PRIMER EJERCICIO PARTE PRÁCTICA DEL PROCESO SELECTIVO DE

A2 GESTIÓN, DESARROLLO, DIGITALIZACIÓN Y SOPORTE DE SISTEMAS DE INFORMACIÓN (ORDEN 6)

DE LA UNIVERSIDAD COMPLUTENSE DE MADRID

Concurso-Oposición Libre
Resolución de fecha 20-11-2024

2 de febrero de 2026

SUPUESTO 1

Para desarrollo de un sistema de gestión de vehículos, se solicita diseñar una jerarquía de clases para un concesionario. Existe una clase base abstracta llamada Vehiculo con atributos marca y precioBase. De esta clase heredan Coche y Moto.
- La clase Coche tiene un atributo propio numPuertas.
- La clase Moto tiene un atributo booleano tieneSidecar.
- Se debe implementar un método calcularPrecioFinal() que en Coche sume 500€ si tiene más de 3 puertas, y en Moto sume 300€ si tiene sidecar.
- Además, se requiere una interfaz Impuesto con un método obtenerImpuestoCirculacion().

PREGUNTAS SUPUESTO 1

1. Si la clase Vehiculo se define como abstract, ¿cuál de las siguientes afirmaciones es correcta?
a) No se puede instanciar la clase Vehiculo directamente.
b) La clase Vehiculo no puede tener atributos, solo métodos.
c) Se pueden crear instancias de Vehiculo usando el operador new.
d) Todos los métodos de Vehiculo deben ser obligatoriamente abstractos.

2. En la clase Moto, el atributo tieneSidecar es de tipo boolean. ¿Cuál es su valor por defecto si no se inicializa?
a) 0
b) true
c) false
d) null

3. Si queremos que el método calcularPrecioFinal() sea obligatorio de implementar en las subclases, ¿cómo debemos declararlo en Vehiculo?
a) public static double calcularPrecioFinal();
b) public void calcularPrecioFinal() {}
c) public final double calcularPrecioFinal();
d) public abstract double calcularPrecioFinal();

4. ¿Cuál es la forma correcta de implementar la interfaz Impuesto en la clase Coche?
a) public class Coche extends Vehiculo override Impuesto
b) public class Coche extends Vehiculo implements Impuesto
c) public class Coche extends Vehiculo, Impuesto
d) public class Coche implements Vehiculo, Impuesto

5. ¿Qué anotación se recomienda usar al implementar calcularPrecioFinal() en Coche para asegurar que estamos sustituyendo el método de la clase padre?
a) @Implement
b) @OverrideMethod
c) @Override
d) @Overload

PREGUNTA DE RESERVA 1

6. Si en el constructor de Coche queremos llamar al constructor de Vehiculo para asignar la marca, ¿qué palabra clave usamos?
a) Vehiculo(marca);
b) parent(marca);
c) this(marca);
d) super(marca);

SUPUESTO 2

Según el siguiente código, responda a las preguntas

```
<?php
declare(strict_types=1);
final class UserService
{
   public function __construct(private PDO $db) {
      $this->db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
   }
   public function register(array $in): array
   {
      $email  = strtolower(trim((string)($in['email'] ?? '')));
      $name   = trim((string)($in['name'] ?? ''));
                  $active = array_key_exists('active',$in) && !is_null($in['active'])
               ? (int)$in['active'] : 1;
      if ($email === '' || !filter_var($email, FILTER_VALIDATE_EMAIL)) {
         throw new InvalidArgumentException();
      }
      $this->db->beginTransaction();
      try {
         $st = $this->db->prepare(
            "SELECT COUNT(*) FROM users u
             WHERE u.deleted_at IS NULL
               AND LOWER(u.email) = :email
               AND u.name = :name
               AND u.active = :active"
         );
         $st->execute([':email'=>$email,':name'=>$name,':active'=>$active]);
         if ((int)$st->fetchColumn() > 0) {
            $this->db->rollBack();
            return ['created'=>false,'user_id'=>null,'duplicate'=>true];
         }
         $ins = $this->db->prepare(
            "INSERT INTO users (email,name,active,created_at)
             VALUES (:email,:name,:active,NOW())"
         );
         $ins->execute([':email'=>$email,':name'=>$name,':active'=>$active]);
         $id = (int)$this->db->lastInsertId();
         $this->db->commit();
         return ['created'=>true,'user_id'=>$id,'duplicate'=>false];
      } catch (Throwable $e) {
         if ($this->db->inTransaction()) $this->db->rollBack();
         throw $e;
      }
   }
}
```

