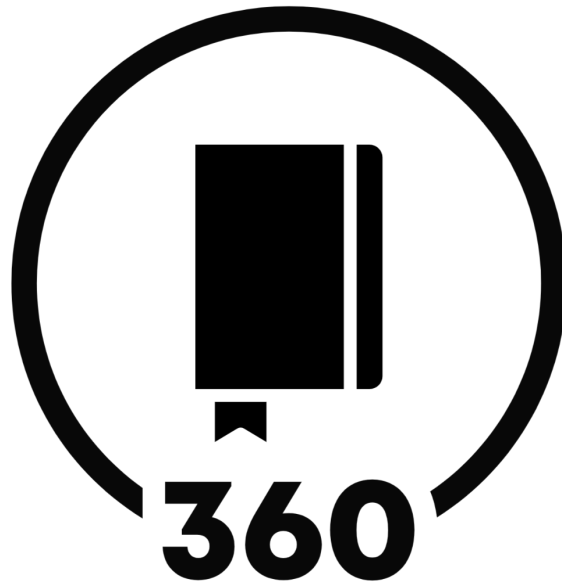


# Aplicaciones Web

*Práctica 3 - Grupo 1 - Campus Virtual*



Lucía Blasco Gallego  
Martín Ruano Crespo  
Diego Pellicer Lafuente  
Andrés Sánchez Casas

# ÍNDICE

## **1. Listado de script para las vistas**

- 1.1 Funcionalidad Cursos*
- 1.2 Funcionalidad Asignaturas*
- 1.3 Funcionalidad Mensajes*
- 1.4 Funcionalidad Calendario*
- 1.5 Login / Logout*

## **2. Listado de scripts adicionales**

- 2.1 Scripts con definiciones de clases*
- 2.2 Scripts con lógica de la aplicación*
- 2.3 Script de abstracción del manejo de la base de datos*
- 2.4 Script css*

## **3. Estructura de la base de datos**

- 3.1 Modelo Relacional*
- 3.2 Tablas*

## **4. Datos de acceso**

## **5. Parte de actividades**

## 1. Listado de scripts para las vistas

- **plantilla.php:** Es utilizada por todas las páginas para generar su vista. E incluye:
  - **Cabecera.php:** sección de encabezado, con título principal "Campus 360" y una sección que muestra un saludo personalizado con el nombre y apellido del usuario, con un botón para cerrar sesión si este ha iniciado sesión, o en caso contrario, se muestra un mensaje de bienvenida para los usuarios desconocidos, con enlaces para iniciar sesión o registrarse en la aplicación.
  - **Pie.php:** en la parte inferior de la página web, el contenido que se muestra es simplemente el texto "Pie de página".
  - **sidebarIzq:** define una barra de navegación que contiene cuatro opciones: Inicio, Calificaciones, Foro y Perfil. Cada opción es un enlace que lleva a una página específica de la aplicación, utilizando la función `resuelve()` de la clase "Aplicación".
  - **sidebarDer:** muestra un calendario en la sección derecha de la página. Se utiliza la clase "Calendar" que se ha definido en el archivo "Calendar.php" y se instancia un objeto de esta clase con la fecha actual.
- **plantillaLogin.php:** Plantilla para la vista del login que contiene el logo de la aplicación.

### 1.1 Funcionalidad Cursos

- **admin.php:** verifica si el usuario actual tiene el rol de administrador y muestra un contenido diferente según el resultado de la verificación. Si el usuario actual es administrador, se muestra un encabezado de consola de administración, de lo contrario, se muestra un mensaje de acceso denegado.
- **cicloAdmin.php:** muestra una lista de ciclos disponibles y permite eliminar ciclos existentes. También proporciona un enlace para ver los cursos asociados a cada ciclo y un enlace para crear un nuevo ciclo.
- **cursoAdmin.php:** muestra una lista de cursos disponibles para un ciclo seleccionado y permite eliminar cursos. Además, muestra un enlace para crear un nuevo curso.
- **asignaturasAdmin.php:** muestra una lista de asignaturas para un curso específico y ofrece la opción de eliminar cada asignatura. Si no se encuentran asignaturas, se muestra un mensaje indicando que no hay asignaturas disponibles para ese curso. Finalmente, se agrega un enlace para crear una nueva asignatura.
- **crear-ciclo.php:** permite crear un nuevo ciclo, comprueba si se ha enviado el formulario. Si es así, recoge el nombre del ciclo enviado a través del formulario y llama a la función "crearCiclo" para agregarlo a la base de datos y redirige al usuario a la página de ciclos. Si no se ha enviado el formulario, se muestra el formulario para que el usuario ingrese el nombre del nuevo ciclo. Finalmente, se genera la vista de la página utilizando la plantilla común `plantilla.php`

