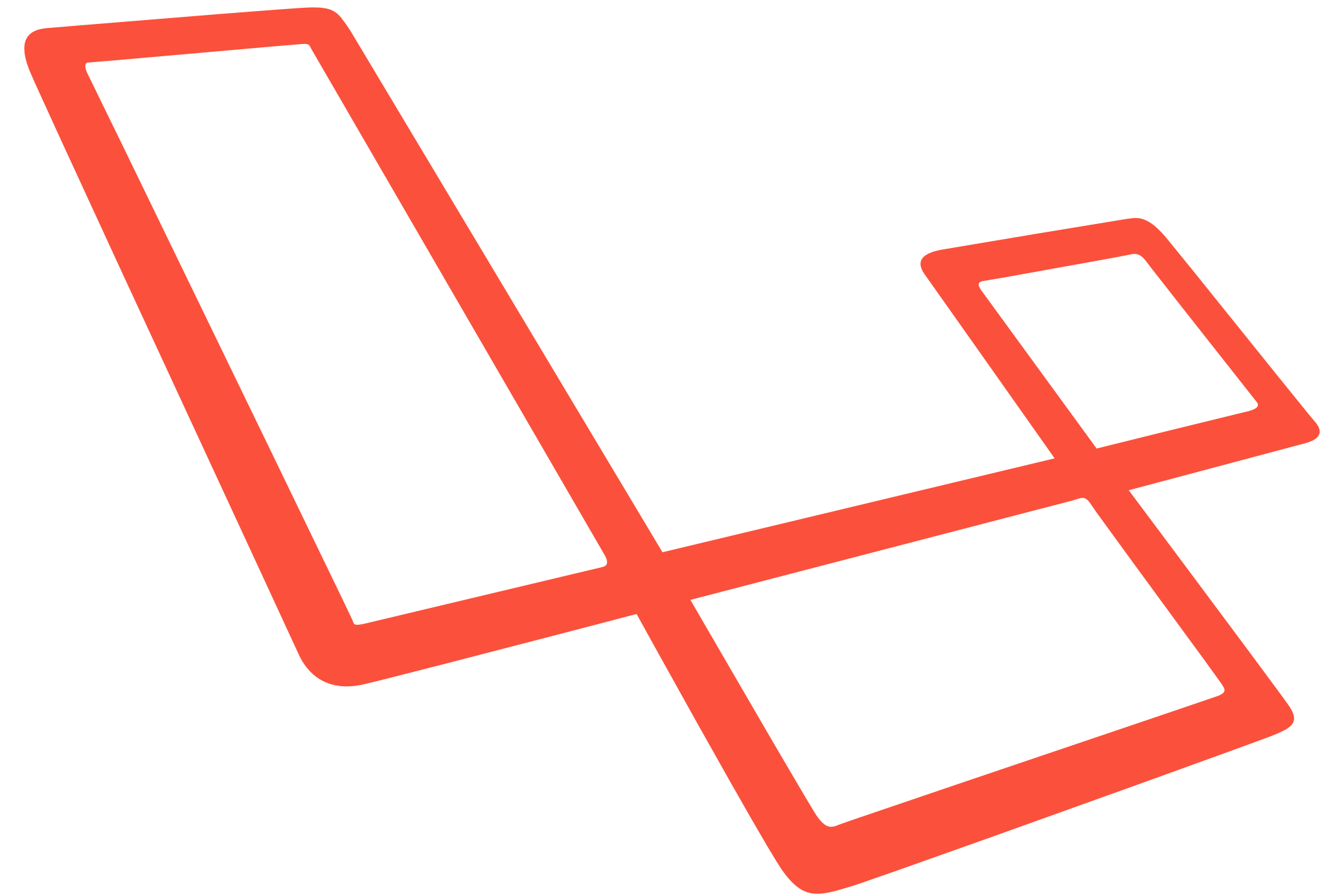
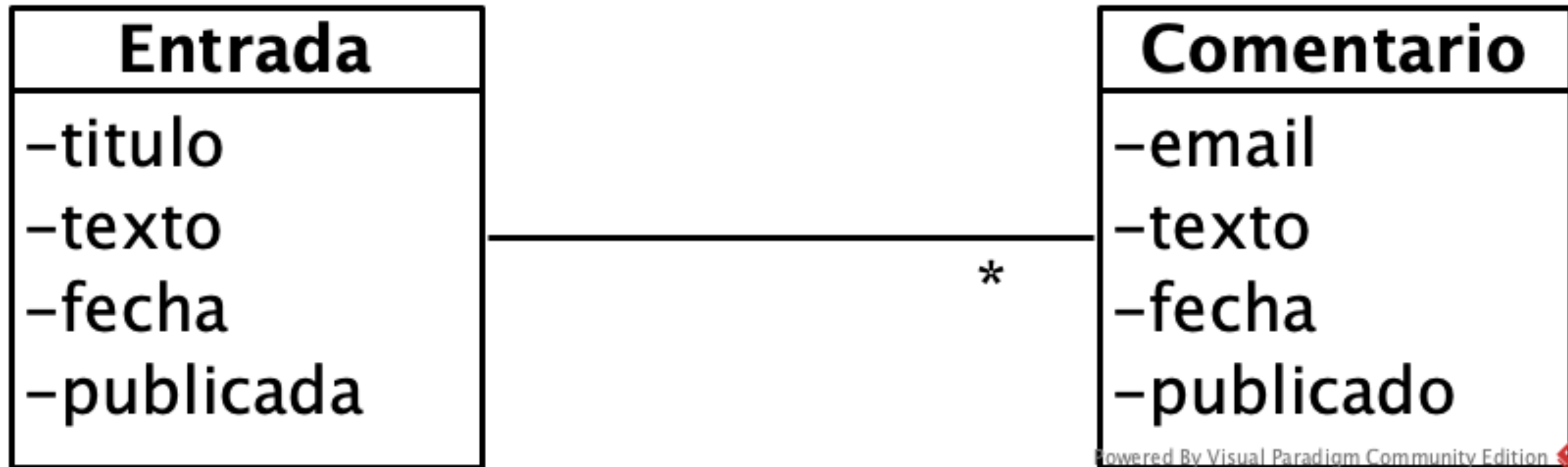


