



Ciclo formativo:	Técnico Superior en Desarrollo de Aplicaciones Web
Módulo:	Desarrollo Web en entorno servidor
Unidad didáctica:	Tarea 1 Obligatoria

Índice

Enunciado.....3

Paso 1.....6

Paso 2.....6

Paso 3.....7

Paso 4.....7

Paso 5.....8

Paso 6.....9

Criterios de evaluación.....10

Entregable.....10


# Enunciado

Con esta tarea vamos a empezar el desarrollo de un sitio web de una biblioteca. El objetivo es crear una aplicación web en la que poner en práctica los contenidos vistos en el primer trimestre:

- Generación dinámica de páginas web.
- Integración con los lenguajes de marcas.
- Etiquetas para inserción de código.
- Tipos de datos. Conversiones entre tipos de datos.
- Variables. Operadores. Ámbitos de utilización.
- Tomas de decisión.
- Bucles.
- Matrices (arrays).
- Funciones.
- Recuperación y utilización de información proveniente del cliente web.
- Procesamiento de la información introducida en un formulario.
- Comentarios.
- Mantenimiento del estado.
- Almacenamiento y recuperación de información en el cliente web.
- Seguridad: usuarios, perfiles, roles.
- Autenticación de usuarios.
- Pruebas y depuración.
- Establecimiento de conexiones.
- Recuperación y edición de información.
- Utilización de conjuntos de resultados.
- Actualización y eliminación de información proveniente de una base de datos.
- Prueba y documentación.

La tarea tendrá que implementar un sistema gestor de libros que admita usuarios y que preste libros a los usuarios. Un usuario se tendrá que registrar para poder entrar en el sistema y ver el catálogo de libros. Una vez registrado podrá consultar los libros y solicitar préstamos de libros que estén disponibles. El administrador del sitio web, aparte de administrar el catálogo, deberá poder administrar los datos de los usuarios en caso de solicitarlo.

La funcionalidad que se pretende se explica a continuación:

INICIO	
	Inicialmente, la aplicación debe mostrar la página principal en el que se muestre el encabezado, el pie de página y el formulario de inicio de sesión. Si el usuario aún no se ha registrado en el sistema, deberá acceder al formulario de registro. Si el usuario ya está registrado, una vez se compruebe que es un usuario válido del sistema, se le redirigirá al catálogo de libros.

## REGISTRO

### Nuevo Lector Persona

Nombre

Primer apellido

Segundo apellido

El formulario de registro recogerá la información necesaria para registrar al usuario. Los campos marcados con asterisco (\*) serán obligatorios, luego hay que controlar que no estén vacíos. Además deberemos comprobar que los campos tienen la información esperada, como un único String el campo login y el campo password o que el correo electrónico está bien formado. Una vez se realice el registro, se informará al usuario que se ha registrado en el sistema y se le remitirá a la página principal.

## CATÁLOGO

Cuando el usuario inicie de sesión, se le reenviará al catálogo de libros. Aquí visualizará en el encabezado su nombre y la posibilidad de cerrar sesión.

▶ Permite seleccionar el campo sobre el que realizamos la búsqueda, simplemente, marcando el campo correspondiente en cualquiera de los registros.

▶ Si la consulta se realiza desde aquí sólo afecta al Catálogo y equivale a la aplicación de un filtro sobre el mismo. No se realiza sobre el/los Depósito/s.

The screenshot shows the 'Catálogo' application window. At the top, there's a menu bar with 'Archivo', 'Ir a', 'Herramientas', and 'Ayuda'. Below it, a navigation bar includes 'Página principal', 'Catálogo', 'Depósito', 'Préstamos', 'Estadística', 'Lectores', and 'Auxiliares'. The 'Catálogo' section has a search bar with 'Buscar' and 'Ejemplar' buttons. A table lists book records with columns: Ejemplar, Título, Autor, Editorial, Registro, and Tipo. The table contains three rows of data. On the left, a sidebar menu offers options like 'Consulta simple', 'Consulta avanzada', and various print functions. At the bottom, there are buttons for 'Nuevo...', 'Detalles', 'Eliminar', 'Actualizar', and 'Lecturas'. A status bar at the very bottom shows 'Pulse F1 para ayuda', 'Administrador', and 'Inversos actualizados'.

