

**ÍNDICE:**

[**1.Introducción: 2**](#_lmz7qro4givw)

[**2. Conexión BBDD 3**](#_gta3ospfbvte)

[**3.Creación de tablas 4**](#_skvz4yfmeefh)

[3.1.Pestañas de Departamentos y Trabajadores 6](#_7voebn551b3p)

[3.2.Buscador en las tablas 7](#_o5xv59ha55tu)

[3.3.Paginación de las Tablas 7](#_g5nrosx4ipjg)

[3.4. Función insertar 9](#_vxf5uyh3tj2s)

[3.5. Función borrar 11](#_cb95u3ywpwwh)

[3.6.Función Editar 13](#_yn8d58to0zrr)

[**4.Cookies 15**](#_wejjsrwbftvd)

[**5.Cerrar sesión 16**](#_spiiac9tliog)

# 

# 1.Introducción:

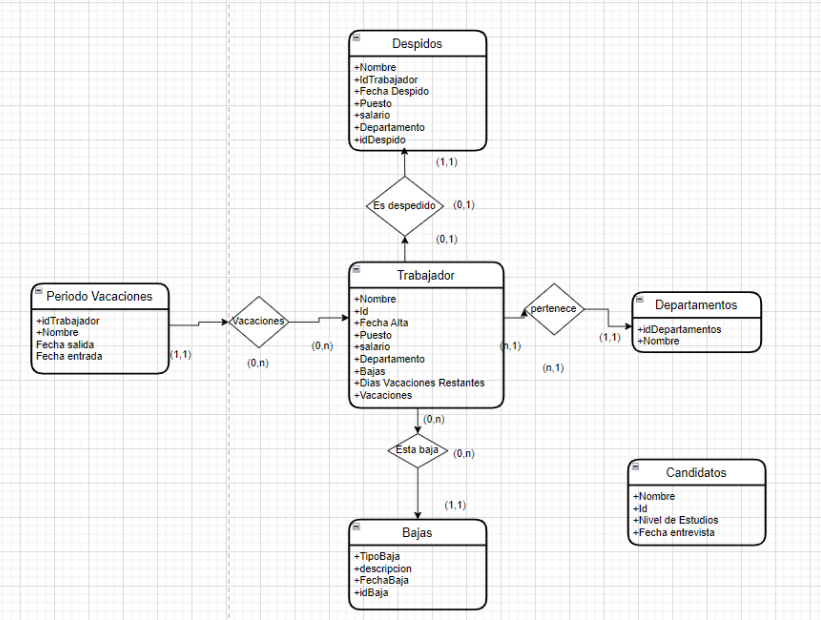
Como se ha especificado en clase, se deja anotado el usuario y la contraseña creados para el profesor. Así se podrá probar la funcionalidad de la página:

Usuario: adminRRHH

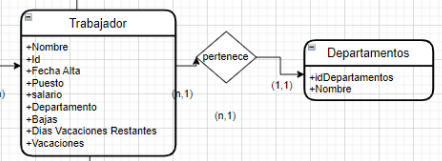
Contraseña: adminrrhh

En el siguiente documento se va a explicar el funcionamiento y las partes importantes del código del módulo de RRHH del proyecto conjunto de clase para crear un ERP .

En nuestro caso empezamos definiendo un modelo Entidad-Relación para crear las tablas en la base de datos, que a medida que se avanzaba en el trabajo se ha visto demasiado ambicioso. La idea original era la siguiente:

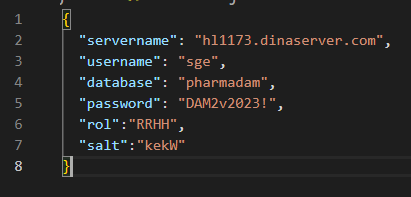


Finalmente se ha optado por hacer la siguiente parte del conjunto:

