



Herramienta  
Dinámica de  
Generación de CVs  
Personalizados

Noviembre

2013

Autor: Alejandro García Pérez  
Director: Fernando Cerdán Cartagena  
Codirectores: Diego García Sánchez  
Juan Suardiáz Muro

UPCT

# Descripción bibliográfica

---

- **Autor:** Alejandro García Pérez
- **E-mail del Autor:** [alejandro.g.murcia@gmail.com](mailto:alejandro.g.murcia@gmail.com)
  
- **Director:** Fernando Cerdán Cartagena
- **E-mail del Director:** [fernando.cerdan@upct.es](mailto:fernando.cerdan@upct.es)
  
- **Codirectores:** Diego García Sánchez y Juan Suardfáz Muro.
  
- **Título del PFC:** Herramienta Dinámica de Generación de CVs Personalizados.
  
- **Descriptores:** PHPrunner, HTML, PHP, SQL, Bases de datos, CV, aplicación web
  
- **Resumen:**

Diseño e implementación de una aplicación web dinámica en la cual se puedan introducir CVs completos y generar plantillas de impresión a partir de los mismos.

Pensada para poder cambiar la configuración establecida de los datos a introducir según las necesidades del usuario (pudiendo añadir campos nuevos) y la posterior salida de los datos en distintos formatos de impresión.
  
- **Titulación:** Ingeniería técnica de telecomunicaciones, especialidad en Telemática.
  
- **Departamento:** Tecnologías de la Información y las comunicaciones.
  
- **Fecha de Presentación:** Noviembre, 2013

# Índice

---

## Contenido

Descripción bibliográfica .....	1
Índice.....	2
<b>1. Introducción.....</b>	<b>3</b>
1.1. Planteamiento inicial .....	3
1.2. Objetivos .....	3
1.3. Organización de la memoria.....	4
<b>2. Herramientas de trabajo.....</b>	<b>5</b>
2.1. Bases de datos .....	5
2.2. Aplicación XAMPP.....	8
2.3. PHPRunner.....	10
<b>3. Desarrollo de la aplicación web.....</b>	<b>21</b>
3.1. Diseño de la base de datos.....	21
3.2. Relación entre tablas .....	36
3.3. Implementación de la interfaz web .....	38
<b>4. Manual de usuario.....</b>	<b>46</b>
<b>5. Conclusiones .....</b>	<b>52</b>
5.1. Dificultades encontradas .....	52
5.2. Líneas futuras de trabajo .....	53
<b>6. Anexo: Bibliografía y referencias .....</b>	<b>55</b>

# CAPÍTULO 1

## 1. Introducción

En esta memoria se detallarán los motivos que nos han llevado a la elaboración de este proyecto, así como todo el proceso de creación de la herramienta web y los recursos empleados para llevarlo a cabo como lenguajes de programación, aplicación de diseño web, etcétera. También se incluirá un manual de usuario donde se explicará cómo utilizar la herramienta de manera adecuada y por último vendrá toda la bibliografía que nos ha servido de soporte para la creación del mismo.

### 1.1. Planteamiento inicial

Se ha detectado una necesidad por la creación de formatos distintos de CVs en función del destino de los mismos. A pesar de que la información es la misma, la forma en que debe presentarse y los rangos de fechas pueden variar considerablemente de un punto de destino a otro. Por ello se plantea una herramienta donde almacenar y gestionar la información de forma global, para que el usuario diseñe el formato de la información en función del destino de presentación, ya sea una convocatoria de becas, proyectos, búsqueda de empleo, etcétera.

### 1.2. Objetivos

- 1) Desarrollar una aplicación de usuario en web 2.0
  - a. Almacenamiento y gestión de la información de manera global
  - b. Personalización de la información
  - c. Familiarizarse y trabajar con el entorno de desarrollo de una web 2.0 mediante PHP desde la aplicación PHPRunner
- 2) Iniciar el desarrollo de aplicaciones completamente dinámicas

### 1.3. Organización de la memoria

La memoria se estructura de la siguiente manera:

- **Herramientas de trabajo:** Se explicará a grandes rasgos qué es una base de datos y cómo trabajar con ella, así como las herramientas empleadas para la creación web y los lenguajes de programación empleados, siendo XAMPP y PHPrunner las principales aplicaciones sobre las que se soporta el proyecto.
- **Desarrollo de la aplicación:** Donde se detallarán los primeros pasos y el proceso de creación de la web y la herramienta de generación de CVs, explicando desde el formato de cada una de las tablas existentes en la base de datos del proyecto así como la relación existente entre ellas y por último veremos el desarrollo de la interfaz web
- **Manual de usuario:** Aquí veremos cómo utilizar la aplicación web a nivel de usuario desde el registro hasta como modificar las tablas de la base de datos añadiendo entradas viendo con detalle todo el entorno web de la aplicación.
- **Conclusiones:** Ideas principales a las que se ha llegado tras la utilización de las herramientas así como problemas encontrados a lo largo de la creación de la aplicación y las líneas futuras de trabajo que se pretenden seguir con este proyecto.
- **Anexo:** Bibliografía y referencias

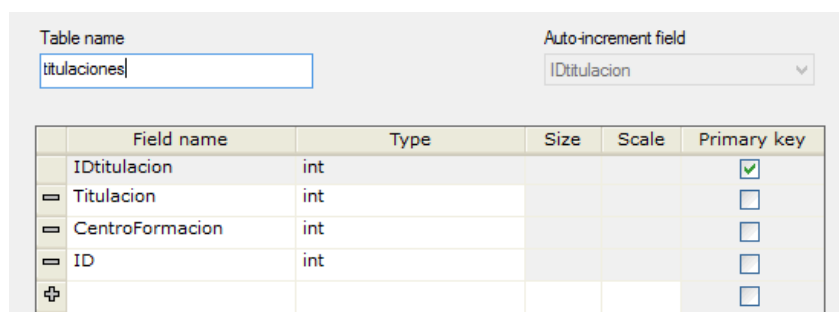
# CAPÍTULO 2

## 2. Herramientas de trabajo

### 2.1. Bases de datos

Este proyecto está basado en la elaboración de una base de datos que contiene una serie de tablas relacionadas entre sí. De ésta manera podemos tener organizada toda la información del CV cada usuario y trabajar con ellas de forma ordenada.

Una base de datos se define como un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Consta principalmente de un conjunto de tablas que están relacionadas entre sí. Cada una de las tablas contiene una serie de campos de información y estos, a su vez, la información útil. Para explicar mejor cada una de las partes de las que consta una tabla se utilizará el siguiente ejemplo:



The screenshot shows a database table definition interface. At the top, there is a text input field for the table name containing 'titulaciones' and a dropdown menu for the auto-increment field set to 'IDtitulacion'. Below this is a table with the following structure:

	Field name	Type	Size	Scale	Primary key
	IDtitulacion	int			<input checked="" type="checkbox"/>
▢	Titulacion	int			<input type="checkbox"/>
▢	CentroFormacion	int			<input type="checkbox"/>
▢	ID	int			<input type="checkbox"/>
⊕					<input type="checkbox"/>

Figura 1 – Ejemplo de una tabla en una base de datos

En el ejemplo de la figura 1 vemos una tabla llamada *titulaciones* que contiene tres campos de información: *IDtitulacion*, *Titulacion*, *CentroFormacion* e *ID*. En este caso los tres campos son de tipo entero (int). Existe un identificador específico para cada uno de los registros que introduzcamos en la tabla y se “Primary Key”, en nuestro ejemplo este valor lo tenemos en el campo *IDtitulacion*, que además se autoincrementa con cada registro. Un registro consta de un valor en cada uno de los campos que forman la tabla.

Como hemos dicho anteriormente, las tablas pueden relacionarse entre sí, un ejemplo de relación entre tablas puede ser el siguiente:

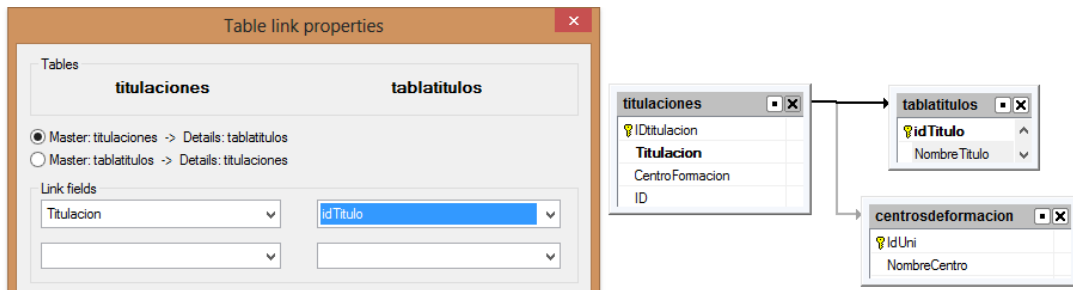


Figura 2 – Relación entre tablas

En la figura 2 vemos que la tabla *titulaciones* está relacionada con las tablas *tablatitulos* y *centrosdeformacion*, además observamos en detalle la relación entre las dos primeras, como se puede apreciar los campos relacionados son el campo *Titulacion* perteneciente a la tabla *titulaciones* y el campo *idTitulo*, que además es la clave primaria de la tabla *tablatitulos*. Esto quiere decir que el valor del campo *Titulacion* será el identificador específico de uno de los registros de la tabla *tablatitulos*. Un ejemplo más específico sería por ejemplo tener un registro específico en la tabla *tablatitulos* que tendría los siguientes valores:

IdTitulo = 1

NombreTitulo = Ingeniero Técnico

Cuando yo especifique que mi titulación es “Ingeniero Técnico”, el valor del campo *Titulacion* de mi tabla *titulaciones* será el identificador específico del registro de la tabla *tablatitulos*, en este ejemplo sería 1.

Se pueden distinguir tres tipos de relaciones entre tablas, estos son:

- **Relación Uno a Uno:** Un registro de una tabla solo se relaciona con un único registro de otra tabla, y viceversa. Ejemplo: una población puede tener sólo un alcalde y un alcalde lo será únicamente de una población.

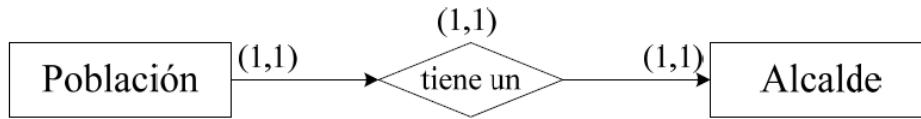


Figura 3 – Relación Uno a Uno

- **Relación Uno a Varios:** Un registro de una tabla puede relacionarse con varios registros de otras tablas. Ejemplo: una tabla de *Cientes* y otra de *pedidos*; cada cliente puede realizar varios pedidos.

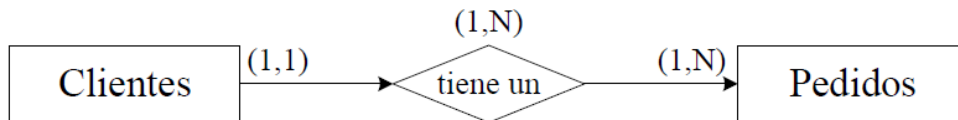


Figura 4 – Relación Uno a Varios

- **Relación Varios a Varios:** Varios registros de una tabla pueden relacionarse con varios registros de otras. Ejemplo: Una tabla *Productos* y otra *Pedidos*. Un pedido puede incluir varios productos, a su vez un único producto puede aparecer en varios pedidos.

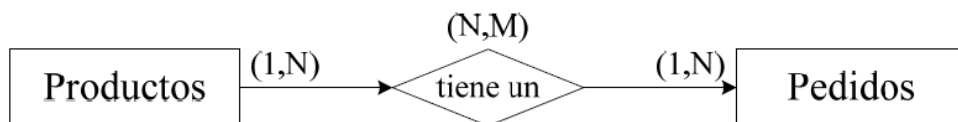


Figura 5 – Relación Varios a Varios

Una vez conocidos los principales aspectos de una base de datos (SGBD), procedemos a ver lo que es un sistema gestor de bases de datos. Se trata de un sistema de software que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación. Un SGBD relacional es un modelo de datos que facilita a los usuarios describir los datos que serán almacenados en la base de datos junto con un grupo de operaciones para manejar los datos.

Los SGBD relacionales son una herramienta efectiva que permite a varios usuarios acceder a los datos al mismo tiempo. Brindan facilidades eficientes y un grupo de funciones con el objetivo de garantizar la confidencialidad, la calidad, la seguridad y la integridad de los datos que contienen, así como un acceso fácil y eficiente a los mismos.



Para este proyecto se ha decidido utilizar MySQL como sistema de gestión de bases de datos, principalmente por ser un sistema de código abierto muy utilizado en aplicaciones web y está muy ligado a PHP (Lenguaje de programación diseñado para la creación de páginas web dinámicas).

Para que todo esto funcione se necesita instalar un servidor web, ya que es necesario para poder gestionar las aplicaciones del lado del servidor, en nuestro caso se ha optado por utilizar Apache por ser también de código abierto.

## 2.2. Aplicación XAMPP

El proceso de instalación de todas las aplicaciones mencionadas anteriormente es un tanto tedioso y para facilitar esta parte del proyecto se ha decidido utilizar una herramienta llamada XAMPP, esta aplicación instala un servidor web Apache, además de MySQL, PHP y Perl (este último no será utilizado en este proyecto). Actualmente existen cuatro versiones del programa distintas dependiendo del sistema operativo en el que vayamos a trabajar (Linux, Windows, Mac OS X y Solaris), siendo todas gratuitas.

Este programa genera una carpeta a nivel local a la cual se puede acceder desde el navegador por medio de la dirección “*http://localhost/xampp/*” Para poder acceder a esta localización primero debemos arrancar el programa

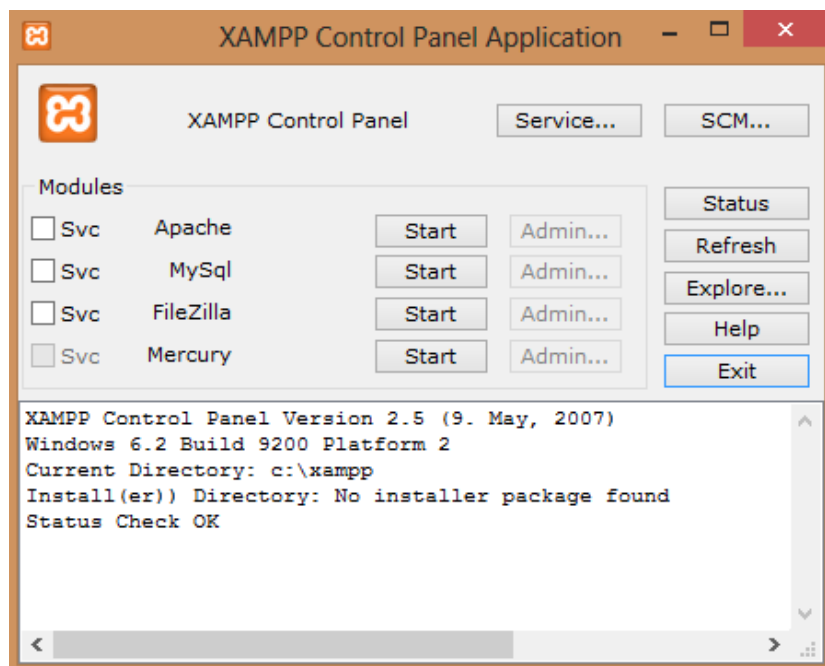


Figura 6 – Inicio aplicación XAMPP

A continuación activaremos el servidor apache y el SGBD MySQL

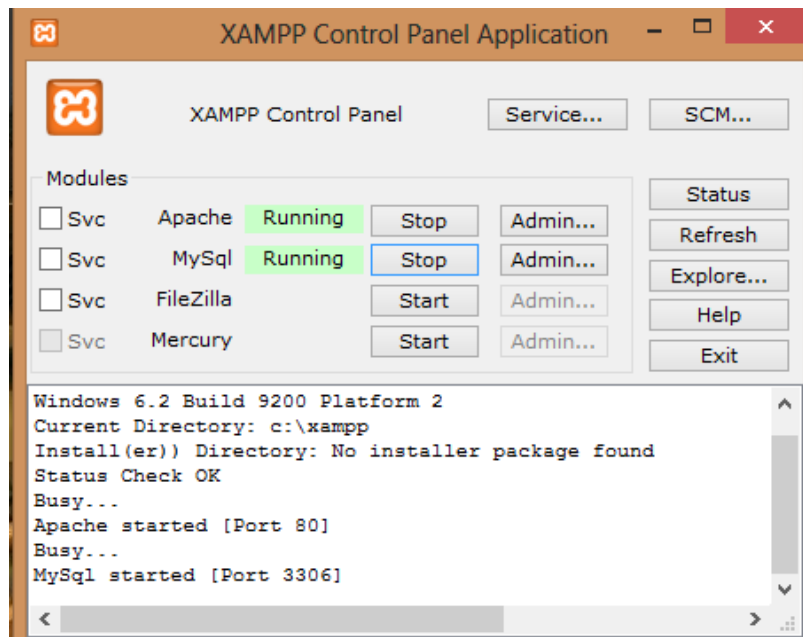


Figura 7 – Iniciación Apache y MySQL en XAMPP

Como podemos ver, la aplicación nos ha arrancado los procesos del servidor Apache y del SGBD MySQL en los puertos 80 y 3306 respectivamente. Ahora si entramos en la dirección <http://localhost/xampp/> desde nuestro navegador web podemos ver la aplicación XAMPP:



Figura 8 – Aplicación XAMPP en el navegador

Lo que vemos rodeado de un círculo azul en la Figura 8 es la herramienta phpMyAdmin proporcionada por XAMPP, donde podemos ver todas las bases de datos

que tenemos almacenadas en nuestro directorio local y poder trabajar con ellas de una manera sencilla.

