

FACULTADE DE INFORMÁTICA, UNIVERSIDADE DA CORUÑA

Trabajo Tutelado ASI

ARQUITECTURA DE LOS SISTEMAS DE INFORMACIÓN

Alejandro Cortiñas – alejandro.cortinas@udc.es

23 de octubre de 2025

1 DESCRIPCIÓN

El objetivo del trabajo tutelado es el diseño y desarrollo de una aplicación web. El trabajo se dividirá en tres iteraciones, descritas en este documento. El dominio de la aplicación será uno de las siguientes:

1. Aplicación de mensajería instantánea con grupos.
2. Aplicación de seguimiento de hábitos diarios.
3. Aplicación para compartir rutas y experiencias de viaje.
4. Aplicación para crear y compartir listas de reproducción.
5. Aplicación para llevar el control de gastos personales.

6. Aplicación para organizar eventos y enviar invitaciones.
7. Gestión de reservas para un restaurante.
8. Gestor de proyectos con seguimiento de hitos.
9. Plataforma de alquiler de objetos entre particulares.
10. Plataforma de alquiler vacacional entre particulares.
11. Plataforma de aprendizaje con cursos y evaluaciones.
12. Plataforma de compraventa de libros usados.
13. Plataforma de crowdfunding para proyectos creativos.
14. Plataforma de donaciones para causas sociales.
15. Plataforma de mentoría académica entre estudiantes.
16. Portal de noticias personalizadas por intereses.
17. Red social para compartir recetas de cocina.
18. Sistema de control de asistencia para clases.
19. Sistema de gestión de academias de música.
20. Sistema de gestión de clínicas veterinarias.
21. Sistema de gestión de congresos académicos.
22. Sistema de gestión de exposiciones en museos.
23. Sistema de gestión de festivales de música
24. Sistema de gestión de inventario para pequeños negocios.
25. Sistema de gestión de tareas colaborativas.
26. Sistema de reservas para instalaciones deportivas.
27. Sistema de seguimiento de pedidos para una tienda online.
28. Sistema de votación online para decisiones grupales.

Solo puede haber un grupo con cada dominio, asignados por orden de entrega de la primera iteración.

1.1 PRIMERA ITERACIÓN

La primera iteración consiste en identificar las parejas que realizarán el trabajo y el dominio elegido por cada pareja. En el campus virtual se ha publicado un elemento de “Elección del Grupo - Trabajo Tutelado” donde hay 28 grupos, uno por dominio. Cada pareja elegirá el grupo en función del dominio que quiera. Es decir, si queréis el dominio *Portal de noticias personalizadas por intereses*, los dos componentes de la pareja deberíais elegir el grupo P16.

El alumnado que tenga *dispensa académica* o que esté matriculado de la *oportunidad adelantada* realizará este trabajo de forma individual, y debe elegir igualmente un grupo/dominio.

1.2 SEGUNDA ITERACIÓN

En esta iteración se realizará tanto el análisis como el diseño de la aplicación a desarrollar. Este diseño debe tener en cuenta una serie de restricciones relacionadas con la complejidad del trabajo:

- Más o menos, la aplicación debe de ser “el triple de grande” que la aplicación de la primera práctica de la asignatura¹. Si es necesario, el profesorado llevará a cabo ajustes sobre el alcance en la revisión de esta iteración.
- La arquitectura debe incluir una de estas dos opciones:
 - Debe integrarse algún servicio web externo existente, encapsulando el código que se comunica con ese servicio en un microservicio separado del backend principal. Por ejemplo, si nuestro dominio fuera “Plataforma de reseñas y puntuaciones de películas”, podríamos usar *The Movie Database* a través de un microservicio *TMDBClient*.
 - Ídem, pero creamos el servicio web externo como parte del trabajo tutelado. Esta alternativa es para el caso de que no se encuentre ningún servicio web externo que resulte conveniente para el dominio. Por ejemplo, si no existiera *The Movie Database*, podríamos

¹Si se realiza de forma individual, debería ser “algo más compleja” que la primera práctica de la asignatura

implementar un pequeño servicio con la información que nos interesaría, *My Movie Database*, y usarlos a través de un microservicio *MMDBClient*.

El primer paso de esta iteración consiste en describir el dominio sobre el que se realizará el trabajo, definiendo los requisitos generales de la aplicación a desarrollar. Esto se plasmará en una **breve descripción textual**, que debería ocupar en torno a una carilla.

A continuación, realizaremos el análisis de los requisitos funcionales de la aplicación, así como el diseño de las vistas y del modelo de datos. Para documentar este análisis y diseño se presentarán, además de la descripción textual ya mencionada, los siguientes artefactos:

- Los requisitos los reflejaremos con un listado de **historias de usuario**. Cada historia de usuario debe ir identificada con una etiqueta (HUXX), un nombre, y una descripción.
- El modelo de datos que da soporte a los requisitos identificados lo representaremos de dos formas diferentes y equivalentes:
 - **Modelo entidad relación**.
 - **Diagrama de clases UML** (únicamente del modelo de datos, no es necesario incluir detalles de “comportamiento”).
- El diseño de las vistas se mostrará mediante los **prototipos de pantalla** más representativos. Para cada prototipo:, **identificar todas las historias de usuario** que aparecen en el mismo, usando para ello las etiquetas de las mismas.
- La arquitectura la mostraremos mediante un **diagrama de componentes UML**.
- El API del servicio web externo (existente o creado por nosotros) debe documentarse detalladamente con el modelo de documentación que se considere adecuado.

