

## Contenido

Caja Blanca: Prueba del camino básico .....	2
Prueba: Actualizar una categoría de producto .....	2
Prueba: Nuevo Contrato .....	3
Prueba: Eliminar una asignación de herramientas .....	5

## Caja Blanca: Prueba del camino básico

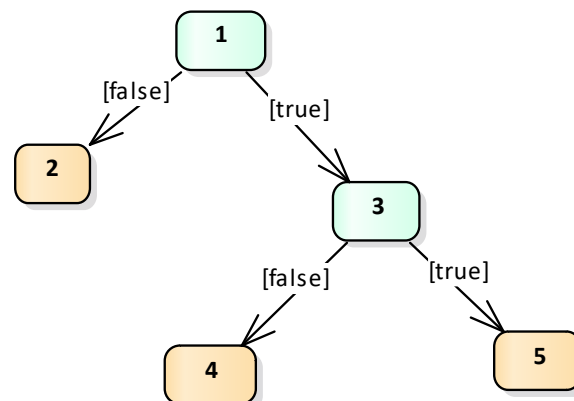
### Prueba: Actualizar una categoría de producto

```
public function update(Request $request, $id)
{
    $this->validate($request, [
        'nombre' => 'required|max:255',
    ]); (1)(2)
    $categoria = Categoria::findOrFail($id); (3)(4)
    $categoria->nombre = $request['nombre'];
    $categoria->update();
    return redirect('categorias'); (5)
}
```

$$V(G) = P + 1$$

$$V(G) = 2 + 1$$

$$V(G) = 3$$



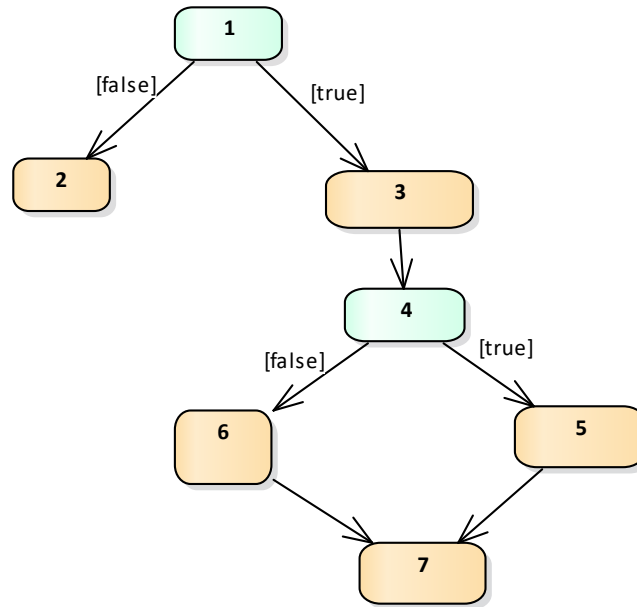
#	Camino	Entrada	Salida
1	1,3,5	{'id':5,'nombre':'mangueras'}	redirect('categorias')
2	1,2	{'id':5, 'nombre':125} (tipo de dato incorrecto)	redirect('categorias/edit/5') ->error('message')
3	1,3,4	{'id':10, 'nombre':'manguera'} (id inexistente)	Página de Error

### Prueba: Nuevo Contrato

```
public function guardarContrato(Request $request){
    $this->validate($request, [
        'fecha_inicio' => 'required|date',
        'fecha_fin' => 'required|date',
        'periodo' => 'required|numeric|min:1',
        'cliente_id' => 'required|numeric|min:1',
        'empleado_id' => 'required|numeric|min:1',
        'documento' => 'nullable|file|mimes:doc,docx,pdf,txt',
    ]); (1)(2)

    $contrato = new Contrato();
    $contrato->fecha_inicio = $request['fecha_inicio'];
    $contrato->fecha_fin = $request['fecha_fin'];
    $contrato->estado = "Vigente";
    $contrato->edicion = true;
    $contrato->periodo = $request['periodo']; (2)
    if (Input::hasFile('documento')) {(4)
        $file = Input::file('documento');
        $file->move(public_path().'/contrato/', $file-
>getClientOriginalName());
        $contrato->documento = $file->getClientOriginalName();(5)
    }else{
        $contrato->documento = 'default.png';(6)
    }
    $contrato->cliente_id = $request['cliente_id'];
    $contrato->empleado_id = $request['empleado_id'];
    $contrato->save();

    return redirect('imonitorio/listaContratos'); (7)
}
```