En la siguiente figura vemos la base de datos correspondiente a este proyecto, donde encontramos todas las tablas correspondientes al mismo con su estructura específica y las diferentes opciones que nos presta la aplicación para trabajar con ellas, así como consultas SQL a las tablas, búsquedas, exportación e importación de información y demás modificaciones.

The screenshot shows the phpMyAdmin interface for a database named 'pfc'. The left sidebar lists various tables, and the main area displays a table with columns: Tabla, Acción, Registros, Tipo, Cotejamiento, Tamaño, and Residuo a depurar. The table lists 26 tables with their respective record counts and sizes.

Tabla	Acción	Registros	Tipo	Cotejamiento	Tamaño	Residuo a depurar
actividadesidi		0	InnoDB	utf8_general_ci	16.0 KB	-
ayudas		0	InnoDB	utf8_general_ci	16.0 KB	-
becas		0	InnoDB	utf8_general_ci	16.0 KB	-
centrosdeformacion		76	InnoDB	utf8_general_ci	16.0 KB	-
contratosi		0	InnoDB	latin1_swedish_ci	16.0 KB	-
contratosn		0	InnoDB	latin1_swedish_ci	16.0 KB	-
contribucionescong		0	InnoDB	utf8_general_ci	16.0 KB	-
contribucionescong		0	InnoDB	latin1_swedish_ci	16.0 KB	-
country		239	InnoDB	utf8_general_ci	48.0 KB	-
datospersonales		0	InnoDB	utf8_general_ci	16.0 KB	-
docencia		0	InnoDB	utf8_general_ci	16.0 KB	-
doctorados		0	InnoDB	utf8_general_ci	16.0 KB	-
informesinvestigacion		0	InnoDB	utf8_general_ci	16.0 KB	-
masters		0	InnoDB	utf8_general_ci	16.0 KB	-
masters		0	InnoDB	utf8_general_ci	16.0 KB	-
otros		0	InnoDB	utf8_general_ci	16.0 KB	-
patentesymodelos		0	InnoDB	utf8_general_ci	16.0 KB	-
proyn		0	InnoDB	utf8_general_ci	16.0 KB	-
publii		0	InnoDB	utf8_general_ci	16.0 KB	-
publin		0	InnoDB	utf8_general_ci	16.0 KB	-
tabladocotorados		0	InnoDB	utf8_general_ci	16.0 KB	-
tablaentradas		181	InnoDB	latin1_swedish_ci	48.0 KB	-
tablaentradas		0	InnoDB	utf8_general_ci	16.0 KB	-
tablamasters		560	InnoDB	latin1_swedish_ci	80.0 KB	-
tablatitulos		142	InnoDB	utf8_general_ci	16.0 KB	-

Figura 9 – Herramienta phpMyAdmin

### 2.3. PHPRunner

Una vez tengamos todas las herramientas descritas anteriormente listas para utilizar se procederá a la elaboración de la base de datos, así como del entorno web para poder utilizar la aplicación. Para esto se ha decidido aprovechar la herramienta PHPRunner, ya que se trata de un programa pensado para desarrollar entornos web basados en bases de datos, lo que hace que sea perfecto para nuestros propósitos.

La aplicación PHPRunner se divide en 11 pantallas de trabajo, las cuales explicaremos cada una más adelante. Lo primero que hay que tener en cuenta para utilizar la aplicación es que debemos disponer de una dirección donde tendremos nuestras bases de datos para poder trabajar con ellas, ya sea una dirección local o remota, por esta razón el primer paso para poder trabajar con PHPRunner es arrancar la aplicación XAMPP y correr tanto el servidor Apache como el SGBD MySQL. Una vez hecho esto debemos ir a la herramienta PHPRunner y decidir si queremos abrir un proyecto ya creado o crear un proyecto nuevo.

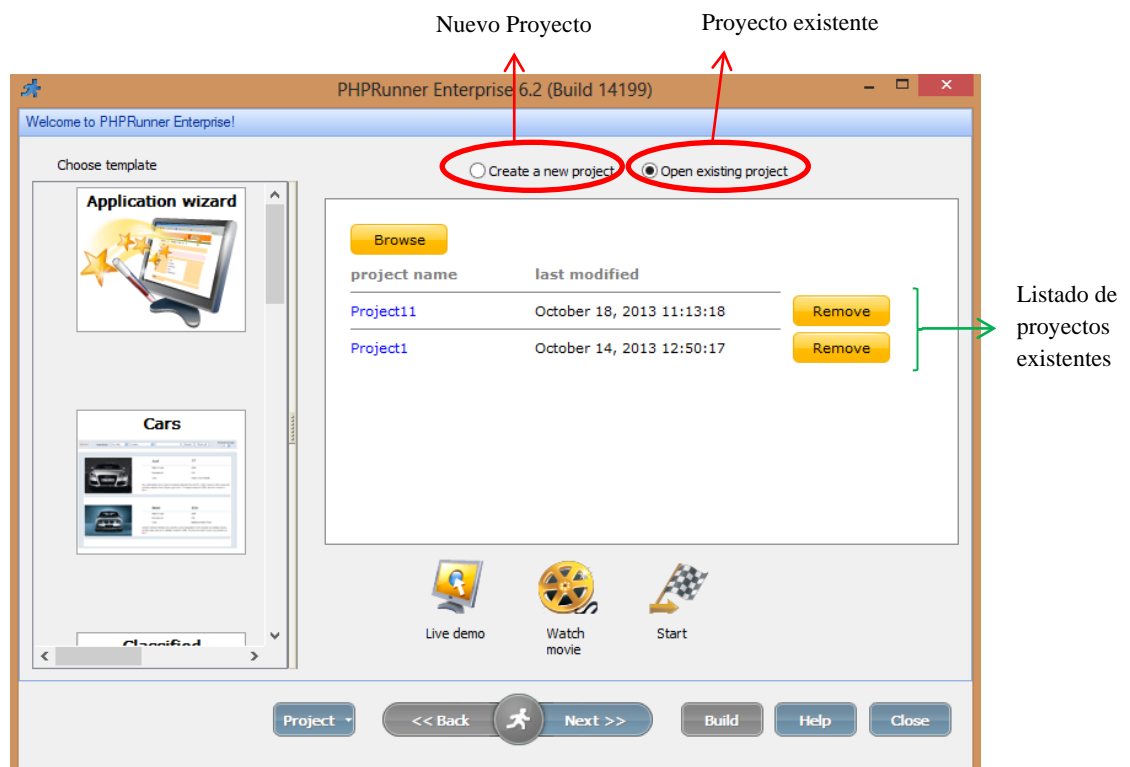


Figura 10 – PHPRunner inicio

Como vemos en la Figura 10, en la parte izquierda tenemos una serie de plantillas que podemos utilizar a la hora de crear un proyecto nuevo si alguna se ajusta a nuestro propósito o podemos seleccionar la opción “Application wizard” para empezar todo el entorno web desde cero sin tablas creadas. Una vez hayamos escogido la plantilla que queremos nos preguntará por el SGBD que queremos utilizar.

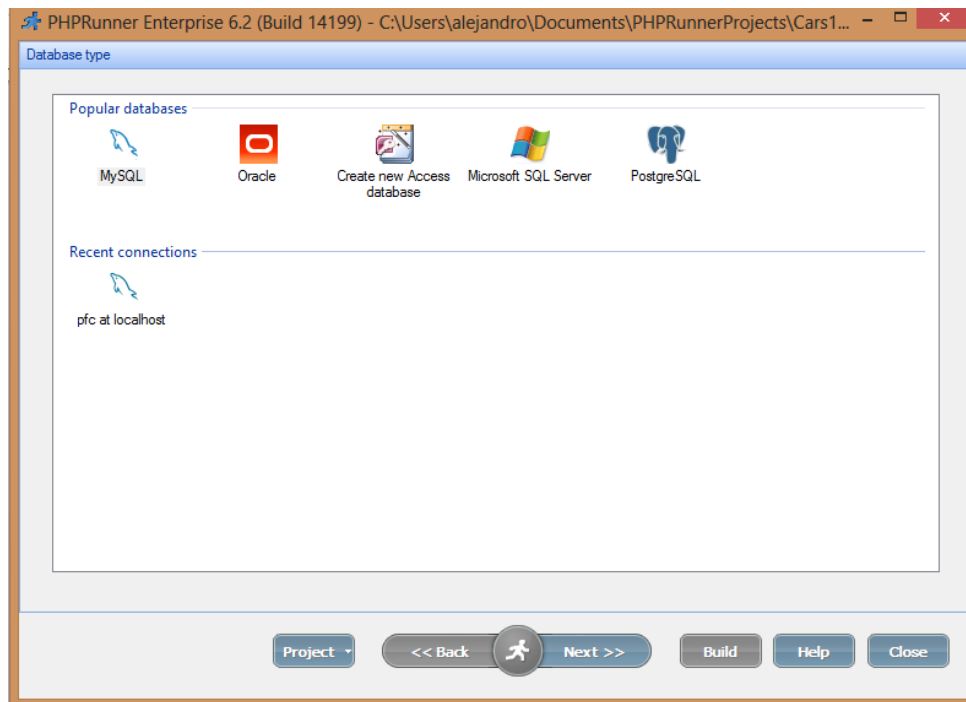


Figura 11 – PHPRunner seleccionar SGBD

En nuestro caso será MySQL. A continuación el programa nos pedirá que le indiquemos donde está ubicada la base de datos para poder trabajar con ella.

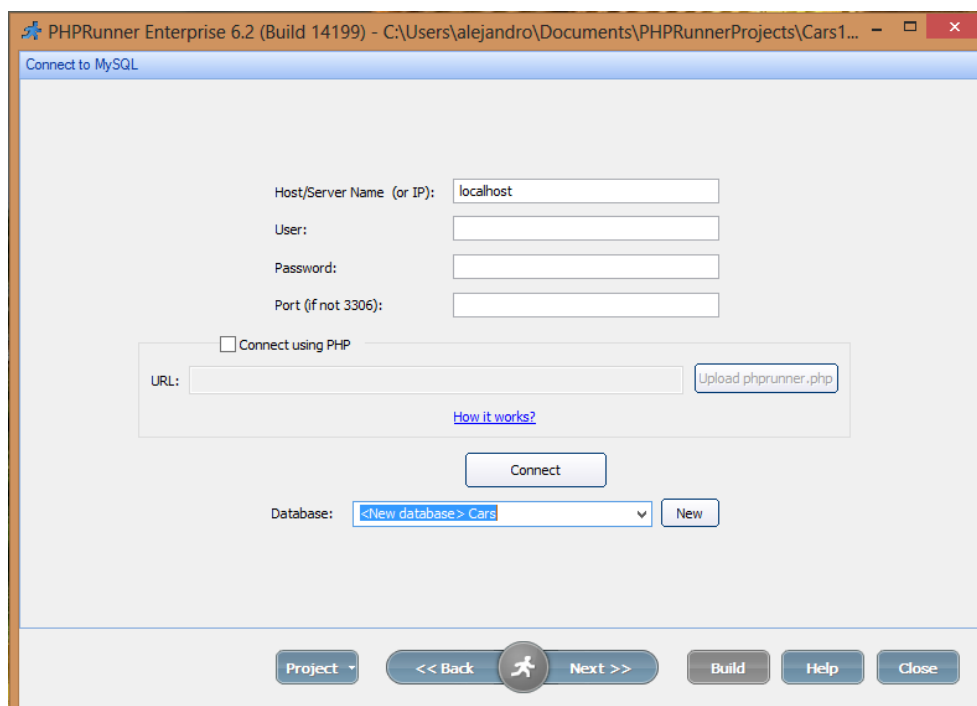


Figura 12 – PHPRunner seleccionar base de datos

Para explicar el funcionamiento del programa se ha elegido la plantilla “Cars”, seleccionamos el host local como ubicación de la misma y como no tenemos creada dicha base de datos, el programa se encargara de crearla añadiendo automáticamente las tablas de la misma, otra opción puede ser seleccionar una base de datos ya creada donde se incluirán estas tablas a las que ya tengamos. Una vez hecho esto se procederá a trabajar sobre el proyecto, pudiendo movernos por un total de 11 pantallas de edición distintas, para movernos por ellas debemos seleccionarlas en el icono central de la parte inferior programa, estas son:



Figura 13 – Pantallas de trabajo en PHPRunner

## Tables

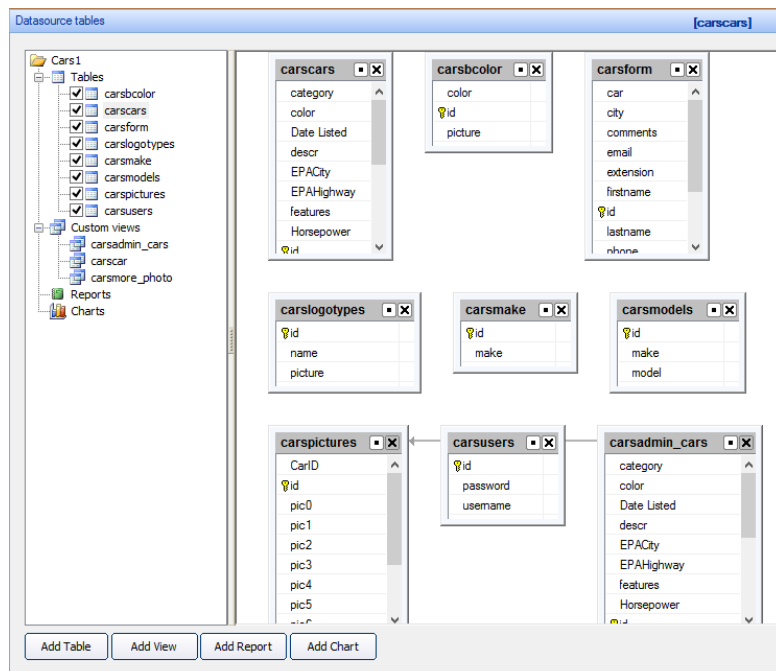


Figura 14 – Tablas PHPRunner

Es la primera pantalla que nos aparece cuando abrimos un proyecto, en la misma podemos ver en la parte izquierda el listado de las tablas que tenemos en la base de datos

seleccionada anteriormente. En la parte derecha tenemos la vista de cada una de las tablas que utilizaremos en la creación de la web y las relaciones que hay entre ellas. En la parte inferior vemos cuatro botones de edición para trabajar con las tablas. Podemos sincronizar todas las tablas con nuestra base de datos en cualquier momento, de forma que se puede trabajar desde el propio PHPRunner, así como desde nuestro servidor a través del portal web, con la herramienta PhpMyAdmin proporcionada por XAMPP.