- **crear-curso.php:** Este script permite crear un nuevo ciclo. Primero define el título y contenido principal de la página, luego comprueba si el método de solicitud es POST (es decir, si se ha enviado el formulario). Si es así, recoge el nombre del ciclo enviado a través del formulario y llama a la función "crearCiclo" para agregarlo a la base de datos. Finalmente, redirige al usuario a la página de ciclos. Si no se ha enviado el formulario, se muestra el formulario para que el usuario ingrese el nombre del nuevo ciclo. Finalmente, se genera la vista de la página utilizando una plantilla común.
- **crear-asignatura.php:** es una página para crear una nueva asignatura. Primero, se define el título de la página y el contenido principal. Luego, se verifica si se ha enviado un formulario de creación de asignaturas utilizando el método POST. Si se ha enviado, se obtienen los valores del nombre de la asignatura, el profesor asignado y la lista de alumnos seleccionados. Estos se envían a la función "crearAsignatura" para que se almacenen en la base de datos y se redirige al usuario a la página de lista de asignaturas. En caso contrario, se obtienen las listas de profesores y alumnos utilizando las funciones "obtenerListaProfesores" y "obtenerListaAlumnos" respectivamente. Luego, se generan opciones para el menú desplegable de profesores y para la lista de alumnos mediante un bucle foreach. El código HTML se genera con la información de las listas y se muestra en el formulario. Finalmente, se pasa el título y el contenido principal a la plantilla de la página utilizando el método generaVista() del objeto \$app.

## 1.2 Funcionalidad Asignaturas

- **AsignaturasAlu.php:** genera una lista con un enlace a la página "contenidoAsignatura.php" para cada asignatura.
- **AsignaturasProfesor.php:** lista de asignaturas disponibles para un determinado profesor identificado por su ID.
- **AsignaturasPadre.php:** se obtienen las asignaturas disponibles para ese padre y se muestra una lista de ellas.

### 1.2.1 Sub Funcionalidad Entrega

- **contenidoEntrega.php:** El script muestra los archivos entregados por un alumno en una determinada entrega de una asignatura. Si hay archivos, los muestra en una lista con enlaces para descargarlos o abrirlos. Además, muestra un formulario para que el alumno pueda subir más archivos.

### 1.2.2 Sub Funcionalidad Recursos.

- **ContenidoAsignatura.php:** muestra una lista de recursos de la asignatura y una lista de entregas y eventos pendientes para esa asignatura. Si el usuario tiene el rol de profesor, se muestran los formularios para subir nuevos archivos y crear eventos o tareas

### 1.3 Funcionalidad Mensajes

- **chat.php:** este script genera una página de chat con una lista de nombres de chat y un formulario para enviar mensajes. La página utiliza una plantilla para generar la página completa y se llama a la función "generaVista()" de la aplicación para construir la página.

### 1.4 Funcionalidad Calendario

- **sidebarDer:** de la plantilla principal contiene siempre una instancia del objeto calendar.

### 1.5 Funcionalidad Log in/ Log out

- **index.php:** verifica si el usuario ha iniciado sesión y muestra el contenido principal de la página si lo ha hecho, y redirige al usuario a la página de inicio de sesión si no lo ha hecho.
- **cambioPassword.php:** muestra un formulario para cambiar la contraseña de un usuario. Para ello, se utiliza la clase FormularioNewPass. Luego, se genera una vista con el título de la página y el contenido principal que incluye el formulario mediante la función generaVista() del objeto \$app. Además, se utiliza la plantilla plantillaLogin.php que se encuentra en la carpeta plantillas.
- **login.php:** muestra un formulario de inicio de sesión en la página utilizando la clase FormularioLogin definida en el archivo FormularioLogin.php del directorio usuarios. Primero se carga el archivo de configuración (config.php), luego se crea una instancia del formulario de inicio de sesión (FormularioLogin) y se llama al método gestiona() para generar el formulario HTML.
- **registro.php:** crea un formulario de registro de usuario mediante la creación de una instancia de la clase FormularioRegistro, y luego utiliza el método gestiona() para obtener el HTML generado por el formulario. Se crea un título de página y contenido principal que incluyen el formulario de registro, y finalmente se genera una vista HTML utilizando la plantilla plantilla.php y se le pasan los parámetros correspondientes.

