**Pruebas Unitarias**

Mariana León

Jorge Sánchez

Juan Melo

Leonardo Gutiérrez

Articulación con la media, Servicio Nacional de Aprendizaje Sena

Ficha: 2464768 ADSI

Fabian Alejandro Rodriguez

5/09/2022

**Introducción**

El propósito de este documento es asegurar el correcto funcionamiento del sistema con diversas pruebas unitarias incluyendo pruebas de tipo caja negra y blanca

**Pruebas Unitarias**

**Objetivos**

El objetivo principal de las pruebas unitarias es probar que cada parte del software funcione correctamente sin ningún error.Esta puede determinar fácilmente el comportamiento exacto del código.

**Beneficios de las pruebas unitarias**

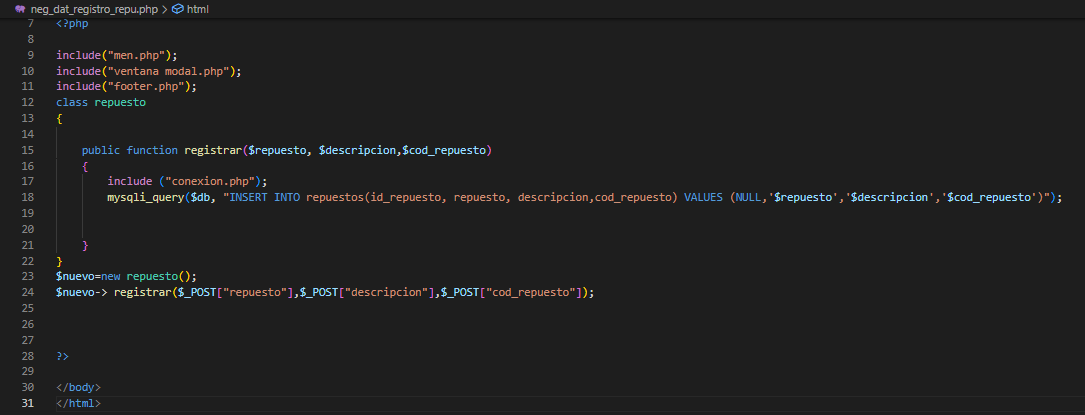
**Calidad del código**

Este tipo de pruebas mejoran la calidad del código y permite a los desarrolladores verificar todos los defectos.

**Detección temprana de errores**

Detectar errores antes de realizar un proceso de integración siempre es beneficioso y ahorra tiempo.Dado que los desarrolladores escriben códigos para pruebas unitarias los problemas pueden ser localizados de una forma más eficiente.Esto ahorra tiempo y mejora la calidad del código

**Capa** neg\_dat\_registro\_repu.php



Después verificamos la funcionalidad de la clase repuesto con la función registrar



