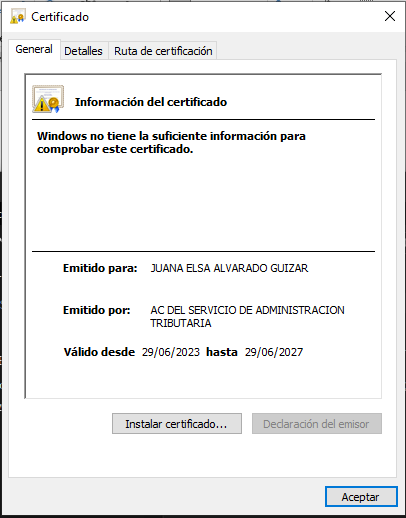
Guía de desarrollador

El principal propósito de este sistema es prevenir el vencimiento de los certificados digitales de la firma o el sello.

Un certificado digital es un archivo con extensión .cer como el siguiente:



Al abrirlo se mira de la siguiente manera:



Como se puede observar, el certificado tiene una fecha de vencimiento.

En el caso de la firma y sellos digitales, la vigencia es de 4 años desde que se hace la renovación o trámite.

El sistema de Asesorías RG se encarga de tener medidos esos tiempo para poder avisar al personal encargado y que puedan tomar medidas al respecto.

La forma en la que el sistema previene un vencimiento es enviar un correo semanalmente con un reporte de Excel en donde muestra los clientes cuyo certificado o certificados están a punto de expirar con una semana de antelación.

Los datos a guardar el certificado son: RFC, nombre del titular, fecha de vencimiento, status y tipo (firma o sello).

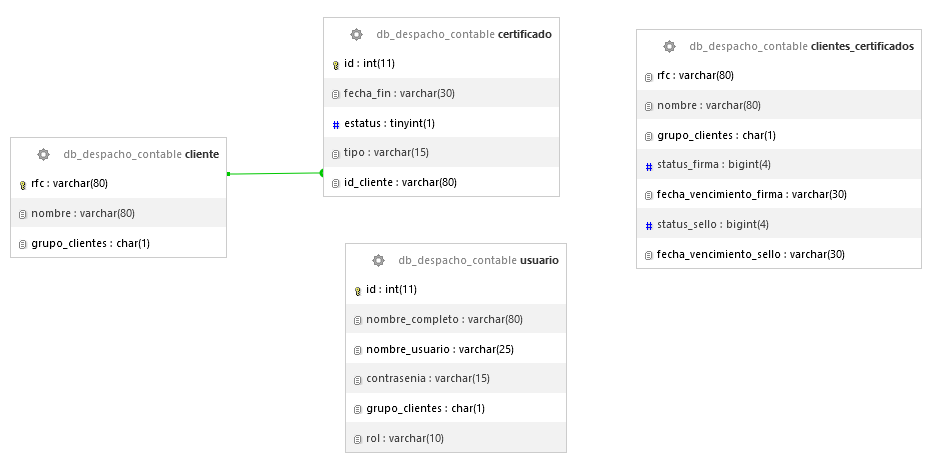
En el despacho los clientes se tienen organizados por grupos (Clientes A, Clientes B y Clientes C) así que tanto los usuarios como clientes es necesario darle un grupo de clientes para tener un mejor orden en el sistema.

Para administradores, podrán ver todos los grupos de clientes así como los usuarios registrados en la base de datos.

Adicionalmente para desarrolladores, se tendrá el servicio “Auto Update Service” del cual se hablará más adelante.

1. **Base de datos:**

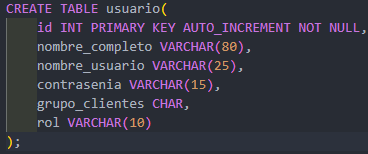
**1.1 El diagrama ER de la base de datos es la siguiente.**



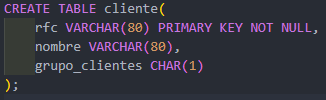
En donde la tabla “cliente” y “certificado” tienen una relación 1:2, la tabla “usuario” contiene todos los registros de las personas que pueden entrar al sistema y “clientes\_certificados” es una vista que muestra de manera más simplificada la relación cliente-certificado.