Ejemplar	Título	Autor	Editorial	Registro	Tipo
000001R	La función del orientador	Tyler, Leona E.	Trillas		Nom
000002W	La acción tutorial en educación secundaria	Sin Autor	Escuela Española		Nom
000003A	El Acceso a la T				

**Datos del fondo**

Tipo	Libro
Título	La función del orientador
Autor	Tyler, Leona E.
Editorial	Trillas
Año ed.	1972

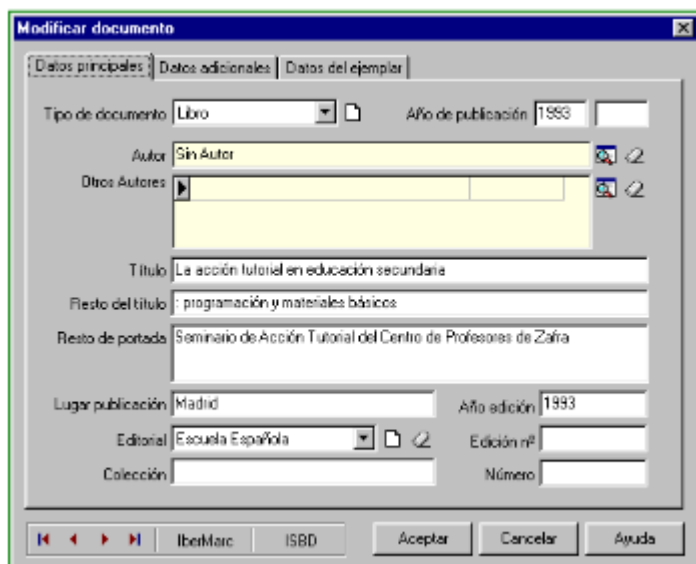
**Datos del ejemplar**

Ejemplar	000001R
Registro	
Tipo	Normal
Signatura	37-TYL-fun
Ubicación	4-2

**Términos significativos por los que podemos realizar búsquedas**

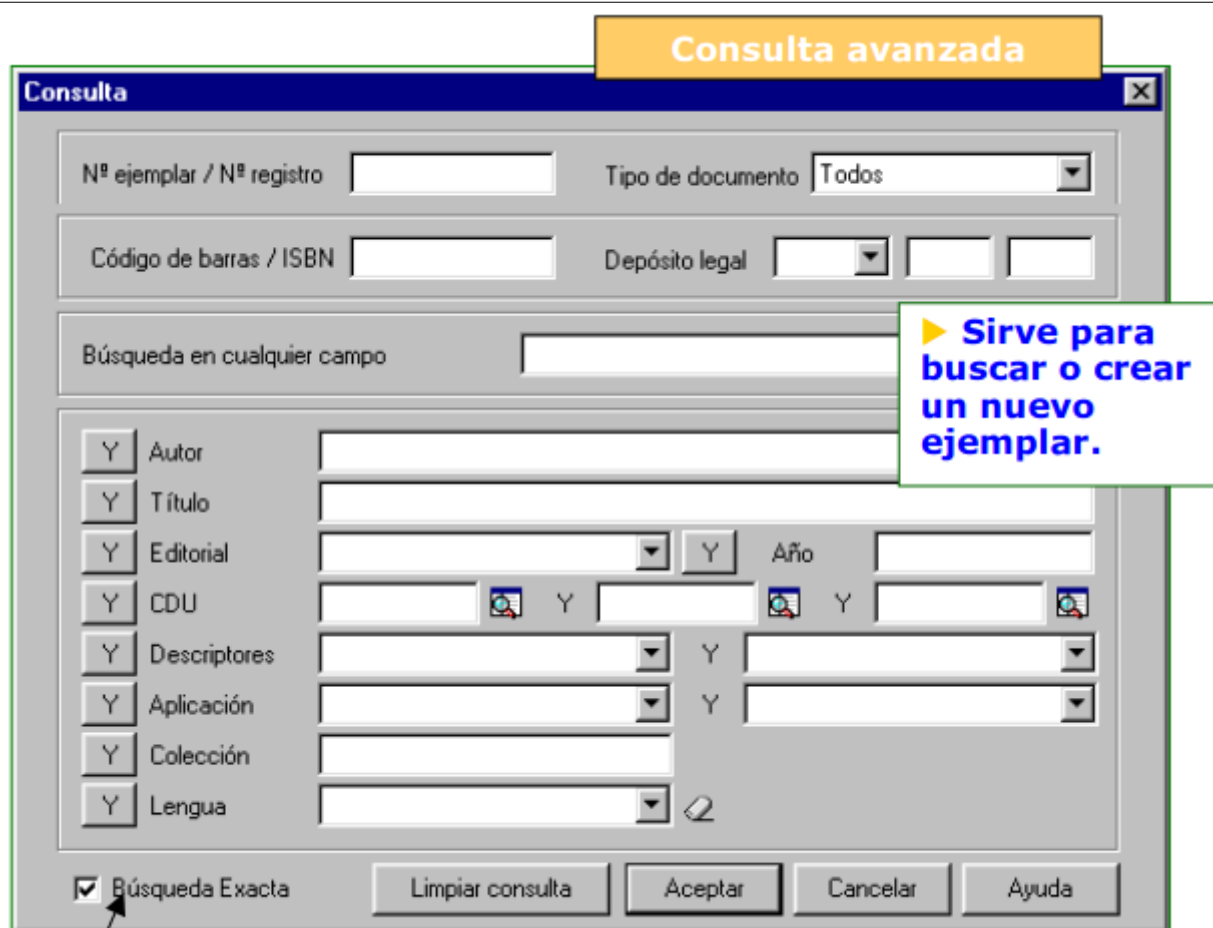
Al seleccionar un libro del catálogo nos mostrará la información detallada y el estado del libro (Disponible | Prestado)