PREGUNTAS SUPUESTO 2

7) El método register(array $in) recibe datos externos. ¿Qué riesgo real existiría si no se castean explícitamente valores como active antes de usarlos en métodos tipados?
a) Ninguno, PHP siempre convierte automáticamente
b) TypeError en tiempo de ejecución
c) Error solo en PHP 7
d) El valor se convierte en null

8) ¿Por qué esta forma $email = trim((string)($in['email'] ?? '')); es más robusta que usar directamente $in['email']?
a) Porque trim() solo acepta strings
b) Porque evita warnings/TypeError si el índice no existe o no es string
c) Porque convierte el email a UTF-8
d) Porque garantiza que el email sea válido

9) Dado este bloque:
```
if ($email === '' || !filter_var($email, FILTER_VALIDATE_EMAIL)) {
    throw new InvalidArgumentException();
}
```
¿Qué aspecto de negocio NO queda cubierto por este bloque de validación?
a) Que el email no esté vacío
b) Que tenga formato válido
c) Que sea único en la base de datos
d) Que sea un string

10) Si se sustituyera array_key_exists('active',$in) por isset($in['active']), ¿qué caso cambiaría de comportamiento?
a) ['active' => 1]
b) ['active' => 0]
c) ['active' => null]
d) Ninguno

11) ¿Qué garantiza realmente este bloque?
```
$this->db->beginTransaction();
```
a) Que no haya duplicados
b) Que todas las operaciones se confirmen o se deshagan juntas
c) Que otros procesos no puedan leer la tabla
d) Que el SELECT bloquee la tabla

PREGUNTA DE RESERVA 2

12) ¿Qué problema de concurrencia puede existir aunque haya transacción, si no existe una restricción UNIQUE en la base de datos?
a) Ninguno, la transacción lo evita completamente
b) Dos procesos concurrentes pueden pasar la comprobación previa y ambos insertar el mismo registro
c) El SELECT COUNT(*) bloquea automáticamente la tabla
d) lastInsertId() puede devolver un valor incorrecto por concurrencia

SUPUESTO 3

Según el siguiente enunciado, responda a las preguntas.
Una empresa quiere gestionar la información de sus clientes, productos y pedidos. Para ello dispone de la siguiente base de datos relacional:

Estructura de tablas

CLIENTES

Campo	Tipo
id_cliente	INT (PK)
nombre	VARCHAR(100)
ciudad	VARCHAR(50)
fecha_alta	DATE

PRODUCTOS

Campo	Tipo
id_producto	INT (PK)
nombre	VARCHAR(100)
precio	DECIMAL(8,2)
stock	INT

PEDIDOS

Campo	Tipo
id_pedido	INT (PK)
id_cliente	INT (FK → CLIENTES.id_cliente)
fecha_pedido	DATE

DETALLE_PEDIDO

Campo	Tipo
id_pedido	INT (FK → PEDIDOS.id_pedido)
id_producto	INT (FK → PRODUCTOS.id_producto)
cantidad	INT

PREGUNTAS SUPUESTO 3

13. ¿Qué consulta obtiene el nombre de los clientes que han realizado al menos un pedido?

a) SELECT nombre FROM clientes;

b) SELECT DISTINCT c.nombre
FROM clientes c
JOIN pedidos p ON c.id_cliente = p.id_cliente;

c) SELECT nombre
FROM clientes
WHERE id_cliente = id_cliente;

d) SELECT nombre
FROM clientes
WHERE id_cliente NOT IN (SELECT id_cliente FROM pedidos);