Es muy importante que se mantenga la congruencia entre los diferentes artefactos entregados. Por ejemplo, si en el diagrama de clases que define la entidad *Película* aparece una propiedad *país de producción*, en

el diseño de la pantalla que sirve para crear una película debe aparecer un campo análogo. Se penalizará mucho la falta de congruencia.

Esta segunda iteración será revisada y modificada tantas veces como sea necesario hasta llegar a una documentación completa y detallada de la aplicación a implementar. **Cada modificación realizada sobre la documentación debe indicarse explícitamente; es decir, debe haber una “lista de cambios” para cada versión del análisis realizado.** No es necesario esperar a tener la documentación completa para empezar a implementar, todo lo contrario, es recomendable ir implementando las HU que estén ya revisadas y dadas por válidas.

1.3 TERCERA ITERACIÓN

Se implementará la aplicación siguiendo el diseño creado en la segunda iteración, y cumpliendo las siguientes restricciones:

- El trabajo se desarrollará utilizando las tecnologías tratadas a lo largo de la asignatura y a partir de los esqueletos base proporcionados para la primera práctica de la asignatura.
- La aplicación debe poder ejecutarse con Docker sin tener nada más instalado en el ordenador, lanzando todos los servicios que sean necesarios. Esto incluye el servicio web externo si se ha creado como parte del proyecto (ver sección 1.2).
- Deben usarse bases de datos PostgreSQL, no H2.
- Cualquier cambio realizado sobre el diseño inicial debe identificarse y describirse, incluyendo la motivación para el cambio. Estos cambios, de no ser consensuados con el profesorado de la materia, pueden valorarse negativamente si no se consideran justificados.

2 NORMAS PARA EL DESARROLLO DEL TRABAJO TUTELADO

- *Grupos de prácticas:* el trabajo se realizará en grupos de dos personas. Aunque el trabajo se reparta, **los dos componentes del grupo**

tienen que conocer y ser capaces de entender e implementar cualquier parte del proyecto realizado, especialmente de cara a la prueba práctica individual.

■ *Entrega:*

- **Iteración 1.** Se seleccionará el grupo usando la tarea del Campus Virtual para ello. El límite de esta iteración es el **29 de octubre**, pero evidentemente al seleccionar un grupo/dominio se da por finalizada la iteración.
- **Iteración 2.** Se entregará un fichero PDF con toda la documentación en una tarea del Campus Virtual. El límite de esta iteración es el **9 de noviembre**.
- **Iteración 3.** La aplicación implementada se entregará como un *tag* de un repositorio de Gitlab.com (<https://gitlab.com/>), al que los usuarios “acortinas” y “gdebernardo” deben tener acceso de “desarrollador/developer” como mínimo para poder acceder al repositorio. En la carpeta raíz del repositorio debe estar el fichero “compose.yaml”, de forma que el proyecto se pueda ejecutar directamente con el comando “docker compose up” desde ese directorio. En la tarea del Campus Virtual se indicará tanto la URL del repositorio como el *tag* en cuestión. Además, en la tarea del Campus Virtual se entregará también la versión final de la documentación. Recordad que los cambios sobre la versión original de la documentación entregada en la segunda iteración deben venir documentados. El límite de esta iteración es el **27 de diciembre**.

- *Revisión y evaluación:* la segunda iteración se revisará en las clases de prácticas. Teniendo en cuenta el número de trabajos, es bastante probable que cada trabajo tenga que ser revisado en el grupo de docencia de la pareja. Debido a esto, las dos personas que formen una pareja deberían estar asignados al mismo grupo de docencia².

Para la tercera iteración se realizará una prueba práctica individual en la que el/la alumno/a tendrá que responder una serie de preguntas sobre el trabajo tutelado, e implementar pequeños cambios o funcionalidades sobre el mismo. Se darán más detalles sobre la realización de esta prueba más adelante, en clase y en la tarea de la entrega en el Campus Virtual.

²En caso contrario, se puede intercambiar el grupo con otro/a alumno/a y avisar al profesorado para tenerlo en cuenta.

- *Criterios de evaluación:* la evaluación del trabajo se basara en la correcta aplicación de los conceptos explicados en la asignatura, tanto en el diseño como en la implementación de la aplicación. Así, se debe prestar atención a aspectos como la arquitectura y diseño de la aplicación, uso correcto de las tecnologías utilizadas, calidad de la implementación (estructura del código, claridad, documentación, errores, etc.) y especialmente el correcto funcionamiento de la aplicación, valorándose negativamente los errores visibles por el usuario final (por ejemplo, un **SQL Exception** visible directamente por el usuario en vez de mostrarle un mensaje amigable).

Se valorará también el acabado final de la aplicación desde el punto de vista de la interfaz de usuario. La aplicación debe tener una interfaz adecuada a sus características, y no ser solo una copia/adaptación de los ejemplos utilizados en las clases de prácticas.

Durante la prueba práctica se valorarán los conocimientos del alumno/a sobre el código desarrollado, la arquitectura definida y la integración de los diferentes componentes de la misma.