

Acto 1 (online)

Teoría ISW

Evaluación Teoría

12-11-2020

2h 30 min

Normas a seguir:

- Este acto de evaluación será objeto de grabación con la finalidad tanto de verificar la identidad del estudiante como para la garantía de los derechos de evaluación y revisión de la prueba.
- Las personas que participan en el acto son responsables de garantizar la ausencia de terceras personas no concernidas.
- La Universitat Politècnica de València no es responsable de la grabación incidental de imágenes que afecten a la esfera de vida privada y familiar si no se adoptaron las recomendaciones de configuración de la estancia.

Finalidad:

- Desarrollo de pruebas de evaluación.
- Prestación del servicio Público de educación superior (artículo 1 de la Ley Orgánica de Universidades).
- Responsable: Universitat Politècnica de València.
- Puede ejercer sus derechos de acceso, rectificación, supresión, portabilidad, limitación u oposición al tratamiento desde una cuenta de correo electrónico de la Universitat Politècnica de València puede escribir a dpd@upv.es. No olvide, incluir documentación justificativa si fuera necesaria.

Ingeniería del Software

ETS Ingeniería Informática

DSIC – UPV

Curso 2020-2021

Acto 1 (online)

Evaluación Teoría

12-11-2020

2h 30 min

Normas a seguir:

- Al realizar esta prueba de evaluación ACEPTO la "cláusula de veracidad" por la que no recibiré ni daré ayuda en esta prueba y garantizo la autoría del 100% de las respuestas. El incumplimiento por mi parte de los deberes derivados de las buenas prácticas de honestidad académica podrá dar lugar a la adopción de las medidas contenidas en la Normativa de Integridad Académica del alumnado de la UPV.
- Se grabará el desarrollo del examen
- El/La alumno/a mantendrá su cámara y micrófono conectados en todo momento.
- El/La alumno/a mantendrá su escritorio compartido en TEAMS en todo momento.
- No es posible realizar preguntas durante el examen. Anota todas las consideraciones que tomes al contestar.
- Se utilizará LucidChart para elaborar el diagrama de clases y Visual Studio para el problema de diseño
- Entrega:
 - El/la alumno/a debe entregar un pdf por email enviado a fjaen@upv.es y avisar de la entrega por el chat de la reunión.
 - El archivo incluirá el nombre del/de la alumno/a, las respuestas a las preguntas teóricas, la imagen del diagrama de clases y el código C# del problema de diseño.

Ingeniería del Software

ETS Ingeniería Informática

DSIC – UPV

Curso 2020-2021

NOMBRE:
GRUPO:

Tiempo: 2 horas 30 minutos

Cuestiones (3 puntos)

1. (1 punto) Asocia los siguientes 'problemas' con conceptos de modelado OO (notación UML).
Un mismo problema podría mapearse con varios conceptos, y viceversa. No todo concepto de Modelado tiene porqué usarse en alguno de estos problemas.

Problema		Concepto de Modelado OO
a) Un vehículo debe tener un calificativo ambiental (0, Eco, A, B, C, ...)		1) Atributo de una clase
b) Un coche es un vehículo		2) Relación de Asociación
c) Un vehículo es de un usuario		3) Relación de Agregación
d) Al inscribirse un socio en una actividad, debe indicarse la fecha de inscripción		4) Cardinalidad 1..*
e) Un vehículo debe tener al menos un propietario registrado		5) Especialización/generalización
		6) Atributo de enlace
		7) Cardinalidad 0..1

2. (1 punto) Explica al menos dos diferencias fundamentales de trabajar con una arquitectura multicapa cerrada versus una arquitectura multicapa abierta.
3. (1 punto) Explica breve, pero razonadamente, qué implicaciones tiene que el software sea un elemento lógico.

Problemas (7 puntos)

4. (3.5 puntos) A partir del siguiente enunciado, construir el diagrama de clases en UML que modela el sistema propuesto, incluyendo los atributos de las clases que creáis oportunos, así como los nombres de todas las relaciones que detectéis (**no hay que incluir los métodos ni los tipos de los atributos**).

