



## Anexo 2

### Pruebas para la obtención de títulos de Técnico y Técnico Superior

#### Convocatoria correspondiente al curso académico 2023-2024

(Resolución de 29 de diciembre de 2023 de la Dirección General de Educación Secundaria, Formación Profesional y Régimen Especial)

DATOS DEL ASPIRANTE			FIRMA
Apellidos:			
Nombre:	D.N.I. N.I.E. o Pasaporte:	Fecha:	

Código del ciclo:	Denominación completa del título:
IFCS02	Técnico Superior en Desarrollo de Aplicaciones Multiplataforma
Clave o código del módulo:	Denominación completa del módulo profesional:
484	Bases de Datos

INSTRUCCIONES GENERALES PARA LA REALIZACIÓN DE LA PRUEBA
<ul style="list-style-type: none"><li>El examen tendrá una duración de 2 horas.</li><li>La prueba consta de un examen tipo test con cuatro opciones de las cuales solamente una es correcta.</li><li>Cada pregunta se responderá en el espacio dejado al efecto en la hoja de respuestas. Se rellenarán los recuadros para señalar la respuesta seleccionada.</li><li>Si se quiere rectificar una respuesta contestada, se borrará toda la casilla de la respuesta incorrecta con Tipp-Ex o corrector, tal y como se puede apreciar en el siguiente ejemplo:<ul style="list-style-type: none"><li>Se elimina la selección de <b>b</b> para seleccionar la opción <b>d</b>: <input type="checkbox"/>a <input type="checkbox"/>b <input type="checkbox"/>c <input checked="" type="checkbox"/>d</li></ul></li><li>Se dispondrá de una hoja para borrador (o de varias si se requieren), que será proporcionada por el centro. Esa hoja se entregará obligatoriamente al final junto con el examen, si bien nada de lo escrito en la hoja de borrador se valorará en la corrección.</li><li>Sólo se utilizará bolígrafo negro o azul y Tipp-Ex o corrector, no permitiéndose usar bolígrafo rojo, lapicero, etcétera.</li><li>No se podrá emplear ningún dispositivo electrónico.</li><li>Cualquier tachadura o borrón en una respuesta podrá invalidar toda la puntuación de la pregunta asociada.</li></ul>

CRITERIOS DE CALIFICACIÓN Y VALORACIÓN
<ul style="list-style-type: none"><li>El test se calificará sobre 10 puntos. Todas las preguntas se calificarán equitativamente con la misma cantidad de puntos. En cada pregunta se plantearán varias respuestas, y se deberá señalar la única que se considere correcta, según el caso. Cada respuesta correcta que se marque se valorará con 0,25 puntos, y si se marca alguna incorrecta, se valorará con una cantidad negativa equivalente a 0,075 puntos, es decir, se descontarán 0,075 puntos. Si no se está seguro de si una respuesta es correcta o no, y no se marca, no sumará ni restará puntos.</li><li>Calificación final del módulo profesional:<ul style="list-style-type: none"><li>El alumno obtendrá en el módulo profesional una calificación entera entre 1 y 10.</li><li>Dicha calificación se calculará:<ul style="list-style-type: none"><li>Si la calificación conseguida en la prueba es inferior a 5: se truncará dicha calificación.</li><li>Si la calificación conseguida en la prueba es igual o superior a 5 y los decimales:<ul style="list-style-type: none"><li>Son inferiores a 0'5: se redondeará al entero inferior más próximo.</li><li>Son iguales o superiores a 0'5: se redondeará al entero superior más cercano.</li></ul></li></ul></li><li>La anterior regla tiene una excepción: las notas de examen inferiores a 1 se redondearán a 1.</li></ul></li></ul>



