Ingeniería del Software

PEC 1: Introducción a la Ingeniería del Software y la Orientación a Objetos

Presentación

Esta PEC es una introducción a la Ingeniería del Software y al paradigma de Orientación a Objetos. La actividad cubre los contenidos estudiados en los módulos 1 y 2 de la asignatura.

En el calendario de vuestra aula tenéis una descripción completa de las competencias y objetivos que esta PEC incluye.

Nota previa

Recordad que, tal como se explica en el Plan docente de la asignatura, esta actividad debe resolverse de forma **individual**. En caso de detectar copias se penalizará la actividad con una D como nota.

Para garantizar que has resuelto esta actividad de forma individual, te pedimos que en tu solución entregada añadas el siguiente texto:

"*Certifico que he realizado la PEC1 de forma completamente individual y solo con la ayuda que el profesorado de esta asignatura considera oportuna según las FAQs sobre plagio.*"

Compromiso de autoresponsabilidad

Escribe aquí el texto con el compromiso de autoresponsabilidad que consta entrecomillado en la **nota previa. No escribirlo implica la no corrección de la PEC y una calificación de D.**

Preguntas del Módulo 1

Pregunta 1 (10% puntuación)

Una de las actividades de la ingeniería del software es la gestión del proyecto. Dentro de la actividad de gestión del proyecto se deben llevar a cabo una serie de tareas pero además es fundamental equilibrar tres variables o restricciones (*constraints*).

Indicad cuáles son estas tres variables o restricciones, añadiendo una breve descripción (no más de 1-2 líneas para cada una) y poned un ejemplo donde se vea claramente que un cambio en cualquiera de las tres variables implica la modificación de, como mínimo, una de las otras dos.

Solución

Las tres variables a equilibrar en la actividad de gestión del proyecto son:

- *El alcance del proyecto: Es todo aquello se debe incluir en el proyecto (y lo que no).*
- *El tiempo: Cuánto tiempo debe destinar a la realización o ejecución del proyecto.*
- *El coste: Cuántos recursos, humanos o materiales, serán necesarios para llevar a cabo el proyecto.*

Un ejemplo de variación en las variables anteriores puede ser un cambio en el alcance del proyecto, solicitando más funcionalidades de las inicialmente previstas. Esto se traduce en que, para abordar estas nuevas funcionalidades será necesario aumentar el tiempo y/o los recursos destinados a la ejecución del proyecto: al tener que implementar más funcionalidades de las inicialmente previstas será necesario aumentar el número de recursos destinados al proyecto para poder terminarlo en el tiempo inicialmente previsto o bien alargarlo en el tiempo manteniendo el mismo número de

recursos, lo que a su vez podría implicar un aumento en el coste al necesitar usar esos recursos más tiempo de lo inicialmente previsto destinados al proyecto.

Otro ejemplo podría ser que se necesitara que el proyecto se ejecute en menos tiempo del inicialmente previsto. En este caso sería necesario aumentar el número de recursos (coste) para poder afrontar el desarrollo de las mismas funcionalidades en menos tiempo o bien reducir el alcance del proyecto para que, manteniendo los recursos, se pueda ejecutar el proyecto en menos tiempo a costa de reducir funcionalidades.

Pregunta 2 (15% puntuación)

En los apuntes del Módulo 1 se han introducido tres métodos de desarrollo de software:

- Ciclo de vida clásico o en cascada
- Ciclo de vida iterativo e incremental
- Desarrollo lean y ágil

Imagina que eres el/la jefe/a de proyecto del departamento de desarrollo de software de una compañía, por lo que tienes la responsabilidad de estudiar y escoger el mejor método de desarrollo de software para tres nuevos proyectos que están a punto de dar comienzo. Los tres proyectos presentan características diferentes, por lo que resulta crucial la elección del método más adecuado en cada caso. Para cada uno de ellos, se solicita hacer lo siguiente:

- Indica el tipo de proyecto que mejor se adecue a cada uno de los tres nuevos proyectos según la tabla del apartado 3.2 del Módulo 1 (Objetivo claro/no claro vs Solución conocida/no conocida).
- Identifica y **justifica** brevemente qué método de desarrollo escogerías para desarrollar el proyecto.