**1.2 Código de las tablas.**

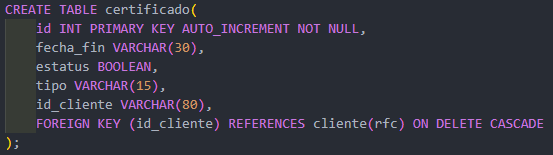
Para la tabla “usuario” en el campo “nombre\_completo” se hizo de 80 caracteres ya que hay clientes que contienen nombres muy largos y era necesario ponerlo así.



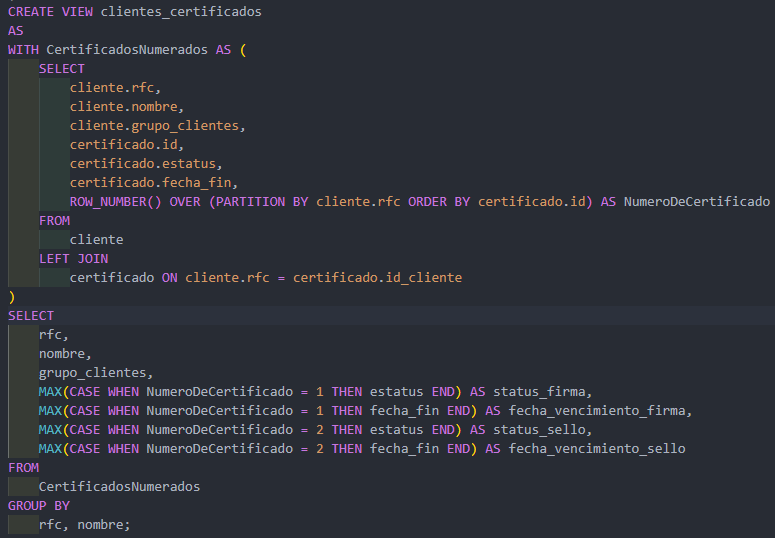
En la tabla “cliente” se define como llave primaria el campo “rfc” definiéndolo con un largo de 80 caracteres porque hay clientes que cuentan con dos tipos de rfc.



Para la tabla “certificado” se guarda la fecha usando el tipo VARCHAR ya que los datos del certificado vienen en otro formato.

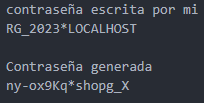


Por último, para la vista “clientes\_certificados” se define de la siguiente manera:



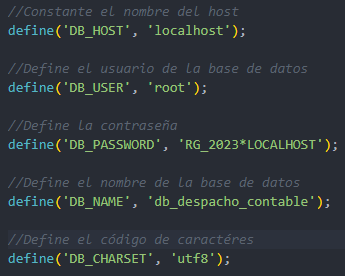
En donde se seleccionan los certificados para mostrarlos en orden, de manera que se pueda diferenciar el sello de la firma.

**Nota: Las credenciales para entrar al DBMS en apache, se usa el usuario root y las siguientes contraseñas:**

****

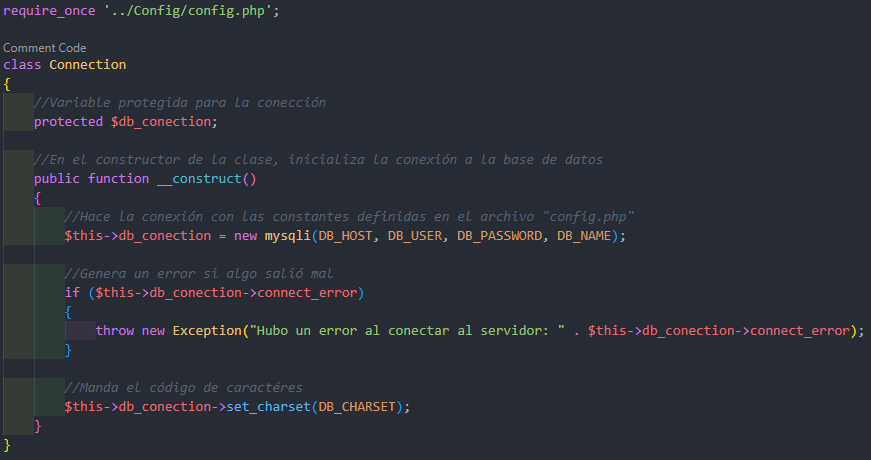
**1.3 Configuración para la conexión del sistema con la base de datos.**

Si acabas de instalar el proyecto, dentro de la ruta “App/Config/” se encuentra un archivo llamado “config-Template.php” el cual se tiene que duplicar, llamar a la copia “config.php” y modificar las constantes definidas en el archivo. La estructura del archivo es la siguiente:



Una vez hecho esto, procedo a explicar la estructura de la conexión.

