

## EL TACÓN DE ORO

**1. Extracción de los sustantivos en la descripción del problema.** EN ESTA PARTE PROCEDO A LEER VARIAS VECES DETALLADAMENTE LA SITUACIÓN EXPUESTA INTENTANDO EXTRAER LO QUE ENTIENDO PARECIDO A LAS ENTIDADES DE UN DIAGRAMA ENTIDAD RELACIÓN

usuarios, web, artículos, zapatos, bolsos, complementos, tienda, campañas, temporada, año, fabricación, firmas, proveedores, socios, pedidos, pago, tarjeta, entidad bancaria, administrativo tienda, empleado Almacén, paquete, empresa de transportes, rutas

PARA MI ESTOS SON LOS SUSTANTIVOS, NOMBRES , ENTIDADES IMPORTANTES, EL RESTO SERIAN ATRIBUTOS O PARTES DE LOS MISMOS , MÉTODOS O COMPORTAMIENTOS Y TIPOS DE RELACIONES ENTRE ESTOS QUE AHORA IREMOS ESPECIFICANDO.

**2. Selección de sustantivos como objetos/clases del sistema.** VOY A AGRUPAR LAS CLASES Y OBJETOS DE LA FORMA QUE ME GUSTARÍA QUE FIGURARAN EN EL POSTERIOR DIAGRAMA. Y LES AÑADIRÉ TAMBIÉN LOS ATRIBUTOS DEL PUNTO

**Obtención de los atributos de los objetos.**

**Clase "Pedido":** es la Superclase o clase padre que relaciona todas las demás

**- Atributos:**

- Fecha de pedido
- Estado del pedido

**1. Superclase "Artículos":**

**- Atributos:**

- Nombre
- Descripción
- Material
- Color
- Precio
- Stock

**- Clases hijas:**

**- Zapatos- Atributos adicionales:** Número, Tipo

**- Bolsos- Atributos adicionales:** Tipo (bandolera, mochila, fiesta, etc.),

**- Complementos - Atributos adicionales:** Tipo (cinturones, guantes), Talla (solo para cinturones y guantes)

**2. Superclase "Tiempo":**

- **Atributos:**
- Nombre (para Año, Temporada y Campaña)
- **Clases hijas:**

- **Año**
- **Temporada**
- **Campaña**

### 3. Superclase "Personas":

- **Atributos:**
- Nombre completo
- Correo electrónico
- Dirección
- **Clases hijas:**

- **Usuario**
- **Trabajador**
- **Proveedor**
- **Transportista**

### 4. Clase "Pago":

- **Atributos:**
- **Tipo de pago**
- **Entidad bancaria**

### 5 Clase "Paquete":

- **Atributos:**
- **Estado del paquete**
- **Fecha de envío**

### 6. Clase "Ruta":

- **Atributos:**
- **Destino**
- **Días de entrega**

### **Obtención de los métodos.**

1. Clase Artículo - obtenerNombre() , obtenerDescripción() , obtenerMaterial() , obtenerColor() , obtenerPrecio() , obtenerStock() , actualizarStock(int cantidad)
2. Clase Zapatos – obtenerNúmero() , obtenerTipo()
3. Clase Bolsos - obtenerTipo()
4. Clase Complementos – obtenerTipo() , obtenerTalla()
5. Clase Temporada - obtenerNombre()
6. Clase Campaña – obtenerNombre() , obtenerFechaInicio() , obtenerFechaFin()
7. Clase Firma – obtenerNombre() , obtenerCIF() , obtenerDomicilioFiscal()
8. Clase Proveedor – obtenerNombre() , obtenerCIF() , obtenerDomicilioFiscal()
9. Clase Usuario – obtenerNombreCompleto() , obtenerCorreoElectrónico() , obtenerDirección()
10. Clase Socio - hacerPedido(List<DetallePedido> detalles) , cancelarPedido(Pedido pedido) , modificarDatosPersonales(String nombreCompleto, String correoElectrónico, String dirección)
11. Clase Pedido – obtenerFechaRealización() , obtenerTotalPagar() , obtenerEstado() , realizarPago(Tarjeta tarjeta) , verificarValidezTarjeta(Tarjeta tarjeta)
12. Clase Empresa de transportes – obtenerNombre() , obtenerCIF() , obtenerDomicilioFiscal()
13. Clase Ruta – obtenerÁreaInfluencia() , obtenerDíasRepartoAsignados() , repartirPedido(Pedido pedido, Fecha fechaReparto) , registrarIncidencia(Fecha fecha, String descripción)

### **Obtención de las relaciones.**

Basándome en el esquema realizado, las relaciones entre las clases que contemplo :

#### **1. Pedido - Artículo:**

- Relación: Un pedido puede contener varios artículos y un artículo puede estar presente en varios pedidos.