14. ¿Qué consulta devuelve el importe total de cada pedido (suma de cantidad × precio)?
a) SELECT id_pedido, SUM(cantidad * precio)
FROM productos
GROUP BY id_pedido;
b) SELECT p.id_pedido, SUM(d.cantidad * pr.precio) AS total
FROM pedidos p
JOIN detalle_pedido d ON p.id_pedido = d.id_pedido
JOIN productos pr ON d.id_producto = pr.id_producto
GROUP BY p.id_pedido;
c) SELECT id_pedido, precio * cantidad
FROM detalle_pedido;
d) SELECT SUM(precio)
FROM productos;

15. ¿Qué consulta muestra los clientes que no han realizado ningún pedido?
a) SELECT nombre
FROM clientes
WHERE id_cliente IN (SELECT id_cliente FROM pedidos);
b) SELECT c.nombre
FROM clientes c
LEFT JOIN pedidos p ON c.id_cliente = p.id_cliente
WHERE p.id_pedido IS NULL;
c) SELECT nombre FROM clientes;
d) SELECT nombre
FROM clientes
JOIN pedidos USING (id_cliente);

16. ¿Qué consulta obtiene el cliente que ha gastado más dinero en total?
a) SELECT c.nombre
FROM clientes c;
b) SELECT c.nombre, SUM(d.cantidad * pr.precio) AS total
FROM clientes c
JOIN pedidos p ON c.id_cliente = p.id_cliente
JOIN detalle_pedido d ON p.id_pedido = d.id_pedido
JOIN productos pr ON d.id_producto = pr.id_producto
GROUP BY c.id_cliente
ORDER BY total DESC
LIMIT 1;
c) SELECT c.nombre, SUM(d.cantidad * pr.precio) AS total
FROM clientes c
JOIN pedidos p ON c.id_cliente = p.id_cliente
JOIN detalle_pedido d ON p.id_pedido = d.id_pedido
JOIN productos pr ON d.id_producto = pr.id_producto
ORDER BY total DESC
FETCH FIRST 1 ROW ONLY;
d) SELECT nombre
FROM clientes
WHERE id_cliente = MAX(id_cliente);

17. ¿Qué consulta devuelve correctamente el nombre de los clientes que han comprado al menos una unidad de cada producto disponible?

a)
```
SELECT c.nombre
FROM clientes c
WHERE NOT EXISTS (
   SELECT 1
   FROM productos p
   WHERE NOT EXISTS (
      SELECT 1
      FROM pedidos pe
      JOIN detalle_pedido dp ON pe.id_pedido = dp.id_pedido
      WHERE pe.id_cliente = c.id_cliente
        AND dp.id_producto = p.id_producto
   )
);
```

b)
```
SELECT c.nombre
FROM clientes c
WHERE c.id_cliente IN (
   SELECT pe.id_cliente
   FROM pedidos pe
   JOIN detalle_pedido dp ON pe.id_pedido = dp.id_pedido
   GROUP BY pe.id_cliente
   HAVING COUNT(dp.id_producto) = (
      SELECT COUNT(*) FROM productos
   )
);
```

c)
```
SELECT c.nombre
FROM clientes c
JOIN pedidos pe ON c.id_cliente = pe.id_cliente
JOIN detalle_pedido dp ON pe.id_pedido = dp.id_pedido
GROUP BY c.id_cliente, c.nombre
HAVING COUNT(dp.id_producto) = (
   SELECT COUNT(*) FROM productos
);
```

d)
```
SELECT c.nombre
FROM clientes c
WHERE EXISTS (
   SELECT 1
   FROM productos p
   JOIN detalle_pedido dp ON p.id_producto = dp.id_producto
   JOIN pedidos pe ON dp.id_pedido = pe.id_pedido
   WHERE pe.id_cliente = c.id_cliente
   GROUP BY p.id_producto
   HAVING COUNT(*) >= 1
);
```