## CATALOGACIÓN



Sólo el administrador podrá dar de alta libros en el catálogo. A través del formulario introducirá los datos en el sistema. Se tendrá que comprobar que el libro ya no está en el sistema como paso previo a introducirlo para evitar datos duplicados en el sistema.

## BUSCADOR



La aplicación tendrá un buscador simple de libros y un buscador avanzado donde poder elegir los criterios por los que realizar la búsqueda

## PRÉSTAMOS

The screenshot shows the 'Préstamos' (Loans) section of a web application. The interface includes a top navigation bar with links like 'Página principal', 'Catálogo', 'Depósito', 'Préstamos', 'Estadística', 'Lectores', and 'Auxiliares'. A sidebar on the left contains a menu with options: 'Prestar ejemplar', 'Devolver ejemplar', 'Prolongar préstamo', 'Anular préstamo', 'Imprimir listado', and 'Informe'. The main content area displays a table of 'Ejemplares en préstamo' with columns for 'Ejemplar', 'Título', 'Autor', 'Lector', and 'Apellidos'. Below the table, there are sections for 'Datos del ejemplar' and 'Datos del lector'. At the bottom, there are filters for 'Filtro por ejemplar', 'Filtro por curso', 'Filtro por lector', and 'Retrasos'. A callout box on the right indicates that buttons for updating and deleting records are only active if there are documents on loan.

La opción por defecto es "Búsqueda por ejemplar".

Hace referencia a todos los libros prestados. Aparece con dos líneas por registro (ejemplar-préstamo y lector).

Barra de desplazamiento para ver los campos que no quedan a la vista.

Icono retrasos

Las operaciones están agrupadas en la misma ventana para facilitar el acceso a las mismas.

Para poder prestar documentos, lógicamente, es necesario tener documentos catalogados y lectores dados de alta y con el carné vigente.

Estos botones sólo pueden activarse si existen documentos prestados que cumplan el requisito correspondiente.

Filtro por ejemplar Filtro por curso Filtro por lector Retrasos

64

Cada usuario podrá consultar los libros prestados del sistema. El administrador tendrá que visualizar los usuarios que tienen libros prestados y los libros que han sacado.

Para facilitar el buen desarrollo de la práctica, se recomienda afrontar el proyecto en diferentes pasos o fases, tal como se describe a continuación.

### Paso 1

En este paso vamos a realizar la parte estática del sitio web. Por ahora, estas páginas contienen datos estáticos que en prácticas posteriores se sustituirán por datos reales obtenidos de una base de datos.

Antes de ponerte a escribir el código HTML, realiza un diagrama de las páginas web que componen tu sitio web. Este diagrama puede tener tanto detalle como tú quieras, pero hay que encontrar un equilibrio entre utilidad y legibilidad: un diagrama muy simple y con poco contenido será poco útil, mientras que un diagrama muy complejo y con mucha información será difícil de entender y no supondrá una ayuda.

