



---

Ingeniería Informática. Curso 2021-2022

---

# **Sistemas Distribuidos**

## **Enunciado Práctica 2.**

V1.0

---



**Versiones:**

Versión	Fecha	Detalle
V1.0	01/04/2022	Versión inicial

---

# 1.- Normativa y consideraciones previas.

## 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.

Dentro del porcentaje de prácticas de la asignatura (60% del total de la misma) la ponderación de esta segunda práctica corresponde al 40%

## Grupos

Por defecto se considerarán los mismos grupos de práctica que fueron formados para la primera práctica.

Sólo en el caso de querer modificar las configuraciones de grupos de la misma, se notificará al profesor, vía correo electrónico del aula-virtual la nueva configuración. En caso de querer realizar modificaciones de los grupos, se permitirán grupos de 2 alumnos, el resto de consideraciones a ellos se mantienen respecto a la primera práctica. Estos cambios de configuración podrán notificarse **hasta el día 17 de abril a las 23:59 h CET**.

# 2.- Enunciado

## 2.1.- Contexto del problema a resolver

Tras la implementación de la Aplicación de Gestión de **BiciURJC**, se desean continuar incluyendo funcionalidades para poder ser automatizadas, desde el propio software de las estaciones y de las bicicletas, para ello se procederá a la contratación de dos empresas independientes para elaborar:

- **API REST de Gestión de Usuarios**
- **API REST de Reserva de Bicicletas**

En el desarrollo de esta práctica se abordará el desarrollo de ambos módulos, simulando por completo la independencia entre ellos.

## 2.2.- Funcionalidades requeridas.

### 2.1.- API REST de Gestión de Usuarios

La aplicación de gestión deberá permitir gestionar los usuarios de BiciURJC en una base de datos H2.

El usuario vendrá definido por las siguientes propiedades

- Identificador único de usuario
- Login: Será un identificador único del usuario en el sistema.
- Contraseña de acceso al sistema (opcional todo el relativo a la codificación y visualización de la misma en los DTO)
- Nombre completo
- (opcional) Foto<sup>1</sup>
- Fecha de Alta en el sistema
- Estado: Activo, Inactivo.
- Saldo: Cantidad en € disponible en su monedero

Se solicita en el pliego las siguientes interfaces API REST:

- CRUD de Gestión de Usuarios:
  - Listado general
  - Búsqueda por identificador único de usuario
  - Insertar usuario.
  - Actualizar todos los parámetros, excepto el identificador único de usuario y la fecha de alta.
  - Borrar (pasará a estado Inactivo)
- Pago para reserva de bicicleta
  - Sólo podrá realizar el pago si el usuario se encuentra ACTIVO y tiene SALDO suficiente
  - Además del pago de la bicicleta, procederá a descontar del saldo del usuario el doble de la cantidad en concepto de fianza
- Devolución de la fianza
  - Deberá incrementar en el saldo del usuario el importe retenido durante la reserva

## 2.2.- API REST de Reserva de bicicletas

Este módulo considerará los conceptos de:

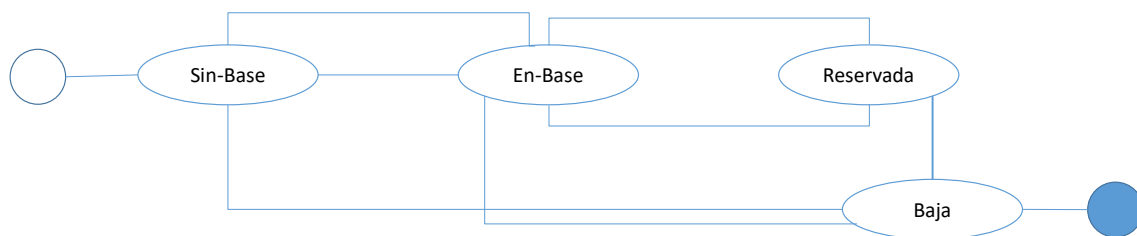
- Estación como el lugar donde se encuentran ancladas las bicicletas para que éstas puedan ser alquiladas y devueltas por los usuarios finales. Cada estación vendrá identificada, como mínimo, por:
  - Identificador único de estación
  - Número de serie.
  - Coordenadas: longitud y latitud donde se encuentra localizada la estación.
  - Capacidad de bicicletas, ya que habrá bases de 5 y 10 bicicletas
  - Estado: Activa / Inactiva

---

<sup>1</sup> Si se implementa el soporte para fotografías del usuario. Será obligatorio NO utilizar fotos de personas reales (en los datos generados de pruebas). Se pueden generar fotos aleatorias de personas que NO existen utilizando la web <https://this-person-does-not-exist.com/>

- Fecha de instalación.
- Bicicleta, caracterizada por:
  - Identificador único
  - Número de serie, será un código alfanumérico de 16 caracteres.
  - Modelo.
  - Fecha de alta en el sistema.
  - Estado en el que se encuentra

El ciclo de vida de una bicicleta en el sistema se representa por el siguiente diagrama de estados:



La representación de los estados será:

- Sin-Base: La bicicleta se ha dado de alta en el sistema, pero aún no se ha asignado estación
- En-Base: Se ha asociado la bicicleta a una estación con capacidad suficiente.
- Reservada: Un usuario está haciendo uso de la bicicleta
- Baja: La bicicleta ha dejado de estar operativa (aunque la Bicicleta se mantendrá en base de datos por razones históricas y de auditoría).

