



Introducción

La Universidad Politécnica de Wisconsin (UPW) a través de su servicio TIC gestiona la compra de las licencias de software para toda la comunidad universitaria. Habitualmente los costes eran asumidos por la propia universidad y las licencias ofertadas a los centros, departamentos e institutos sin coste para ellos. El aumento de los costes y el mal uso de las licencias por parte de algunos usuarios ha llevado a una situación donde es necesario que se establezca un sistema de copago de las licencias por parte de los centros, departamentos o institutos donde se van a utilizar.

Para gestionar la solicitud de licencias por parte de las entidades de la UPW se necesita desarrollar una aplicación informática en la intranet, llamada Tienda de Software donde los gestores de las entidades entren a seleccionar el tipo de licencias y el número requerido.

Funcionalidades de la aplicación

La aplicación debe permitir que los gestores entren y seleccionen las licencias, que están organizadas por la categoría que tienen asignada.

Las licencias disponibles podrán ser con control de stock disponible o sin control de stock (lo que significa que son licencias ilimitadas). Para cada licencia se indica su precio y una imagen.

Las categorías de las licencias están organizadas de forma jerárquica multinivel, indicando en cada categoría cuál es su categoría padre.

En los pedidos que realicen los gestores, deberán indicar el centro de coste en el que se cargará el importe de las licencias y el conjunto de licencias solicitadas, que compondrán las líneas del pedido.

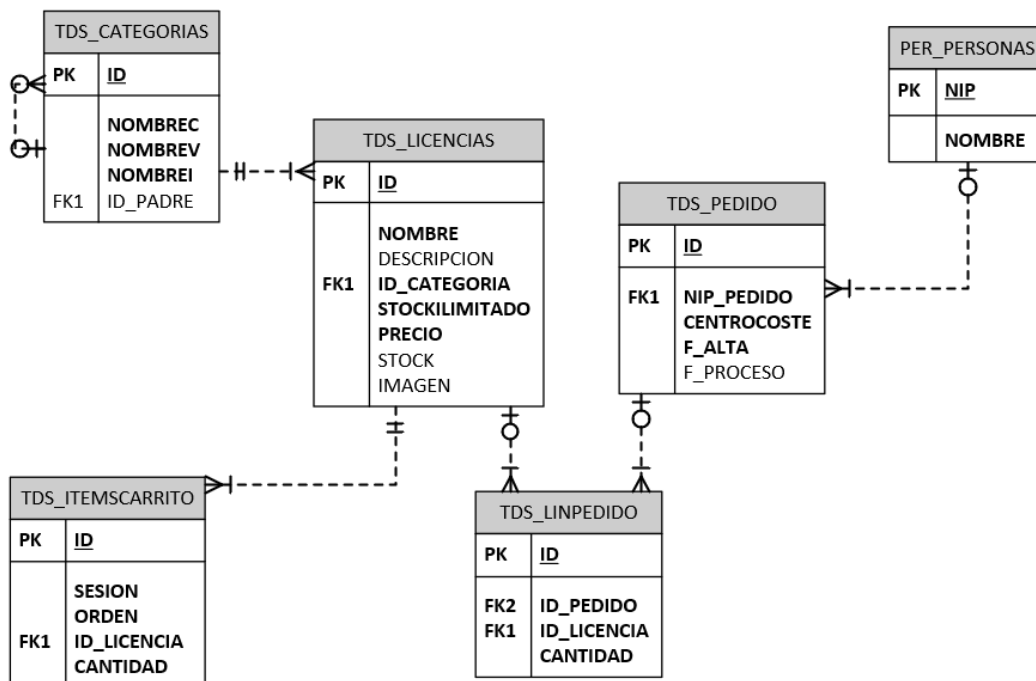
Para identificar a los gestores se usará la tabla de PER_PERSONAS de la universidad donde todos los usuarios ya están dados de alta.

El funcionamiento de la aplicación será como el de una tienda online donde el usuario, estando ya identificado, podrá ir incluyendo las licencias que quiere adquirir a su carrito. Cuando desee finalizar la compra, se le mostrará el carrito y tras completar el dato del centro de coste se le generará el pedido correspondiente.