## Query

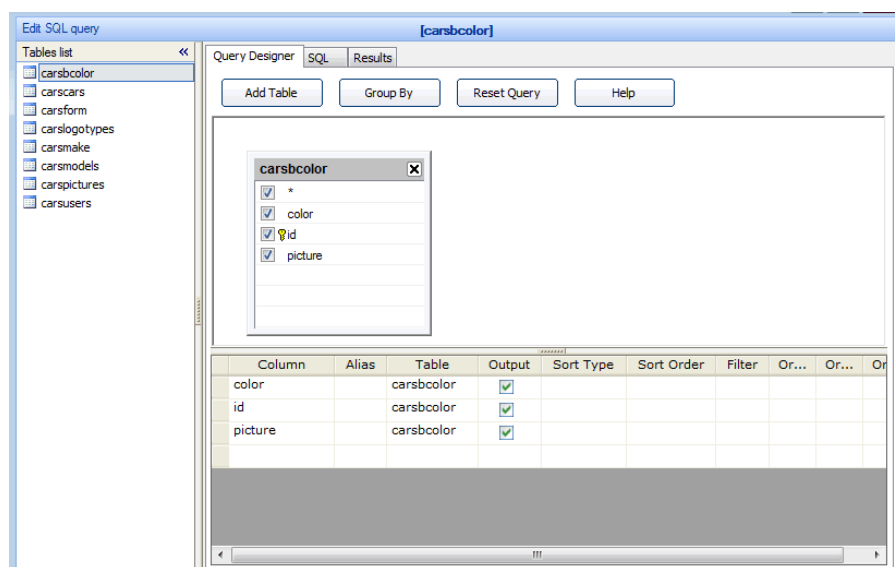


Figura 15 – Consultas PHPRunner

En esta pantalla se pueden generar consultas SQL de las tablas que tengamos en nuestra base de datos asociada al proyecto. En esta pantalla podemos ver una lista de nuestras tablas en la parte izquierda y a la derecha el entorno de trabajo, que tiene 3 pestañas en la parte superior por las que podemos navegar. En la primera pestaña podemos diseñar consultas SQL de manera visual relacionando los elementos de nuestras tablas, en la segunda se pueden escribir directamente sentencias SQL y por último vemos los resultados que producen las sentencias SQL de cada tabla. Esta página no la hemos tocado mucho en este proyecto, ya que cuando hemos necesitado ver la información de la base de datos hemos utilizado la aplicación web que proporciona XAMPP ya que es más completa.

## Pages

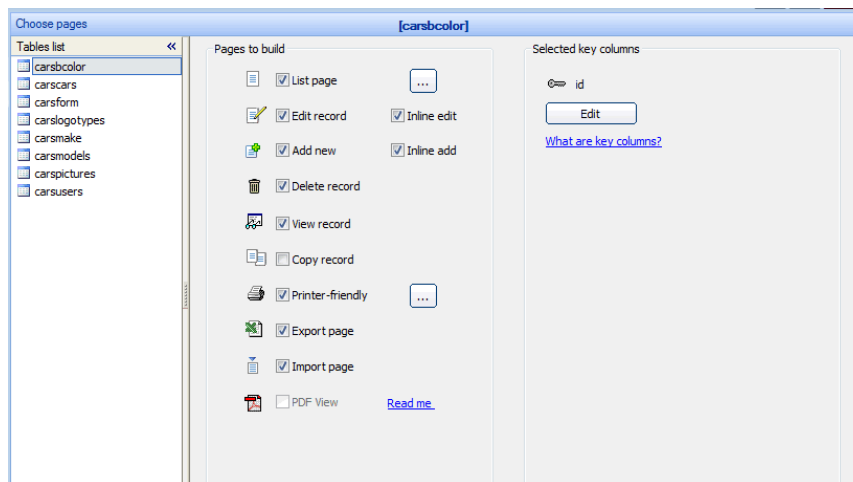


Figura 16 – Páginas PHPRunner

En la parte izquierda de esta pantalla vemos las tablas correspondientes a la base de datos asociada al proyecto. Aquí especificaremos qué páginas queremos que el usuario pueda ver en la web para cada una de las tablas. Si quisiéramos una página para añadir campos de una tabla, o para exportar la información, eliminarla o cualquiera de las otras opciones deberemos especificarlo en esta pantalla.

## Fields

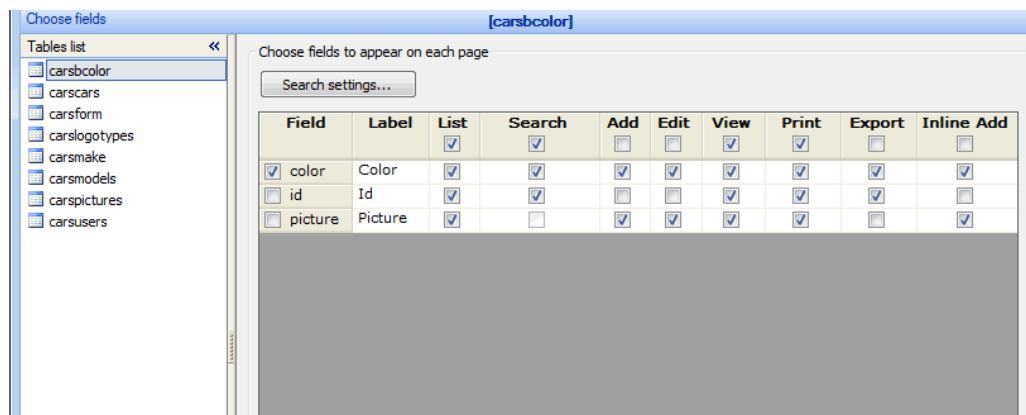


Figura 17 – Campos PHPRunner

En esta pantalla también disponemos de toda nuestra base de datos en la parte izquierda. Aquí podemos especificar qué campos de nuestra tabla queremos que aparezcan en cada una de las páginas. En el ejemplo de la Figura 17 vemos que el campo “color”



aparece en todas las páginas correspondientes a la tabla “carsbcolor”, sin embargo vemos que el campo “picture” no aparece en las páginas de búsqueda y exportación

## Totals

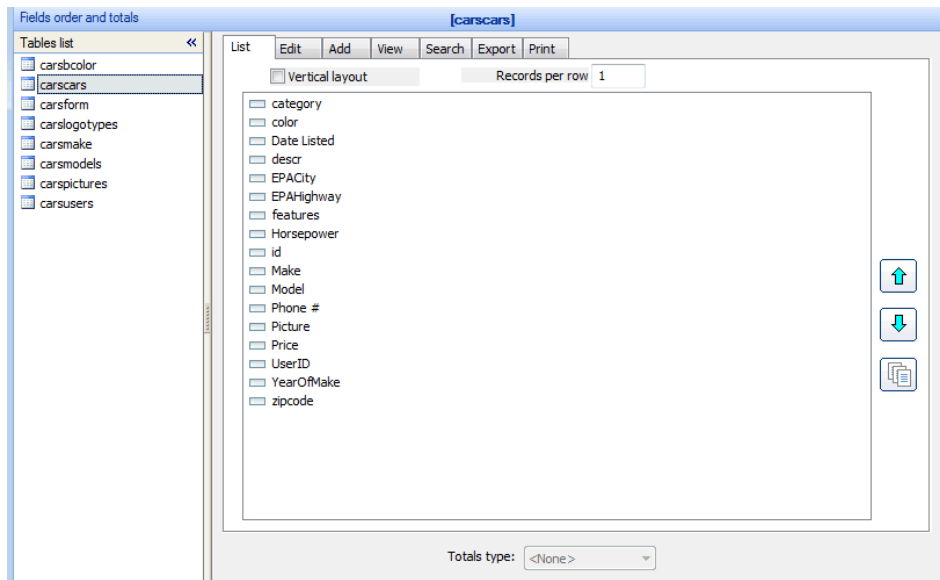


Figura 18 – Totales PHPRunner

Aquí podemos especificar el orden en el que se mostrarán los campos seleccionados en la pantalla “fields” para cada una de las páginas correspondientes a cada tabla, pudiendo añadir también secciones adicionales si lo consideramos necesario.

## Misc

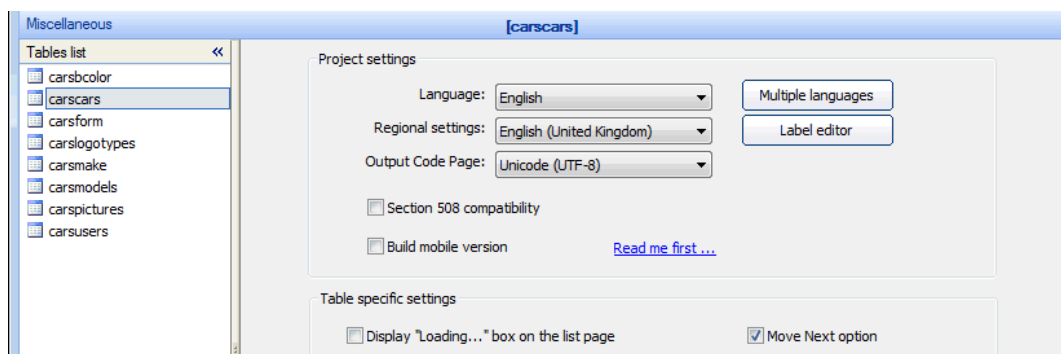


Figura 19 – Miscelánea PHPRunner

En esta pantalla podemos especificar las opciones generales para la web como el idioma, la codificación de salida, incluso si queremos que se pueda adaptar una versión para móvil. Además de especificaciones propias para cada una de las tablas de forma global o de forma única.

## Security

The screenshot displays the Security configuration page for the 'carscars' application. On the left, a 'Tables list' sidebar shows various database tables. The main configuration area is titled 'Security [carscars]' and features four radio button options: 'No Login', 'Hardcoded', 'Database' (which is selected), and 'Active Directory'. Under the 'Database' option, there are several dropdown menus: 'Table' set to 'carsusers', 'Username field (login)' set to 'username', 'Password field' set to 'password', and 'Full name field' set to 'username'. A checkbox labeled 'Add Login with Facebook option' is checked. Below this, there are input fields for 'Facebook APP ID' (containing '181638395212821') and 'Facebook Secret ID' (masked with asterisks). On the right side of the configuration area, there are five blue buttons: 'User login settings...', 'Advanced...', 'Permissions...', 'Locking and Audit...', and 'Encryption...'.

Figura 19 – Seguridad PHPRunner

En la pantalla de seguridad podemos especificar las condiciones propias a la seguridad de nuestra web, si requiere o no de un registro, la encriptación de directorios y demás. Si queremos un sistema de autenticación, deberemos disponer en nuestra base de datos una tabla dedicada a tal fin, donde los usuarios puedan introducir su nombre y su contraseña para poder comprobarlos cuando quieran entrar a la aplicación. El programa permite además registro a través de la red social Facebook. También hay que señalar que el programa permite encriptación de la información y poder hacer distintos grupos de privilegios entre los usuarios, pudiendo así especificar que usuarios tienen permisos para ver o modificar los datos de la base de datos.

## Style

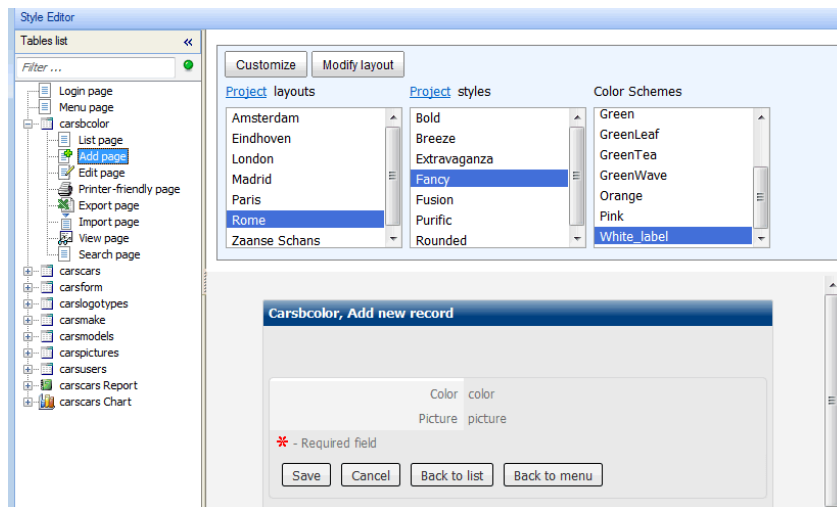


Figura 20 – Estilo PHPRunner

En la pantalla de estilo podemos seleccionar el aspecto que tendrá nuestra página web, tanto el colorido como la forma de los botones, las tablas y demás por medio de una serie de plantillas y colores definidos que se pueden combinar a nuestro gusto. Además se puede modificar tanto de forma visual como a través de código PHP.

## Editor

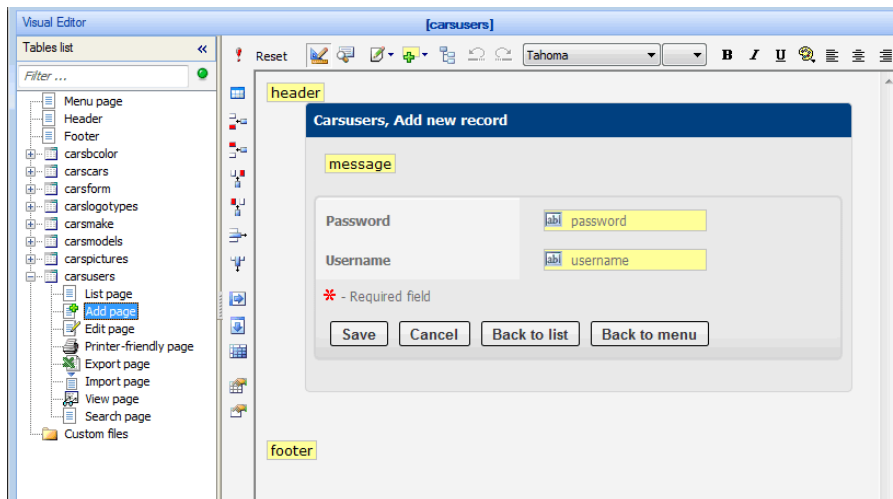


Figura 21 – Editor PHPRunner

En la pantalla de edición se puede configurar el aspecto visual más detalladamente, aquí podemos movernos con la lista que tenemos a la izquierda a través de todas las

páginas que se mostrarán en nuestra web, y se puede especificar el aspecto visual de cada una, así como la información que vemos en las mismas, pudiendo así modificar la posición de todos los elementos y tanto añadir como eliminar los que consideremos oportunos.

Como vemos en la figura 21, en la parte de la derecha de la pantalla vemos que tenemos una barra de herramientas con muchas más opciones que en la pantalla de estilo, pudiendo modificar, entre otras cosas, los tipos de letra. En la barra de herramientas lateral también tenemos más opciones para poder añadir más elementos como tablas y poder especificar la posición de los campos en cada una de ellas. En esta pantalla también tenemos la opción de trabajar directamente con el código HTML que se va generando automáticamente con forme vamos dándole forma a nuestra web, con lo que, si las herramientas de PHPRunner no fueran suficientes, podemos escribir código HTML para darle la forma final a cada una de las páginas de nuestra web.