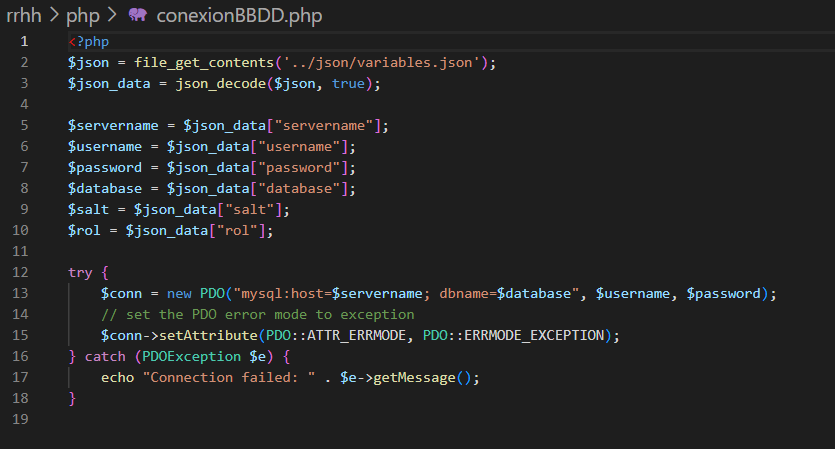


# 2. Conexión BBDD

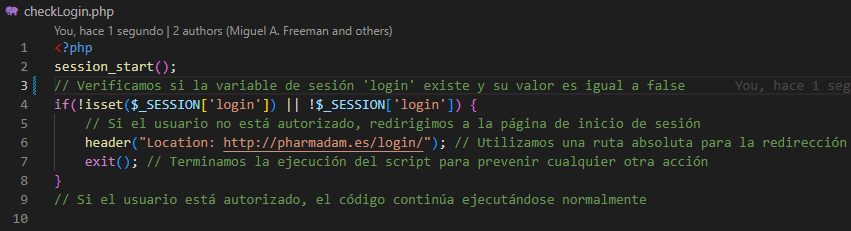
Lo primero que se ha creado es la conexión a la base de datos. Se ha hecho un json que tendrá todos los datos necesarios para conectarse a la BBDD.



Y una clase de php para poder utilizar las diferentes variables y conectarse:



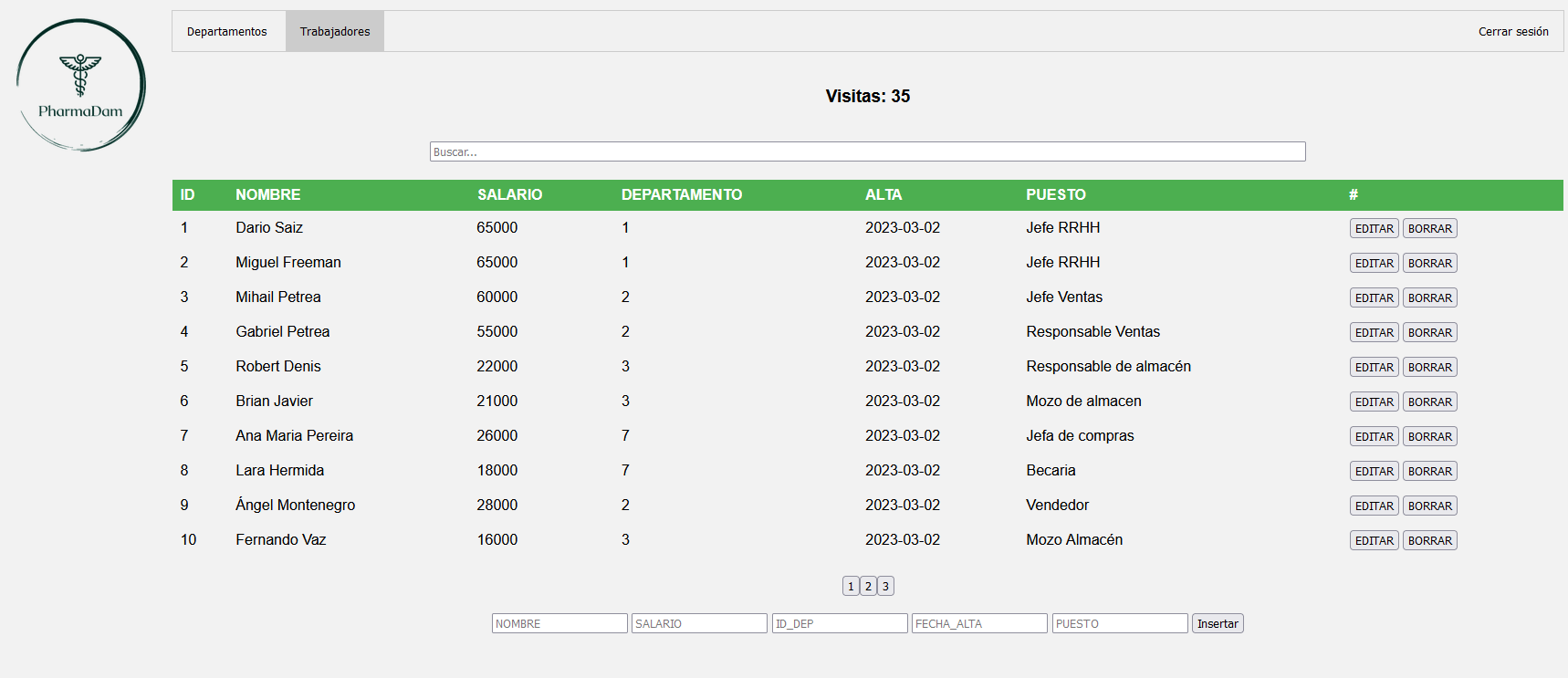
También se ha creado una clase de php para chequear si un usuario está autorizado para entrar. Este fichero se utilizará en alguna de las páginas, como en la principal llamada index.php que se verá más adelante.



