

Asignatura 240305

Segundo curso del Grado en Ingeniería Informática

Ingeniería del Software

Exámenes

Federico Fariña Figueredo

18 de enero de 2018

Examen Final 14-Enero-2014

Ejercicio 1. Test (5 puntos)

Seleccione la respuesta correcta (sólo una) rodeando su opción con un círculo. Tenga en cuenta que cada respuesta correcta vale 0,5 puntos y que cada fallo resta 0,25 puntos a la puntuación de este ejercicio :

1. La ingeniería del software se define como:
 - a) Otra forma de denominar a la ciencia de la computación
 - b) La aplicación de enfoques sistemáticos, disciplinados y cuantificables hacia el desarrollo, operación y mantenimiento del software
 - c) Los aspectos teóricos legales que permiten la puesta en marcha de una empresa que produzca software.
 - d) La aplicación de teorías científicas conocidas a la resolución de problemas numéricos.
2. En los procesos software la documentación es:
 - a) Una actividad incluida sólo en las fases de análisis de requisitos y diseño
 - b) Una actividad presente en todas las actividades del proceso
 - c) Una actividad independiente de todas las demás que debe realizarse tras el despliegue del producto
 - d) Un rollo que no es absolutamente necesario
3. Denominamos requisitos funcionales a:
 - a) Los que explican como debe reaccionar el sistema ante entradas concretas
 - b) Restricciones sobre los servicios del sistema (como temporales o de cumplimiento de estándares)
 - c) A los que se refieren a las características del dominio de la aplicación
4. El diagrama de clases de UML permite mostrar:
 - a) La funcionalidad del sistema en términos de actores y relaciones entre actividades
 - b) Los estados y transiciones de estado del sistema
 - c) Las clases del sistema, sus atributos y cómo se relacionan entre ellas
5. La validación del software se refiere a:
 - a) La comprobación de que se ha construido el software que pedía el cliente
 - b) La realización de pruebas para comprobar que no hay fallos en la ejecución del software
 - c) La comprobación de que los subsistemas se integran sin problemas
 - d) Lo mismo que la verificación
6. La verificación y validación de software es:
 - a) Un proceso caro e innecesario
 - b) Un proceso caro, por lo que debe estar perfectamente planificado
 - c) Un proceso barato que se realiza sin necesidad de planificación
7. La calidad del software tiene que ver fundamentalmente con:
 - a) La utilización de lenguajes de programación modernos
 - b) La utilización de interfaces gráficos
 - c) El seguimiento de estándares

8. El desarrollo de software está siempre sujeto a modificaciones en los requisitos. Ello obliga a:
 - a) Llevar un control sobre los nombres de los documentos y programas que se generan
 - b) Firmar un contrato que impida al cliente modificaciones
 - c) Realizar diversas reuniones en la fase de mantenimiento
9. En el modelo de proceso en espiral :
 - a) Cada fase comienza cuando se ha concluido la anterior
 - b) El sistema se crea añadiendo componentes funcionales al sistema (incrementos)
 - c) Podemos acomodar otros modelos
10. Los métodos ágiles de desarrollo de software :
 - a) Ponen énfasis en la adaptabilidad pues consideran el cambio un aspecto natural del proyecto
 - b) Eliminan la realización de pruebas unitarias
 - c) Limitan las interacciones con el usuario

Ejercicio 2. Casos de uso (2,5 puntos)

Cruceros Schettino está desarrollando un sistema de reservas y emisión de billetes vía web.

Para reservar pasajes el sistema solicita un listado de tres países por los que el cliente desea pasar, el mes en el que se desea viajar y la cantidad de pasajeros. El sistema presenta un listado de cruceros que cumplen los requisitos del pasajero (y disponen de camarotes libres). El pasajero selecciona el crucero deseado y el sistema procede a pedirle los DNIs de los pasajeros, realiza la reserva y muestra un número de reserva.

Un pasajero con reserva (de la que aún no se ha emitido billete) puede pedir la emisión de un billete. Para ello debe indicar su DNI y su número de reserva. El sistema le muestra un listado de camarotes con espacio. El usuario elige camarote y el sistema emite el billete.

Se pide:

- Modelo de casos de uso
- Detalle del caso de uso: emitir billete

Ejercicio 3. Diagrama de clases (2,5 puntos)

Se va a implementar un programa que representa una reserva natural. Una reserva natural contiene un conjunto de seres vivos: plantas, animales carnívoros, animales herbívoros y animales omnívoros. Todos los seres vivos nacen, crecen, se alimentan, se reproducen y mueren. Obviamente las plantas se alimentan de aire y tierra mientras que los animales cada uno de lo que le toque. Además, los carnívoros y omnívoros cazan herbívoros. De cada especie animal y vegetal nos interesa conocer sus fechas de nacimiento y muerte, además del nombre de su especie y un código unitario que la identifique. En el caso de los animales nos interesa también que cada espécimen incorpore información sobre quiénes son sus descendientes.

Cree un diagrama de clases para la reserva natural. Debe incluir las propiedades y métodos que se derivan de lo expuesto en el enunciado.

Examen Práctico Final 14-Enero-2014

La práctica 8 que entregó no se adapta a lo que debería ser en esta asignatura. Procure mejorar su contenido teniendo en cuenta:

- Concordancia del diseño con el código: Si su código se ha obtenido a partir de su diseño no tiene sentido que no concuerden
- Documentación: El código debe incorporar documentación que permita entender qué hace.
- Trazabilidad: Se supone que usted ha modificado el código que le entregó alguien. ¿Que modificaciones ha introducido? ¿quién era ese alguien? y, ¿quién es usted?
- Usabilidad: No tiene sentido que el usuario del programa tenga que pensar cómo hay que meter los datos para que todo funcione, que haya fallos por que se ha introducido un dato erróneo sin que ni siquiera se haya avisado previamente de cómo son los datos no erróneos, etc.

Es decir, no le pido que implemente nada que no esté ya implementado (salvo aspectos de interfaz con el usuario). De hecho, no se lo recomiendo no vaya a ser que al final entregue algo que no compile. Le pido que su entrega tenga una pinta más profesional.

Debe entregar:

- Su diagrama de clases en formato pdf.
- Los ficheros .java de la aplicación
- Si su fichero .zargo incluya información relevante no incluida en los .java, el .zargo
- Si dispone de algún fichero adicional explicando aspectos relevantes, ese fichero

Para ello cree una carpeta que incluya todo lo anterior cuyo nombre sea sus dos apellidos seguidos de su nombre y comprímala en .tar.gz

Examen Final 24-Enero-2014

Ejercicio 1. Test (5 puntos)

Seleccione la respuesta correcta (sólo una) rodeando su opción con un círculo. Tenga en cuenta que cada respuesta correcta vale 0,5 puntos y que cada fallo resta 0,15 puntos a la puntuación de este ejercicio :

1. Las herramientas CASE:
 - a) Permiten mejorar la productividad en el desarrollo y mantenimiento del software
 - b) Es como se denominan a los compiladores en el entorno de la ingeniería del software
 - c) Son un conjunto de estándares de obligado cumplimiento
 - d) Se aplican en la fase de Análisis.
2. El paradigma de programación orientada a objetos:
 - a) Es una forma de denominar a los lenguajes orientados a objetos
 - b) Es una filosofía de desarrollo de programas
 - c) Está relacionada con mejorar el tiempo de ejecución de los programas
 - d) Surge a partir de la aparición de Java
3. El documento de requisitos de software:
 - a) Se escribe al final de la fase de despliegue
 - b) Debe incluir requisitos de usuario y de sistema
 - c) Es un documento de formato libre. No existen estándares.
4. UML es:
 - a) Un lenguaje de programación de bajo nivel
 - b) Un conjunto de diagramas y lenguajes para representar software
 - c) Un diagrama que permite representar clases
 - d) Un lenguaje de programación de alto nivel
5. Verificar el software consiste en:
 - a) Conseguir venderlo
 - b) Comprobar que hace lo que el usuario pedía
 - c) Comprobar que se ha desarrollado sin errores
6. Algunas técnicas de verificación de software son:
 - a) Las inspecciones de software y las pruebas de caja negra
 - b) Las pruebas de caja blanca y las auditorías de calidad
 - c) El seguimiento de standards en el desarrollo.
 - d) Las técnicas de relajación conductistas.
7. En el marco de la ingeniería del software cuando hablamos de gestión de configuración nos referimos a
 - a) La inclusión de opciones de configuración en los interfaces de los programas
 - b) A los procesos que facilitan el desarrollo de diferentes versiones de un mismo producto
 - c) A los cambios organizativos en la empresa de desarrollo de software

