

Práctica y Trabajo Tutelado de Internet y Sistemas Distribuidos

3º Curso – Grado en Informática
Curso académico 2024-2025

1 Introducción

La práctica y los trabajos tutelados de la asignatura “Internet y Sistemas Distribuidos” consistirán en la aplicación de los conceptos y tecnologías aprendidos en la asignatura para el desarrollo de un servicio simplificado de gestión de inscripciones en cursos de formación.

Se desarrollará una aplicación que siga una arquitectura en capas como la estudiada en la asignatura, incluyendo la capa Acceso a Datos, la capa Lógica de Negocio, la capa Servicios, la capa Acceso a Servicios y la capa Interfaz de Usuario. La aplicación podrá ser invocada remotamente usando REST y, si se hace el trabajo tutelado, Apache Thrift. La capa Acceso a Datos utilizará una Base de Datos relacional para guardar la información pertinente.

El apartado 2 especifica la funcionalidad de la práctica. El apartado 3 especifica la funcionalidad del trabajo tutelado.

2 Práctica

2.1 Visión global

FicTraining se dedica a ofrecer a sus usuarios cursos de formación presenciales en diferentes ciudades.

Cuando un usuario quiere hacer un curso de formación, debe inscribirse en el mismo y efectuar el pago con una tarjeta bancaria. En el momento de realizar la inscripción se le asignará un identificador (que sirve como justificante de la inscripción), y se guardará el día y hora a la que se hizo la inscripción. Solamente se podrán hacer inscripciones en cursos aún no comenzados. Además, las inscripciones se podrán cancelar siempre que falten más de 7 días para el comienzo del curso.

El objetivo de la práctica es desarrollar una aplicación que dé soporte a algunos aspectos de la operación de *FicTraining*. Más concretamente, debe permitir dar de alta cursos, buscar cursos por localidad y fecha, buscar un curso por su identificador, inscribirse en un curso, cancelar una inscripción, y buscar todas las inscripciones de un usuario.

2.2 Funcionalidad de la Capa Modelo

En esta sección se proporciona más información sobre la funcionalidad que debe soportar la capa Modelo. Nótese que deben tratarse las condiciones de error que se desprenden de la descripción realizada en esta y en la sección anterior.

1. Para dar de alta un curso, se indicará la ciudad donde se celebra, su nombre, la fecha y hora de comienzo, el precio y el número máximo de plazas. Además, se guardará la fecha y hora en la que se ha dado de alta el curso. La fecha de comienzo del curso tendrá que ser, al menos, 15 días posterior al momento en que se da de alta.
2. Será posible buscar cursos que se celebren en una ciudad a partir de una fecha. Se devolverán únicamente los cursos que se celebren en esa ciudad y cuya fecha de inicio sea posterior a la indicada, ordenados por fecha de celebración. La información devuelta de los cursos incluirá, además del resto de datos, el número de plazas disponibles en ese momento.
3. Será posible buscar cursos por su identificador. Al igual que en el punto anterior, la información devuelta del curso incluirá, además del resto de datos, el número de plazas disponibles en ese momento.
4. Será posible que un usuario se inscriba en un curso siempre y cuando esté abierto aún su plazo de inscripciones (es decir, siempre que el curso aún no haya empezado). Además de otros parámetros que puedan ser necesarios, recibe como entrada un e-mail para identificar al usuario y un número de tarjeta bancaria para realizar el pago. En caso de ejecutarse con éxito, se almacena la inscripción, quedando registrada la fecha y hora a la que se hizo, y devuelve el identificador asignado a ella.
5. Será posible que un usuario cancele una inscripción siempre y cuando esté abierto aún su plazo de cancelación (es decir, siempre que falten más de 7 días para el comienzo del curso), y no se haya cancelado ya previamente. Recibe como entrada el identificador de la inscripción que desea cancelar y el email del usuario que está cancelando la inscripción (un usuario solamente puede cancelar sus inscripciones). En caso de ejecutarse con éxito, quedará registrada la fecha y hora a la que se canceló la inscripción.
6. Será posible obtener todas las inscripciones que un usuario ha realizado a lo largo del tiempo, ordenadas por fecha de creación de la inscripción (primero las más recientes). Deben devolverse todos los datos almacenados para cada inscripción.

IMPORTANTE: Para simplificar la implementación de la práctica, NO se guardarán datos de los usuarios del servicio. Sólo se guardará su e-mail al hacer una reserva.

2.3 Detalles de la Capa Servicios

La capa Servicios expondrá la funcionalidad de la capa modelo a las aplicaciones remotas usando un servicio web REST que trabajará con datos en formato JSON. Sin embargo:

- En la búsqueda de cursos se podrá especificar únicamente la ciudad y se devolverán todos los cursos que empiecen a partir de la fecha actual.
- Los datos de los cursos devueltos (e.g. como resultado de una búsqueda) no incluirán la fecha de alta del curso.
- Los datos de las inscripciones devueltas (e.g. como resultado de una búsqueda) incluirán solamente los últimos 4 dígitos de la tarjeta bancaria utilizada para el pago.

