Eliot Cruz

Código

Equipo 1, pagina web

1. cruz ponce eliot gabriel
2. luna fonseca javier neftali
3. manjarrez carrillo ander osvaldo
4. romero gonzalez roger

**Insertar Grupo**

Creamos un Script llamado “PaginaInsertarGrupo.php”, el cual podrá llamar el WebService de InsertarGrupo. Creamos una función Aceptar que es la que permitirá la conexión al WebService.

Nombramos una variable $cliente que es la que tendrá la dirección del servidor que proporcionara la conexión a el WebService.

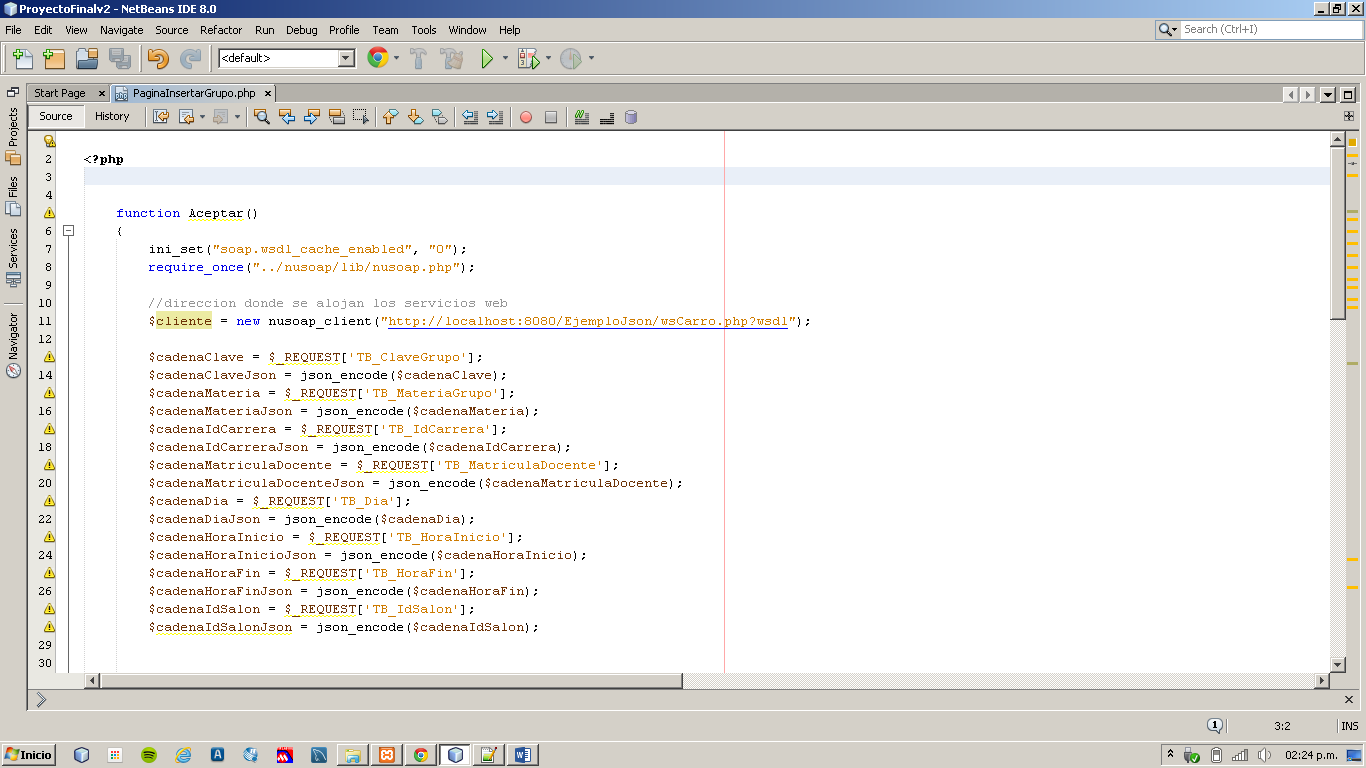
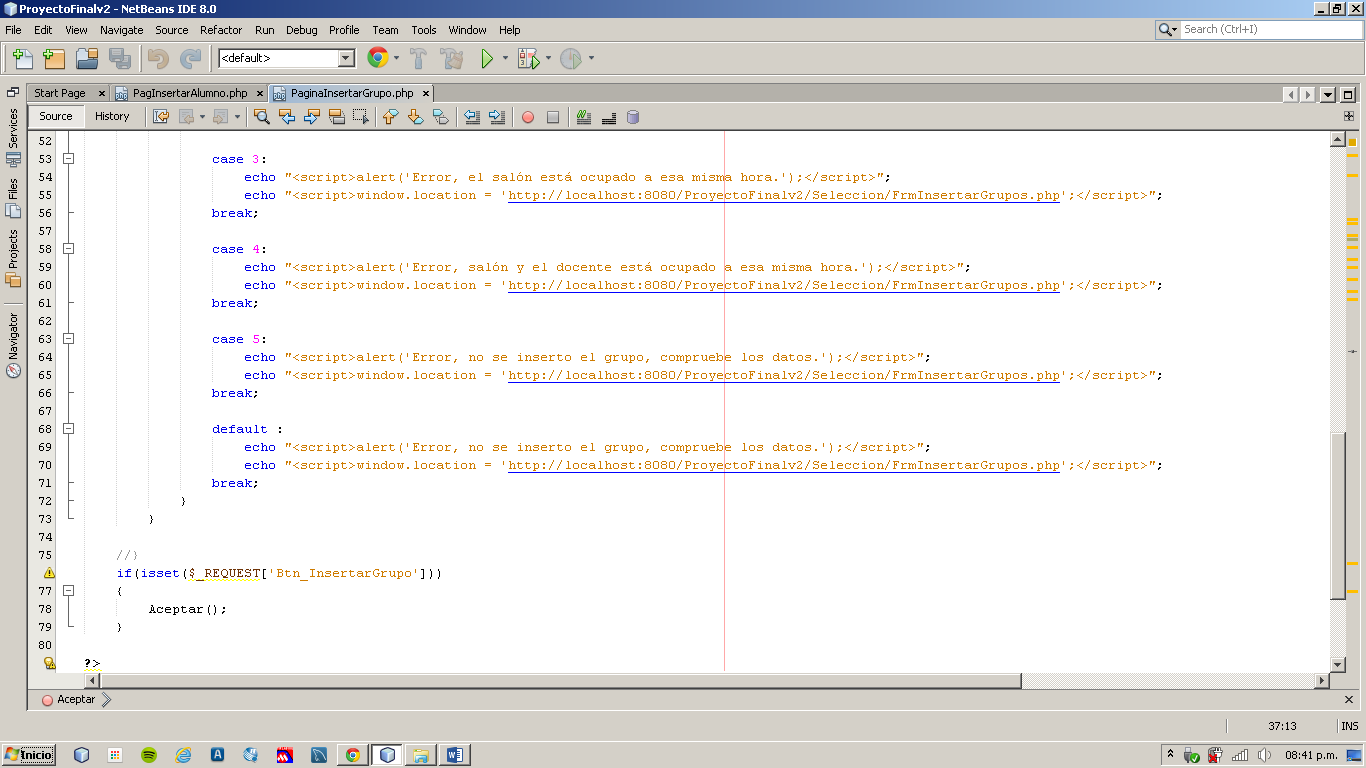
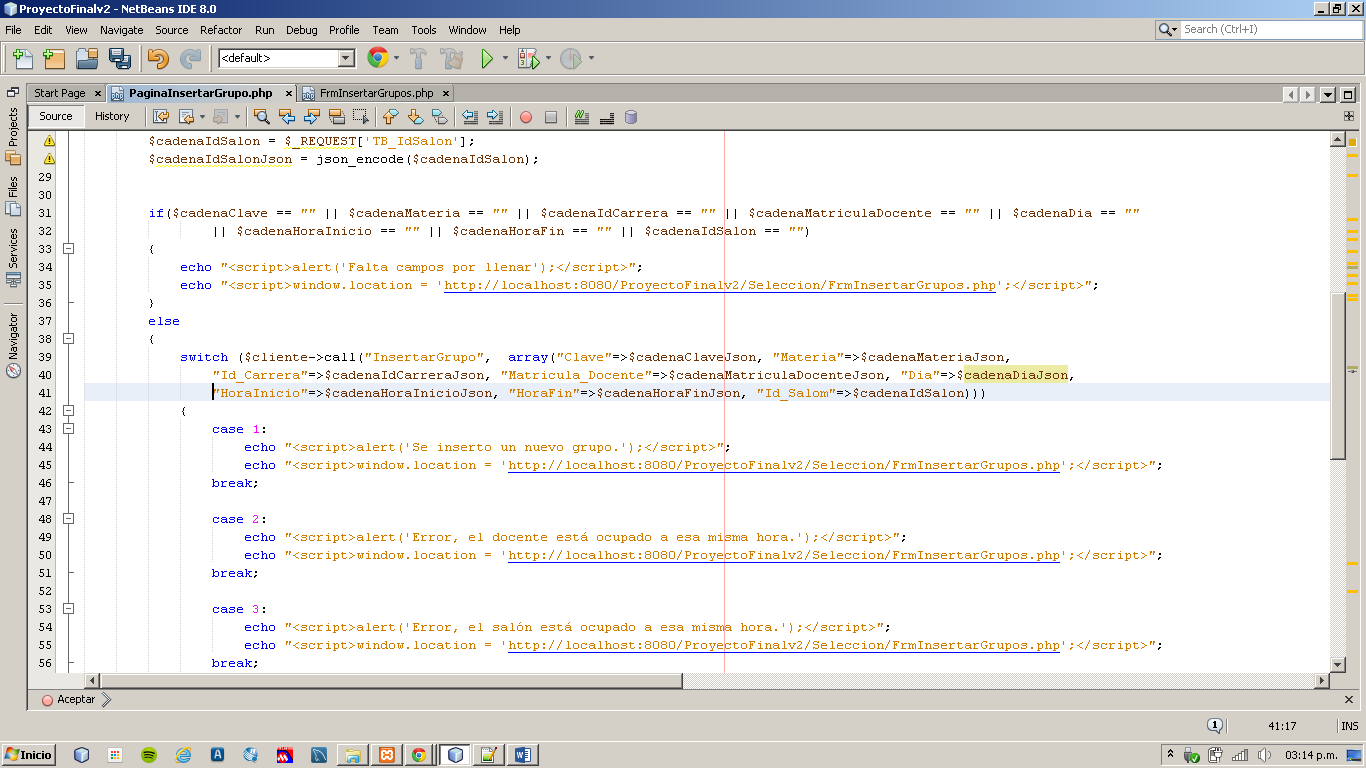
Las variables de la línea 13 a la línea 28, tomaran los valores correspondientes que fueron mandados desde el Html “FrmInsertarGrupo.php”, las encriptamos con Json todas las variables, las cuales mandaremos a la petición.

