

Informática
Nivel medio
Prueba 2

Miércoles 18 de noviembre de 2015 (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 – 13

Opción A — Bases de datos

1. Paul, el dueño de una pequeña tienda, decide crear e implementar una base de datos de clientes después de asistir a un curso sobre bases de datos.

(a) (i) Identifique **un** beneficio de crear una base de datos. [1]

(ii) Identifique **un** costo de la implementación. [1]

Considere la siguiente sección del archivo de base de datos de clientes.

TablaClientes

Nombre	Fecha	ArticuloPedido	CantidadPedida	TotalFactura
Ann Low	18/04/2015	1713	4	200,00
Boris Nicke	18/04/2015	1324	3	180,00
Greta Pink	18/04/2015	1713	3	150,00
Rob Nool	19/04/2015	1648	7	360,00
Ann Low	19/04/2015	1713	5	250,00
Ivor Turk	20/04/2015	1423	6	105,00

(b) (i) Indique el número de campos de cada registro de la tabla `TablaClientes`. [1]

(ii) Describa, usando un ejemplo de la tabla anterior, por qué el nombre del cliente **no** es una clave principal adecuada. [2]

(iii) Defina el término *clave secundaria*. [1]

(c) Describa los pasos que hay que llevar a cabo en una consulta para obtener una lista de todos los artículos de los que se pidieron más de cinco el 19/04/2015. [4]

Paul descubre que el tiempo de respuesta de las consultas de la base de datos es muy lento.

(d) Explique qué podría realizar Paul para mejorar el tiempo de respuesta sin realizar cambios de hardware. [5]

(La opción A continúa en la página siguiente)

(Opción A: continuación)

2. Una compañía se dedica a alquilar bicicletas a turistas. Los datos de las bicicletas y los clientes se almacenan en un archivo de base de datos. En la siguiente tabla se muestran los datos para un día.

TablaAlquilerBicicletas

ClienteId	CNombre	CTelefono	Salida	Entrada	IdBicicleta	MarcaBicicleta	ModeloBicicleta
BL567	Boris Lok	99123456	09.00	11.00	DU12	BMS_11	A
CL167	Ivy Lok	93123455	09.00	11.00	DU14	AVG_00	B
AL751	Ann Summer	43453657	09.00	17.00	DU54	AVG_00	A
FC345	Fred Cohen	38321432	10.00	15.00	DU23	XYZ_94	C
...

La estructura de la tabla se puede resumir usando la **notación abreviada** siguiente.

TablaAlquilerBicicletas

(ClienteId, CNombre, CTelefono, Salida, Entrada, IdBicicleta,
MarcaBicicleta, ModeloBicicleta)

- (a) Identifique **tres** funciones de un sistema de gestión de bases de datos en una aplicación como la de alquiler de bicicletas. [3]

Hay varios aspectos de la TablaAlquilerBicicletas que no son satisfactorios y se ha decidido, por tanto, normalizar esa base de datos.

- (b) Resuma el objetivo de la normalización. [2]
- (c) Indique las características de
- (i) primera forma normal (1FN); [1]
 - (ii) segunda forma normal (2FN); [1]
 - (iii) tercera forma normal (3FN). [1]
- (d) Elabore la base de datos en 3FN. Usando la **notación abreviada**, muestre con claridad la estructura de la base de datos en **1FN, 2FN y 3FN**. [7]

(La opción A continúa en la página siguiente)

Véase al dorso

(Opción A: continuación)

3. (a) Distinga entre el diseño lógico y físico de una base de datos. [4]
- (b) Identifique **tres** elementos de información que estén contenidos en un diccionario de datos. [3]
- (c) Explique **dos** responsabilidades de un administrador de base de datos. [4]

Un servicio de información en línea cuenta con una base de ofertas de trabajos y detalles de los usuarios. Los usuarios pueden ver las vacantes disponibles y responder electrónicamente a los trabajos en los que están interesados.