## 2. Listado de scripts adicionales

### 2.1 Scripts con definiciones de clases

- **Alumno.php:** el script define clase llamada "Alumno" que tiene métodos para buscar alumnos por ID, listar todos los alumnos, agregar y eliminar alumnos. La clase también tiene métodos para obtener y establecer asignaturas para cada alumno. La clase utiliza la funcionalidad "MagicProperties" y utiliza una instancia de la clase "Aplicación" para conectarse a la base de datos.
- **Padre.php:** define la clase "Padre". Esta clase tiene métodos para buscar, insertar y eliminar padres en la base de datos. También tiene un método para obtener los hijos de un padre a partir de su ID.
- **Profesor.php:** este script es una clase PHP llamada "Profesor", que contiene métodos para interactuar con una base de datos que almacena información de profesores, como su identificador, despacho, tutorías y asignaturas impartidas. La clase utiliza la funcionalidad "MagicProperties" para crear propiedades dinámicamente, y tiene métodos para insertar, borrar y buscar profesores en la base de datos, así como para obtener una lista de todos los profesores y sus asignaturas.
- **Ciclo.php:** clase que representa un ciclo educativo. Contiene métodos para crear, borrar y consultar.
- **Asignatura.php:** clase PHP que representa una asignatura. La clase tiene métodos para crear una nueva asignatura, buscar asignaturas por diferentes criterios y para insertar una asignatura en la base de datos. La clase también utiliza la clase Aplicación para obtener la conexión a la base de datos y realizar consultas.
- **Entregas\_Eventos.php:** define clase "Entregas\_Eventos" que se utiliza para manejar eventos relacionados con entregas de tareas en una aplicación web. La clase incluye métodos para crear, buscar y eliminar eventos por diferentes criterios. Además, la clase tiene propiedades privadas para almacenar información sobre el evento, como su ID y la fecha de finalización. Esta clase utiliza una instancia de la clase "Aplicación" a través del método getInstance() para obtener la conexión a la base de datos y realizar consultas.
- **Usuario.php:** Este script en PHP define una clase "Usuario" que se encarga de gestionar usuarios en una aplicación. La clase tiene métodos estáticos para buscar, crear y autenticar usuarios en la aplicación. También define constantes para los roles de usuario en la aplicación.
- **Calendar.php:** clase Calendar que puede usarse para generar un calendario HTML con la capacidad de agregar eventos a días específicos.
- **Recurso.php:** este script define una clase llamada "Recursos" que se utiliza para interactuar con recursos de una asignatura en una base de datos. Proporciona métodos para crear, obtener, actualizar y borrar recursos.
- **EntregasAlumno.php:** La clase EntregasAlumno tiene varios métodos estáticos que permiten crear, insertar y obtener entregas de los alumnos en una base de datos, tiene varios métodos para acceder y modificar las propiedades de una entrega. También tiene un método que borra la entrega de la base de datos.
- **Aplicación.php:** clase Aplicación, que se encarga de mantener el estado global de una aplicación. Tiene varios atributos, como la ruta de la aplicación, la conexión de la base de datos, y los atributos pendientes de la petición.

