Proyecto final

1º Evaluación – 2º ASIR

ASGBD

Santiago Gutiérrez Romero

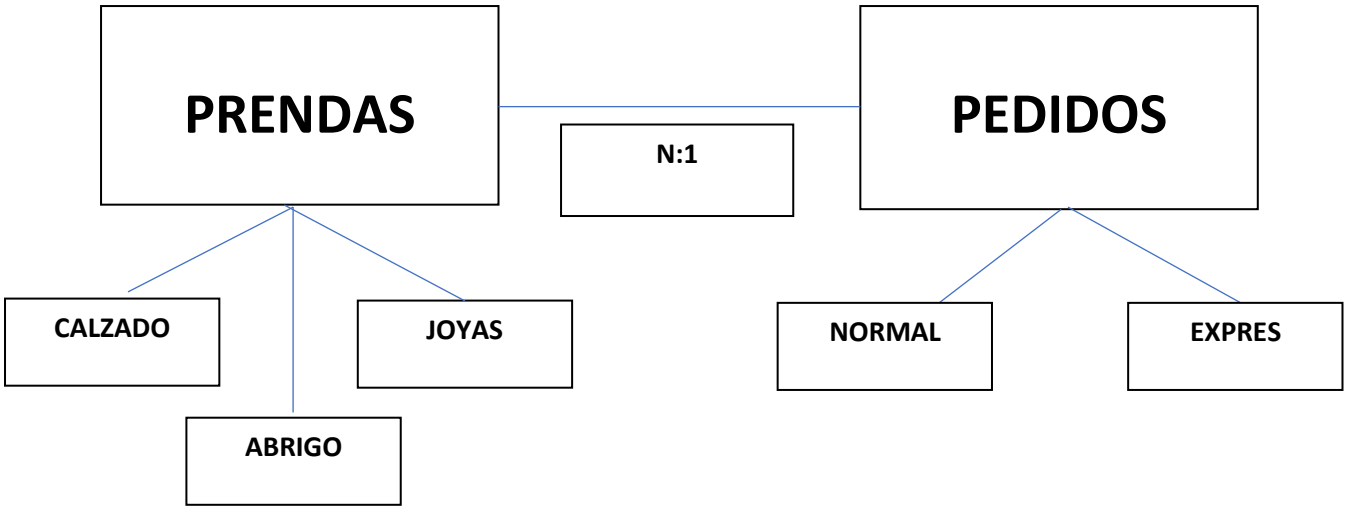
Tabla de contenido

Introducción:

El siguiente proyecto es una en una aplicación tipo CRUD basada en un gestor de pedidos de una tienda de ropa cualquiera. En este caso hemos escogido un tópico que se ajusta a la vida real ya que son muchas las tiendas las que cuentan con un servicio de envío de prendas, además, es vital para un negocio un posterior análisis económico de dichas ventas.

El proyecto realizado cuenta con una jerarquía de dos clases conectadas por una relación, dichas clases y relación serían las siguientes

.



Como se puede apreciar tenemos una super clase llamada Prendas desde la que parten tres subclases llamadas calzado, abrigo y joyas, cada una de estas subclases cuenta con campos exclusivos, en el caso de los pedidos ocurriría lo mismo solo que en este caso partirían las subclases pedido Normal y pedido Express. La lógica de la relación sería que un pedido lleva muchas prendas.

Para la puesta en marcha de este proyecto hemos hecho uso de el lenguaje de programación typescript, el entorno en tiempo de ejecución multiplataforma NODE.JS, MONGO atlas y Mongoose además de una serie de módulos necesarios para la programación del proyecto.

Clases y subclases:

A continuación, se mostrarán capturas de cada clase y subclase y se comentarán los aspectos más relevantes:

Super clase PRENDA:

```
export class Prenda {
    protected _id: number;
    protected _precioXmayor: number;
    protected _precioPublico: number;
    private _fechaCompra: Date;
    protected _material: string;
    private _paisFabric: string;
    private _pedi: number;

    constructor(
        id: number,
        precioXmayor: number,
        precioPublico: number,
        fechaCompra:Date,
        material:string,
        paisFabric:string,
        pedi:number) {

        this._id = id;
        this._precioXmayor = precioXmayor;
        this._precioPublico = precioPublico;
        this._fechaCompra = fechaCompra;
        this._material = material;
        this._paisFabric = paisFabric;
        this._pedi = pedi;
    }

    // Metodos GET
get id() {
    return this._id;
}
get precioXmayor() {
    return this._precioXmayor;
}
get precioPublico() {
    return this._precioPublico;
}
get fechaCompra() {
    return this._fechaCompra;
```

Aquí podemos ver todos los campos que heredarán las subclases, entre los cuales es de especial importancia el campo _pedi, este es el encargado de unir las dos jerarquías de clases.