## **CONTENIDO DE LA PRUEBA**

- 1) **Considerando las características de los diferentes tipos de ficheros (planos, indexados, de acceso directo, entre otros), ¿cuál de las siguientes afirmaciones es correcta cuando se requiere optimizar el tiempo de búsqueda y actualización de registros en un sistema que gestiona un volumen masivo de datos?**
  - a. Los ficheros planos son más eficientes debido a su estructura simple y fácil manejo.
  - b. Los ficheros indexados ofrecen un mejor rendimiento mediante el uso de índices que agilizan las búsquedas.
  - c. Los ficheros de acceso directo son menos recomendables debido a la complejidad en su estructuración y mantenimiento.
  - d. Los ficheros de marcas son preferibles por su capacidad para almacenar datos de manera no estructurada, facilitando búsquedas rápidas.
- 2) **¿Qué tipo de base de datos sería más apropiada para una aplicación que requiere almacenar y consultar grandes volúmenes de datos relacionales, que además se actualizan frecuentemente y están distribuidos geográficamente?**
  - a. Base de datos relacional centralizada
  - b. Base de datos NoSQL distribuida
  - c. Base de datos relacional distribuida
  - d. Base de datos jerárquica
- 3) **Entre las funciones de los SGBD, una esencial para garantizar la integridad de la información durante transacciones que involucran múltiples operaciones es:**
  - a. El control de redundancia
  - b. La gestión de vistas
  - c. El manejo de transacciones y su atomicidad
  - d. La indexación de datos
- 4) **Considerando las diferencias entre SGBD comerciales y libres, ¿cuál de las siguientes afirmaciones es verdadera en términos de soporte y personalización para una empresa que busca una solución a medida con un soporte técnico garantizado?**
  - a. Los SGBD libres ofrecen alta personalización pero el soporte técnico, no garantizado, depende de la comunidad o servicios externos.
  - b. Los SGBD comerciales no permiten personalización, pero ofrecen soporte técnico integral.
  - c. Los SGBD libres ofrecen soporte técnico más efectivo y personalizado que los comerciales.
  - d. Los SGBD comerciales permiten personalización avanzada y proporcionan soporte técnico garantizado por el proveedor.
- 5) **En el modelo relacional, ¿cuál de las siguientes afirmaciones describe adecuadamente la relación entre relaciones (tablas), atributos (columnas) y tuplas (filas)?**
  - a. Una relación es un conjunto de tuplas donde cada tupla está compuesta por atributos, que definen propiedades únicas de la relación.
  - b. Los atributos son colecciones de relaciones que definen las características de las tuplas en una base de datos.
  - c. Las tuplas son agrupaciones de atributos que determinan las relaciones entre distintas tablas dentro de una base de datos.



- d. Los atributos y las tuplas son independientes entre sí y no contribuyen a la definición de una relación.
- 6) **¿Por qué es crucial seleccionar el tipo de dato adecuado para una clave primaria en una base de datos relacional?**
- a. Para garantizar una organización estética de la base de datos.
  - b. Para asegurar la eficiencia en el uso de espacio de almacenamiento y la velocidad de acceso a los datos.
  - c. Porque el tipo de dato de la clave primaria determina el tipo de datos de todas las otras columnas en la relación.
  - d. Porque simplifica el diseño de índices secundarios.
- 7) **Considerando las características de los índices en bases de datos relacionales, ¿cuál de las siguientes opciones describe correctamente el impacto de permitir valores NULL en una columna indexada?**
- a. Los valores NULL incrementan la velocidad de búsqueda en los índices porque representan ausencia de datos.
  - b. Permitir valores NULL en columnas indexadas no afecta el rendimiento de las operaciones de búsqueda.
  - c. Los valores NULL pueden complicar las operaciones de búsqueda en los índices y potencialmente aumentar el uso del espacio.
  - d. Los valores NULL en columnas indexadas eliminan la necesidad de índices secundarios.
- 8) **¿Cómo contribuyen los roles al manejo de la seguridad en una base de datos relacional?**
- a. Permitiendo a los usuarios ejecutar cualquier acción sin restricciones, para simplificar la administración de la base de datos.
  - b. Facilitando la asignación y administración de privilegios a través de un conjunto de permisos agrupados, mejorando la seguridad y la gestión.
  - c. Restringiendo el acceso a la base de datos únicamente a los usuarios con roles asignados, sin importar sus privilegios específicos.
  - d. Eliminando la necesidad de privilegios, ya que los roles determinan completamente el acceso a la base de datos.
- 9) **En el contexto de la administración de bases de datos, ¿qué lenguaje se utiliza específicamente para crear o modificar la estructura de la base de datos, como tablas y relaciones?**
- a. Lenguaje de Manipulación de Datos (DML)
  - b. Lenguaje de Definición de Datos (DDL)
  - c. Lenguaje de Control de Datos (DCL)
  - d. Lenguaje de Consulta Estructurada (SQL)