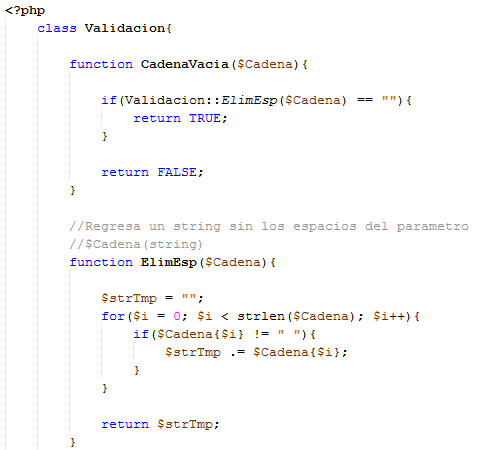
Figura Variables y Función

Primero verificamos si las variables no contienen campos vacíos, en caso de que alguna variable contenga campos vacíos, le indicara al usuario y la petición terminara. En caso contrario, mandara la petición del servicio al WebService junto con las variables que son necesarias para poder insertar un nuevo grupo, en donde habrá 5 respuestas por parte del servicio, en donde indicara si fue exitosa la inserción, donde el docente está ocupado a esa misma hora, el salón está ocupado a esa hora, donde el docente y el salón estén ocupados a esa misma hora y un error general en donde fallo algo en la inserción, cualquier respuesta nos mandara directamente a la página de Insertar Grupo.



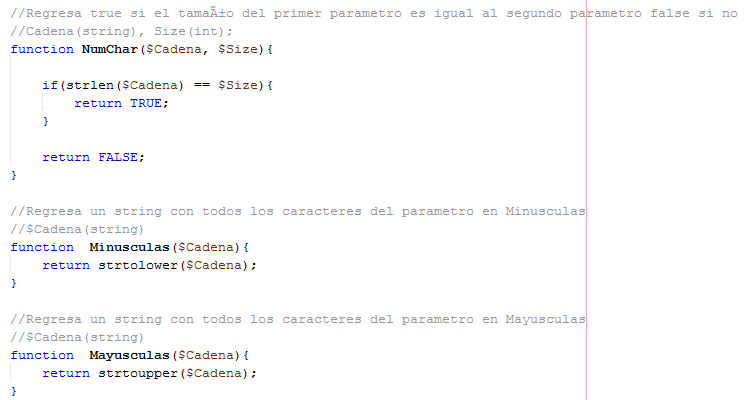
***Página: ClaseValidacion.php***

Esta clase se encarga de realizar diferentes validaciones para que las diferentes páginas utilicen las funciones de esta clase, los métodos son genéricos y la mayoría ayudan a tratar strings.



La función **CadenaVacia** se encarga de verificar que el parámetro que recibe tenga al menos un carácter si lo tiene regresa true si no regresa false.

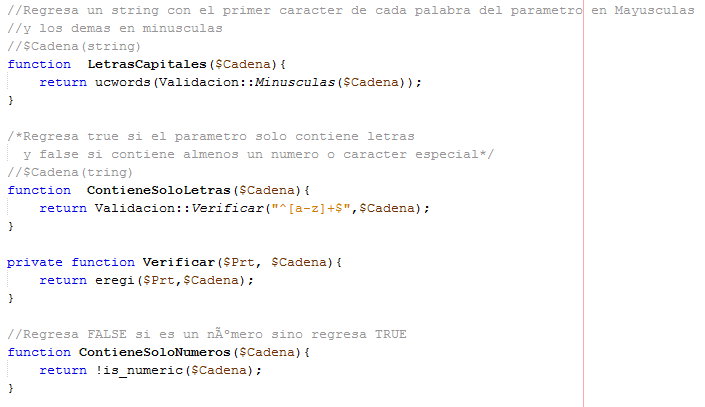
La función **ElimEsp** se encarga de tratar el parámetro que recibe de forma que elimina todos los espacios del mismo dejando solo los demás caracteres.



La función **NumChar** verifica que la longitud del primer parámetro es el segundo parámetro de ser así regresa true caso contrario regresa false.

La función **Minusculas** regresa el string que recibe como parámetro pero convierte todos caracteres a minúsculas.

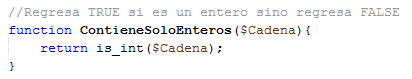
La función **Mayusculas** regresa el string que recibe como parámetro pero convierte todos caracteres a mayúsculas.



La función **LetrasCapitales** regresa un string con el primer carácter en mayúscula del parametro y los demás en minúsculas.

La función **ContieneSoloLetras** verifica que el parámetro que recibe no contenga ningún carácter que no sea una letra y regresa su true en caso de que tenga regresa false.

La función **ContieneSoloNumeros** regresa true si no es un numero en caso contrario regresa false si es un número.



La función **ContieneSoloEnteros** regresa true si el parámetro es un entero en caso contrario regresa false.

***Página: PagActualizarAlumno.php***



Las primeras tres líneas agregan las referencias para poder utilizar la clase Validación codificada por nosotros mismos y los recursos necesarios para trabajar con Web Services.

Las siguientes cinco líneas son declaraciones donde se almacena la información relevante del formulario para después manejar esa información, la información de pasa por el método post o variable súper global post.

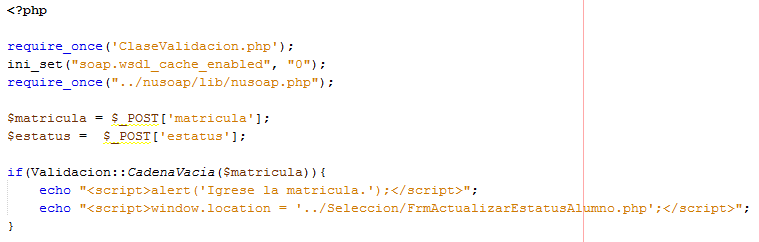
El siguiente if else es para verificar que ningún campo este vacío si mínimo uno lo es muestras un mensaje y regresa a la página ActualizarAlumno.php en caso contrario en el elseif se verifica que los campos o variables Nombre, Apellido Paterno y Apellido Materno contengan solo letras sin acentos en caso de que tampoco se cumpla se ejecuta el else.



Donde se da formato de Letras capitales a las cadenas y se convierte a entero la edad ya que se recibe como string y a continuación se crea una instancia de nusoap con la dirección ip respectiva del cliente para poder invocar el Web Service necesario y se crean nuevas variables donde se cifra o serializa la información recibida de formulario ya almacenadas en las variables anteriores y por último se invoca el web service necesario en este caso ActualizarAlumno y se mandan como parámetros las variables que actualizaran el alumno a partir de la matrícula ya serializadas.

Si el método ActualizarAlumno regresa true se muestra un mensaje donde se informa esto y se regresa al inicio o página principal, en caso contrario se muestra un mensaje donde se muestra el error y se mantiene en la misma página.

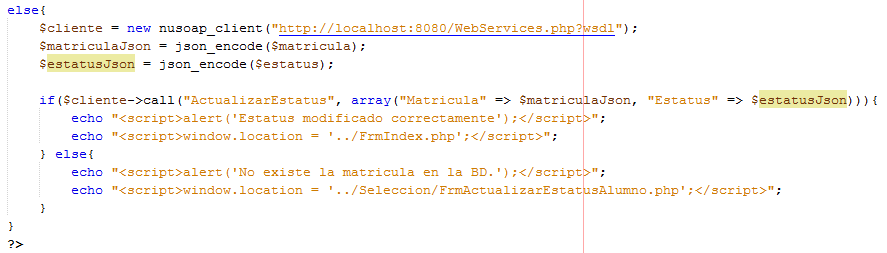
***Página: PagActualizarAlumno.php***



Las primeras tres líneas agregan las referencias para poder utilizar la clase Validación codificada por nosotros mismos y los recursos necesarios para trabajar con Web Services.

Las siguientes dos líneas son declaraciones donde se almacena la información relevante del formulario para después manejar esa información, la información de pasa por el método post o variable súper global post.

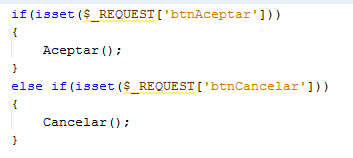
El siguiente if else es para verificar el campo matricula no este vacío si lo es muestras un mensaje donde solicita la pagina y se mantiene en la misma página.



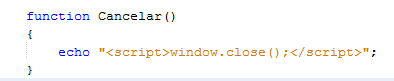
En caso contrario se crea una instancia de nusoap con la dirección respectiva del cliente para poder invocar el Web Service necesario y se crean nuevas variables donde se cifra o serializa la información recibida de formulario ya almacenadas en las variables anteriores y por último se invoca el Web Service necesario en este caso ActualizarEstatus y se mandan como parámetros las variables que actualizaran el alumno a partir de la matrícula ya serializadas.