# 3.Creación de tablas

Una vez hecho el las clases que nos permiten la conexión a la BBDD y que nos autentifican, se procedió a crear la página principal de RRHH que contendrá las tablas de Departamento y Trabajador.

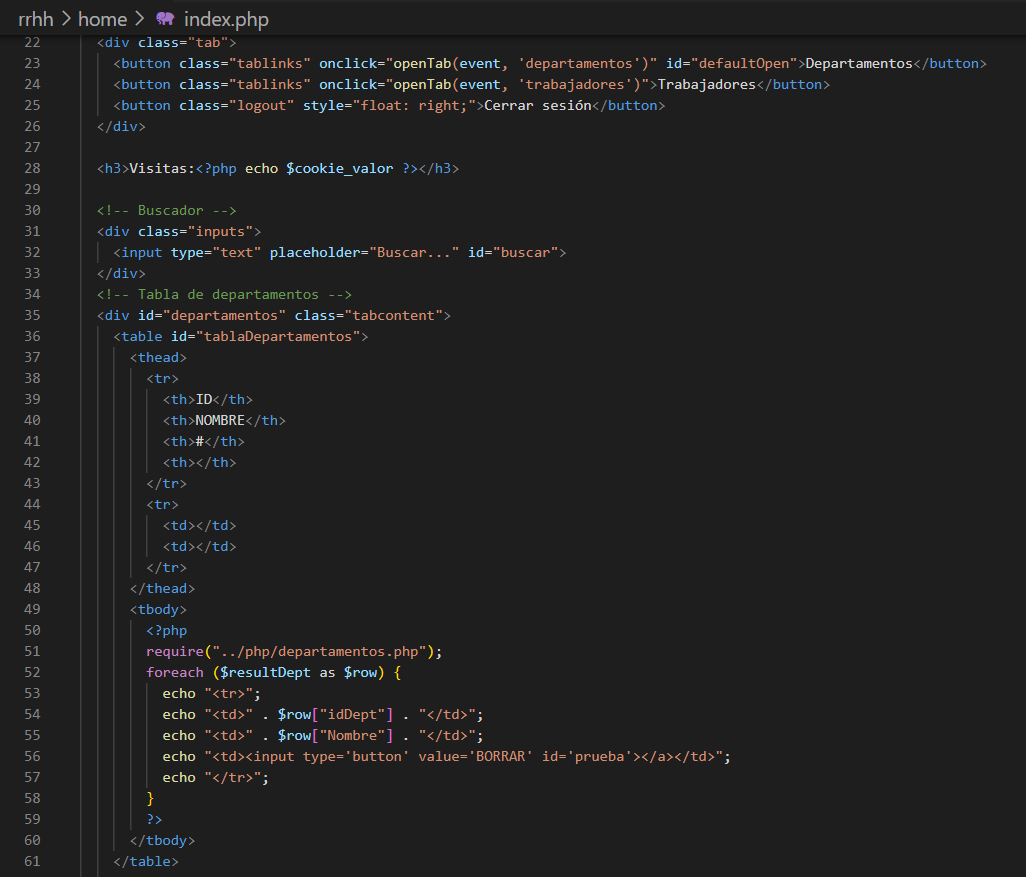
La vista principal de esta página que se ha llamado index.php sería la siguiente:

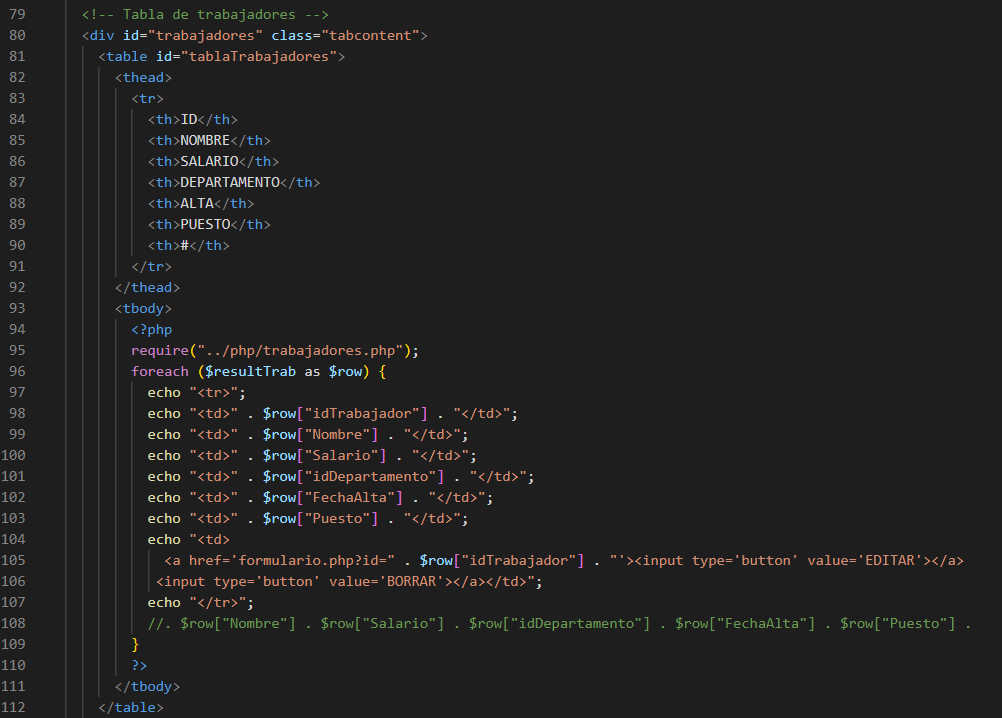


Como se puede observar hay un mecanismo de pestañas para poder elegir entre las tablas de Departamentos y Trabajadores. Tiene un buscador para poder localizar antes al trabajador o departamento y se ha introducido a la derecha los botones de editar y borrar que en el caso de departamentos se ha optado por solo introducir el de borrar. También se podrán insertar nuevos trabajadores y departamentos con los campos que se localizan abajo de la pantalla además de que se tiene paginación para cada tabla. Por último se ha creado un botón de cerrar sesión que te devolverá al login de PharmaDam.

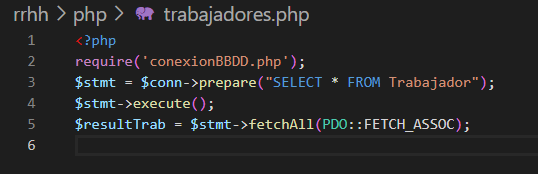
Para poder mostrar las dos tablas se ha creado un css que nos controla las pestañas, la página principal index.php que contiene la estructura de las tablas y los métodos para rellenarlas.

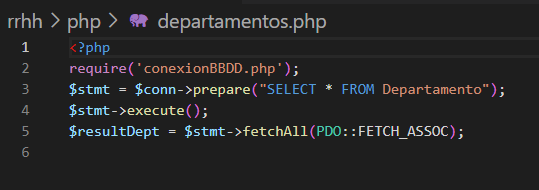
El código siguiente sería de la tabla de departamentos y la tabla trabajadores que recogen los datos de las clases departamentos.php y trabajadores.php. Se recorre cada columna a través de un foreach y se va escribiendo en cada columna un dato.





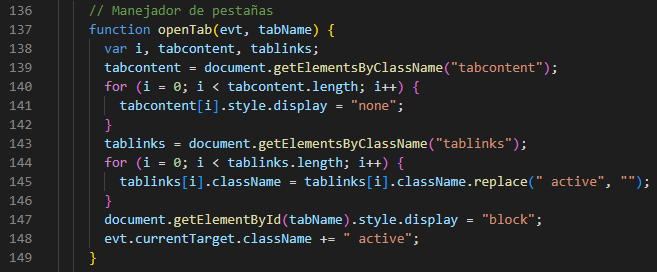
El código de los php sería el siguiente y utiliza un “select \* “ para recoger todos los datos de cada tabla.





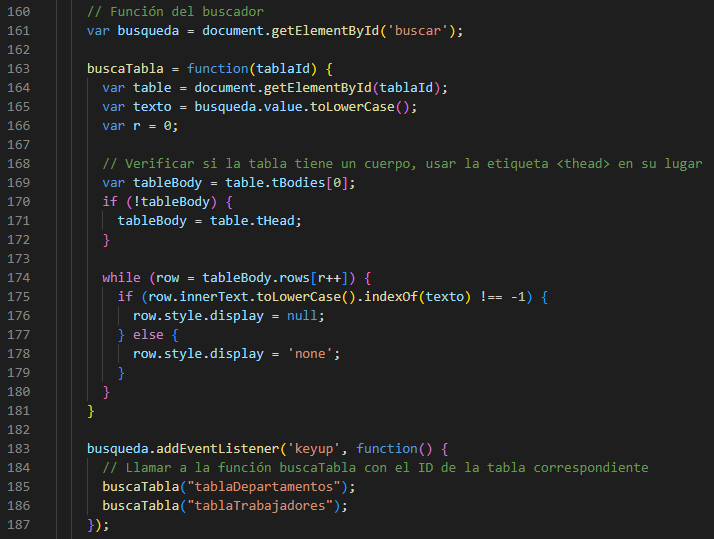
## 3.1.Pestañas de Departamentos y Trabajadores

La función openTab se encarga de manejar las pestañas de la página. Toma como argumentos el evento que ha activado la función (evt) y el nombre de la pestaña a la que se desea acceder (tabName).