- (d) Discuta **una** ventaja y **una** desventaja de usar una base de datos para este servicio de información en línea. [4]

Fin de la opción A

Opción B — Modelos y simulaciones

4. La contaminación atmosférica es una preocupación para la sanidad pública. Una medida de la contaminación del aire es la cantidad de PM10 (partículas que miden menos de 10 micrómetros) en el aire. Existe ya legislación europea que requiere que cualquier ciudad con más de 100.000 habitantes debe mantenerse dentro de los siguientes límites de PM10:

1. El valor medio PM10 en un día no debe exceder 50 microgramos por metro cúbico ($50 \mu\text{g m}^{-3}$) más de 35 veces en el año.
2. Los valores diarios promedios del año completo (medidos desde el 1 de enero) no deben exceder los 40 microgramos por metro cúbico ($40 \mu\text{g m}^{-3}$).

En una ciudad industrial, las medidas de PM10 se toman una vez cada hora y posteriormente se registran y almacenan. Durante el año, se elabora un modelo completo con las medidas tomadas cada hora. Al final del año la ciudad genera un informe que incluye:

- un diagrama de la media de los valores diarios
- el número de días en que la media ha excedido los $50 \mu\text{g m}^{-3}$
- los valores diarios promediados durante el año.

(a) Resuma, usando un dibujo u otro recurso, un método para organizar los datos almacenados en una hoja de cálculo para facilitar la referencia y el análisis. [3]

Todos los días se publica en la prensa local el valor medio del día anterior, junto con el número de días, si ha habido alguno, en que se ha sobrepasado el límite ($50 \mu\text{g m}^{-3}$) desde el 1 de enero.

(b) Describa cómo se podrían usar los resultados de cada hora para mostrar los resultados diarios necesarios así como los necesarios para generar el informe a final de año. [5]

El modelo se puede usar para generar datos sobre patrones de contaminación a lo largo del tiempo.

(c) (i) Identifique **dos** de esos patrones. [2]

(ii) Explique cómo uno de los patrones de la parte (c)(i) podría ayudar a la ciudad a planificar el futuro. [2]

En cierto momento del año, los valores diarios de la ciudad, promediados desde el 1 de enero, son de $45 \mu\text{g m}^{-3}$. Se introduce el fin de semana “sin automóvil”, en el que la gente no puede conducir sus vehículos por la ciudad. Esto reduce satisfactoriamente el promedio de PM10 pero tiene un impacto negativo en el comercio local.

(d) Explique cómo se podría usar el modelo para estimar el número mínimo de fines de semana “sin automóvil” para mantener los valores diarios promediados en el año por debajo de $40 \mu\text{g m}^{-3}$. [5]

(La opción B continúa en la página siguiente)

Véase al dorso

(Opción B: continuación)

5. Tenemos un juego de mesa para niños muy pequeños que usa un tablero de 4×4 casillas. El juego requiere dos jugadores.

El objetivo es ser el primer jugador en alcanzar la casilla INICIO, es decir, la casilla inferior derecha. Los jugadores juegan por turnos para moverse por las casillas.

Los jugadores pueden mover a una casilla adyacente que, desde su posición, puede ser:

- hacia abajo verticalmente
- horizontalmente a la derecha
- diagonalmente en la dirección de INICIO.

Un jugador puede saltar por encima del otro si uno de los posibles movimientos está bloqueado por otro jugador.

En el diagrama siguiente, el jugador A se encuentra en la casilla 1 y el jugador B en la casilla 6.

A	2	3	4
5	B	7	8
9	10	11	12
13	14	15	INICIO

- (a) Enumere las casillas a las que puede moverse el jugador A.

[1]

(La opción B continúa en la página siguiente)

(Continuación: opción B, pregunta 5)

- (b) El jugador A se mueve a la casilla 2 y el B a la 10.

1	A	3	4
5	6	7	8
9	B	11	12
13	14	15	INICIO

Para cada uno de los movimientos que **pueda realizar** el jugador A, sugiera y justifique el mejor movimiento que **podría realizar** el jugador B.

[3]

Se realiza una simulación del juego en un programa informático.

- (c) (i) Resuma **una** forma de representar a los jugadores A y B y sus posiciones en memoria. [2]
- (ii) Usando la estructura de la parte (c)(i), indique las posiciones actuales de los jugadores A y B, como se muestra en el diagrama de la parte (b). [1]
- (iii) Usando la respuesta a la parte (c)(i), resuma cómo se podría identificar cada **posible** movimiento desde la posición actual del jugador. [2]
- (iv) Sugiera cómo se puede elegir el **mejor movimiento** de entre todos los posibles. [3]

El computador siempre intentará encontrar la mejor ruta a la casilla INICIO.

