

Práctica 4

Desarrollo de una aplicación multiplataforma

4.1. Programación y objetivos

Esta práctica constará de cuatro sesiones. Tendrá una puntuación en la nota final de prácticas de 4 puntos sobre 10.

4.1.1. Objetivos generales de la práctica 4

1. Llevar a cabo el análisis, diseño, implementación y prueba de una aplicación multiplataforma
2. Demostrar capacidad para generar código automático (desarrollo dirigido por modelos)
3. Cumplir las especificaciones en base a la realización de una descripción arquitectónica (DA) reducida
4. Adquirir destrezas en el autoaprendizaje de nuevas IDEs o entornos de trabajo o el uso de las ya conocidas para desarrollar aplicaciones Web
5. Incorporar al sistema diversos componentes software externos

4.1.2. Planificación y tareas específicas

A diferencia de las prácticas anteriores, el contenido concreto de esta práctica es ideado por cada grupo pequeño de teoría. Será cada equipo el que decida el sistema multiplataforma que

va a desarrollar. Se aconseja que sea de un tema que os motive y os resulte interesante. Por ejemplo, un gestor de partidas de rol, un videojuego de navegador, una pequeña red social de películas o deportes, un bot que recoja información de twitter y muestre información, etc. El profesor guiará al equipo para garantizar que el sistema elegido tenga las características necesarias para cumplir con los objetivos de la práctica y que su implementación completa sea factible durante el tiempo de duración de las prácticas (considerando 4 horas de trabajo fuera de clase por cada sesión de 2 horas de prácticas). En la Sección 4.3 se dan dos propuestas posibles para que los estudiantes se hagan una idea de la envergadura pedida a los proyectos.

Algunos de sus requisitos son:

- El sistema debe incluir una app en Flutter y una aplicación web con Ruby on Rails.
- Una vez acordados los requisitos funcionales y antes de pasar al diseño de alto nivel (descripción arquitectónica), el grupo se dividirá en dos equipos de desarrollo (uno para la app y otro para la aplicación web) que formarán sendos equipos de prueba para la parte no desarrollada por ellos.

Planificación

La planificación dependerá del modelo del ciclo de vida que se elija. Aquí se da una sugerencia para el modelo en V.

Sesión	Semana	Tareas
S1	30 abril-5 mayo	Análisis y diseño del sistema y las pruebas de sistema
S2	7-12 mayo	Diseño detallado y pruebas de integración y unidad e inicio de la codificación
S3	14-19 mayo	Codificación
S4	21-26 mayo	Depuración, pruebas, puesta a punto y despliegue

Una vez entregada la práctica, se dedicarán las siguientes sesiones a exponerlas, de la siguiente forma:

Tareas específicas

Se documentarán las siguientes tareas a realizar:

1. Fase de análisis: Se plasmará el análisis en una DA somera que incluya:

¹Por haber terminado el período docente el 9 de junio, los estudiantes del grupo DS2 realizarán esta sesión de evaluación concertando una tutoría en esta semana (incluido el sábado día 12).

Sesión	Semana	Tareas
SE1	28 mayo-2 junio	Exposición de la práctica al resto de compañeros del grupo grande de prácticas y realización de una encuesta personal de valoración por parte de cada estudiante del grupo grande
SE2	8-12 junio ¹	Evaluación de la práctica por parte del profesor

- Requisitos funcionales (se puede incluir un diagrama de casos de uso) y no funcionales
 - Listado somero de partes interesadas y preocupaciones de cada una
 - Diagrama que muestre la arquitectura del sistema desde el punto de vista del contexto (diagrama de componentes, diagrama de paquetes, ...)
 - Listado de criterios de calidad (perspectivas) a partir de los requisitos no funcionales que cumplan con las expectativas de los distintos interesados en el sistema, la normativa legal y los criterios de ética profesional que el equipo de desarrollo considere de importante cumplimiento
 - Diseño de pruebas del sistema e integración: condiciones de prueba y casos de prueba
2. Fase de diseño: Se deberá realizar al menos un diagrama de clases de diseño (o varios) de forma que se automatice todo lo posible la producción de código. Se realizará el diseño de las pruebas de unidad: condiciones de prueba y casos de prueba
 3. Codificación: Debe generarse todo el código que sea posible a partir de los diagramas de clase
 4. Prueba: Se deben realizar todas las pruebas necesarias para que la aplicación sea completamente operativa

Salvo el código, que se subirá aparte, la documentación de todas las demás tareas será incluida en un único pdf.

4.2. Criterios de evaluación

1. Capacidad demostrada de trabajo en equipo (reparto equitativo de tareas) y uso de sistemas de gestión proyectos y tareas (como Trello), sistemas de control de versiones (como Git o SVN), etc.

2. Reutilización de métodos (ausencia de código redundante)
3. Validez (se cumple con la intencionalidad de las partes interesadas)
4. Cumplimiento con los requisitos no funcionales
5. Éxito de las pruebas

4.3. Ejemplos de propuestas

Aunque se recomienda que cada grupo pequeño elija libremente la aplicación que quiere desarrollar, se proponen como ejemplos las siguientes dos opciones:

1. Desarrollo de una aplicación multiplataforma que pueda ser usado por farmacias y sus clientes
2. Aplicación Web RESTful receptiva y adaptable a diferentes dispositivos para visitar un museo

Una descripción de los requisitos funcionales del primer ejemplo consiste en considerar la implementación de un sistema de gestión de pedidos (reserva) de medicamentos por parte de un consorcio de farmacias, con las siguientes dos partes:

1. Para los consumidores: App que interaccione con un servidor REST desplegado en Internet para que una persona pueda:
 - Acceder al catálogo de medicamentos en el almacén del consorcio, con la descripción de cada medicamento
 - Localizar en un mapa las farmacias geográficamente más cercanas a su ubicación actual, elegir una y reservar² medicamentos con su móvil, usando un “carrito de la compra” virtual
2. Para las farmacias: Aplicación Web que permita al farmacéutico, o a un distribuidor de medicamentos autorizado, acceder al catálogo de productos farmacéuticos para dar de alta/baja o modificar la descripción de un medicamento en dicho catálogo. Esta aplicación también incluirá un enlace para descargarse el app Android desde el sitio Web donde esté ubicada. La interfaz del servidor con sus clientes ha de desarrollarse utilizando tecnología REST

²No se realizará el pago, sólo una reserva.

Debe tenerse en cuenta que en esta propuesta sólo se describen informalmente los requisitos funcionales de la aplicación, sin decirse nada de los requisitos no funcionales (salvo los mínimos obligatorios) ni, más concretamente, de los criterios de calidad. Constituye una parte importante en la evaluación de esta práctica la exhaustividad en la identificación de requisitos no funcionales y, en especial, criterios de calidad, así como la implementación de las medidas necesarias para cumplirlos.

4.4. Plazos de entrega y presentación de la práctica

Esta práctica será subida a PRADO en dos tareas:

1. Entrega del trabajo de la primera semana de la práctica: un pdf con el análisis y el diseño. El plazo terminará justo antes del inicio de la segunda sesión de esta práctica (a las 15:30 h. del día de la segunda sesión).
2. Entrega del resto de la práctica. El plazo terminará justo antes del inicio de la primera sesión de exposición de la misma (a las 15:30 horas del día de la primera sesión de exposición). Se deberán subir dos ficheros separados:
 - a) Fichero pdf con una memoria del desarrollo de la práctica (pasos seguidos, software utilizado, errores encontrados, dificultades más importantes, etc.).
 - b) Archivo con todo el software desarrollado.