8. La calidad del software:
 - a) Depende de atributos cuya medida es difícil
 - b) Depende, fundamentalmente del uso de herramientas CASE
 - c) Depende del dominio del problema
9. El desarrollo basado en componentes :
 - a) Facilita la gestión de configuración
 - b) Reduce tiempos y costes de desarrollo
 - c) Facilita la programación: un componente se programa más rápido que un módulo ad-hoc
10. El desarrollo en sala limpia o síntesis automática :
 - a) Permite mucha participación del usuario en el desarrollo del producto
 - b) Disminuye el tiempo de verificación de código
 - c) Se adecúa a proyectos pequeños de rápido desarrollo

Ejercicio 2. Casos de uso (2,5 puntos)

Se le contrata para desarrollar un sistema de apoyo a la docencia que permite a los profesores calificar a sus estudiantes. Su aplicación parte de los siguientes requisitos:

- Un profesor puede calificar a sus alumnos
- Las calificaciones del profesor van a parar a la base de datos de la universidad
- Un profesor también puede modificar (durante un año) la nota de uno de sus alumnos. Para ello debe seleccionar la asignatura, y el estudiante.
- Los estudiantes pueden ver un listado con las notas que han sacado en todas las asignaturas
- Un profesor puede ver un listado de las notas de todos los alumnos de su asignatura
- Un usuario sólo puede trabajar en el sistema tras autenticarse por medio de nombre de usuario y contraseña
- El profesor puede imprimir en un documento pdf el listado de notas de una asignatura

Se pide:

1. Un diagrama de casos de uso. Se exige que el diagrama tenga al menos una inclusión o una extensión
2. Detalles del caso de uso modificar nota

Ejercicio 3. Diagrama de clases (2,5 puntos)

Un sistema de gestión bancaria opera con, al menos, dos tipos de cuentas bancarias: Cuenta corriente y cuenta de ahorro. Ambas están asociadas a una persona (el titular), se identifican por un número de cuenta y disponen de un saldo. El titular de una cuenta puede ver el listado de movimientos (entradas o salidas de una cantidad por un concepto en una fecha dada) de la cuenta, ingresar una cantidad, reintegrar una cantidad o ver el saldo actual. Los dos tipos de cuentas se diferencian en el modo en que se gestiona el inters. En las cuentas corrientes es un valor fijo mientras que en las de ahorro es un valor que se recalcula trimestralmente. Además las cuentas corrientes están sujetas a una comisión fija anual y sobre ella se pueden realizar transferencias y adeudos

Cree un diagrama de clases para representar esta parte del banco.

Examen Práctico Final 24-Enero-2014

Tras usar durante un par de años la agenda que diseñó en la práctica siete su cliente le pide una modificación. Para mejorar la capacidad de organización se desea que exista el concepto de grupo. Un grupo es una colección de contactos de la agenda que tienen algo en común.

Se desea que se pueda

- Crear grupos
- Añadir a un grupo contactos que ya estaban en la agenda
- Ver el listado de grupos
- Eliminar un grupo. Cuando se elimine un grupo se puede decidir si se eliminan todos los contactos del grupo o sólo el grupo

Se pide que modifique los documentos que entregó en la práctica 7 de modo que, además de lo que se pedía en dicha práctica incluya:

- Diagrama de casos de uso de la agenda que incluya los nuevos casos de uso
- Diagrama de clases de la aplicación completa. Dicho diagrama puede ser sin detalles si lo entregó así en su momento, pero la clase Grupo debe aparecer con todas sus propiedades y métodos

Debe entregar la misma documentación que se pedía en la práctica 7 en una carpeta de nombre Apellido1Apellido2Nombre comprimida en .tar.gz

Examen Final 12-Junio-2015

APELLIDOS Y NOMBRE:.....

Debe comenzar el examen haciendo el test. Dispone de 30 minutos para ello. Una vez lo finalice, debe dejarlo debajo del resto de hojas del examen y seguir con el resto de preguntas. El profesor le indicará cómo debe entregarlo.

Ejercicio 1. Test (4 puntos)

Seleccione la respuesta correcta (sólo una) rodeando su opción con un círculo. Tenga en cuenta que cada respuesta correcta vale 0,4 puntos y que cada fallo resta 0,2 puntos a la puntuación de este ejercicio :

1. Dados los siguientes requisitos para un editor de texto, ¿cual es funcional?:
 - a) El sistema puede ejecutarse en cualquier sistema operativo.
 - b) El sistema propone correcciones si detecta faltas ortográficas.
 - c) El programa dispone de mecanismos de protección contra la copia.
 - d) El programa debe evolucionar continuamente.
2. Las herramientas CASE permiten:
 - a) Reducir la calidad del software.
 - b) Mejorar la velocidad de finalización de la versión beta de un proyecto.
 - c) Mejorar la gestión de proyectos complejos.
 - d) Decidir que metodología de desarrollo va a dar mejores resultados en el desarrollo de un proyecto.
3. EL paradigma orientado a objetos define un programa como:
 - a) Una secuencia de instrucciones que trabaja con datos
 - b) Un grupo de funciones que llaman a funciones
 - c) La respuesta lógica a una expresión lógica mediante reglas lógicas
 - d) Un grupo de objetos interaccionando entre sí.
4. Los diagramas de flujo de datos DFDs:
 - a) Indican como se transforman los datos del sistema.
 - b) Indican como se estructuran los datos del sistema.
 - c) Indican como se almacenan los datos del sistema.
 - d) Indican como se comunican los datos del sistema.
5. El modelo arquitectónico Cliente/Servidor:
 - a) Es escalable (si los servidores de servicios son independientes)
 - b) Es la única opción para aplicaciones multiplataforma
 - c) Obliga a que todos los subsistemas usen el mismo modelo de datos
 - d) Se basa en capas que usan los servicios de capas inferiores y ofrecen servicios a las superiores

6. El proceso unificado (UP)
 - a) Es otra forma de llamar a la metodología en cascada
 - b) Es Iterativo e incremental
 - c) No tiene en cuenta la arquitectura
 - d) Aplaza la gestión de riesgos a las últimas iteraciones
7. El patrón modelo, vista, controlador indica que:
 - a) Conviene mantener por separado los componentes referidos a la presentación.
 - b) La lógica de programa depende del interfaz
 - c) El controlador debe formar siempre parte del modelo
 - d) Los interfaces deben ser preferentemente textuales
8. Se denominan pruebas de integración a las pruebas que se realizan sobre:
 - a) el sistema completo.
 - b) grupos de módulos interconectados .
 - c) unidades.
 - d) el sistema completo en el entorno real y con intervención del usuario final
9. La documentación del software :
 - a) no tiene nada que ver con la calidad del software
 - b) es un proceso que debe estar estandarizado.
 - c) es un proceso que depende del desarrollador
10. El coste de un proyecto:
 - a) es independiente de la cantidad de programadores necesarios para su desarrollo
 - b) no se ve alterado por factores externos al proyecto
 - c) es fácil de estimar en base al documento preliminar de requisitos
 - d) puede estimarse con procedimientos standard (COCOMO)

Ejercicio 2. Cuestiones (1 punto)

Defina de forma breve y concisa (no es necesario usar más de una frase) los siguientes conceptos:

- Verificación de software
- Validación de software
- Inspección de software

Ejercicio 3. Casos de uso (3 puntos)

Su empresa recibe a un posible cliente cuyo problema es muy sencillo. El garaje de su comunidad dispone de un único portón que se emplea tanto para entrar como para salir. Además, el pasillo de acceso es largo y estrecho (no permite el paso de dos vehículos simultáneamente). Para evitar problemas se ha pensado en poner semáforos en todos los puntos de acceso al pasillo. Los usuarios autorizados usarán mandos a distancia codificados para indicarle al sistema que desean entrar o salir. Se instalarán sensores para detectar la presencia de vehículos en el el pasillo.

Se pide:

1. Diagrama de casos de uso del sistema
2. Descripción detallada (caso de uso expandido) de algún caso de uso incluyendo un diagrama de secuencia para el flujo principal.

Ejercicio 4. Diagrama de clases (2 puntos)

LoBajoTo es una empresa que se dedica a alquilar videos en *streaming* por Internet. Cuando el cliente accede a la aplicación encuentra un catálogo en el que aparecen diferentes elementos alquilables identificados por un *identificador* único y una descripción *artística* (el título, director, actores, temática, ...). Los elementos alquilables pueden ser películas, series, temporadas de una serie o capítulos individuales de una serie. Obviamente, a partir de la Serie podemos navegar entre sus temporadas y sus capítulos. En cualquier caso, cuando alquilamos una serie, la bajamos capítulo a capítulo.

