

Profesores:

*Neiner Maximiliano, Villegas Octavio*

## 1.- Verificar usuario

El formulario de la pagina login.html, establecer el valor del “action” en ‘backend/verificarUsuario.php’.

En la página **verificarUsuario.php** se recibirán por POST el DNI y el apellido de un usuario. Con estos datos se deberá determinar si existe un empleado dentro del archivo de texto (empleados.txt) que coincida con los valores requeridos.

Si no existe un empleado en el archivo de texto, mostrar un mensaje de error y habilitar un link para volver a **login.html**.

Si se encuentra al empleado, se deberá redireccionar hacia **mostrar.php**.

***Nota:** Para realizar el direccionamiento utilizar la función header. Ver funcionamiento en <http://php.net/manual/es/function.header.php>*

## 2.- Validar la sesión del usuario

En la página **verificarUsuario.php**, cuando el usuario sea verificado, establecer la variable de sesión **DNIEmpleado** con el valor del DNI del usuario que se ha ‘logueado’ al sitio.

Dicha variable de sesión permitirá saber si un visitante está identificado o no, y con esta información determinar si puede acceder a una página de nuestro sitio o no.

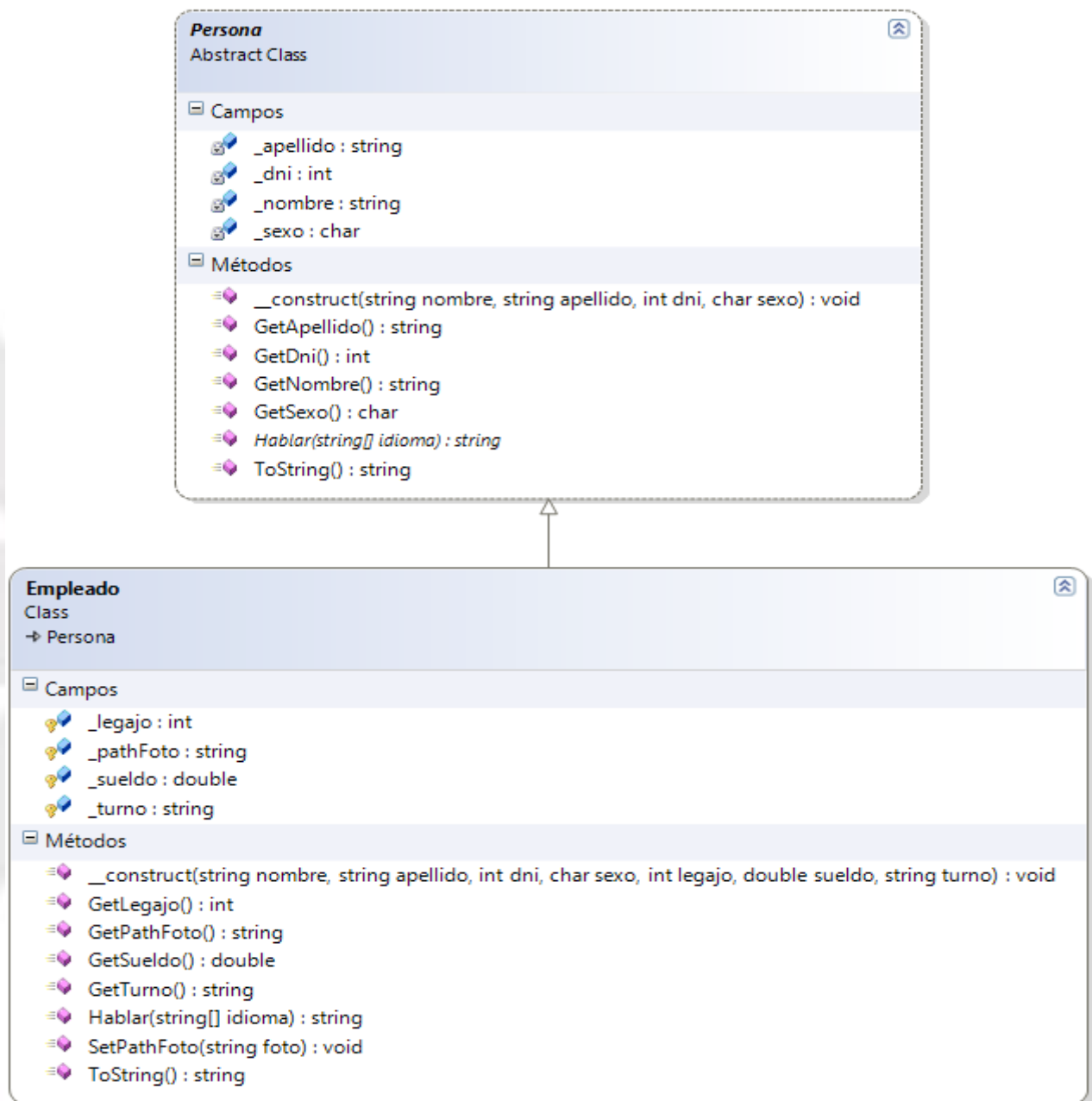
Agregar la página **validarSesion.php** que verifique si la variable de sesión **DNIEmpleado** existe o no. En el caso de no existir, redireccionar hacia **login.html**.