$$V(G) = P + 1$$

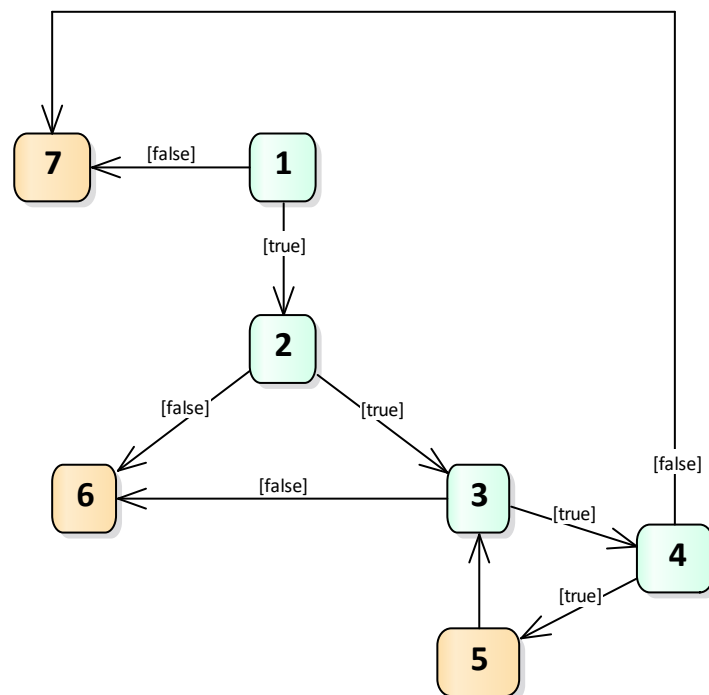
$$V(G) = 2 + 1$$

$$V(G) = 3$$

#	Camino	Entrada	Salida
1	1,2	{'fecha_inicio' : '54.3', 'fecha_fin' : 'abril', 'periodo' : 5, 'cliente_id' : 1, 'empleado_id': 2, 'documento': 'contx.pdf'} (Tipos de datos erroneos)	Redirect('imonitorio/nuev oContrato') ->error('message')
2	1,3,4,6,7	{'fecha_inicio' : '02/02/21', 'fecha_fin' : '02/02/22', 'periodo' : 5, 'cliente_id' : 1, 'empleado_id': 2, 'documento': null} (No se está enviando el file)	redirect('imonitorio/ listaContratos')
3	1,3,4,5,7	{'fecha_inicio' : '02/02/21', 'fecha_fin' : '02/02/22', 'periodo' : 5, 'cliente_id' : 1, 'empleado_id': 2, 'documento': 'contx.pdf'} (Correcto)	redirect('imonitorio/ listaContratos')

## Prueba: Eliminar una asignación de herramientas

```
public function eliminarAsignacion($id)
{
    $asignacion = AsignacionHerramienta::findOrFail($id); (1)(7)
    if ($asignacion->estado == 'Activa'){ (2)
        foreach ($asignacion->detalles as $detalle) {(3)
            $herramienta =
                Herramienta::withTrashed()->findOrFail($detalle-
>herramienta_id);(4)(7)
            $herramienta->cantidad_asignada =
                $herramienta->cantidad_asignada - $detalle->cantidad;
            $herramienta->cantidad_taller =
                $herramienta->cantidad_taller + $detalle->cantidad;
            $herramienta->update(); (5)
        }
    }
    $asignacion->delete();
    return redirect('herramientas/listaAsignaciones'); (6)
}
```



$$V(G) = P + 1$$

$$V(G) = 4 + 1$$

$$V(G) = 5$$

#	Camino	Entrada	Salida
1	1,7	{id:999999999} (id inexistente)	Página Error
2	1,2,6	{id:1} (asignación finalizada)	View("herramientas/listaAsignaciones")
3	1,2,3,6	{id:4} (asignación sin detalle)	View("herramientas/listaAsignaciones")
4	1,2,3,4,7	{id: 3} (asignación con detalle erróneo)	Página Error
5	1,2,3,4,5,3,6	{id:1} (asignación con detalle correcto)	View("herramientas/listaAsignaciones")