Cada elemento alquilable puede estarlo en diferentes formatos y calidades, lo que lleva a diferentes precios.

Considerando que el empleado de la empresa debe poder añadir y borrar todos los elementos del catálogo y todos los videos mencionados, cree un diagrama de clases que represente las relaciones entre las clases del modelo de negocio (conceptual) y los atributos y métodos principales de dichas clases

Examen Práctico Final 12-Junio-2015

La práctica 9 que entregó no cumple los mínimos necesarios para aprobar la asignatura. Procure mejorar su contenido teniendo en cuenta:

- Obviamente, el programa debe compilar sin errores.
- También debe funcionar correctamente, o que significa:
 - Debe tener en cuenta el peso de los camiones para calcular lo que pagan.
 - Debe funcionar cuando hay menos de tres cabinas abiertas.
 - Debe ofrecer resultados que tengan sentido (no es posible que aumentando la frecuencia de llegada disminuya el número de vehículos atendidos)
- Si su código se ha obtenido a partir de su diseño no tiene sentido que no concuerden.
- Aspectos de usabilidad. Ejecutar algunos programas es una tortura:
 - No admiten equivocaciones en la entrada de datos
 - No informan de los tipos de los valores a introducir
 - Obligan a teclear excesos de datos innecesarios
- Debe existir información que permita saber qué cambios incluye la nueva versión y quién es el autor de dichos cambios.
- El código debe estar comentado

Es decir, no le pido que implemente nada que no esté ya implementado (salvo aspectos de interfaz con el usuario y eliminación de errores). De hecho, no se lo recomiendo no vaya a ser que al final entregue algo que no compile. Le pido que su entrega tenga una pinta más profesional.

Debe entregar:

- Su fichero .zargo con los diagramas de diseño
- Cualquier otro fichero que considere ofrece información necesaria
- Los ficheros de su proyecto
- Un fichero de texto plano que indique que ha hecho durante el examen

Para ello cree una carpeta que incluya todo lo anterior cuyo nombre sea sus dos apellidos seguidos de su nombre y comprímala en .tar.gz. Deposite el fichero comprimido en la carpeta **recoger** que encontrará en el escritorio

Examen Final 23-Junio-2015

APELLIDOS Y NOMBRE:.....

Debe comenzar el examen haciendo el test. Dispone de 30 minutos para ello. Una vez lo finalice, debe dejarlo debajo del resto de hojas del examen y seguir con el resto de preguntas. El profesor le indicará cómo debe entregarlo.

Ejercicio 1. Test (4 puntos)

Seleccione la respuesta correcta (sólo una) rodeando su opción con un círculo. Tenga en cuenta que cada respuesta correcta vale 0,4 puntos y que cada fallo resta 0,2 puntos a la puntuación de este ejercicio :

1. Para evitar errores en los requisitos debemos:
 - a) validar cada requisito con el usuario.
 - b) planificar las entregas de modo concienzudo y de acuerdo con las necesidades del cliente/usuario
 - c) actualizar el documento de requisitos según avanzamos en el desarrollo
 - d) verificar su validez, consistencia, completitud, realismo y verificabilidad
2. Los diagramas entidad-relación:
 - a) reflejan los datos del sistema y sus relaciones
 - b) son necesarios única y exclusivamente cuando se diseña una base de datos
 - c) son una herramienta que permite reflejar requisitos funcionales
 - d) son una herramienta que permite reflejar requisitos no funcionales
3. El diagrama de paquetes de UML:
 - a) se emplea para indicar cómo se relacionan los paquetes lógicos con el hardware
 - b) se utiliza para indicar cómo se agrupan los componentes de un sistema grande en subsistemas más pequeños
 - c) se utiliza para mostrar las clases, interfaces, métodos, propiedades y relaciones que forman parte de un paquete.
 - d) reflejan la forma en que diferentes objetos colaboran entre sí
4. El diseño orientado a eventos considera que el programa se compone de:
 - a) sensores que captan los eventos, distribuidores que los envían y manejadores que los procesan
 - b) un interfaz de usuario, un controlador de casos de uso y el modelo de negocio
 - c) un interfaz táctil que permite lanzar ordenes a la aplicación
 - d) un interfaz gráfico y un conjunto de métodos
5. Los métodos ágiles de desarrollo:
 - a) son metodologías en cascada con más interacción del usuario
 - b) también se denominan Scrum
 - c) se basan en transformaciones formales que permiten obtener el desarrollo como refinamiento de la especificación
 - d) son de naturaleza iterativa e incremental

6. Los procesos de inspección de software:
 - a) consisten en ejecutar el programa y analizar que los resultados se ajusten a las especificaciones
 - b) consisten en localizar mediante un debugger los errores de programación
 - c) consisten en señalar los errores de codificación
 - d) permiten comprobar que se cumplen estándares de calidad
7. La mejora de procesos tiene que ver con:
 - a) la aplicación de estándares sobre la ingeniería del software
 - b) la mejora de la calidad de los procesos de una aplicación
 - c) la mejora de la calidad de la metodología de desarrollo aplicada por la empresa
 - d) La mejora del proceso de pruebas de aplicaciones.
8. Una desventaja del desarrollo basado en componentes es:
 - a) Da más problemas en la gestión de configuración
 - b) En general, aumenta costes de desarrollo
 - c) Asegura una mayor fiabilidad de los componentes
 - d) Obliga a diseñar los componentes antes de la lógica de negocio
9. Las actividades de gestión de un proyecto informático incluyen:
 - a) Estimación de coste, redacción de informes y selección del personal
 - b) Redacción de la propuesta y planificación de las pruebas
 - c) Redacción de la propuesta, análisis y diseño
 - d) Análisis, diseño y desarrollo
10. Denominamos gestión de riesgos a :
 - a) estudiar el mercado para comprobar si existen competidores para nuestro software
 - b) planificar métodos que eviten que si algo sale mal los efectos sean muy perniciosos
 - c) el proceso de eliminación de problemas que pueden hacer que el proyecto no finalice
 - d) el análisis de qué cosas pueden ir mal durante el proyecto

Ejercicio 2. Cuestiones (1 punto)

Defina de forma breve y concisa los siguientes conceptos:

- Ingeniería del software
- Análisis del software
- Arquitectura del software

Ejercicio 3. Casos de uso (2,5 puntos)

La empresa elCineDeMiBarrio está actualizandose. Le han contratado para que haga un programa que les permita utilizar Internet como espacio comercial. Se desea que el programa permita que los clientes compren sus entradas desde su página web y, de paso, le piden otras funcionalidades.

Se pretende que cada vez que un cliente compre por Internet una o varias entradas para alguna de las películas emitidas en cualquiera de las salas del cine, se le muestre un código de compra con el que podrá retirar sus entradas en una máquina expendedora automática que estará situada en la entrada del cine. También se le mostrará por pantalla los asientos que le han sido asignados. Para conseguir que la información sobre asientos libres sea correcta, las taquilleras deberán, cada cierto tiempo, volcar la información sobre los asientos ocupados al sistema (el sistema automático va a convivir con uno de taquilla tradicional).

Como funcionalidades adicionales se pretende que el encargado pueda modificar desde su casa la programación de películas a emitir. Para ello el encargado accederá al sistema e indicará las películas que se van a proyectar, su duración y la sala en que se proyectará cada una. Los operadores de sala pueden acceder al sistema para obtener el listado de cada sala para que sepan qué película deben poner a cada hora. El encargado también desea poder acceder en un instante dado a un informe sobre la ocupación de cada sala en el instante actual (en realidad, verá el número de entradas vendidas para cada sala).

Un último requisito del sistema consiste en que los operadores de sala, si comprueban que se ha programado en su sala una película no disponible, puedan solicitar mediante la aplicación una copia al distribuidor.