Subclase ABRIGO, extiende de PRENDA

```
import { Prenda } from './prenda';

export class Abrigo extends Prenda {
    private _manga: string;
    private _cremallera: boolean;
    private _cuello: string;
    constructor(
        id: number,
        precioXmayor: number,
        precioPublico: number,
        fechaCompra:Date,
        material:string,
        paisFabric:string,
        pedi:number,
        manga:string,
        cremallera:boolean,
        cuello:string) {

            super(id, precioXmayor, precioPublico, fechaCompra, material, paisFabric, pedi)
        this._manga = manga;
        this._cremallera = cremallera;
        this._cuello =cuello;
    }
    //Metodos GET
    get manga(){
        return this._manga;
      }
    get cremallera(){
       return this._cremallera;
      }
    get cuello(){
        return this._cuello;
      }

//Sobreescitura de los calculos propios de la subclase


precioXmayorPrenda(): number {
    let xMayorPrenda: number
    let materiaX: string
    materiaX = this._material
    let cremalleraX: boolean
    cremalleraX = this._cremallera
    xMayorPrenda = super.precioXmayorPrenda()

    if (materiaX == "Algodon"){
        xMayorPrenda= xMayorPrenda+1
        if (cremalleraX == true){
            xMayorPrenda= xMayorPrenda+3
```

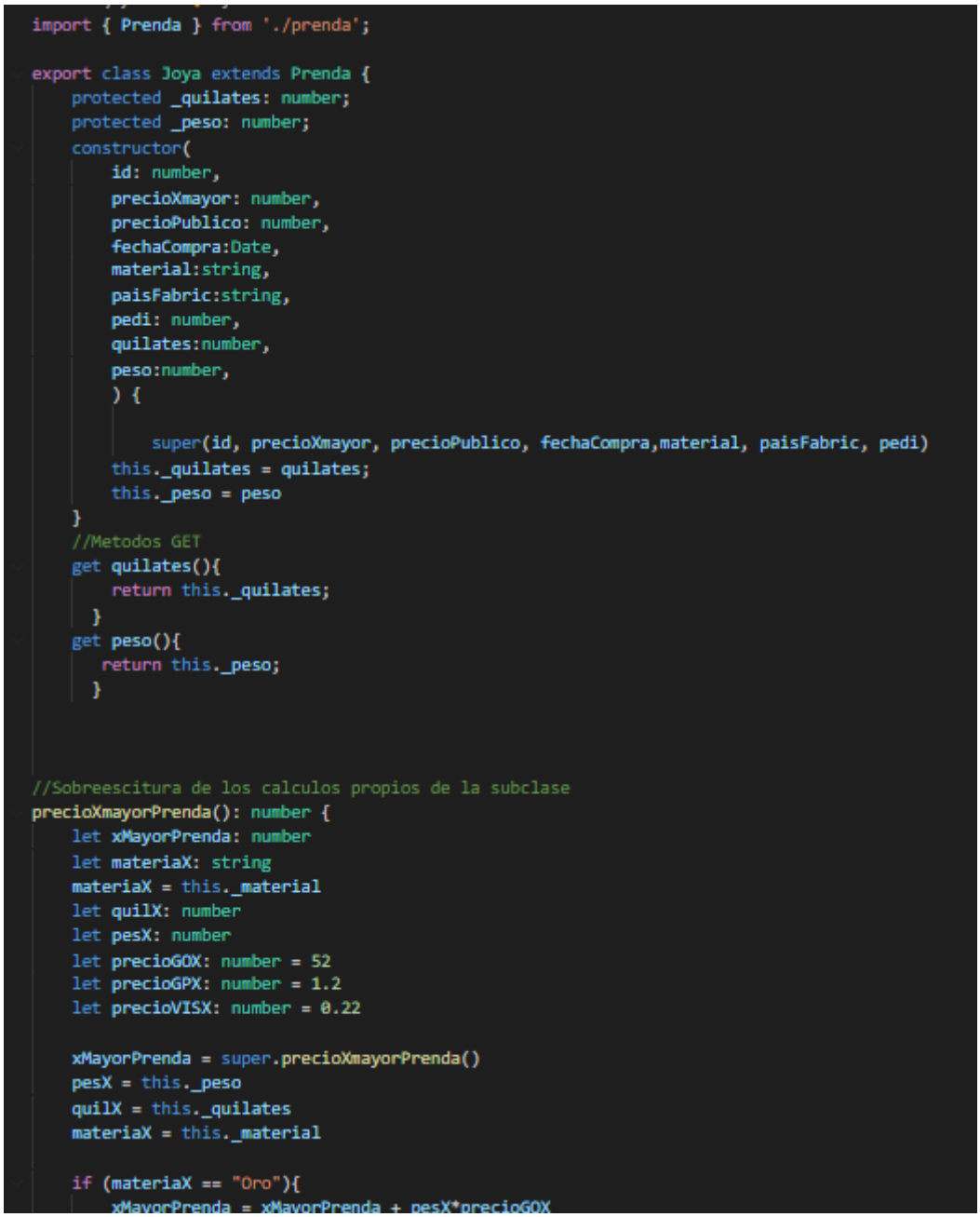
En este caso podemos ver los campos exclusivos de los abrigos.