En la ruta “App/Libraries/” hay un archivo llamado “Connection.php” el cual contiene la lógica necesaria para realizar el enlace a la base de datos. Su estructura es la siguiente:



Esta clase heredará a todos los servicios en el proyecto y, por lo tanto, los constructores de estos servicios usarán el constructor de la clase padre y usarán la propiedad protegida “$db\_conection” para poder hacer modificaciones o lectura a la base de datos.

Cómo se puede observar, la clase “Connection” usa el archivo “config.php” para de ahí tomar las constantes y realizar la conexión.

**2.0 Servicio de usuarios**

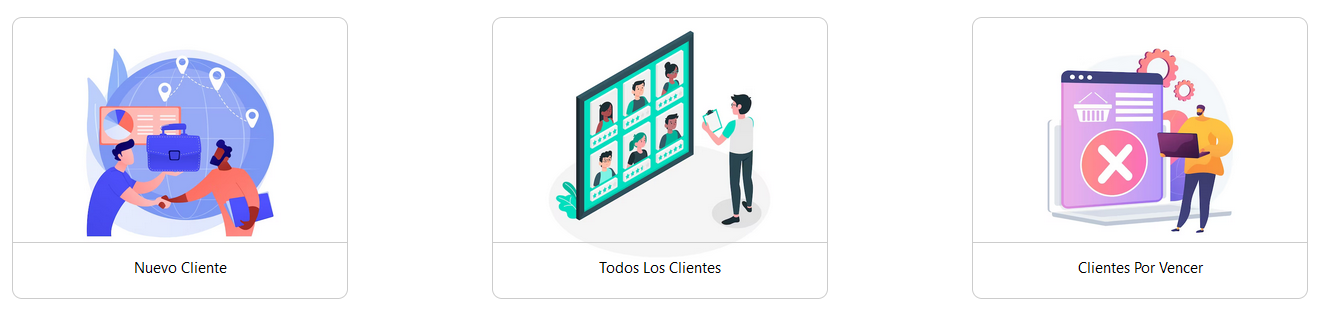
El servicio tiene la siguiente estructura:

****

En la imagen anterior se pueden observar dos sub servicios, adicionalmente se tiene el sub servicio de inicio de sesión, cerrar sesión y validar usuario en sesión.

**2.1 Nuevo usuario.**

Para este sub servicio se



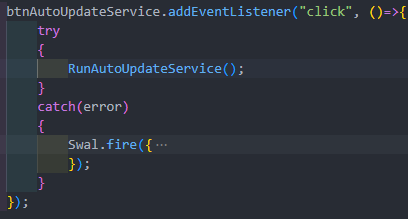
1.1.1 Nuevo Clientes: Este es el sub servicio que se encarga de la inserción a la base de datos

La única diferencia entre un administrador y un desarrollar radica en un servicio llamado “Auto Update Service” el cuál se encarga de agregar registros de nuevos clientes de forma masiva a la base de datos.

Auto Update Service o A.U.S no tiene interfaz gráfica ya que desde la página principal realiza la petición HTTP mediante Javascript al controlador de PHP.

La funcionalidad de este servicio es la siguiente:

**Botón ubicado en la página principal**

****

**Método que realiza la petición**

****

En la URL se debe enviar el parámetro “status” e igualarlo a “run”. Esto como medida adicional al usar el servicio.

La razón por la que solo un desarrollador puede usar este servicio es que es muy probable encontrar errores.

**Posibles errores:**

Los errores radican en la discrepancia de certificados.

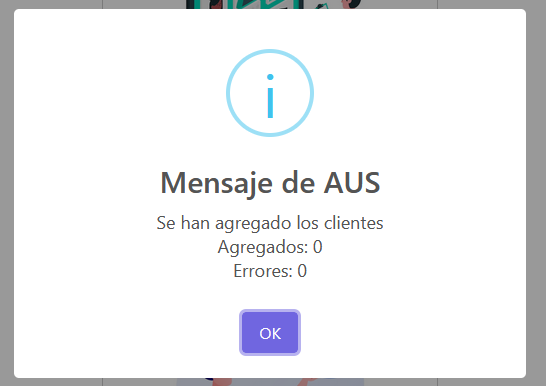
A.U.S realiza una búsqueda en la carpeta con todos los certificados para así buscar dos certificados iguales. La forma de diferenciar entre firma y sello es la carpeta en la que está el certificado.