Dada esta información

- identifique los actores de la aplicación
- Haga un diagrama de casos de uso
- Para los casos de uso `comprarEntradaViaWeb` y `cogerLaEntradaDeLaMaquina` se desea el caso de uso detallado incluyendo un diagrama de colaboración o de secuencia (el que considere más adecuado)

Ejercicio 4. Diagrama de clases (2,5 puntos)

La aplicación `VotemosTo2` permite establecer referéndums entre todos sus usuarios. Cualquier persona puede convertirse en usuario simplemente con rellenar un formulario en el que se le piden sus datos personales junto con el DNI y un número de cuenta para posibles pagos (obviamente, el sistema comprobará este último dato conectándose al banco). También se le pide si desea poder proponer referéndums (usuario VIP), lo que lleva aparejado un pago o ser un simple votante. Cualquier usuario registrado puede pasar a ser VIP (a poder proponer referéndums) pagando el importe adecuado. Los usuarios VIP pueden proponer un referéndum (pagando por ello un precio fijo). Para ello establecen la pregunta, un número mínimo de votantes (quórum) y una fecha. El resto de usuarios recibirán información sobre la votación que se ha propuesto. Cualquier usuario puede pasar a formar parte del censo de un referéndum anunciado (pagando también un precio módico). Cuando llegue la fecha de una votación, si el censo supera el quórum el sistema avisa a los participantes del censo de que deben votar. Los socios del censo tienen 24 horas para votar tras lo que el sistema cierra la votación y calcula los resultados.

El sistema debe almacenar información sobre:

- Los referéndums que se han realizado y sus resultados (a favor, en contra o abstención)
- Los referéndums propuestos por cada usuario VIP (se hayan realizado o no)
- Los referéndums en los que cada usuario ha formado parte del censo

Dada esta información se solicita un diagrama de clases que represente la lógica de negocio. Las clases deben incluir las propiedades y métodos que permitan hacerse socio, proponer un referéndum, apuntarse a un referéndum y votar

Examen Práctico Final 23-Junio-2015

Una empresa desea proporcionar a sus empleados la posibilidad de mantener en sus servidores una agenda personalizada. A cada trabajador de la empresa se le suministra un nombre de usuario y una clave que les permiten entrar en la aplicación. Una vez dentro cada empleado puede agregar tanto contactos personales como contactos de empresa. También puede crear citas tanto con conocidos como con contactos de empresa. Todos los usuarios pueden ver los contactos de empresa de todos los usuarios (pero no modificarlos ni borrarlos). Además, cada usuario puede ver, modificar y borrar sus contactos personales así como todas sus citas.

El único usuario que puede modificar y borrar contactos de empresa es el jefe. Obviamente él es también el único que puede añadir nuevos trabajadores y borrar trabajadores (es decir, añadir y eliminar usuarios de la aplicación)

Como puede ver esta agenda es, en cierto modo, similar a la que presentó para la práctica 7. En base a eso, tiene dos formas de resolver este examen:

- Si considera que lo que entregó para la práctica mencionada estaba suficientemente bien hecho, modifique la documentación de modo que :
 - Se reflejen los nuevos casos de uso en el diagrama de casos de uso.
 - Para el caso de uso `borrarContactoDeEmpresa` debe dar su especificación completa y crear un diagrama de secuencia.
 - Modifique el diagrama de clases de modo que se reflejen las nuevas clases y sus relaciones así como las propiedades y métodos relacionados con el caso de uso, `borrarContactoDeEmpresa`
- Si no entregó la práctica siete, no ha traído la documentación que entregó o no cree que su entrega sea lo suficientemente correcta debe::
 - Crear un diagrama de casos de uso con todos los casos de uso de la aplicación
 - Para el caso de uso `borrarContactoDeEmpresa` debe dar su especificación completa y crear un diagrama de secuencia.
 - Crear un diagrama de clases conceptual con las clases que tengan que ver con la lógica de negocio
 - El diagrama de clases debe reflejar las propiedades y métodos relacionados con el caso de uso, `borrarContactoDeEmpresa`

Si decide resolver el examen eligiendo la primera opción debe entregar una carpeta que contenga la misma documentación que entregó cuando tuvo que entregar la práctica 7 y otra carpeta con el nuevo .zargo y un documento de texto con el caso de uso expandido.

Si elige la segunda opción debe entregar el fichero .zargo con los diagramas comentados y un fichero de texto plano con la especificación completa del caso de uso.

En ambos casos, introduzca los ficheros/carpetas que le he indicado en una carpeta cuyo nombre sea sus dos apellidos seguidos de su nombre, comprímala en formato `.tar.gz` y entreguela en la carpeta que encontrará para ese fin en el escritorio

Examen Final 7-Junio-2016

APELLIDOS Y NOMBRE:.....

Debe comenzar el examen haciendo el test. Dispone de 30 minutos para ello. Una vez lo finalice, debe dejarlo debajo del resto de hojas del examen y seguir con el resto de preguntas. El profesor le indicará cómo debe entregarlo.

Ejercicio 1. Test (4 puntos)

Seleccione la respuesta correcta (sólo una) rodeando su opción con un círculo. Tenga en cuenta que cada respuesta correcta vale 0,4 puntos y que cada fallo resta 0,2 puntos a la puntuación de este ejercicio :

1. ¿Cuál de las siguientes afirmaciones considera que es cierta?:
 - a) La calidad de un producto sólo puede evaluarse cuando se ejecuta.
 - b) El trabajo de un ingeniero informático no finaliza con la entrega del producto.
 - c) Para recuperar retrasos en las entregas de un proyecto basta con aumentar la cantidad de personal.
 - d) La modificación de requisitos no perjudica el cumplimiento de los plazos de entrega.
2. EL paradigma imperativo define un programa como:
 - a) Un grupo de funciones que llaman a funciones
 - b) La respuesta lógica a una expresión lógica mediante reglas lógicas
 - c) Un grupo de objetos interaccionando entre sí.
 - d) Una secuencia de instrucciones que trabajan con datos
3. Los requisitos no funcionales:
 - a) pueden ser muy difíciles de verificar, por lo que, si es posible, deben cuantificarse.
 - b) indican qué es lo que el sistema no debe hacer
 - c) se refieren a características del dominio de aplicación
 - d) no afectan al modelo de desarrollo
4. Dada una aplicación con múltiples servicios descentralizados, ¿cuál de estos modelos arquitectónicos considera que debe seguir?:
 - a) Modelo de capas.
 - b) Modelo cliente/servidor.
 - c) Modelo repositorio.
5. El proceso unificado (UP) se dice iterativo e incremental porque:
 - a) resuelve antes de nada las facetas que generan más riesgos
 - b) se basa en la arquitectura
 - c) construye múltiples versiones añadiendo funcionalidad a las anteriores
 - d) está dirigido por casos de uso

6. En un diagrama de casos de uso un actor es:
 - a) un elemento externo al sistema que interactúa con el sistema
 - b) un elemento del sistema que interactúa con otros elementos del sistema
 - c) un elemento externo del sistema que no interactúa con el sistema
 - d) un elemento del sistema que invoca a otras partes del sistema
7. El diagrama de despliegue o emplazamiento muestra:
 - a) Como se agrupan los componentes de un sistema en subsistemas
 - b) Como se relacionan las clases entre sí
 - c) Como se relacionan los paquetes lógicos con el hardware del sistema
 - d) Como colaboran diversos objetos entre sí
8. A menudo se utilizan las 10 reglas de Nielsen para evaluar interfaces de usuario. ¿Cuál de las siguientes no es una de dichas normas?
 - a) Utilice convenciones del mundo real, no del sistema (ej. usa iconos que recuerden cosas)
 - b) Cumpla convenciones y estándares, los usuarios están habituados a ellas
 - c) Utilice la mínima información en menús y gráficos sencillos
 - d) Limite las posibilidades de opción del usuario para que el uso del programa sea uniforme
9. Denominamos métodos ágiles a
 - a) Scrum
 - b) Aquellos que ponen énfasis en la adaptabilidad pues consideran el cambio de requisitos un aspecto natural del proyecto
 - c) Todos los métodos de desarrollo iterativos e incrementales
 - d) Los métodos basados en escribir los tests antes que el código
10. En la gestión de riesgos se denomina *estrategia de prevención* a aquella que
 - a) Reduce la probabilidad de ocurrencia del riesgo
 - b) Reduce el impacto de la ocurrencia del riesgo
 - c) Se prepara para el peor escenario posible
 - d) Analiza los riesgos cada uno por su cuenta

Ejercicio 2. Cuestiones (1 punto)

Defina de forma breve y concisa los siguientes conceptos:

- Pruebas de caja negra
- Pruebas de caja blanca

Ejercicio 3. Casos de uso (2,5 puntos)

Lo que aparece a continuación es la descripción que dio el Diario de Navarra de la aplicación para móvil que permite pagar la zona azul:

Los ciudadanos deben registrarse previamente en la página web de XXX, donde cumplimentarán los datos personales, el número de tarjeta de crédito o débito y hasta tres matrículas. A continuación, el usuario puede descargarse la aplicación en dispositivos Android, Iphone y Windows Phone desde la que posteriormente se gestiona el pago.