Si este método regresa true se muestra un mensaje donde se notifica y se regresa al inicio o página principal en caso de contrario se muestra un mensaje notificando el error y se mantiene en la misma página.

Descripción de código LOGIN.



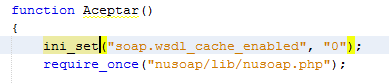
Primero se localiza el botón que fue presionado el de cancelar sesión o iniciar sesión una vez localizado se manda llamar el método correspondiente.



Si el usuario presiona cancelar, se irá al método cancelar y la página se cerrará utilizando código de java script.



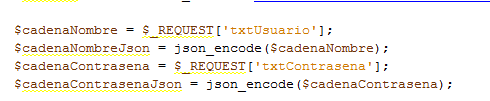
Esta línea de código funciona para mandar llamar la clase Validar Función que se hizo para validación estándar del programa.



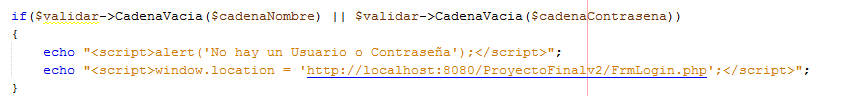
Si el usuario inicia sesión empieza el método aceptar, las dos primeras líneas se usan para llama las librerías del nusoap que se usan dentro del programa para la conexión con el servidor web.



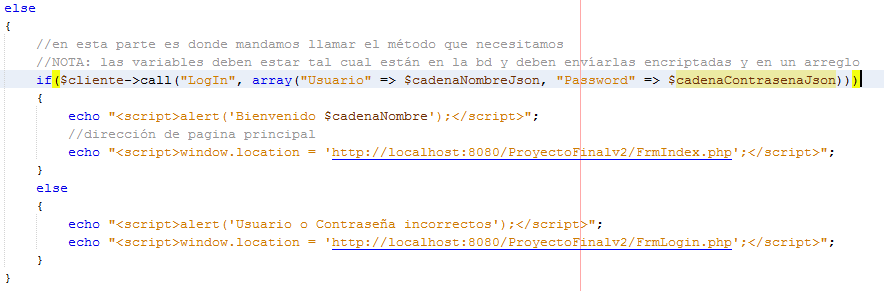
Se Crea una nueva instancia de conexión con el servidor donde está alojado el servicio web, dando como parámetro la dirección y la ruta donde está el servicio.



Se crean variables con los datos de las text box y de encriptan con Json, para la seguridad de los mismos al momento de enviarlos al servidor.

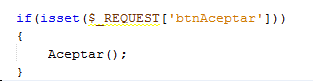


Se utilizan los métodos de la clase Validar para validar que las text box no estén vacías. En caso de que no haya datos en las text box, manda un mensaje con java script de que no hay datos y se direcciona otra vez la página principal.



En caso de haber datos en las text box manda llamar un método que está alojado en el servicio web. Este método regresa un true si los datos son correctos y un false sino lo son. Si los el método retorna un true entonces se redirecciona a la página principal del proyecto, en caso contrario manda un mensaje de error.

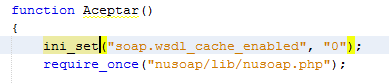
Descripción de código Insertar Alumno.



Valida si el botón Aceptar se pulso, de ser cierto manda llamar la función Aceptar.



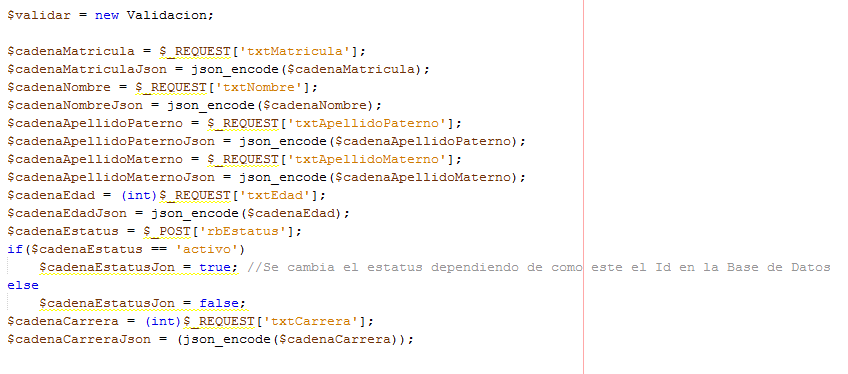
Esta línea de código funciona para mandar llamar la clase Validar Función que se hizo para validación estándar del programa.



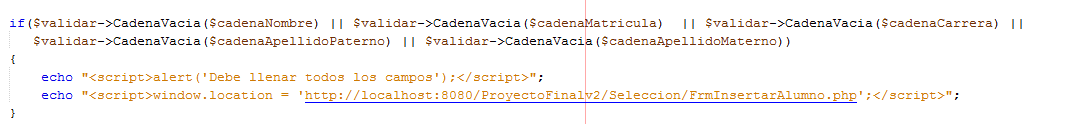
Si el usuario inicia sesión empieza el método aceptar, las dos primeras líneas se usan para llama las librerías del nusoap que se usan dentro del programa para la conexión con el servidor web.



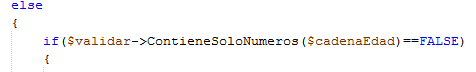
Se Crea una nueva instancia de conexión con el servidor donde está alojado el servicio web, dando como parámetro la dirección y la ruta donde está el servicio.



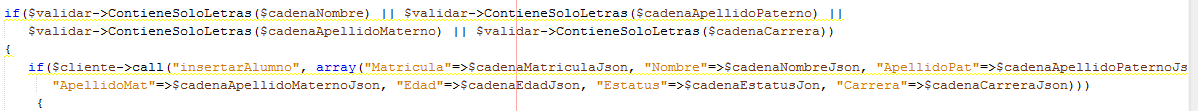
Se crea una instancia de la clase validación para poder usarla en el programa. Se obtienen los datos de cada text box se crea una variable extra para encriptarlos con Json.



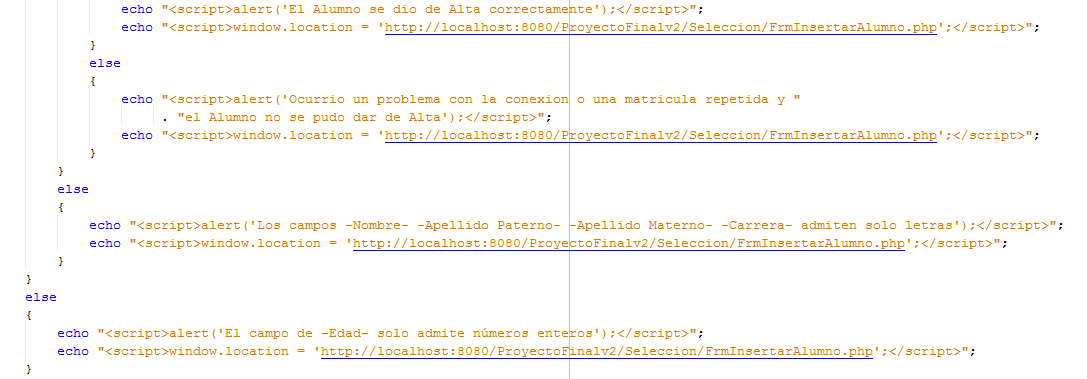
Se valida que las cajas de texto tengan texto, que no estén vacías. En caso de que estén vacías manda un mensaje.



En caso de que las text box tengan texto entra a otra validación que es, validar si la edad contiene solo números si contiene solo números el método regresa un false y entra a la siguiente validación



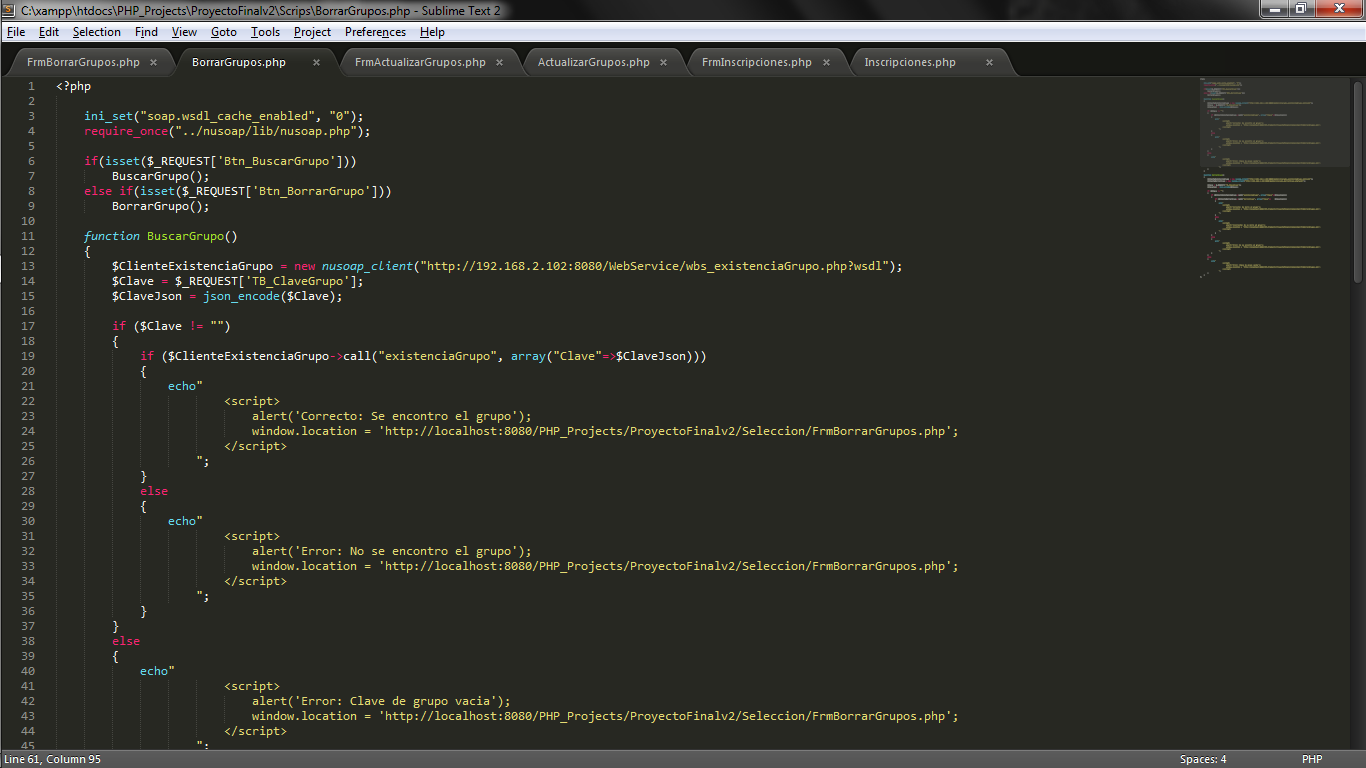
La siguiente validación es para que las textbox contengan solo letras en caso de ser verdad, manda llamar el método insertar Alumno y le manda en un string encriptado con json todos los valores que se están usando.



