



88147025



INFORMÁTICA
NIVEL MEDIO
PRUEBA 2

Martes 18 de noviembre de 2014 (mañana)

1 hora

INSTRUCCIONES PARA LOS ALUMNOS

- No abra esta prueba hasta que se lo autoricen.
- Conteste todas las preguntas de una de las opciones.
- La puntuación máxima para esta prueba de examen es *[45 puntos]*.

Opción	Preguntas
Opción A — Bases de datos	1–3
Opción B — Modelos y simulaciones	4–6
Opción C — Ciencia de la Web	7–9
Opción D — Programación orientada a objetos	10–12

Opción A — Bases de datos

1. (a) (i) Describa, utilizando un ejemplo, la diferencia entre una operación y una transacción. [3]
- (ii) Identifique un motivo que haría necesario revertir una transacción. [1]
- (iii) Indique qué efecto tiene revertir una transacción. [1]

Un pequeño banco va a usar una base de datos para procesar depósitos y retiros de fondos, consultas sobre saldos de cuentas y la creación de nuevas cuentas.

- (b) Resuma **dos** razones para usar una base de datos en esta aplicación. [4]
- (c) Explique, ofreciendo ejemplos, cómo podría usar esta aplicación la validación y la verificación de datos al procesar un depósito. [4]
- (d) Discuta, dando un ejemplo, qué importancia tiene el aislamiento (la propiedad ACID) en esta aplicación. [3]

2. Una compañía vende a través de Internet sombreros fabricados a mano. Están planeando implantar un sistema de gestión de bases de datos (SGBD) como parte de su tienda en línea. Los datos que la compañía almacena para cada sombrero son: tamaño, precio y nombre de la persona que lo fabricó.

- (a) Identifique los tipos de datos para los datos de cada sombrero. [3]
- (b) Identifique **tres** capacidades que deberán soportar las herramientas del SGBD. [3]

Una de las características de un SGBD es la seguridad.

- (c) En el contexto anterior, discuta **dos** características de seguridad necesarias para este SGBD. [4]
- (d) Resuma la naturaleza de un diccionario de datos. [2]
- (e) Explique la importancia que tiene el modelado de datos en el diseño de una base de datos. [3]

(La opción A continúa en la página siguiente)

(Opción A: continuación)

3. Una librería le ha encargado el diseño de una base de datos para mantener un registro de sus libros. Para ello, ofrecen la descripción siguiente:

- cada **libro** se introduce en la base de datos con un título, un autor, una fecha de publicación, un precio de lista y el número de copias que hay en la librería
- cada **estantería** contiene estantes y tiene un número de pasillo y de estantería (por ejemplo, la estantería 23 tiene 5 estantes y está en el pasillo 4)
- cada **estante** contiene títulos de libros
- un **registro de venta** contiene el título de un libro, el nombre del cliente, el precio pagado y la fecha de la venta.

- (a) Describa qué características tiene una base de datos normalizada. [2]
- (b) Resuma los problemas generados por los datos redundantes. [2]
- (c) Elabore un diagrama entidad-relación (ERD) para la base de datos de la librería que represente de manera exacta la información de los cuatro puntos mencionados anteriormente. [3]
- (d) (i) Identifique un componente del ERD que no sea coherente con la tercera forma normal (3FN). [1]
- (ii) Resuma qué se podría hacer para transformar el ERD en 3FN. [3]
- (e) Discuta **un** problema relacionado con la privacidad que podría surgir si la policía accediera a los datos en la base de datos. [3]

Fin de la opción A

Opción B — Modelos y simulaciones

4. La cantidad de antibiótico suministrado a un paciente en cada dosis depende del peso, en kg, y de la edad del paciente al inicio del tratamiento.

La cantidad suministrada a cada paciente se calcula de la forma siguiente.

Edad	Cantidad de medicina por kg
18 – 40	1,50 mg
41 – 60	1,75 mg
más de 60	2 mg

- (a) Calcule la cantidad de cada dosis para un paciente de 20 años y 50 kg. [1]
- (b) Indique las variables y el cálculo necesario para modelar la cantidad necesaria para un paciente. [2]

Considere el caso en el que hay que suministrar la medicina cada 8 horas durante 10 días.

- (c) Elabore un algoritmo que use este modelo para crear una lista de días, horas y la cantidad en cada dosis para un paciente, suponiendo que la primera dosis se suministra a las 8.00 de la mañana del primer día. La lista debe referirse a las fechas como día1, día2, etc y la hora debe aparecer en formato de 24 horas. [4]

Con el paso del tiempo el cuerpo del paciente absorbe parte de la medicina. Justo **antes** del siguiente tratamiento, la cantidad de medicina en el cuerpo es del 20% de la que estaba presente inmediatamente **después** del último tratamiento. Es necesario llevar un registro de las cantidades de medicina que hay en el cuerpo en cada fase del tratamiento.