2.4 Detalles de la Aplicación Cliente

Para simplificar, en la práctica se desarrollará un único cliente que permitirá invocar todas las operaciones ofrecidas por la capa servicios (si bien, en un caso real habría diferentes clientes para los usuarios finales y para los administradores o empleados de *FicTrainings*).

En la búsqueda de cursos por ciudad y fechas, la aplicación mostrará el número de plazas reservadas en lugar del número de plazas disponibles.

Debe implementarse la capa acceso al servicio usando REST.

Aunque no se realice el trabajo tutelado de Apache Thrift (ver siguiente apartado), la arquitectura de la práctica deberá contemplar la posibilidad de que se desarrollasen en un futuro varias implementaciones de las capas Servicios y Acceso al Servicio, de forma que el cliente pudiese utilizar una capa de Acceso al Servicio u otra modificando simplemente un parámetro de configuración.

3 Trabajo tutelado

El trabajo tutelado es INDIVIDUAL y no es necesario hacerlo para poder aprobar la asignatura, aunque sí es necesario para optar a la nota máxima. Dentro de un mismo grupo de prácticas, cada alumno puede decidir individualmente si lo hace o no.

Como trabajo tutelado se propone implementar la capa Servicios usando un servicio Thrift. Además, los alumnos que realicen este trabajo tutelado, deberán implementar la capa Acceso al Servicio del cliente usando dicha tecnología. El cliente podrá cambiar de la versión REST a la versión Thrift del Servicio modificando simplemente un parámetro de configuración.

4 Normativa y evaluación

4.1 Composición de los grupos y trabajo a realizar por cada alumno

La práctica se realizará en grupos de 3 personas.

4.2 Distribución de las tareas de desarrollo de la práctica

Los miembros de un grupo serán responsables de dividir el desarrollo de la práctica según consideren que es más adecuado para su caso, teniendo en cuenta las siguientes normas:

- En cada una de las **iteraciones** de la práctica será necesario **implementar todas las funcionalidades pedidas**.
- **Cada miembro del grupo debe contribuir de forma significativa al desarrollo de cada iteración de la práctica. Concretamente, cada miembro debe haber contribuido de forma relevante al desarrollo de la capa Modelo (tanto DAOs, como Servicios y sus tests de integración), de la capa Servicios, de la capa Acceso a Servicios y de la capa Interfaz de Usuario del cliente de línea de comandos.**
 - Para determinar la contribución de cada alumno a la práctica, el profesor que la corrija **utilizará EXCLUSIVAMENTE el grafo de commits del repositorio Git** asociado (en caso de dudas podrá concertar una reunión con los miembros que estime necesario).
 - **Si un alumno no tiene commits significativos en alguna de las capas enumeradas, no superará la práctica.**
 - Aun considerándose suficientes los commits de todos los alumnos de un grupo, la nota de la práctica podría ser diferente en función del análisis del grafo de commits.
 - **Una posible forma de dividir el trabajo (aunque no obligatoria) es por funcionalidades.** Cada uno de los 3 miembros del grupo puede encargarse de implementar 2 de las 6 funcionalidades en cada capa. Pero dentro de un mismo grupo, también es posible que distintos miembros contribuyan al desarrollo de una misma funcionalidad si lo consideran oportuno.

- **Los mensajes de los commits deberán seguir OBLIGATORIAMENTE el formato que se explica a continuación:**
 - *[FUNC-X] Mensaje detallado.*
 - Para commits en los que se esté implementando una funcionalidad, siendo X el número que tiene asignado en el enunciado de la práctica esa funcionalidad (entre 1 y 6).
 - Por ejemplo:
 - [FUNC-1] Añadida e implementada la operación create en el DAO de Curso.
 - [FUNC-1] Implementado dar de alta Curso en servicio de la capa Modelo.
 - [FUNC-1] Añadida prueba de ejecución correcta para dar de alta Curso.
 - *[VARIOS] Mensaje detallado.*
 - Para commits de cosas comunes a varias funcionalidades (e.g. entidades del modelo, factorías, interfaces, esqueleto de una clase etc.).
 - Por ejemplo:
 - [VARIOS] Añadida entidad Curso.
 - [VARIOS] Añadida interfaz para el DAO de Curso.
 - [VARIOS] Añadida factoría para el DAO de Curso.
 - [VARIOS] Añadido conversor entre DTO y entidad Curso.