- Tipo de relación: Asociación muchos a muchos.

#### **2. Pedido - Usuario:**

- Relación: Un pedido está asociado a un usuario que lo ha realizado.

- Tipo de relación: Asociación uno a uno o muchos a uno

### **3. Pedido - Pago:**

- Relación: Un pedido está asociado a un pago cuando se realiza la transacción.
- Tipo de relación: Asociación uno a uno.

### **4. Pedido - Paquete:**

- Relación: Un pedido está asociado a un paquete que contiene los artículos pedidos.
- Tipo de relación: Asociación uno a uno.

### **5. Artículo - Campaña:**

- Relación: Un artículo puede ser parte de una campaña.
- Tipo de relación: Asociación uno a muchos.

### **6. Artículo - Temporada:**

- Relación: Un artículo puede pertenecer a una temporada específica.
- Tipo de relación: Asociación muchos a uno.

### **7. Artículo - Firma:**

- Relación: Un artículo puede ser de una firma específica.
- Tipo de relación: Asociación uno a uno.

### **8. Artículo - Proveedor:**

- Relación: Un artículo puede ser provisto por uno o varios proveedores.
- Tipo de relación: Asociación muchos a muchos.

### **9. Usuario - Socio:**

- Relación: Un usuario puede ser socio del negocio.
- Tipo de relación: Herencia (subclase).

### **10. Usuario - Trabajador:**

- Relación: Un usuario puede ser un trabajador del negocio.
- Tipo de relación: Herencia (subclase).

## **11. Usuario - Proveedor:**

- **Relación:** Un usuario puede ser un proveedor.
- **Tipo de relación:** Herencia (subclase).

## **12. Paquete - Ruta:**

- **Relación:** Un paquete está asociado a una ruta de entrega.
- **Tipo de relación:** Asociación uno a uno o muchos a uno

### **Añadir getters, setters y constructores.**

**He añadido constructores en visual paradigm , pero getters y setters lo haré desde netbeans**

### **Primer refinamiento.**

En el refinamiento lo pensé mejor y reestructuré las clases quitando pedido de ser una mas para convertirla en el núcleo principal que une todo. Se hace un PEDIDO de uno o mas ARTÍCULOS, por parte de un CLIENTE, que hace un PAGO del pedido por un medio, dicho pedido moviliza a un TRABAJADOR administrativo, otro de almacén y un transportista que lleva el paquete o paquetes del pedido por una ruta. Dicho pedido sino es fabricado es de FIRMA y se prepara a través de un PROVEEDOR , teniendo en cuenta que el pedido sea de TEMPORADA , y este o no en CAMPAÑA, una de las varias que hay en el AÑO .

**COMO CONCLUSIÓN DEL DOCUMENTO , ME PUEDO HABER DEJADO ALGUNA COSA POR IMPLEMENTAR O EN EL TINTERO, SE ME HA HECHO MUY EXTENSO Y COMPLICADO PARA REFLEJAR TODO LO QUE QUERÍA , SOBRE TODO SI QUERÍA HACERLO DE UNA MANERA ENTENDIBLE SIMPLIFICADA CON UN PROGRAMA NUEVO PARA MI (LA CONEXIÓN CON NETBEANS DE VP NO QUEDA MUY CLARIFICADA... TUVE QUE BUSCARLO POR FUERA ) POR ELLO PENSÉ QUE LA MANERA MAS SIMPLE FINALMENTE. MAS QUE UNA AMALGAMA DE INTERRELACIONES Y CLASES CON MUCHOS ATRIBUTOS PARECIDAS ERA UN PEDIDO QUE CONTIENE VARIAS CLASES DE LAS QUE CADA UNA DE LAS RESTANTES HEREDAN A SU VEZ TODOS ESOS ATRIBUTOS Y MÉTODOS QUE EN VEZ DE SER REPETIDOS SON HEREDADOS Y RESCRITOS ..... AL FINAL HERENCIA Y POLIMORFISMO QUE AL FIN Y AL CABO SON IMPRESCINDIBLES EN PROGRAMACIÓN.**

**GRACIAS, ME HA PARECIDO UNA PRACTICA MAS COMPLEJA DE LO QUE A SIMPLE VISTA PUEDA PARECER, PERO SÚPER ÚTIL Y NECESARIA DE COMO PASAR DEL PROBLEMA AL PROGRAMA.**

**FINALMENTE COMO NO PODÍA PASARLO A CÓDIGO EN NETBEANS LOS GETTERS Y SETTERS LOS HICE CON VISUAL PARADIGM**