- (d) (i) En relación con cualquier software con el que esté familiarizado, explique cómo se podría construir un modelo para calcular la cantidad de medicina en la sangre de un paciente concreto inmediatamente **después** de suministrar cada dosis. [4]
- (ii) Identifique qué efecto tienen los errores de entrada que pueden surgir en este modelo. [1]
- (iii) Sugiera una forma adecuada de comprobar la precisión del modelo. [3]

(La opción B continúa en la página siguiente)

(Opción B: continuación)

5. Una nueva localidad ha aumentado la población un 7% respecto del año pasado, mucho más de lo esperado. Ahora tiene 25 000 habitantes. A efectos de planificación, es importante predecir el tamaño futuro de la población de la localidad.
- (a) Indique cómo se podría realizar una simulación para predecir la población en los próximos 10 años usando sólo la tasa actual de crecimiento. [1]
 - (b) Identifique **tres** factores que afecten al crecimiento de la población de una localidad. [3]
 - (c) Resuma una forma de mejorar la simulación usando los tres factores identificados en la parte (b). [3]
 - (d) Discuta la precisión de esta simulación del crecimiento de la población. [4]
 - (e) Prediga **dos** consecuencias sociales de una simulación imprecisa del crecimiento de la población en una localidad. [4]
6. El diseño asistido por computador incorpora la visualización en 2D y 3D. Un fabricante de coches decide modificar un diseño actual para incorporar características adicionales. El diseñador empieza cargando una versión del *wireframe* del diseño actual.
- (a) Resuma cómo un gráfico del *wireframe* representa el diseño del vehículo. [2]
 - (b) (i) Identifique **un** cambio que se podría hacer en el diseño del vehículo usando el *wireframe*. [1]
 - (ii) Resuma cómo se podría manipular el *wireframe* para hacer este cambio. [2]
 - (iii) Resuma cómo podría interactuar el diseñador con el computador para conseguirlo. [2]
- Después de realizar los cambios en el diseño básico se pueden realizar cambios en el interior y el exterior del vehículo.
- (c) Identifique **dos** cambios en el interior y el exterior que podría realizar el diseñador. [2]
- Para realizar estos cambios el diseñador necesita una vista diferente.
- (d) Describa las técnicas de visualización que se podrían aplicar para convertir el *wireframe* en una vista en 3D del vehículo, tal y como aparecería una vez fabricado. [6]

Fin de la opción B

Opción C — Ciencia de la Web

7. Un estudiante del Reino Unido está leyendo una página Web de un periódico con sede en Sudáfrica.

(a) Utilizando este ejemplo, distinga entre Internet y la World Wide Web. [3]

Muchos periódicos tienen una versión en Internet a través de la cual los usuarios pueden leer las noticias.

(b) Identifique otros **dos** medios electrónicos usados por los periódicos para ofrecer información mediante la tecnología proporcionada por la evolución de la Web. [2]

El Localizador Uniforme de Recursos (URL) se usa para identificar un recurso concreto de Internet. El siguiente es un ejemplo de URL extraído de un periódico internacional

<http://www.southafricantimes.com/football/mon/rt>

(c) Usando el ejemplo del URL anterior, identifique **tres** características de una URL. [3]

Una página HTML de un sitio Web concreto muestra un formulario que contiene botones de radio. Estos botones permiten que el usuario elija una opción concreta. Esta opción determinará el contenido de la siguiente página Web que se va a entregar desde el servidor del sitio Web.

(d) En relación directa con la interfaz de pasarela común (CGI), explique cómo se realiza este proceso. [4]

(e) A medida que aumenta el número de sitios Web, el número de direcciones también debe aumentar. Sugiera, aportando razones, qué protocolo se ha actualizado para que esto pueda ocurrir. [3]

(La opción C continúa en la página siguiente)

(Opción C: continuación)

8. Se han publicado en línea dos nuevas revistas científicas.

- (a) Haciendo referencia al contenido y los enlaces, describa cómo un motor de búsqueda que use el algoritmo HITS podría generar índices de relevancia de páginas diferentes para las dos publicaciones. [4]

Ambas publicaciones permiten que se descargue contenido. Este contenido puede ser, por ejemplo, texto o imágenes.

- (b) Resuma por qué se podrían usar diferentes técnicas de compresión para la transferencia de esta información. [4]

- (c) Explique, con un ejemplo, cómo los mismos términos de búsqueda en el mismo motor de búsqueda podrían generar distintos resultados para distintos usuarios. [3]

- (d) Identifique **dos** tipos distintos de sistemas distribuidos, ofreciendo un ejemplo de cómo se podría usar cada uno. [4]

9. Actualmente se están comercializando computadores portátiles que usan únicamente la informática en la nube para todos los servicios y aplicaciones.