Diagrama de Tablas

La aplicación ya tiene las tablas definidas en Oracle 19c y son estas:



Leyenda:

Los campos NO NULOS están marcados en NEGRITA

FK: Claves ajenas o Foreign Keys

PK: Clave primaria o Primary Key



Supuesto 1

Se desea realizar ciertos cambios en la aplicación para permitir lo siguiente:

Pregunta 1 (5 puntos): Se desea obtener una vista en Oracle 19c con la relación de categorías clasificada en estructura de árbol. Se deberá indicar el nivel de la categoría dentro del árbol, el identificador y nombre de la categoría. Deberá mostrarse ordenado jerárquicamente, teniendo en cuenta que el nodo raíz es el único que tiene ID_PADRE a nulo y tiene como ID=0.

Completar el siguiente cuadro de creación de la vista.

Respuesta 1:

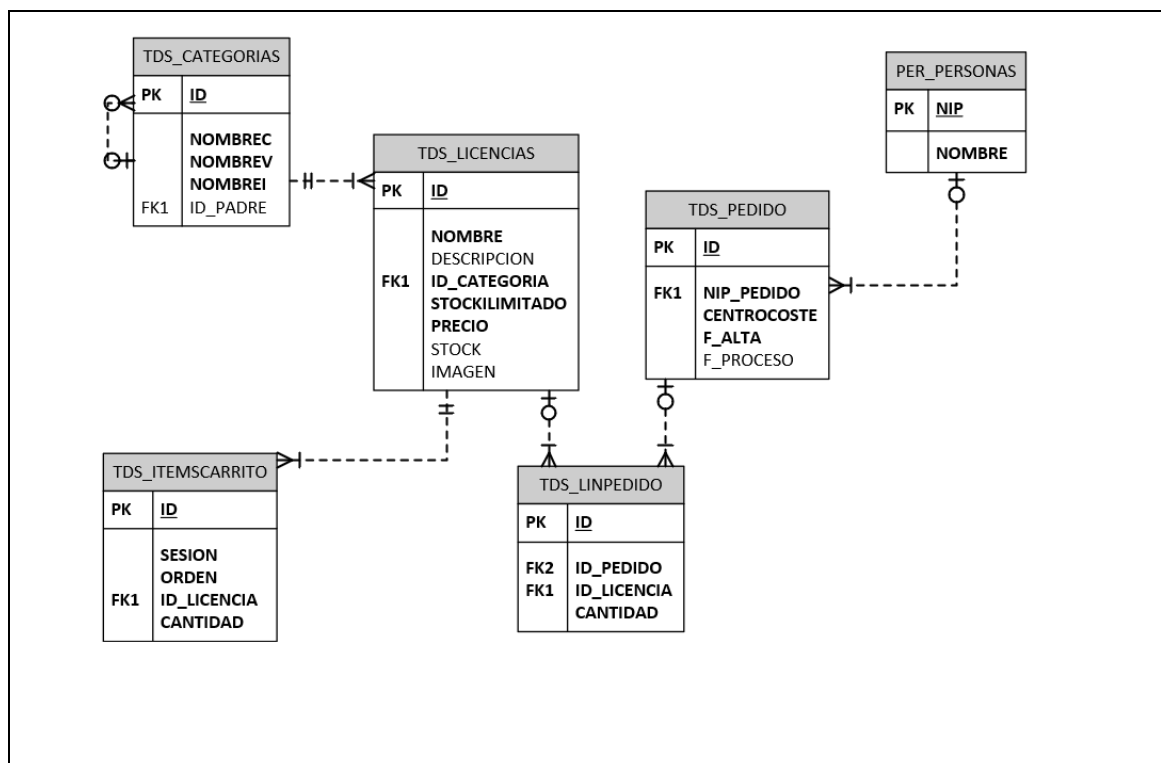
```
CREATE VIEW TDS_ARBOL_CATEGORIAS_VW AS
```

```
SELECT
```

```
FROM TDS_CATEGORIAS
```

Pregunta 2 (5 puntos): Se quiere identificar a las personas que podrán usar las licencias compradas por el gestor de la entidad. Para esto será necesario modificar la estructura de datos de tal manera que se puedan identificar las personas para las que se adquieren licencias en cada línea de pedido. Completar en el diagrama la nueva tabla con los campos necesarios y sus relaciones, teniendo en cuenta que todas las personas están ya en la tabla PER_PERSONAS. Esta tabla se utilizará posteriormente para registrar la fecha de activación y de caducidad de cada licencia.