## Events

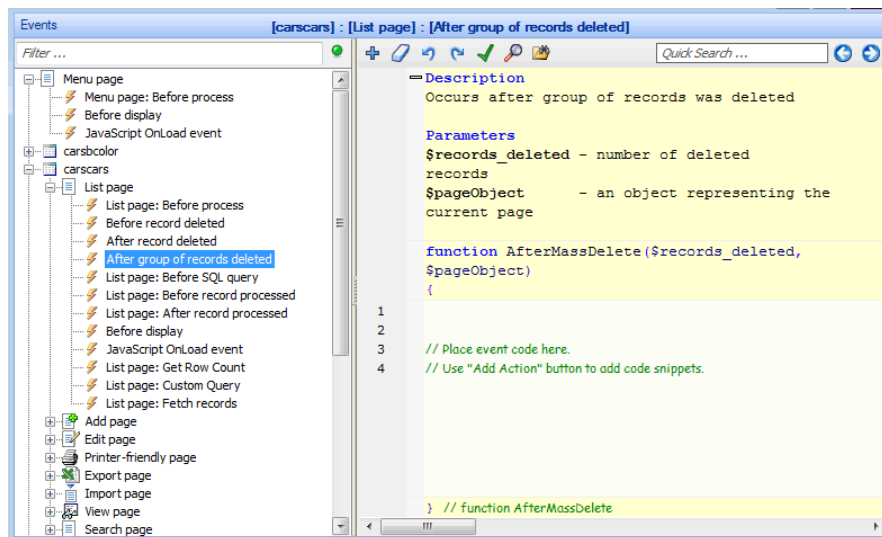


Figura 21 – Eventos PHPRunner

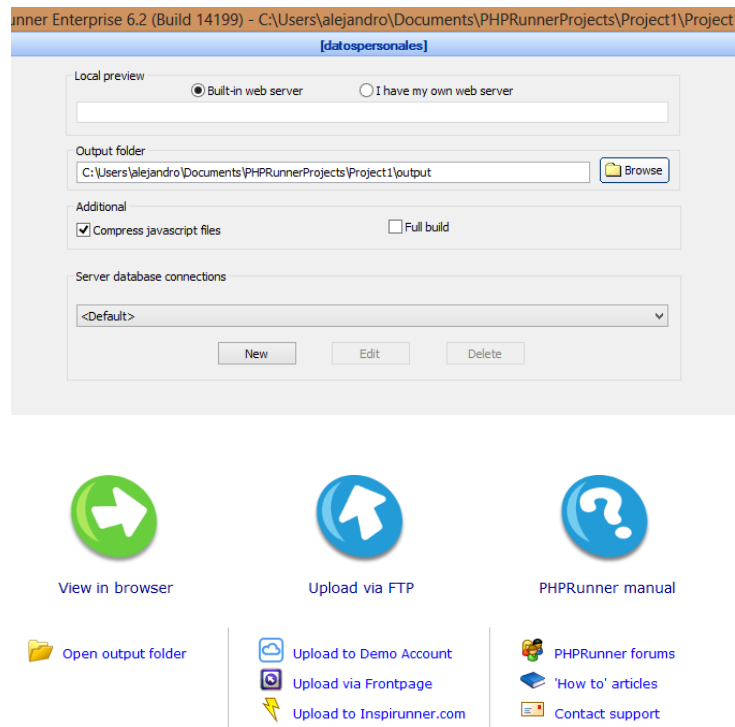
Los eventos son los fragmentos de código PHP ejecutados antes o después de que se produzcan determinadas acciones como editar, eliminar, que un nuevo usuario se registre en el sistema, etcétera. Por lo tanto, gracias a los eventos se pueden definir algunas acciones que se activarán cuando se cumplen ciertas condiciones.

Algunos ejemplos de los eventos más comunes son:

- Enviar un correo electrónico con los datos de una modificación.
- Guardar datos en otra tabla.

- Comprobar la unicidad de un registro.
- Mostrar información relacionada con el registro actual.

## Output



Por último tenemos la pantalla de “salida” del proyecto. Aquí podemos especificar si queremos subir el proyecto a un servidor web específico o por el contrario, guardarlo de manera local. Además, en esta pantalla nos aparecerá la opción de construir la web, que el programa lo hará en un servidor web o de manera local, según lo hayamos especificado. Una vez construida la web, el sistema te da la opción de ejecutar la web creada desde un explorador que tengamos instalado en el sistema y así podemos ver el resultado final de la misma o bien subirlo a un servidor web mediante FTP.

Cuando construimos un proyecto, el programa genera automáticamente ficheros HTML y hojas de estilo conforme hayamos indicado a lo largo de todo el proyecto, de forma que todo el código se pueda interpretar correctamente desde cualquier navegador web.

# CAPÍTULO 3

## 3. Desarrollo de la aplicación web

Una vez que conozcamos todas las herramientas y lenguajes de programación necesarios para nuestro proyecto, deberemos empezar con la elaboración del mismo. Lo primero que necesitamos es definir las tablas que nos harán falta. A la hora de decidir qué tablas son las que necesitamos para nuestro proyecto se ha cogido un currículum genérico. A partir de éste hemos podido definir la cantidad de tablas necesarias, así como el número y los tipos de entrada de cada una de ellas.

### 3.1. Diseño de la base de datos

Tras decidir las tablas que necesitamos, debemos proceder con su creación. Para ello tenemos dos opciones, como hemos visto anteriormente, ya que podemos crearlas directamente desde la aplicación web de XAMPP o desde el propio PHPRunner. Cada uno tiene sus ventajas y desventajas, así pues lo que se ha decidido es utilizar cada uno según la tabla que vayamos a crear, ya que al hacerlas desde la aplicación PHPRunner, nos ahorramos tener que ir sincronizando la aplicación con la base de datos y a su vez es más ágil porque no se necesitan definir tantos como nos pide la aplicación XAMPP. Sin embargo, algunas de nuestras tablas son casi idénticas, con modificaciones mínimas, pero desde PHPRunner no tenemos la opción de copiar los campos de una tabla a otra, por lo que si estamos hablando de una cantidad relativamente grande de campos el trabajo se hace muy lento e iterativo. Con lo cual, cuando queramos hacer una tabla que es casi idéntica a otra ya creada nos iremos a la aplicación XAMPP, que no nos da la opción de copiar tablas como tal, pero sí que tenemos la posibilidad de construir sentencias SQL para modificarlas, y a través de esto sí que podemos crear una nueva tabla copiando los campos de una tabla existente. Concretamente, los pasos a seguir para llevar a cabo esta acción serán:

- Una vez arrancado el servidor Apache y el SGBD MySQL vamos a nuestro navegador web y escribimos “localhost”. Una vez dentro de la aplicación XAMPP vamos a la herramienta PhpMyAdmin. A continuación seleccionamos la base de datos correspondiente a nuestro proyecto y en el menú de herramientas de la parte derecha de la pantalla entramos en SQL, donde se podrán escribir sentencias. Por último escribiremos la siguiente sentencia SQL:

```
- CREATE TABLE tablaNueva AS (  
- SELECT *  
- FROM tablaExistente  
- WHERE 1 = 0) ;
```

- Le damos a continuar y ya se nos habrá creado una tabla con el mismo número y la misma forma de campos que teníamos en una tabla existente.

Para este proyecto se necesitarán un total de 27 tablas. Para entender mejor el proyecto, separaremos estas tablas en cuatro tipos distintos de tablas.

El primer tipo de tabla sería la asociada al registro de los usuarios, dónde se almacenarán todos ellos. Sólo hay una tabla de este tipo y se le ha llamado “TablaLoguin”.

Después tendremos las distintas tablas necesarias para cumplimentar todo el curriculum, este es el tipo más numeroso.

El tercer tipo sería una tabla donde se haga referencia a cada una de las tablas del grupo anterior, de manera que cada usuario pueda seleccionar cada una de las tablas que desea ver o imprimir desde la web a esta tabla se le ha llamado “TablaEntradas”.

Por último tendremos un tipo de tabla que tendrá información útil para todos los usuarios, estas tablas ya vendrán con una cantidad importante de entradas, de manera que el usuario puede utilizarlas para rellenar información relativa a su curriculum, por ejemplo la titulación cursada o la escuela de formación dónde realizó dicha titulación. Este último tipo de tablas tienen una particularidad a la hora de introducirlas en el proyecto, ya que debemos instanciar una serie de entradas con los valores deseados, más adelante también se explicara este procedimiento.

Cada una de las tablas se verá con más detalle a continuación:

- **TablaLoguin**

En esta tabla se almacenarán los usuarios registrados en la aplicación y sus respectivas contraseñas. Los campos son:

- **IdLog** [int(11)]: tendrá autoincremento, el hecho de que tenga un tamaño de 11 es por defecto de la aplicación PHPRunner para todos los campos de tipo

entero, si quisiéramos modificarlo tendríamos que ir a PhpMyAdmin, en la aplicación web de XAMPP. El valor será de -2147483648 a 2147483647 si tuviera signo o de 0 a 4294967295 en caso de no tener signo. Este campo almacenará el identificador único asignado a cada usuario.

- **Usuario** [varchar(50)]: Tendrá un tamaño por defecto de 50 caracteres. Este campo contendrá el nombre de usuario único para cada persona registrada.
- **Contraseña** [varchar(10)]: Tendrá un tamaño por defecto de 10 caracteres y almacenará la contraseña de cada usuario.
- **Email** [varchar(60)]: Tendrá un tamaño por defecto de 60 caracteres. Este campo contendrá la dirección de e-mail de cada usuario registrado.

- **Datos Personales**

En esta tabla se almacenarán los datos personales de los usuarios del sistema. Los campos son:

- **IdDatos** [int(11)]: tendrá autoincremento. Este campo almacenará el identificador único asignado a cada entrada de datos personales de la tabla.
- **Nombre** [varchar(50)]: Tendrá un tamaño por defecto de 50 caracteres. En este campo el usuario introducirá su nombre.
- **Apellidos** [varchar(50)]: Tendrá un tamaño por defecto de 50 caracteres. En este campo el usuario introducirá sus apellidos.
- **Nacimiento** [date]: Este es un tipo de campo de fecha. Aquí cada usuario podrá introducir su fecha de nacimiento.
- **PaisNacimiento** [varchar(3)]: Este campo es de tipo carácter y tiene un tamaño de 3 caracteres. Aquí cada usuario podrá introducir su país de nacimiento. Es de tipo carácter de tamaño 3 porque está enlazado a un campo de otra tabla que tiene el mismo formato, se explicara más adelante el motivo.
- **DNI** [int(11)]: Este campo es un entero. Aquí cada usuario podrá introducir su Documento Nacional de Identidad.
- **Correo** [varchar(50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí cada usuario podrá introducir su correo electrónico.



- **Telefono** [varchar(15)]: Tendrá un tamaño por defecto de 15 caracteres. Aquí cada usuario podrá introducir su teléfono. Se ha decidido ponerlo en forma de caracteres para que el propio usuario pueda escribirlo con el formato que lo considere oportuno (separado por barras, guiones, espacios o simplemente todo junto).
- **ID** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de cada usuario asignada en el momento del registro.

- **Country**

En esta tabla se almacenarán el nombre de los países para que los usuarios puedan seleccionar su lugar de nacimiento, por lo tanto esta sería una tabla del cuarto tipo. Esta tabla en concreto además se pudo encontrar fácilmente en internet en formato SQL, por lo que solo hubo que añadirla a nuestra base de datos sin más. Los campos son:

- **Code** [varchar(3)]: será el identificador único de cada país. Este campo almacenará el identificador único asignado a cada entrada de la tabla.
- **Name** [varchar(52)]: Aquí tendremos el nombre de cada país.

- **Titulaciones**

En esta tabla se almacenarán las titulaciones de los usuarios del sistema. Los campos son:

- **IdTitulacion** [int(11)]: tendrá autoincremento. Este campo almacenará el identificador único asignado a cada entrada de la tabla.
- **Titulacion** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de una titulación asignada por el usuario.
- **CentroFormacion** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de un centro de formación asignado por el usuario.
- **ID** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de cada usuario asignada en el momento del registro.

- **TablaTitulos**

En esta tabla se almacenarán el nombre las titulaciones para que los usuarios las puedan seleccionar, por lo tanto esta sería una tabla del cuarto tipo. Los campos son:

- **IdTitulo** [int(11)]: tendrá autoincremento. Este campo almacenará el identificador único asignado a cada entrada de la tabla.
- **NombreTitulo** [varchar(50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí vendrá el nombre de la titulación.
- **CentroFormativo** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación de un centro de formación y estará asociada a otra tabla de la base de datos.

- **Masters**

En esta tabla se almacenarán los títulos de Master de los usuarios del sistema. Los campos son:

- **IdMaster** [int(11)]: tendrá autoincremento. Este campo almacenará el identificador único asignado a cada entrada de la tabla.
- **Master** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de un Master asignada por el usuario.
- **CentroFormacion** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de un centro de formación asignado por el usuario.
- **ID** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de cada usuario asignada en el momento del registro.

- **TablaMasters**

En esta tabla se almacenarán el nombre los Masters para que los usuarios los puedan seleccionar, por lo tanto esta sería una tabla del cuarto tipo. Los campos son:

- **IdMaster** [int(11)]: tendrá autoincremento. Este campo almacenará el identificador único asignado a cada entrada de la tabla.
- **NombreMaster** [varchar(50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí vendrá el nombre del Master.
- **CentroFormativo** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación de un centro de formación y estará asociada a otra tabla de la base de datos.

- **Doctorados**

En esta tabla se almacenarán los Doctorados de los usuarios del sistema. Los campos son:

- **IdDoctorado** [int(11)]: tendrá autoincremento. Este campo almacenará el identificador único asignado a cada entrada de la tabla.
- **Doctorado** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de un Doctorado asignado por el usuario.
- **CentroFormacion** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de un centro de formación asignado por el usuario.
- **ID** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de cada usuario asignada en el momento del registro.

- **TablaDoctorados**

En esta tabla se almacenarán el nombre los Doctorados para que los usuarios los puedan seleccionar, por lo tanto esta sería una tabla del cuarto tipo. Los campos son:

- **IdDoctorado** [int(11)]: tendrá autoincremento. Este campo almacenará el identificador único asignado a cada entrada de la tabla.
- **NombreDoctorado** [varchar(50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí vendrá el nombre del Doctorado.

- **CentroFormativo** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación de un centro de formación y estará asociada a otra tabla de la base de datos.

- **CentrosDeFormacion**

En esta tabla se almacenarán el nombre de una serie de centros de formación, principalmente universidades, para que los usuarios los puedan seleccionarlos, por lo tanto esta sería una tabla del cuarto tipo.

- **IdUni** [int(11)]: será el identificador único de cada centro de formación. Este campo almacenará el identificador único asignado a cada entrada de la tabla.
- **NombreCentro** [varchar(50)]: Aquí tendremos el nombre de cada Centro de formación..

- **Publii y Publin**

Estas dos tablas serán idénticas en cuanto al número y a la forma de los campos, aquí se podrán guardar las publicaciones en artículos y revistas de cada usuario. El hecho de que sean dos tablas es porque una tendrá las publicaciones dentro del ámbito internacional (Publii) y la otra contendrá las publicaciones dentro del ámbito nacional (Publin).

- **IdPubli** [int(11)]: tendrá autoincremento. Este campo almacenará el identificador único asignado a cada entrada de la tabla.
- **Libro\_revista** [varchar(50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí vendrá el nombre del libro o la revista donde se publicó.
- **Paginas** [int(11)]: Este campo es un entero. Aquí se podrá indicar el número de página donde comienza la publicación en dicho libro o revista.
- **Impacto** [varchar(50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí se podrá indicar si se trata de una publicación con impacto o sin impacto.
- **Fecha** [date]: Tendrá un formato de tipo fecha. Aquí vendrá la fecha en la que se hizo la publicación.