- (d) Resuma qué efecto tendría sobre la ejecución del programa en un computador usar un tablero mucho mayor. [4]

(La opción B continúa en la página siguiente)

(Opción B: continuación)

6. Un vehículo sin conductor depende de una serie de láseres y cámaras que se combinan con mapas en alta definición para ofrecer una vista tridimensional (3D) del entorno. Se programan reglas en el sistema de forma que los vehículos circulen a una velocidad óptima, mantengan una distancia de seguridad con el vehículo que tiene delante, utilicen el combustible de forma económica y respeten las normas de tráfico.
- (a) Resuma la necesidad de almacenar mapas de alta definición en memoria. [2]
- (b) Resuma cómo el análisis de imágenes 3D sucesivas en memoria puede contribuir a que el automóvil conduzca con mayor seguridad. [3]
- (c) Resuma el tiempo y la memoria necesarios para visualizar en 3D esta situación. [3]
- (d) Discuta las consecuencias sociales de los vehículos autoconducidos. [4]

Fin de la opción B

Opción C — Ciencia de la Web

7. El código que se muestra a continuación es parte de un documento XML (*Extensible Mark-up Language*, lenguaje de marcas extensible) que contiene los detalles de una colección de DVD.

```
<colección>
  <dvd>
    <título>El Hobbit</título>
    <género>Fantasia</género>
  <dvd>
    <título>Algo para recordar</título>
    <género>Romantica</género>
  </dvd>
<!--introducir mas DVD aqui-->
</colección>
```

- (a) Identifique el error en la sección del código XML anterior. [1]

En algunas aplicaciones, se usa XML en lugar de HTML, fundamentalmente por su extensibilidad.

- (b) Resuma el significado de esta propiedad en el contexto del código XML. [2]
- (c) Describa **un** beneficio de almacenar la información de formato para HTML en un archivo CSS. [2]

Un documento XML contiene detalles de la colección de DVD de una persona. El código XML se enviará al servidor web usando el conjunto de protocolos de Internet, que incluye los protocolos TCP e IP.

- (d) Describa cómo funcionan estos dos protocolos cuando envían conjuntamente datos por Internet. [4]

El siguiente formulario web forma parte de una página de acceso a una tienda en línea.

Nombre	<input type="text"/>
Correo electrónico	<input type="text"/>
País	<input type="text"/>

- (e) (i) Explique cómo se podrían usar scripts de cliente en esta página de acceso antes de que la página se envíe al servidor web de la tienda en línea. Debe hacer referencia a cualquier software que se haya usado. [3]
- (ii) Resuma los beneficios de usar scripts de cliente para el sitio de compras. [2]

Una red social almacena varias fotografías de perfil de cada cliente en su servidor web.

- (f) Explique cómo el uso de scripts puede permitir al usuario cambiar su fotografía de perfil sin recargar toda la página. [4]

(La opción C continúa en la página siguiente)

Véase al dorso

(Opción C: continuación)

8. Cuando un usuario solicita un archivo de un sitio web concreto, ese sitio web usa compresión con pérdidas para enviar el archivo a través de Internet.
- (a) Discuta cómo podría afectar el uso de compresión con pérdidas a la experiencia de usuario. [5]
- Los tres procesos esenciales de un motor de búsqueda web son:
- inspeccionar mediante crawlers
 - indexar
 - buscar.
- (b) Resuma las funciones de cada uno de estos tres procesos. [6]
- (c) Explique cómo el algoritmo PageRank podría discriminar a los nuevos sitios. [3]
- (d) Explique cómo puede un motor de búsqueda mantener un índice actualizado cuando la Web se expande continuamente. [3]
9. Una gran empresa minorista en línea se traslada a un nuevo sitio especialmente diseñado y está considerando incorporar la informática en la nube a su estrategia de TI.
- (a) Describa **dos** características arquitectónicas que sean componentes esenciales de un modelo informático en la nube público y sostenible. [4]
- A la hora de tomar decisiones relacionadas con sus instalaciones informáticas, la compañía debe tener en cuenta los factores siguientes:
- almacenamiento de datos económicos sensibles
 - uso intenso de correo electrónico con fines publicitarios
 - aumento de la demanda durante la época navideña
 - cualquier proyecto futuro de desarrollo.
- Hoy en día, muchas compañías usan un enfoque híbrido para la informática en la nube, empleando para ello nubes privadas y públicas.
- (b) Explique de qué modo esta compañía podría usar un enfoque híbrido. [6]