Proyecto 1

Actualmente, la empresa dispone de una aplicación web para que cada empleado pueda imputar las horas que dedica a cada tarea. Debido al surgimiento de nuevas formas de trabajo, híbridas y flexibles, se quiere desarrollar una aplicación móvil que permita realizar estas mismas tareas pero

desde cualquier dispositivo móvil. Aunque se dispone de un equipo con conocimientos de desarrollo para iOS y Android, se quiere explorar una nueva tecnología que permita el desarrollo multiplataforma (tipo Xamarin o Ionic) para unificar el desarrollo y el mantenimiento de las aplicaciones de ambas plataformas en un único código. El equipo de desarrollo no tiene experiencia con esta tecnología, pero se quiere aprovechar el proyecto para obtener experiencia y poder valorar su posible uso para futuros proyectos.

Solución

*Proyecto de **Tipo 2,** objetivo claro y solución poco conocida.*

*Metodología de desarrollo **Iterativo e incremental o ágil**. En este caso, el objetivo que se quiere alcanzar es claro: replicar el comportamiento de la aplicación web para que cada empleado pueda imputar las horas de cada tarea, pero en una aplicación móvil en lugar de la aplicación web. No obstante, la solución no está del todo clara porque, aunque el equipo tiene experiencia en desarrollo de aplicaciones móviles para plataformas específicas (iOS y Android), se quiere explorar una nueva tecnología que permita realizar el desarrollo para ambas plataformas con un mismo código. Es por ello que podría ser necesario realizar algún cambio sobre la solución a medida que se avanza con el proyecto y se obtiene conocimiento de la tecnología, por lo que una de las dos metodologías con enfoque incremental es la mejor opción.*

Proyecto 2

El CTO (Chief Technology Officer) de la compañía ha decidido mover parte de sus servicios al Cloud, ya que para ciertos servicios puede ser beneficioso en términos de ahorro de costes de desarrollo, mantenimiento y hospedaje. Hasta ahora la compañía ha sido muy reticente en su migración al Cloud, es por ello que ha decidido crear un pequeño equipo multidisciplinario para que se forme en temas de Cloud y pueda llevar a cabo un primer proyecto: analizar varias de las aplicaciones propias que ha desarrollado la compañía y ver si alguna de ellas puede ser candidata a ser migrada al proveedor Cloud seleccionado. Una vez seleccionada la aplicación, si la hubiera, se explorará cuáles de los servicios del proveedor Cloud pueden ser válidos para su hospedaje (máquinas virtuales, plataformas de aplicaciones, software como servicio, etc.)

Solución

*Proyecto de **Tipo 4**, objetivo poco claro y solución poco conocida.*

*Metodología **lean-agile o iterativo i incremental**. En este caso el objetivo no está del todo claro ya que se quiere explorar el uso del Cloud pero no se sabe muy bien aún con qué aplicaciones o para qué casos de uso. Es por ello que se va a realizar primero una exploración para encontrar alguna posible aplicación candidata. Por otro lado, tampoco se conoce muy bien la tecnología ya que el equipo no tiene experiencia y además se debe comenzar a formar desde cero. Es por ello que un enfoque iterativo es el más adecuado ya que nos permite ir adaptando los objetivos y la solución del proyecto en cada iteración a medida que se va avanzando en el desarrollo de la solución.*

Proyecto 3

Una empresa externa ha realizado una auditoría en la página web principal de la empresa para detectar problemas de usabilidad que afecten negativamente a usuarios con diversidad funcional. Dicha empresa externa ha elaborado un documento con todos aquellos aspectos que se deberían mejorar. Aunque estos cambios no implican un rediseño de la página web, sí que es necesario llevar a cabo un pequeño proyecto para implementar las mejoras enumeradas en el documento. Una vez completada la nueva versión se sustituirá la versión actual y en la siguiente auditoría ya se valorará si es necesario algún cambio futuro.