Diagrama de clases



Gist

<http://bit.ly/20t22su>

Modelo

Modelo y recursos asociados

Model

Son representaciones de las entidades de datos de nuestra aplicación

Factory

Definen una plantilla para generar datos de prueba

Migration

Define la estructura de las tablas/entidades

Seeder

Inserta datos de prueba en la base de datos

(Resource) Controller

Controlador que define las acciones CRUD para un modelo concreto

Entrada

Crear el modelo y sus recursos asociados

```
php artisan make:model Entrada -a
```

<https://documentacion-laravel.com/eloquent.html>

Editar la migración

// database/migrations/2020..._create_entradas_table.php

```
public function up()  
{  
    Schema::create('entradas', function (Blueprint $table) {  
        $table->bigIncrements('id');  
  
        $table->string('titulo');  
        $table->text('texto')->nullable();  
        $table->dateTimeTz('fecha')->nullable();  
        $table->boolean('publicada')->nullable()->default(false);  
  
        $table->timestamps();  
    });  
}
```

<https://documentacion-laravel.com/migrations.html>

Lanzar la migración

```
php artisan migrate
```

Editar el factory

```
// database/factories/EntradaFactory.php
```

```
$factory->define(Entrada::class, function (Faker $faker) {  
    return [  
        'titulo' => $faker->sentence(3),  
        'texto' => $faker->text(200),  
        'fecha' => now(),  
        'publicada' => false,  
    ];  
});
```

<https://github.com/fzaninotto/Faker>

Editar el seeder

// database/seeds/EntradaSeeder.php

```
class EntradaSeeder extends Seeder
{
    public function run()
    {
        for ($i = 0; $i < 5; $i++) {
            factory(Entrada::class)->create();
        }
    }
}
```

<https://documentacion-laravel.com/database-testing.html#usando-seeders>

Activar el seeder

// database/seeds/DatabaseSeeder.php

```
class DatabaseSeeder extends Seeder
{
    public function run()
    {
        $this->call(EntradaSeeder::class);
    }
}
```

Lanzar el seeder

```
php artisan db:seed
```

```
composer dump-autoload
```

Recrear la base de datos completa

```
php artisan migrate:fresh --seed
```



Añadir los campos *fillable* y *dates* al modelo

```
// app/Entrada.php
```

```
class Entrada extends Model
{
    protected $fillable = [
        'titulo', 'texto', 'fecha', 'publicada'
    ];

    protected $dates = [
        'created_at', 'updated_at', 'fecha'
    ];
}
```

<https://documentacion-laravel.com/eloquent.html#insertando-y-actualizando-modelos>

Comentario

Crear el modelo y sus recursos asociados

```
php artisan make:model Comentario -a
```

Migración

```
// database/migrations/2020_..._create_comentarios_table.php
```

```
public function up()  
{  
    Schema::create('comentarios', function (Blueprint $table) {  
        $table->bigIncrements('id');  
  
        $table->string('email');  
        $table->text('texto')->nullable();  
        $table->dateTimeTz('fecha')->nullable();  
        $table->boolean('publicado')->nullable()->default(false);  
  
        $table->timestamps();  
    });  
}
```

Factory

```
// database/factories/ComentarioFactory.php

$factory->define(Comentario::class, function (Faker $faker) {
    return [
        'email' => $faker->email,
        'texto' => $faker->text(200),
        'fecha' => now(),
        'publicado' => $faker->boolean(30),
    ];
});
```