Incluir la página **validarSesion.php** en las páginas **index.php** y **mostrar.php**.

***Nota:** Agregar en las páginas **mostrar.php** y **index.php** un link para poder ‘desloguearse’. Crear la página **cerrarSesion.php** para lograr dicho objetivo. Redirigir hacia **login.html**.*

## 3.- Reformular la jerarquía de clases

Agregar en la clase **empleado**, un atributo (con sus métodos getters y setters) para poder almacenar el path de la foto del empleado. **Ver diagrama de clases**.



Modificar también el método `ToString()` para que contemple dicho atributo.

### 3.1 - Modificar formulario y validaciones (Front-end)

Modificar ***index.html***, para poder seleccionar una foto por cada empleado que se dé de alta.

☐ Noche

Foto:  Ningún archivo seleccionado

---

Agregar en la función **AdministrarValidaciones** la validación de campo vacío para la foto del empleado.

**Nota:** Recordar establecer el valor del atributo “enctype” con el valor ‘multipart/form-data’ para el formulario.

#### 4.- Validar y guardar la imagen (Back-end)

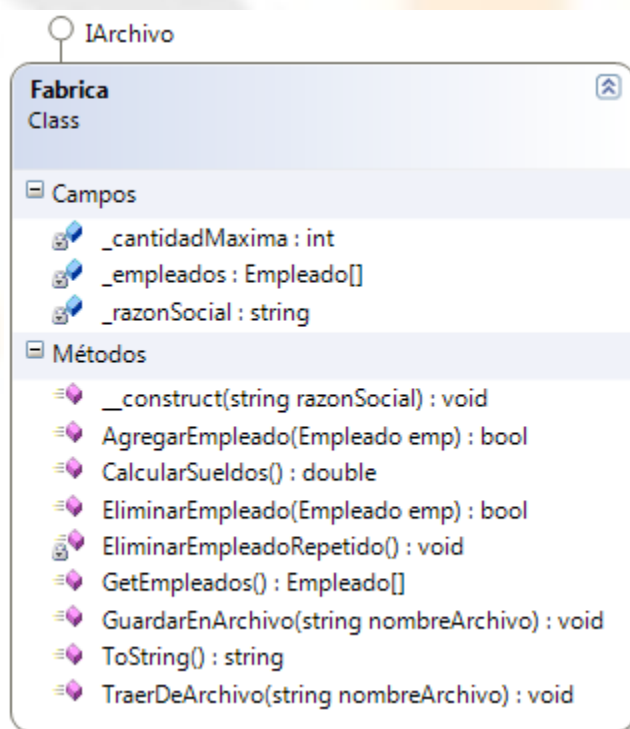
En **administracion.php**, se deberá validar:

- Que el archivo sea una imagen.
- Que sus extensiones válidas sean JPG, BMP, GIF, PNG o JPEG.
- Que su tamaño máximo sea de 1MB.
- Que no exista un archivo con el mismo nombre en el repositorio final (**./fotos**)

Si todas las validaciones son pasadas, se guardará la instancia de la clase empleado y su foto se almacenará en el subdirectorio **./fotos**, renombrando la imagen recibida según el siguiente formato:  
**dni-apellido.extension**

#### 5.- Modificar la clase fabrica

Agregar en la clase **fabrica**, un método getter para poder obtener el array de empleados.  
**Ver diagrama de clases.**



Modificar el funcionamiento del método **TraerDeArchivo** para que, por cada empleado que se recupere del archivo de texto, almacene el path de la foto de dicho empleado y lo almacene en el atributo correspondiente.

## 6.- Mostrar fotos de los empleados

Modificar la página **mostrar.php** para que se recuperen los empleados del archivo de texto, mediante una instancia de la clase **fabrica**, y se agregue una etiqueta `<img>` que muestre la foto de cada empleado.