Subclase Calzado extiende de PRENDAS:

```
import { Prenda } from './prenda';

export class Calzado extends Prenda {
    protected _suela: string;
    protected _unidadesEnmercado: number;
    protected _calidad: string;
    constructor(
        id: number,
        precioXmayor: number,
        precioPublico: number,
        fechaCompra:Date,
        material:string,
        paisFabric:string,
        pedi:number,
        suela:string,
        unidadesEnmercado:number,
        calidad:string) {

            super(id, precioXmayor, precioPublico, fechaCompra, material, paisFabric, pedi)
        this._suela = suela;
        this._unidadesEnmercado = unidadesEnmercado;
        this._calidad = calidad;
    }

    //Metodos GET
    get suela(){
        return this._suela;
      }
    get unidadesEnmercado(){
       return this._unidadesEnmercado;
      }
    get calidad(){
        return this._calidad;
      }

//Sobreescitura de los calculos propios de la subclase
precioXmayorPrenda(): number {
    let xMayorPrenda: number
    let materiaX: string
    materiaX = this._material
    let unidadesX: number
    unidadesX = this._unidadesEnmercado
    let calidadX: string
    calidadX = this._calidad
    let suelaX = this._suela
    let preciogomaX: number = 2
    let preciosinteticaX: number = 1
    xMayorPrenda = super.precioXmayorPrenda()
```

Con campos específicos de su clase.

Clase Joyas extiende de PRENDAS:



```
import { Prenda } from './prenda';

export class Joya extends Prenda {
    protected _quilates: number;
    protected _peso: number;
    constructor(
        id: number,
        precioXmayor: number,
        precioPublico: number,
        fechaCompra:Date,
        material:string,
        paisFabric:string,
        pedi: number,
        quilates:number,
        peso:number,
        ) {

            super(id, precioXmayor, precioPublico, fechaCompra,material, paisFabric, pedi)
        this._quilates = quilates;
        this._peso = peso
    }
    //Metodos GET
    get quilates(){
        return this._quilates;
      }
    get peso(){
       return this._peso;
      }



//Sobreescitura de los calculos propios de la subclase
precioXmayorPrenda(): number {
    let xMayorPrenda: number
    let materiaX: string
    materiaX = this._material
    let quilX: number
    let pesX: number
    let precioGOX: number = 52
    let precioGPX: number = 1.2
    let precioVISX: number = 0.22

    xMayorPrenda = super.precioXmayorPrenda()
    pesX = this._peso
    quilX = this._quilates
    materiaX = this._material

    if (materiaX == "Oro"){
        xMayorPrenda = xMayorPrenda + pesX*precioGOX
```

Campos específicos para hacer cálculos.

