

Modelado de Datos

DESICIONES DE DISEÑO:

- Pensamos en que Categoria sea solamente un atributo de tipo VARCHAR pero luego viendo que podemos tener distintos tipos de categorias elegimos ir por la opcion donde vamos a tener una clase Categoria la cual tiene un nombre
- La prioridad de los pedidos es segun el id ya que al ser autoincremental representaría el orden de llegada.
- La composición de productos la resolvemos con la tabla **ProductoCompuesto**, que tiene FK producto_id (que hace referencia al "nombre" del producto en si msmo, por ejemplo "cajita feliz") y el componente_id (que hace referencia al producto que lo compone, por ejemplo "papas fritas", "nuggets", etc.)

SAHIRA AYLÉN ALVAREZ...

GRUPO 14

ENLACE AL
REPOSITORIO

SAHIRA AYLÉN ALVAREZ...

Persistencia de Datos (desde el modelo de clases hacia el modelo de datos)

TODAS LAS TABLAS CUENTAN CON LAS COLUMNAS:

- id
- estaActivo
- fechaDeAlta
- fechaDeBaja
- fechaUltimaModificacion

SAHIRA AYLÉN ALVAREZ...

DESICIONES DE DISEÑO:

- Decidimos utilizar la estrategia de mapeo de herencia "single table" para encaapsular los personajes y sus roles, ya que al tener pocos atributos en las clases hijas, es más performante tener todos los datos en una misma tabla (incluso con el costo de tener campos nulos). Debido al rapido acceso, se evitan los joins.
- Creamos 2 converters, uno para los LocalDate y otro para la interfáz de ElementoDefensor.
- Utilizamos la etiqueta @ElementCollection para persistir la lista de elementos defensores que tiene un personaje (ya que es una lista de objetos que implementan una misma interfaz), junto con el @Convert para almacenar la interfaz como String

SAHIRA AYLÉN ALVAREZ...