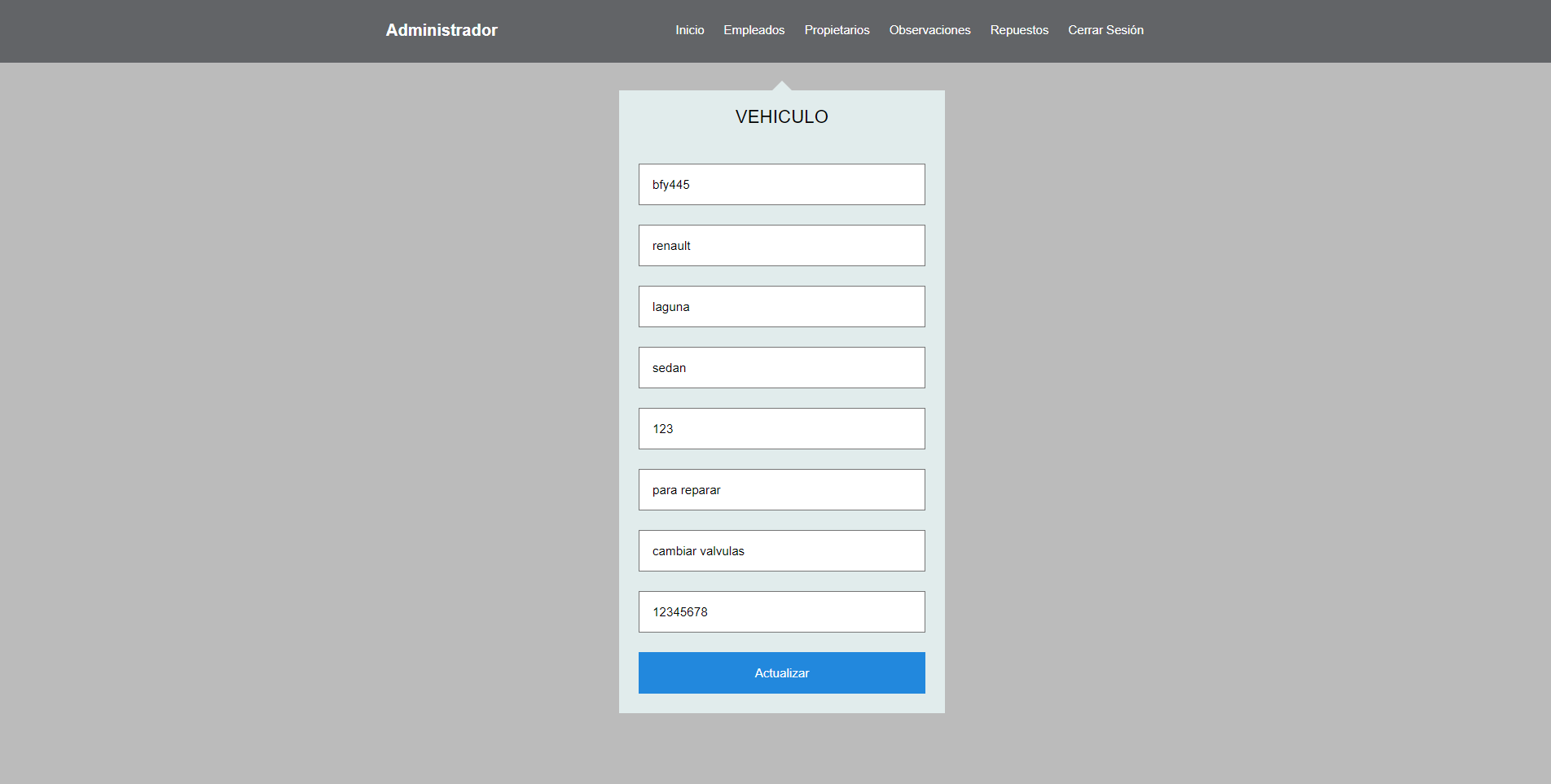
Podemos visualizar que al realizar dicha prueba se puede observar la funcionalidad que este conllevo

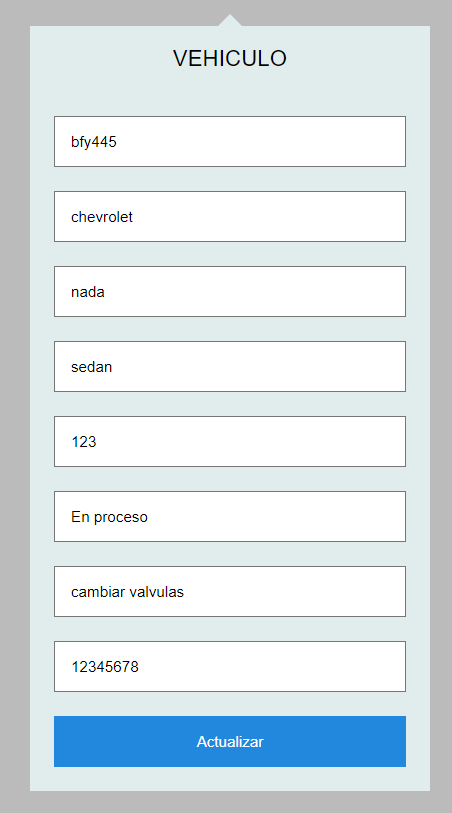
| **Resultados esperados** | **Resultado obtenido** |
| --- | --- |
| Los datos ingresados en el formulario se almacenan en la base de datos | Se pueden visualizar los datos insertados anteriormente |

**Pruebas de caja Negra**

**Prueba 1**

Para este fragmento comprobaremos la funcionalidad del código a partir de unos datos observando que el resultado arrojado sea el correcto para esto solo necesitamos un dato de entrada y esperar la ejecución