Crea una página web "esqueleto" a partir de la cual puedas crear todas las páginas de tu sitio web. En esta página incluye todos los elementos que se repitan en todas las páginas (cabecera y pie de página, barra de navegación). Asegúrate que el código de esta página web es correcto. Mira el código de la página del W3C y copia lo mínimo necesario para crear una página web correcta.

Cuando escribas el código HTML a mano, utiliza la herramienta W3C Markup Validation. La herramienta de validación del W3C sólo permite validar las páginas web de una en una. Si necesitas validar todo un sitio web, la herramienta Multipage Validator 30 permite validar múltiples páginas a partir de una única dirección.

### Paso 2

En este paso tienes que crear el estilo visual de las páginas web. El esquema mostrado es el resultado de mostrar el sitio web "en crudo", esto es, sin ningún aspecto visual.

Para definir el estilo visual tienes que emplear el lenguaje de estilos CSS. No tienes que crear un fichero distinto para cada página, sino que el mismo fichero lo utilizarás en todas las páginas, a no ser que existan algunas páginas con estilos visuales muy diferentes (por ejemplo, la página de inicio, la página principal y el resto de páginas).

Tienes varias formas de crear una hoja de estilo CSS:

1. Crear una hoja CSS desde cero: esto lo podrás hacer cuando ya tengas un conocimiento amplio del lenguaje CSS.
2. Tomar un CSS como “inspiración” para crear un CSS propio. En este caso, es una buena práctica citar al autor de la o de las hojas CSS en que te has basado, pero si existen grandes diferencias no es del todo necesario. La cita la puedes indicar mediante un comentario en la misma hoja CSS.
3. Adaptar una hoja CSS a la estructura de tu página o, al revés, adaptar la estructura de tu página a una hoja CSS. Cuando usas directamente el CSS de otra persona, primero tienes que consultar si es posible hacerlo, es decir, si el autor de la hoja CSS te da permiso. Lo normal es que te pida (y si no lo hace, hazlo) que cites claramente al autor del CSS en la página web, pero a veces con un simple comentario en la misma hoja CSS puede ser suficiente.

El estilo que definas no debe ser muy simple (no vale con cambiar el color de fondo de la página y definir un estilo para el texto), pero tampoco es necesario que sea muy complejo. Con unas pocas reglas de CSS bien elegidas se puede lograr un estilo visual impactante.

## Paso 3

En este paso tienes que validar los datos de los formularios del lado del cliente. Por ahora sólo podemos utilizar las etiquetas y atributos que me permite HTML5. Más adelante, esta validación se realizará implementando funciones en el Back-End (PHP).

## Paso 4

En este paso tienes que dividir la estructura de las páginas de tu sitio web en múltiples ficheros y tienes que utilizar PHP para combinarlos en uno solo. Detecta la partes que sean comunes a diferentes páginas de tu sitio web (por ejemplo, la cabecera con el título, la barra de navegación y el pie de página) y ponlas en ficheros independientes. De este modo te puedes crear una plantilla a partir de la cual se genera cualquier página web de tu sitio web.