La aplicación permite pagar el estacionamiento, ampliar el tiempo de aparcamiento o anular algunas denuncias.

Pero, ¿cómo sabe el vigilante de la ORA que se ha efectuado el abono mediante un teléfono móvil? Puesto que no existe un tique que justifique el abono, el vigilante introduce en una aplicación interna la matrícula del coche. En ese momento, el programa indica si el conductor ha efectuado el pago mediante la aplicación móvil o no. En caso afirmativo, se comprueba si está dentro de los límites horarios en función del importe realizado. De lo contrario, se procede a realizar la denuncia.

Dada esta información y sabiendo que desde la aplicación del móvil se pueden modificar las matrículas:

- identifique los actores del sistema
- Haga un diagrama de casos de uso
- Para el caso de uso pagarZonaAzul se desea el caso de uso detallado incluyendo un diagrama de colaboración o de secuencia (el que considere más adecuado)

Ejercicio 4. Diagrama de clases (2,5 puntos)

Si lo piensa un poco, el lado servidor de las aplicaciones de mensajería instantánea es, conceptualmente, sencillo. Simplemente permite que los usuarios dados de alta (reconocidos por un número de móvil) y conectados envíen/reciban mensajes a/de otros usuarios dados de alta o a grupos de usuarios dados de alta. También permiten crear grupos de usuarios dados de alta y se aseguran de que un usuario dado de alta cuando se conecta reciba todos los mensajes que se hallan enviados a él o sus grupos mientras no estaba conectado. Los mensajes pueden ser de texto básico o bien imágenes o vídeo que pueden estar asociados a un mensaje de texto.

Dada esta información se solicita un diagrama de clases que represente la lógica de negocio de estas aplicaciones

Examen Práctico Final 7-Junio-2016

Cree un documento que contenga:

- Casos de uso de la aplicación que entregó en la práctica 3
- Diagramas de secuencia de dichos casos de uso
- Un breve comentario indicando qué mejoras considera que deberían incluirse en el diseño del código que entregó en dicha práctica

Examen Final 20-Junio-2016

APELLIDOS Y NOMBRE:.....

Debe comenzar el examen haciendo el test. Dispone de 30 minutos para ello. Una vez lo finalice, debe dejarlo debajo del resto de hojas del examen y seguir con el resto de preguntas. El profesor le indicará cómo debe entregarlo.

Ejercicio 1. Test (4 puntos)

Seleccione la respuesta correcta (sólo una) rodeando su opción con un círculo. Tenga en cuenta que cada respuesta correcta vale 0,4 puntos y que cada fallo resta 0,2 puntos a la puntuación de este ejercicio :

1. ¿Cuál de las siguientes NO es una característica deseable del software:
 - a) El software debe utilizar la mayor cantidad posible de los recursos hardware del usuario .
 - b) El software debe ser fácil de usar.
 - c) El software no debe causar daños en caso de fallo.
 - d) El software debe permitir evoluciones poco costosas.
2. En lenguajes orientados a objeto llamamos herencia a:
 - a) la capacidad de restringir el acceso a los elementos del objeto.
 - b) la posibilidad de crear múltiples objetos de idéntico comportamiento.
 - c) el mecanismo que permite a una clase presentar algunos de los comportamientos programados en otra clase.
 - d) a la existencia dentro de una clase de diversos métodos que se diferencian entre sí sólo por el tipo de sus parámetros.
3. El estudio de viabilidad de un proyecto es
 - a) la fase de análisis
 - b) la fase de obtención de requisitos
 - c) un estudio a partir de requisitos de usuario preliminares que puede recomendar abandonar el proyecto
 - d) un estudio de los requisitos de sistema que puede recomendar indicaciones sobre el presupuesto
4. El análisis estructurado utiliza diversos modelos gráficos para describir el sistema, como:
 - a) diagramas de flujo de datos para modelar el comportamiento.
 - b) diagramas entidad-relación para modelar la funcionalidad
 - c) diagramas de transición de estado para modelar representar la estructura de los datos
 - d) diagramas de casos de uso para modelar las clases
5. El modelo de control basado en eventos considera que:
 - a) los subsistemas responden a solicitudes externas
 - b) los subsistemas colaboran en llevar el control.
 - c) un subsistema se encarga de iniciar y finalizar la ejecución del resto

6. Dentro del proceso unificado (UP) el desarrollo:
 - a) se realiza después de la elaboración (definición de la arquitectura)
 - b) puede llevar a modificar la planificación
 - c) puede llevar a modificar los plazos de entrega
 - d) consiste en la definición de los casos de uso
7. En un diagrama de secuencia representamos:
 - a) Clases y actores
 - b) Clases y mensajes
 - c) Objetos y actores
 - d) Objetos y mensajes
8. El diagrama de paquetes:
 - a) Como se agrupan los componentes de un sistema en subsistemas
 - b) Como se relacionan las clases entre sí
 - c) Como se relacionan los paquetes lógicos con el hardware del sistema
 - d) Como colaboran diversos objetos entre sí
9. ¿Qué tipo de interacción con el usuario considera preferible para un juego para niños pequeños?
 - a) Interfaz mediante comandos
 - b) Interfaz mediante menús
 - c) Interfaz mediante menús
 - d) Interfaz mediante manipulación directa
10. CMMI define diferentes niveles de madurez. Un proceso se denomina gestionado cuando
 - a) No satisface las metas
 - b) Satisface las metas y el equipo conoce el proceso
 - c) Existen planes documentales y monitorización de procedimientos
 - d) Se utilizan procesos de mejora basados en métricas

Ejercicio 2. Cuestiones (1 punto)

Defina de forma breve y concisa (no es necesario usar más de una frase) los siguientes conceptos:

- Ingeniería del software
- Proceso de software

Ejercicio 3. Casos de uso (2,5 puntos)

Se le ha contratado desde el gobierno para desarrollar una aplicación que permita realizar vía web consultas a la ciudadanía (usted se encargará de programar el servidor, la interfaz web la lleva otra empresa). La aplicación permitirá participar en votaciones a todas aquellas personas que formen parte del censo y se den de alta en la aplicación (la aplicación utilizará las bases de datos del estado para cotejar los datos de los que pretendan darse de alta y suministrará una clave única a cada persona dada de alta).

La aplicación permitirá plantear dos tipos de consultas, las propuestas por el gobierno y las propuestas por otros usuarios dados de alta que hayan sido autorizados a proponer votaciones. Todos las personas dadas de alta en la aplicación pueden votar en un referendum propuesto por el gobierno.

Para poder proponer una votación, los usuarios no gubernamentales deben pagar un canon. El pago da derecho a proponer un número limitado de votaciones. Cuando un usuario que haya pagado el canon proponga una votación, esta debe ser autorizada por el administrador del sistema, que dará un plazo para que el resto de usuarios se den de alta en el censo de la votación. Si en dicho plazo el tamaño del censo alcanza un cierto valor, la votación se producirá un mes después. Si no, no se llevará a cabo. El usuario que haya propuesto la votación puede ver cuántos votantes se han apuntado y anular la votación (hasta una semana antes de la fecha final para confeccionar el censo).

Todos los usuarios recibirán un mail informándole de las votaciones propuestas (del gobierno o de otros usuarios) para que se den de alta en algún censo si así lo desean. Cuando llegue la fecha de una votación en la que puedan participar, podrán votar. También podrán ver los resultados de aquellas votaciones en las que se han apuntado.

Se pide:

1. Diagrama de casos de uso del sistema
2. Descripción detallada (caso de uso expandido) de algún caso de uso incluyendo un diagrama de secuencia para el flujo principal.

Ejercicio 4. Diagrama de clases (2,5 puntos)

Una empresa de transporte de mercancías quiere implantar un sistema de gestión de nóminas y almacen. La empresa distribuye a su personal en diferentes niveles, almaceneros, transportistas y jefes de almacén. La aplicación debe gestionar nóminas, por lo que de todo el personal se deben conocer los siguientes datos: Nombre, apellidos, Nif, teléfono y número de cuenta (estos dos últimos datos son modificables). Cada jefe de almacén (uno por almacén) decide los horarios de los almaceneros del almacén y la carga y horario de salida de sus furgonetas (identificadas por una matrícula y con una capacidad diferente cada una). Cada transportista conduce siempre la misma furgoneta.