Hay que verificar que todos los certificados de firma tengan un sello y viceversa.

También hay que verificar que se hayan agregado todos.

El sistema está preparado para esos escenarios, pero hay que seguir buscando errores.

Al usar A.U.S aparece un mensaje informativo con lo siguiente

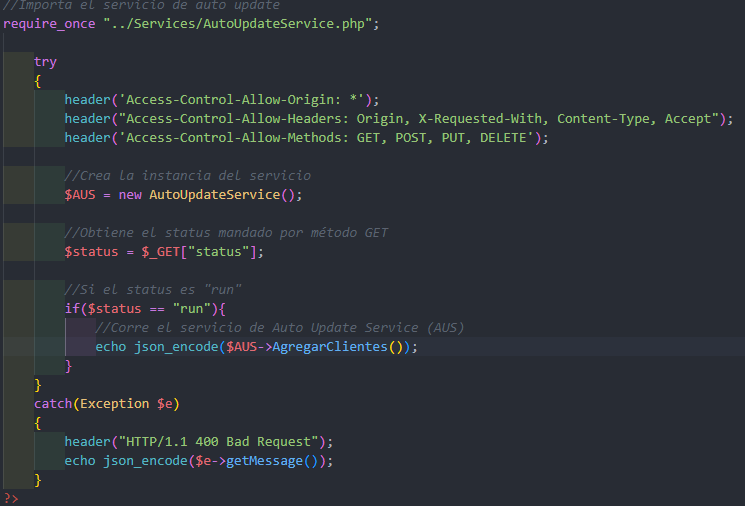


En donde “Agregados” es la cantidad de pares de certificados que se insertaron correctamente en la base de datos y “Errores” es la cantidad de certificados que no se agregaron.

**Algoritmo de A.U.S.**

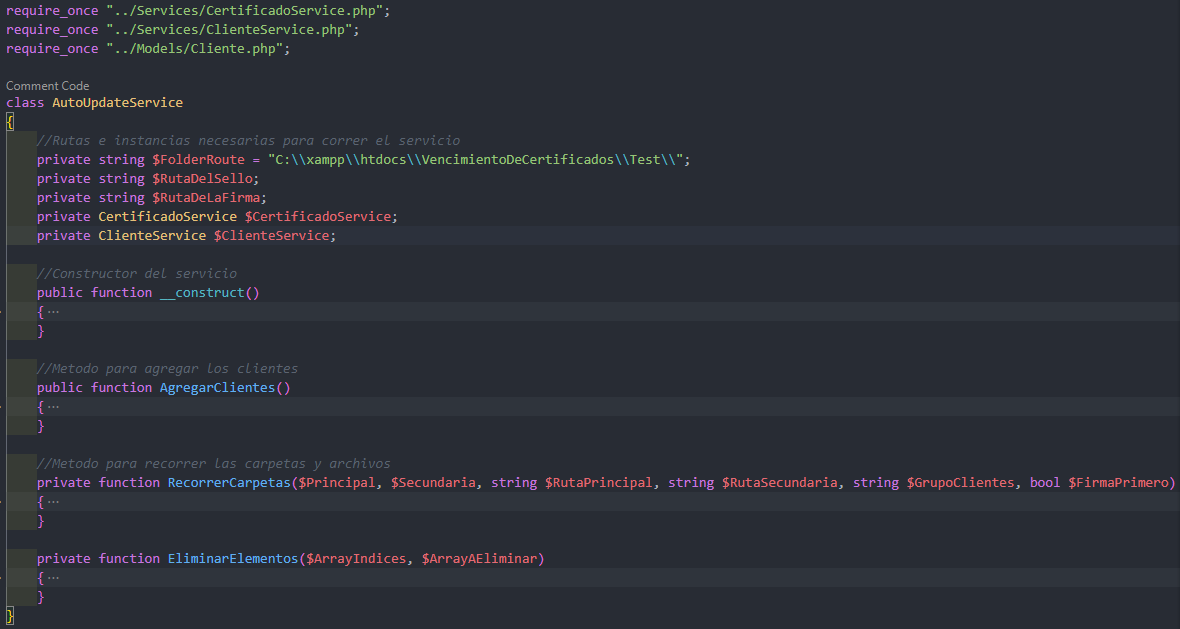
AUS funciona de la siguiente manera:

Recibe la petición HTTP en el controlador



Y este invoca al sub servicio “AgregarClientes”.