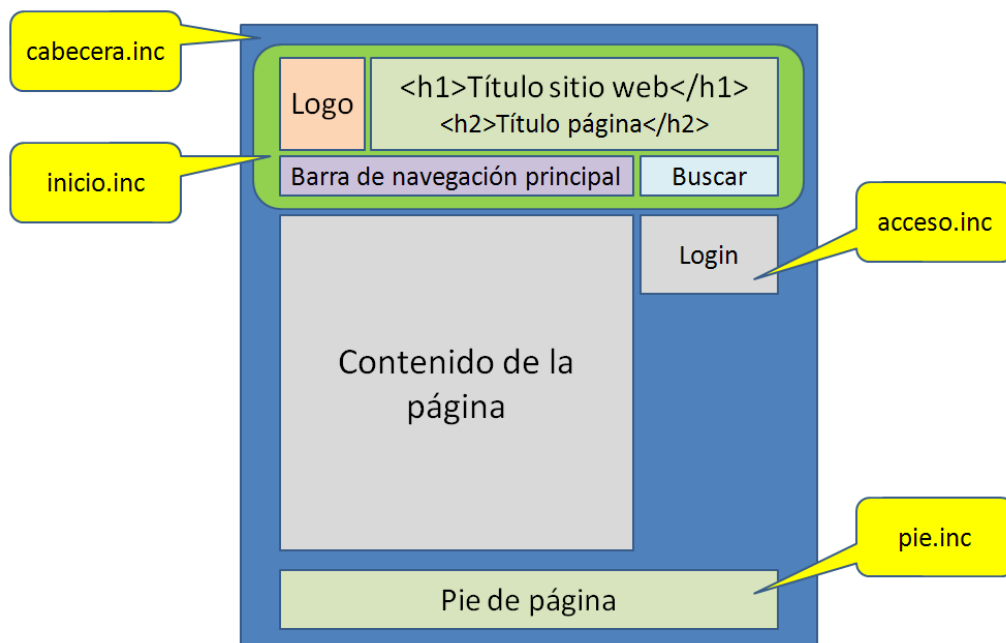
Además, tienes que programar con PHP las páginas web que reciban los datos enviados a partir de los formularios que has realizado en las prácticas anteriores. Por ahora, estas páginas simplemente tienen que mostrar los datos recibidos para comprobar que todo funciona correctamente. En próximas prácticas almacenarás estos datos en una base de datos.

Un sitio web correcto debe mantener una coherencia, tanto visual como de estructura y contenido, entre todas sus páginas web. Para lograrlo, lo normal es trabajar con una plantilla a partir de la cual se crean todas las páginas web. Cuando se quiera hacer un cambio, se modifica la plantilla y todas las páginas web se verán afectadas por el cambio.

En PHP, para incluir en un fichero el contenido de otro fichero se emplean dos funciones muy parecidas, `require(fichero)` e `include(fichero)`:

- `require(fichero)`: si el fichero no existe, se produce un mensaje de error y finaliza la ejecución.
- `include(fichero)`: si el fichero no existe, se produce un mensaje de advertencia y continúa la ejecución

Además, existen `require_once(fichero)` e `include_once(fichero)`: sólo incluyen el fichero indicado una vez, por lo que evitan problemas de redefinición de funciones, reasignación de variables, etc.



Por ejemplo, en la Figura se muestra una plantilla sencilla para un sitio web, donde la estructura y contenido de todas las páginas se ha dividido en cinco partes: la cabecera, el inicio, el control de acceso a la parte privada, el contenido principal de la página y el pie de la página.

## Paso 5

En este paso tienes que emplear las sesiones de PHP para controlar el correcto acceso como usuario registrado. Ahora mismo existe un problema de seguridad en el sitio web: si se conoce la URL, se puede acceder directamente a la página con el menú de usuario registrado sin tener que pasar por la página de control de acceso. Tienes que programar un sistema para que sólo se muestre la página con el menú de usuario registrado a aquellos usuarios que hayan introducido un nombre de usuario y una contraseña correctos.

En el encabezado tienes que mostrar el nombre de usuario del usuario que ha accedido y todas las operaciones que realice el usuario dentro del sitio web tienen que estar autorizadas para los usuarios registrados. (Los usuarios no registrados deben ver una página informativa y no pueden hacer nada)

**Nota:** La parte privada de la aplicación y su integración con la parte pública la puedes plantear de varias formas. Dos formas típicas son:

- **Separada** La parte privada es completamente independiente de la parte pública, posee su propio menú o barra de navegación e incluso puede poseer su propio estilo visual (CSS). Evidentemente, debe existir una opción en el menú o barra de navegación que permita pasar de la parte pública a la parte privada y viceversa.
- **Integrada** La parte privada se integra como una opción más en el menú o barra de navegación de la parte pública. La parte privada aparece como un apartado más de la parte pública, que sólo está disponible cuando el usuario se ha identificado.



En la realización de esta práctica puedes aplicar cualquiera de estas dos estrategias o cualquiera similar.

Para controlar el acceso a una página web de una zona privada se suele emplear una variable de sesión que se inicializa con cierto valor en la página de control de acceso; en las páginas donde se quiere controlar si el usuario tiene permiso para acceder se consulta el valor de la variable de sesión para ver si tiene el valor esperado. Si no contiene el valor esperado, lo normal es mostrar una página con un mensaje de error o redirigir al usuario a la página principal del sitio web.