Seeder

```
// database/seeds/ComentarioSeeder.php
```

```
class ComentarioSeeder extends Seeder
{
    public function run()
    {
        $entrada = Entrada::find(1);

        for ($i = 0; $i < 5; $i++) {
            factory(Comentario::class)->create([
                'entrada_id' => $entrada->id
            ]);
        }
    }
}
```

Activar el seeder

// database/seeds/DatabaseSeeder.php

```
class DatabaseSeeder extends Seeder
{
    public function run()
    {
        $this->call(EntradaSeeder::class);
        $this->call(ComentarioSeeder::class);
    }
}
```

Añadir los campos *fillable* y *dates* al modelo

```
// app/Comentario.php
```

```
class Comentario extends Model
{
    protected $fillable = [
        'email', 'texto', 'fecha', 'publicado'
    ];

    protected $dates = [
        'created_at', 'updated_at', 'fecha'
    ];
}
```

Relación $1 \rightarrow N$

Añadir las *foreign keys* a las tablas

```
// database/migrations/2020_..._create_comentarios_table.php
```

```
public function up()
{
    Schema::create('comentarios', function (Blueprint $table) {
        $table->bigIncrements('id');

        $table->string('email');
        $table->text('texto')->nullable();
        $table->dateTimeTz('fecha')->nullable();
        $table->boolean('publicado')->nullable()->default(false);

        $table->bigInteger('entrada_id')->unsigned()->index();
        $table->foreign('entrada_id')->references('id')->on('entradas')->onDelete('cascade');

        $table->timestamps();
    });
}
```

<https://documentacion-laravel.com/eloquent-relationships.html>

Definir las relaciones entre entidades

```
// app/Entrada.php
```

```
class Entrada extends Model
{
    ...

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

Definir las relaciones entre entidades

```
// app/Comentario.php
```

```
class Comentario extends Model
{
    ...

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

Rutas

Definir las rutas

```
// routes/web.php
```

```
Route::resource('entradas', 'EntradaController');  
Route::resource('comentarios', 'ComentarioController');
```

Ver las rutas

```
php artisan route:list
```

Ver las rutas

| Domain | Method | URI | Name | Action | Middleware |
|--------|-----------|-------------------------------|---------------------|---------------------------------------------------|--------------|
| | GET HEAD | / | | Closure | web |
| | GET HEAD | api/user | | Closure | api,auth:api |
| | GET HEAD | comentarios | comentarios.index | App\Http\Controllers\ComentarioController@index | web |
| | POST | comentarios | comentarios.store | App\Http\Controllers\ComentarioController@store | web |
| | GET HEAD | comentarios/create | comentarios.create | App\Http\Controllers\ComentarioController@create | web |
| | GET HEAD | comentarios/{comentario} | comentarios.show | App\Http\Controllers\ComentarioController@show | web |
| | PUT PATCH | comentarios/{comentario} | comentarios.update | App\Http\Controllers\ComentarioController@update | web |
| | DELETE | comentarios/{comentario} | comentarios.destroy | App\Http\Controllers\ComentarioController@destroy | web |
| | GET HEAD | comentarios/{comentario}/edit | comentarios.edit | App\Http\Controllers\ComentarioController@edit | web |
| | GET HEAD | entradas | entradas.index | App\Http\Controllers\EntradaController@index | web |
| | POST | entradas | entradas.store | App\Http\Controllers\EntradaController@store | web |
| | GET HEAD | entradas/create | entradas.create | App\Http\Controllers\EntradaController@create | web |
| | GET HEAD | entradas/{entrada} | entradas.show | App\Http\Controllers\EntradaController@show | web |
| | PUT PATCH | entradas/{entrada} | entradas.update | App\Http\Controllers\EntradaController@update | web |
| | DELETE | entradas/{entrada} | entradas.destroy | App\Http\Controllers\EntradaController@destroy | web |
| | GET HEAD | entradas/{entrada}/edit | entradas.edit | App\Http\Controllers\EntradaController@edit | web |

Controlador

Acciones de un controlador de recursos

| Verbo HTTP | URI | Acción | Nombre de ruta |
|------------|-------------------------|---------|------------------|
| GET | entradas | index | entradas.index |
| GET | entradas/create | create | entradas.create |
| POST | entradas | store | entradas.store |
| GET | entradas/{entrada} | show | entradas.show |
| GET | entradas/{entrada}/edit | edit | entradas.edit |
| PUT/PATCH | entradas/{entrada} | update | entradas.update |
| DELETE | entradas/{entrada} | destroy | entradas.destroy |