Solución

*Proyecto de **Tipo 1,** objetivo conocido y solución conocida.*

*Metodología de **Ciclo de vida en cascada**. En este caso el objetivo es conocido ya que es necesario realizar algunas mejoras de usabilidad a nivel de interfaz, sin añadir ni modificar ninguna funcionalidad. Además, estos cambios están claros, ya que una empresa externa especializada ha realizado un documento con los cambios que se deben implementar. La solución también es conocida ya que se trabaja sobre la misma tecnología y no se añade ninguna funcionalidad sustancial nueva, por lo que la incertidumbre es baja. Es por ello que la metodología en cascada es el mejor enfoque ya que se realiza una única nueva versión de golpe, donde no existe incertidumbre en la solución y, además, se conocen bien los requisitos a implementar.*

Pregunta 3 (25% puntuación)

Dentro de la ingeniería del software existen una serie de actividades principales que agrupan un conjunto de tareas específicas. En el apartado 2.3 del Módulo 1 se indican cuales son estas actividades:

- Gestión del proyecto
- Identificación y gestión de requisitos
- Modelización
- Construcción y pruebas
- Calidad
- Mantenimiento y reingeniería

Adicionalmente, existen una serie de roles principales que intervienen en los proyectos de ingeniería del software tal y como se detallan en el apartado 2.4 del Módulo 1:

- Jefe/a de proyecto
- Expertos/as del dominio
- Analista funcional
- Arquitecto/a
- Analista orgánico o analista técnico
- Programadores/as
- Experto/a de calidad
- Encargado/a del despliegue
- Responsable del producto

A continuación se listan una serie de tareas propias de un proyecto de ingeniería del software. Analízalas e indica a cuál de las actividades del apartado 2.3 pertenece cada una y cuál de los roles del apartado 2.4 es el responsable de llevarla a cabo.

1. Crear el diseño de las clases y asociaciones que intervendrán en la gestión de las notificaciones de una aplicación. ***(Modelización - Analista funcional)***
2. Definir la arquitectura del sistema e identificar qué tecnologías se usarán para su implementación. ***(Arquitecto/a - Modelización)***
3. Programar las pruebas unitarias automatizadas. ***(Programadores/as - Construcción y pruebas)***

4. Revisar los hitos conseguidos hasta la fecha y validar que se cumplen según la planificación inicial. ***(Gestión del proyecto - Jefe/a del proyecto)***
5. Validar que el rendimiento de la aplicación es aceptable y cumple con los requisitos de velocidad de uso definidos. ***(Experto/a de calidad - Calidad)***
6. Realizar un correctivo sobre un bug detectado en la aplicación. ***(Programadores - Mantenimiento y reingeniería)***
7. Desplegar la nueva versión de la aplicación en el entorno de producción. ***(Encargado/a del despliegue - Mantenimiento y reingeniería)***
8. Revisar las pruebas realizadas para validar que se tienen en cuenta todas las posibles casuísticas. ***(Experto/a de calidad - Calidad)***
9. Revisar que todos los requisitos se han implementado al finalizar el proyecto. ***(Identificación y gestión de requisitos - Responsable del producto)***
10. Escribir el código de una nueva funcionalidad. ***(Programadores/as - Construcción y pruebas)***

Preguntas del Módulo 2

Pregunta 4 (15% puntuación)

Algunos de los conceptos que se tratan en el Módulo 2 de la asignatura son la herencia y las asociaciones. Asimismo, también hemos visto que las clases están formadas por atributos, asociaciones y operaciones.

A continuación se detalla un listado de clases. **Para cada una de ellas**, propón:

- Dos subclases
- Una clase con la que pueda tener una relación de asociación y el nombre y la multiplicidad de dicha asociación (ninguno o uno, uno, uno o más de uno, varias) entre ambas clases.
- Un atributo, tanto para la superclase como para cada una de las dos subclases (no es necesario para la clase propuesta para la asociación).

Por ejemplo, para la clase Mascota:

- Subclases: Perro y Pájaro

- Clase asociación: Dueño. Una Mascota **tiene uno o más de un** Dueño y un Dueño **se responsabiliza de una o más** Mascotas.
- Atributos:
    - Mascota: peso
    - Perro: raza
    - Pájaro: tamañoAlas

Nota: Intentad evitar, en la medida de lo posible, usar verbos demasiado genéricos en las asociaciones. Fijaos que, en el ejemplo anterior, hemos indicado “un Dueño **se responsabiliza de una o más** mascotas” en lugar de “un Dueño **tiene una o más** mascotas”.