Para el siguiente código en SQL:

```
SELECT Autor.Nombre, COUNT(Libro.ID) AS TotalLibros, SUM(Venta.Cantidad) AS TotalVendidos
FROM Autor
JOIN Libro ON Autor.ID = Libro.AutorID
JOIN Venta ON Libro.ID = Venta.LibroID
WHERE Venta.Fecha BETWEEN '2023-01-01' AND '2023-12-31'
```



**GROUP BY Autor.Nombre**

**HAVING SUM(Venta.Cantidad) > 100**

**ORDER BY TotalVendidos DESC;**

**10. ¿Qué tipo de JOIN se utiliza en la consulta para combinar las tablas Autor, Libro, y Venta?**

- a. LEFT JOIN
- b. RIGHT JOIN
- c. INNER JOIN
- d. FULL OUTER JOIN

**11. ¿Qué indica la cláusula HAVING SUM(Venta.Cantidad) > 100 en esta consulta?**

- a. Filtra autores que han vendido más de 100 libros en total antes de agrupar.
- b. Limita los resultados a libros individuales que han vendido más de 100 copias.
- c. Filtra los grupos de autores cuyos libros han vendido más de 100 copias en total.
- d. Establece una condición para mostrar solo las ventas superiores a 100 en el resultado final.

**12. ¿Qué representa COUNT(Libro.ID) AS TotalLibros en la consulta?**

- a. El número total de libros en la base de datos.
- b. El número total de libros vendidos por autor.
- c. El número total de libros distintos escritos por cada autor.
- d. El número total de ventas de libros.

**13. ¿Qué función de agregado se utiliza para determinar el volumen total de ventas por autor?**

- a. COUNT()
- b. SUM()
- c. AVG()
- d. MAX()

**14. En la cláusula WHERE Venta.Fecha BETWEEN '2023-01-01' AND '2023-12-31', ¿qué tipo de datos se espera que tenga el campo Venta.Fecha?**

- a. TEXT
- b. INTEGER
- c. DATE
- d. VARCHAR

**15. ¿Cómo se ordenan los resultados de la consulta final?**

- a. Alfabéticamente por el nombre del autor.
- b. Por el número total de libros en orden ascendente.
- c. Por el número total de ventas en orden descendente.
- d. Por la fecha de venta de los libros.

Para el siguiente programa en SQL:

BEGIN TRANSACTION;

-- Inserción de un nuevo empleado utilizando datos proporcionados



INSERT INTO Empleado (Nombre, DepartamentoID, Salario)

VALUES ('Juan Pérez', (SELECT ID FROM Departamento WHERE Nombre = 'Desarrollo'), 30000);

-- Actualización del salario de todos los empleados del departamento de 'Desarrollo' en un 10%

UPDATE Empleado

SET Salario = Salario \* 1.10

WHERE DepartamentoID = (SELECT ID FROM Departamento WHERE Nombre = 'Desarrollo');

-- Borrado de empleados que no han sido asignados a ningún proyecto

DELETE FROM Empleado

WHERE ID NOT IN (SELECT EmpleadoID FROM ProyectoEmpleado);

COMMIT;

**16. ¿Qué garantiza el uso de la instrucción BEGIN TRANSACTION al inicio del programa?**

- a. Que las consultas se ejecuten más rápidamente.
- b. Que las modificaciones en la base de datos sean permanentes inmediatamente.
- c. Que todas las operaciones se realicen como una única unidad de trabajo.
- d. Que se puedan ejecutar múltiples consultas simultáneamente.

**17. ¿Cómo se asegura que el nuevo empleado 'Juan Pérez' sea asignado al departamento correcto?**

- a. Utilizando un JOIN con la tabla Departamento.
- b. Mediante una subconsulta que recupera el ID del departamento de 'Desarrollo'.
- c. Asignando un ID de departamento fijo.
- d. A través de una transacción separada.

**18. ¿Qué efecto tiene la cláusula WHERE DepartamentoID = (SELECT ID FROM Departamento WHERE Nombre = 'Desarrollo') en la actualización del salario?**

- a. Incrementa el salario de todos los empleados de la empresa.
- b. Reduce el salario de los empleados del departamento de 'Desarrollo'.
- c. Incrementa el salario solo de los empleados del departamento de 'Desarrollo'.
- d. Cambia el departamento de todos los empleados a 'Desarrollo'.