El servicio de AUS contiene varios sub servicios pero solamente uno es público. La estructura del AUS es la siguiente:



AUS usa el servicio de clientes y certificados para poder funcionar, adicionalmente usa el modelo de Cliente para así facilitar la inserción masiva a la base de datos.

En las propiedades tenemos:

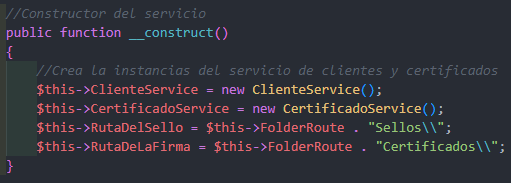
**$FolderRoute:** Que contiene la ruta en donde están guardadas las carpetas de sellos y firmas.

**$RutaDelSello y $RutaDeLaFirma:** Que se concatena a la ruta del folder en el constructor de AUS.

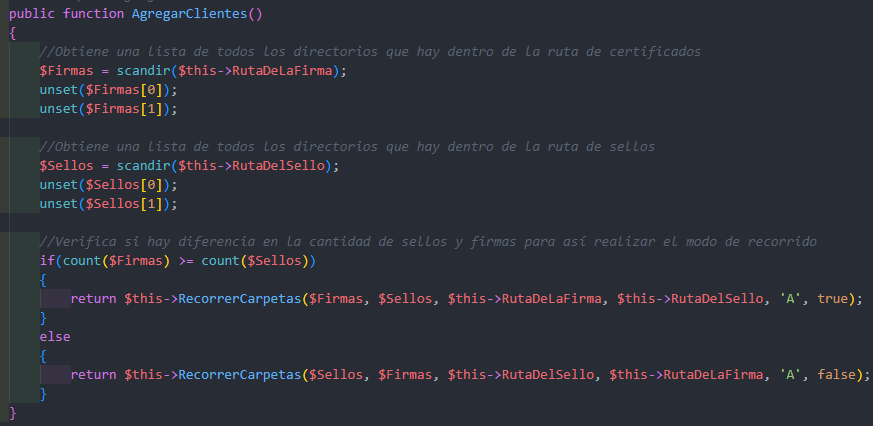
**$CertificadoService:** Instancia del servicio de certificados

**$ClienteService:** Instancia del servicio del cliente.

**Constructor:**



**Sub servicio para Agregar Clientes:** Este módulo del servicio de AUS emplea el sub servicio privado para recorrer carpetas de la siguiente manera.



En donde usamos el método “scandir” para obtener un listado de los directorios dentro de la ruta de las firmas y sellos, adicionalmente quitamos los primeros dos elementos del arreglo obtenido ya que los certificados comienzan a partir del índice 2.

Después tenemos una condición que no sirve para verificar discrepancias entre firmas y sellos y así elegir el modo de recorrido (firmas primero o sellos primero).

Dentro de cada posible escenario se manda a llamar al método “RecorrerCarpetas”, el cual recibe 6 argumentos:



**$Principal:** Es la carpeta de la cual se tienen más elementos.

**$Secundaria:** Es la carpeta de la cual se tienen menos elementos.

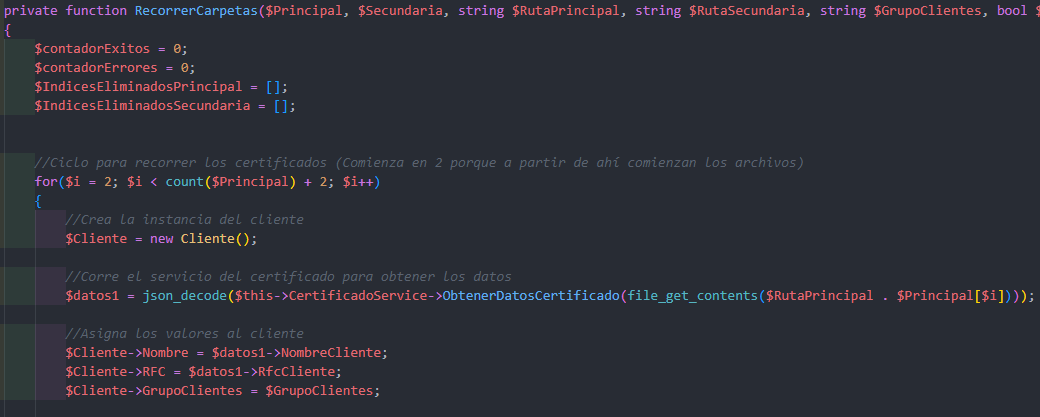
**$RutaPrincipal:** Es la ruta de la carpeta que se va a tomar como principal.