Clases:

1. Vehículo
    - *Subclases: Coche y Moto*
    - *Clase asociación: Conductor. Un Vehículo **es conducido por uno o más** Conductor y un Conductor **conduce uno o más** Vehículos.*
    - *Atributos:*
        - *Vehículo: matrícula*
        - *Coche: numAirbags*
        - *Moto: longitudManillar*
2. DispositivoElectrónico
    - *Subclases: Smartphone y Ordenador*
    - *Clase asociación: Seguro. Un DispositivoElectrónico **tiene ningún o un** Seguro y un Seguro **cubre a un** DispositivoElectrónico.*
    - *Atributos:*
        - *DispositivoElectrónico: marca*
        - *Smartphone: tiene5G*
        - *Ordenador: resoluciónWebCam*
3. PiezaDeRopa
    - *Subclases: Pantalón y Camisa*
    - *Clase asociación: Diseñador. Una PiezaDeRopa **es diseñada por un** Diseñador y un Diseñador **diseña varias** PiezasDeRopa.*

- *Atributos:*
  - *PiezaDeRopa: talla*
  - *Pantalón: tallaCintura*
  - *Camisa: longitudMangas*

4. Mueble
   - *Subclases: Armario y Sofá*
   - *Clase asociación: Vivienda. Una Vivienda* ***contiene varios*** *Muebles y un Mueble* ***está ubicado en una*** *Vivienda.*
   - *Atributos:*
     - *Mueble: color*
     - *Armario: numEstanterías*
     - *Sofá: numPlazas*
5. Instrumento
   - *Subclases: Guitarra y Piano*
   - *Clase asociación: Músico. Un Músico* ***toca uno o más*** *Instrumentos y un Instrumento* ***es tocado por uno o más*** *Músicos.*
   - *Atributos:*
     - *Instrumento: nombreFabricante*
     - *Guitarra: esEléctrica*
     - *Piano: numTeclas*

Nota de la solución*: es importante que los atributos propuestos para la superclase sean comunes a las dos subclases y que los atributos propuestos para cada subclase sean específicos de cada subclase.*

Pregunta 5 (10% puntuación)

Supongamos que tenemos el siguiente sistema orientado a objetos de una plataforma de contenido en *Streaming*. Por simplicidad, supongamos que el sistema está formado por las siguientes clases con sus atributos:

- Contenido (clase abstracta): título, género, director, costeProducción
- Película (clase concreta, subclase de Contenido): duración, añoEstreno

- Serie (clase concreta, subclase de Contenido): numCapítulos, numTemporadas, duraciónCapítulo
- Usuario (clase concreta): alias, géneroFavorito, email

A partir de los datos anteriores, responde a las siguientes preguntas:

1. ¿Es posible instanciar las clases Película, Serie, Contenido y Usuario?
   No es posible instanciar la clase contenido ya que se trata de una clase abstracta. Únicamente será posible instanciar sus subclases, Película y Serie ya además son clases concretas. La clase Usuario también se podrá instanciar ya que se trata de una clase concreta.

2. Supongamos que la clase Contenido define la operación polimórfica calcularDuraciónTotal. Explica textualmente cómo se calcularía en cada función para cada una de sus subclases.
   - *Subclase Película: la operación calcularDuraciónTotal retornaría la duración total de la película.*
   - *Subclase Serie: la operación calcularDuraciónTotal retornaría la suma de la duración de cada capítulo.*

3. Propón una asociación entre las clases Contenido y Usuario. Indica también la multiplicidad de la asociación entre ambas clases. ¿Se trata de una asociación polimórfica?
   Asociación *Visualiza: Un usuario visualiza cualquier número de Contenidos y un Contenido es visualizado por cualquier número de Usuarios.*
   Este caso se trata de una asociación polimórfica, ya que la asociación de Usuario se realiza con la superclase Contenido, sin diferenciar si se trata de la subclase Película o Serie.

4. ¿Qué visibilidad debería tener el atributo email de la clase Usuario si queremos que sea visible por las instancias de la clase Usuario, pero que no sea visible por las instancias de Contenido ni de ninguna de sus subclases?
   Privado

5. ¿Qué visibilidad debería tener el atributo costeProducción de la clase Contenido para que sea visible desde las instancias de la clase Contenido y sus subclases Película y Serie pero no desde las instancias de la clase Usuario?
   Protegido

6. ¿Qué visibilidad debería tener el atributo título de la clase Contenido si queremos que tanto las instancias de la clase Contenido y sus subclases como las de la clase Usuario tengan acceso al atributo título?
   Público

Pregunta 6 (25% puntuación)

Enunciado

Suponed que queremos modelar, utilizando la orientación a objetos, una aplicación muy sencilla para gestionar las ligas de fútbol de una región (la gestión de los resultados y la clasificación quedan fuera del alcance de la aplicación).