**19. ¿Cuál es el propósito de DELETE FROM Empleado WHERE ID NOT IN (SELECT EmpleadoID FROM ProyectoEmpleado);?**

- a. Eliminar todos los proyectos que no tienen empleados asignados.
- b. Borrar los registros de proyectos de empleados específicos.
- c. Eliminar empleados que no estén asignados a ningún proyecto.
- d. Asignar empleados a proyectos existentes.



**20. ¿Qué efecto tendría en la base de datos si la instrucción COMMIT; fuese reemplazada por ROLLBACK;?**

- a. Confirmaría todas las operaciones realizadas en la transacción.
- b. No tendría ningún efecto en las operaciones realizadas.
- c. Desharía todas las operaciones realizadas desde BEGIN TRANSACTION.
- d. Ejecutaría nuevamente todas las operaciones para verificar errores.

**21. Considerando la integridad referencial, ¿qué precaución debe tenerse al borrar empleados no asignados a proyectos?**

- a. Asegurarse de que no existan referencias a esos empleados en otras tablas relacionadas.
- b. Actualizar los salarios de los empleados antes de borrarlos.
- c. Insertar nuevos empleados para reemplazar a los que se borraron.
- d. Verificar que los empleados tengan salarios asignados antes de borrarlos.

Para el siguiente código en SQL:

```
CREATE PROCEDURE RegistrarPrestamo(  
  
    IN LibroID INT,  
  
    IN UsuarioID INT,  
  
    IN FechaDevolucion DATE  
  
)  
  
BEGIN  
  
    DECLARE EstadoLibro VARCHAR(20);  
  
  
    -- Comprobar el estado actual del libro  
  
    SELECT Estado INTO EstadoLibro FROM Libros WHERE ID = LibroID;  
  
  
    IF EstadoLibro = 'Disponible' THEN  
  
        -- Actualizar el estado del libro a 'Prestado'  
  
        UPDATE Libros SET Estado = 'Prestado' WHERE ID = LibroID;  
  
  
        -- Registrar el préstamo  
  
        INSERT INTO Prestamos(LibroID, UsuarioID, FechaPrestamo, FechaDevolucion)  
  
        VALUES (LibroID, UsuarioID, CURDATE(), FechaDevolucion);  
  
    ELSE  
  
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El libro no está disponible para préstamo.';
```



END IF;

END;

**22. ¿Qué operación realiza el procedimiento almacenado RegistrarPrestamo si el libro está disponible?**

- a. Elimina el registro del libro de la base de datos.
- b. Actualiza el estado del libro a 'Prestado' y registra el préstamo.
- c. Registra solamente la fecha de devolución del libro.
- d. No realiza ninguna operación y finaliza la ejecución del procedimiento.

**23. ¿Cómo se maneja la situación cuando el libro solicitado no está disponible para préstamo?**

- a. Continuando la ejecución sin realizar cambios.
- b. Actualizando el estado del libro a 'En reparación'.
- c. Lanzando una excepción con un mensaje de error específico.
- d. Cambiando el estado del libro a 'Disponible' automáticamente.

**24. ¿Qué significa DECLARE EstadoLibro VARCHAR(20); dentro del procedimiento almacenado?**

- a. Selecciona el estado del libro desde la base de datos.
- b. Crea una variable temporal para almacenar el estado del libro.
- c. Define una constante que no puede cambiar durante la ejecución del procedimiento.
- d. Establece el valor predeterminado del estado del libro a una cadena de 20 caracteres.

**25. En el contexto de este procedimiento, ¿cuál es el propósito de SIGNAL SQLSTATE '45000'?**

- a. Notificar al sistema de que la operación ha sido exitosa.
- b. Lanzar una excepción controlada para indicar que el libro no está disponible.
- c. Crear un nuevo estado en la base de datos para el libro.
- d. Registrar el estado del préstamo en un log de errores.

**26. ¿Qué permite CURDATE() en la sentencia INSERT?**

- a. Insertar la fecha actual como la fecha de préstamo.
- b. Calcular automáticamente la fecha de devolución basada en la fecha actual.
- c. Registrar la fecha y hora exactas en que se ejecutó el procedimiento.
- d. Comparar la fecha de devolución con la fecha actual para validar la entrada.