En caso de que no haya entrado alguna vez manda un mensaje de error por cada parte a la que no entro. Es validación básica de letras y números de las text box.

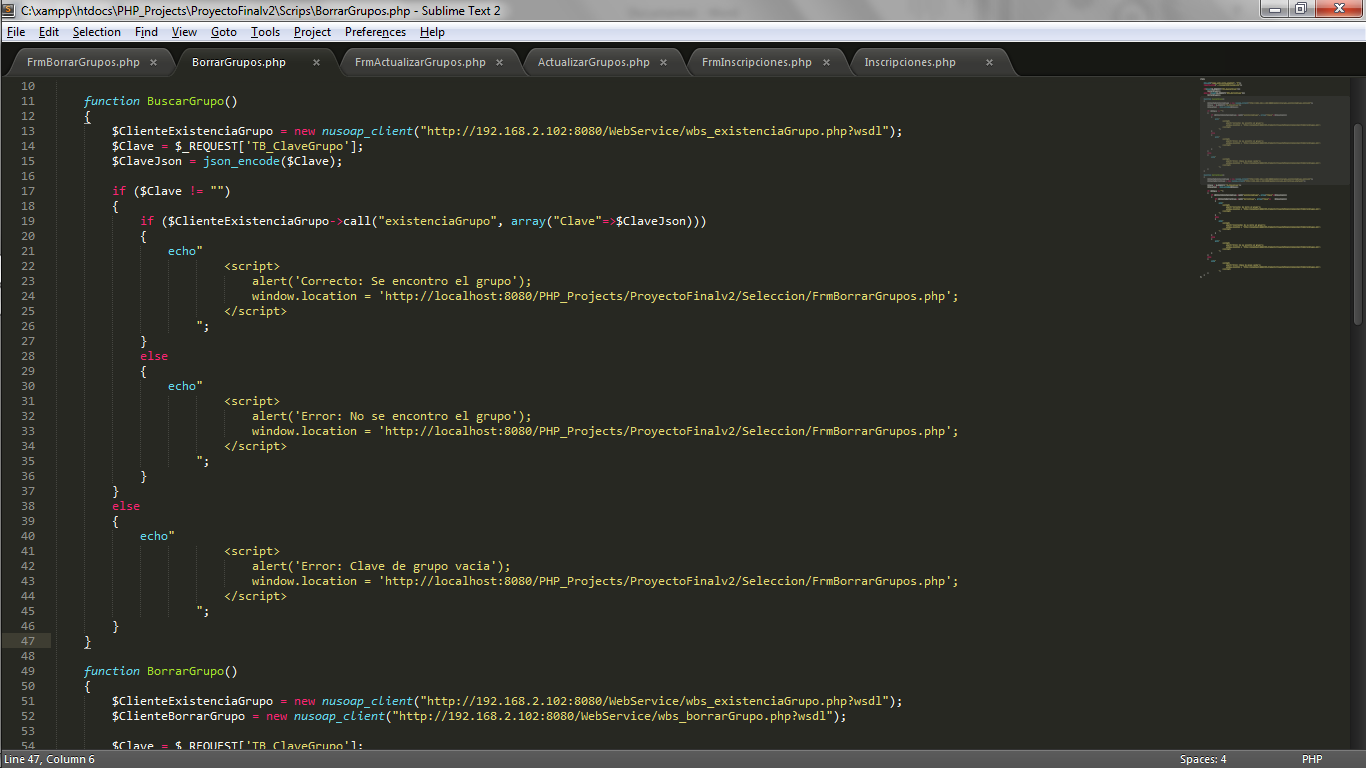
**Borrar grupo.**

Buscar Grupo



Líneas 3 y 4: Referencias para poder invocar los servicios web.

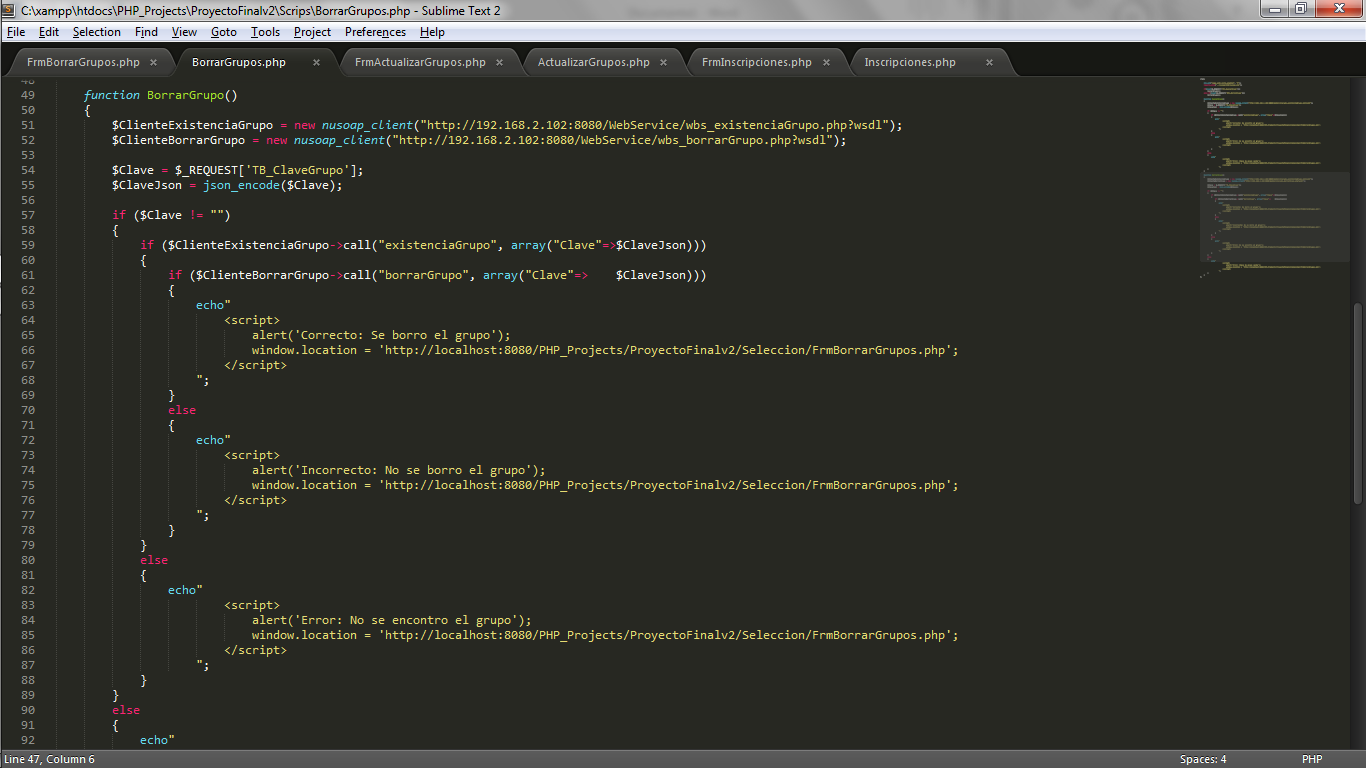
Línea 6 a 9: Verifica a que botón se le dio clic en el formulario.