Las cargas de las furgonetas son paquetes identificados por un código que van dirigidos a un destino. Los almaceneros agrupan los paquetes en función de la ciudad de destino e informan en el sistema de las dimensiones de cada grupo y de los paquetes que contiene para que el jefe los distribuya entre las furgonetas. Obviamente, una furgoneta puede transportar paquetes que vayan a ciudades diferentes, pero una norma de la empresa indica que nunca la carga de una furgoneta puede contener paquetes que vayan a dos provincias diferentes.

Se solicita un diagrama de clases que represente la lógica de negocio. Las clases deben incluir las propiedades y métodos que se obtengan de la descripción dada.

Examen Práctico Final 20-Junio-2016

Su cliente ha quedado relativamente satisfecho con el trabajo de simulación de supermercado que le entregó. Pero lo solicita una pequeña modificación:

Su cliente quiere que por cada producto que se instancie, el simulador decida qué producto es y localice el precio del producto en una base de datos del supermercado. También quiere que, a la salida, cada caja indique información sobre el dinero que ha recaudado en la simulación. Para ello le basta con saber que debe existir una clase que haga de interfaz con la base de datos. Dicha clase, a la que vamos a llamar **IBDD** debe instanciarse cuando arranque la aplicación. Dispondrá de dos métodos públicos que usted va a utilizar:

1. `public int getCodigo();` Este método devuelve un valor entero que coincide con el código de un producto del supermercado. El valor que devuelve sigue una distribución acorde a las ventas habituales del supermercado.
2. `public float getPrecioProducto(int codigo);` Este método devuelve un valor real que coincide con el precio unitario de los productos del código indicado

Cuando recibe esta solicitud, usted recuerda que la aplicación de supermercado que entregó en su día no cumplía los estándares de calidad que se le suponen a su empresa, por lo tanto decide:

1. Modificar los documentos de diseño incluyendo las nuevas características en el diseño de su simulador.
2. modificar el código de sus clases incluyendo, donde sea necesario, las llamadas a los métodos de **IBDD** para incluir la información económica en el simulador
3. Mejorar la calidad del software entregado anteriormente

Para entregar su examen

1. Cree una carpeta con nombre **Apellido1Apellido2Nombre**
2. Guarde en dicha carpeta: el documento de análisis y diseño de su aplicación y el proyecto de netBeans con el código
3. Comprima la carpeta en formato **.tar.gz**
4. Depositela en la carpeta de entrega

Examen Final 19- enero-2017

APELLIDOS Y NOMBRE:.....

Debe comenzar el examen haciendo el test. Dispone de 30 minutos para ello. Una vez lo finalice, debe dejarlo debajo del resto de hojas del examen y seguir con el resto de preguntas. El profesor le indicará cómo debe entregarlo.

Ejercicio 1. Test (4 puntos)

Seleccione la respuesta correcta (sólo una) rodeando su opción con un círculo. Tenga en cuenta que cada respuesta correcta vale 0,4 puntos y que cada fallo resta 0,2 puntos a la puntuación de este ejercicio :

1. ¿Cuál de las siguientes afirmaciones sobre los casos de uso considera que es cierta?:
 - a) Los verificadores especifican los requisitos de los casos de uso.
 - b) Los casos de uso se usan para planificar iteraciones.
 - c) Analistas y diseñadores localizan los casos de uso a partir de las clases.
 - d) El modelo de casos de uso debe obtenerse antes de la captura de requisitos.
2. Existen diferentes (falsos) mitos sobre el software que dificultan su desarrollo. ¿Cual de estas afirmaciones es uno de ellos?
 - a) La calidad del producto software se puede comprobar antes de que se haya ejecutado
 - b) Un proyecto retrasado puede cumplir su planificación si se le asigna más personal.
 - c) La variación de requisitos a mitad de proyecto siempre supone un problema
 - d) No se puede empezar a programar en un proyecto hasta que el cliente no ha dejado muy claro lo que quiere
3. Cuando hablamos de herramientas CASE nos referimos
 - a) A herramientas que ayudan a codificar (por ejemplo NetBeans).
 - b) A herramientas que facilitan el análisis y diseño (por ejemplo visual Paradigm)
 - c) A herramientas que facilitan el análisis, diseño y desarrollo de software
 - d) A herramientas que facilitan el uso de distintas metodologías propias de la ingeniería del software
4. En los lenguajes en los que existe el concepto de herencia suele aparecer el concepto de polimorfismo que consiste en :
 - a) Sobrecargar los métodos de la superclase de modo que se comporten de muchas formas
 - b) Sobrecargar los métodos de la subclase de modo que se comporten de muchas formas.
 - c) Reprogramar métodos en la subclase y permitir que acceda también al comportamiento de la superclase.
 - d) Reprogramar métodos en la subclase haciendo inaccesible el comportamiento de la superclase.
5. El diagrama entidad-relación es:
 - a) Un diagrama UML que permite representar la funcionalidad de una clase
 - b) Un diagrama que muestra como se relacionan las diferentes entidades de una empresa
 - c) Un diagrama que se usaba en desarrollo imperativo y que no tiene ninguna utilidad en la actualidad
 - d) Un diagrama que se utiliza para representar los datos que se han de almacenar de modo persistente

6. ¿Cuál de las siguientes NO es una característica del proceso unificado de desarrollo (UP) :
- a) Es Iterativo e Incremental
 - b) Está dirigido por sprints
 - c) Está centrado en la arquitectura
 - d) Se focaliza en los riesgos
7. El diagrama de paquetes (UML) muestra:
- a) Como se agrupan los componentes de un sistema en subsistemas
 - b) Como se relacionan las clases entre sí
 - c) Como se relacionan los paquetes lógicos con el hardware del sistema
 - d) Como colaboran diversos objetos entre sí
8. ¿Cuál de entre las siguientes pruebas deben realizarse antes?
- a) Pruebas de integración
 - b) Pruebas de validación
 - c) Pruebas de unidades
 - d) En la actualidad se realizan todas a la vez, una vez finalizado el desarrollo
9. Dentro de la Ingeniería del software denominamos *Control de calidad a*
- a) Los procesos que vigilan el desarrollo para comprobar que el software verifica sus requisitos funcionales
 - b) La definición de un plan de calidad
 - c) Los procesos que vigilan el desarrollo para comprobar que se cumplen los estándares
 - d) Los sistemas de mejora de procesos
10. El modelo en espiral
- a) Es un nombre alternativo para el proceso unificado (UP)
 - b) Más que un modelo de desarrollo es una forma de pensar adaptable a cualquier actividad del ciclo de vida del software
 - c) Es un modelo antiguo que no permitía iteraciones ni vueltas atrás
 - d) Consiste en construir múltiples prototipos

Ejercicio 2. Cuestiones (1 punto)

Defina de forma breve y concisa los siguientes conceptos:

- Planificación de riesgos
- Supervisión de riesgos

Ejercicio 3. Análisis y Diseño: La biblioteca (2,5 puntos)

La biblioteca de la universidad quiere una nueva aplicación para gestionar sus fondos documentales. En esencia la aplicación debe permitir:

- Que los empleados de la biblioteca den de alta/baja nuevos documentos en el sistema
- Que los usuarios de la biblioteca puedan coger prestados libros de modo que quede registrado donde está cada libro

La biblioteca presta documentos a distintos tipos de usuarios, cada uno con unas condiciones diferentes:

- Los profesores de la universidad pueden coger documentos sin un plazo de entrega
- Los alumnos de la universidad pueden coger documentos, pero tienen que devolverlos en un plazo determinado (por ejemplo una semana)
- La biblioteca también puede prestar documentos a otros usuarios no relacionados con la universidad. En este caso hay un plazo de devolución más corto.

El sistema de préstamo para profesores y alumnos de la universidad es sencillo. Simplemente hay que utilizar la tarjeta de la universidad para activar un lector donde se lee el código de barras del libro. El sistema imprime un documento en el que se indica la fecha de devolución. Obviamente, antes de autorizar el préstamo el sistema comprueba que la tarjeta es válida accediendo al sistema informático de la universidad. Para los otros usuarios el sistema es más complejo. En primer lugar tienen que darse de alta en el sistema proporcionando a los empleados de la biblioteca sus datos, que se almacenarán en una base de usuarios externos del sistema. En el proceso de darles de alta, el empleado les proporcionará una tarjeta con la que pueden pedir libros igual que profesores y alumnos

El sistema no permite que un usuario que tenga un documento cuyo plazo de préstamo ha expirado pida prestado un nuevo documento.