Super clase PEDIDO:

```typescript
import { Prenda } from './prenda';

export class Pedido {
    protected _id: number;
    protected _precioBase: number;
    protected _diasAprox: number;
    private _compañia: string;
    protected _fechaEnvio: Date;
    private _paisSalida: string;
    protected _estado: boolean
    private _arrayPrendas: Array<Prenda>

    constructor(
        id: number,
        precioBase: number,
        diasAprox: number,
        compañia: string,
        fechaEnvio:Date,
        paisSalida:string,
        estado: boolean ) {

        this._id = id;
        this._precioBase = precioBase;
        this._diasAprox = diasAprox;
        this._compañia = compañia;
        this._fechaEnvio = fechaEnvio;
        this._paisSalida = paisSalida;
        this._estado = estado;
        this._arrayPrendas = new Array<Prenda>()
    }

    // Metodos GET
get id() {
    return this._id;
}
get precioBase() {
    return this._precioBase;
}
get diasAprox() {
    return this._diasAprox;
}
get compañia() {
    return this._compañia;
}
get fechaEnvio() {
    return this._fechaEnvio;
}
get paisSalida() {
    return this._paisSalida
```

Subclase NORMAL extiende de Pedido:

SRC > clases > pedidosN.ts > Normal > costoPedido

```ts
import { Pedido } from './pedido';
import { Prenda } from './prenda';

export class Normal extends Pedido {
    protected _incremento: number;
    protected _impuesto: number;

    constructor(
        id: number,
        precioBase: number,
        diasAprox: number,
        compañia: string,
        fechaEnvio: Date,
        paisSalida: string,
        estado: boolean,
        incremento:number,
        impuesto:number) {

        super(id, precioBase, diasAprox, compañia, fechaEnvio, paisSalida, estado)

        this._incremento = incremento;
        this._impuesto = impuesto;
    }
    //Metodos GET
    get incremento() {
        return this._incremento;
    }
    get impuesto() {
        return this._impuesto;
    }

    //Sobreescitura de los calculos propios de la subclase

    costoPedido(): number {
        let costo: number
        let imp: number
        let incre: number

        imp = this._impuesto
        incre = this._incremento
        costo = super.costoPedido()
        costo = costo + imp + incre
        return costo
    }


}
```

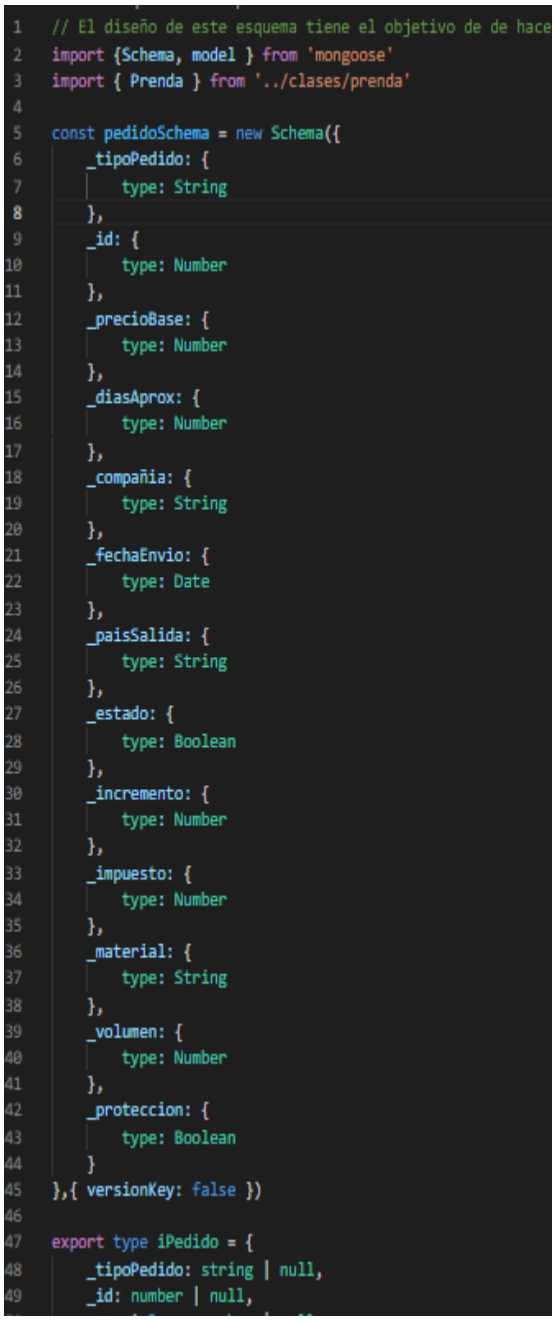
Sub clase Express extiende de pedido:

```
import { Pedido } from './pedido';
import { Prenda } from './prenda';

export class Expres extends Pedido {
    protected _material: string;
    protected _volumen: number;
    protected _proteccion: boolean;

    constructor(
        id: number,
        precioBase: number,
        diasAprox: number,
        compañia: string,
        fechaEnvio: Date,
        paisSalida: string,
        estado: boolean,
        material:string,
        volumen: number,
        proteccion: boolean,
        ) {

        super(id, precioBase, diasAprox, compañia, fechaEnvio, paisSalida, estado)

        this._material = material;
        this._volumen = volumen;
        this._proteccion = proteccion;
    }
    //Metodos GET
    get material() {
        return this._material;
    }
    get volumen() {
        return this._volumen;
    }
    get proteccion() {
        return this._proteccion;
    }

    //Sobreescitura de los calculos propios de la subclase

    costoPedido(): number {
        let costo: number
        let mat: string
        let vol: number
        let protec: boolean

        mat = this._material
        vol = this._volumen
        protec = this._proteccion
```

