



# Frameworks Modernos para desarrollo Ágil

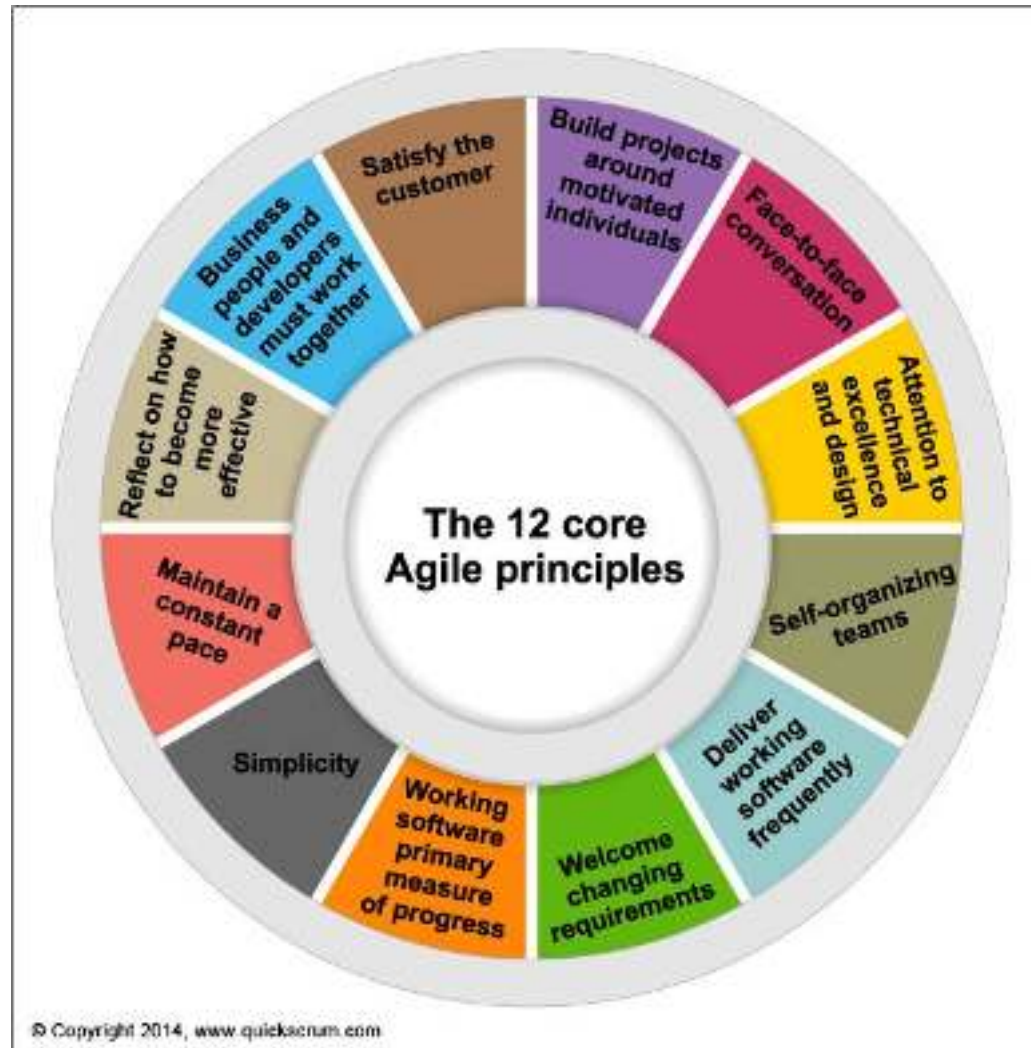
Juan Fajardo Barrero



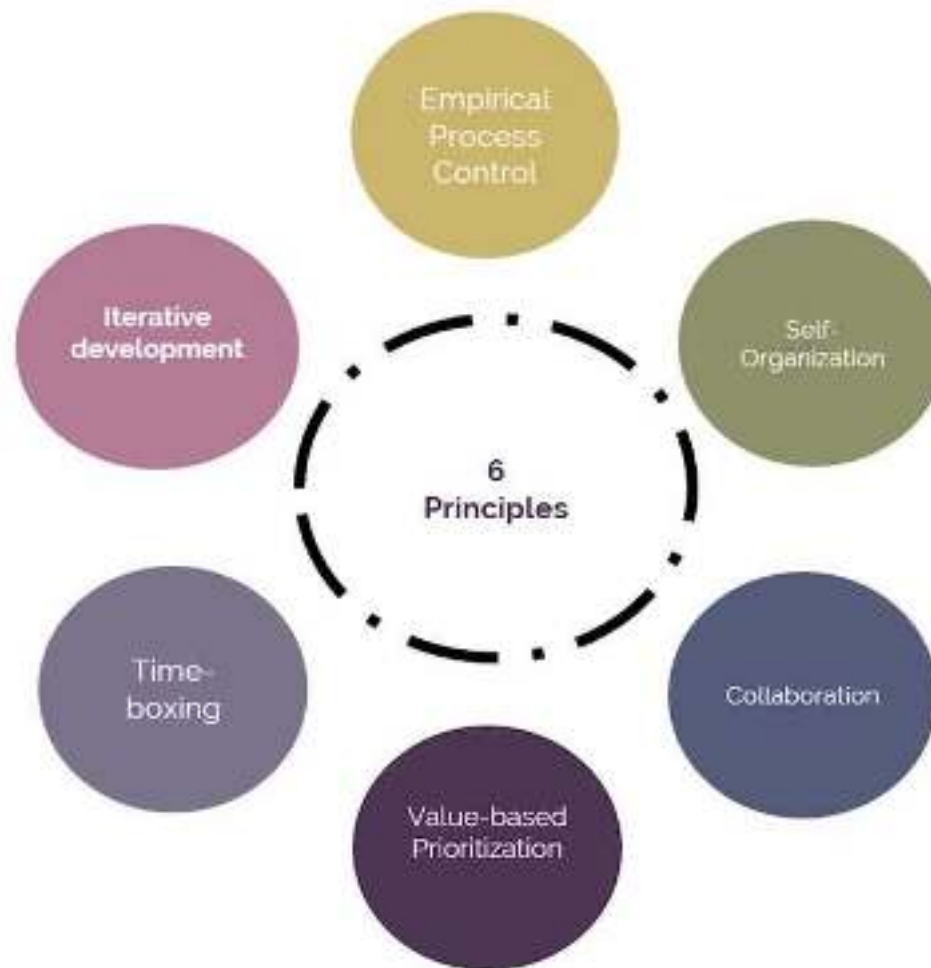
# Agenda

- **Agile Development: En que consiste**
- **Metodologías ágiles**
- **Frameworks**
- **Frameworks FrontEnd**
- **Frameworks BackEnd**
- **MVC**
- **Laravel Como ejemplo**
- **Otras Herramientas**
- **Conclusiones**