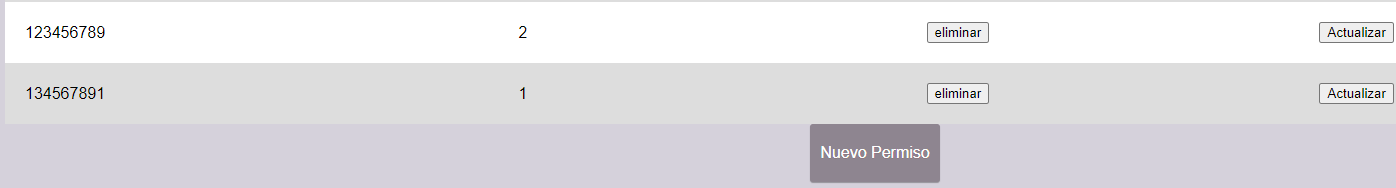
****



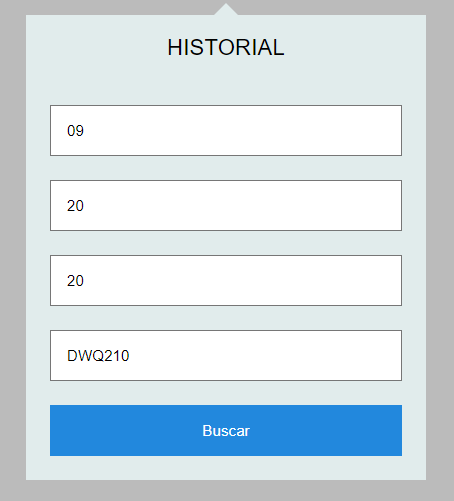
| **Resultados esperados** | **Resultado obtenido** |
| --- | --- |
| Los datos ingresados en el formulario se almacenan en la base de datos | Se pueden visualizar los datos insertados anteriormente |

**Prueba 2**

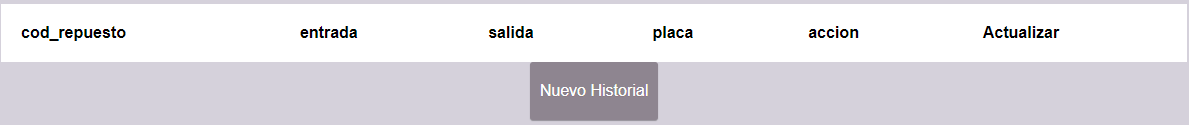




| **Resultados esperados** | **Resultado obtenido** |
| --- | --- |
| Se esperaba que solo se pudieran registrar los roles tipo mecánicos. | El resultado obtenido es que se puede modificar el campo rol a un rol tipo administrador. |

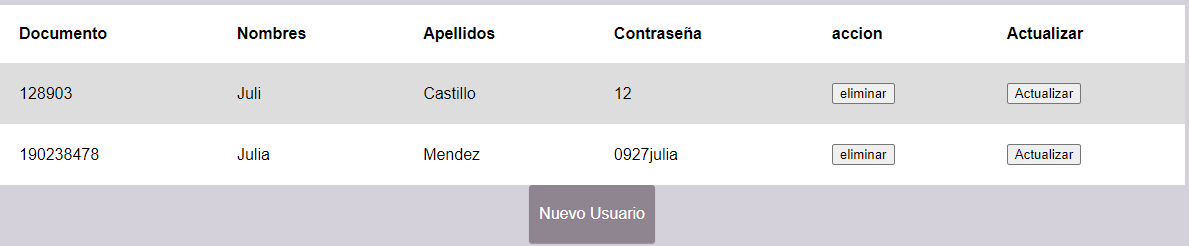






| **Resultados esperados** | **Resultado obtenido** |
| --- | --- |
| Se esperaba que solo se pudieran eliminar solo una fila ya con datos actualizados | El resultado obtenido es que se puede modificar los diversos campos pero al momento de actualizarlos si los dos cuentan con el mismo código del repuesto se eliminan las dos filas |