<https://documentacion-laravel.com/controllers.html#controladores-de-recursos>

Crear los métodos del controlador

<http://bit.ly/20t22su>

Vistas

Crear las vistas

<http://bit.ly/20t22su>

Ajustar la zona horaria del servidor

```
// config/app.php
```

```
'timezone' => 'Europe/Madrid',
```

<https://documentacion-laravel.com/localization.html>

Tests

Tests

- Si aplicamos TDD, se escriben antes que el código del controlador
- Nos permiten verificar que el controlador funciona correctamente
- Ayudan a encontrar errores debidos a interacciones inesperadas
- Para escribirlos seguiremos el orden "Given→When→Then"

Crear la clase de test

```
php artisan make:test EntradaTest
```

<https://documentacion-laravel.com/testing.html>

Escribir los tests

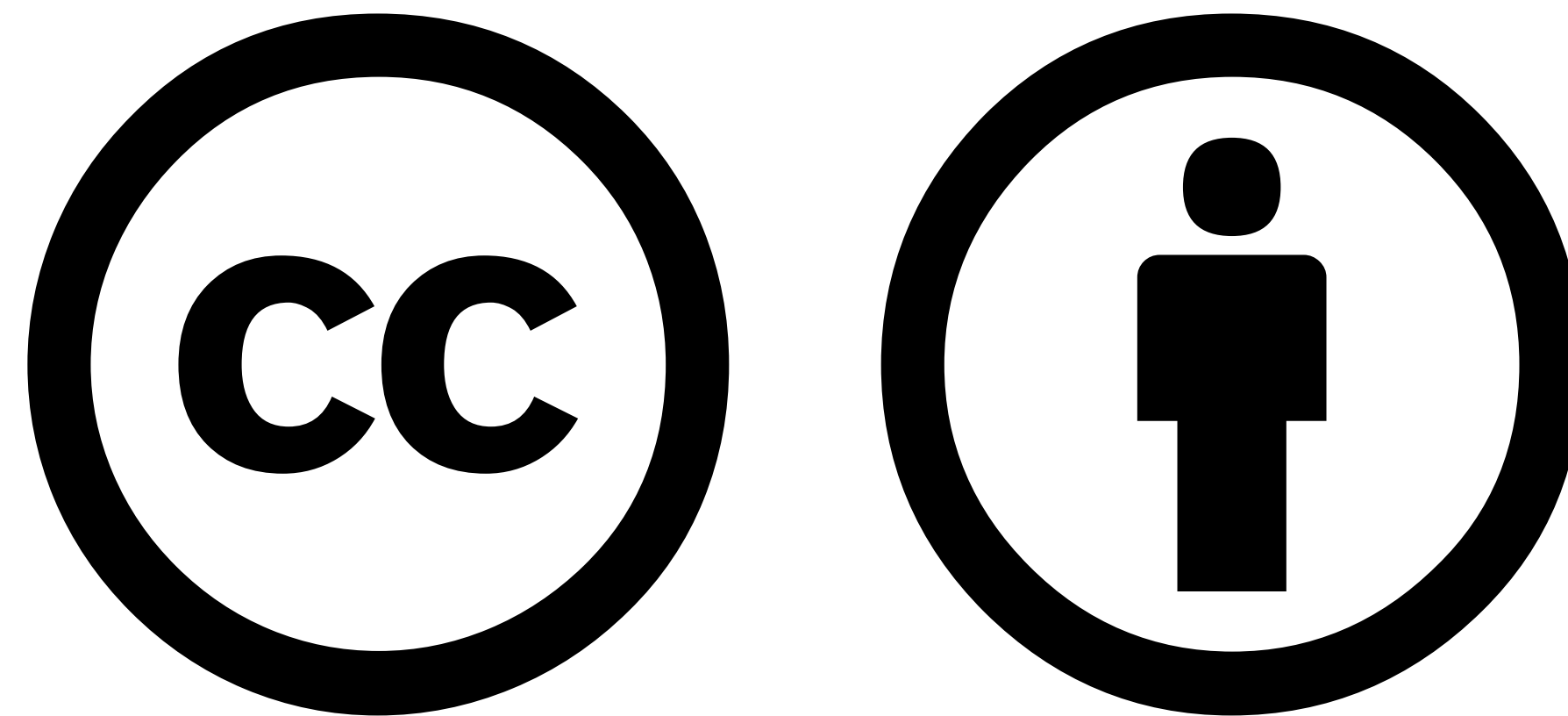
<http://bit.ly/20t22su>

Lanzar los tests

```
vendor/bin/phpunit tests/Feature/EntradaTest.php
```

Buenas prácticas

<http://bit.ly/2vb30Tw>



Excepto si se especifica lo contrario, esta presentación está bajo licencia

<https://creativecommons.org/licenses/by/4.0/>

© 2020 Ion Jaureguialzo Sarasola. Algunos derechos reservados.