4- Trabajando la Home

Luego de trabajar con el layout en el articulo anterior, vamos a continuar trabajando en la Home del sitio. Recuerda que siempre estamos atentos a los comentarios y en el slack de Laraveles para dudas, consultas, y/o sugerencias.

Empecemos con la primer y única ruta por ahora del proyecto

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

Si ya hemos trabajado con laravel este código es trivial: (recordar que es el archivo **/resources/views/home.blade.php**)

```
@extends('template.layout')
@section('title', 'Home')
@section('content')
@endsection
@section('script')
@endsection
```

Con este código hemos creado la Home sin contenido pero con los links y los css ya cargados.

Para no estar repitiendo una y otra vez las cosas todo nuestro código html irá siempre en la **section content** y nuestro código Vue en la **section script**

Para no hacer el post demasiado largo pondremos el código html inicial completo. (recordar que **todo** este contenido debe ir en la **section content**)

```
 <div class="container">
        <div class="columns personal-menu text-center vertical-center margin0">
            <div class="column">
                Zona de pruebas
            </div>
        </div>
        <div class="columns margin0 tile">
            <div class="column is-2 line-der">
                <aside class="menu">
                    <p class="menu-label">
                        Menu Principal
                    </p>
                    <ul class="menu-list">
                        <li @click="menu=0" class="hand-option"><a
                                    :class="{'is-active' : menu==0 }">Dashboard</a></li>
                        <li @click="menu=1" class="hand-option"><a :class="{'is-active' : menu==1
}">Departamentos</a>
                        </li>
                        <li @click="menu=2" class="hand-option"><a
                                    :class="{'is-active' : menu==2 }">Cargos</a></li>
                        <li @click="menu=3" class="hand-option"><a
                                    :class="{'is-active' : menu==3 }">Empleados</a></li>
                    </ul>
                </aside>
            </div>
            <div class="column personal-content" v-if="menu==0">
                <div class="columns text-center">
                    <div class="column">
                        <h3>Dashboard</h3>
                    </div>
                </div>
                <div class="columns text-center">
                    <div class="column">
                        <h1>Bienvenido</h1>
                    </div>
                </div>
            </div>
            <div class="column" v-if="menu==1">
                <div class="columns">
                    <div class="column text-center">
                        <h3>Departamentos</h3>
                    </div>
                </div>
                <div class="columns">
                    <div class="column">
                        Tabla de  departamentos
                    </div>
                </div>
            </div>
            <div class="column" v-if="menu==2">
                <div class="columns">
                    <div class="column text-center">
                        <h3>Cargos</h3>
                    </div>
                </div>
                <div class="columns">
                    <div class="column">
                       Tabla Cargos
                    </div>
                </div>
            </div>
            <div class="column" v-if="menu==3">
                <div class="columns">
                    <div class="column text-center">
                        <h3>Empleado</h3>
                    </div>
                    <div class="column">
                       Tabla Empleados
                    </div>
                </div>
            </div>
        </div>
        <div class="columns margin0 text-center vertical-center personal-menu">
            <div class="column">Empleados 0</div>
            <div class="column">Departamentos 0</div>
            <div class="column">Cargo 0</div>
        </div>
</div>
```

Para quienes ya conozcan Vue el código es transparente pero para quienes no aclaremos el código diferente que aparece en determinados

Tomemos este ejemplo

```
<li @click="menu=0" class="hand-option">
<a :class="{'is-active' : menu==0 }">Dashboard</a></li>
```

Lo primero que se ve es **@click** esto es el evento click pero controlado por Vue. Como puede observarse lo que hace al hacer click es transparente. No es necesario llamar una función, se podría hacer, pero también se puede hacer directamente cuando es una acción única.

En conclusión indica que al hacer click la variable **menu** pase a tener el valor 0

Lo siguiente que vemos es **:class**. Esto indica a Vue que tiene un código para resolver y que el resultado será un class. En este caso le decimos que coloque la clase **is-active** siempre y cuando menú sea igual a 0.

Esta misma lógica la repetimos 4 veces. una vez por cada opción del menú con la variable menu asociada a diferentes valores.

El otro código Vue que puede verse aquí es **v-if** que es utilizado para mostrar determinado contenido siempre y cuando se cumpla una condición. por ejemplo:

```
<div class="column personal-content" v-if="menu==0">
   <div class="columns text-center">
        <div class="column">
             <h3>Dashboard</h3>
         </div>
   </div>
   <div class="columns text-center">
        <div class="column">
             <h1>Bienvenido</h1>
         </div>
   </div>
</div>
```

Aquí estamos indicando que ese bloque de información se muestre siempre y cuando la variable menú sea igual a 0

Esta condición la tenemos para los siguientes 3 valores posibles de menú

Pero todo esto no funcionaria sin Vue. Veamos que código seria necesario en el **section script** para que esto funcione:

```
<script>
        let elemento = new Vue({
            el: '.app',
            data: {
                menu:0
            }
        })
</script>
```

Simple verdad? Ahora repasemos un tema que dejamos pendiente. Observemos que Vue tiene la siguiente línea **el: '.app'** lo que le indica que debe actuar sobre un elemento que tiene la clase app. Podría ser #app y buscaría un elemento con id app. Es por esto que en el layout que hicimos en el post anterior teníamos esta línea

```
<div class="app">
    @yield('content')
</div>
```

Nos estamos asegurando que todo el contenido esté dentro de un div con la clase app y por lo tanto que Vue tenga efecto sobre todo el contenido.

Volviendo a Vue lo único que necesitamos definir es el valor inicial de menú. Luego según donde se haga click el valor cambiará y tendremos un resultado u otro.

No hay que hacer nada más. Lo que se debe mostrar o ocultar lo resuelve Vue por medio del valor de la variable **menu**

Código: Github

Siguiente post: Departure: migración, modelo y controlador