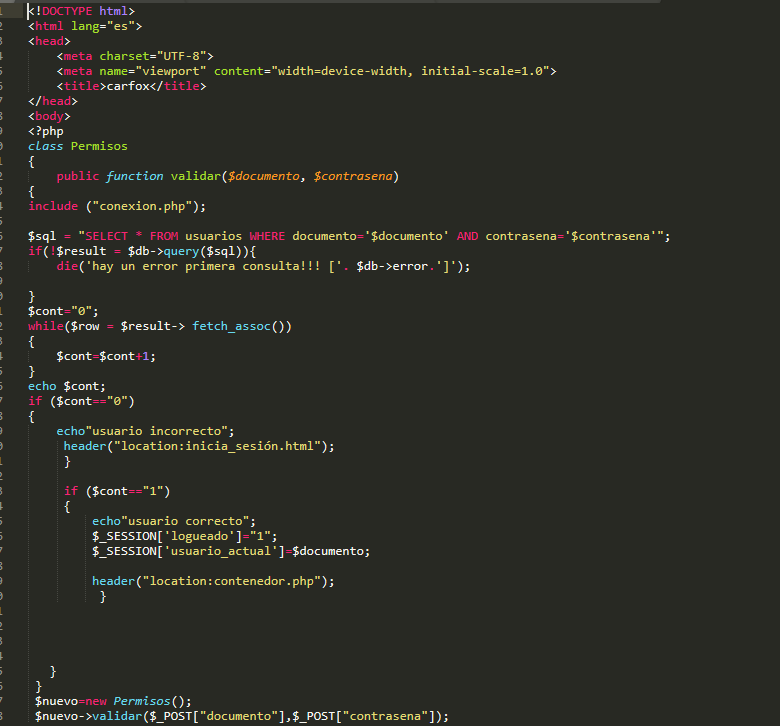
| **Resultados esperados** | **Resultado obtenido** |
| --- | --- |
| Se esperaba que solo se pudiera modificar el documento | El resultado obtenido son datos almacenados previamente ingresados en el registro pero que en el apartado de actualizar el documento esta tipo hidden debido a que no se podrá modificar. |

**Pruebas de caja blanca**

En las pruebas de caja blanca lleva como objetivo el diseño de casos de prueba que utiliza la estructura de control y diseño junto con las funciones internas que estas conllevan.

Con esta prueba se pretende verificar las diferentes ramas de decisión,bucles y declaraciones en el código

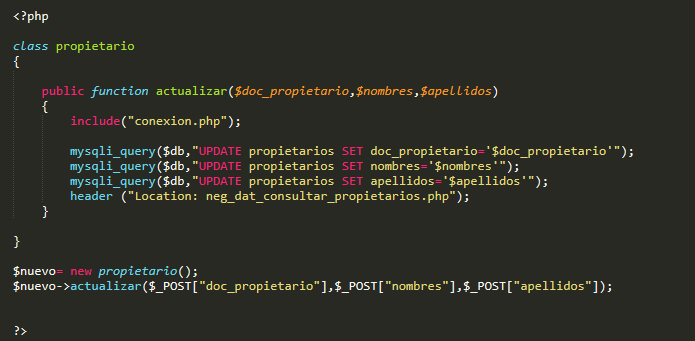
En el siguiente código se empieza a validar tanto el documento como la contraseña junto con condicionales respecto al usuario que inicio sesion en el sistema



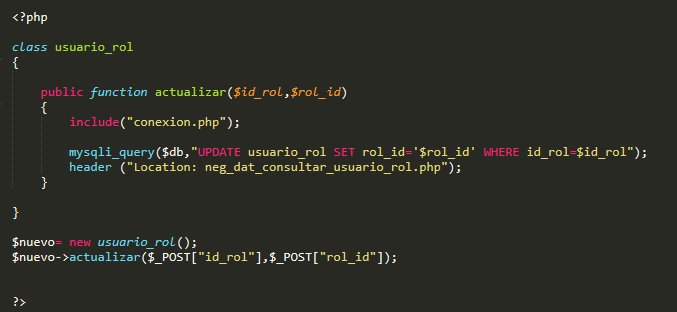
| **Resultados esperados** | **Resultado obtenido** |
| --- | --- |
| La clase permisos debe validar documento,contraseña para acceder al aplicativo donde se realizará la consulta y verificación con la base de datos. | En el aplicativo la función validar hace una consulta en la base de datos donde verifica que la información almacenada sea verdadera y dejar ingresar al usuario permitido |



| **Resultados esperados** | **Resultado obtenido** |
| --- | --- |
| Realiza un conteo de 0 a 1 en donde si dicho resultado es 0 no permite ingresar pero en el caso contrario ya el acceso es correcto. | Al momento de hacer el conteo y de verificar la información el aplicativo permite o niega el acceso al usuario registrado en la base de datos. |

****

| **Resultados esperados** | **Resultado obtenido** |
| --- | --- |
| Al tener que actualizar los campos se debe actualizar dicha información y quedar almacenado en la base de datos | al momento que actualizamos la información muestra el cambio en la tabla de consultas así mismo hace el cambio en la base de datos |



| **Resultados esperados** | **Resultado obtenido** |
| --- | --- |
| Al tener que actualizar los campos de la tabla usuario\_rol se debe ejecutar dicha función en la base de datos la cual almacena la información actualizada previamente. | Se realizarán actualizaciones y cambio de información en la base de datos y se debe mostrar el cambio realizado o ejecutado. |