

## BIBLIOTECA

Para la gestión de una biblioteca se propone el siguiente modelo relacional compuesto por tres tablas:

### 1) `libros`:

- Código: número entero (3), no nulo.
- Título: cadena de texto (50), no nulo.
- Autor: cadena de texto (30), no nulo.
- Año: número entero (4), no nulo.
- Número de Ejemplares: número entero (2), no nulo.
- La clave primaria de `libros` es el código.

### 2) `socios`:

- Código: número entero (3), no nulo.
- Nombre: cadena de texto (30), no nulo.
- Domicilio: cadena de texto (50), no nulo.
- Teléfono: número entero (9), no nulo.
- Correo: cadena de texto (30), no nulo.
- La clave primaria de `socios` es el código.

### 3) `prestamos`:

- Código de Libro: número entero (3), no nulo, clave ajena de `libros`.
- Código de Socio: número entero (3), no nulo, clave ajena de `socios`.
- Fecha de Inicio de Préstamo: fecha (DD/MM/YYYY), no nulo.
- Fecha de Fin de Préstamo: fecha (DD/MM/YYYY), no nulo.
- Fecha de Devolución Real: fecha (DD/MM/YYYY).
- La clave primaria de `prestamos` es compuesta y está formada por el código del libro, el código del socio y la fecha de inicio del préstamo.

Notas:

- El número de ejemplares de un libro indica cuántos ejemplares o copias de ese libro hay disponibles actualmente en la biblioteca. Por tanto:
  - Cuando la biblioteca presta un libro a un socio, su número de ejemplares se decrementa en una unidad.
  - Cuando el socio devuelve dicho libro a la biblioteca, su número de ejemplares se incrementa en una unidad.
- Como máximo, un socio sólo puede sacar un libro de la biblioteca al día.

## EJERCICIO 1

Con la aplicación web phpMyAdmin de administración de bases de datos MySQL, crea una base de datos llamada `biblioteca` y con cotejamiento `utf8_general_ci`. Crea también la tabla `libros` e inserta en ella varias filas de algunos libros conocidos.

En NetBeans, crea un nuevo proyecto PHP denominado `biblioteca` y codifica los siguientes ficheros usando la extensión **MySQLi** de PHP:

- A) **`insertar_libro.php`**. Esta página web muestra un formulario que contiene todos los campos del libro. Cuando el usuario introduce los datos de un nuevo libro y pulsa el botón *Insertar Libro*, entonces esta página se conecta con MySQL e inserta el nuevo libro en la tabla `libros`. Se debe visualizar un mensaje de salida, tanto si la inserción se ha realizado con éxito como si ha habido algún error.
- B) **`modificar_libro.php`**. Esta página web muestra un formulario que contiene todos los campos del libro. Cuando el usuario introduce el código del libro a modificar y los nuevos datos del libro y pulsa el botón *Modificar Libro*, entonces esta página se conecta a MySQL y modifica el libro en la tabla `libros`. Se debe visualizar un mensaje de salida, tanto si la modificación se ha realizado con éxito como si ha habido algún error.
- C) **`eliminar_libro.php`**. Esta página web muestra un formulario que contiene el código del libro. Cuando el usuario introduce el código del libro a eliminar y pulsa el botón *Eliminar Libro*, entonces esta página se conecta a MySQL y elimina el libro de la tabla `libros`. Se debe visualizar un mensaje de salida, tanto si la eliminación se ha realizado con éxito como si ha habido algún error.
- D) **`consultar_libros.php`**. Esta página web se conecta a MySQL, obtiene todos los libros (con todos sus campos) almacenados en la tabla `libros` y los visualiza en una tabla HTML.

## EJERCICIO 2

Con la aplicación web phpMyAdmin de administración de bases de datos MySQL, crea la tabla `socios` e inserta en ella varias filas de algunos socios ficticios.