Fin de la opción C

Opción D — Programación orientada a objetos

Una pequeña clínica que cuenta con tres doctores presta servicios en un pueblo. En la base de datos de la clínica se almacena la información de todos los clientes. Los pacientes que acuden a la clínica reciben un número de prioridad (de 1 a 3) y se sientan en una sala de espera hasta que un doctor se quede libre. Cuando llega su turno, los pacientes abandonan la sala de espera para entrar en la consulta del doctor asignado, que realiza un diagnóstico, ofrece tratamiento y receta los medicamentos.

El sistema informático de la clínica está programado en Java. El sistema está formado por muchos objetos y algunos de ellos se enumeran a continuación.

Objeto	Descripción
Doctor	Profesional habilitado que trata a pacientes en la clínica.
Paciente	Enfermo que solicita una consulta con un doctor.
SalaDeEspera	Lugar en que el paciente espera la consulta.
Consulta	Cita entre un médico y un paciente que genera diagnóstico, tratamiento y receta la medicación.
Tratamiento	Registro fechado de todas las acciones y medicaciones recetadas para tratar la condición diagnosticada al paciente.

Los objetos `Paciente`, `SalaDeEspera` y `Tratamiento` se han definido en los diagramas UML siguientes:

Paciente
Integer id String nombre Integer prioridad String doctor
setId (Integer id) setNombre (String nombre) setPrioridad (Integer prioridad) setDoctor (String doctor) Integer getId() String getNombre() Integer getPrioridad() String getDoctor() String toString()

SalaDeEspera
Paciente[10]pacientes
agregar(Paciente nuevoPaciente) void llamarSiguientePaciente() Integer buscarIndiceSiguientePaciente() quitar(Integer n)

Tratamiento
String fecha Integer idPaciente String doctor String acciones String medicacion
setFecha (String fecha) setIdPaciente (Integer id) setDoctor (String doctor) setAcciones (String acciones) setMedicacion (String medicacion) String getFecha() Integer getIdPaciente() String getDoctor() String getAcciones() String getMedicacion() String toString()

(La opción D continúa en la página siguiente)

(Opción D: continuación)

Los objetos `Paciente` y `SalaDeEspera` se implementan de la siguiente manera:

```
public class Paciente
{
    private int id;
    private String nombre;
    private int prioridad;
    private String doctor;

    public Paciente(int i, String n, int p)
    {
        id = i;
        nombre = n;
        prioridad = p;
        doctor = null;
    }
    public void setId(int i) { id = i; }
    public void setNombre(String n) { nombre = n; }
    public void setPrioridad(int p) { prioridad = p; }
    public void setDoctor(String d) { doctor = d; }
    public int getId() { return id; }
    public String getNombre() { return nombre; }
    public int getPrioridad() { return prioridad; }
    public String getDoctor() { return doctor; }
    public String toString() { return id+" "+nombre+" "+prioridad+" "+doctor; }
}

public class SalaDeEspera
{
    private Paciente[] pacientes = new Pacientes[10];

    // usa el constructor por defecto

    public void agregar(Paciente nuevoPaciente)
    // agrega el nuevo paciente a la siguiente posicion vacia de la matriz
    {
        int i = 0;
        while ((pacientes[i] != null) && (i < 10))
        {
            i=i+1;
        }
        if (i==10) { System.out.println("No hay mas sitio en la sala de espera."); }
        else { pacientes[i] = nuevoPaciente; }
    }
}
```

(La opción D continúa en la página siguiente)