- **ISBN** [varchar(50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí se podrá indicar el ISBN de la publicación.
- **ISSN** [varchar(50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí se podrá indicar el ISSN de la publicación.
- **Editorial** [varchar(50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí se podrá indicar la editorial de la revista o el libro donde se hizo pública.
- **ID** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de cada usuario asignada en el momento del registro.

- **Proyi y Proyn**

Estas dos tablas serán idénticas en cuanto al número y a la forma de los campos, aquí se podrán guardar los proyectos de cada usuario. El hecho de que sean dos tablas es porque una tendrá los proyectos dentro del ámbito internacional (Proyi) y la otra contendrá los proyectos dentro del ámbito nacional (Proyn).

- **IdProy** [int(11)]: tendrá autoincremento. Este campo almacenará el identificador único asignado a cada entrada de la tabla.
- **Título** [varchar(160)]: Tendrá un tamaño por defecto de 160 caracteres. Aquí vendrá el título del proyecto realizado.
- **EntidadFinanciadora** [varchar (160)]: Tendrá un tamaño por defecto de 160 caracteres. Aquí vendrá el nombre de la entidad financiadora del proyecto.
- **EntidadParticipante** [varchar (160)]: Tendrá un tamaño por defecto de 160 caracteres. Aquí vendrá el nombre de la entidad participante en el proyecto.
- **FechaInicio** [date]: Tendrá un formato de tipo fecha. Aquí vendrá la fecha en la que se inició el proyecto.
- **Duración** [varchar(50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí se podrá indicar la duración del proyecto.

- **InvestigadorPrincipal** [varchar(80)]: Tendrá un tamaño por defecto de 80 caracteres. Aquí se podrá indicar el nombre del investigador principal del proyecto.
- **NumParticipantes** [int(11)]: Este campo es un entero. Aquí se podrá indicar el número de participantes en el proyecto.
- **Subvencion** [int(11)]: Este campo es un entero. Aquí se podrá indicar la cantidad de subvención que hubo en el proyecto.
- **Otro** [varchar(300)]: Tendrá un tamaño por defecto de 300 caracteres. Aquí se podrá indicar información adicional que se necesite por parte del usuario.
- **ID** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de cada usuario asignada en el momento del registro.

- **Contratoi y Contraton**

Estas dos tablas serán idénticas en cuanto al número y a la forma de los campos, aquí se podrán guardar los contratos de cada usuario. El hecho de que sean dos tablas es porque una tendrá los contratos realizados dentro del ámbito internacional (Contratoi) y la otra contendrá los contratos dentro del ámbito nacional (Contraton).

- **IdContrato** [int(11)]: tendrá autoincremento. Este campo almacenará el identificador único asignado a cada entrada de la tabla.
- **Titulo** [varchar(260)]: Tendrá un tamaño por defecto de 260 caracteres. Aquí vendrá el título del contrato realizado.
- **EntidadFinanciadora** [varchar (160)]: Tendrá un tamaño por defecto de 160 caracteres. Aquí vendrá el nombre de la entidad financiadora del contrato.
- **EntidadParticipante** [varchar (160)]: Tendrá un tamaño por defecto de 160 caracteres. Aquí vendrá el nombre de la entidad participante en el contrato.
- **FechaInicio** [date]: Tendrá un formato de tipo fecha. Aquí vendrá la fecha en la que se inició el contrato.
- **Duración** [varchar(50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí se podrá indicar la duración del contrato.

- **InvestigadorPrincipal** [varchar(80)]: Tendrá un tamaño por defecto de 80 caracteres. Aquí se podrá indicar el nombre del investigador principal del contrato.
- **NumParticipantes** [int(11)]: Este campo es un entero. Aquí se podrá indicar el número de participantes en el contrato.
- **Subvencion** [int(11)]: Este campo es un entero. Aquí se podrá indicar la cantidad de subvención que hubo en el contrato.
- **TipoContrato** [varchar(150)]: Tendrá un tamaño por defecto de 150 caracteres. Aquí se podrá indicar el tipo de contrato.
- **Ref** [varchar(50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí se podrá indicar la referencia del contrato.
- **Otro** [varchar(300)]: Tendrá un tamaño por defecto de 300 caracteres. Aquí se podrá indicar información adicional que se necesite por parte del usuario.
- **ID** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de cada usuario asignada en el momento del registro.

- **ContribucionesCongi y ContribucionesCongn**

Estas dos tablas serán idénticas en cuanto al número y a la forma de los campos, aquí se podrán guardar las contribuciones a congresos de cada usuario. El hecho de que sean dos tablas es porque una tendrá los congresos realizados dentro del ámbito internacional (ContribucionesCongi) y la otra contendrá los congresos dentro del ámbito nacional (ContribucionesCongn).

- **IdCont** [int(11)]: tendrá autoincremento. Este campo almacenará el identificador único asignado a cada entrada de la tabla.
- **Participantes** [varchar(200)]: Tendrá un tamaño por defecto de 200 caracteres. Aquí vendrán los nombres de los participantes en el congreso.
- **Congreso** [varchar (150)]: Tendrá un tamaño por defecto de 150 caracteres. Aquí vendrá el nombre del congreso.
- **Lugar** [varchar (150)]: Tendrá un tamaño por defecto de 150 caracteres. Aquí vendrá el nombre del lugar donde se realizó el congreso.

- **Fecha** [date]: Tendrá un formato de tipo fecha. Aquí vendrá la fecha en la que se llevó a cabo el congreso.
- **Otro** [varchar(300)]: Tendrá un tamaño por defecto de 300 caracteres. Aquí se podrá indicar información adicional que se necesite por parte del usuario.
- **ID** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de cada usuario asignada en el momento del registro.

- **TesisDoctorales**

En esta tabla se almacenarán las tesis doctorales en las que han participado los usuarios del sistema. Los campos son:

- **IdTesis** [int(11)]: tendrá autoincremento. Este campo almacenará el identificador único asignado a cada entrada de la tabla.
- **Título** [varchar (150)]: Tendrá un tamaño por defecto de 150 caracteres. Aquí vendrá el título de la tesis doctoral.
- **Doctorando** [varchar (50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí vendrá el nombre de la persona que se está doctorando.
- **Universidad** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de un centro de formación asignado por el usuario.
- **Facultad** [varchar (50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí vendrá el nombre de la facultad a la que pertenece el doctorado.
- **Fecha** [date]: Tendrá un formato de tipo fecha. Aquí vendrá la fecha en la que se llevó a cabo el congreso.
- **Calificación** [varchar (50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí vendrá la calificación final del doctorado.
- **ID** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de cada usuario asignada en el momento del registro.



- **PatentesYmodelos**

En esta tabla se almacenarán las patentes y modelos de los usuarios del sistema. Los campos son:

- **IdPyM** [int(11)]: tendrá autoincremento. Este campo almacenará el identificador único asignado a cada entrada de la tabla.
- **Título** [varchar (150)]: Tendrá un tamaño por defecto de 150 caracteres. Aquí vendrá el título del modelo o la patente.
- **Autores** [varchar (350)]: Tendrá un tamaño por defecto de 350 caracteres. Aquí vendrá el nombre de los autores del modelo o la patente
- **NumSolicitud** [varchar (50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí vendrá el número de solicitud de la patente.
- **Fecha** [date]: Tendrá un formato de tipo fecha. Aquí vendrá la fecha en la que se presenta el modelo o la patente.
- **Titular** [varchar (50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí vendrá el nombre del titular del modelo o la patente.
- **LicenciasVendidas** [int(11)]: Este campo es un entero. Este campo tendrá la cantidad de licencias vendidas del modelo o la patente.
- **ImporteAcumulado** [varchar (50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí vendrá la cantidad de dinero acumulado gracias al modelo o la patente.
- **ID** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de cada usuario asignada en el momento del registro.

- **Becas**

En esta tabla se almacenarán las Becas concedidas a los usuarios del sistema. Los campos son:

- **IdBeca** [int(11)]: tendrá autoincremento. Este campo almacenará el identificador único asignado a cada entrada de la tabla.
- **Tipo** [varchar (50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí vendrá el tipo de Beca concedida.

- **ConcursoPublico** [varchar (50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí se indicará si la beca fue de concurso público o no.
- **Duracion** [varchar (50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí vendrá la duración de la Beca.
- **Fecha** [date]: Tendrá un formato de tipo fecha. Aquí vendrá la fecha en la que se concede la Beca.
- **InstitucionFinanciera** [varchar (150)]: Tendrá un tamaño por defecto de 150 caracteres. Aquí vendrá el nombre de la institución que concede la Beca.
- **Universidad** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de un centro de formación asignado por el usuario.
- **ID** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de cada usuario asignada en el momento del registro.

- **Ayudas**

En esta tabla se almacenarán las ayudas destinadas a proyectos concedidas a los usuarios del sistema. Los campos son:

- **IdAyuda** [int(11)]: tendrá autoincremento. Este campo almacenará el identificador único asignado a cada entrada de la tabla.
- **Título** [varchar (150)]: Tendrá un tamaño por defecto de 150 caracteres. Aquí vendrá el título del proyecto para el que se recibió ayuda.
- **Financiacion** [varchar (150)]: Tendrá un tamaño por defecto de 150 caracteres. El nombre del programa de financiación.
- **Entidad** [varchar (150)]: Tendrá un tamaño por defecto de 150 caracteres. Aquí vendrá la entidad a la que fue destinada la ayuda.
- **Fecha** [date]: Tendrá un formato de tipo fecha. Aquí vendrá la fecha en la que se concede la Ayuda.

- **InvestigadorPrincipal** [varchar (50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí se podrá indicar el nombre del investigador principal del contrato.
- **Cantidad** [varchar (50)]: Tendrá un tamaño por defecto de 50 caracteres. Este campo tendrá almacenada la cantidad correspondiente a la ayuda obtenida.
- **ID** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de cada usuario asignada en el momento del registro.

- **InformesDeInvestigación**

En esta tabla se almacenarán los informes de investigación de los usuarios del sistema. Los campos son:

- **IdInformes** [int(11)]: tendrá autoincremento. Este campo almacenará el identificador único asignado a cada entrada de la tabla.
- **Autores** [varchar (350)]: Tendrá un tamaño por defecto de 350 caracteres. Aquí vendrá el nombre de los participantes de la investigación.
- **Titulo** [varchar (150)]: Tendrá un tamaño por defecto de 150 caracteres. El nombre del informe de investigación.
- **Referencia** [varchar (150)]: Tendrá un tamaño por defecto de 150 caracteres. Aquí vendrá la referencia del informe de investigación.
- **Fecha** [date]: Tendrá un formato de tipo fecha. Aquí vendrá la fecha en la que se concede la Ayuda.
- **ID** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de cada usuario asignada en el momento del registro.

- **Ayudas**

En esta tabla se almacenarán las ayudas destinadas a proyectos concedidas a los usuarios del sistema. Los campos son:

- **IdAyuda** [int(11)]: tendrá autoincremento. Este campo almacenará el identificador único asignado a cada entrada de la tabla.

- **Título** [varchar (150)]: Tendrá un tamaño por defecto de 150 caracteres. Aquí vendrá el título del proyecto para el que se recibió ayuda.
- **Financiaci3n** [varchar (150)]: Tendrá un tamaño por defecto de 150 caracteres. El nombre del programa de financiaci3n.
- **Entidad** [varchar (150)]: Tendrá un tamaño por defecto de 150 caracteres. Aquí vendrá la entidad a la que fue destinada la ayuda.
- **Fecha** [date]: Tendrá un formato de tipo fecha. Aquí vendrá la fecha en la que se concede la Ayuda.
- **InvestigadorPrincipal** [varchar (50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí se podrá indicar el nombre del investigador principal del contrato.
- **Cantidad** [varchar (50)]: Tendrá un tamaño por defecto de 50 caracteres. Este campo tendrá almacenada la cantidad correspondiente a la ayuda obtenida.
- **ID** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificaci3n 3nica de cada usuario asignada en el momento del registro.

- **Docencia**

En esta tabla se almacenará la actividad docente de los usuarios del sistema. Los campos son:

- **IdDocencia** [int(11)]: tendrá autoincremento. Este campo almacenará el identificador 3nico asignado a cada entrada de la tabla.
- **Asignatura** [varchar(150)]: Tendrá un tamaño por defecto de 150 caracteres. Aquí vendrá el nombre de la asignatura.
- **Centro** [int(11)]: Tendrá un tamaño por defecto de 11 caracteres. Referencia al centro de formaci3n donde se impartió la asignatura.
- **CicloUniversitario** [varchar(50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí vendrá el ciclo formativo en el que se impartió la asignatura.

- **Periodo** [varchar(50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí vendrá el periodo durante el cual se impartió la asignatura.
- **ProfesorResponsable** [varchar(50)]: Tendrá un tamaño por defecto de 50 caracteres. Aquí se indicará si el usuario fue o no el profesor responsable de la asignatura impartida.
- **Programa** [varchar(450)]: Tendrá un tamaño por defecto de 450 caracteres. Aquí se indicará si el programa de la asignatura impartida.
- **ID** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de cada usuario asignada en el momento del registro.

- **Otros**

En esta tabla se almacenará información adicional que quieran incluir los usuarios del sistema. Los campos son:

- **IdOtro** [int(11)]: tendrá autoincremento. Este campo almacenará el identificador único asignado a cada entrada de la tabla.
- **Descripcion** [varchar(550)]: Tendrá un tamaño por defecto de 550 caracteres. Aquí vendrá toda la información que necesite añadir el usuario.
- **ID** [int(11)]: Este campo es un entero. Este campo tendrá almacenada la identificación única de cada usuario asignada en el momento del registro.

- **TablaEntradas**

En esta tabla será la correspondiente al tercer tipo, aquí todos los campos serán de tipo entero [int(11)] y habrá un campo ID con el identificador único de cada usuario y luego una campo por cada una de las tablas del segundo tipo, es decir, las tablas orientadas a recopilar toda la información del CV de cada usuario.

### 3.2. Relación entre tablas

*En cuanto a las relaciones que se han establecido entre las tablas anteriores hay que decir que existe un campo ID en todas las tablas que corresponde al identificador único de cada usuario, este identificador se genera automáticamente cuando un usuario*

nuevo se registra en el sistema, por la propiedad de autoincremento y es la clave primaria de la tabla *TablaLoguin*.

Además de esta relación están las existentes entre la tabla de entrada de datos y el resto de tablas de información del sistema. Esta relación se consigue asignando un campo por cada una de las tablas de tipo entero y luego se relaciona cada uno de los campos con el identificador único de su tabla correspondiente.

Otra relación es la que existe entre la tabla de datos personales y la tabla de los países, en este caso el campo *PaisDeNacimiento* de la tabla *DatosPersonales* tiene el mismo formato que el campo *Code* de la tabla *Country*, que además es la clave primaria de la tabla, por lo que es único para cada entrada existente. De esta forma el sistema no puede tener fallos.

La siguiente relación entre tablas que vamos a ver es la existente entre las tablas correspondientes a la formación del usuario, estas son: *Titulaciones*, *Master* y *Doctorados*. Cada una de estas tablas tiene dos campos que están relacionados con otras dos tablas, uno de esos campos es común a las tres, se trata del campo *CentroFormacion*, que es un entero relacionado con el campo *IdUni* de la tabla *CentrosDeFormacion*, que además es la clave primaria de la tabla. El otro campo que está relacionado es el que corresponde al nombre del título, para cada tabla este campo es: *IdTitulacion*, *IdMaster* y *IdDoctorado* respectivamente. Cada uno de estos campos se relaciona con el correspondiente a la clave primaria de las tablas *Tablaitulos*, *TablaMaster* y *TablaDoctorados*, estos campos son todos de tipo entero.