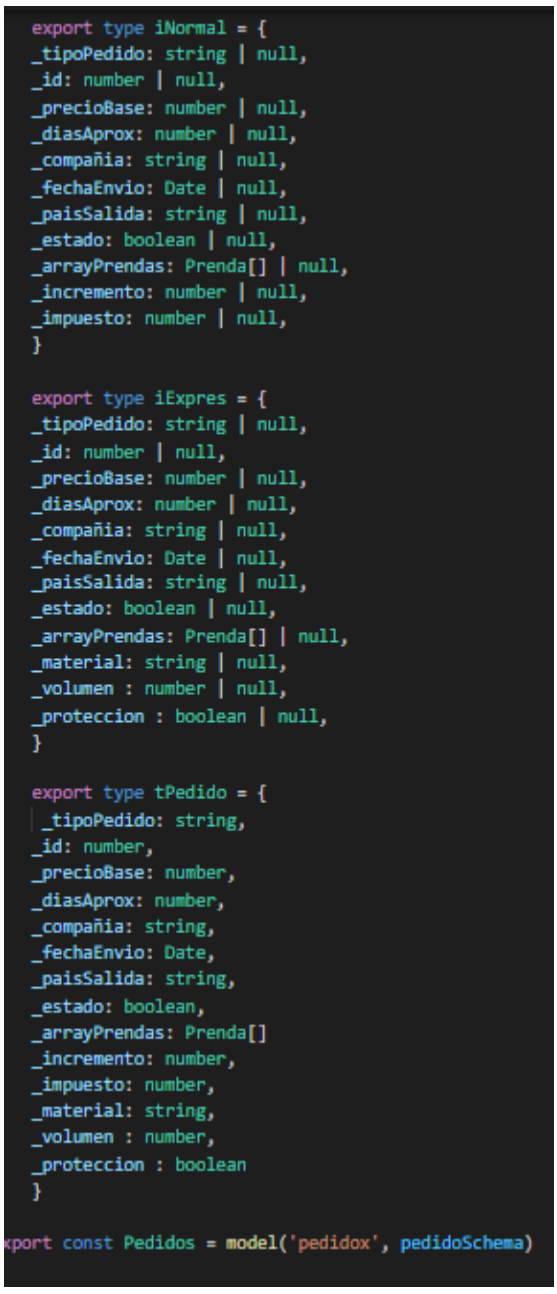
Schemas:

A continuación, se mostrarán los Schemas de nuestro proyecto 2 de ellos referentes a nuestras jerarquías de clases y otros dos que han sido creados con el fin de darle persistencia a objetos determinados:

Schema Pedidos:



```
// El diseño de este esquema tiene el objetivo de de hace
import {Schema, model } from 'mongoose'
import { Prenda } from '../clases/prenda'

const pedidoSchema = new Schema({
    _tipoPedido: {
        type: String
    },
    _id: {
        type: Number
    },
    _precioBase: {
        type: Number
    },
    _diasAprox: {
        type: Number
    },
    _compañia: {
        type: String
    },
    _fechaEnvio: {
        type: Date
    },
    _paisSalida: {
        type: String
    },
    _estado: {
        type: Boolean
    },
    _incremento: {
        type: Number
    },
    _impuesto: {
        type: Number
    },
    _material: {
        type: String
    },
    _volumen: {
        type: Number
    },
    _proteccion: {
        type: Boolean
    }
},{ versionKey: false })

export type iPedido = {
    _tipoPedido: string | null,
    _id: number | null,
```



```
export type iNormal = {
_tipoPedido: string | null,
_id: number | null,
_precioBase: number | null,
_diasAprox: number | null,
_compañia: string | null,
_fechaEnvio: Date | null,
_paisSalida: string | null,
_estado: boolean | null,
_arrayPrendas: Prenda[] | null,
_incremento: number | null,
_impuesto: number | null,
}

export type iExpres = {
_tipoPedido: string | null,
_id: number | null,
_precioBase: number | null,
_diasAprox: number | null,
_compañia: string | null,
_fechaEnvio: Date | null,
_paisSalida: string | null,
_estado: boolean | null,
_arrayPrendas: Prenda[] | null,
_material: string | null,
_volumen : number | null,
_proteccion : boolean | null,
}

export type tPedido = {
_tipoPedido: string,
_id: number,
_precioBase: number,
_diasAprox: number,
_compañia: string,
_fechaEnvio: Date,
_paisSalida: string,
_estado: boolean,
_arrayPrendas: Prenda[]
_incremento: number,
_impuesto: number,
_material: string,
_volumen : number,
_proteccion : boolean
}

xport const Pedidos = model('pedidox', pedidoSchema)
```

Schema Prendas:

```
import {Schema, model } from 'mongoose'

const prendaSchema = new Schema({
    _tipoPrenda: {
        type: String
    },
    _id: {
        type: Number
    },
    _precioXmayor: {
        type: Number
    },
    _precioPublico: {
        type: Number
    },
    _fechaCompra: {
        type: Date
    },
    _material: {
        type: String
    },
    _paisFabric: {
        type: String
    },
    _pedi: {
        type: Number
    },
    _manga: {
        type: String
    },
    _cremallera: {
        type: Boolean
    },
    _cuello: {
        type: String
    },
    _suela: {
        type: String
    },
    _unidadesEnmercado: {
        type: Number
    },
    _calidad: {
        type: String
    },
    _quilates: {
        type: Number
    },
```

```
_precioPublico: number | null,
_fechaCompra: Date | null,
_material: string | null,
_paisFabric: string | null,
_pedi: number | null,
}

export type iAbrigo = {
    _tipoPrenda: string | null,
    _id: number | null,
    _precioXmayor: number | null,
    _precioPublico: number | null,
    _fechaCompra: Date | null,
    _material: string | null,
    _paisFabric: string | null,
    _pedi: number | null,
    _manga: string | null,
    _cremallera: boolean | null,
    _cuello: string | null,
}

export type iCalzado = {
    _tipoPrenda: string | null,
    _id: number | null,
    _precioXmayor: number | null,
    _precioPublico: number | null,
    _fechaCompra: Date | null,
    _material: string | null,
    _paisFabric: string | null,
    _pedi: number | null,
    _suela: string | null,
    _unidadesEnmercado: number | null,
    _calidad: string | null,
}
export type iJoya = {
    _tipoPrenda: string | null,
    _id: number | null,
    _precioXmayor: number | null,
    _precioPublico: number | null,
    _fechaCompra: Date | null,
    _material: string | null,
    _paisFabric: string | null,
    _pedi: number | null,
    _quilates: number | null,
    _peso: number | null,
}

export const Prendas = model('prendax', prendaSchema)
```

Schema enviados:

```
import {Schema, model } from 'mongoose'

// Definimos el Schema
const enviadoSchema = new Schema({
    _id: {
        type: Number
    },
    _compañia: {
        type: String
    },
    _paisSalida: {
        type: String
    },
    _fechaEntrega: {
        type: Date
    }
},{ versionKey: false })


export type iEnv = {
    _id: Number | null,
    _compañia: String | null,
    _paisSalida: String | null,
    _fechaEntrega: Date | null
  }

// La colección de la BD (Plural siempre)
export const Enviados = model('enviados', enviadoSchema)
```