- (a) (i) Identifique **dos** características de este tipo de computadores portátiles en relación con la informática en la nube. [2]

- (ii) Resuma **dos** posibles desventajas de este tipo de computador portátil para el usuario. [4]

- (b) Discuta, considerando **un** aspecto positivo y **un** aspecto negativo, las consecuencias implícitas de que la Web no esté controlada de forma centralizada. [5]

- (c) Para **un** ejemplo de propiedad intelectual, evalúe **una** medida que se haya tomado para la protección de la misma en la Web. [4]

Fin de la opción C

Página en blanco

Opción D — Programación orientada a objetos

Una empresa de transporte de paquetes usa trenes en sus operaciones. Usa un programa orientado a objetos para realizar un seguimiento de los trenes y de los paquetes que entrega.

La empresa tiene varios objetos en su programa, aquí están algunos de ellos.

Objeto	Descripción
Tren	Cada <code>Tren</code> está formado por objetos <code>MaterialRodante</code> , cada uno de los cuales es un <code>Vagón</code> o una <code>Máquina</code> .
<code>MaterialRodante</code>	Un objeto <code>MaterialRodante</code> puede ser una <code>Máquina</code> (que puede remolcar) o un <code>Vagón</code> (que necesita ser remolcado). Cada <code>MaterialRodante</code> tiene un número de identificación único (ID) y un peso.
<code>Máquina</code>	Una variedad de <code>MaterialRodante</code> . Cada <code>Máquina</code> tiene un peso máximo que puede remolcar.
<code>Vagón</code>	Una variedad de <code>MaterialRodante</code> . Cada <code>Vagón</code> tiene un peso máximo de carga.
<code>Paquete</code>	Cada <code>Paquete</code> está etiquetado con un número de seguimiento, las direcciones de origen y destino y su peso.

El código de las páginas siguientes implementa la clase `Tren` que se usa en este programa.

(La opción D continúa en la página siguiente)

(Opción D: continuación)

```

public class Tren
{
    private Máquina[] mMáquinas;
    private Vagón[] mVagones;
    private int mNúmeroMáquinas;
    private int mNúmeroVagones;
    private int mNúmeroTren;
    private double mPeso;           // Peso total en kilogramos
    public Tren(int número)
    {
        mNúmeroTren = número;
        mMáquinas = new Máquina[6]; // El tren puede tener un máximo de 6
                                     // máquinas
        mNúmeroMáquinas = 0;
        mVagones = new Vagón[100]; // El tren puede tener un máximo de 100
                                     // vagones
        mNúmeroVagones = 0;
        mPeso = 0;
    }
    public void agregarMáquina( Máquina nuevaMáquina )
    {
        mMáquinas[mNúmeroMáquinas] = nuevaMáquina;
        mNúmeroMáquinas++;
    }
    public Máquina quitarMáquina()
    {
        mNúmeroMáquinas--;
        return mMáquinas[mNúmeroMáquinas];
    }
    public void agregarVagón( Vagón nuevoVagón )
    {
        mVagones[mNúmeroVagones] = nuevoVagón;
        mNúmeroVagones++;
    }
    public Vagón quitarVagón()
    { // Código que se debe escribir
    }
    public double getPeso()
    { // Código que se debe escribir
    }
    ...
}

public class MaterialRodante
{
    private int mNúmeroID;
    private double mPeso;

```

(La opción D continúa en la página siguiente)

(Opción D: continuación)

```

public MaterialRodante(int ID, double peso)
{
    mNúmeroID = ID;
    mPeso = peso; // Peso en kilogramos
}
// Métodos accesores
public double getPeso() { return mPeso; }
public int getID() { return mNúmeroID; }
...
// Otros métodos
...
}

public class Máquina extends MaterialRodante
{
    private double mPesoRemolcado; // peso máximo que puede remolcar la
                                    // máquina

    public Máquina(int ID)
    {
        super(ID, 120000); // Las máquinas pesan 120000
                            // kilogramos
        mPesoRemolcado = 1400000; // Las máquinas pueden remolcar
                                    // 1400000 kilogramos
    }
    // Métodos accesores
    public double getPeso() { return super.getPeso(); }
    ...
    // Otros métodos
    ...
}

public class Vagón extends MaterialRodante
{
    private Paquete[] mPaquetes;
    private int mNúmeroPaquetes;
    public Vagón(int ID)
    {
        super(ID, 32000); // Un vagón vacío pesa 32000 kilogramos
        mPaquetes = new Paquete[100];
        mNúmeroPaquetes = 0;
    }
    // Métodos accesores
    public int getIDVagón() { return this.getID(); }
    public double getPeso()
    {
        // Código que se debe escribir
    }
    ...
    // Otros métodos
    ...
}