Por último hay que señalar que también existen tres relaciones más entre las tablas: *TesisDoctorales*, *Becas* y *Docencia*. Estas tres tablas tienen un campo de tipo entero para designar una universidad o centro de formación y están relacionadas exactamente igual que en el apartado anterior con los campos de centro de formación de las tablas de formación.

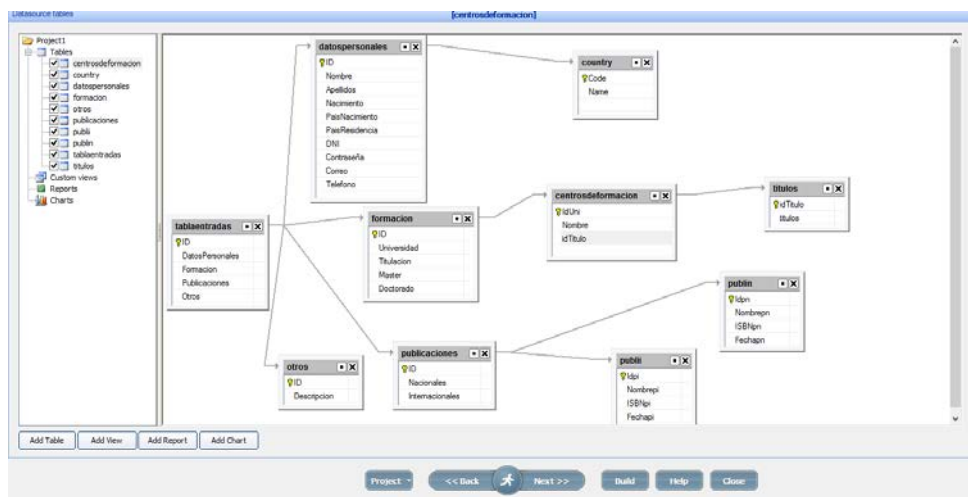


Figura 22 – Ejemplo relación entre tablas PHPRunner

### 3.3. Implementación de la interfaz web

En la parte del diseño de las tablas de la base de datos se ha utilizado tanto la aplicación XAMPP como el PHPRunner, sin embargo, para el desarrollo de la interfaz web se ha utilizado exclusivamente la aplicación PHPRunner, exprimiendo su potencial. Para explicar este apartado lo vamos a hacer “navegando” por todas las pantallas que tiene la aplicación y viendo cómo se ha trabajado en cada una.

#### - Páginas

En esta parte debemos asignar las páginas que se verán para cada una de las tablas. En esta parte de la memoria vamos a volver a aprovechar los cuatro grupos distintos de tablas que se hicieron anteriormente.

- Tipo uno (login): Esta tabla solo se utilizará para almacenar a los usuarios registrados, por lo que no tendrá ninguna página relativa a la misma.
- Tipo dos (información): para cada una de este tipo de tablas se asignará una página para listar la información, otra para editar la información, añadir un registro y una última para eliminar un registro.
- Tipo tres (tabla entradas): esta tabla no tendrá ninguna página de edición como tal, al igual que la tabla del tipo uno.
- Tipo cuatro (tablas instanciadas): de estas tablas solo queremos utilizar la información que viene ya en ellas para rellenar los campos del tipo dos, pero queremos que se puedan añadir más registros por parte del usuario, por eso para este tipo de tablas sólo vamos a necesitar la página de añadir un registro.

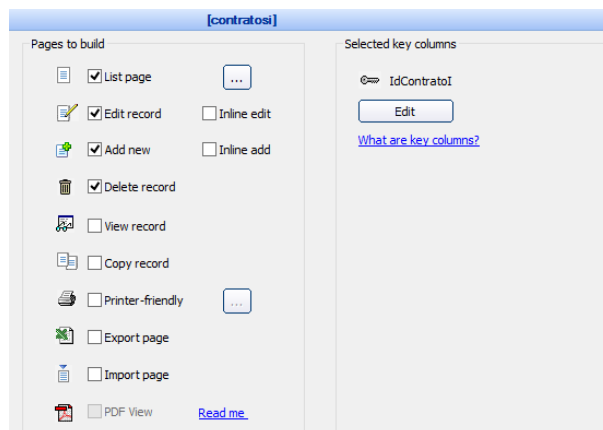


Figura 23 – Páginas de tipo dos

- Fields

En esta pantalla vamos a seleccionar los campos que queremos que se muestren en cada página, por lo tanto en las tablas del tipo uno y tres no tendremos que especificar ninguno, ya que nos saldrán todos deshabilitados.

En cuanto a los otros tipos, especificaremos que queremos mostrar todos los campos excepto el de la clave primaria de cada registro, ya que es información que no necesita el usuario y podemos hacerla transparente a él, y el campo de “ID” que tiene cada tabla para identificar a qué usuario pertenece, por la misma razón que el anterior.

Field	Label	List	Search	Add	Edit	View	Print	Export	Inline Add	Inline Edit
<input type="checkbox"/> IdContratoI	Id Contrato I	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> Titulo	Titulo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> EntidadFinanciadora	Entidad Financiadora	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> EntidadParticipante	Entidad Participante	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> FechaInicio	Fecha Inicio	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> Duracion	Duracion	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> InvestigadorPrincipal	Investigador Principal	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> NumParticipantes	Num Participantes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> Subvencion	Subvencion	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> TipoContrato	Tipo Contrato	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> Ref	Ref	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> Otro	Otro	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> ID	ID	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 24 – Campos de tipo dos

- Totals

Aquí debemos especificar el orden en el que queremos que se muestren los campos. En principio si los campos están añadidos a las tablas en el orden que los quieres mostrar, se verán de esa forma por defecto y no tendremos que modificar nada, como es nuestro caso. Sin embargo, si por lo que sea queremos añadir un campo en un futuro podemos añadirlo al final de una tabla y especificar en esta pantalla en qué posición queremos que se muestre al usuario.

Además diremos que los campos se tienen que mostrar en vertical en la página de “listado” de los campos, para una visualización más óptima de la información.

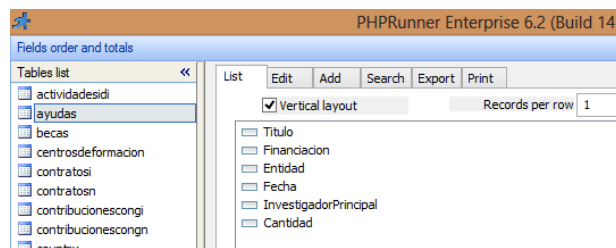


Figura 24 – Totals de tipo dos



- Miscelánea

Aquí seleccionaremos algunas especificaciones globales de la web, por ejemplo el idioma, que lo tendremos en español, la aplicación acepta múltiples lenguajes, pero para nuestros propósitos tampoco es necesario.

Después podemos seleccionar algunos ajustes que tienen que ver con la forma en la que se muestran las tablas, aquí diremos que se muestren 10 resultados por página, ya que por defecto nos salen 20 y lo normal es que tengamos menos resultados, no obstante especificaremos que el usuario pueda cambiar el número de resultados visibles por página por si le conviene ver más o menos

The image shows a screenshot of a web application's settings interface. It is divided into three main sections: Project settings, Table specific settings, and Enterprise Edition settings. In the Project settings section, 'Language' is set to 'Spanish', 'Regional settings' to 'Español (Español, tradicional)', and 'Output Code Page' to 'Unicode (UTF-8)'. There are buttons for 'Multiple languages' and 'Label editor'. In the Table specific settings section, 'Move Next option' is checked, 'Rollover row highlighting' is checked, and 'Truncate large text fields' is checked. The 'Records per page' is set to 10, and the 'Records per page selection' dropdown is set to '5, 10, 20, all'. There is a 'Search settings...' button and an 'Apply to all tables' button. In the Enterprise Edition settings section, 'Web Reports' is unchecked.

Figura 25 – Especificaciones miscelánea

- Seguridad

En cuanto a la seguridad se ha establecido que la identificación se hará a través de la base de datos, hay que especificar la tabla que tendrá este fin, en nuestro caso será *TablaLoguin*, el campo de esa misma tabla donde se almacenará el usuario, que será *Usuario* y por último el campo donde vendrá el password del usuario, que será *Contraseña*.

Además indicaremos que queremos que exista una pantalla de registro para nuevos usuarios, además de otra para poder cambiar la contraseña y otra para recordarla por correo. Para esto último debemos seleccionar un campo de la misma tabla donde tendremos almacenada la dirección de correo de cada usuario.

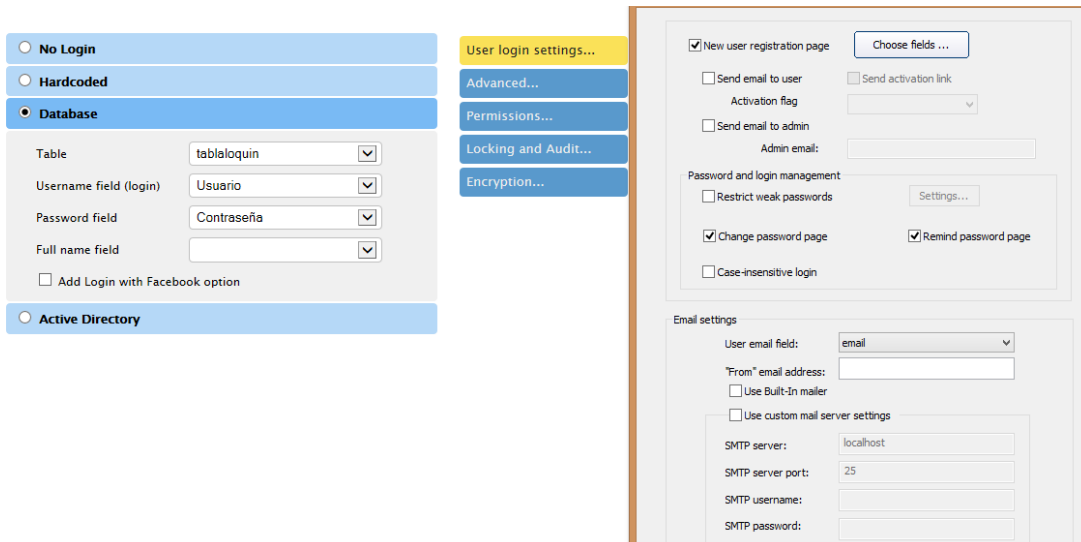


Figura 26 – Especificaciones de seguridad

- Estilo

Sobre el estilo de la web se han seleccionado una combinación de colores más bien claros con las plantillas que nos proporciona la propia aplicación. Aquí se podría especificar uno a uno el aspecto de cada elemento de las diferentes páginas de la web.

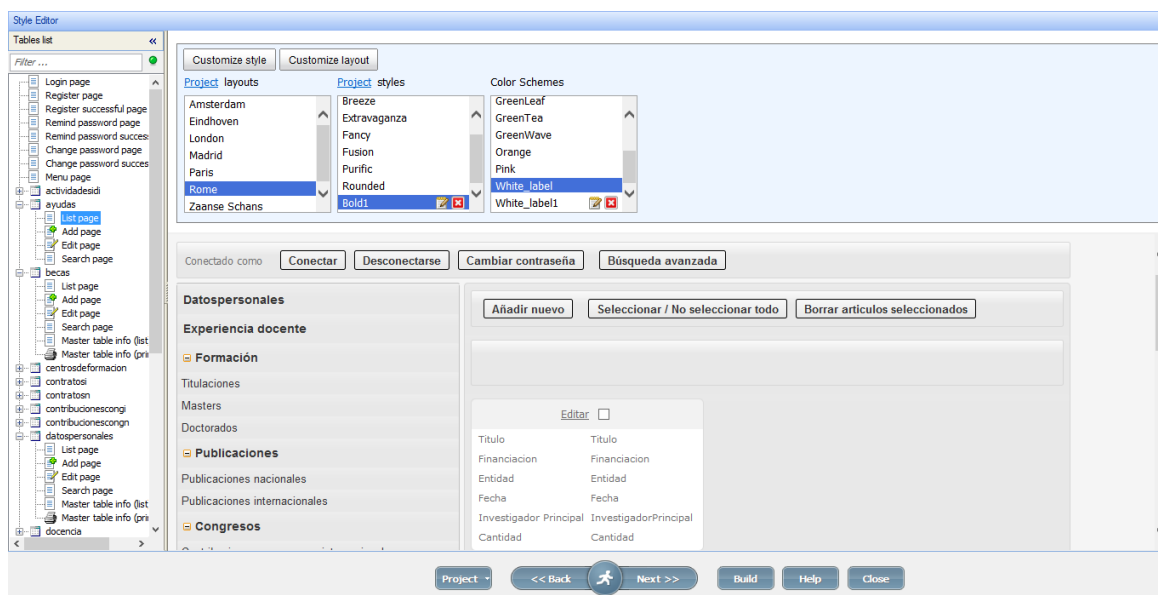


Figura 27 – Estilo de la web

Concretamente en “layouts” se ha seleccionado *Rome*, en “styles” hemos asignado *Blod1* y en “Color Schemes” se ha elegido *White\_label*.

En esta pantalla vemos que a la vez que se puede ajustar todo de manera gráfica también se genera código en segundo plano, un ejemplo es el siguiente código que se ha generado automáticamente de una de las páginas de la web:

```
23         brick color1 vrecsperpage
24     brick searchpanel
25     endcontainer
26 endblock
27
28 beginblock center
29     begincontainer vertical style=1 width=100% bigblock
30         begincontainer horizontal color=1 style=2 width=100% recordcontrols
31             brick left recordcontrols_new
32             brick left recordcontrol
33         endcontainer
34         begincontainer horizontal color=1 style=2 width=100% message
35             brick center filename=message_bare.htm message
36         endcontainer
37         begincontainer grid delimx=10 delimy=10 color=1 style=grid grid
38             brick grid
39         endcontainer
40         begincontainer horizontal color=1 style=2 width=100% pagination
41             brick center pagination
42         endcontainer
43     endcontainer
44 endblock
```

Figura 28 – Código generado por PHPRunner en la pantalla de estilo

Entendemos que el programa genera ese código para que a la hora de construir el entorno web genere las hojas de estilo (archivos .css) apropiadas según ha especificado el usuario de acuerdo a ese código.

#### - Editor

Aquí deberemos especificar aspectos más concretos de la visualización final de la web y el manejo de las tablas. Lo primero que debemos especificar es de qué forma queremos que se muestre el menú principal de navegación de la web, para que el usuario pueda moverse por las distintas páginas que la conforman.

El esquema que hemos considerado mejor para movernos por la web es el que se muestra en la siguiente figura:

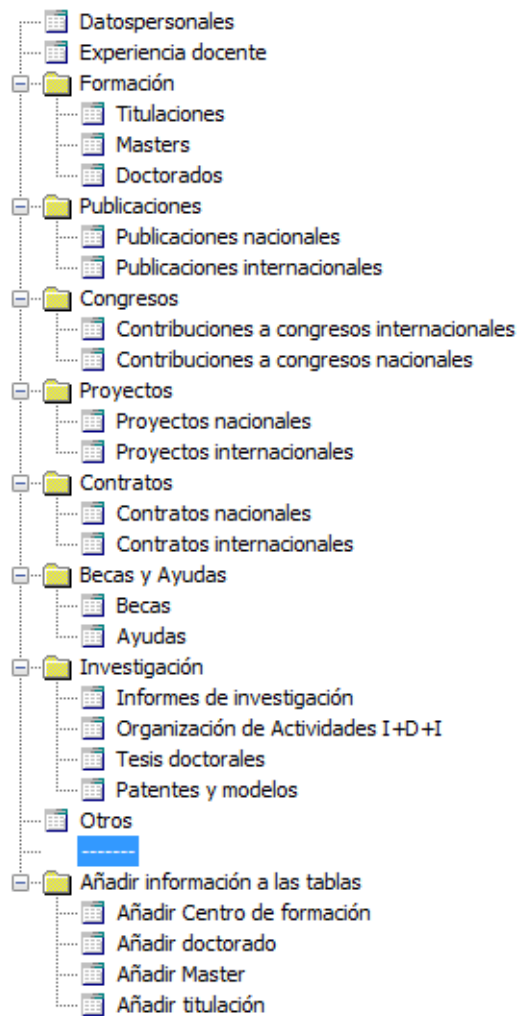


Figura 29 – Esquema del menú principal de la web

Así pues se han agrupado las tablas como vemos en la figura, de forma que el usuario despliega los grupos para ver las pantallas de las que dispone y seleccionar a cuál desea ir.