Para devolver los libros prestados el usuario debe entregarlos a los empleados de la biblioteca, que se encargarán de informar al sistema de la devolución una vez hayan dejado el documento en el lugar que le corresponda. Además de esto y de dar de alta a los usuarios externos, los empleados de la biblioteca se limitan a informar a los usuarios de la situación de documentos concretos (prestados, prestados a profesor, en estantería o no lo tenemos). Cada cierto tiempo solicitarán al sistema un listado con los documentos prestados que deberían haber sido ya devueltos para ponerse en contacto con los usuarios que los tienen.

Se pide:

- Identifique los actores del sistema
- Haga un diagrama de casos de uso del sistema
- Haga un diagrama de clases que represente la lógica de negocio del sistema de la biblioteca. Deben aparecer los atributos que usted considere lógicos
- Seleccione un caso de uso relacionado con el préstamo de documentos. Describalo como caso de uso detallado incluyendo un diagrama de colaboración o de secuencia (el que considere más adecuado).
- Incluya en el diagrama de clases anterior todas las clases de diseño que obtenga del diseño de dicho caso de uso. Incluya también los métodos que se obtengan

Examen Práctico Final 19- enero-2017

Si usted está aquí es por una de las siguientes razones:

- No me ha enviado las prácticas que le he pedido (quizá se le pasó)
- No he conseguido descomprimir de forma correcta sus ficheros y, por tanto, no se lo que ha enviado (quizá hubo un problema de formato/transmisión)
- No he conseguido compilar el código que me ha enviado (quizá no me envió la versión correcta)
- No he conseguido ejecutar sin errores el código que me ha enviado (quizá no me envió la versión correcta)

En vista de ello, quiero chequear todo su trabajo. Por ello tiene que

1. Crear una carpeta con su nombre
2. Copiar en ella todo lo que haya hecho relacionado con las prácticas de Ingeniería del software organizado en carpetas llamadas **Practica1**, **Practica2**, etc. Asegurese que el material que incluye es suyo
3. Comprimir en formato **.tar.gz** la carpeta
4. Entregarla

Tiene exactamente media hora.

Examen Final 9-enero-2018

APELLIDOS Y NOMBRE:.....

Debe comenzar el examen haciendo el test. Dispone de 30 minutos para ello. Una vez lo finalice, debe dejarlo debajo del resto de hojas del examen y seguir con el resto de preguntas. El profesor le indicará cómo debe entregarlo.

Ejercicio 1. Test (4 puntos)

Seleccione la respuesta correcta (sólo una) rodeando su opción con un círculo. Tenga en cuenta que cada respuesta correcta vale 0,4 puntos y que cada fallo resta 0,2 puntos a la puntuación de este ejercicio :

1. La ingeniería del Software puede definirse como:
 - a) La aplicación metodológica de la ciencia de la computación.
 - b) La metodología que permite codificar algoritmos de forma eficiente.
 - c) La metodología que permite codificar algoritmos eficientes.
 - d) La aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software.
2. En lenguajes orientados a objeto llamamos clase a:
 - a) un modelo que define las variables y métodos comunes a todos los objetos de un cierto tipo.
 - b) un modelo que define las variables de clase de los objetos de un cierto tipo.
 - c) un conjunto de módulos que están fuertemente interrelacionados.
 - d) un objeto.
3. El estudio de viabilidad de un proyecto
 - a) debe realizarse antes de comenzar la obtención de requisitos del sistema
 - b) se realiza a partir de requisitos de usuario preliminares
 - c) se realiza a partir de requisitos de sistema preliminares
 - d) se realiza una vez se han obtenido los requisitos de sistema a partir de los de usuario
4. Un diagrama entidad/relación muestra:
 - a) cómo se transforman los datos del sistema.
 - b) cómo se modifica el estado del sistema
 - c) cómo son los datos del sistema y cómo se relacionan entre ellos
 - d) cómo se relacionan los datos del sistema con las entidades que tienen estados
- 5.Cuál de las siguientes NO es una característica del proceso unificado (UP):
 - a) es iterativo e incremental.
 - b) está organizado en cascada.
 - c) está centrado en la arquitectura.
 - d) Se focaliza en los riesgos.

6. De los siguientes diagramas que forman parte del estandar UML 2.x, ¿cuál es de comportamiento?:
 - a) Diagrama de Clases
 - b) Diagrama de Casos de Uso
 - c) Diagrama de Secuencia
 - d) Diagrama de Paquetes
7. ¿Cual de las siguientes prácticas está en contra del código ético para los Ingenieros del Software publicado por ACM/IEEE-CS Joint Task Force?:
 - a) Pedir la opinión de otros profesionales en aquellas áreas que estén fuera de nuestra competencia
 - b) Violar alguna ley referida a nuestra profesión si dicha ley es inconsistente con el interés público
 - c) Utilizar en beneficio propio los medios materiales de un cliente
 - d) Revelar a las autoridades los peligros potenciales del software que se esté desarrollando
8. Un buen diseño de la interfaz de puede :
 - a) empeorar la confiabilidad del sistema.
 - b) reducir el número de errores del usuario
 - c) reducir el número de usuarios del sistema
 - d) aumentar el coste del proyecto sin mejorar sus posibilidades de venta
9. Las características de los programas orientados a eventos incluyen:
 - a) El programa, al ser reactivo, deja muy poco tiempo libre al procesador
 - b) El usuario debe esperar a que el programa le solicite datos
 - c) El control del programa es centralizado
 - d) La gestión de eventos debe ser rápida
10. Las inspecciones de software:
 - a) son un metodo de validación del código
 - b) son un método de verificación válido a cualquier documento
 - c) son el método de verificación más usado pues, aunque son poco eficientes, son muy baratas
 - d) Se utilizan tan sólo en procesos de mejora basados en métricas

Ejercicio 2. Cuestiones (1 punto)

Defina de forma breve y concisa los siguientes conceptos:

- Modelo de proceso en cascada
- Modelo de proceso iterativo e incremental

Ejercicio 3. Un caso práctico (5 puntos)

Le han contratado para programar el subsistema de cobro de un supermercado. En el supermercado hay un conjunto de cajas. Cada caja está conectada a un lector de códigos de barras de los productos, a un sistema de identificación del personal de la tienda, a una pantalla para ver los productos que la cajera va cobrando, a un sistema de pago por tarjeta y a un sistema de impresión para imprimir los recibos que las cajeras entregan a los clientes. Las cajas también se conectan con las base de datos que el supermercado emplea para almacenar la información de sus empleados y sus ventas.

Para abrir una caja, un empleado del supermercado debe identificarse mediante nombre de usuario y clave en la caja. Si la combinación es correcta (en función de la información recogida en la base de datos de empleados), la caja puede comenzar a usarse. Si el empleado es un cobrador, lo único que puede hacer es comenzar a cobrar a clientes. Para cobrar a un cliente los cobradores van identificando los productos comprados. Pueden hacerlo de dos modos: pasando el código de barras por el lector de códigos (que se activa al abrir la caja) o pulsando una tecla de la caja para indicar que se va a introducir manualmente un código. En ambos casos, el cobrador puede escribir antes un número indicando que el código que se va a introducir después se debe cobrar multiplicado por dicho número. El cobrador ve por una pantalla el resultado de la lectura (código de producto, precio unitario del producto, número de unidades y precio total). Si comprueba que ha habido un error, el cobrador puede eliminar un producto pulsando una tecla de la caja y volviendo a introducir el código del producto (indicando antes, si es distinto de uno, el número de instancias del producto que deben eliminarse). Para acabar de cobrar a un cliente el cobrador pulsa en otro botón de la caja con lo que se habilitan el terminal de pago mediante tarjeta y el sistema de apertura de la caja. Si el pago es por tarjeta, el cliente inserta la tarjeta en el terminal de pago y la caja cobra al banco. Si el pago es en metálico, el cobrador cobra e indica pulsando otra tecla que se ha realizado el pago. En ambos casos, una vez se ha realizado el pago el sistema imprime un ticket en el que por cada artículo comprado se refleja su nombre, el número de instancias compradas, el precio unitario y el precio total. El ticket también indica el total de la compra, el cajero que ha realizado el cobro y un código para identificar la venta. El sistema también vuelca en las bases de datos de la tienda la información de los productos vendidos, el cobrador que ha emitido el ticket y el método de pago.