PREGUNTA DE RESERVA 3

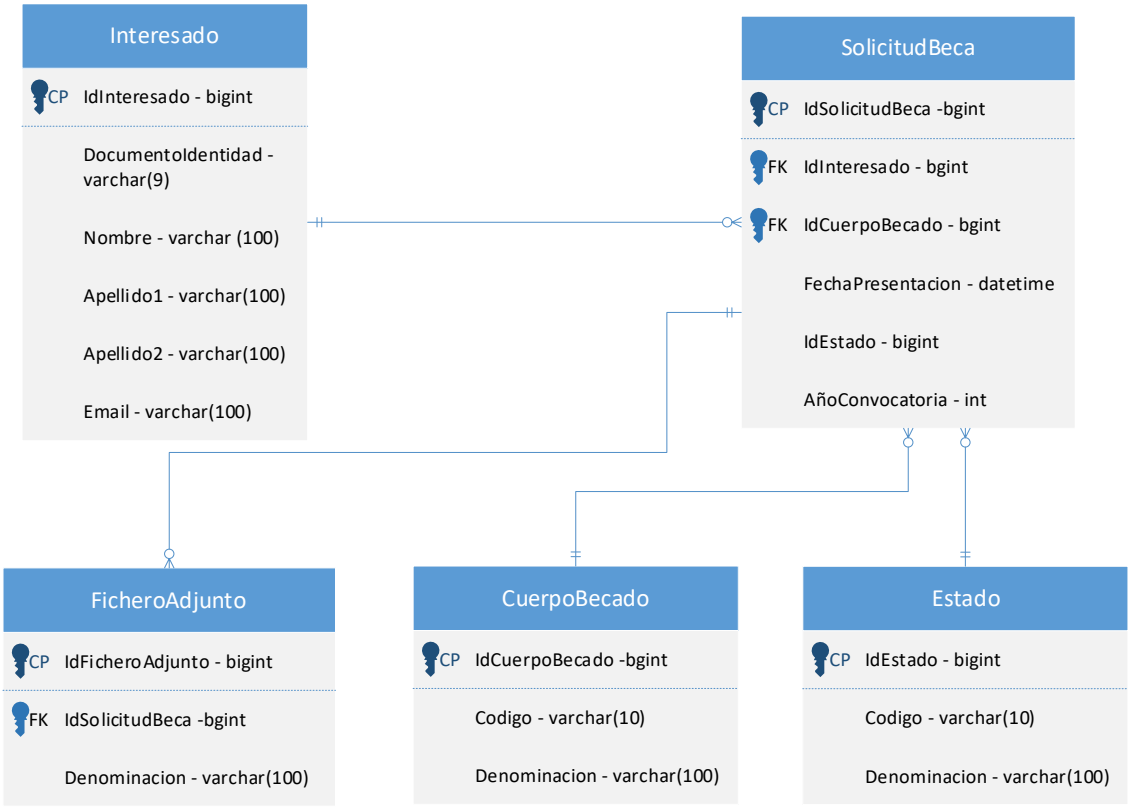
18. ¿Qué tipo de JOIN devuelve todos los clientes, tengan o no pedidos?
a) INNER JOIN
b) RIGHT JOIN
c) LEFT JOIN
d) FULL JOIN

SUPUESTO 4

En una universidad quieren desarrollar un sistema integrado de gestión de becas para alumnos/profesores/personal, donde se pueda:

- Permitir que el personal universitario solicite becas
- Validar documentación PDF (boletines, notas, justificantes)
- Publicar listados provisionales/definitivos
- Integrar autenticación corporativa de la universidad (@uni.es)

A continuación, se expone parte del modelo de datos del sistema, con su correspondientes claves primarias y foráneas:



El sistema, en su parte frontend, permitirá la firma y presentación de solicitudes, subsanaciones y alegaciones por parte de los interesados una vez se autentiquen. También permitirá la descarga de los distintos listados que se publiquen por parte de los empleados públicos de la universidad que tramitan estas becas.
Por otra parte, el backend dará el servicio necesario al frontend y, además, permitirá a los empleados públicos de la universidad, desde una aplicación de gestión, visualizar la información y documentación de las solicitudes, subsanaciones y alegaciones presentadas, así como la publicación de los listados mencionados anteriormente.
Para el desarrollo del frontend se utilizará un framework de desarrollo web con JavaScript, HTML5 y CSS3, mientras que el backend podrá desarrollarse con .NET o Java, dependiendo de la experiencia y conocimientos de los desarrolladores de la universidad.
También se ha decidido usar GitHub (plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git) para poder compartir el trabajo entre los desarrolladores en la cual se puede almacenar, compartir y trabajar de forma conjunta en el código. De esta forma se permite seguir y administrar los cambios en el código a lo largo del tiempo.

PREGUNTAS SUPUESTO 4

19. ¿Qué elemento HTML utilizaría para impedir que el e-mail exceda los 100 caracteres cuando el interesado lo cumplimente?
    a) <input type="email" id="email" name="email" size="100">
    b) <input type="email" id="email" name="email" max="100">
    c) <input type="email" id="email" name="email" maxlength="100">
    d) <input type="email" id="email" name="email" length="100">

20. Se decide usar ORM (Object Relational Mapping), según la solución backend escogida (Entity Framework si se opta por .NET o bien Hibernate si se opta por Java) ¿Cuál es la función principal de un ORM?
    a) Genera automáticamente código HTML a partir de las tablas de base de datos
    b) Permite trabajar con clases y objetos Java/C# que se mapean automáticamente a tablas y filas de la base de datos relacional
    c) Compila el código fuente Java/C# a sentencias SQL ejecutables
    d) Valida la sintaxis SQL antes de enviarla a la base de datos

21. ¿Qué tipo de correspondencia hay entre las tablas “SolicitudBeca” y “FicheroAdjunto”?
    a) Un registro de SolicitudBeca puede tener cero o más registros de FicheroAdjunto
    b) Un registro de SolicitudBeca puede tener como máximo un registro de FicheroAdjunto
    c) N registros de SolicitudBeca pueden tener M registros de FicheroAdjunto
    d) Un registro de SolicitudBeca puede tener cero o como máximo un registro de FicheroAdjunto

22. El formulario de solicitud será extenso y tendrá muchas consultas de datos ¿qué usarías para validar email de base de datos sin recargar página entera?
    a) Server Sent Events (updates servidor)
    b) Ajax (Asynchronous JavaScript And XML)
    c) WebSocket (conexión persistente)
    d) Server Side Rendering (página pre-renderizada)

23. El proceso de presentación de solicitud implica la autenticación del usuario, la firma de la solicitud, el guardado en base de datos, la generación de un asiento registral y la generación de un justificante de presentación. ¿Qué diagrama UML utilizaría para representar las acciones implicadas en este flujo?
    a) Diagrama de flujo de datos
    b) Diagrama de despliegue
    c) Diagrama de secuencia
    d) Diagrama de Entidad/Relación

PREGUNTA DE RESERVA 4

24. ¿Qué herramienta analiza código detectando bugs, vulnerabilidades y code smells automáticamente?
    a) Mercurial
    b) Git
    c) SonarQube
    d) Linter JS

SUPUESTO 5

Esquema (común a todo el examen)

```
ALUMNO (
   id_alumno INT PRIMARY KEY,
   nombre     VARCHAR(50),
   carrera    VARCHAR(50)
);

ASIGNATURA (
   id_asignatura INT PRIMARY KEY,
   nombre        VARCHAR(50),
   creditos      INT
);

MATRICULA (
   id_alumno     INT,
   id_asignatura INT,
   nota          DECIMAL(4,2),
   PRIMARY KEY (id_alumno, id_asignatura),
   FOREIGN KEY (id_alumno) REFERENCES ALUMNO(id_alumno),
   FOREIGN KEY (id_asignatura) REFERENCES ASIGNATURA(id_asignatura)
);
```

