

Obligatoriedad

Superar esta práctica es de carácter **obligatorio** para aprobar la asignatura. Es necesario obtener una calificación de al menos 4 puntos sobre 10 para superar la práctica.

Grupos

En grupos de (como máximo) 3 personas.

Parte obligatoria

1. Desarrollar un conjunto de clases Java que sirvan para almacenar en memoria y gestionar (parcialmente) datos sobre los huéspedes y las reservas de habitaciones en un hotel. Como mínimo deberán proporcionar la siguiente funcionalidad:
 - 1.1. Una lista o tabla de huéspedes que han hecho reservas o han tenido en el pasado alguna estancia en el hotel.
 - a) Por cada huésped, se almacenarán: nombre, apellidos, NIF (número de identificación fiscal), fecha de nacimiento, domicilio (dirección, localidad, código postal, provincia).
 - b) Además podrán almacenarse opcionalmente un número de teléfono fijo, un número de teléfono móvil y/o una dirección de email.
 - c) Se almacenaran solamente los datos de los huéspedes que han realizado la reserva de habitación, no los de sus acompañantes.
 - d) La lista deberá tener un constructor sin parámetros que meta en la lista unos huéspedes de ejemplo, que deberán incluir como mínimo un huésped llamado José Pérez Pérez con NIF 12345678A.
 - 1.2. Una lista o tabla de reservas de habitaciones (pasadas, presentes o futuras), con los siguientes datos: huésped que ha hecho la reserva, habitación reservada, fecha de entrada y fecha de salida.
 - a) No se considerarán reservas de grupos de habitaciones.
 - b) Tendrá un constructor sin parámetros que meta unas cuantas reservas de ejemplo, que deberán incluir como mínimo una reserva del huésped con NIF 12345678A para la habitación 101 con fecha de entrada el 1 de marzo del 2015 y fecha de salida el 2 de marzo del 2015.
 - 1.3. Pueden añadirse todas las clases, atributos, métodos, etc que se consideren necesarios para implementar las aplicaciones de los apartados siguientes.
2. Desarrollar una aplicación distribuida para que los empleados del hotel puedan consultar y gestionar las reservas, con las siguientes características:
 - 2.1. El servidor se implementará con un servlet desplegado sobre el servidor Web GlassFish. Este servidor:
 - a) Almacenará y gestionará los datos de huéspedes, habitaciones y reservas, llamando al iniciarse a los constructores de cada lista que meten los datos de ejemplo.
 - b) Proporcionará como un servicio Web tipo REST las siguientes operaciones:
 - Consultar huéspedes por NIF o por apellidos y nombre.
 - Añadir un huésped con sus datos.
 - Dado el NIF de un huésped, eliminar o modificar sus datos,
 - Dada una fecha, consultar una lista de las reservas que tienen esa fecha de entrada,