# Desarrollo Ágil



# Scrum



# Frameworks

## SOFTWARE FRAMEWORK

Sets of libraries or classes

Built-in generic functionalities,  
Deals with standard low-level details

Reusable software environment

Working template application

Can be modified by writing additional code






<http://inderpsingh.blogspot.com/>



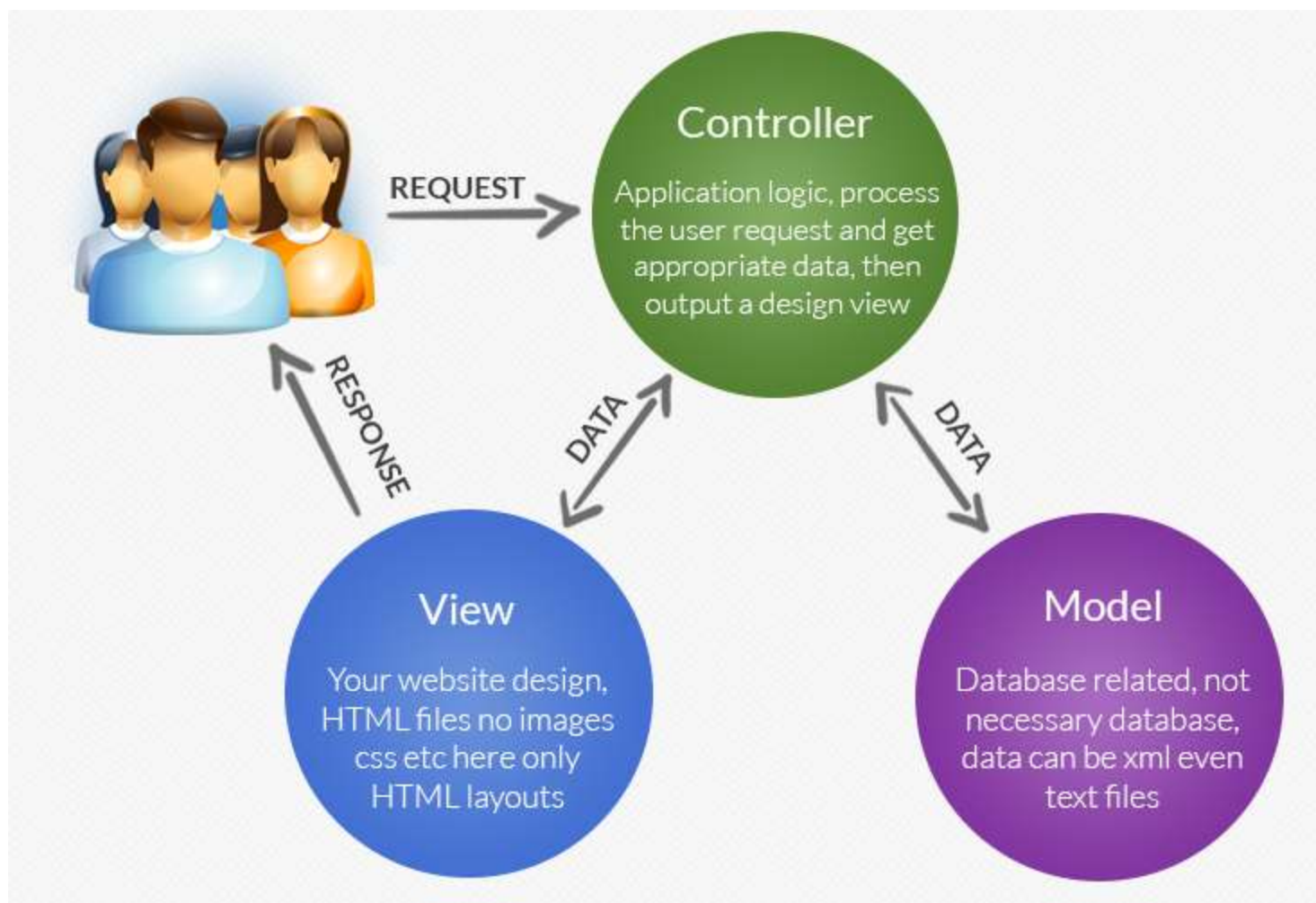
# FrontEnd



# BackEnd

FrameWork	Lenguaje / Herramienta	
MVC 4,5	DotNet: C# / VB	
Rails	Ruby	
Django	Python	
Spring	Java	
Laravel	PHP	

# MVC en 10 minutos





# Ejemplo MVC

1	2	0
0	1	0
0	0	2

Modelo



Usuario



Controlador



# No MVC

```
#include <stdio.h>
```

```
void show() {  
    int i;  
    for ( i=0;i<9;i++)  
    {  
        if (m.get(i)==0) printf(" ");  
        if (m.get(i)==1) printf("X");  
        if (m.get(i)==2) printf("O");  
        if (i%3==0) printf("\n");  
    }  
}
```

```
int main()  
{  
    int board[9]={0,0,0,0,0,0,0,0,0};  
    int turn=0;  
    int i,pos;  
  
    for (int i=0;i<9;i++)  
    {  
        printf("Play:");  
        scanf("%d",&pos);  
        board[pos]=1+(i%2);  
        show(board);  
    }  
}
```

# Modelo

```
public class TTTModel {  
    int board[9];  
    int turn;  
public:  
    void play(int pos)  
    {  
        if (turn%2==0)  
            board[pos]=1;  
        else  
            borad[pos]=1;  
        turn++;  
    }  
    int get(int pos)  
    {  
        return board[pos];  
    }  
}
```