## 2.2 Scripts con lógica de la aplicación

- **Formulario.php:** Este script es una clase base para la gestión de formularios en PHP. Incluye métodos para generar mensajes de error, etiquetas HTML y atributos para formularios. Todos los formularios harán uso del método “generaCamposFormulario” para generar el html necesario del formulario y “procesaFormulario” que se encarga de validar los datos introducidos a través del formulario, crear un nuevo objeto con esos datos y por ultimo guardarlo en la base de datos:
  - **FormularioCreaAsignatura.php:** permite al administrador crear una nueva asignatura.
  - **FormularioCreaCiclo.php:** permite al administrador crear un nuevo ciclo educativo.
  - **FormularioNuevoUsuario.php:** permite al administrador crear un nuevo usuario, en este formulario podrá indicar el rol del nuevo usuario.
  - **FormularioNuevoAlumno.php:** permite añadir nuevos alumnos al campus.
  - **FormularioNuevoProfe.php:** permite al administrador dar de alta nuevos profesores.
  - **FormularioParticipantesAsignatura.php:** permite al administrador añadir alumnos a una asignatura.
  - **FormularioCreaEvento.php:** permite a los profesores crear un nuevo evento o tarea.
  - **FormularioLogin.php:** clase genera un formulario HTML para que los usuarios puedan iniciar sesión en el campus. Comprueba y valida los datos. Si son correctos, iniciará sesión en la aplicación y el usuario será redirigido a la página principal, si no, se mostrará un mensaje de error.
  - **FormularioLogout.php:** permite al usuario cerrar sesión en la aplicación.
  - **FormularioNewPass.php:** permite a un usuario cambiar su contraseña. El formulario incluye campos para el email del usuario, la nueva contraseña y la confirmación de la nueva contraseña. Si hay errores de validación, se muestran mensajes de error para que el usuario los corrija.
  - **FormularioRegistro.php:** esta clase es generar un formulario de registro para que los usuarios se registren en una aplicación web. El formulario contiene los siguientes campos de la tabla usuarios con una doble confirmación para la contraseña y un botón de envío "Registrar". Si los datos son válidos, se crea un nuevo usuario y se inicia sesión en la aplicación. Si hay errores en los datos, se generan mensajes de error para informar al usuario.
  - **FormularioSubeArchivo.php:** generar un formulario que permite a los usuarios subir un archivo. El formulario tiene un campo de entrada de archivo y un botón de envío. Cuando se envía el formulario, el archivo se carga en el servidor y se procesa. El método "procesaFormulario()" valida y mueve el archivo de la ubicación temporal a su ubicación correcta. También se definen métodos auxiliares para validar y sanitizar el nombre del archivo cargado.
  - **FormularioEditaEvento.php:** genera un formulario igual al de crear evento y permite a un profesor editar la información de un evento.

### 2.3 Scripts de abstracción del manejo de la base de datos

- **Alumnos.php**
- **Asignaturas.php**
- **Ciclo.php**
- **Calificaciones.php**
- **EntregasAlumno.php**
- **EstudianAsignaturas.php**
- **Eventos\_Tareas.php**
- **Padres.php**
- **Profesores.php**
- **Recursos.php**
- **Usuarios.php**

### 2.4 Script css

Para definir el estilo de nuestro campus hemos utilizado el script estilo.css

Contenedor:

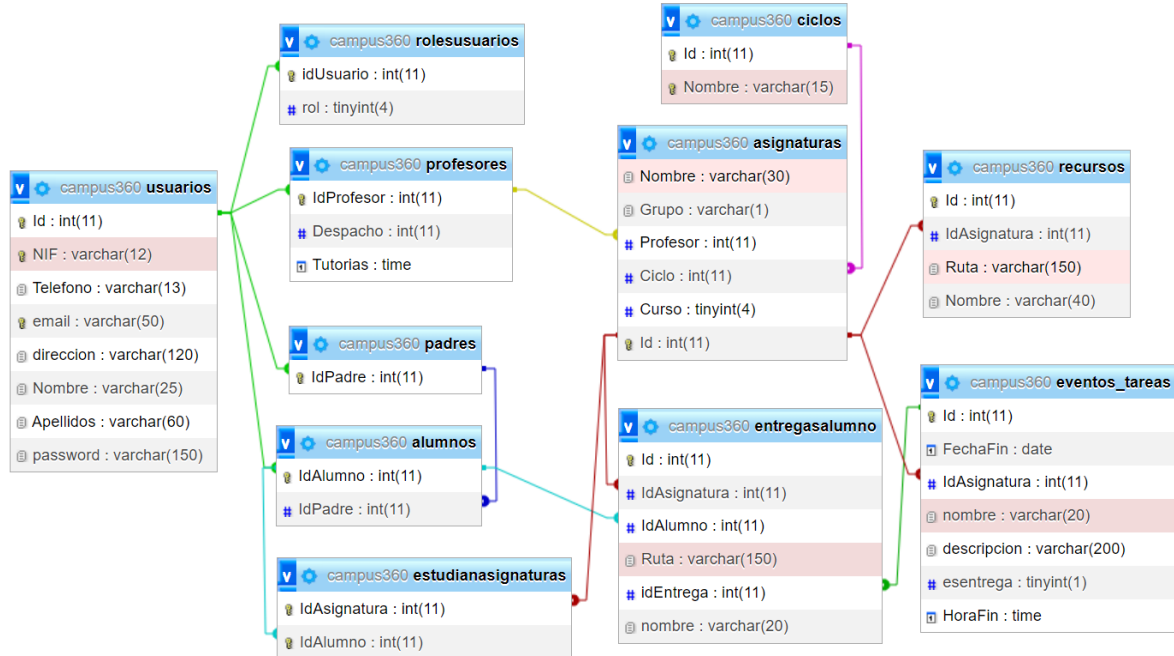
- **Header:** utiliza flex, contiene el nombre de la aplicación y la información de inicio de sesión
- **Container:** utiliza flex con flex-direction: row, para que al añadir en orden sidebarIzq, main y sidebarDer se ubiquen correctamente.
  - **SidebarIzq:** incluye en navegador (nav)
  - **Main:** contiene el contenido principal de la aplicación y exclusivamente en el foro se encuentra dentro otro contenedor(containerC)
    - **Foro:**
      - **ContainerC:** con flex, display flex, y con flex-direction: row, para que el listado de chats ocupe la mitad izquierda (left) y la redacción de mensajes ocupe la parte derecha (right)
  - **SidebarDer:** contiene el calendario (calendar).

Cabe destacar que también se han implementado los scripts correspondientes a la eliminación y edición de asignaturas, ciclos y usuarios por parte del administrador y también los relacionados con la edición y eliminación de eventos, tareas y entregas por parte de un profesor. Si se borra una asignatura, se eliminará también toda la información que estuviera relacionada con ella(recursos, tareas, eventos y entregas).



### 3. Estructura de la base de datos

#### 3.1 Modelo Relacional:



### 3.2 Tablas

#### 3.2.1 Alumnos.

##### 1 Alumnos

Creation: Apr 07, 2023 at 12:36 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
IdAlumno	int(11)		No			-> Usuarios.Id ON UPDATE CASCADE ON DELETE CASCADE		
IdPadre	int(11)		Yes	NULL		-> Padres.IdPadre ON UPDATE SET_NULL ON DELETE SET_NULL		

**Propósito:** en la tabla alumnos establecemos una relación directa entre un padre y un alumno, en la que un alumno es siempre revisado por un padre.

**Scripts que acceden a la tabla:** Alumnos.php (modifica y consulta)

**Campos:**

- IdAlumno: identifica a los alumnos. int(11)
- IdPadre: Clave foránea que se conecta con la tabla de padres para identificarlos. int (11)

#### 3.2.2 Asignaturas.

## 2 Asignaturas

Creation: Apr 06, 2023 at 05:00 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
Nombre	varchar(30)		No					
Grupo	varchar(1)		No					
Profesor	int(11)		No			-> Profesores.IdProfesor ON UPDATE CASCADE ON DELETE CASCADE		
Ciclo	int(11)		No			-> Ciclos.Id ON UPDATE CASCADE ON DELETE CASCADE		
Curso	tinyint(4)		No					
id	int(11)		No		auto_increment			

**Propósito:** Almacenar la información relacionada con una asignatura y poder identificar cada asignatura con un id único que se asigna de manera automática y de forma incremental.

**Scripts que acceden a la tabla:** Asignaturas.php (modifica y consulta)

### Campos:

- nombre: nombre de la asignatura. varchar(30)
- grupo: letra de la clase. varchar(1)
- profesor: Clave foránea que conecta a un profesor con una asignatura que imparte. int (11)
- ciclo: Clave foránea que conecta una asignatura con el ciclo al que pertenece. varchar(11)
- curso: curso en el que se imparte la asignatura. [1,4]. tinyint(4), ya que existe un máximo de cuatro cursos por ciclo.
- id: identificador de la asignatura, se genera de manera automática y de forma incremental.