Respuesta 2:





Pregunta 3 (5 Puntos): El stock deberá actualizarse automáticamente cuando se cree el pedido. Para solucionar esta funcionalidad se debe crear un TRIGGER sobre la tabla de líneas de pedido que actualizará el stock de licencias. Se debe asumir que no se contemplan actualizaciones ni borrados de líneas de pedido.

Respuesta 3:

```
CREATE OR REPLACE TRIGGER TDS_LINPEDIDO_TR1
```

```
AFTER ..... ON .....
```

```
FOR EACH ROW
```

```
DECLARE
```

```
BEGIN
```

```
END
```



Supuesto 2

Pregunta 4 (4 puntos): Se desea implementar un servicio REST dentro de la aplicación, que permita realizar operaciones en la base de datos relativas al carrito (TDS_ITEMSCARRITO) cada vez que el usuario añade o elimina un ítem del carrito.

Respuesta 4a: Completa el siguiente cuadro de especificación del servicio para añadir un ítem al carrito

Método HTTP	
Justificación del método HTTP utilizado	
Ruta (endpoint) con versionado	
Ejemplo del cuerpo (body) de la solicitud en formato JSON.	
Código HTTP devuelto por el REST en caso de éxito	
Ejemplo del cuerpo de la respuesta (HTTP CODE y response) en formato JSON en caso de éxito	



Respuesta 4b: Completa el siguiente cuadro de especificación del servicio para eliminar un ítem del carrito

Método HTTP	
Justificación del método HTTP utilizado	
Ruta (endpoint) con versionado	
Ejemplo invocación REST	
Código HTTP devuelto por el REST en caso de éxito	
Ejemplo del cuerpo de la respuesta (HTTP CODE y response) en formato JSON en caso de éxito	



Pregunta 5 (5 puntos): A partir del Diagrama Entidad-Relación de la aplicación, complete las anotaciones JPA de las siguientes clases Java de entidad.

Como requisito de lógica de negocio, implemente el método de instancia public double calcularImporteTotal() dentro de la clase TdsPedido. Este método debe calcular y retornar el coste total del pedido.

Respuesta 5a: Completa la Clase TdsLicencias

```
@
@
public class TdsLicencia {

    @
    @
    private Long id;

    @
    private String nombre;

    @
    private double precio;

    @
    private Integer stock;

    @
    @
    private TdsCategorias categoria;

    public TdsLicencia() {}

    public TdsLicencia(Long id, double precio) {
        this.id = id;
        this.precio = precio;
    }
    /* Getters y Setters se presuponen NO ES NECESARIO COMPLETARLO*/
}
```



Respuesta 5b: Completa la Clase TdsPedido

```
@
@
public class TdsPedido {

    @
    @
    private Long id;

    @
    @
    private TdsPersonas persona;

    @
    private String centroCoste;

    @
    private Date fechaAlta;

    @
    private Date fechaProceso;

    @
    private List<TdsLinPedido> lineasPedido;

    public TdsPedido() {}

    public TdsPedido(Long id, List<TdsLinPedido> lineasPedido) {
        this.id = id;
        this.lineasPedido = lineasPedido;
    }

    // =====
    // MÉTODO DE CÁLCULO DEL IMPORTE TOTAL DEL PEDIDO
    // =====

    /**
     * Calcula el importe total del pedido.
     * @return El importe total del pedido como un valor double.
     */
    public double calcularImporteTotal() {

    }

    /* Getters y Setters se presuponen NO ES NECESARIO COMPLETARLO */
}
```



Pregunta 6 (3 puntos): Se quiere crear la pantalla de compra de licencias mediante carrito de la compra. Para ello se ha realizado una pantalla con el componente CARDS de APEX para que sea lo más parecido a otras tiendas on-line.