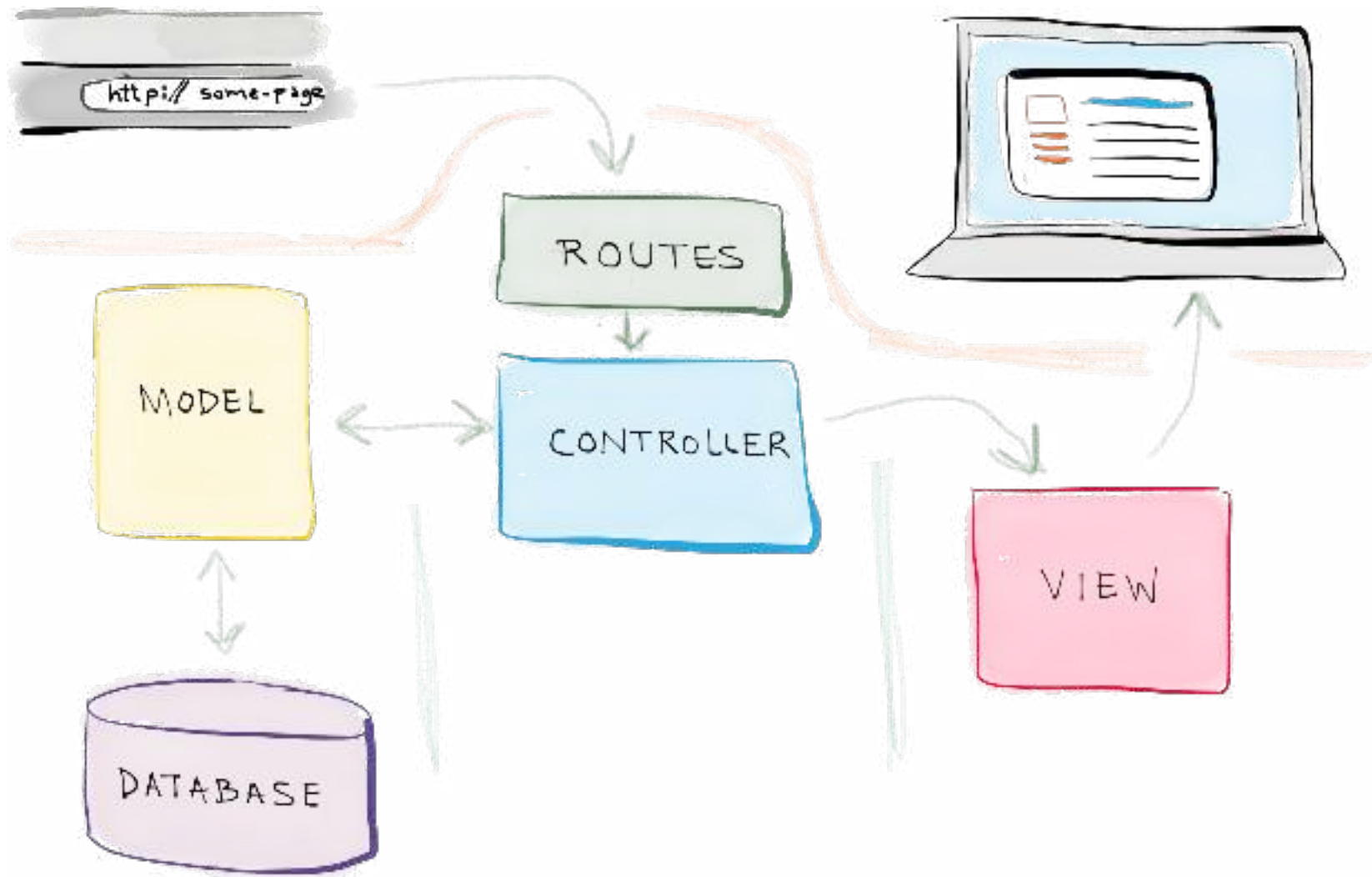
# Vista

```
public class TTTView {
    public:
    void show(TTTModel m)
    {
        for (int i=0;i<9;i++)
        {
            if (m.get(i)==0) cout<<"_ ";
            if (m.get(i)==1) cout<<"X";
            if (m.get(i)==2) cout<<"O";
            if (i%3==0) cout<<endl;
        }
    }
    int read()
    {
        int pos;
        cout<<"Play:";
        cin>>pos;
        return pos;
    }
}
```