Los jefes de caja, además de cobrar, también pueden realizar devoluciones. Para ello deben indicar que van a realizar dicha operación, tras lo cual por cada producto que se vaya a devolver deben introducir el código del ticket que se va a modificar y el código del producto (indicando antes el número de instancias del producto). Para finalizar deben pulsar una tecla de fin de devolución. Si el ticket indicaba que el ticket se había abonado en metálico se abre la caja para que el jefe de caja pueda devolver el dinero; en caso contrario el sistema ingresa directamente en la cuenta del cliente la cantidad que deba devolversele e informa al jefe de caja que se ha hecho el ingreso. Para finalizar la devolución el jefe de caja debe indicar que vuelve al modo normal de facturación.

Se pide:

1. Diagrama de casos de uso del sistema
2. Diagrama de clases que represente las clases conceptuales, aquellas que aparecen en la descripción que le he dado del sistema) con sus atributos
3. Elija un caso de uso y haga una descripción detallada
4. Diagrama de secuencia del caso de uso que haya detallado
5. Modifique el diagrama de clases anterior incluyendo los métodos que hayan surgido durante el diseño del caso de uso.

Examen Final 19-enero-2018

APELLIDOS Y NOMBRE:.....

Ejercicio 1. Test (4 puntos)

Seleccione la respuesta correcta (sólo una) rodeando su opción con un círculo. Tenga en cuenta que cada respuesta correcta vale 0,4 puntos y que cada fallo resta 0,2 puntos a la puntuación de este ejercicio:

1. La validación de requisitos consiste en:
 - a) comprobar si merece la pena desarrollar el sistema.
 - b) localizar los requisitos de usuario.
 - c) definir los requisitos del sistema.
 - d) comprobar si hemos especificado lo que quería el cliente.
2. Entre los modelos arquitectónicos básicos, el modelo cliente/servidor
 - a) es útil en sistemas con grandes cantidades de datos compartidos por diferentes subsistemas
 - b) es ideal para aplicaciones con múltiples servicios deslocalizados.
 - c) es muy adecuado para sistemas que se desarrollan incrementalmente
 - d) es el modelo de referencia para aplicaciones guiadas por interrupciones
3. Una de las reglas de Nielsen para evaluar la interfaz de usuario indica que
 - a) debemos emplear un lenguaje formal para evitar interpretaciones erróneas.
 - b) debemos limitar al máximo la libertad de elección del usuario para que no se equivoque
 - c) debemos utilizar un diseño detallista, con gráficos complejos que atraigan el interés del usuario
 - d) debemos ser consistentes con otras aplicaciones/versiones para aprovechar los recuerdos del usuario
4. El modelo conceptual del sistema trata de reflejar la lógica de negocio. Representa clases conceptuales como:
 - a) los actores del sistema (Empleado, Cajero, ...)
 - b) el hardware del sistema (LectorDeCodigos, LectorDeTarjetas, ...).
 - c) partes de una transacción (LineaDeProducto, FormaDePago, ...)
 - d) la utilización de métodos de un objeto por parte de otro (consultarPrecio(), aniadirLinea(), ...).
5. El plan de pruebas es un documento que debe indicar:
 - a) qué herramientas deben usarse para realizar el diseño del sistema
 - b) cuándo se debe implementar cada uno de los requisitos
 - c) quiénes son los encargados de realizar las inspecciones del código
 - d) cómo y cuándo debe el cliente dar el visto bueno a las pruebas realizadas

6. La calidad de un proyecto se consigue gracias:
 - a) al cumplimiento de un plan de calidad basado en estándares
 - b) a dedicar mucho tiempo al análisis
 - c) a la utilización de herramientas CASE
 - d) a la realización frecuente de inspecciones
7. Los procedimientos de gestión de configuración definen como:
 - a) organizar los recursos materiales y personales de la empresa
 - b) evitar sobrecostos elevados sobre la estimación inicial
 - c) construir el software siguiendo criterios de calidad
 - d) construir y registrar la entrega de las diferentes versiones
8. Según el código ético publicado por la ACM/IEEE Computer Society los ingenieros de software en funciones de dirección o liderazgo:
 - a) deben asignar trabajo entre sus empleados considerando exclusivamente criterios de beneficio empresarial
 - b) no pueden castigar a nadie por expresar preocupaciones éticas sobre un proyecto
 - c) deben ofrecer estimaciones cuantitativas de coste, plazo, personal, calidad y producto sin considerar el nivel de certidumbre
 - d) deben abstenerse de tomar medidas si conocen que se están incumpliendo las políticas del empresario o las indicaciones de este código
9. Los denominados *métodos ágiles* suelen:
 - a) considerar que los requisitos del sistema son casi inmutables
 - b) minimizar la participación de clientes/usuarios en el proceso de desarrollo
 - c) llevar a reescribir muy a menudo el código
 - d) ser muy estrictos
10. Dentro de la gestión de riesgos de un proyecto informático llamamos estrategias de minimización a las que:
 - a) tratan de reducir la probabilidad de que suceda el riesgo analizado
 - b) tratan de reducir el impacto que tiene el riesgo si llegase a suceder
 - c) preparan planes considerando que sucede el peor escenario al que puede llevar la ocurrencia del riesgo
 - d) analizan de modo mínimo los escenarios de riesgo

Ejercicio 2. Cuestiones (1 punto)

Defina de forma breve y concisa los siguientes conceptos:

- Verificación de software
- Validación de software

Ejercicio 3. Análisis y Diseño: Sistema de examen automático (5 puntos)

Estoy pensando en poner en marcha un sistema automático de evaluación de la parte de test de esta asignatura. Para ello necesito una aplicación que genere y corrija automáticamente los exámenes. La aplicación utilizaría una base de datos de preguntas y respuestas. La idea es que yo, el profesor, según vaya avanzando el curso, pueda creando un banco de preguntas y respuestas. Es decir, en cualquier momento pueda ir añadiendo nuevas preguntas y/o respuestas correctas e incorrectas a las preguntas previamente añadidas. El profesor también debería dar de alta en la aplicación a los alumnos están autorizados a usarla (obviamente, la aplicación me indicaría un nombre de usuario y clave por cada alumnos que yo debería suministrar a cada alumno). La aplicación también debería permitir que el profesor defina un horario de examen. En el día y hora definidos, la aplicación permitiría que los alumnos se conectasen y solicitasen un examen. Se generaría un test con preguntas y respuestas diferentes para cada alumno. El alumno iría respondiendo secuencialmente las preguntas hasta que respondiese diez o pasase el tiempo de examen. Posteriormente el profesor podría solicitar a la aplicación que le diese los resultados de los alumnos, bien pidiendo el de un alumno concreto o un listado con todos los alumnos.

Se pide:

- Identifique los actores del sistema
- Haga un diagrama de casos de uso del sistema
- Haga un diagrama de clases que represente la lógica de negocio del sistema de examen. Deben aparecer los atributos que usted considere lógicos
- Uno de los casos de uso que ha escrito es hacerExamen. De una descripción detallada incluyendo un diagrama de secuencia.
- Incluya en el diagrama de clases anterior todas las clases de diseño que obtenga del diseño de dicho caso de uso. Incluya también los métodos que se obtengan

Examen Práctico Final 19-enero-2018

En la practica 7 se le pedía un documento de análisis y diseño de una agenda. Complete usted dichos documentos añadiendo:

1. En el documento de análisis la especificación completa de un nuevo caso de uso (debe estar al final).
2. En el documento de diseño el diagrama de secuencia del nuevo caso de uso y la actualización del diagrama de clases de modo que incluya, además de lo que ya incluía, todo lo necesario para poner en marcha este nuevo caso de uso (deben estar al final).

NOTA: Si usted ha trabajado con la agenda en su casa es probable que la versión de visualParadigm del laboratorio no le permita abrir el fichero que entregó. En tal caso, cree un nuevo fichero en visual paradigm que contenga sólo lo necesario para el nuevo caso de uso.

Para entregar:

1. Cree una carpeta con nombre **Apellido1Apellido2Nombre**
2. Cree dentro de dicha carpeta dos subcarpetas **documentos** y **modelos**
3. Guarde en **documentos** los nuevos documentos de análisis y diseño
4. Guarde en **modelos** el fichero de visualParadigm, el nuevo diagrama de secuencia (en formato pdf o gif) y el diagrama de clases (en formato pdf o gif). El fichero de visualParadigm debe llamarse Agenda2 si es una evolución del que entregó o nuevaAgenda si ha tenido que crear uno nuevo.
5. Comprima en formato .tar.gz la carpeta **Apellido1Apellido2Nombre** y deposite el archivo comprimido en la carpeta de entregas