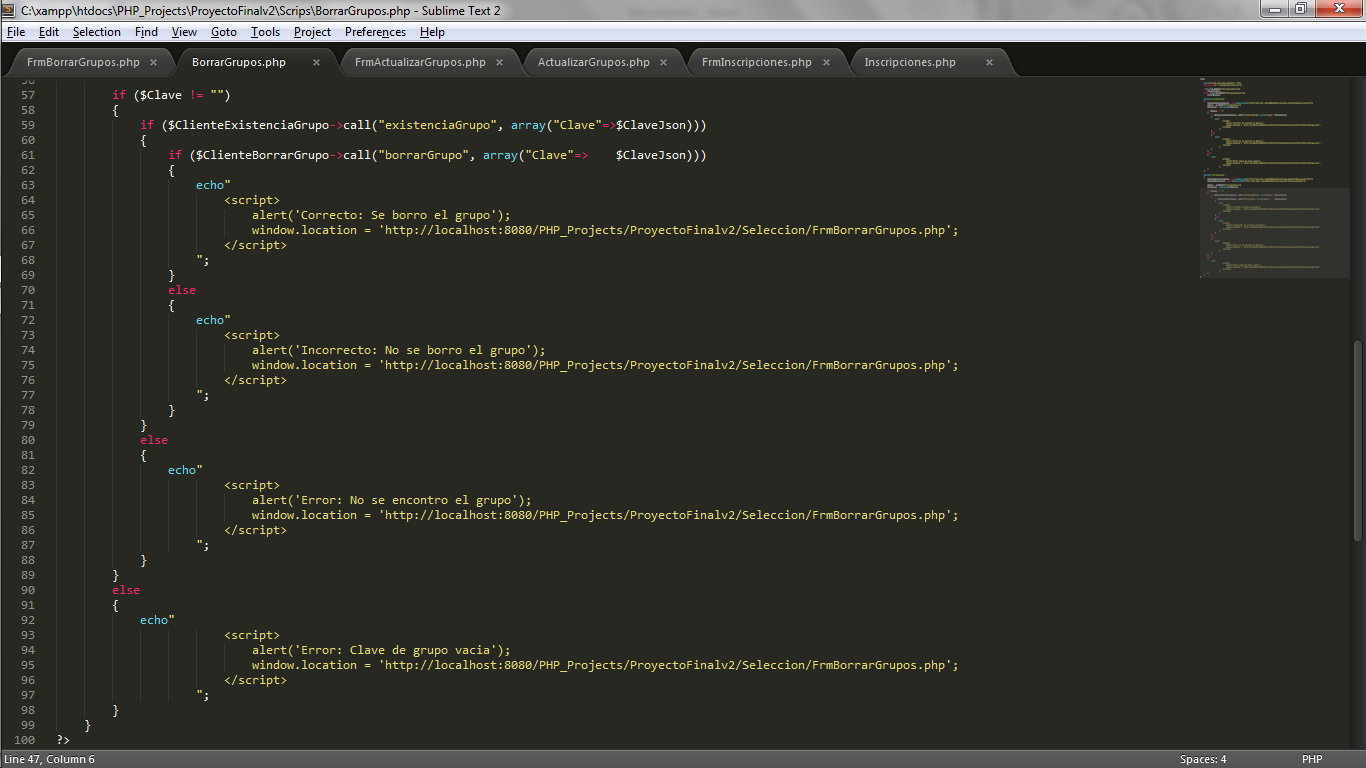
Línea 11 a 47:

* Línea 13 a 15: Creación de la instancia con la dirección donde se aloja el servicio web.
* Línea 17 a 46: Se verifica primero si la clave del grupo no está vacía, si esta vacía mandara un mensaje de error, si no está vacía se verifica la existencia del grupo que se está buscando, si el grupo se encuentra se mandara un mensaje de que el grupo se encontró si no, mandara un mensaje indicando que el grupo no se encontró.

Borrar grupo

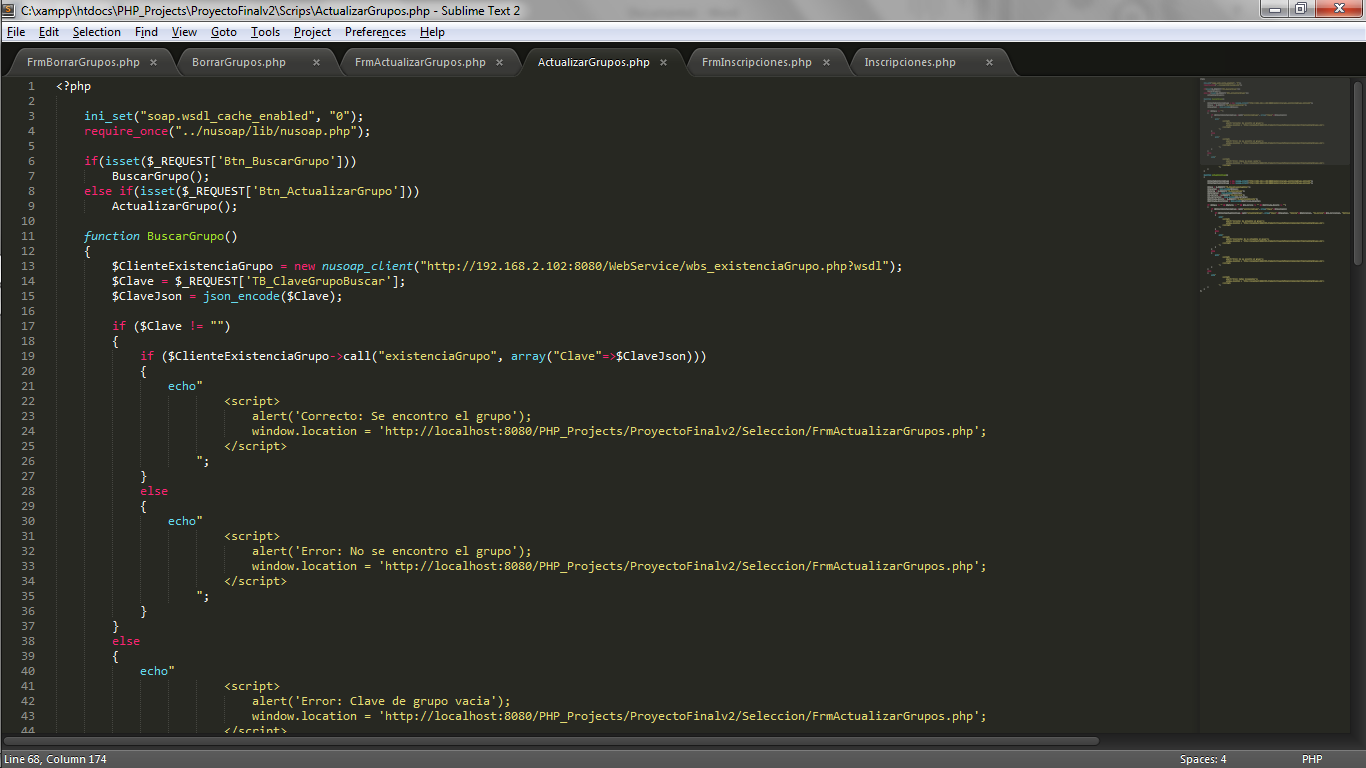


Linea 51 a 55: Creación de las instancias con la dirección donde se aloja el servicio web y creación de variables necesarias.



Línea 57 a 98: Se verifica si la clave del grupo que queremos borrar, si la clave esta vacia se mandara un mensaje de error, si la clave no está vacía se verifica la existencia del grupo que queremos borrar, si el grupo no existe, se mandara un mensaje de error indicando que el grupo que se quiere borra no existe, y si el grupo existe se borra y se manda el mensaje de que el grupo se borro conectamente.

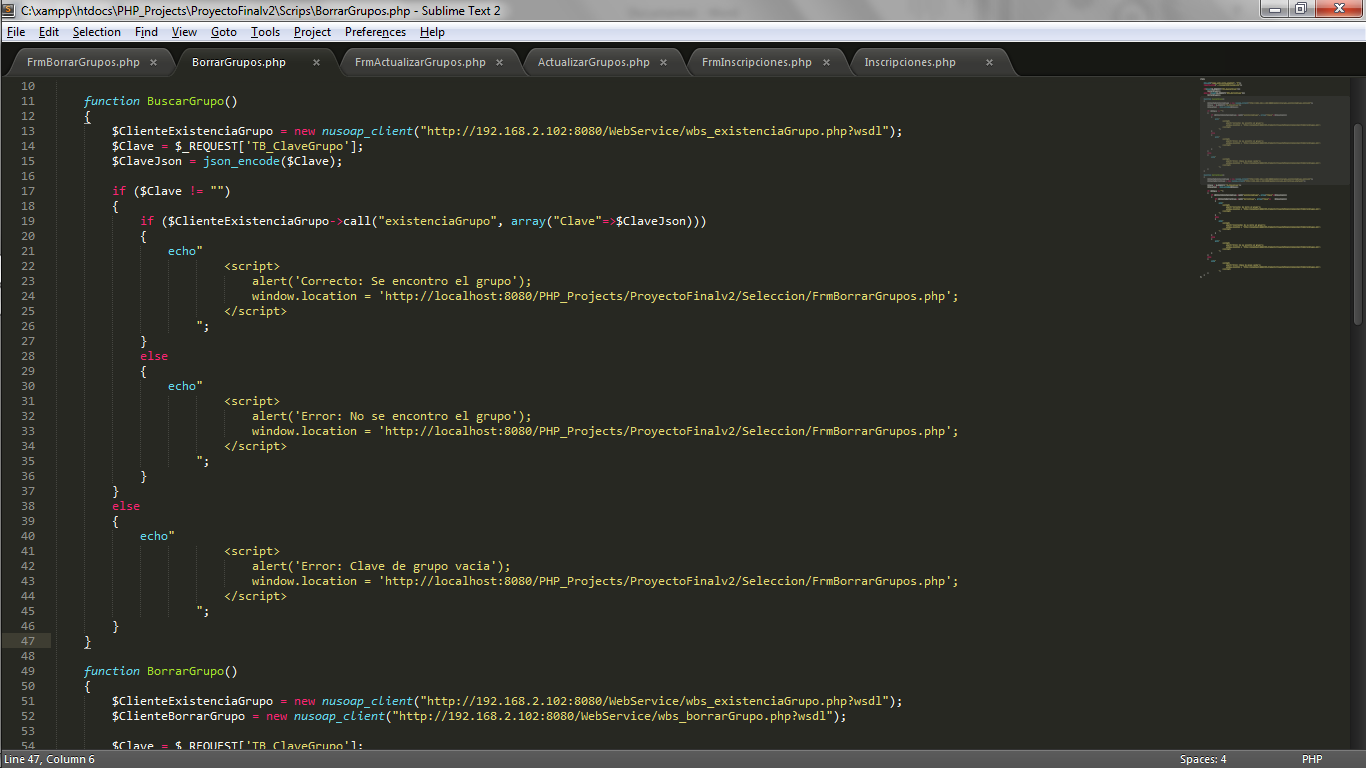
**Actualizar grupo**



Líneas 3 y 4: Referencias para poder invocar los servicios web.

Línea 6 a 9: Verifica a que botón se le dio clic en el formulario.