# Controlador

```
public class TTTController
{
    TTTModel model;
    TTTView view;
    public:
    TTTController()
    {
        model=new TTTModel();
        view=new TTTView();
    }
    void play()
    {
        int i,pos;
        for (i=0;i<9;i++)
        {
            view.show();
            pos=view.read();
            model.play(pos);
        }
    }
}
```

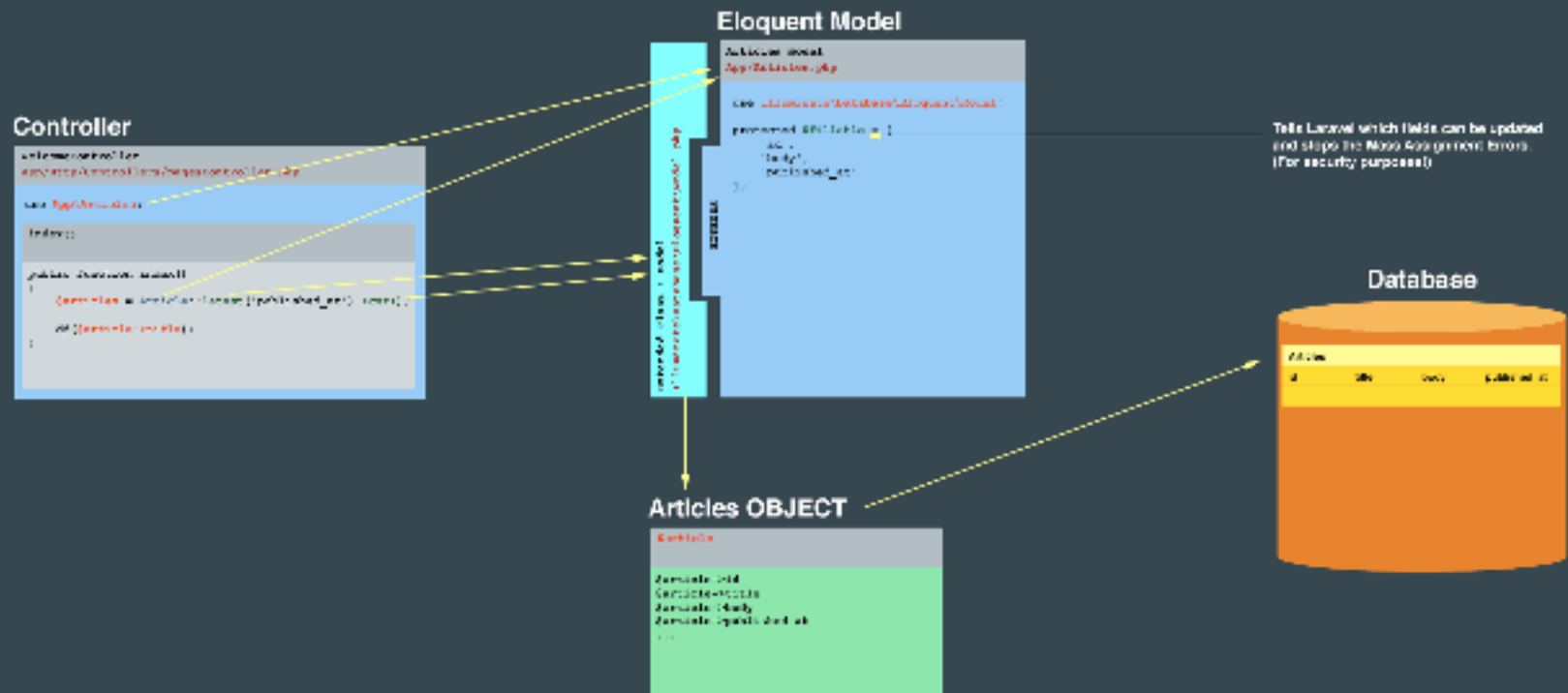
# MVC En Web



# Laravel Como Ejemplo

## Eloquent ORM

The Model.php class (part of the eloquent library) gives us a much easier way of interacting with the Database through the methods it provides.