## Paso 6

En este paso tienes que crear una base de datos en MySQL para almacenar los datos que emplea el sitio web. Los datos que se reciben a través de Internet se tienen que validar y filtrar en el servidor antes de insertarlos en la base de datos, ya que pueden provenir de una fuente no segura. Aunque los datos provengan de un formulario donde se han verificado previamente, no hay que confiar en ellos.

Las funciones de filtrado que tiene PHP (funciones Filter) se pueden emplear para realizar dos tipos de filtrados:

### Validación

- Aplica reglas estrictas de formato (por ejemplo, URL o email).
- Devuelve el dato original o false si no pasa el filtro.

### Saneamiento

- Se emplean para permitir o no permitir ciertos caracteres en los datos.
- No emplea reglas de formato.
- Devuelve la cadena original tras eliminar los caracteres no permitidos.

Otra opción es emplear el uso de las expresiones regulares. Recordemos que una expresión regular es un patrón que define un conjunto de cadenas sin enumerar todos sus elementos. Una expresión regular está formada de caracteres (letras, números y signos) y metacaracteres que tienen una función definida (representar otros caracteres o indicar la forma de combinar los caracteres). Los metacaracteres reciben este nombre porque no se representan a ellos mismos, sino que son interpretados de una manera especial.

Las contraseñas de los usuarios no se almacenan en texto claro en la base de datos. En su lugar se usa el hash de contraseñas que es una de las consideraciones de seguridad más elementales. Sin una función hash, cualquier contraseña que se almacene en la base de datos podrá ser robada si la base de datos se ve comprometida. Al aplicar un algoritmo hash a las contraseñas antes de almacenarlas en la base de datos, dificultamos que el atacante obtenga la contraseña del usuario.

Cuando un archivo es crítico para el negocio, se almacena en la base de datos como objeto binario grande (BLOB). Ahora bien, esta práctica encarece el peso de la base de datos y el manejo de la misma. Una solución cuando no es crítico almacenar los archivos en la base de datos, es almacenar la ruta del sistema de ficheros en la base de datos para que se pueda recuperar en la aplicación. Implementando un robusto sistema de permisos del sistema de ficheros en la ruta que almacena los archivos y un buen sistema de backup, esta opción es igualmente válida.

También es buena práctica crear un archivo de configuración de la base de datos que centralice las variables de conexión de la base de datos con la aplicación. De esta forma, al desplegar la aplicación en otro entorno modificamos la configuración de una manera sencilla y segura.

Es frecuente, además, crear bibliotecas de funciones de acceso a base de datos, gestión de sesiones... Todo diseño que me permita obtener una buena modularidad en el sitio web es bien recibido por la descomposición del sistema complejo en otros subsistemas más sencillos, la interoperabilidad entre los diferentes módulos y la facilidad de comprensión del sistema en cada una de sus partes.

## Criterios de evaluación

<b>Tarea 1 -Escala de valoración diferenciada</b>		
<b>C. E.</b>	<b>Criterio</b>	<b>Puntos</b>
2.c)	Utiliza etiquetas para incluir código en HTML	0,5
2.g)	Utiliza distintos tipos de variables y operadores	0,5
3.a)	Utiliza mecanismos de decisión	0,5
3.c)	Utiliza arrays para almacenar y recuperar datos	1
3.d)	Crea bibliotecas de funciones y las usa en la aplicación	1
3.f)	Emplea métodos para recuperar información del usuario	1
3.g)	Utiliza comentarios en la aplicación	0,5
4.b)	Utiliza mecanismos para mantener el estado de la aplicación	1
4.e)	La aplicación integra mecanismos de autenticación de usuarios	1
6.d)	La aplicación maneja información de la base de datos	1
6.f)	La aplicación manipula información de la base de datos	1
6.g)	La aplicación funciona y está documentada	1

## Entregable

Se considera una practica completa si se entrega los siguientes ficheros:

- Memoria de la práctica en la que se explica cómo se ha implementado la aplicación web y justifica las decisiones tomadas en el diseño.
- Código fuente de los archivos PHP, CSS....
- Scripts de creación e inserción inicial de datos en la base de datos
- Carpeta del proyecto que contendrá:
  - Directorios y ficheros de la aplicación
  - Script de creación e inserción inicial de datos de la aplicación
  - Memoria de la práctica