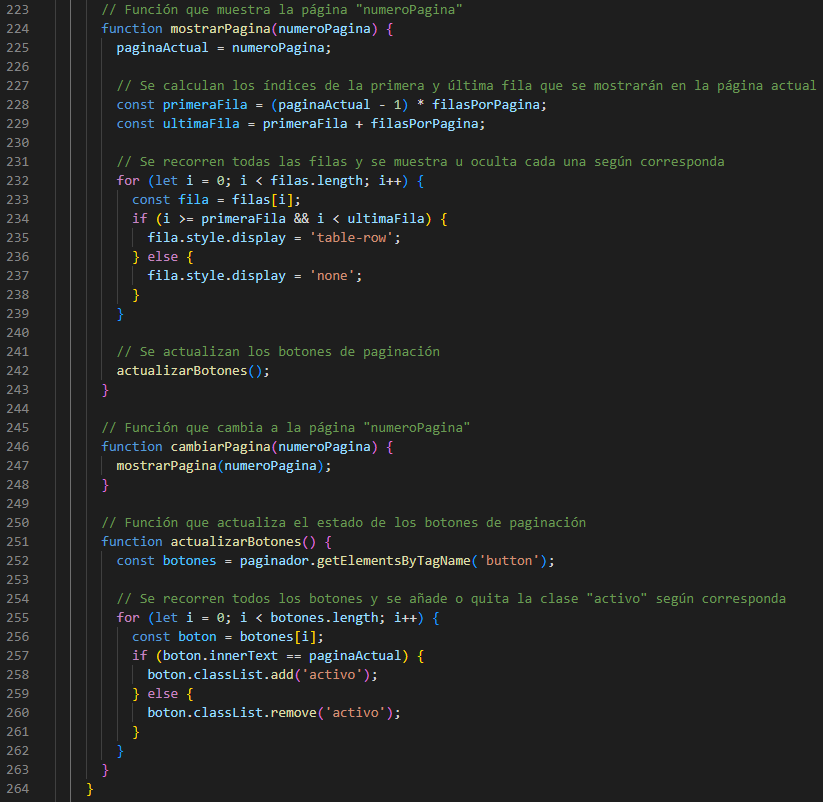


## 3.2.Buscador en las tablas

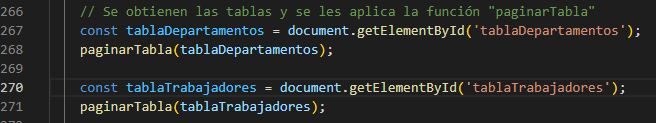
La función buscarTabla se encarga de filtrar una tabla a partir del texto ingresado en un campo de búsqueda. La función obtiene el ID de la tabla a filtrar como parámetro y luego utiliza el método getElementById para obtener una referencia a dicha tabla en el DOM.



## 3.3.Paginación de las Tablas

La función "paginarTabla" permite la paginación de tablas en HTML. Toma como parámetro la tabla a paginar y muestra un número determinado de filas por página. Crea un elemento "div" que contiene los botones de paginación, los cuales permiten cambiar entre las diferentes páginas de la tabla. También contiene funciones para mostrar la página actual, cambiar a otra página y actualizar el estado de los botones de paginación. En resumen, esta función crea una interfaz de paginación para tablas HTML.

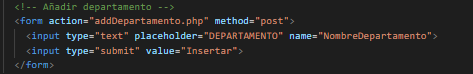
Una vez hecha la función solo queda aplicarla a las tablas deseadas:

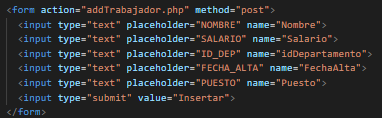


## 3.4. Función insertar

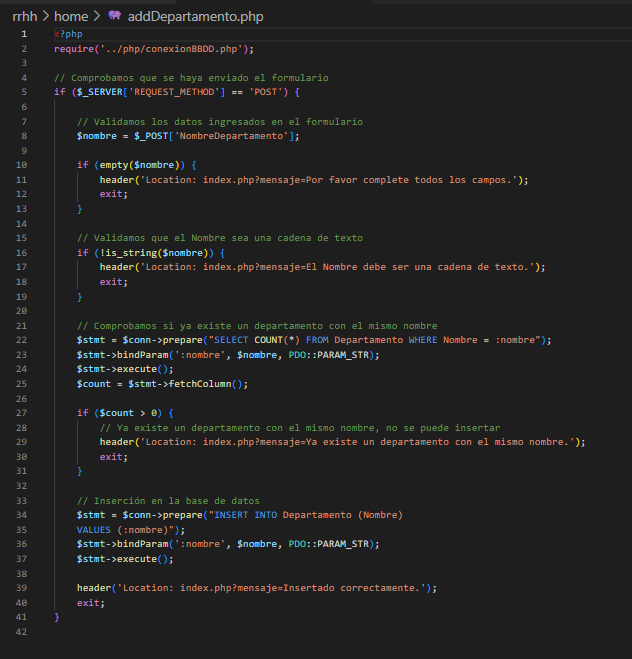
Debajo de las tablas encontraremos los espacios para introducir las diferentes variables para insertar un nuevo departamento o trabajador dependiendo en la pestaña que nos encontremos.