Se solicita en el pliego las siguientes interfaces API Rest con gestión sobre ellas sobre base de datos H2:

- Reserva de bicicleta
  - Se progresarán los parámetros: identificador de estación, identificador de bicicleta, identificador de usuario
  - Se deberá validar que la estación está ACTIVA, la bicicleta EN-BASE y en dicha estación
  - Se procederá a la invocación, en forma de cliente REST para proceder con el pago de la reserva y fianza al API REST del módulo de Gestión de Usuarios.
  - La bicicleta pasará a estado RESERVADA dejando un sitio libre en la estación.
- Devolución de bicicleta
  - Se progresarán los parámetros: identificador de estación, identificador de bicicleta, identificador de usuario
  - Se deberá validar que la estación está ACTIVA, la bicicleta RESERVADA y existe sitio en dicha estación

- Se procederá a la invocación, en forma de cliente REST para proceder con la devolución de la fianza al API REST del módulo de Gestión de Usuarios.
- La bicicleta pasará a estado EN-BASE en dicha estación

## 2.3.- Funcionalidades excluidas de esta práctica.

Destacar que:

- No será necesario realizar más funcionalidades que las marcadas en el enunciado.
- No debe generarse el código automáticamente con Swagger o similar, sino creando los servicios REST tal y como se ha visto en clase.

## 3.- Entrega de la práctica.

### 3.1.- ¿Qué debe incluir la práctica?

#### Código fuente y ejecución de la aplicación.

Implementar 2 proyectos Spring REST:

- Proyecto 1 – API REST Gestión de Usuarios
  - El API Rest deberá estar perfectamente documentado y seguirá el Nivel 2 del nivel de madurez de Richardson
  - Base de datos H2 con datasource y consola habilitada:  
`spring.datasource.url=jdbc:h2:mem:bdSDPractica2-Usuarios`
  - Servidor Spring-boot con Tomcat embebido
  - Documentación Swagger habilitada en la WEB
  - El API Rest debe de estar accesible en el **puerto 8081** del servidor donde se ejecute la práctica.
- Proyecto 2 – API Rest Reserva de Bicicletas
  - El API Rest deberá estar perfectamente documentado y seguirá el Nivel 2 del nivel de madurez de Richardson
  - Base de datos H2 con datasource y consola habilitada:  
`spring.datasource.url=jdbc:h2:mem:bdSDPractica2-Estaciones`
  - Servidor Spring-boot con Tomcat embebido
  - Documentación Swagger habilitada en la WEB
  - El API Rest debe de estar accesible en el **puerto 8080** del servidor donde se ejecute la práctica.

---

## Datos de precarga de la aplicación.

En la práctica que se entregue se utilizarán métodos anotados con `@PostConstruct` para cargar a través de un script en la base de datos un pequeño número de datos de ejemplo, que deberán incluir **como mínimo**:

- Proyecto 1 – API Rest Gestión de Usuarios:
  - 5 usuarios.
- Proyecto 2 – API Rest Reserva de Bicicletas:
  - 2 estaciones de bicicletas (una con capacidad de 5 bicicletas y otra de 10 bicicletas)
  - 9 bicicletas
  - En relación a las bicicletas anteriores:
    - 4 asignadas a la primera estación
    - 3 asignadas a la segunda estación
    - 2 sin asignar

## Breve memoria descriptiva.

Será necesario acompañar una breve memoria descriptiva de la solución implementada detallando:

- Arquitectura de la solución: controladores (`@RestController`), repositorios (`JpaRepository`), entidades (`@Entity`), servicios (`@Service`), etc...
- Modelo conceptual de la base de datos.

Además de la memoria descriptiva cada miembro deberá, de forma individual, a través de correo o mensajería del aula virtual un mensaje con un breve mensaje de en el que incluya una consideración personal sobre su forma de abordar/realizar la práctica: problemas encontrados, cómo se ha abordado la resolución, experiencia personal en el desarrollo de la práctica.

## 3.2.- Fecha y modo de entrega

Una persona exclusivamente de cada grupo, procederá a la entrega de la práctica a través del Aula Virtual, utilizando el enlace proporcionado al efecto en la sección de Evaluación. Cada grupo entregará DOS ficheros .zip (no rar ni otros formatos de compresión), uno por cada proyecto con:

- Todos los archivos del proyecto realizado con Spring STS. Se comprimirá el proyecto del workspace desarrollado incluyendo archivos de configuración, maven de la aplicación, fuentes, etc...
- En la raíz del proyecto, existirá un fichero “autores.txt” con los nombres y correos de los componentes del grupo

Se realizarán 2 sesiones de aclaración de dudas de la práctica los días:

Vicálvaro	Móstoles	
27/04/22	27/04/22	Práctica 1 – Día 1
04/05/22	04/05/22	Práctica 1 – Día 2

(\*) Fechas tentativas

**La fecha tope para entregar la práctica es el 22 de mayo de 2022 a las 23:59 CEST**

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

## 4.- Criterios de calificación

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

- Implementación de todos los requisitos solicitados.
  - La implementación de los requisitos marcados como opcionales se valorarán complementariamente a la puntuación máxima, pudiendo obtener hasta 0,5 puntos adicional por ellos.
- El funcionamiento correcto de la aplicación según se especifica en el enunciado.
- La corrección y adecuación de los elementos de todo tipo utilizados (clases Java, anotaciones, arquitectura, clases DTO, clases @Entity, etc...).
- Legibilidad y claridad del código fuente.
- Trazabilidad del código de acuerdo a los niveles de error: trace, debug, info, warn, error
- Detalle de la solución implementada en la memoria descriptiva de la solución.

**NOTA:**

*Si se considera necesario se realizará un examen oral a los componentes de los grupos para poder aclarar determinados aspectos de la implementación y el grado de participación de los alumnos.*