

Práctica 3

Desarrollo de una aplicación multiplataforma

3.1. COVID-19: Cambios relacionados con las prácticas, por las exigencias de virtualización

El número de prácticas se reduce a 3, quitándose la práctica 4, que valía 1,5 puntos. De esta forma, la puntuación total será de 8,5 puntos (luego se prorrateará de forma adecuada, para considerar que el peso de las prácticas en la asignatura es del 50 %), con la siguiente distribución:

- P1: 3 puntos sobre 8,5
- P2: 1,5 puntos sobre 8,5
- P3: 4 puntos sobre 8,5

3.2. Programación y objetivos

Esta práctica constará de cuatro sesiones. Tendrá una puntuación en la nota final de prácticas de 4 puntos sobre 8,5.

3.2.1. Objetivos generales de la práctica 3

1. Llevar a cabo el análisis, diseño e implementación completos 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

3.2.2. Planificación y tareas específicas

A diferencia de las prácticas anteriores, el contenido concreto de esta práctica es ideado por cada equipo de desarrollo (grupo pequeño). 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 además 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). Para cumplir con lo especificado en la guía de la asignatura, también es posible desarrollar uno de dos sistemas que se especifican en la Sección 3.4 de este guión.

Planificación

Sesión	Semana	Tareas
S1	20-23 abril	Análisis y diseño del sistema
S2	27-30 abril	Aprendizaje de frameworks y/o IDEs que se consideren necesarios e inicio de la codificación
S3	4-7 mayo	Codificación
S4	11-14 mayo	Depuración, puesta a punto y despliegue

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

Sesión	Semana	Tareas
SE1	18-21 mayo	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	25-28 mayo ¹	Evaluación de la práctica por parte del profesor

Tareas específicas

1. Fase de análisis: Se plasmará el análisis en una DA somera que incluya:
 - 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
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
3. Autoaprendizaje: Se recomienda
 - profundizar en el aprendizaje de Android Studio o aprender alguna IDE para iOS
 - aprender alguna tecnología para desarrollo de aplicaciones Web. Por ejemplo, para Java, servlets y Java Server Pages (JSP) ²
 - instalar un servidor web, un servidor de aplicaciones Web y un SGBD. Por ejemplo, se puede instalar el servidor Web Apache, el servidor de aplicaciones Apache Tomcat y el SGBD MySQL. Netbeans y Eclipse permiten su gestión dentro del desarrollo de aplicaciones Web
 - aprender a usar un entorno de trabajo para generar y probar aplicaciones Web (por ejemplo, Ruby on Rails si se implementan las aplicaciones en Ruby), o profundizar en el uso de los ya conocidas (NetBeans o Eclipse, si se implementan dichas aplicaciones en Java)
 - OPCIONAL: implementar el patrón arquitectónico REST (Representational State Transfer) para crear los servicios web, valiéndose de algún framework que reduzca el trabajo de codificación:
 - Existe una API –Java API for RESTful Web Services (JAX-RS)– que se define sobre los servlets, y un framework, Jersey, que implementa el estándar JAX-RS
 - Spring: otro framework con su propia implementación del patrón REST

¹Por haber terminado el período docente el 27 de mayo, los estudiantes del grupo DS2 realizarán esta sesión de evaluación concertando una tutoría.

²Un ejemplo de tutorial en internet, es <https://www3.ntu.edu.sg/home/ehchua/programming/java/JavaServlets.html>.

- OPCIONAL: se tendrá en cuenta el aprendizaje y uso de cualquier otro software de ayuda al desarrollo de la práctica que el grupo pequeño considere
- 4. Codificación: debe generarse todo el código que sea posible a partir de los diagramas de clase
- 5. Prueba: se deben realizar todas las pruebas necesarias para que la aplicación sea completamente operativa

3.3. 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 GitHub o SVN), etc.
2. Implementación completa y verificabilidad (sin errores de ejecución)
3. Reutilización de métodos (ausencia de código redundante)
4. Validez (se cumplen los requisitos funcionales)
5. Calidad (se cumplen los requisitos no funcionales)
6. Claridad y completitud de la DA y del diseño subsiguiente en cuanto a los requisitos funcionales (punto de vista de contexto) y diagramas de diseño
7. Claridad y completitud de la DA en cuanto a los requisitos no funcionales (perspectivas)
8. Autoaprendizaje realizado de herramientas, marcos de trabajo, IDEs, etc.

3.4. Propuestas opcionales

Aunque se recomienda que cada grupo pequeño elija libremente la aplicación que quiere desarrollar, también puede escoger entre una de las siguientes dos opciones:

1. Desarrollo de una aplicación multiplataforma para la virtualización del comercio de oficinas de farmacia.- Se trata de un sistema de gestión de pedidos (reserva) de medicamentos, proporcionado por un consorcio de farmacias a sus farmacias y a los consumidores, con dos grandes subsistemas:
 - a) 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 reservar³ medicamentos con su móvil, usando un “carrito de la compra” virtual
- b) 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
2. Aplicación Web RESTful receptiva y adaptable a diferentes dispositivos para visitar un museo

Debe tenerse en cuenta que en estas propuestas sólo se describen informalmente los requisitos funcionales de la aplicación que se pide, sin decirse nada de los requisitos no funcionales 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.

3.5. 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, lenguajes de programación usados, errores encontrados, dificultades más importantes, etc.).
 - b) Archivo con todo el software desarrollado.

³No se realizará el pago, sólo una reserva.