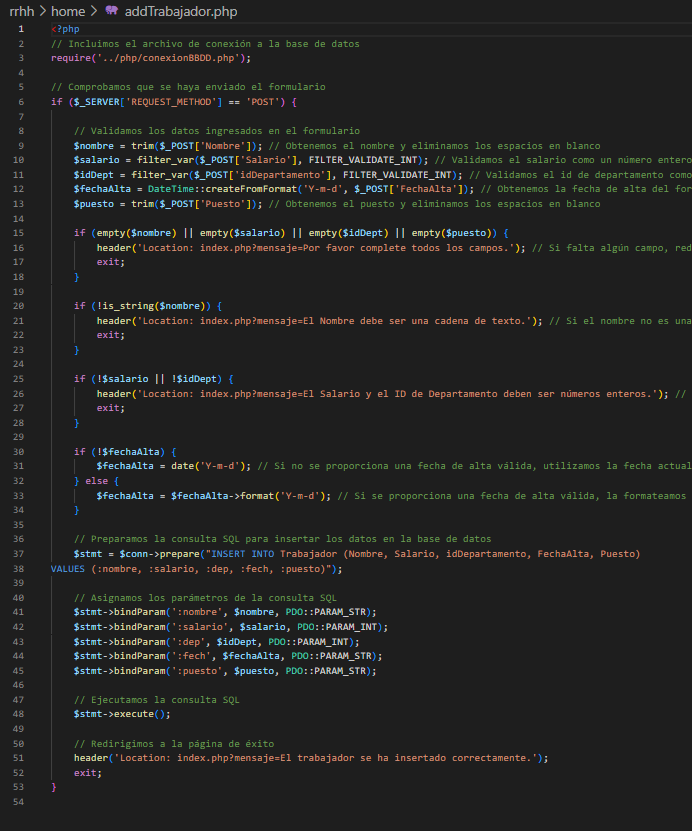
El código del index.php para enviar los datos de los campos rellenos desde el formulario de la página principal es el siguiente:





Se han creado dos php para controlar la llamada a la base de datos para añadir tanto departamentos como trabajadores. Cada variable se validará para que se introduzca en el formato correcto, que no haya repetidos y que no se puedan dejar campos en blanco en el formulario que llega desde la página principal.





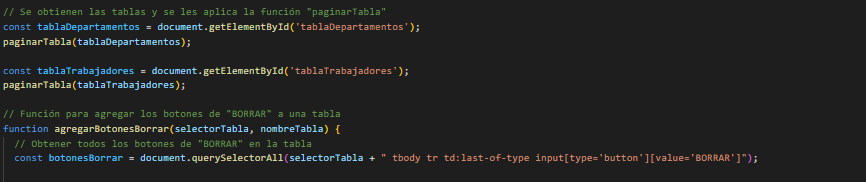
Finalmente se hace un CRUD con un “INSERT INTO” donde corresponda.

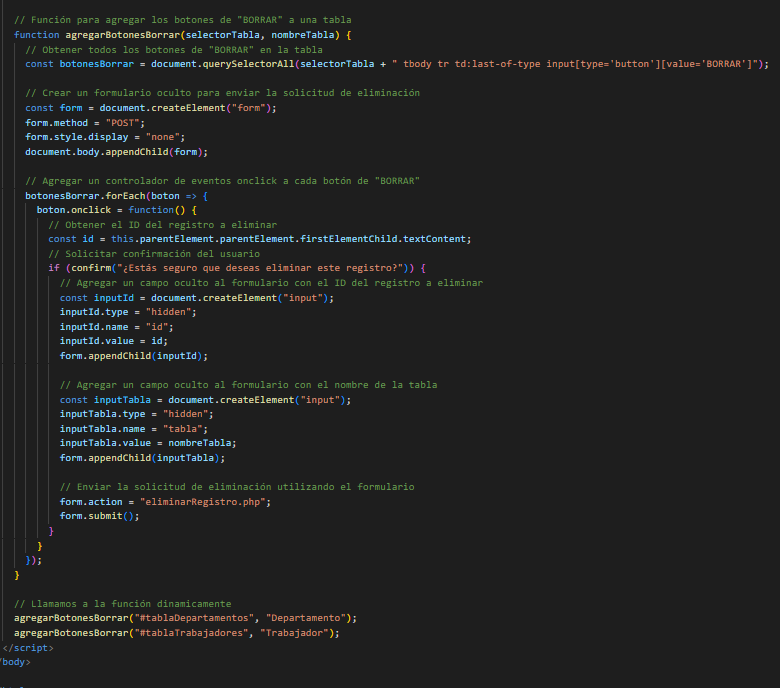
## 3.5. Función borrar

En cada fila de la tabla, en la parte derecha estará un botón de eliminar. Con este botón eliminaremos los datos de la fila que seleccionemos. Si pulsamos el botón llamaremos a la función “OnClick” de ese botón, en este caso el de borrar.

