

## Entrega DAO

El patrón DAO (Data Access Object) propone separar la lógica de negocio de la lógica para acceder a datos, el DAO proporcionará los métodos necesarios para el CRUD. Por otra parte, la capa de negocio solo se preocupa por la lógica de negocio y de usar el DAO para interactuar con la fuente de datos.

En el proyecto la clase Ingredient representa un objeto con la lógica de negocio.

- **IConfiguration**: interface que contiene los métodos para obtener los datos para acceder a la BBDD.
- **Configuration**: clase que implementa la interface IConfiguration, cuyo constructor devuelve una nueva configuración con los parámetros de acceso a BBDD.
- **Entity**: clase abstracta de la que van a extender el resto de objetos de la lógica de negocio en la que está el parámetro id que es necesario para todos los objetos, así como sus métodos get y set. Además está el método validate que comprueba que la id sea válida.
- **IRunnable**: interface que contiene los métodos getSQL, para obtener la sentencia SQL que se le pasa al objeto y la función run.
- **Runnable**: clase que implementa la interface IRunnable. Tiene como parámetros un String donde almacena la sentencia SQL a ejecutar, un T entity que es un genérico donde se pasa el tipo de objeto a tratar y un statement de T donde se pasan los parámetros necesarios para ejecutar la sentencia SQL.
- **ResultSet**: interface que contiene el método run que recibe como parámetro un ResultSet y objeto genérico sobre el que vamos a hacer la ejecución.
- **Statement**: interface que contiene el método run que recibe como parámetro un PreparedStatement y objeto genérico sobre el que vamos a hacer la ejecución.
- **IEntityManager**: interface que contiene los métodos necesarios para las operaciones CRUD.
- **EntityManager**: clase que implementa la interface IEntityManager. Aquí están inyectados una lista de objetos runnable donde están almacenadas las operaciones a realizar y un objeto Configuración que tiene los parámetros de conexión. En esta clase también están las siguientes funciones:
  - o **buildConnection**: devuelve una nueva configuración con los parámetros de acceso a BBDD.
  - o **addStatement**: genera un objeto Runnable con una nueva operación a realizar y lo almacena en la lista.
  - o **addRangeStatement**: genera objetos Runnable en función de lo que recibe en el objeto iterable y los almacena en la lista.
  - o **save**: crea una nueva conexión a BBDD, una vez que está hecho esto, recorre la lista de objetos Runnable para ir realizando las operaciones correspondientes. Una vez completados todos hace un commit para guardar todos los datos de manera atómica. En caso de haber alguna excepción no pararía la ejecución y se hace un rollback. Para terminar, vacía la lista de Runnable.
  - o **select**: crea una nueva conexión a BBDD, una vez que está hecho esto, coge el primero objeto de la lista de Runnable y genera un ResultSet ejecutando el statement que tiene la consulta select. Luego recorre el ResultSet asignando a

entity una nueva instancia del objeto que se recibe como parámetro y llama a la función run de Resultset.