### 3.2.3 Ciclo.

## 4 Ciclos

Creation: Apr 04, 2023 at 01:34 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
Nombre	varchar(15)		No					

**Propósito:** Almacena los distintos ciclos disponibles en el campus.

**Scripts que acceden a la tabla:** Ciclo.php (modifica y consulta)

### Campos:

- id: identificador del ciclo. int(11)
- nombre: nombre del ciclo. varchar(15)

### 3.2.4 EntregasAlumno.

## 5 EntregasAlumno

Creation: Apr 11, 2023 at 08:53 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
Id	int(11)		No		auto_increment			
IdAsignatura	int(11)		No			-> Asignaturas.Id ON UPDATE CASCADE ON DELETE CASCADE		
IdAlumno	int(11)		No			-> Alumnos.IdAlumno ON UPDATE CASCADE ON DELETE CASCADE		
Ruta	varchar(150)		No					
idEntrega	int(11)		No			-> Eventos_Tareas.Id ON UPDATE CASCADE ON DELETE CASCADE		
nombre	varchar(20)		No					

**Propósito:** identificar y almacenar las distintas versiones que hace un alumno de una entrega.

**Scripts que acceden a la tabla:** EntregasAlumno.php (modifica y consulta)

### Campos:

- id: identifica de manera inequívoca la entrega. int(11) se genera de forma automática y con valores incrementales.
- idAsignatura: clave foránea que relaciona la entrega con la asignatura a la que pertenece. int(11)
- idAlumno: clave foránea que relaciona al alumno con su entrega. int(11)
- ruta: ubicación del archivo donde se encuentra la entrega. varchar(150)
- idEntrega: clave foránea que relaciona el archivo subido por el alumno con la tarea correspondiente. int(11)
- nombre: nombre que identifica al archivo. varchar(20)

### 3.2.5 EstudianAsignaturas.

## 6 EstudianAsignaturas

Creation: Apr 06, 2023 at 05:13 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
IdAsignatura	int(11)		No			-> Asignaturas.Id ON UPDATE CASCADE ON DELETE CASCADE		
IdAlumno	int(11)		No			-> Alumnos.IdAlumno ON UPDATE CASCADE ON DELETE CASCADE		

**Propósito:** mantener una lista sobre los alumnos que cursan cada asignatura.

**Scripts que acceden a la tabla:** Asignaturas.php y Alumnos.php (modifica y consulta)

### Campos:

- idAsignatura: clave foránea que identifica la asignatura. int(11) se corresponde al id de

## ASIGNATURAS

- idAlumno: clave foránea identifica al alumno. int(11) se corresponde con el idAlumno de ALUMNOS.

### 3.2.6 Eventos\_tareas.

#### 7 Eventos\_Tareas

Creation: Apr 13, 2023 at 09:25 AM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
Id	int(11)		No		auto_increment			
FechaFin	date		No					
IdAsignatura	int(11)		No			-> Asignaturas.Id ON UPDATE CASCADE ON DELETE CASCADE		
nombre	varchar(20)		No					
descripcion	varchar(200)		No					
esentrega	tinyint(1)		No					
HoraFin	time		No					

**Propósito:** almacena la información sobre los distintos eventos.

**Scripts que acceden a la tabla:** Eventos\_Tareas.php (modifica y consulta)

#### Campos:

- id: identifica un evento concreto. int(11).
- fechaFiN: fecha de expiración del evento. date.
- idAsignatura: clave foránea que relaciona el evento/tarea con una asignatura. int(11)
- nombre: nombre del evento/tarea. varchar(20)
- descripción: breve descripción del evento/tarea. varchar(200)
- esentrega: utilizamos este campo para saber si tenemos una tarea(1) o un evento(0). tinyint(1).

### 3.2.7 Padres.

#### 8 Padres

Creation: Apr 06, 2023 at 05:08 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
IdPadre	int(11)		No			-> Usuarios.Id ON UPDATE CASCADE ON DELETE CASCADE		

**Propósito:** listado de los padres

**Scripts que acceden a la tabla:** Padres.php (modifica y consulta)

#### Campos:

- idPadre: clave foránea que identifica a un padre como usuario. int(11). No nulo.