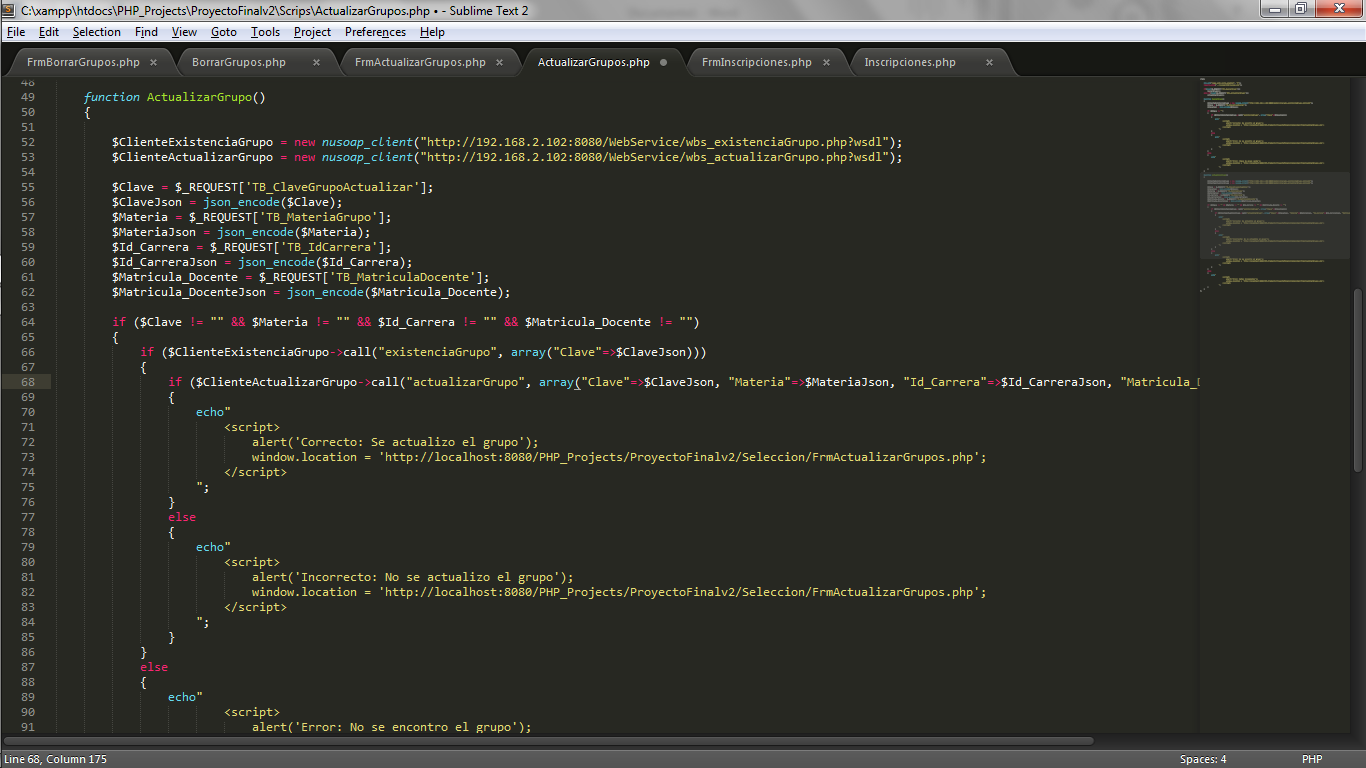
Buscar Grupo



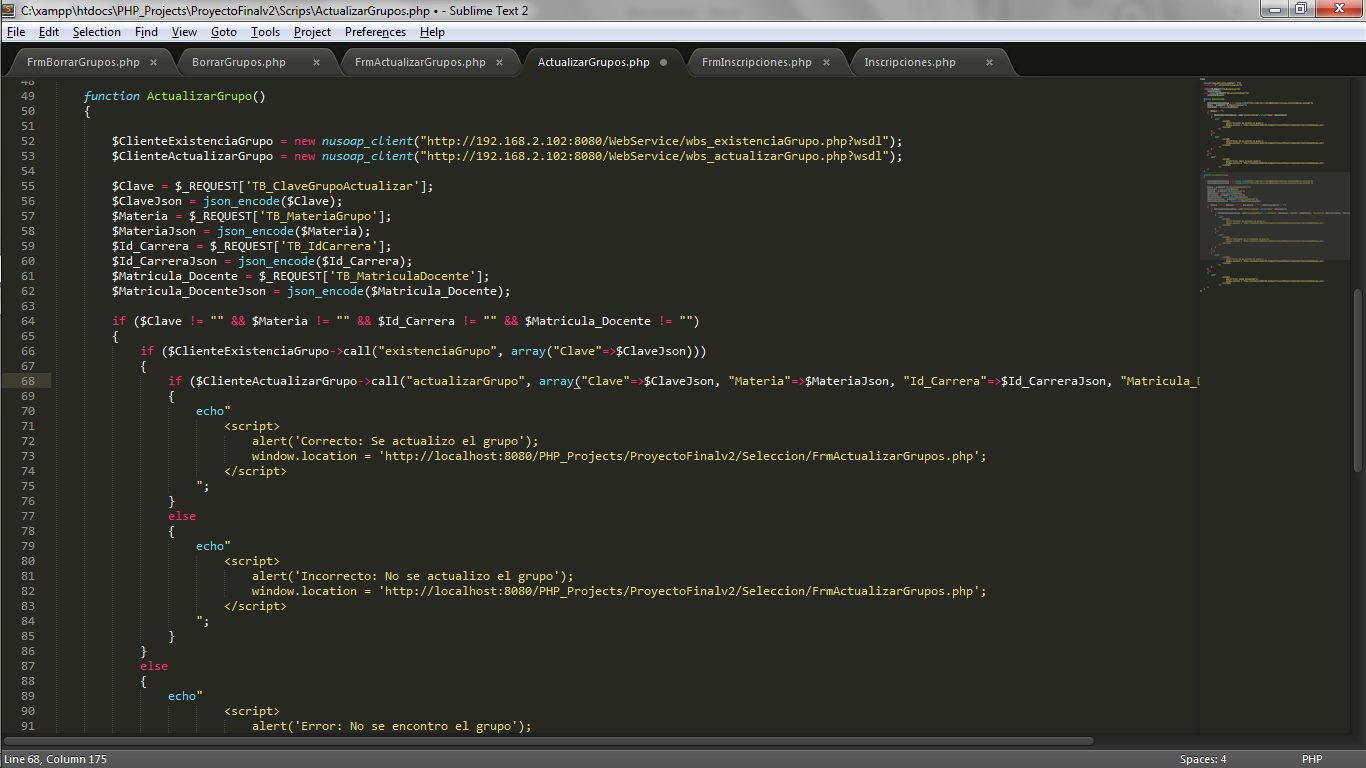
Línea 11 a 47:

* Línea 13 a 15: Creación de la instancia con la dirección donde se aloja el servicio web.
* Línea 17 a 46: Se verifica primero si la clave del grupo no está vacía, si esta vacía mandara un mensaje de error, si no está vacía se verifica la existencia del grupo que se está buscando, si el grupo se encuentra se mandara un mensaje de que el grupo se encontró si no, mandara un mensaje indicando que el grupo no se encontró.

Actualizar Grupo

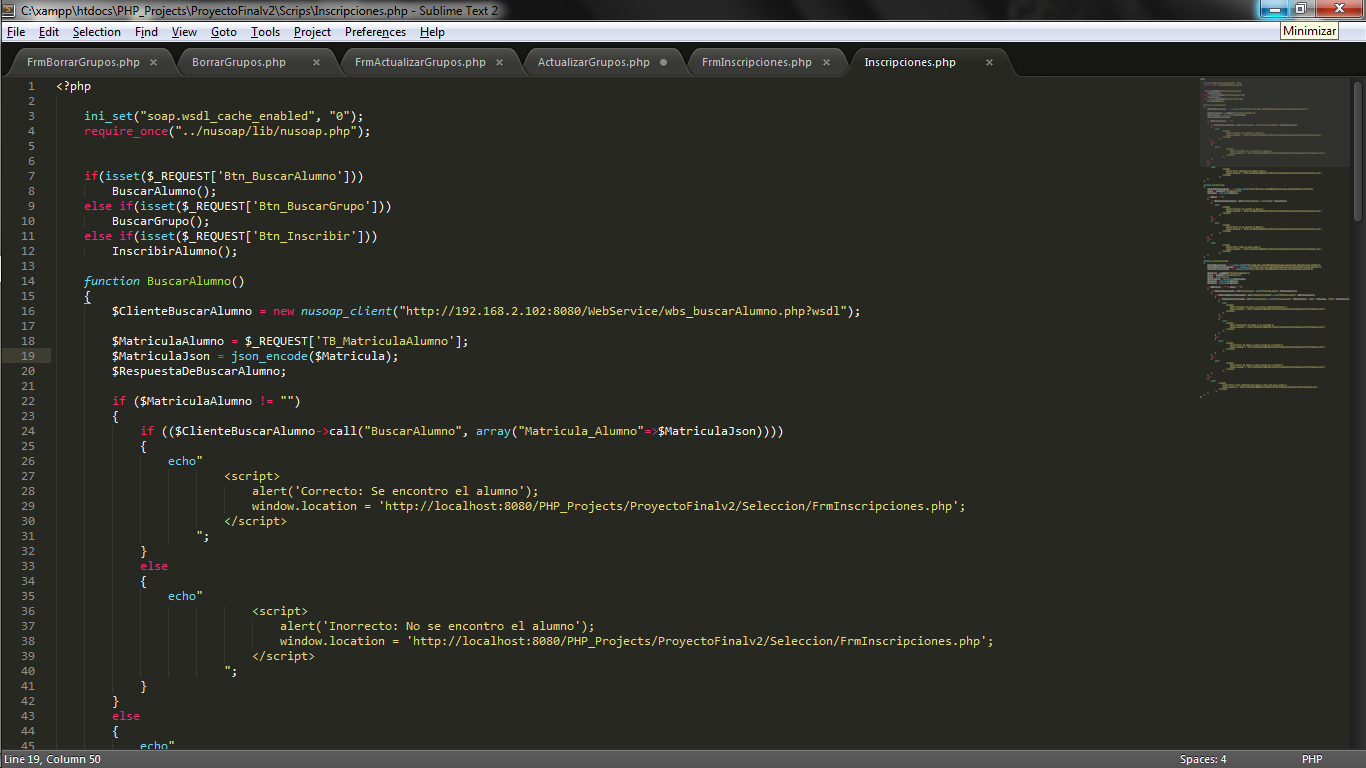


Línea 52 a 62: Creación de las instancias con la dirección donde se aloja el servicio web y creación de variables necesarias.