**27. ¿Cuál de las siguientes afirmaciones es verdadera respecto al ámbito de la variable EstadoLibro?**

- a. Puede ser accedida y modificada por cualquier procedimiento dentro de la base de datos.
- b. Solo existe dentro del procedimiento RegistrarPrestamo y no es accesible fuera de él.
- c. Se almacena de forma permanente en la base de datos una vez que el procedimiento termina.
- d. Su valor se resetea a NULL cada vez que se invoca el procedimiento.

**Se tiene las siguientes entidades y sus atributos:**

#### **Empleado**

Atributos: EmpleadoID (PK), Nombre, Apellido, Email, DepartamentoID (FK)

#### **Proyecto**

Atributos: ProyectoID (PK), NombreProyecto, FechaInicio, FechaFin, GerenteID (FK)



## Tarea

Atributos: TareaID (PK), Descripción, FechaInicio, FechaFin, Estado, ProyectoID (FK)

## Departamento

Atributos: DepartamentoID (PK), NombreDepartamento, Ubicación

## Relaciones:

**Empleado-Departamento:** Un departamento puede tener muchos empleados, pero un empleado pertenece a un solo departamento. (1:N)

**Empleado-Proyecto:** Un empleado puede trabajar en muchos proyectos, y un proyecto puede tener muchos empleados asignados. (M:N, relación "Asignación")

**Proyecto-Tarea:** Un proyecto consiste en muchas tareas, pero una tarea pertenece a un solo proyecto. (1:N)

## Diagrama E/R

Imagina un diagrama E/R que visualmente representa las entidades descritas con sus atributos y las relaciones entre ellas, incluyendo cardinalidades y posiblemente identificando entidades fuertes vs. débiles.

### 28. ¿Qué indica la relación 1:N entre la entidad "Departamento" y "Empleado"?

- a. Un empleado puede pertenecer a muchos departamentos.
- b. Un departamento puede tener muchos empleados.
- c. Cada empleado puede gestionar un departamento.
- d. Los departamentos están subdivididos en empleados.

### 29. En la relación M:N entre "Empleado" y "Proyecto", ¿qué estructura se necesita para implementar esta relación en el modelo relacional?

- a. Una tabla de unión que contenga las claves primarias de ambas entidades como claves foráneas.
- b. Un atributo adicional en cada entidad para apuntar a la otra entidad.
- c. Una clave foránea en la entidad "Empleado" que apunte a "Proyecto".
- d. Una restricción de unicidad en los atributos de ambas entidades.

### 30. ¿Cuál de los siguientes atributos sería más adecuado como clave primaria para la entidad "Tarea"?

- a. Descripción
- b. FechaInicio
- c. TareaID
- d. Estado

### 31. ¿Qué representa el atributo GerenteID en la entidad "Proyecto"?

- a. Una clave foránea que enlace a la entidad "Empleado" indicando quién gestiona el proyecto.
- b. Un identificador único para cada gerente de proyecto.
- c. Un atributo descriptivo del proyecto.
- d. La cantidad de gerentes en un proyecto.





**32. ¿Cómo se modelaría la relación entre "Proyecto" y "Tarea" para asegurar que todas las tareas estén asociadas a un proyecto?**

- a. Mediante una restricción de integridad referencial que obligue a cada "Tarea" a tener un "ProyectoID" válido.
- b. Asignando un "TareaID" único a cada "Proyecto".
- c. Creando una tabla separada que liste todas las tareas posibles para cada proyecto.
- d. Usando el atributo "Estado" en "Tarea" para indicar a qué proyecto pertenece.

**33. Considerando la normalización hasta la Tercera Forma Normal (3FN), ¿cuál sería un indicativo de que la entidad "Empleado" está correctamente normalizada?**

- a. Todos los atributos no clave dependen del todo de la clave primaria.
- b. "EmpleadoID" es parte de la clave compuesta de otra entidad.
- c. Los atributos "Nombre" y "Apellido" se combinan en un único atributo "NombreCompleto".
- d. La entidad "Empleado" incluye un atributo "NombreProyecto" para identificar proyectos asignados.