Al presionar el botón se selecciona la tabla donde se esta y el id que corresponda a la fila. Se hace un método POST que nos redirige a nuestro php que contiene el código CRUD para eliminar los datos seleccionados a través de la id. Además al pulsar aparecerá una alerta de confirmación de borrado.

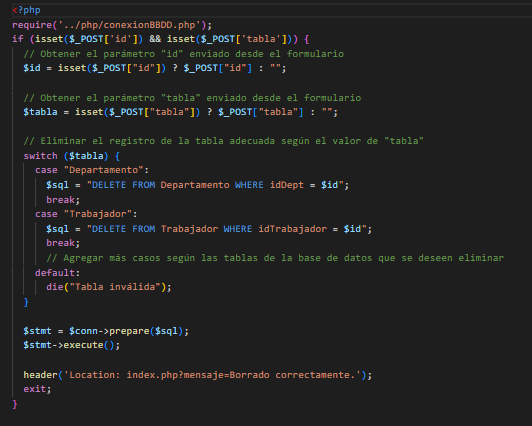
A continuación, esta el código que se introduce en index.php:





Cuando se tiene el nombre de la tabla y el id se envia el formulario oculto con los datos a la clase eliminarRegistro.php que incluye tanto la eliminación de los datos de la tabla departamentos como de la tabla trabajadores. La función OnClick comentada anteriormente, coge el nombre de la tabla y el id y lo manda, por lo tanto, el código puede diferenciar de que tabla y registro se tratan los datos.

Se crea en este php un SWITCH para diferenciar la tabla y una sentencia CRUD para eliminar el registro según el id. Además, incluimos un header que nos devolverá a la página principal cuando se termine de ejecutar todo el código.



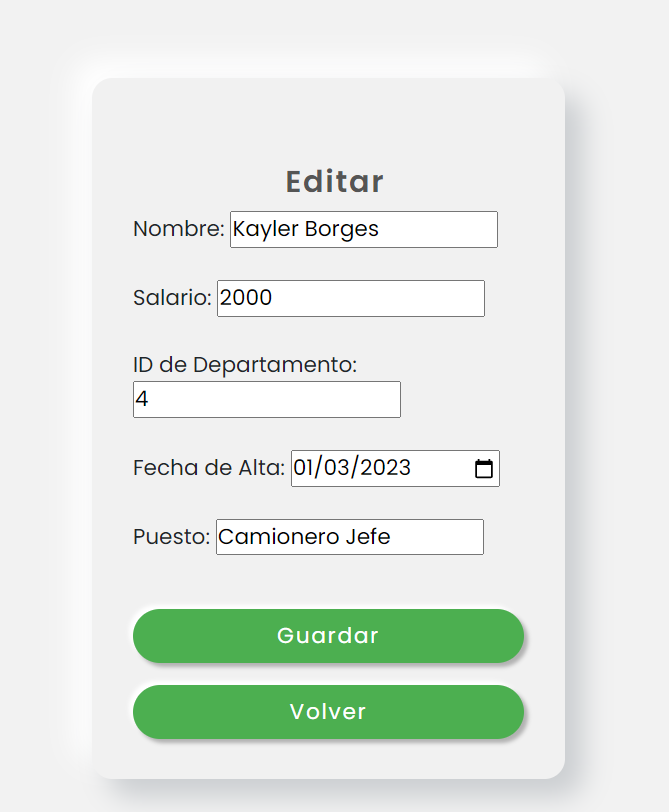
## 3.6.Función Editar

En este caso y durante la creación de esta función, se decidió que solo estuviera activa en la tabla de trabajadores, ya que, se veía innecesario y de poca seguridad que esto ocurriera y se pudieran eliminar departamentos.

En el index.php lo primero que se tiene, es coger con el botón editar el “id” al que nos estamos refiriendo, además de que el botón nos redirigirá a una página llamada formulario.php.