Línea 64 a 106: Se verifica que la clave, materia, Id de la carrera y matricula del docente a actualizar en el grupo no estén vacías, si están vacías se mandara un mensaje de error, indicando que los datos están incorrectos, si los datos están completos se verifica la existencia del grupo que se quiere actualizar, si el grupo no existe, se mandara un mensaje de error indicando que el grupo no se encuentra, si el grupo si se encuentra se actualiza y mandara un mensaje de que el grupo se actualizo.

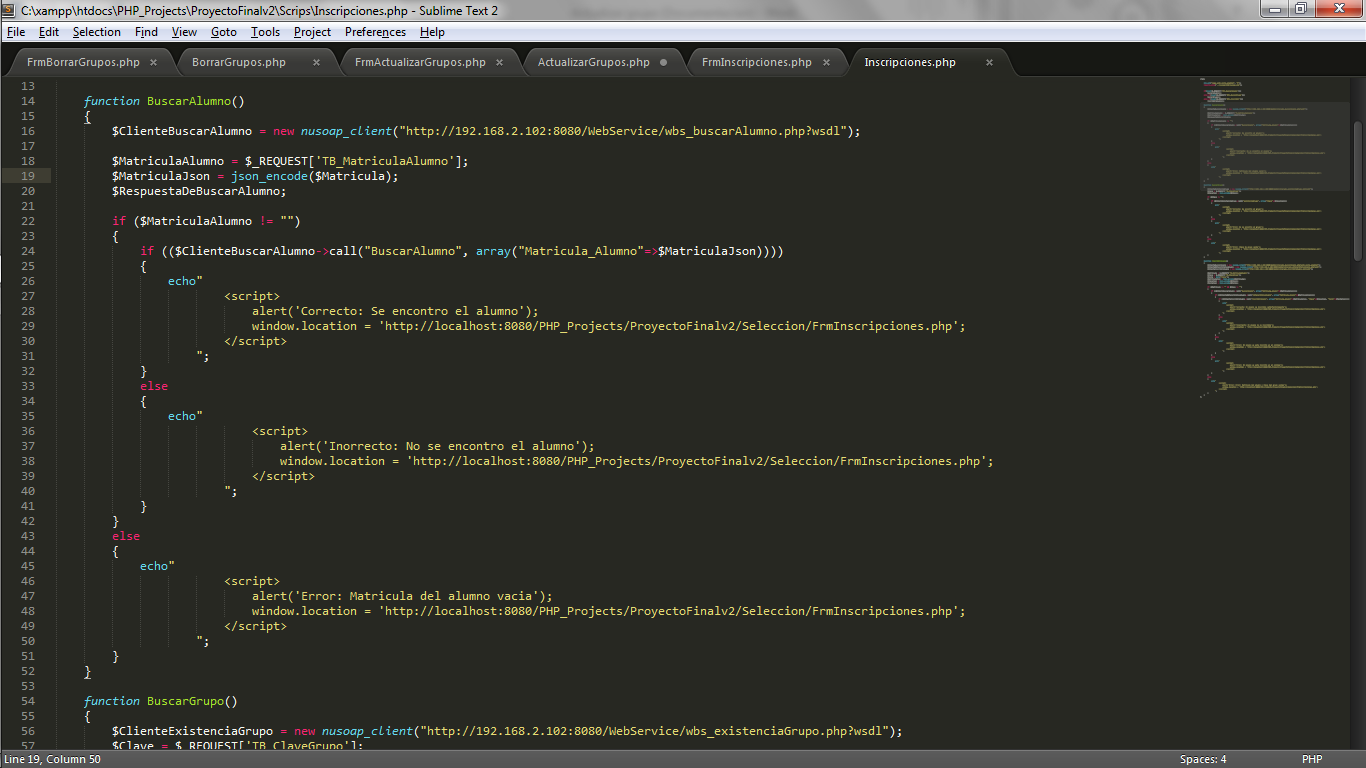
**Inscribir**



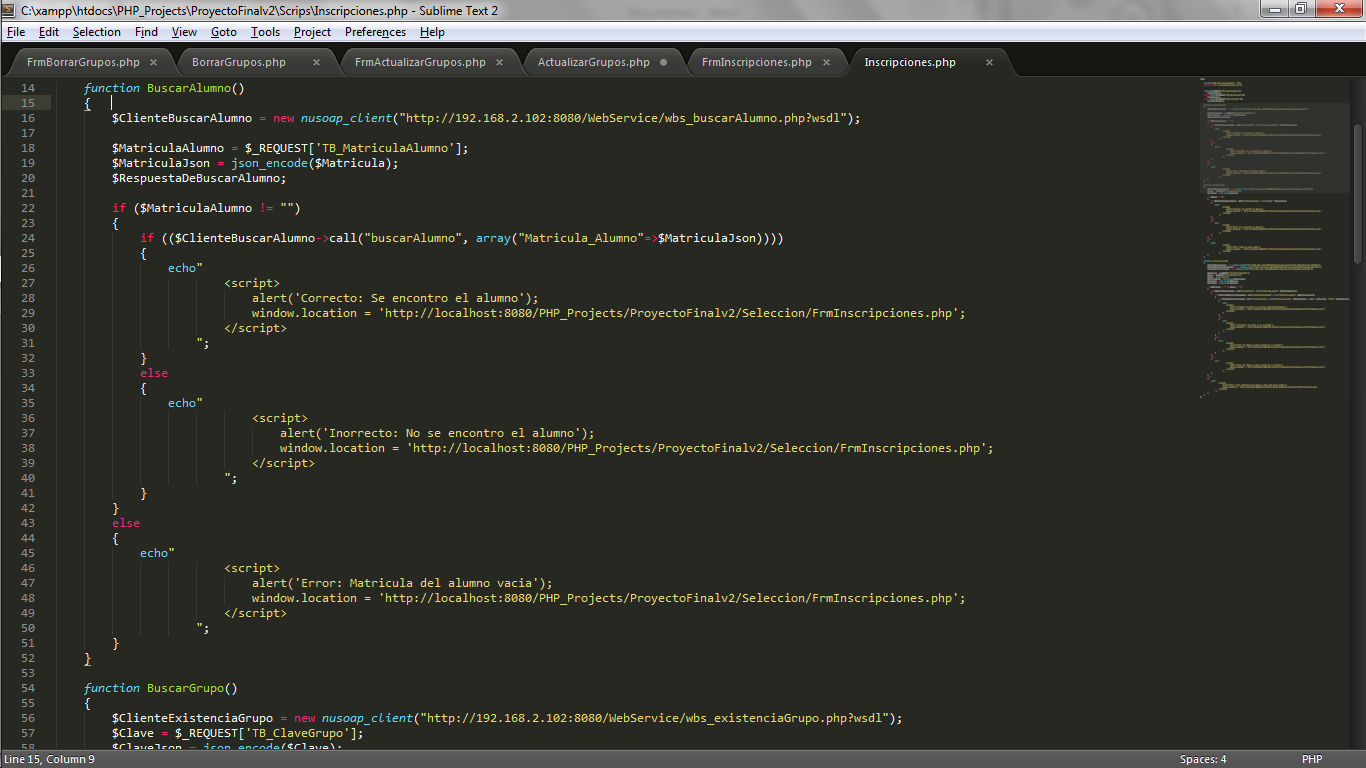
Líneas 3 y 4: Referencias para poder invocar los servicios web.

Línea 7 a 12: Verifica a que botón se le dio clic en el formulario.

