

Sistemas de bases de datos II

Ingeniería Informática

2024/2025

Trabajo-II

Desarrollar una aplicación Java que gestione los menús de un restaurante a través de una base de datos PostgreSQL.

Base de datos

Requisitos (si lo crees necesario puedes incluir más campos):

- Por cada menú es necesario tener registrado:
 - id**: Permite identificar a cada menú. Su valor se obtendrá de una secuencia SEQ_MENUS.
 - nombre**: cadena obligatoria con un máximo de 100 caracteres que representa el nombre del menú (por ejemplo: "Menú Navidad", "Menú Noche Vieja",)
 - precio**: representa el precio del plato (valor numérico con un máximo de tres dígitos en la parte entera y 2 dígitos en la parte decimal).
 - desde**: fecha a partir de la que los clientes pueden pedir el menú (el dato es obligatorio)
 - hasta**: fecha tope hasta la que los clientes pueden pedir el menú (el dato es obligatorio)
- Por cada plato es necesario tener registrado:
 - id**: Permite identificar a cada plato. Su valor se obtendrá de una secuencia SEQ_PLATOS.
 - nombre**: cadena de 150 caracteres máximo que indica nombre del plato (el dato es obligatorio). No puede haber dos platos con el mismo nombre.
 - descripcion**: cadena de 450 caracteres máximo que representa la descripción del plato (el dato es obligatorio).
 - precio**: representa el precio del plato (valor numérico con un máximo de tres dígitos en la parte entera y 2 dígitos en la parte decimal).
 - tipo**: cadena que representa el tipo del plato. Los únicos valores permitidos son: ENTRANTE, PRINCIPAL, POSTRE e INFANTIL (el dato es obligatorio).
- Por cada ingrediente es necesario saber:
 - id**: Permite identificar a cada plato. Su valor se obtendrá de una secuencia SEQ_INGREDIENTES.
 - nombre**: cadena de 150 caracteres máximo que indica nombre del plato (el dato es obligatorio). No puede haber 2 ingredientes con el mismo nombre.

Además, hay que tener en cuenta que:

- Un menú estará formado por muchos platos
- Un plato estará compuesto por muchos ingredientes. En esta relación, habría que indicar por cada ingrediente la cantidad utilizada y la unidad de medida. Por ejemplo: 100 gramos, 4 unidades, 250 centilitros. Las unidades de medida únicamente podrán ser: GRAMOS, UNIDADES, CENTILITROS. La cantidad será un entero con un máximo de 4 dígitos.

Casos de uso:

Plato registrarPlato(String nombre, String descripcion, double precio, Tipo tipo, List<CantidadIngrediente> cantidadesIngredientes)

Esta funcionalidad añade un nuevo plato que podrá servirse en el restaurante. El identificador del plato se obtendrá de la secuencia SEQ_PLATOS. El parámetro cantidadesIngredientes identifica cuáles son los ingredientes del plato y que cantidad hay que utilizar de cada uno de ellos. Un objeto CantidadIngrediente consta de tres propiedades: nombre (del ingrediente), cantidad y unidad medida. Si no existiera alguno de los ingredientes, se añadiría a la base de datos (su identificador se obtendría de la secuencia SEQ_INGREDIENTES). El objeto Plato devuelto incluirá los datos del plato (incluyendo su identificador) así como una lista de objetos Compuesto que representan los componentes del plato (un objeto Compuesto incluye un objeto Ingrediente, así como la cantidad y unidad de medida utilizados para el plato)

Menu registrarMenu(String nombre, LocalDate desde, LocalDate hasta, List<String> platos)

Esta funcionalidad registra un nuevo menú en la base de datos. El parámetro platos representa los nombres de los platos que conforman el menú. El identificador del menú se obtendrá de la funcionalidad SEQ_MENU. El precio del menú tendrá un 15% de descuento respecto al precio de cada uno de los platos. El objeto Menu devuelto incluye los datos del menú así como un map que representa los platos. En este map, la clave representa el tipo de plato (ENTRANTE, PRINCIPAL, POSTRE) y el valor es una lista formada por los platos del menú correspondientes a ese tipo. Si a la hora de registrar un menú no existiera alguno de los platos, se propagaría una excepción y no se crearía el menú.

List<Menu> buscarMenu(LocalDate fecha)

Esta funcionalidad devuelve los objetos menús disponibles para la fecha indicada por el parámetro. Cuando el programa muestre en pantalla un menú lo hará utilizando la siguiente plantilla:

```
Menú: Nombre del menú
      Precio: precio del menú
      Platos:
        ENTRANTE:
          Nombre del plato
          Nombre del plato
        PRINCIPAL:
          Nombre del plato
          Nombre del plato
        POSTRE:
          Nombre del plato
          Nombre del plato
          Nombre del plato
```

List<Plato> buscarPlato(Tipo tipo, List<String> ingredientes)

Esta funcionalidad devuelve los Platos que carecen de los ingredientes cuyos nombres se pasan como parámetro.

void subirPrecioPlato(String nombre, double porcentaje)

Esta funcionalidad incrementa el precio del plato en un determinado porcentaje. Esto implicará recalcular el precio de todos los menús en los que se sirva dicho plato. El porcentaje es un valor comprendido entre 0 y 1 (0.8 representaría un incremento del 80%)

Métodos del Dao:

Deberás definir qué funcionalidades necesitas incluir en el Dao.

Consideraciones para entregar el trabajo:

- El trabajo se realizará de forma individual o en grupo formado por un máximo de tres personas (matricula ordinaria). El nombre del archivo ZIP debe coincidir con el nombre y apellidos del alumno/a (o nombre y apellidos de los miembros del equipo)
- Se entregará obligatoriamente un fichero ZIP cuyo nombre será trabajo1.zip. Este fichero incluirá dos carpetas: doc y app.
 - o Carpeta **doc**:
 - Incluirá obligatoriamente un documento: trabajo2.pdf
 - Documento trabajo2.pdf. Este documento incluirá:
 - ♦ Nombre del estudiante (o estudiantes que conforman el grupo)
 - ♦ Descripción funcional de la aplicación.
 - ♦ Tareas realizadas por cada miembro del equipo
 - o Carpeta **app**:
 - Incluirá el proyecto IntelliJ correspondiente a la aplicación desarrollada. El proyecto deberá estar limpio, es decir, carecerá de la subcarpeta target.

Calificación

- Aprobado: Definir correctamente la base de datos e implementar correctamente las funcionalidades: **registrarPlato()**, **registrarMenu()** y **buscarMenu()**. El proyecto IntelliJ debe incluir un fichero **database.sql** con las sentencias SQL correspondientes para la creación de las tablas y secuencias. La aplicación deberá incluir una capa CasoDeUso donde se incluirán las funcionales solicitadas y una capa Dao a través de la que se accederá a la base de datos. La función main() deberá crear ocho platos de distinto tipo y tres menús (que se servirán en las mismas fechas), luego deberá buscar los menús que podrán servirse en una determinada fecha y se mostrarán en pantalla.
- Si la aplicación se hace en grupo, la funcionalidad buscarMenu() deberá implementarse a través de una funcionalidad del Dao en la que se realice una consulta SELECT utilizando JOINS.
- Notable: Sólo se podrá optar a la calificación Notable si todo lo especificado para la calificación Aprobado está correctamente implementado. Además habría que implementar correctamente las funcionalidades **buscarPlato()** y **subirPrecioPlato()**.
- Sobresaliente: Dependerá de la calidad del código implementado (utilización de JOINS, reutilización de sentencias, no repetición de código).