- Dada la fecha de entrada de una reserva y NIF del huésped, eliminar o modificar sus datos.
 - Añadir una reserva a la lista.
 - Se especificarán como parámetros de entrada:
 - El NIF del huésped. El servidor comprobará que el huésped está en la lista. En caso contrario no añadirá la reserva y devolverá un error.
 - Las fechas de entrada y de salida.
 - El servidor generará y asignará aleatoriamente un número de habitación entre 100 y 599. No es necesario que devuelva ese número ni que compruebe que la habitación no está ya reservada para esas fechas.
 - c) Para invocar los servicios de consulta de información se utilizará HTTP GET, y HTTP POST en el resto de los casos.
 - d) Todas las operaciones utilizarán la URL del servlet, añadiendo un parámetro para distinguir entre las diferentes operaciones.
 - e) Los parámetros de entrada de las operaciones se pasarán como parámetros de la forma nombre=valor (en la URL para los GET y en el cuerpo de la petición HTTP para los POST). Por ejemplo, al añadir un huésped se enviarán cada uno de sus atributos (NIF, nombre, apellido, etc) como un parámetro.
 - f) En las operaciones de consulta de información se devolverán los resultados aplanados o serializados a XML, con la biblioteca XStream.
 - g) Se utilizarán los códigos de estado HTTP para indicar el resultado de la operación (p.ej. 404 para indicar que no existe un huésped o una reserva con esos datos). Donde esto no sea suficiente se definirán y devolverán mensajes, también serializados a XML con XStream, con los elementos que se consideren necesarios.
- 2.2. El cliente será una aplicación Java nativa que permita al usuario invocar cualquiera de las operaciones del servicio tantas veces como desee y que invoque las respectivas operaciones en el servidor, utilizando la API de HTTP, y que muestre los resultados.
- a) El nombre de la clase principal del cliente deberá empezar por "Cliente".
 - b) Después de las operaciones de consulta se dará inmediatamente al usuario la opción de eliminar o modificar los datos de los resultados.
 - c) Si al intentar añadir una reserva el servidor detecta que no está el huésped en la lista se le dará al usuario la opción de añadirlo, introduciendo sus datos, y completar la reserva.
3. Desarrollar una aplicación Web compuesta de:
- 3.1. En el lado del cliente una o varias páginas y formularios HTML, estáticos o generados dinámicamente por el servidor, que permitan al usuario realizar tantas veces como desee las mismas operaciones que en la aplicación del apartado 2. En la memoria deberá especificarse la URL inicial de la aplicación.
 - 3.2. El servidor se implemente con un servlet desplegado sobre el servidor Web GlassFish. Los resultados de las operaciones (incluyendo mensajes que indiquen el éxito o el fracaso de una operación) se enviarán generando dinámicamente páginas HTML, que deberán incluir algún mecanismo (enlace, botón, etc) para facilitar que el usuario realice una nueva operación sin tener que pulsar el botón de volver a la página anterior del navegador. El nombre de la clase del servlet deberá empezar por "Servlet".
4. Escribir una memoria en formato pdf en la que se especifiquen, en la portada, los autores de la práctica, y se expliquen brevemente la organización en proyectos, aplicaciones, carpetas, paquetes, clases Java y documentos HTML y CSS que la componen, con un máximo de 2 líneas por cada elemento, y añadiendo donde sea apropiado figuras o esquemas que clarifiquen las relaciones entre elementos. . Además deberá documentarse adecuadamente el código añadiendo comentarios.

La práctica deberá desarrollarse utilizando el **entorno Netbeans** y deberá funcionar correctamente en **Windows 7, Netbeans 8.0, Oracle Java 1.8 y Google Chrome 40**

Partes opcionales

Adicionalmente, se valorará:

- A) La utilización de los elementos HTML específicos para el tipo de información a introducción o presentar (como address o date). Para convertir las fechas de y hacia el formato utilizado por HTML puede utilizarse la clase java.text.SimpleDateFormat.
 - B) Añadir hojas de estilo CSS para mejorar la estética y portabilidad de las páginas HTML.
- En la portada de la memoria deberán especificarse las opciones realizadas.

Criterios de calificación

Al evaluar la práctica se tendrán en cuenta, entre otros, los siguientes aspectos:

- La realización de todos los elementos especificados.
- El funcionamiento correcto de las aplicaciones según se especifica en el enunciado.
- La corrección de las páginas HTML .
- Que no se transmita innecesariamente información entre servidor y cliente (p. ej. en una consulta que se transmita solamente la información que cumple los criterios de la consulta)
- En la parte opcional A, la adecuación de los elementos utilizados.
- En la parte opcional B:
 - Que todos los elementos de estilo se especifiquen en las hojas CSS (en lugar de en las páginas HTML).
 - La estética del resultado.
 - El número y tipo de los aspectos de estilo especificados.

Fecha y modo de entrega

La fecha tope para entregar la práctica es el **miércoles 25 de marzo de 2015 a las 20 h.**

Las clases de teoría y ejercicios se reanudarán:

- **En el grupo de Vicávaro el lunes 16 de marzo.**
- **En el grupo de Móstoles el miércoles 18 de marzo.**

La práctica se entregará a través del Campus Virtual, utilizando el enlace proporcionado al efecto en la sección de Contenidos. Cada grupo entregará un fichero .zip con los archivos que tenga la práctica (memoria de la práctica y todos los archivos de(l/los) proyecto(s) que componen la práctica).

Se recomienda encarecidamente conservar una copia de la práctica entregada, ya que es frecuente olvidar entregar alguno de los archivos.