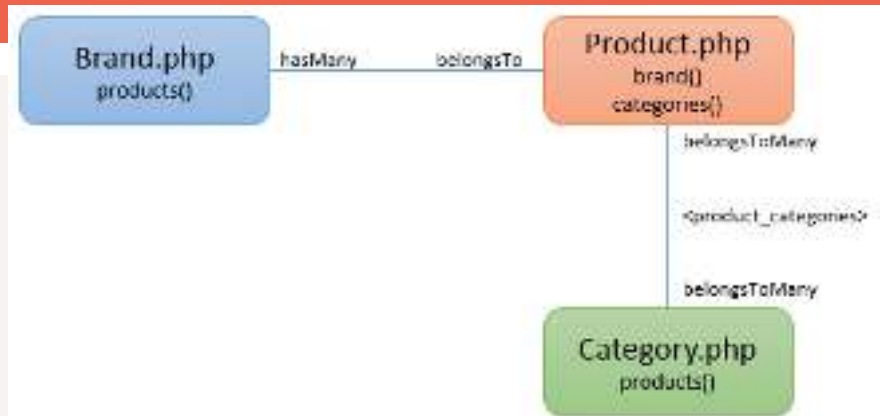
# Laravel Model

```
class Brand extends Model
{
    use SoftDeletes;

    /**
     * The attributes that should be mutated to dates.
     *
     * @var array
     */
    protected $dates = ['deleted_at'];

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name',
        'description',
    ];

    /**
     * Get the products for the brand.
     */
    public function products()
    {
        return $this->hasMany('Larashop\Models\Product', 'brand_id', 'id');
    }
}
```





# Laravel : Model

```
<?php
```

```
namespace App;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class TipoEstudio extends Model
```

```
{
```

```
    //
```

```
    protected $table = "tipo_estudio";
```

```
    public $timestamps=false;
```

```
    protected $primaryKey = 'tipo_estudio'; // or null
```

```
    public $incrementing = false;
```

```
    protected $fillable = ['tipo_estudio', 'descripcion'];
```

```
}
```

# Laravel View

```
<table id="mytable" class="table table-bordered table-striped">
<thead>
  <th>Tipo de Estudio</th>
  <th>Descripcion</th>
</thead>
<tbody>
  @if($tiposestudio->count())
  @foreach($tiposestudio as $t)
    <tr>
      <td>{{$t->tipo_estudio}}</td>
      <td>{{$t->descripcion}}</td>
    @endforeach
  @else
  <tr>
    <td colspan="2">No se han encontrado datos !!</td>
  </tr>
  @endif
</tbody>
</table>
```

# Laravel Controller

```
namespace App\Http\Controllers;
use App\TipoEstudio;
use Illuminate\Http\Request;

class TipoEstudioController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        //
        $tiposestudio = TipoEstudio::orderBy('tipo_estudio','ASC')->paginate(5);

        return view('TipoEstudio.index',compact('tiposestudio'));
    }
}
```

# Resultado

Inicio » Dashboard

Dashboard » Inicio

Tipos de estudio

Tipo de estudio

Descripcion

Tipo de estudio	Descripcion	
EMBI	Estudio de Movilidad de Bajo Impacto	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
EMAI	Estudio de Movilidad de Alto Impacto	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

# Otras Herramientas

- **Scrum como metodología**
- **Herramientas para comunicación efectiva: Kanboard, Wekan**
- **Historias de usuario como herramienta de análisis**
- **Control de versiones: Git, SVN**
- **Un buen sistema operativo: Linux**
- **Un buen motor de bases de datos: Postgres**
- **Diseño de interfaces (Web, Escritorio) : Bootstrap, Foundation**

# Conclusiones

- **Muchas alternativas**
- **Desarrollo ágil = adaptable**
- **Desarrollo colaborativo**
- **Control de versiones**
- **Más seguridad en las aplicaciones**
- **Aprovechar el framework**

# Preguntas - Referencias

<https://laravel.com/>

<https://www.djangoproject.com/>

<https://spring.io/>

<https://rubyonrails.org/>

[\*\*https://letsclearitup.com.ua/books/2017/  
laravel\\_5\\_cookbook.pdf\*\*](https://letsclearitup.com.ua/books/2017/laravel_5_cookbook.pdf)