Datos

ALUMNO

id_alumno	nombre	carrera
1	Ana	Informática
2	Luis	Informática
3	Marta	Matemáticas
4	Juan	Informática

ASIGNATURA

id_asignatura	nombre	creditos
10	Bases de Datos	6
11	Programación	6
12	Estadística	4

MATRICULA

id_alumno	id_asignatura	nota
1	10	8.5
1	11	7.0
2	10	4.5
3	12	9.0
4	10	NULL

PREGUNTAS SUPUESTO 5

Teniendo en cuenta el esquema y los datos anteriores:

25. ¿Qué consulta devuelve el nombre del alumno y la nota obtenida en cada asignatura?
a) SELECT nombre, nota
FROM ALUMNO, MATRICULA;
b) SELECT A.nombre, M.nota
FROM ALUMNO A
JOIN MATRICULA M ON A.id_alumno = M.id_alumno;
c) SELECT nombre, nota
FROM ASIGNATURA JOIN MATRICULA;
d) SELECT nombre
FROM ALUMNO;

26. ¿Qué consulta devuelve todos los alumnos, aunque no estén matriculados en ninguna asignatura?
a) SELECT *
FROM ALUMNO
JOIN MATRICULA ON ALUMNO.id_alumno = MATRICULA.id_alumno;
b) SELECT *
FROM ALUMNO LEFT
JOIN MATRICULA ON ALUMNO.id_alumno = MATRICULA.id_alumno;
c) SELECT *
FROM MATRICULA LEFT JOIN ALUMNO;
d) SELECT *
FROM ASIGNATURA RIGHT JOIN ALUMNO;

27. ¿Qué consulta devuelve la nota media por asignatura?
a) SELECT nombre, AVG(nota)FROM ASIGNATURA;
b) SELECT A.nombre, AVG(M.nota)
FROM ASIGNATURA A
JOIN MATRICULA M ON A.id_asignatura = M.id_asignatura
GROUP BY A.nombre;
c) SELECT AVG(nota) FROM MATRICULA;
d) SELECT nombre, nota FROM MATRICULA;

28. ¿Qué consulta devuelve las asignaturas cuya nota media es mayor o igual que 7?
a) SELECT A.nombre, AVG(M.nota)
FROM ASIGNATURA A
JOIN MATRICULA M ON A.id_asignatura = M.id_asignatura
WHERE AVG(M.nota) >= 7 GROUP BY A.nombre;
b) SELECT A.nombre, AVG(M.nota)
FROM ASIGNATURA A
JOIN MATRICULA M ON A.id_asignatura = M.id_asignatura
GROUP BY A.nombre HAVING AVG(M.nota) >= 7;
c) SELECT nombre FROM ASIGNATURA WHERE nota >= 7;
d) SELECT nombre FROM MATRICULA GROUP BY nombre;

29. ¿Qué consulta devuelve los alumnos que no se han matriculado en ninguna asignatura?
a) SELECT *
   FROM ALUMNO A
   JOIN MATRICULA M ON A.id_alumno = M.id_alumno
   WHERE M.id_asignatura IS NULL;
b) SELECT *
   FROM ALUMNO A
   LEFT JOIN MATRICULA M ON A.id_alumno = M.id_alumno
   WHERE M.id_asignatura IS NULL;
c) SELECT * FROM MATRICULA WHERE id_alumno IS NULL;
d) SELECT * FROM ALUMNO WHERE id_alumno NOT EXISTS;

PREGUNTA DE RESERVA 5

30. ¿Qué consulta devuelve los alumnos con matrícula pero sin nota asignada?
a) SELECT * FROM MATRICULA WHERE nota = NULL;
b) SELECT * FROM MATRICULA WHERE nota IS NULL;
c) SELECT * FROM MATRICULA WHERE nota = 0;
d) SELECT * FROM MATRICULA WHERE nota < 5;