4.3 Distribución de las tareas de desarrollo del trabajo tutelado

Dado que el trabajo tutelado es opcional e individual, cada miembro que decida hacer el trabajo tutelado deberá escoger un grupo de dos funcionalidades de entre los que se enumeran a continuación, e implementar con Apache Thrift las partes de las capas Servicios y Acceso a Servicios correspondientes a esas funcionalidades. En caso de realizar el trabajo tutelado más de un miembro de un mismo grupo, deben ponerse de acuerdo para elegir diferente grupo de funcionalidades.

- Grupo 1: Funcionalidades 1 y 2.
- Grupo 2: Funcionalidades 3 y 4.
- Grupo 3: Funcionalidades 5 y 6.

Todos los commits relativos a un grupo de funcionalidades deben ser realizados por el alumno que haya elegido ese grupo. **Los mensajes de los commits deben seguir el formato:**

[TT-X] Mensaje detallado.

Siendo X el grupo de funcionalidades elegido. Por ejemplo:

- [TT-1] Añadidas operaciones al IDL
- [TT-1] Implementado dar de alta Curso en la capa Servicios.

4.4 Iteraciones y entregas

Para la realización de la aplicación se seguirá un enfoque basado en iteraciones, de manera que cada iteración incorpora más funcionalidad sobre la anterior, hasta que en la última iteración se termina con un software que implementa toda la funcionalidad. En particular, la aplicación se hará en tres iteraciones.

- **[OBLIGATORIA] Primera iteración.** Se entregará la capa Modelo de la práctica (incluidas sus pruebas de integración). **Plazo de entrega: domingo 10 de noviembre.** La corrección de esta iteración se realizará durante la semana del 11 de noviembre y **NO será presencial.** El profesor enviará un **correo a los miembros de cada grupo con los problemas que haya identificado** al analizar el código de su práctica y que **puede ser necesario corregir para poder implementar correctamente la segunda iteración.**
- **[OBLIGATORIA] Segunda iteración.** Se completará la práctica con la implementación del servicio web REST que permite el acceso remoto a la capa Modelo, así como el cliente de

línea de comandos necesario para acceder al mismo. Opcionalmente, se entregará el trabajo tutelado. **Plazo de entrega: domingo 15 de diciembre.** La corrección de esta iteración se realizará durante la semana del 16 de diciembre y **NO será presencial.**

- **Tercera iteración.** En esta iteración se permite entregar el trabajo tutelado en caso de no haberlo entregado en la segunda iteración. **Plazo de entrega: domingo 26 de enero.** La corrección de esta iteración se realizará una vez finalizado el plazo de entrega y **NO será presencial.**

Para la entrega de cada iteración, se utilizará el repositorio Git asignado a cada grupo. Se deben subir al repositorio **sólo los ficheros fuente** (e.g. .java, pom.xml, ficheros de configuración, etc.), **y no los ficheros objeto** (e.g. .class, .war, etc.). **Antes de la entrega de cada iteración se proporcionarán instrucciones detalladas sobre cómo realizar la entrega.**

4.5 Evaluación

La práctica y los trabajos tutelados se evaluarán de la siguiente manera:

- **Práctica.** Cada una de las dos primeras iteraciones se puntúan **de 0 a 5**, siendo la nota final de la práctica la **suma de ambas**. Tal y como especifican las normas de evaluación de la asignatura, se recuerda que en la evaluación de cada una de las dos iteraciones es necesario obtener un **mínimo de 2,5 puntos** y en caso de no alcanzarse esa nota mínima en alguna de ellas, la nota máxima de la práctica será un 4. Además, para poder presentar la segunda iteración, es necesario haber obtenido la nota mínima en la primera iteración. Dentro de un grupo, la nota puede ser diferente para cada miembro del grupo en función de su contribución y la calidad del diseño e implementación de la parte que haya hecho.
- **Trabajos tutelados.** Puntuación: **de 0 a 10 puntos**. En este caso no es preciso obtener ninguna nota mínima.

Para la **corrección de cada iteración** el profesor descargará el código del repositorio Git y evaluará su correcto funcionamiento, la calidad del diseño y la calidad del código.

Una práctica copiada significará un suspenso para el grupo que ha dejado copiar y para el que ha copiado; a todos los efectos, no se hará ninguna distinción. Los suspensos por práctica copiada tendrán que realizar una práctica distinta, que además deberán proponer (y ser aceptada).

Si alguno de los miembros de un grupo no alcanza la nota mínima en la segunda iteración pero el resto sí lo hace, el alumno suspenso (o no presentado) deberá desarrollar en solitario en la segunda oportunidad una versión extendida de la práctica con funcionalidades adicionales.

En la convocatoria correspondiente a la segunda oportunidad se presentará la misma práctica y trabajo tutelado (excepto los suspensos en alguno de los dos casos anteriores), sin posibilidad de entregar la primera iteración: se presentará directamente la versión final de la práctica y del trabajo tutelado.