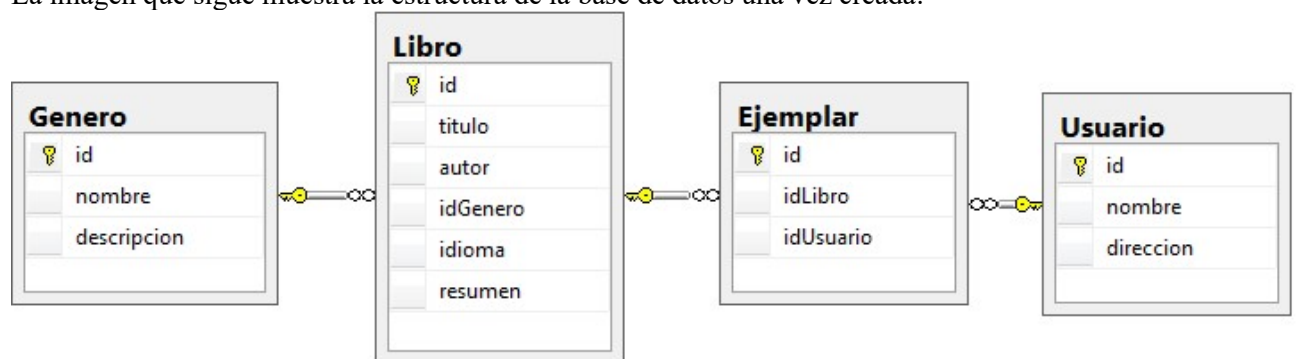


## Ejercicio sobre la gestión de préstamos de libros de una biblioteca con consulta y actualización de una BD

Se trata de crear una aplicación web PHP que permita realizar la devolución y préstamo de los libros de una biblioteca. Para ello,

1. Crear en el SGBD SQL Server 2019 una base de datos con el nombre `Biblioteca`
2. La base de datos `Biblioteca` tendrá la siguiente estructura:
  - Una tabla `Genero` para mantener información sobre los géneros literarios. Esta tabla debe contar con los siguientes campos:
    - `nombre`, una cadena de 50 caracteres de longitud variable y codificada en UTF
    - `descripcion`, una cadena de 2000 caracteres de longitud variable y codificada en UTF
    - La clave primaria será un entero auto incremental (`idGenero`)
  - Una tabla `Libro` para mantener información sobre los libros de los que la biblioteca tiene algún ejemplar. Esta tabla debe contar con los siguientes campos:
    - `titulo`, una cadena de 100 caracteres de longitud variable y codificada en UTF
    - `autor`, una cadena de 100 caracteres de longitud variable y codificada en UTF
    - `idGenero`, una referencia a la tabla `Genero`
    - `idioma`, una cadena de 20 caracteres de longitud variable y codificada en UTF
    - `resumen`, una cadena de 2000 caracteres de longitud variable y codificada en UTF que mantiene un resumen sobre el contenido del libro
    - La clave primaria será un entero auto incremental (`idLibro`)
  - Una tabla `Ejemplar` para mantener información individualizada de los ejemplares físicos de los libros de los que la biblioteca tiene algún ejemplar. Esta tabla es una tabla de estado, no es una tabla histórica, por lo que los datos que mantiene se refieren a hechos actuales, no pasados. Esta tabla debe contar con los siguientes campos:
    - `idLibro`, una referencia a la tabla `Libro`, del cual éste es un ejemplar concreto
    - `idUsuario` mantiene el estado de *prestado* del libro: si su valor es `null`, el libro no está prestado; si su valor es un `idUsuario`, el libro está prestado y el dato informa del usuario que lo tomó prestado
    - La clave primaria será un entero auto incremental (`idEjemplar`)
  - Una tabla `Usuario` para mantener información sobre los usuarios que pueden tomar libros prestados de la biblioteca. Esta tabla debe contar con los siguientes campos:
    - `nombre`, una cadena de 80 caracteres de longitud variable y codificada en UTF
    - `direccion`, una cadena de 200 caracteres de longitud variable y codificada en UTF

La imagen que sigue muestra la estructura de la base de datos una vez creada:



3. Poblar las tablas con información. Para ello buscar por internet información sobre géneros y libros. En cuanto a géneros, incluir los siguientes géneros: Sátira, Cuento, Novela, Poema épico, Cantar de gesta, Romance, Comedia, Drama, Tragedia, Historia, Novela histórica, Viajes y Novela policíaca. En cuanto a libros, incluir, como mínimo, los siguientes libros: Rinconete y Cortadillo, Much Ado About Nothing, La vida es sueño, La tragedia de Hamlet, Fausto, L' Avare, Decamerón, Veinte mil leguas de viaje submarino, Guerra y paz, Tres tristes tigres, Pantaleón y las visitadoras, Ficciones, Bestiario, Una Historia de España, After Dark (en castellano), The Caves of Steel, Las damas de oriente, La balsa de piedra, El médico y El ocho.
4. La aplicación puede desarrollarse de la siguiente manera:
  - a. Si el usuario quiere tomar prestado un ejemplar:
    - i. Podrá consultar los fondos de la biblioteca realizando búsquedas de libros por género, título, idioma o autor. Si todos los ejemplares de algún libro estuvieran prestados el sistema informará de ello. Las búsquedas pueden ser combinadas: género y autor; idioma y autor; idioma, género y autor, etc.
    - ii. Como resultado de búsqueda anterior, el usuario obtiene una relación de los libros que resulten. Esa relación debe incluir: título, nombre del género, autor, idioma, y resumen del libro. Si el usuario necesita conocer lo que un determinado género significa, podrá pinchar en su nombre para obtener esa información. El usuario podrá seleccionar un libro o volver a la vista anterior para realizar otra búsqueda diferente.
    - iii. Si el usuario selecciona un libro en el paso anterior, el sistema le mostrará los ejemplares del libro sin prestar, pinchará en un libro y se le solicitará su número de usuario, tras lo cual el sistema actualizará ese ejemplar como prestado con el número de usuario introducido
    - iv. El usuario se acerca al mostrador y mediante su carnet de usuario solicitará que le entreguen el libro reservado.
  - b. Si el usuario devuelve un ejemplar:
    - i. Se acercará al mostrador y entregará el ejemplar al operario de la biblioteca que procederá a buscar ese ejemplar en el sistema y a informarlo como devuelto.

## Comenzar

1. Copia la estructura MVC y renómbrala como `biblioteca`
2. Crea la vista para consultar libros con datos constantes, no es necesario que incluya datos de los libros que contiene la base de datos, es un prototipo de la vista. La vista puede presentar una serie de libros recomendados del mes y los controles para realizar búsquedas de libros. Al principio, no es necesario incluir todos los controles, puedes empezar mostrando un control para buscar por título o por autor, y luego agregar el autor o el título, ...
3. Una vez creado el prototipo de la vista, crea en el repositorio el código que obtenga la información que debe incluir la vista anterior a partir de la opción de búsqueda realizada por el usuario
4. Completa la vista `base.php` para que incluya la imagen común a todas las vistas y las posibles interacciones de los usuarios y operarios de la biblioteca
5. Crea una clase controladora y el método adecuado que muestre la vista de consulta de libros, le solicite al repositorio que encuentre los libros relacionados con la búsqueda solicitada por el usuario y, como resultado, muestre los resultados al usuario en la vista consulta de libros. (Tras dar nombre al método, no te olvides de cumplimentar la tabla de rutas para identificar con nombre de la interacción -valor `ctl` especificado en el `href` que recoge la interacción del usuario en la vista `base.php`- el nombre de la clase controladora y el método especializado que atiende la petición del usuario)

- a. Una vez obtengas esta funcionalidad, agrega nuevos controles de búsqueda y completa la funcionalidad de esta parte del ejercicio
- 6. Realiza de forma iterativa y de igual manera el resto del ejercicio:
  - a. Primero, creas un prototipo de vista
  - b. Luego, el repositorio que cumplimente las necesidades de la vista (solicitando datos a la BD, o modificando esos datos al prestar o devolver un ejemplar)
  - c. Por último, crea, en el controlador correspondiente, la lógica de código que gobierna el proceso y sin olvidarte de informar la tabla de rutas