### 3.2.8 Profesores.

#### 9 Profesores

Creation: Apr 06, 2023 at 05:00 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
IdProfesor	int(11)		No			-> Usuarios.Id ON UPDATE CASCADE ON DELETE CASCADE		
Despacho	int(11)		No					
Tutorias	time		No					

**Propósito:** información acerca de los despachos y tutorías de los profesores.

**Scripts que acceden a la tabla:** Profesores.php (modifica y consulta)

**Campos:**

- idProfesor: clave foránea que identifica un profesor como usuario.
- despacho: número de despacho del profesor.
- tutorías: hora a la que el profesor realiza sus sesiones de tutorías.

### 3.2.9 Recursos.

#### 10 Recursos

Creation: Apr 13, 2023 at 09:26 AM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
idAsignatura	int(11)		No			-> Asignaturas.Id ON UPDATE CASCADE ON DELETE CASCADE		
Ruta	varchar(150)		No					
Nombre	varchar(40)		No					

**Propósito:** tener identificados los recursos, la asignatura a la que pertenecen y su ubicación.

**Scripts que acceden a la tabla:** Recursos.php (modifica y consulta)

**Campos:**

- id: identifica recursos de manera inequívoca, asigna este valor de forma automática al crear el recurso y los valores se van incrementando. int (11)
- idAsignatura: clave foránea que relaciona un recurso con la asignatura a la que pertenece. int (11)
- ruta: ubicación del recurso en el sistema. varchar(150)
- nombre: nombre del archivo. varchar(40)

### 3.2.10 RolesUsuario.

#### 11 RolesUsuarios

Creation: Apr 06, 2023 at 05:00 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
idUsuario	int(11)		No			-> Usuarios.Id ON UPDATE CASCADE ON DELETE CASCADE		
rol	tinyint(4)		No					

**Propósito:** nos permite saber que rol tiene cada usuario en la aplicación **Scripts que acceden a la tabla:** Usuarios.php (modifica y consulta)

**Campos:**

- idUsuario: clave foránea que identifica al usuario. int (11).
- rol: tinyint(4) Admin(1), Alumno(2), Padre(3), Profe(4)

### 3.2.11 Usuarios.

## 12 Usuarios

Creation: Apr 04, 2023 at 01:34 PM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
Id	int(11)		No		auto_increment			
NIF	varchar(12)		No					
Telefono	varchar(13)		No					
email	varchar(50)		No					
direccion	varchar(120)		No					
Nombre	varchar(25)		No					
Apellidos	varchar(60)		No					
password	varchar(150)		No					

**Propósito:** tabla que contenga toda la información común de los usuarios.

**Scripts que acceden a la tabla:** Usuarios.php (modifica y consulta)

### Campos:

- id: identificador de usuario. int (11), asigna de forma incremental y automática al crearse un nuevo usuario. No nulo.
- NIF: documento nacional de identidad. varchar(12). No nulo.
- Teléfono: teléfono del usuario. varchar(13). No nulo.
- email: correo electrónico. varchar(50). No nulo.
- dirección: dirección del usuario. varchar(120). No nulo.
- nombre: nombre de usuario. varchar(25). No nulo.
- apellidos: apellidos del usuario varchar(60) No nulo.
- password: contraseña del usuario. varchar (150). No nulo. Se almacena un hash y no la contraseña por seguridad.

#### 4. Datos de acceso

☐ **Acceso alumno:**

[user@campus.es](mailto:user@campus.es)

contraseña: userpass

☐ **Acceso profesor:**

[profe@campus.es](mailto:profe@campus.es)

contraseña: userpass

☐ **Acceso padre:**

[padre@campus.es](mailto:padre@campus.es)

contraseña: userpass

☐ **Acceso admin**

[admin@campus.es](mailto:admin@campus.es)

contraseña: userpass

#### 5. Parte de actividades

<b>Diego Pellicer Lafuente</b>	Funcionalidad de gestión de cursos
<b>Andrés Sánchez Casas</b>	Funcionalidad gestión de recursos en un curso
<b>Martin Ruano Crespo</b>	CSS de la aplicación
<b>Lucia Blasco Gallego</b>	