(Opción D: continuación)

```
public void llamarSiguientePaciente()
// busca el siguiente paciente, muestra sus datos
// y lo extrae de la matriz
{
    int indice = 0;
    if (pacientes[0]==null)
    {
        System.out.println("La sala de espera esta vacia.");
    }
    else
    {
        indice = buscarIndiceSiguientePaciente();
        quitar(indice);
    }
}

private int buscarIndiceSiguientePaciente()
// devuelve el indice del primer paciente con la
// mayor prioridad en la matriz de pacientes
{
    int max = 0;
    //... falta el codigo ...
    return max;
}

private void quitar(int n)
// imprime los datos de la instancia del paciente en el índice n de la matriz
// y quita al paciente desplazando al resto de pacientes en un indice en
// direccion al inicio de la matriz
{
    //... falta el codigo ...
}
}
```

(La opción D continúa en la página siguiente)

(Opción D: continuación)

10. (a) Defina el término *constructor*, usando un ejemplo del código de las páginas 12 y 13. [2]
- (b) Describa **un** campo adicional que se podría haber incluido en la clase `Paciente`. Incluya un tipo de datos y datos ejemplo en su respuesta. [2]
- (c) Describa la relación entre el objeto `Paciente` y el objeto `SalaDeEspera`. [2]

Considere la clase `SalaDeEspera` que se presenta en las páginas 12 y 13.

- (d) Elabore las líneas de código que faltan en el método `buscarIndiceSiguientePaciente()` para que devuelva el índice del primer paciente con la prioridad más alta en la matriz `pacientes`.
Nota: la máxima prioridad posible es 3. [3]
- (e) Elabore el método `quitar(int n)` que muestre los datos del objeto paciente con índice `n` y a continuación quite ese objeto moviendo todos los objetos paciente restantes en un índice hacia el inicio de la matriz `pacientes`. Puede asumir que `n` es un índice válido entre 0 y 9 y que existe una instancia de `Paciente` en ese índice. [6]
11. (a) En relación con la clase `Paciente`, resuma **una** ventaja de la encapsulación. [2]
- (b) En relación con el objeto `Tratamiento`, discuta **una** consideración ética durante el diseño del software que almacene pacientes y sus enfermedades. [4]

La clínica desea empezar a guardar datos en un objeto `Doctor`, incluyendo el nombre completo, número de teléfono y si el médico está presente o no. Por ejemplo:

nombre:	Dra. Henriëtte Mănescu-Rața
teléfono:	0734511122
presente:	sí

- (c) Diseñe el objeto `Doctor` usando un diagrama UML. [3]
- (d) En relación con el objeto `Doctor`, resuma la necesidad de usar los conjuntos de caracteres extendidos usados por los lenguajes de programación actuales. [3]

(La opción D continúa en la página siguiente)

(Opción D: continuación)

12. Los objetos `Tratamiento` se instancian durante todo el día y se añaden a una colección. El objeto `archivoTratamiento` contiene los métodos siguientes que actúan sobre esta colección:

- `getNext()` lee el siguiente tratamiento de la colección y lo devuelve
- `hasNext()` devuelve falso cuando no hay más tratamientos en la colección.

Construya el método `mostrarMedicacionPorDoctor()`, que reciba el nombre del doctor como parámetro y muestre la medicación para cada tratamiento en la colección proporcionada por ese doctor. Puede asumir que se ha declarado la variable global `archivoTratamiento`, que está abierta para lectura y que la primera vez que se llame a `getNext()` devolverá el primer tratamiento de la colección.

[6]

13. Es necesario desarrollar posteriormente el objeto `Tratamiento`. Hay tres tipos posibles de tratamiento que se van a grabar.

- `ambulatorio` – el paciente recibe el tratamiento y posteriormente se va a su domicilio
- `hospitalario` – el paciente pasa una o más noches en la clínica
- `derivacion` – el paciente es trasladado al hospital de una ciudad cercana.

Todos los tratamientos tienen campos comunes, como el identificador (ID) del paciente, la fecha y un objeto doctor, pero otros campos son diferentes. Por ejemplo, los tratamientos `ambulatorio` y `hospitalario` incluyen medicación, mientras que la `derivacion` no la incluye. Por otra parte, la `derivacion` incluye el nombre del hospital al que se ha enviado al paciente y si se ha usado una ambulancia para el traslado. El tratamiento `hospitalario` incluye un número de habitación.

(a) Elabore diagramas que muestren cómo se puede usar la herencia para rediseñar la clase `Tratamiento`.

[6]

(b) Describa **tres** ventajas de la modularidad en el desarrollo de programas.

[6]

Fin de la opción D