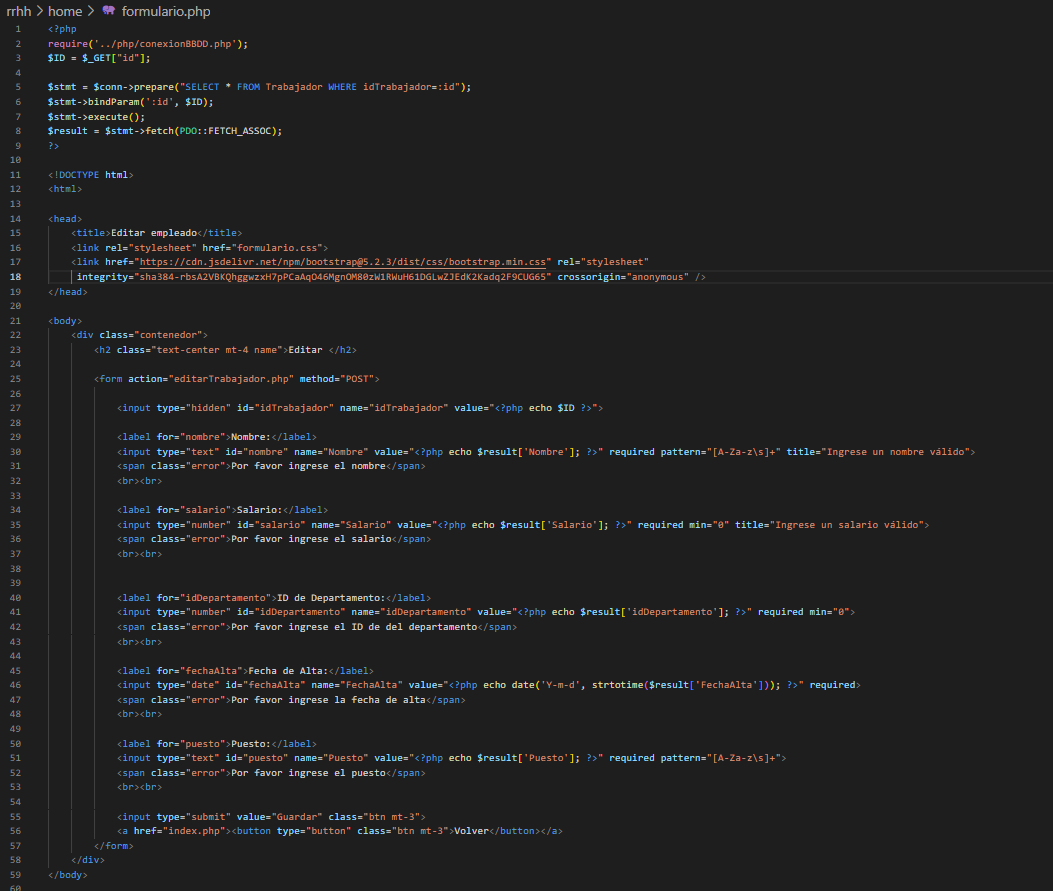
Esta página es un formulario. Los campos se rellenarán automáticamente con los datos del trabajador de la fila que se haya seleccionado. El único campo que no aparecerá y que no se podrá editar será el de ID ya que un trabajador siempre utilizará el mismo aunque le cambien los datos. En este caso se ha elegido a Kayler para hacer el ejemplo.



Para conseguir que los campos se vean rellenos en el formulario, se hace una llamada GET a la base de datos para recibir a través del ID todos los datos.

Todos los campos del formulario están validados tanto con formato como no dejar huecos en blanco.

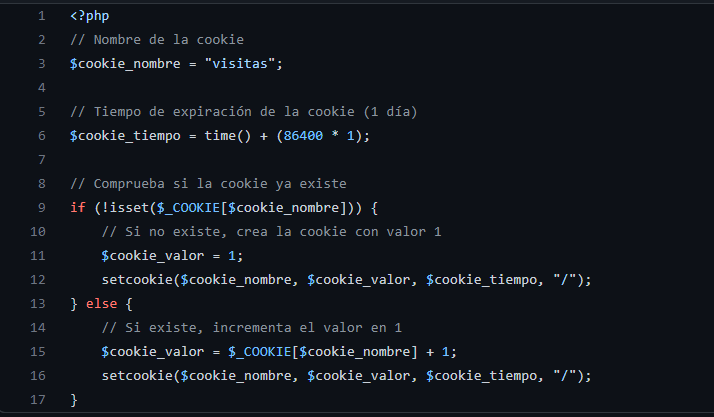
Por último tenemos dos botones, el de guardar que creará un submit para actualizar la base de datos como el de volver que nos redirigirá a la página principal.



# 4.Cookies

Se ha creado una cookie que se crea cuando el usuario entra por primera vez y cuenta cuantas veces se conecta a nuestra página. Esta cookie esta insertada en nuestro index.php que hace una llamada al valor de esta cookie.

El código de la cookie está en la clase cookieVisitas.php:



# 5.Cerrar sesión

Para cerrar sesión el botón cerrar sesión de el index del home tiene anclado una url que redirige a cerrarSesion.php que básicamente acaba con la sesión actual de el usuario y redirige a el login.