**Para el siguiente programa en SQL**

-- Definición de un tipo de objeto para Empleado

CREATE TYPE EmpleadoTipo AS OBJECT (

EmpleadoID NUMBER,

Nombre VARCHAR2(100),

Email VARCHAR2(100),

Posicion REF PosicionTipo,

Proyectos LIST<REF ProyectoTipo>

);

-- Definición de un tipo de objeto para Posición

CREATE TYPE PosicionTipo AS OBJECT (

PosicionID NUMBER,

Titulo VARCHAR2(100),

SalarioBase NUMBER

);

-- Definición de un tipo de objeto para Proyecto

CREATE TYPE ProyectoTipo AS OBJECT (

ProyectoID NUMBER,



```
NombreProyecto VARCHAR2(100),  
  
FechaInicio DATE,  
  
FechaFin DATE  
  
);
```

-- Inserción de un objeto Posición

DECLARE

```
pos1 PosicionTipo := PosicionTipo(1, 'Desarrollador Senior', 60000);
```

BEGIN

```
INSERT INTO Posiciones VALUES (pos1);
```

END;

-- Inserción de un objeto Empleado con referencia a Posición

DECLARE

```
emp1 EmpleadoTipo;
```

```
posRef REF PosicionTipo;
```

BEGIN

```
SELECT REF(p) INTO posRef FROM Posiciones p WHERE p.PosicionID = 1;
```

```
emp1 := EmpleadoTipo(1, 'Juan Pérez', 'juan.perez@empresa.com', posRef, NULL);
```

```
INSERT INTO Empleados VALUES (emp1);
```

END;

**34. ¿Qué característica de las bases de datos objeto-relacionales permite definir el tipo EmpleadoTipo?**

- a. Polimorfismo
- b. Herencia
- c. Encapsulación
- d. Definición de tipos de objeto

**35. ¿Cómo se establece la relación entre un Empleado y su Posición en este programa?**

- a. Mediante un identificador único
- b. Usando una lista de referencias
- c. A través de una referencia a PosicionTipo
- d. Con una clave foránea tradicional

**36. ¿Qué representa LIST<REF ProyectoTipo> en la definición de EmpleadoTipo?**



- a. Un tipo de datos escalar
- b. Una colección de identificadores únicos
- c. Un tipo de datos colección que almacena referencias a objetos Proyecto
- d. Una lista de proyectos serializados como cadenas de texto

**37. ¿Qué operación realiza la sentencia `SELECT REF(p) INTO posRef FROM Posiciones p WHERE p.PosicionID = 1`;**

- a. Inserta una nueva posición en la base de datos
- b. Actualiza la referencia de una posición existente
- c. Recupera una referencia a un objeto PosicionTipo existente
- d. Elimina la referencia a una posición

**38. En la inserción de `emp1`, ¿cómo se inicializa el atributo `Proyectos`?**

- a. Con una lista vacía
- b. Con un objeto ProyectoTipo específico
- c. Con una referencia nula
- d. Mediante una subconsulta que recupera proyectos existentes

**39. ¿Qué ventaja ofrece el uso de `REF PosicionTipo` en `EmpleadoTipo` para la gestión de la base de datos objeto-relacional?**

- a. Permite almacenar múltiples posiciones para un solo empleado, aumentando la flexibilidad del modelo.
- b. Facilita la integridad referencial automática sin necesidad de triggers o procedimientos almacenados adicionales.
- c. Proporciona navegabilidad directa desde un objeto EmpleadoTipo a su correspondiente PosicionTipo, mejorando la eficiencia de las consultas.
- d. Reduce la cantidad de espacio necesario para almacenar información de la posición, ya que solo se almacena una vez en la base de datos.

**40. Dado que el atributo `Proyectos` en `EmpleadoTipo` es un `LIST<REF ProyectoTipo>`, ¿cuál sería la manera correcta y efectiva de asociar un nuevo proyecto a un empleado existente en la base de datos?**

- a. Mediante la actualización directa de la lista `Proyectos` con el nuevo `REF ProyectoTipo` usando una operación de inserción SQL estándar.
- b. Utilizando una sentencia `UPDATE` para reemplazar la lista `Proyectos` existente con una nueva que incluye el nuevo proyecto.
- c. Creando un procedimiento almacenado que maneje la lógica de añadir una nueva referencia a proyecto en la lista `Proyectos` de un empleado.
- d. Inserción directa del nuevo proyecto en la tabla `Proyectos` sin modificar la lista `Proyectos` en `EmpleadoTipo`, ya que la actualización es automática.