Como hemos dicho, la aplicación permitirá gestionar diferentes ligas. Para cada liga, nos interesará saber el nombre de la liga y el nombre de la categoría a la que pertenece.

En cada liga, jugarán varios equipos, de los cuales se debe tener en cuenta:

- Un equipo no podrá jugar en más de una liga.
- Para cada equipo guardaremos su nombre, su color o colores principales (máximo 2), su año de fundación y el número de socios, aunque hay que tener en cuenta que puede haber equipos que no tengan socios.

Por otro lado, la aplicación deberá gestionar también a las personas que forman parte de la liga y de los equipos, concretamente, estas pueden ser:

- Jugadores
- Entrenadores

- Árbitros.

La aplicación no gestionará personas que no tengan ninguno de estos roles (y ninguna persona podrá tener más de un rol a la vez). Para cada una de estas personas deberemos cumplimentar su nombre, sus apellidos, su año de nacimiento y su número de identificación (DNI o NIE).

Además, para cada jugador:

- Deberemos conocer el nombre de la posición que desempeña (únicamente se permitirá registrar a un jugador con, como máximo, dos posiciones “oficiales”) y si es capitán o no.
- Un jugador solo podrá jugar en un equipo y un equipo deberá tener al menos 11 jugadores, aunque, al tratarse de ligas regionales, no se pone límite a los jugadores que puede inscribir cada equipo.

Por otro lado, para cada entrenador registrado:

- Deberemos conocer los años de experiencia que tiene entrenando equipos.
- Igual que en el caso de los jugadores, un entrenador únicamente puede entrenar a un equipo pero un equipo puede tener uno o, como máximo, dos entrenadores.

Finalmente, para los árbitros registrados:

- Deberemos conocer los partidos que lleva arbitrados.
- Pueden arbitrar en más de una liga y cada liga tendrá varios árbitros designados para poder cubrir todos los partidos.

A partir de la información anterior, responde a los siguientes apartados:

a) (10%) Haz una lista de todas las clases que puedas identificar en el enunciado anterior. Para cada clase, indica si es abstracta o concreta y, en caso de que sea una subclase, de qué otra clase lo es. Además, para cada clase, haz una lista de sus atributos. Para aquellos que sean opcionales y/o multivaluados, indícalo al lado de su nombre.
b) (15%) Haz una lista de todas las asociaciones que identifiques. Para cada asociación, utiliza la siguiente plantilla sustituyendo las palabras entre paréntesis por lo que corresponda (y eliminando los paréntesis):

- (nombreAsociación): Asocia cada instancia de (Clase 1) con (ninguna o una / una y sólo una / una o más de una / n o más de n / cualquier número de) instancia/as de (Clase 2) y cada instancia de (Clase 2) con (ninguna o una / una y sólo una / una o más de una / n o más de n / cualquier número de) instancia/as de (Clase1).

NOTA: Para determinar las asociaciones y multiplicidades haced las suposiciones que consideréis adecuadas e indicadlas junto a la respuesta.

Solución

a)

- *Liga (Concreta): nombre, categoría*
- *Equipo (Concreta): nombre, colores (multivaluado), añoFundación, numSocios (es válido considerarlo opcional para el caso en que no tenga socios)*
- *Persona (Abstracta): nombre, apellidos, añoNacimiento, numIdentificación*
- *Jugador (Concreta, subclase de Persona): posición (multivaluado), esCapitán*
- *Entrenador (Concreta, subclase de Persona): añosExperiencia*
- *Árbitro (Concreta, subclase de Persona): numPartidosArbitrados*

b)

- *JuegaEn: Asocia cada instancia de Equipo con 11 o más instancias de Jugador y cada instancia de Jugador con una única instancia de Equipo.*
- *EntrenaA: Asocia cada instancia de Equipo con 1 o 2 instancias de Entrenador y cada instancia de Entrenador con una única instancia de Equipo.*
- *AsignadoA o ArbitraEn: Asocia cada instancia de Árbitro con cualquier número de instancias de Liga y cada instancia de Liga con cualquier número de instancias de árbitro.*
- *PerteneceA: Asocia cada instancia de Equipo con una única instancia de Liga y cada instancia de Liga con cualquier número de instancia de Equipo.*