En NetBeans, utiliza el proyecto `biblioteca` creado anteriormente y codifica los siguientes ficheros usando **PHP Data Objects (PDO)**:

- A) **`insertar_socio.php`**. Esta página web muestra un formulario que contiene todos los campos del socio. Cuando el usuario introduce los datos de un nuevo socio y pulsa el botón *Insertar Socio*, entonces esta página se conecta con MySQL e inserta el nuevo socio en la tabla `socios`. Se debe visualizar un mensaje de salida, tanto si la inserción se ha realizado con éxito como si ha habido algún error.
- B) **`modificar_socio.php`**. Esta página web muestra un formulario que contiene todos los campos del socio. Cuando el usuario introduce el código del socio a modificar y los nuevos datos del socio y pulsa el botón *Modificar Socio*, entonces esta página se conecta a MySQL y modifica el socio en la tabla `socios`. Se debe visualizar un mensaje de salida, tanto si la modificación se ha realizado con éxito como si ha habido algún error.
- C) **`eliminar_socio.php`**. Esta página web muestra un formulario que contiene el código del socio. Cuando el usuario introduce el código del socio a eliminar y pulsa el botón *Eliminar Socio*, entonces esta página se conecta a MySQL y elimina el socio de la tabla `socios`. Se debe visualizar un mensaje de salida, tanto si la eliminación se ha realizado con éxito como si ha habido algún error.

- D) **consultar\_socios.php**. Esta página web se conecta a MySQL, obtiene todos los socios (con todos sus campos) almacenados en la tabla `socios` y los visualiza en una tabla HTML.

### EJERCICIO 3

Con la aplicación web phpMyAdmin de administración de bases de datos MySQL, crea la tabla `prestamos`.

En NetBeans, utiliza el proyecto `biblioteca` creado anteriormente y codifica los siguientes ficheros usando **transacciones** con **MySQLi** o **PHP Data Objects (PDO)**:

- A) **insertar\_prestamo.php**. Esta página web muestra un formulario que contiene todos los campos del préstamo excepto la fecha de devolución real. Cuando el usuario introduce los datos de un nuevo préstamo y pulsa el botón *Insertar Préstamo*, entonces esta página se conecta con MySQL, modifica el libro referenciado (decrementando su número de ejemplares en una unidad) en la tabla `libros` e inserta el nuevo préstamo en la tabla `prestamos`. Se debe visualizar un mensaje de salida, tanto si la inserción se ha realizado con éxito como si ha habido algún error.
- B) **modificar\_prestamo.php**. Esta página web muestra un formulario que contiene todos los campos del préstamo. Cuando el usuario introduce los datos identificativos (código de libro, código de socio y fecha de inicio) del préstamo a modificar y la nueva fecha de devolución real y pulsa el botón *Modificar Préstamo*, entonces esta página se conecta a MySQL, modifica el préstamo en la tabla `prestamos` y modifica el libro referenciado (incrementando su número de ejemplares en una unidad) en la tabla `libros`. Se debe visualizar un mensaje de salida, tanto si la modificación se ha realizado con éxito como si ha habido algún error.

### EJERCICIO 4

En NetBeans, utiliza el proyecto `biblioteca` creado anteriormente y codifica los siguientes ficheros usando **sentencias preparadas** con **MySQLi** o **PHP Data Objects (PDO)**:

- A) **consultar\_libros\_por\_autores.php**. Esta página web muestra un formulario que contiene los autores de los libros. Cuando el usuario introduce los autores (separados por comas) y pulsa el botón *Consultar Libros por Autores*, entonces esta página se conecta a MySQL, utiliza una consulta preparada (con parámetros por nombre) para obtener todos los libros (con todos sus campos) de cada autor de la tabla `libros` y los visualiza en varias tablas HTML, una para cada consulta referida a un autor. Se debe visualizar un mensaje de salida, si ha habido algún error durante las consultas.
- B) **consultar\_socios\_con\_libros\_prestados.php**. Esta página web se conecta a MySQL, obtiene los datos relevantes (código de libro, código de socio y fecha de fin de préstamo) de los préstamos no devueltos, utiliza consultas preparadas (con parámetros por posición) para obtener el título del libro y el nombre del socio relacionados con cada préstamo, y los visualiza junto con la fecha de fin en una tabla HTML. Se debe visualizar un mensaje de salida, si ha habido algún error durante las consultas.