**Nota:** recordar establecer el valor del atributo “src” con el valor obtenido del método de instancia **GetPathFoto** de la clase empleado. Establecer el valor de los atributos “width” y “height” en ‘90px’.

### Listado de Empleados

#### Info

1000001 - Juan - Perez - M - 101 - 8500 - T - fotos/1000001-Perez.jpg



[Eliminar](#)

1000033 - Susana - Gimenez - F - 112 - 9500 - N - fotos/1000033-Gimenez.gif



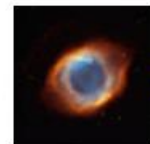
[Eliminar](#)

1000008 - Gustavo - Cordera - M - 112 - 25000 - N - fotos/1000008-Cordera.jpg



[Eliminar](#)

1000038 - Roberto - Sanchez - M - 106 - 20000 - M - fotos/1000038-Sanchez.jpg



[Eliminar](#)

1000002 - Joaco - Hernandez - M - 153 - 9000 - M - fotos/1000002-Hernandez.jpg



[Eliminar](#)

Agregar la eliminación de la imagen del repositorio físico cuando se invoque al método de instancia **EliminarEmpleado** de la clase **fabrica** (link Eliminar).

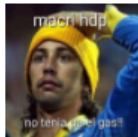



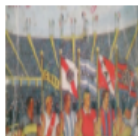
## 7.- Modificar datos de los empleados

Agregar en la página **mostrar.php** una nueva columna con un input (type=button) con la leyenda **Modificar**.

Asociar en evento onclick de los botones con la función *AdministrarModificar*, la cual recibe como parámetro el DNI del empleado.

Agregar un formulario que contenga un input (type=hidden) para ser enviado hacia la página **index.php**. Este hidden servirá para cambiar el funcionamiento del formulario de alta de empleados.

**Nota:** renombrar el archivo **index.html** a **index.php** para poder mostrar los datos del empleado a ser modificado.

Listado de Empleados		
Info		
1000001 - Juan - Perez - M - 101 - 8500 - T - fotos/1000001-Perez.jpg		<a href="#">Eliminar</a> <input type="button" value="Modificar"/>
1000033 - Susana - Gimenez - F - 112 - 9500 - N - fotos/1000033-Gimenez.gif		<a href="#">Eliminar</a> <input type="button" value="Modificar"/>
1000008 - Gustavo - Cordera - M - 112 - 25000 - N - fotos/1000008-Cordera.jpg		<a href="#">Eliminar</a> <input type="button" value="Modificar"/>
1000038 - Roberto - Sanchez - M - 106 - 20000 - M - fotos/1000038-Sanchez.jpg		<a href="#">Eliminar</a> <input type="button" value="Modificar"/>
1000002 - Joaco - Hernandez - M - 153 - 9000 - M - fotos/1000002-Hernandez.jpg		<a href="#">Eliminar</a> <input type="button" value="Modificar"/>

La función *AdministrarModificar* deberá establecer el valor del input (type=hidden) con el valor recibido como parametro y luego 'submitear' el formulario que lo contiene.

En **index.php** se deberá recuperar del archivo de texto al empleado que coincida con el DNI recibido por POST.

Los datos recuperados serán mostrados en cada uno de los elementos del formulario de ingreso.

Para esto, se deberá:

- Crear una instancia de tipo Fabrica (obteniendo los empleados por medio del metodo *TraerDeArchivo*)
- Realizar una busqueda y obtener al empleado cuyo DNI coincida con el recibido.
- Establecer los valores del objeto a cada elemento del formulario.
- Cambiar el titulo de nivel 2 de 'Alta de Empleado' a 'Modificar Empleado'.
- Cambiar el titulo del navegador <title> de 'HTML 5 Formulario Alta de Empleado' a 'HTML5 Formulario Modificar Empleado'.
- Colocar a los inputs **txtDni** y **txtLegajo** como de solo lectura (readonly).
- Cambiar el texto del input **btnEnviar** de 'Enviar' a 'Modificar'.
- Agregar un input (type=hidden), **hdnModificar**, para agregar las siguientes acciones en **administracion.php**:
  - Obtener la instancia del empleado a ser modificado, realizando una busqueda por DNI en la lista de empleados de la fabrica.
  - Invocar al método de instancia *EliminarEmpleado* de la fabrica, pasándole como parámetro el empleado obtenido.

***Nota:** Para poder modificar un empleado (con su foto), es necesario eliminar al empleado (con su foto) y luego agregarlo como un empleado nuevo.*