

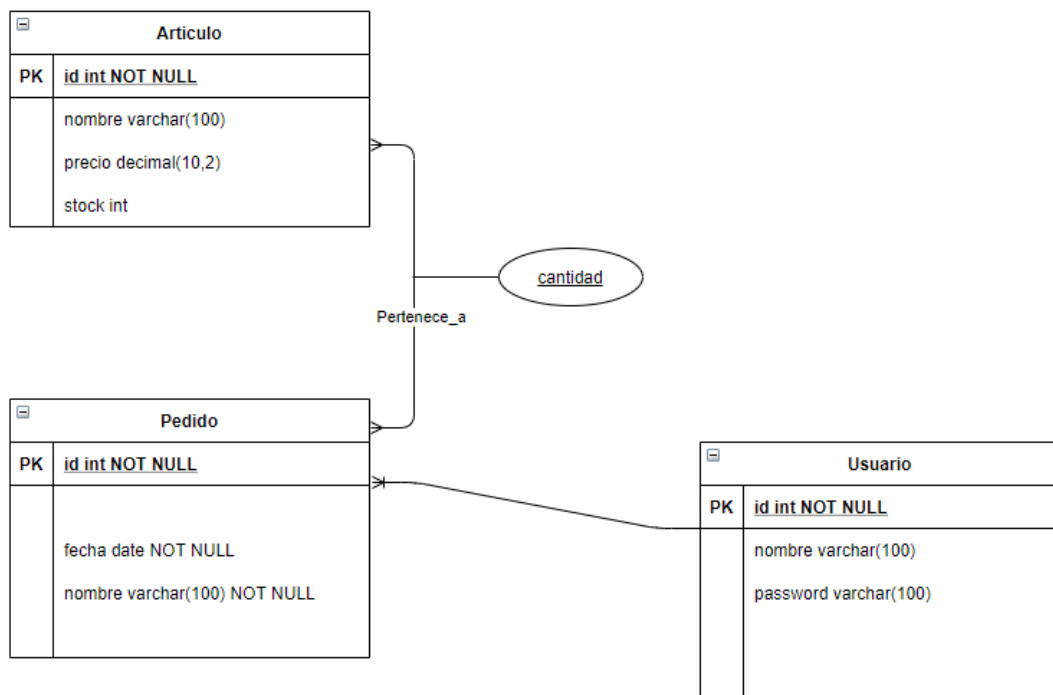
PROYECTO MARKETPLACE EOI

El proyecto va a consistir en desarrollar una aplicación Web con Angular 11 y un Backend en JAVA que funcione como un **MarketPlace** básico:

Un usuario podrá acceder al Marketplace con su user y pw, seleccionar artículos y generar un pedido con todos los artículos seleccionados.

Pese a que os facilitamos un modelo de entidades, si alguno quiere implementarlo con otro modelo como ya sabéis es elección vuestra, lo importante es que la salida del API REST se cómo indicamos en la parte de los DTO's.

1. **ENTIDADES** : Artículo, Usuario, Pedido y Pertenece_A (Artículo pertenece a Pedido, aquí se guardan todos los artículos que componen un pedido con sus cantidades)



2. Operaciones API a implementar

USUARIO

*POST*marketplace/usuario/login (Comprobar Login de usuario)

*POST*marketplace/usuario/login (Crear nuevo usuario)

*PUT*marketplace/usuario/:id (Actualizar un usuario)

*GET*marketplace/usuario (Obtener todos los usuarios)

ARTICULO

*POST*marketplace/articulo (Crear nuevo artículo)

*PUT*marketplace/articulo/:id (Actualizar un artículo)

*GET*marketplace/articulo/:id (Obtener un artículo por su id)

*GET*marketplace/articulo/:nombreParcial (Obtener todos los artículos que contengan en su atributo nombre, nombreParcial)

PEDIDO

*POST*marketplace/pedido (Crear nuevo pedido)

*PUT*marketplace/pedido/:id (Actualizar un pedido)

*GET*marketplace/pedido/:id (Obtener un pedido por su id)

*GET*marketplace/pedido/:nombreParcial (Obtener todos los pedidos que contengan en su atributo nombre, nombreParcial)

*DELETE*marketplace/pedido/:id (Eliminar un pedido por su id)

Recomendación / Propuesta Modelo DTO's

USUARIO

```
{
  "id": 1,
  "nombre": "Nombre",
  "password": "Password"
}
```

ARTICULO

```
{
  "id": 1,
  "nombre": "Nombre",
  "precio": 12,
  "stock": 1
}
```

PEDIDO

```
{
  "id": 1,
  "nombre": "Nombre",
  "fecha": "12/12/2020",
  "articulos": [
    {
      "id": 1,
      "cantidad": 12
    },
    {
      "id": 2,
      "cantidad": 20
    }
  ]
}
```

Consideraciones:

BBDD: Podéis elegir la BBDD que queráis pero recomendamos una BBDD en memoria H2 que facilite su despliegue y prueba. Para los datos iniciales utilizar un archivo que se llame data.sql y meter las inserts necesarias.

Seguridad:

Crearemos una autenticación en memoria para un usuario genérico llamado USER que tendrá el ROLE_USER y otro usuario de TIPO ADMIN con su contraseña y que tendrá el ROLE_ADMIN.

Todas las operaciones de delete serán únicamente permitidas a un usuario de tipo Admin para evitar perder información.

Para ello utilizar las @Secure de Spring Security.

Testing:

Validaremos la cobertura de testing unitario y puntuaremos según cobertura. A mayor cobertura de testing mayor nota.

OPTATIVO:

Para las pruebas de integración, si alguien quiere hacerlo con RestTemplate, lo explicaremos en clase y podremos ampliar la cobertura.

Entregable: Proyecto BASE: <https://gitlab.com/jjimerod/marketplace.git>

Los que no tengáis cuenta en GITLAB tendréis que crearla, y posteriormente crearemos una rama para cada uno de vosotros y cuyo código utilizaremos mas adelante para la integración con el front

