

Biblioteca	
<p>Vamos a gestionar una pequeña biblioteca para nuestra organización, en la que cada empleado, familiar o colaborador pueda tomar prestado un libro durante 10 días.</p>	
<p>Crearemos un array (biblioteca) donde mantendremos todos los libros de la biblioteca, cada libro será un objeto de la clase Libro, cuyas propiedades serán:</p> <ul style="list-style-type: none"> • título • autor • isbn • referencia • fechaPrestamo • fechaDevolucion • usuario <p>Crearemos un array (usuarios) donde mantendremos a los usuarios de la biblioteca, cada usuario será un objeto de la clase Usuario con una única propiedad:</p> <ul style="list-style-type: none"> • nombre 	
Disposición	1
<p>En esta versión tendremos 2 botones. Nuevo Libro y Nuevo Usuario.</p> <p>Nuevo Libro mostrará un formulario con los siguientes datos:</p> <ul style="list-style-type: none"> • Título • Autor • ISBN <p>Dispondrá de un botón Confirmar, que invocará a la función nuevoLibro que añadirá un objeto Libro al array <i>biblioteca</i>.</p> <p>Y también de un botón Cancelar, que limpiará los datos del formulario y lo ocultará</p> <p>Nuevo Usuario mostrará un formulario con el <i>siguiente</i> dato:</p> <ul style="list-style-type: none"> • Nombre <p>Con un botón Confirmar, se invoca a la función nuevoUsuario que añadirá un objeto Usuario al array <i>usuarios</i></p> <p>Un botón Cancelar, que limpiará los datos del formulario y lo ocultará.</p> <p>La disposición incluirá un listado de libros que se generará con la función listarLibros que se describe en un punto posterior.</p> <p>Los formularios Nuevo Libro y Nuevo Usuario, estarán ocultos una vez que se hayan procesado (o cancelado) los datos, hasta que no se vuelvan a activar con los botones.</p>	
nuevoLibro	2
<p>Codificaremos la función nuevoLibro, que se llamará desde el botón Confirmar del formulario Nuevo Libro.</p> <p>Adicionalmente codificaremos una función calculoReferencia, que devolverá una nueva referencia que será un número aleatorio entre 1 y 10000, que no exista entre las referencias de los libros de la biblioteca.</p> <p>La función, tomará los datos del formulario e invocará al constructor de la clase Libro creando un objeto y la añadirá al array <i>biblioteca</i>, con los datos del formulario y la referencia obtenida por la llamada a la función anterior.</p> <p>Cada vez que se añada un libro se refrescará el listado de libros con la función refrescarListado.</p>	

nuevoUsuario	0,5
<p>Codificaremos la función nuevoUsuario, que se llamará desde el botón Confirmar del formulario Nuevo Usuario.</p> <p>La función, tomará el dato del formulario (nombre) e invocará al constructor de la clase Usuario creando un objeto añadiéndolo al array <i>usuarios</i>.</p>	
listarLibros	2
<p>Esta función devolverá un elemento (objeto) Tabla de HTML que recogerá todos los libros en formato tabla: En cada fila mostrará los siguientes datos:</p> <ul style="list-style-type: none"> • Título • Autor • ISBN • Referencia • Fecha de Préstamo (vacío si el libro no está prestado) • Fecha de Devolución (vacío si el libro no está prestado) • Usuario (vacío si el libro no está prestado) <p>Los libros que estén prestados presentarán a continuación de la fila el botón Devolver, que invocará al método devolver, y los que no estén prestados, el botón Prestar que invocará al método prestar. Sólo se mostrará el botón Prestar si hay usuarios en nuestra tabla de usuarios.</p>	
refrescarListado	1
<p>Nuestro desarrollo tendrá una variable global que se denominará tablaHTML. Estará inicializada a <i>undefined</i>.</p> <p>La función refrescarListado() se invocará cada vez que se realice un cambio en los datos de los objetos de la biblioteca.</p> <p>Si tablaHTML es undefined, quiere decir que nunca hemos mostrado nuestro listado.</p> <p>Si tablaHTML no es undefined, quiere decir que tenemos que refrescar el listado, por lo que antes de invocar a listarLibros para actualizar su valor, borraremos el elemento tabla del documento HTML.</p> <p>En cualquier caso, tablaHTML recogerá el valor (objeto HTML) de la función listarLibros y lo desplegaremos en nuestro documento HTML la tabla utilizando, append o insertAdjacentElement.</p>	
Método prestar	3
<p>Este método de la clase <i>Libro</i> se invocará al accionar el botón Prestar y mostrará en la línea del listado, en una lista desplegable, el nombre del usuario al que se le prestará el libro.</p> <p>Se mostrarán los botones confirmar y cancelar, para en su caso confirmar el préstamo o cancelar la operación.</p> <p>Si se confirma el préstamo, se asignará la fecha del día a la propiedad <i>fechaPrestamo</i>, y la propiedad <i>fechaDevolucion</i> contendrá la fecha del día + 10 días. La propiedad <i>usuario</i> recogerá el objeto usuario seleccionado (habrá que buscar el objeto en el array utilizando el nombre).</p> <p>Se refrescará el listado mediante la función <i>refrescarListado</i> para que se recojan las modificaciones.</p>	
Método devolver	0,5
<p>Este método de la clase <i>Libro</i> se invocará al accionar el botón Devolver.</p> <p>Una vez pulsado, se mostrarán los botones confirmar y cancelar.</p> <p>Al confirmar la devolución, se asignarán a undefined las propiedades <i>fechaPrestamo</i>, <i>fechaDevolucion</i> y <i>usuario</i></p> <p>Se refrescará el listado mediante la función <i>refrescarListado</i> para que se recojan las modificaciones.</p>	