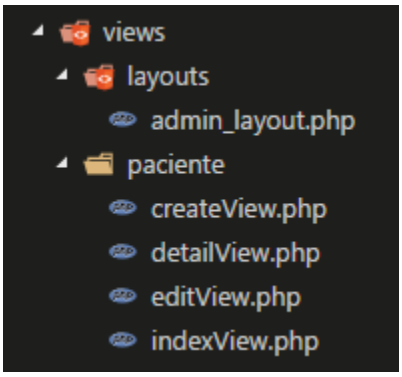
Definición de plantilla o layout del proyecto

Actualmente nuestra estructura de proyecto formativo tiene varias vistas que están repitiendo el código lo cual se puede constituir en un problema porque a la hora de hacer algún cambio si tenemos por ejemplo 20 vistas tendremos que cambiar el código en 20 archivos. Para resolver este problema integraremos a nuestra base de proyecto una plantilla o layout que nos permitirá optimizar este proceso.

Primero se define una variable ***$layout*** que nos permite determinar la plantilla general que tendrá el proyecto. En este caso en el constructor se define con el valor *"**admin_layout.php**"*

```php
<?php

class BaseController
{
    protected $layout;

    public function __construct()
    {
        // Se asigna a la variable layout un valor de plantilla por defecto
        $this->layout = "admin_layout.php";

        require_once 'Connection.php';
        require_once 'BaseModel.php';

        //Incluir todos los modelos
        /*foreach (glob("models/*.php") as $file) {
            require_once $file;
        }*/

        // Incluir todos los modelos de forma manual
        require_once 'models/Paciente.class.php';
    }

    // Crear los Métodos que sean comunes para todos los controladores
}
```

La carpeta **views** ahora tendrá una carpeta 'layouts' donde guardaremos todas las plantillas que requiera el proyecto y también reorganizaremos los demás archivos para que estén organizados por carpetas, para el ejemplo inicial crearemos una carpeta llamada **pacientes** y allí tendremos todas las vistas asociadas al mismo.



Cambiaremos nuestra **vista** del ejemplo y solo dejaremos el contenido como tal (El DIV container) para renderizar los registros de pacientes y el formulario de ingreso. El archivo **pacientes/IndexView.php** quedará de la siguiente forma:

```
<div class="container">
    <div class="row">
        <div class="col-lg-4">
            <form action = "<?php echo "index.php?controller=paciente&action=create"; ?>" method="post">···
            </form>
        </div>
        <div class="col-lg-8">
            <div>
                <h2>Pacientes</h2>
                <hr/>
            </div>
            <div class="row">···
            </div>
            <?php
            if (isset($allPacientes)) {
                foreach ($allPacientes as $row) { //recorremos el array de objetos y obtenemos el valor de las propiedades
            ?>
            <div class="row">
                <div class="col-2"><?php echo $row->documento ?></div>
                <div class="col-3"><?php echo $row->nombre; ?></div>
                <div class="col-3"><?php echo $row->telefono; ?></div>
                <div class="col-2"><?php echo $row->eps; ?></div>
                <div class="col-2">
                    <div class="right col-1">
                        <a href = "<?php echo "index.php?controller=Paciente&action=borrar&id=". $row->documento; ?>" class="btn btn-danger">Borrar</a>
                    </div>
                </div>
            </div><br>
            <?php
                }
            }
            ?>
        </div>
    </div>  <!-- row -->
    <br><br>
</div>
```

La definición de nuestro layout tendrá todos los elementos comunes a todas las vistas, es decir, que tendrá las etiquetas <html>, <head> y <body> al igual que la definición de los estilos, scripts y pie de página (Footer) que se repetirá en todas las vistas. Utilizaremos el siguiente código para poder incluir la vista dentro del layout.

```
<?php
    require_once 'views/'. $current_view
?>
```

El siguiente código irá en el archivo **admin_layout.php:**

admin_layout.php

```php
<!doctype html>
<html lang="es">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <link rel="stylesheet" href="css/mycss.css">
</head>
    <body>
        <header class="container-fluid backg1">
            HEADER MENU
        </header></br>

        <?php
            require_once 'views/'. $current_view
        ?>

        <footer class="container-fluid backg1">
            Ejemplo PHP POO/MVC/PDO - <?php echo date("Y"); ?>
        </footer>

        <!-- Optional JavaScript -->
        <!-- jQuery first, then Popper.js, then Bootstrap JS -->
        <script src="js/jquery-3.3.1.min.js"></script>
        <script src="js/popper.min.js"></script>
        <script src="js/bootstrap.min.js"></script>
    </body>
</html>
```

Por último cambiaremos las acciones de nuestro controlador para que primero se defina la variable de la vista que queremos llamar y luego que se incluya el layout el cual hará finalmente la inclusión de la vista

PacienteController.php

```php
<?php

class PacienteController extends BaseController
{

    public function __construct()
    {
        parent::__construct();
    }

    public function index()
    {
        //Creamos el objeto usuario
        $paciente_obj = new Paciente();

        //Conseguimos todos los usuarios
        $allPacientes = $paciente_obj->getAll();

        // Archivo VISTA que debe incluirse en el Layout
        $current_view = "pacientes/IndexView.php";
        // Se incluye el Layout actual
        require_once 'views/layouts/' . $this->layout;
    }

```

Actividad 1

Los aprendices deberán incluir al Proyecto base **Senamás_EPS**, toda la parte de layouts tal como está especificado en esta guía, teniendo el CRUD completo y funcional de la tabla Pacientes

Luego de esto, se procederá a realizar el CRUD completo de las tablas médico y especialidad