Como vemos en la figura, las últimas pantallas que aparecen son las destinadas a que el usuario pueda añadir información a las tablas de tipo cuatro, por eso en las opciones del “lincado” de esas tablas debemos asignar que tienen que enlazar con la página de añadir campos de cada una de estas tablas. El resto de tablas se pondrán por defecto en enlace a la página de listado de las tablas, así que no lo modificaremos. Una vez estemos en la página de listado de cada tabla podremos añadir entradas, modificar las existentes o eliminarlas, tal y como especificamos en la pantalla de páginas.

También debemos asignar cómo se tienen que comportar las entradas de datos de todos los campos, por defecto serán todos de tipo texto, por lo que debemos especificar cuáles son de tipo numérico o si queremos que coja alguna estructura concreta de datos, si

es un teléfono, una dirección de correo, un código postal, etcétera. Esto debemos hacerlo ya que si no lo especificamos se podrían crear inconsistencias en las tablas por no introducir los datos de manera correcta.

Además los campos de las tablas que ya tenemos instanciadas debemos seleccionar la forma en la que están enlazados con otros campos de otras tablas, ya que el programa no lo hace. Para especificar esto debemos entrar a las propiedades de la entrada del campo que queremos que dependa de estas tablas instanciadas y especificar el campo de qué tabla es el que está enlazado con la tabla.

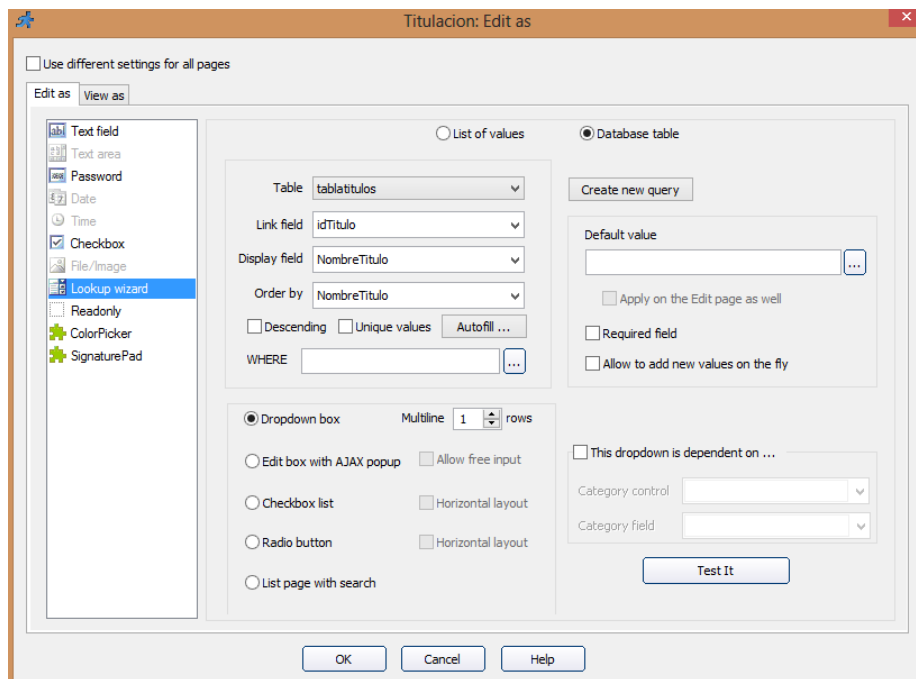


Figura 30 – Propiedades de campo que depende de otro

Como vemos en la figura 40, corresponde a la entrada de información del campo *Titulo* de la tabla *Titulacion*. Para que se pueda seleccionar información de otra tabla desde un desplegable debemos darle a la opción “Lookup wizard”, en el menú de opciones debemos seleccionar la opción “Database table” y finalmente seleccionamos que la información se encuentra en la tabla *TablTitulos*, el campo que se debe almacenar finalmente es el valor que haya en *IdDoctorado*, y la información a mostrar en el desplegable se encuentra en el campo *NombreDoctorado*. Por último se especifica que los muestre en orden alfabético y ascendente según este campo.

Finalmente, como curiosidad, cabe añadir que la aplicación PHPRunner también nos ofrece distintos formatos de entrada de los campos de tipo “Date” o fechas. Estos los podemos ver en la siguiente figura.

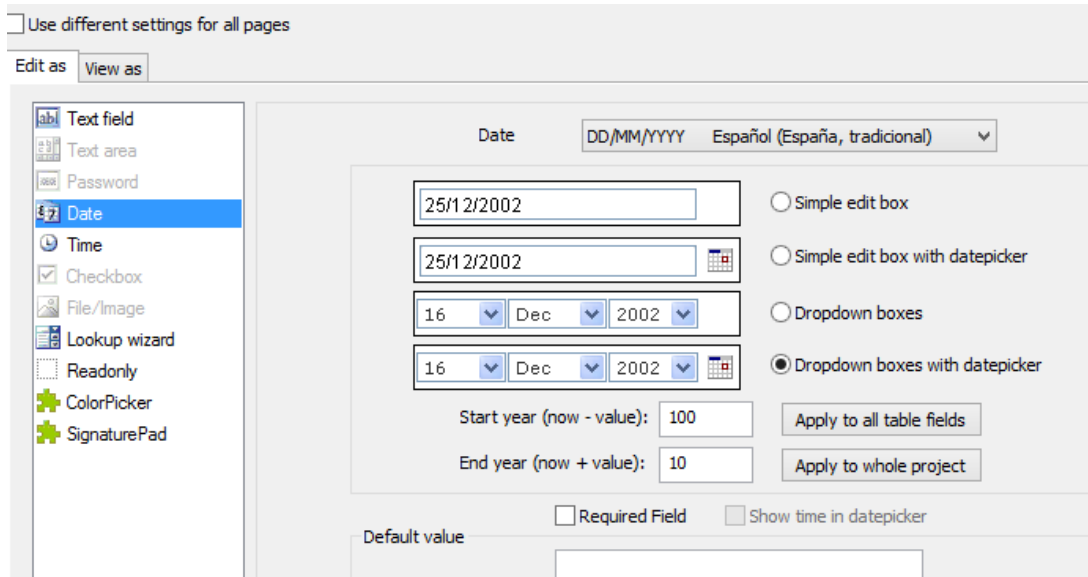


Figura 31 – Distintos formatos de entrada para un campo de tipo “Date”

En esta pantalla también se genera código en segundo plano en formato HTML que es completamente editable, en la siguiente figura vemos un ejemplo de este:

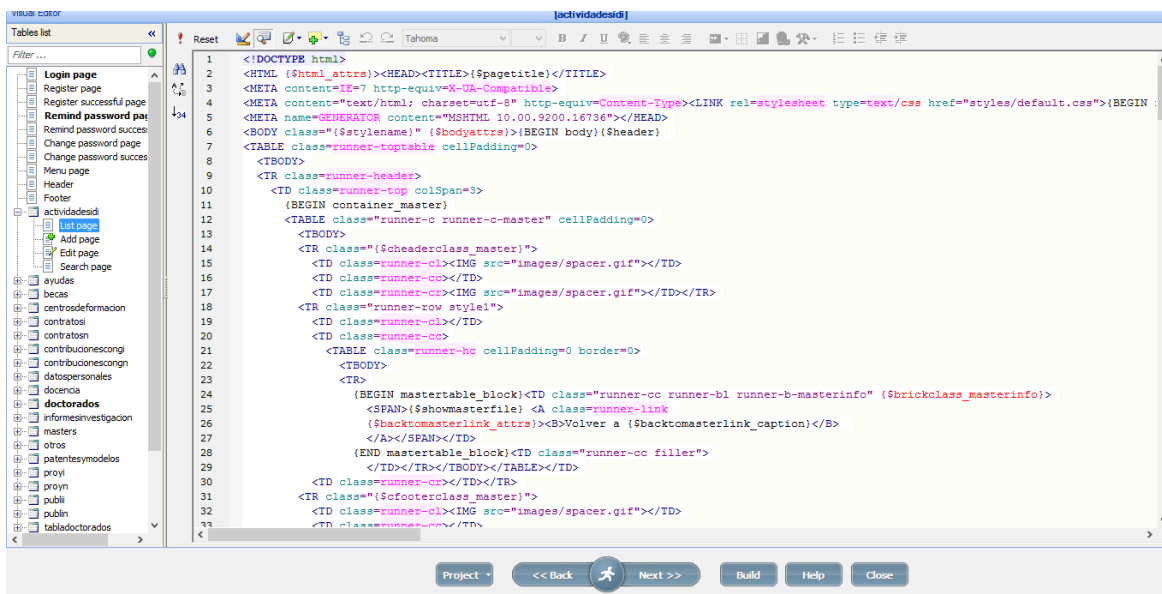


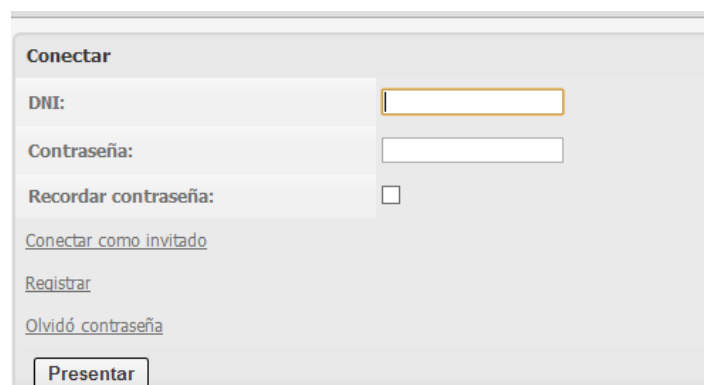
Figura 32 – Código HTML generado automáticamente por PHPRunner

# CAPÍTULO 4

## 4. Manual de usuario

Aquí se va a exponer un sencillo tutorial de uso de la aplicación con las opciones que tiene por el momento.

La primera página que le aparecerá al usuario cuando accede al sistema es la página de identificación



Conectar

DNI:

Contraseña:

Recordar contraseña:

[Conectar como invitado](#)

[Registrar](#)

[Olvidó contraseña](#)

Figura 33 – Identificación de usuario

La primera vez que se accede al sistema tienes dos opciones, o conectarte como invitado o solicitar un registro. Con la primera opción no se podrá editar nada de la base de datos, pero se podrá ver información genérica para que tengan una idea de cómo funciona la aplicación, pudiendo moverse por todos los menús de la web.

Con la segunda opción solicitamos acceder al sistema registrándonos con un nombre de usuario, una contraseña y una dirección de correo electrónico, como vemos a continuación:

Registrar

Usuario: ejemplo \*

Contraseña: \*\*\*\*\* \*

Re-escribir la contraseña: \*\*\*\*\* \*

Email: ejemplo@web.com \*

\* - Campo de requerimiento

[Regresar a la página de conexión](#)

Presentar

Figura 34 – registro de usuario

Si alguno de estos campos creara algún conflicto con la tabla existente no se realiza el registro, esto es, que ya haya un usuario en el sistema con el mismo nombre, que la contraseña no sea idéntica en los dos campos o que el formato de correo no sea el correcto. Además vemos también que los cuatro campos son obligatorios para poder realizar el registro.

Una vez estemos registrados en el sistema tenemos la opción de recuperar la contraseña en caso de que se nos haya olvidado, esta es la razón de que se necesite introducir el correo electrónico en la página de registro.

Tras registrarnos en el sistema de manera satisfactoria volveremos a la página de autenticación y ya podremos introducir nuestro usuario y contraseña que especificamos anteriormente.

Una vez dentro del sistema registrado como usuario nos encontramos el menú principal de navegación por la web en la parte izquierda de la pantalla, como hemos explicado antes las distintas páginas están agrupadas en secciones y para verlas deberemos darle al desplegable que hay a la izquierda de cada nombre de sección. Algunas páginas como la correspondiente a los datos personales o la correspondiente a la experiencia docente no está dentro de ninguna sección, por lo que podemos acceder a ella directamente clicando en directamente en el nombre. A continuación se muestra el aspecto de este menú.



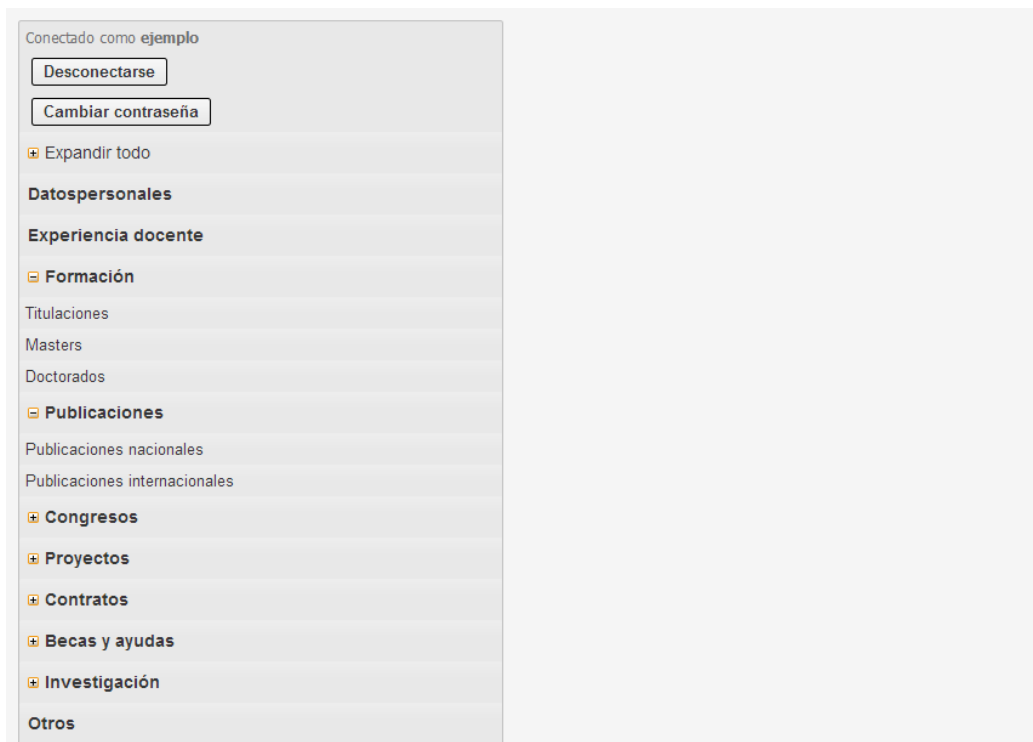


Figura 35 – Menú principal de la web

Como vemos en la figura 35 las secciones serían:

- Formación
- Publicaciones
- Congresos
- Proyectos
- Contratos
- Becas y ayudas
- Investigación

Dentro de cada una se pueden encontrar las tablas correspondientes. En la parte superior de la web vemos que aparece nuestro nombre de usuario, la opción de desconectarnos del sistema y cambiar nuestra contraseña. En el menú principal además podemos encontrar esto en la parte inferior:

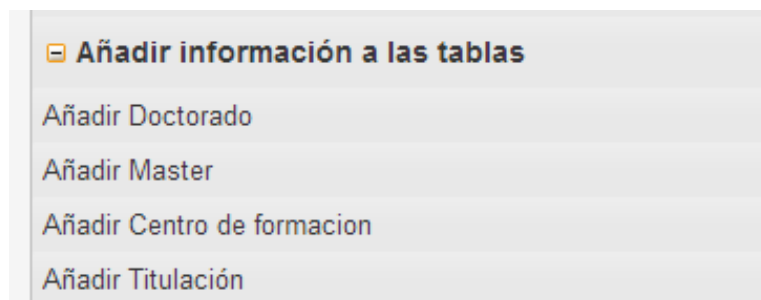


Figura 36 – Sección “añadir información”

Esta es la última sección que encontramos en el menú principal, aquí podemos añadir información en las cuatro tablas disponibles para ello. Cuando pulsamos cualquiera de estos cuatro enlaces nos aparecerá lo siguiente, por ejemplo si pulsamos en “añadir Centro de formación”:

A screenshot of a web form titled "Centros de formación, Añadir nuevo registro". The form has a blue header bar with the title. Below the header, there is a text input field labeled "Nombre Centro". At the bottom of the form, there are two buttons: "Guardar" and "Volver a la lista".