```

(La opción D continúa en la página siguiente)

(Opción D: continuación)

- 10.** (a) Defina la función de un *constructor*. [2]
- (b) Resuma las ventajas del polimorfismo, usando la clase `MaterialRodante` como ejemplo. [3]
- (c) Elabore un diagrama en lenguaje unificado de modelado (UML) de la clase `Tren`. [3]
- (d) Elabore un método `getNúmeroDeVagones()`, perteneciente a la clase `Tren`, que devuelva el número de vagones acoplados actualmente al tren. [2]
- (e) Elabore el método `quitarVagón()` que quite un vagón de un tren y devuelva el objeto que se ha quitado. Incluya una verificación de errores adecuada. [5]

(La opción D continúa en la página siguiente)

(Opción D: continuación)

- 11. (a) Resuma **una** ventaja de usar colecciones de biblioteca estándares. [2]
- (b) Describa **dos** formas en que podría variar la programación en función de que la lleve a cabo un equipo o un individuo que trabaje solo. [4]

El código siguiente implementa la clase `Paquete` que se usa en el programa de reparto de la empresa de transporte.

```
public class Paquete
{
    private int IDseguimiento;
    private double peso;
    public String direcciónDestino;
    public String direcciónOrigen;
    public Paquete(int ID)
    {
        IDseguimiento = ID;
        peso = 0;
    }
    public void setPeso(double nuevoPeso) { peso = nuevoPeso; }
    public double getPeso() { return peso; }
}
```

Las direcciones de origen y destino se almacenan en un objeto `Paquete` como cadenas simples. No obstante, las direcciones son complejas y hay distintos componentes de información que podrían estar presentes o no, como el nombre de pila o el nombre de la empresa, así como el número de casa, el nombre de la calle, la ciudad y el país.

Se ha decidido crear una nueva clase `Dirección` que contenga esta información.

- (c) Indique el tipo de datos adecuado que se va a usar en la clase `Dirección` para almacenar:
 - (i) el nombre de la calle; [1]
 - (ii) el número de edificio; [1]
 - (iii) una indicación de si es una dirección de empresa o no. [1]
- (d) Identifique qué cambios hay que hacer en la clase `Paquete` para que use la nueva clase `Dirección`. [3]

Se crearán las clases separadas `DirecciónOrigen` y `DirecciónDestino`. La dirección de destino puede contener instrucciones especiales para la persona que realiza la entrega. La dirección de origen contiene una variable que indica si el paquete se ha obtenido de la casa del cliente o de la oficina local de correos.

- (e) Resuma cómo se pueden crear estas **dos** nuevas clases duplicando el mínimo código posible. [3]

(La opción D continúa en la página siguiente)

(Opción D: continuación)

12. (a) Considere el fragmento de código siguiente.

```
Tren A = new Tren(123);
Máquina B = new Máquina(7);
A.agregarMáquina(B);
Vagón C = new Vagón(23);
A.agregarVagón(C);
Vagón D = new Vagón(66);
A.agregarVagón(D);
Vagón E = new Vagón(71);
A.agregarVagón(E);
A.agregarMáquina(new Máquina(9));
```

- (i) Dibuje la matriz `mMáquinas` después de ejecutar el fragmento de código. [2]
- (ii) Indique el valor de `mNúmeroMáquinas` después de ejecutar el fragmento de código. [1]
- (iii) Dibuje la matriz `mVagones` después de ejecutar los fragmentos de código anterior y siguiente. [2]

```
Vagón F = A.quitarVagón();
F = A.quitarVagón();
A.agregarVagón(new Vagón(214));
```

Los paquetes cargados en un vagón no pueden pesar más que la capacidad del vagón. Las máquinas de un tren deben tener una potencia combinada suficiente para tirar de los vagones que remolca. La compañía debe ser capaz de comprobar que se cumplan esos requisitos.

- (b) Elabore el método `getPeso()` de la clase `Vagón` que devuelva el peso total **combinado** de los paquetes que tiene el vagón en el momento del viaje y del vagón mismo. [4]
- (c) Elabore el método `getPeso()` de la clase `Tren` que devuelva el peso total **combinado** de todos los paquetes, máquinas y vagones de un tren. [4]
- (d) Explique por qué tener un método `getPeso()` en las clases `Tren` y `Vagón` no genera un error de compilación, incluso aunque la clase `Tren` no herede de la clase `MaterialRodante`. [2]

Fin de la opción D