

- **Documento explicativo del código DAO**

- **Interfaz Statement**

- Gracias a esta interfaz, las órdenes SQL individuales son enviadas a la base de datos a través del controlador JDBC, y son recogidos los resultados de estas. Es importante señalar que esta interfaz protege la aplicación de las inyecciones SQL, y con dicha protección se evita que cualquier usuario introduzca sentencias no deseadas en la base de datos.

- **Clase Ingredient**

- En esta clase se definen los atributos correspondientes a los ingredientes, que son nombre y precio, y dicha clase hereda de Entity para obtener el id y los métodos relacionados con el mismo (generarlo de forma aleatoria, establecerlo y obtenerlo).

- **Interfaz Runables**

- En esta interfaz se hayan dos métodos: el primero es getSQL(), cuya finalidad es llamar a la transacción o consulta SQL. El segundo método, que es llamado run, sirve para ejecutar un PreparedStatement, que definirá la sentencia SQL que permitirá insertar el id, el nombre y el precio de los ingredientes.

- **Implementación de la interfaz Runables**

- En este archivo se implementa la interfaz Runables. Se trata de una clase genérica, ya que toma el valor de cualquier cosa que le sea indicada. Sus atributos son sql, entity y statement. Hay dos métodos: el primero devuelve el atributo SQL y el segundo hace una PreparedStatement de los atributos entity y statement.

- **Interfaz Resultset**

- En esta interfaz se encuentra el método run, que recibe como parámetros un Resultset con los resultados de la sentencia SQL y una variable entity genérica, lanzando una excepción SQL si hay un error.

- **Interfaz Configuration**

- En esta interfaz son definidos los métodos getUser, getPassword y getUrl, para obtener sus respectivos atributos.

- **Implementación de la interfaz Configuration**

- En este archivo se implementa la interfaz Configuración, inicializándola a null. Se define un método de esta interfaz donde, si fuera null, se crease una nueva configuración y se devuelve esa misma configuración. Para terminar, se definen las variables de entorno en el sistema operativo donde serán introducidos el usuario, el password y la url de la base de datos para la conexión con la misma.

- **Clase Entity**

- En esta clase se crea la Entity, una clase abstracta que representa una tabla de la base de datos. En ella, se genera un id aleatorio usando el paquete java.util.UUID, así como el getter para devolverlo y el setter para establecerlo. El método validate se encarga de validar dicho id pasándolo a String, y de lo contrario lanza una excepción si hay un fallo. El método equals comprueba si la instancia de Entity es falsa, y después devuelve el id del getter igualándolo a la instancia de Entity llamada tmpEntity. Y por último el método hashCode devuelve un entero al id para la base de datos.

- **Interfaz EntityManager**

- En esta interfaz se definen 4 métodos: el primero es un método save en el que se guardan las transacciones. Los otros tres métodos son genéricos, perteneciendo los dos primeros a la misma interfaz. AddStatement sirve para añadir un statement a una lista que será ejecutada para la base de datos(runables) y con addRangeStatement se le pasa varios statement en lugar de uno. El último es un ResultSet, que es un conjunto de resultados de la sentencia SQL, como ya se explicó anteriormente.

- **Implementación de la interfaz EntityManager**

- Aquí se realiza la implementación de EntityManager, donde además de los cuatro métodos anteriormente mencionados, se crea una lista de runables para llamar a la transacción y definir la sentencia SQL, crea una nueva configuración que inicialmente está vacía para luego construirla en el método buildConnection y gracias al método select se obtiene la url, el password y el usuario ejecutándose la consulta deseada.

- **Clase Converter**

- En esta clase hay dos métodos: fromUUIDtoByteArray para convertir el id en bytes para guardarlo en la máquina virtual, y fromByteArrayToUUID para realizar la operación contraria, convertir el array de bytes en el id del ingrediente.

- **Clase App**

- Esta es la clase principal, ya que en ella reside el método main que ejecutará la aplicación. Primero se genera un UUID para el ingrediente y se le pasa, así como un nombre y un precio. Después se crea la conexión para insertar un ingrediente mediante buildConnection, ejecutando la query insertando el id, el nombre y el precio del ingrediente, para después guardarlo. Luego se crea la conexión para obtener los ingredientes guardados en la base de datos y así obtenerlos mediante una consulta.

