

Desafío 1

Crear nuevo proyecto

```
composer create-project laravel/laravel crud2
```

Configuración: BD, SMTP, REDIS archivo en el archivo **.env**

Motor Base de datos: MYSQL

```
DB_CONNECTION=mysql
DB_HOST=localhost
DB_PORT=3306
DB_DATABASE=crud
DB_USERNAME=root
DB_PASSWORD=*****
```

Servidor de Correo: SMTP (MailTrap)

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=5a585aba5fb36d
MAIL_PASSWORD=20d24664a663a6
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS=null
MAIL_FROM_NAME="${APP_NAME}"
```

Servidor de estructura de datos: REDIS

```
REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379
```

Desafío 2

```
class Publication extends Model
{
    public $timestamps = false;
    public function comments()
    {
        return $this->hasMany('App\Comment');
    }
    protected $fillable = [
        'title', 'content', 'auth'
    ];
}
```

```
class Comment extends Model
{
    public function publication()
    {
        return $this->belongsTo('App\Publication');
    }
    protected $fillable = [
        'status', 'content'
    ];
}
```

Desafío 3

```
$comments_hola_aprobado = Publication::find()->comments()
->where('content', 'like', 'HoLa')
->where('status', 'APROBADO')->get();
```

Otra manera:

```
$comments_hola_aprobado = Publication::find()->comments()
->whereColumn([
    ['content', 'like', 'Hola'],
    ['status', '=', 'Aprobado'],
])->get();
```

Desafío 4

Crear Migraciones

Las migraciones diseñan las tablas como si de modelos se tratarán y luego las puedes generar al Motor de base de datos, con pocos comandos. Controlando la cronología de la creación de las tablas. Las migraciones son de gran utilidad siempre y cuando se tenga en cuenta las precauciones previas antes de realizarla.