Figura 37 – Añadir nuevo Centro de formación

Aquí ya podemos introducir el nombre del Centro de formación que queremos añadir y le damos a guardar. Se añadirá a la tabla de los centros de formación y cuando queramos añadir un nuevo algún registro nuevo en alguna de las tablas que dependa de esta nos aparecerá el centro que hemos añadido además de los ya existentes.

Si lo que queremos es añadir un nuevo registro en cualquiera de las otras tablas debemos seguir los siguientes pasos, por ejemplo, si quisiéramos añadir una nueva tesis doctoral a nuestro CV deberíamos, primero clicamos en la sección “investigación” y a continuación en “Tesis Doctorales”, si no tenemos ningún registro añadido anteriormente nos encontraremos lo siguiente:

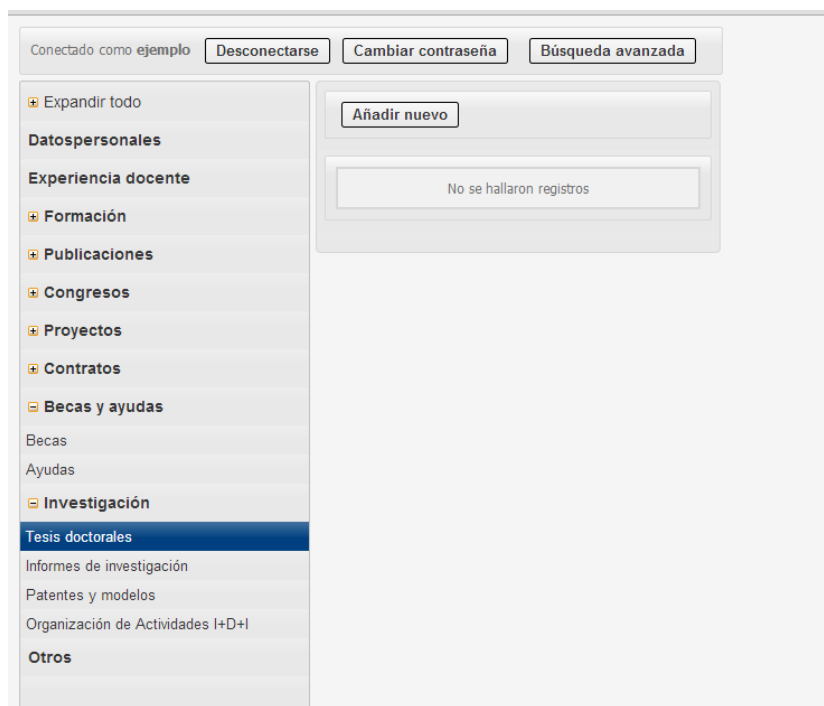


Figura 38 – Página Tesis doctorales vacía

A continuación pulsamos en “Añadir nuevo” y nos llevará a la página de añadir nuevo registro de la tabla *TesisDoctorales*, que tendrá este aspecto:

The image shows a form titled 'Tesisdoctorales, Añadir nuevo registro'. The form contains the following fields: 'Titulo' (text input), 'Doctorando' (text input), 'Universidad' (dropdown menu with the text 'Por favor seleccione'), 'Facultad' (text input), 'Fecha' (date picker with three dropdown menus for day, month, and year), and 'Calificacion' (text input). At the bottom of the form, there are two buttons: 'Guardar' and 'Volver a la lista'.

Figura 39 – Página para añadir nueva Tesis doctoral

Tras rellenar los campos y darle a “Guardar” nos aparecerá el siguiente mensaje:

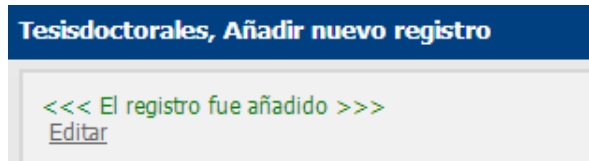


Figura 40 – Nuevo registro añadido

A continuación podemos volver al listado pulsando en “Volver a la lista” y podemos ver que nuestro registro ya aparece en ella.

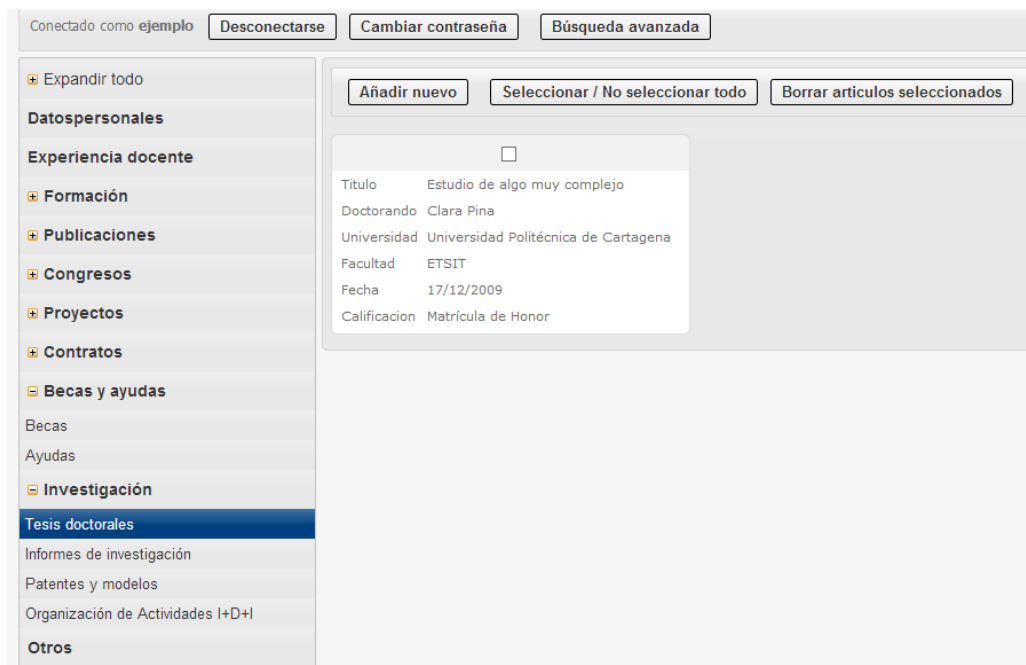


Figura 41 – Página Tesis doctorales con registro

Además también vemos que nos han salido dos nuevos botones de edición, el de “Seleccionar/No seleccionar todo” y el de “Borrar artículos”. Si marcamos la entrada que acabamos de introducir y le damos a “Borrar artículos” esta se eliminará de la tabla.

# CAPÍTULO 5

## 5. Conclusiones

La conclusión principal a la que se ha llegado tras la realización de este trabajo es que el controlar una base de datos medianamente grande es un proceso bastante laborioso, ya que tienes que tener en cuenta muchos factores. Además tenemos que añadir a este hecho que la idea es montar la aplicación en un entorno “web 2.0” mediante lenguaje PHP. Todos estos factores hacen que la aplicación PHPRunner sea imprescindible, ya que encaja “como anillo al dedo” para nuestros propósitos. Aunque a primera vista es una herramienta aparentemente difícil de aprender, al final resulta ser más intuitiva de lo aparenta y nos hemos dado cuenta de que tiene un potencial importante que podemos explotar, esto nos da pie a pensar que esta herramienta se puede optimizar mucho, como veremos más adelante en este capítulo.

### 5.1. Dificultades encontradas

Las dificultades encontradas han sido principalmente aprender a utilizar la aplicación PHPRunner desde cero y aprender cómo funciona una base de datos. Sabemos que con un buen uso de sentencias SQL y del lenguaje PHP podemos optimizar la aplicación muchísimo más. Ante estas dificultades, decidimos centrarnos en aprender a manejar bien la aplicación PHPRunner y en crear una base de datos robusta y que funcione para que nos sirva de base de una aplicación mucho más desarrollada.

Como vimos anteriormente, existe una forma de agilizar la creación de una tabla a partir de otra que ya está generada. Esto nos suponía trabajar el doble y conseguimos solventarlo de la siguiente manera:

Si queremos añadir una tabla creada en Excel en nuestra base de datos lo que tenemos que hacer es guardar el archivo, pero no como documento de Excel, sino como csv. En las opciones de “Guardar como” seleccionaremos la opción “CSV (Delimitado por comas)(\* .csv)“. Ahora abrimos el archivo con un editor de texto y vamos a eliminar la

primera línea. Esta es la que tiene el título de los campos, y no nos interesa, si la dejamos se rellenaría nuestra base de datos en MySQL con esos valores. A continuación vamos a la herramienta PhpMyAdmin y creamos una nueva base de datos, acto seguido creamos una tabla, y añadiremos los campos correspondientes según nuestra tabla. Por último tenemos que escribir una sentencia SQL que lo que hará será tomar un archivo como referencia, leerlo, e interpretarlo según dos parámetros que le indiquemos:

- Los datos que van en cada campo se separan mediante una coma (“,”)
- Los diferentes registros se separan por un salto de línea (“\n”)

La sentencia la debemos realizar desde la sección de “SQL”, que está en el menú superior y es la siguiente:

```
LOAD DATA INFILE 'ubicación/del/archivo' INTO TABLE 'nombre_de_la_tabla'  
FIELDS TERMINATED BY ','  
LINES TERMINATED BY '\n';
```

Hay que tener en cuenta que la ubicación del archivo dependerá del lugar donde hayamos instalado nuestro programa XAMPP, por defecto en nuestro disco “C://” dentro de la carpeta XAMPP debemos buscar dónde se encuentra la base de datos que contiene la tabla a partir de la cual queremos generar la nueva.

## 5.2. Líneas futuras de trabajo

Para concluir este capítulo vamos a ver los puntos que vamos a seguir para mejorar la aplicación y explotar todos los recursos del PHPRunner. Una de esas partes sería el aspecto final de la aplicación, lo mejor para hacer esta parte es tener un “feedback” de los propios usuarios, preguntando cuál es el aspecto que a ellos les gustaría que tuviese la aplicación o qué funcionalidades adicionales consideran que debería tener. De esta forma podemos hacer un diseño de la aplicación centrado en el usuario, que va a ser el quien hará uso de ella en última instancia, por lo que consideramos que este tipo de diseño iría mucho más acorde con el proyecto. En esta primera toma de contacto no se ha podido realizar todo lo bien que nos hubiera gustado, pero la idea es seguir por ese camino.

En cuanto a mejoras aplicables de la aplicación, se pretenden crear una serie de plantillas de impresión que el usuario pueda utilizar para exportar toda la información. Además queremos que estas plantillas puedan ser personalizables por el propio usuario y que también pueda generar nuevas plantillas.

En cuanto al volcado de la información, se pretende conseguir que a través de las plantillas que acabamos de explicar, se pueda exportar la información en diferentes formatos (PDF, Word, etcétera).

También se quiere conseguir una herramienta mucho más dinámica y personalizable, de forma que el usuario pueda ir añadiendo campos a todas las tablas o quitando los que no necesite.

Otra mejora que nos permite hacer la aplicación PHPRunner es la encriptación de la información, esto es interesante ya que se trata de una aplicación donde se escribe información personal y a la que pueden acceder muchos usuarios. Además sabemos que la aplicación PHPRunner permite tener diferentes grupos de usuarios, esto nos puede servir para tener distintos niveles de privilegios, según el tipo de usuario, de forma que además de tener el usuario que entra como invitado, que solo es capaz de ver el aspecto de la aplicación, se podrían generar dos tipos diferentes de usuarios registrados, uno básico y gratuito en el que se pueda almacenar toda la información del CV en la base de datos de la aplicación y otro “Premium”, con más herramientas como por ejemplo la exportación de la información, o el poder editar las tablas. También se podría hacer un sistema que los usuarios tuvieran que solicitar un permiso de exportación de la información, por lo tanto tendríamos dos formas posibles de realizar esto. La aplicación también permite avisar por correo a los usuarios cuando se realiza una acción, de manera que este sepa en todo momento lo que está ocurriendo con su información.

Es sabido que normalmente las personas se muestran reticentes a poner su información en la web y para solventar esto también sería posible desarrollar la misma aplicación de uso off-line, instalando una serie de paquetes necesarios para poder trabajar como un servidor de bases de datos y un sistema gestor de la misma.

# CAPÍTULO 6

## 6. Anexo: Bibliografía y referencias

- [1] UM, Sistema Gestor de Bases de Datos  
<http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/sgbd.html>
- [2] Ecured, Servidor de Bases de datos  
[http://www.ecured.cu/index.php/Servidor\\_Bases\\_de\\_Datos](http://www.ecured.cu/index.php/Servidor_Bases_de_Datos)
- [3] Ecured, Bases de datos  
[http://www.ecured.cu/index.php/Bases\\_de\\_Datos](http://www.ecured.cu/index.php/Bases_de_Datos)
- [4] Ecured, PHP  
<http://www.ecured.cu/index.php/PHP>
- [5] Ecured, Servidor web Apache  
[http://www.ecured.cu/index.php/Apache\\_Servidor\\_Web](http://www.ecured.cu/index.php/Apache_Servidor_Web)
- [6] Ecured, Servidor web  
[http://www.ecured.cu/index.php/Servidor\\_web](http://www.ecured.cu/index.php/Servidor_web)
- [7] Apachefriends, XAMPP  
<http://www.apachefriends.org/es/xampp.html>
- [8] Aimbox, PHPRunner  
<http://www.aimbox.net/introduccion-phprunner/>
- [9] Conclase  
[www.conclase.net](http://www.conclase.net)
- [10] w3school  
[www.w3schools.com](http://www.w3schools.com)
- [11] PHP  
<http://www.php.net/>
- [12] Digitallearning, primeros pasos XAMPP  
<http://www.digitallearning.es/blog/xampp-instalacion-servidor-multiplataforma-curso-php/>
- [13] stackoverflow  
<http://stackoverflow.com/questions/tagged/database>



- [14] Nlinesoft, PHPRunner  
<http://xlinesoft.com/phprunner/docs/>
- [15] Wikipedia, PHPRunner  
<http://en.wikipedia.org/wiki/PHPRunner>
- [16] Wikipedia, HTML  
<http://es.wikipedia.org/wiki/Html>
- [17] UV, Guía HTML  
<http://www.uv.es/jac/guia/>
- [18] Aulaclíc, Guías varias  
<http://www.aulacli.c.es/>
- [19] Wikipedia, PHP  
<http://es.wikipedia.org/wiki/Php>
- [20] desarrolloweb, Guía PHP  
<http://www.desarrolloweb.com/php/>
- [21] Wikipedia, SQL  
<http://es.wikipedia.org/wiki/SQL>
- [22] 1keydata, Guía SQL  
<http://www.1keydata.com/es/sql/>
- [23] Desarrolloweb, Tutorial SQL  
<http://www.desarrolloweb.com/manuales/9/>
- [24] Sqlcourse, Guía SQL  
<http://www.sqlcourse.com/index.html>
- [25] Wikipedia, Aplicación web  
[http://es.wikipedia.org/wiki/Aplicaci%C3%B3n\\_web](http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web)