Buscar Alumno

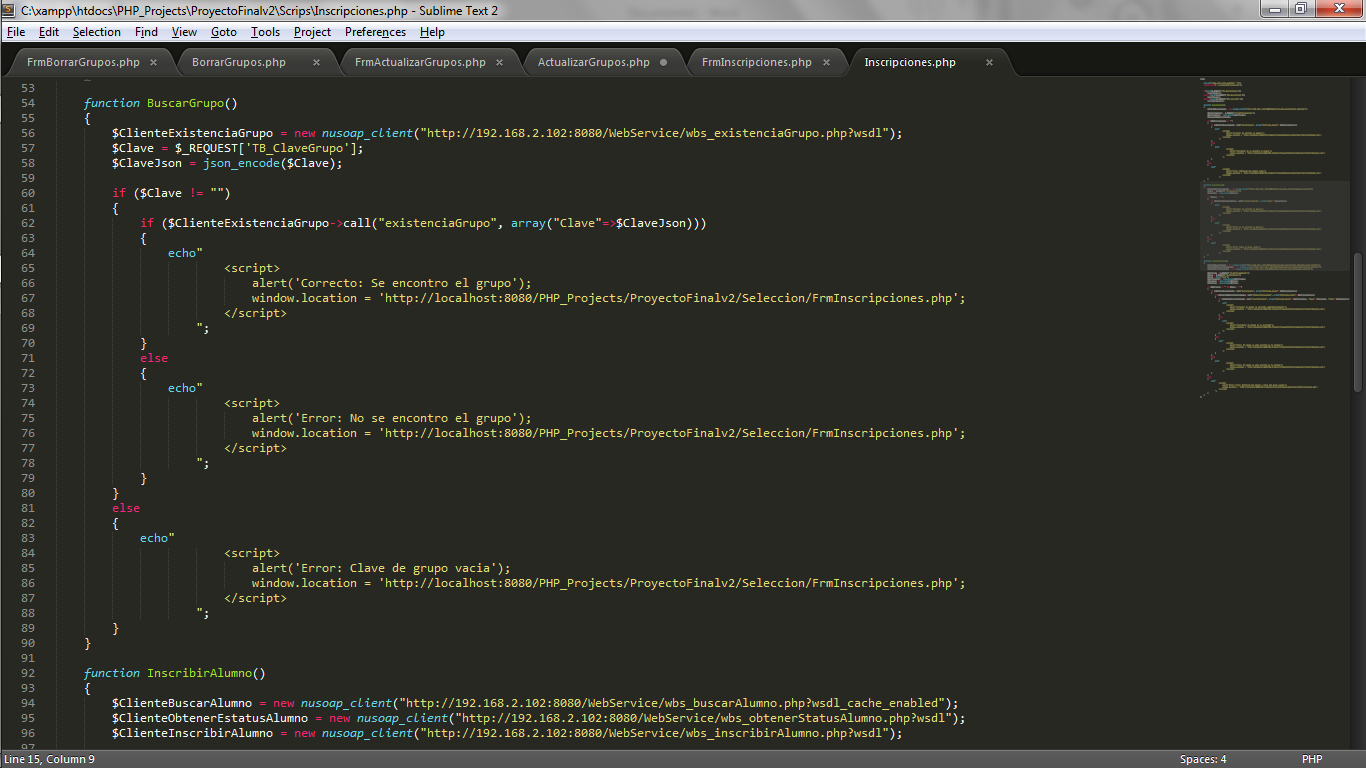


Línea 16 a 20: Creación de las instancias con la dirección donde se aloja el servicio web y creación de variables necesarias.



Línea 22 a 52: Se verifica que la matricula del alumno que queremos buscar no este vacía, si está vacía se mandara un mensaje de error, si no está vacía se verificara la existencia del alumno, si el alumno se encuentra se mandara mensaje de que se encontró el alumno, en caso contrario se mandara mensaje de que no se encontró el alumno.

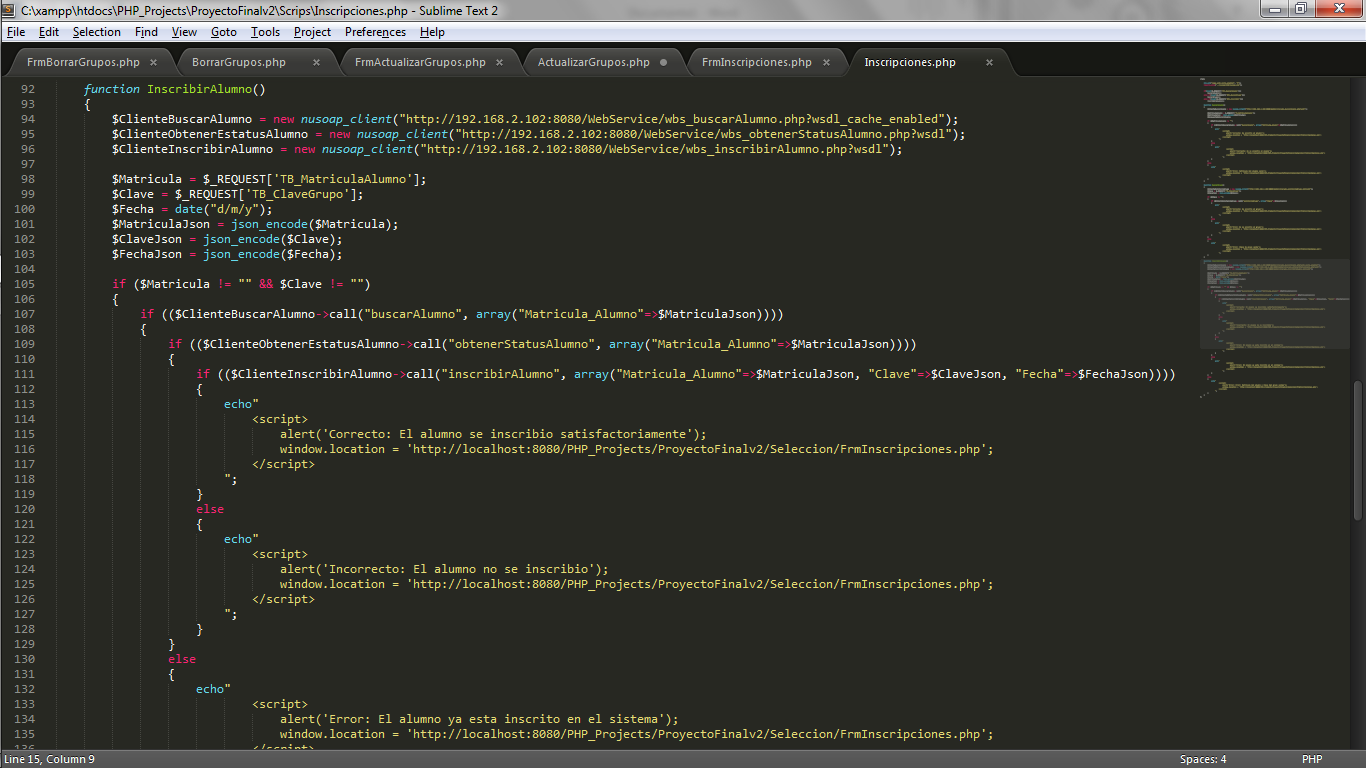
Buscar grupo



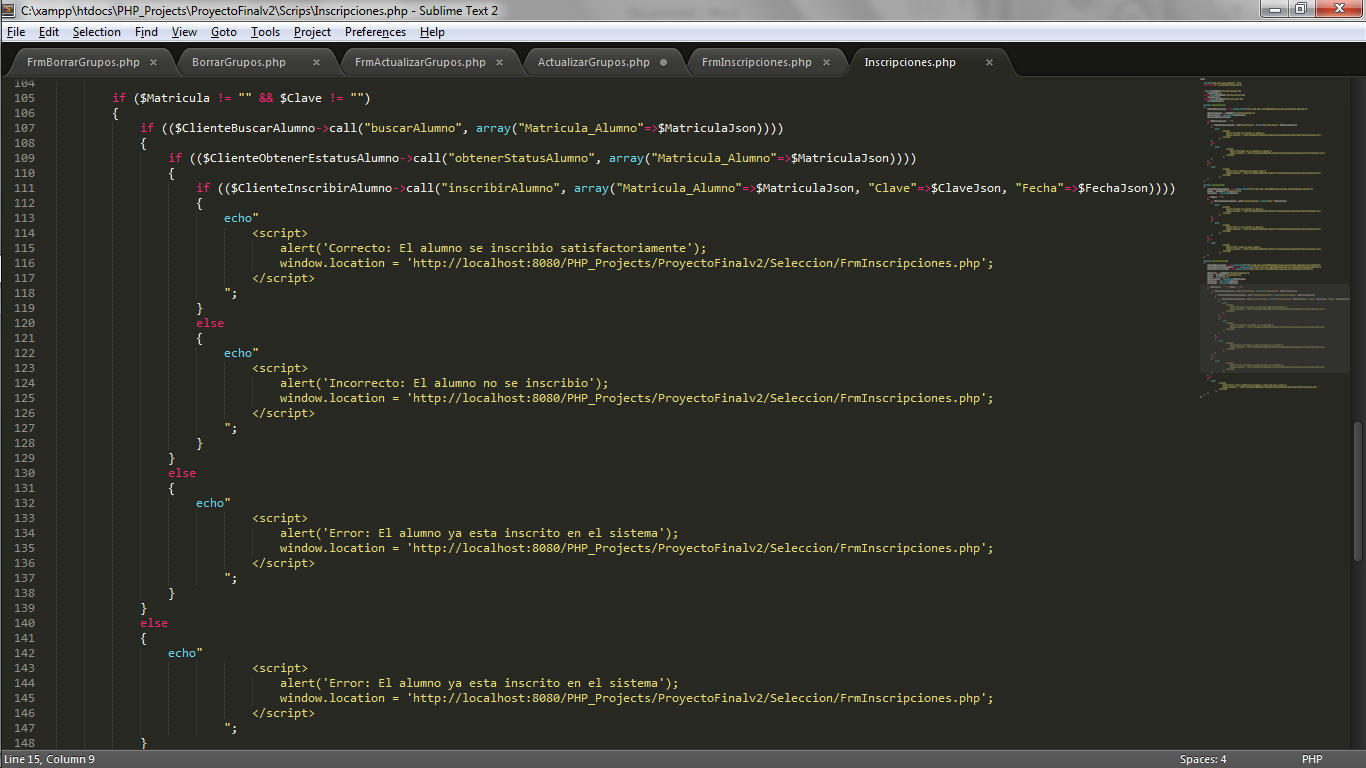
Línea 56 a 58: Creación de la instancia con la dirección donde se aloja el servicio web y creación de variables necesarias.

Línea 60 a 90: Se verifica que la clave del grupo que se quiere buscar no está vacía, si la clave está vacía se mandara un mensaje de error, sin no está vacía, se verificara la existencia del grupo, si el grupo se encontró, mandara un mensaje indicando lo anterior dicho, en caso de no encontrarse el grupo, de igual manera se mandara el mensaje indicando que el grupo no se encontró.

Inscribir alumno



Línea 94 a 103: Creación de la instancia con la dirección donde se aloja el servicio web y creación de variables necesarias.



Línea 105 a 149: Se verifica que la matricula del alumno y la clave no se encuentren vacíos, si alguno de los 2 datos o los 2 están vacíos, se mandara un mensaje de error, si no están vacíos se verificara el estatus del alumno, si el alumno ya está inscrito en un grupo, mandara un mensaje de error indicando lo anterior dicho, si el alumno no está inscrito en algún grupo inscribirá al alumno en el grupo indicado y mandara un mensaje si se inscribió o si no se inscribió el grupo