**$RutaSecundaria:** Es la ruta de la carpeta que se va a tomar como secundaria.

**$GrupoClientes:** Es el grupo que se va a agregar (Clientes A, B o C)

**$FirmaPrimero:** Esta variable nos sirve como bandera para saber si $Principal es firma (true) o sello (false).

A continuación, se explicará el algoritmo que emplea el sub servicio “RecorrerCarpetas” de AUS



Al inicio, la declaración de variables: contadorExitos, contadorErrores, IndicesEliminadosPrincipal e IndicesEliminadosSecundaria.

Los arreglos mencionados anteriormente nos servirán para saber los índices de los certificados que se han agregado a la base de datos.

Al final, estos se eliminarán de **$Principal** y **$Secundaria** dejando solo aquellos que por alguna razón no pudieron agregarse.

Después tenemos el primero ciclo que recorre el arreglo principal. Dentro de este se usa el servicio de certificados para obtener los datos y guardar solo el nombre, rfc y el grupo de clientes



Después tenemos el segundo ciclo que recorre el arreglo secundario, en este se verifica si el valor del índice está dentro de los índices eliminados. Si es así, significa que ese certificado ya ha sido agregado y, por lo tanto, no se necesita realizar todo el proceso y se pasa al siguiente índice para agilizar la búsqueda.

Después verifica si $j es menor al tamaño del arreglo secundario + 2 (Esto porque se eliminaron previamente 2 elementos del array).

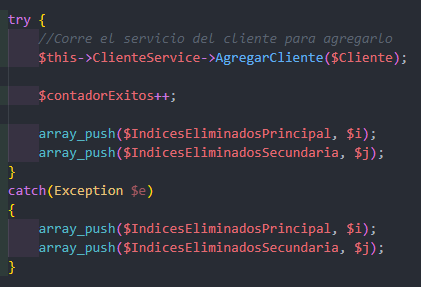
En caso de que sea verdadero, se obtienen los datos del certificado en la posición $j del arreglo.

Si el RFC previamente guardado es igual a la de los datos obtenidos del segundo certificado. Se procede a guardar la información en el objeto $Cliente.

Aquí nuestra bandera ($FirmaPrimero) juega un rol muy importante ya que, en caso de ser true, significa que $Principal es un arreglo de certificados de firmas y, por lo tanto, estos datos se guardarán en la propiedad “Firma” del objeto cliente.

De lo contrario, se guardará en la propiedad “Sello”.

Una vez hecho lo anterior. Se procede a guardar el objeto en la base de datos como un nuevo registro.

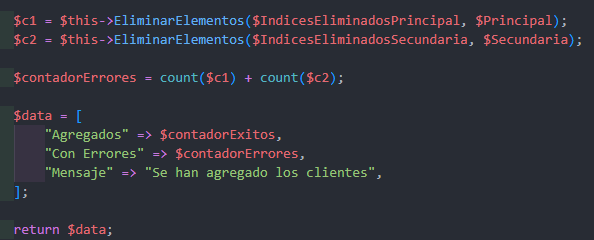


En caso de que el servicio de clientes no haya dado error, incrementa el contador de éxitos y guarda el índice como un nuevo valor en el array de los índices eliminados.

El método genera una excepción cuando el usuario ya ha sido agregado anteriormente (Con esto se refiere a que un empleado ya agregó ese cliente usando el servicio de clientes o por AUS).

Por lo tanto, también se guarda el índice en el array de eliminados.

Una vez realizadas todas las iteraciones de los ciclos, se proceden a eliminar los elementos en con los índices guardados.



Aquí es donde entra el segundo sub servicio de AUS, el cual recibe como argumentos el arreglo de los índices y el arreglo en donde se van a eliminar esos índices, el sub servicio tiene la siguiente estructura.