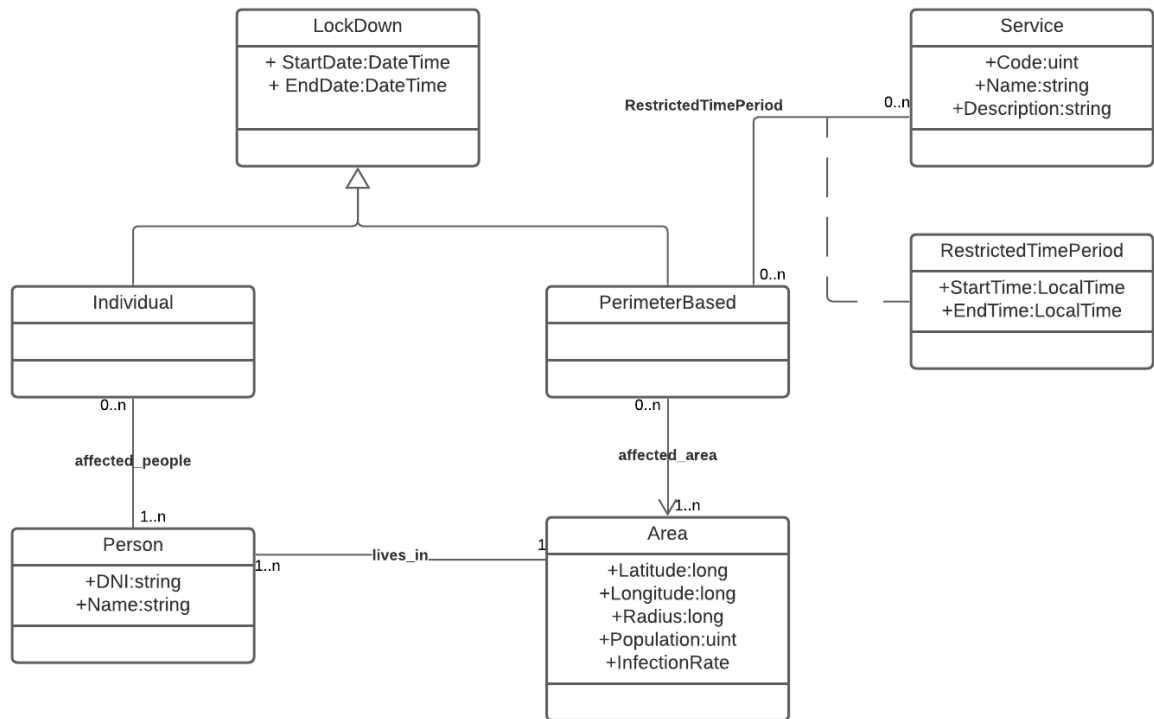
La empresa ISWSoft ha decidido lanzar un nuevo proyecto de desarrollo de software para que los pequeños y medianos comercios puedan adaptarse más fácilmente a las restricciones de movilidad de sus clientes. La aplicación a desarrollar, denominada *RepartoAPP*, gestiona únicamente el reparto a domicilio de los pedidos que los clientes realizan a los comercios por distintos medios (teléfono, WhatsApp, página web, etc.), al que se denomina paquete. En la primera versión de *RepartoAPP* no gestionará el pago de los pedidos, el cliente lo abonará directamente al comercio.

Todo comercio que lo desee puede registrarse en *RepartoAPP*. Para ello debe indicar su nombre, dirección, CIF, actividad a la que se dedica (elegida de entre las opciones que le indica la aplicación: alimentación, hostelería, mercería, ...). Para facilitar la gestión de sus clientes, los comercios irán registrándolos, conforme estos vayan utilizando el servicio, siendo el DNI del cliente la forma de identificarlo (si el cliente ya existe en *RepartoAPP*, no se volverá a registrar). Además del DNI, se indicará su nombre, teléfono, dirección y código postal. El propio comercio, cuando debe entregar un paquete a un cliente, lo introduce en la aplicación, diferenciando si es paquete urgente o no. En el caso de paquete urgente, se indica la hora acordada de entrega (en el mismo día) y el importe adicional (extra) que el comercio abonará al repartidor sobre la tarifa fijada. En caso de un paquete no urgente, se le asigna una franja de entrega acordada previamente con el cliente. En ambos casos, el paquete se etiqueta como pequeño, mediano o grande, en función de su peso y dimensiones, y si es frágil o no. También se puede incluir una pequeña descripción.

Por otra parte, los repartidores también pueden registrarse en la aplicación indicando su DNI, alias, teléfono y sus zonas de reparto (identificadas como un código postal) en las que ofrecen sus servicios como repartidores. Si el repartidor dispone de vehículo propio (moto, coche o furgoneta) para el reparto, indicará la matrícula y el peso máximo del paquete a repartir, pudiendo, en este caso, indicar más de un código postal como zona de reparto. Los repartidores que no disponen de vehículo propio sólo podrán realizar entregas de paquetes de tamaño pequeño en una única zona de reparto (código postal). Además, cada repartidor deberá tener un repartidor sustituto, que se haría cargo de sus entregas de paquetes en caso de cualquier imprevisto. Un sustituto puede serlo como máximo de un repartidor.

RepartoAPP implementa el algoritmo de asignación de paquetes a los repartidores, en función de su zona o zonas de reparto y su disponibilidad, nada más se crea el paquete. La disponibilidad será indicada por el propio repartidor en tiempo real. Si está disponible, se le podrán ir asignando paquetes, teniendo información en todo momento de los paquetes asignados, pero por recoger en tienda; de los paquetes recogidos, pero aún no entregados; del paquete actual que está en reparto; y finalmente, los paquetes ya entregados, en cuyo caso se deberá registrar la fecha y hora de entrega exactas. El comercio podrá visualizar el estado de sus paquetes y el repartidor asignado. El administrador de *RepartoAPP* ordenará las liquidaciones de pago a los repartidores y cobro a los comercios cada 15 días.

5. (3.5 puntos) A partir del siguiente fragmento de diagrama de clases en UML:



Nota 1. Existe una restricción de navegabilidad entre *PerimeterBased* y *Area*

Nota 2. Solo se piden los atributos, no los métodos.

Nota 3. *LockDown* es una clase abstracta.

- (1.5 puntos) Realiza el diseño en C# atendiendo a los patrones vistos en clase.
- (1 punto) Proporciona la cabecera de los métodos constructores necesarios (sin la implementación del cuerpo del método).
- (1 punto) Implementa el código necesario para crear un *LockDown* de tipo *PerimeterBased* con un servicio restringido y que afecta a un área. El sistema ha de tener también otro *LockDown* de tipo individual que afecta solo a una persona. Inventa el valor de todos los parámetros necesarios (**solo los imprescindibles**) para dejar el sistema en un estado consistente tras dicha creación.