Teniendo en cuenta que los datos de la tabla TDS_LICENCIAS son los siguientes

ID	NOMBRE	ID_CATEGORIA	PRECIO	STOCK	IMAGEN
1	LICENCIA WORD 2025	1	299,9	450	(BLOB)
2	LICENCIA EXCEL 2020	2	123	100	(BLOB)
3	LICENCIA ADOBE 2022	3	450	200	(BLOB)
4	LICENCIA WORD 2023	1	100	25	(BLOB)
5	LICENCIA WORD 2020	1	25	10	(BLOB)
6	LICENCIA EXCEL 2025	2	300	125	(BLOB)
7	LICENCIA ADOBE 2025	3	790	350	(BLOB)

El aspecto que debe presentar la pantalla es el siguiente:

Compra de licencias sw

nobody

 LICENCIA WORD 2025 450

PRECIO: 299.9 euros

Añadir al carrito

 LICENCIA EXCEL 2020 100

PRECIO: 123 euros

Añadir al carrito

 LICENCIA ADOBE 2022 200

PRECIO: 450 euros

Añadir al carrito

 LICENCIA WORD 2023 25

PRECIO: 100 euros

Añadir al carrito

 LICENCIA WORD 2020 10

PRECIO: 25 euros

Añadir al carrito

 LICENCIA EXCEL 2025 125

PRECIO: 300 euros

Añadir al carrito

 LICENCIA ADOBE 2025 350

PRECIO: 790 euros

Añadir al carrito



Respuesta 6a: Rellenar todos los campos que se consideren necesarios del apartado SOURCE del componente CARDS para mostrar las licencias de las categorías 1,2 y 3.

Page 666: Licencias

Pre-Rendering

Components

Body

Licencias

Actions

Añadir al carrito

Post-Rendering

Filter

Identification

Source

Location: Local Database

Type:

Table Owner: Parsing Schema

Table Name:

Include ROWID Column: ☐

Where Clause:

Page Items to



Respuesta 6b: Rellenar todos los campos que se consideren necesarios de los apartados CARD, TITLE, BODY e ICON AND BADGE para que el aspecto de la pantalla sea el mostrado en la imagen (no es necesario rellenar los campos CSS Classes)

Page 666: Licencias

Pre-Rendering

Components

Body

Licencias

Actions

Añadir al c

Post-Rendering

Filter

Appearance

Card

CSS Classes

Primary Key Column 1

Primary Key Column 2

Title

Advanced Formatting

Column

CSS Classes

Subtitle

Body

Advanced Formatting

HTML Expression

Secondary Body

Icon and Badge

Icon Source

Image Column

Icon CSS Classes

Icon Position

Icon Description

Badge Column

Badge Label

Badge CSS Classes



Pregunta 7 (3 puntos): Se tiene el backoffice de la aplicación implementado en APEX y se tiene un INTERACTIVE GRID donde hacer el mantenimiento de las licencias tal y como se observa en la imagen.

Compra de licencias sw

nobody

Search: All Text Columns

Go

Actions

Edit

Save

Add Row

Reset

☐		Id	Nombre	Precio	Categoría	Stock	
☒		1	LICENCIA WORD 2025	299.9	WORD	450	Subir imagen
☐		4	LICENCIA WORD 2023	100	WORD	25	Subir imagen
☐		5	LICENCIA WORD 2020	25	EXCEL	10	Subir imagen
☐		2	LICENCIA EXCEL 2020	123	ADOBE	100	Subir imagen
☐		6	LICENCIA EXCEL 2025	300	EXCEL	125	Subir imagen
☐		3	LICENCIA ADOBE 2022	450	ADOBE	200	Subir imagen
☐		7	LICENCIA ADOBE 2025	790	ADOBE	350	Subir imagen

1 rows selected

Total 7



Respuesta 7: Rellene las propiedades necesarias del campo CATEGORIA para que recupere los valores de la tabla y aparezca el desplegable con la descripción de las categorías.

Page 777: Mantenimiento de Llcencia

Pre-Rendering

Components

- Body
 - Mantenimiento de Llcencias
 - Columns
 - ☒ APEX\$ROW_SELECTOR
 - ☒ APEX\$ROW_ACTION
 - ☐ ID
 - ☐ NOMBRE
 - ☒ CATEGORIA
 - ☐ PRECIO
 - ☐ STOCK
 - ☐ Imagen

Column Groups

Saved Reports

Post-Rendering

Filter

List of Values

Type: SQL Query

SQL Query

Display Extra Values: ☐

Display Null Value: ☐

Cascading List of Values

Parent Column(s)

Link

Target: No Link Defined

Source

Type: Database Column

Database Column

Data Type

Query Only: ☐

Primary Key: ☐

Master Detail

Master Column: - Select -

Default

Type: - Select -