Schema rendimientos:

```
// El diseño de este esquema tiene el objetivo de de hacer como base par
import {Schema, model } from 'mongoose'

// Definimos el Schema
const rendimientoSchema = new Schema({
    _id: {
        type: Number
    },
    _precioCompra: {
        type: Number
    },
    _precioVenta: {
        type: Number
    },
    _rendimiento: {
        type: Number
    }
},{ versionKey: false })


export type iRend = {
    _id: Number | null,
    _precioCompra: Number | null,
    _precioVenta: Number | null,
    _rendimiento: Number | null,
  }

// La colección de la BD (Plural siempre)
export const Rendimientos = model('rendimientos', rendimientoSchema)
```

Demostración de uso de la aplicación

1.1. Creación de objetos y persistencia en BD (Opción 1 - 7):

Creación de objeto abrigo:

```
1.- Crear nuevo abrigo en memoria
2.- Crear nuevo calzado en memoria
3.- Crear nueva joya en memoria
4.- Crear nuevo pedido normal en memoria
5.- Crear nuevo pedido expres en memoria
6.- SALVAR PRENDAS EN BD
7.- SALVAR PEDIDOS EN BD
8.- Mostrar pedidos listos
9.- Hacer persistentes pedidos ya enviados
10.- Mostrar precio total de todo el pedido
11.- Cambiar una prenda de pedido
12.- Mostrar el beneficio de cada pedido y el beneficio total
13.- Borrar un pedido y todas sus prendas de la BD
14.- Hacer persistentes los mejores rendimientos
0.- Salir
opción: : 1
ID de la prenda: 5
Precio de compra al por mayor: 45
Precio de Venta en tienda: 70
Fecha de Compra del producto: 1
Material de fabricación: Algodon
Pais de fabricación del producto: España
Indica el ID del pedido al que pertenece esta prenda: 3
Tipo de manga del abrigo: Corto
Abrigo con cremallera:
Tipo de cuello del abrigo: Corto
```

Creación de objeto calzado:

```
1.- Crear nuevo abrigo en memoria
2.- Crear nuevo calzado en memoria
3.- Crear nueva joya en memoria
4.- Crear nuevo pedido normal en memoria
5.- Crear nuevo pedido expres en memoria
6.- SALVAR PRENDAS EN BD
7.- SALVAR PEDIDOS EN BD
8.- Mostrar pedidos listos
9.- Hacer persistentes pedidos ya enviados
10.- Mostrar precio total de todo el pedido
11.- Cambiar una prenda de pedido
12.- Mostrar el beneficio de cada pedido y el beneficio total
13.- Borrar un pedido y todas sus prendas de la BD
14.- Hacer persistentes los mejores rendimientos
0.- Salir
opción: : 2
ID de la prenda: 6
Precio de compra al por mayor: 56
Precio de Venta en tienda: 67
Fecha de Compra del producto: 4
Material de fabricación: Cuero
Pais de fabricación del producto: China
Indica el ID del pedido al que pertenece esta prenda: 3
Tipo de suela del calzado: No goma
Unidades existentes en mercado: 89
Calidad del calzado: Bajo
```

Creación de objeto Joya:

```
1.- Crear nuevo abrigo en memoria
2.- Crear nuevo calzado en memoria
3.- Crear nueva joya en memoria
4.- Crear nuevo pedido normal en memoria
5.- Crear nuevo pedido expres en memoria
6.- SALVAR PRENDAS EN BD
7.- SALVAR PEDIDOS EN BD
8.- Mostrar pedidos listos
9.- Hacer persistentes pedidos ya enviados
10.- Mostrar precio total de todo el pedido
11.- Cambiar una prenda de pedido
12.- Mostrar el beneficio de cada pedido y el beneficio total
13.- Borrar un pedido y todas sus prendas de la BD
14.- Hacer persistentes los mejores rendimientos
0.- Salir
opción: : 3
ID de la prenda: 7
Precio de compra al por mayor: 89
Precio de Venta en tienda: 120
Fecha de Compra del producto: 9
Material de fabricación: Oro
Pais de fabricación del producto: Argentina
Indica el ID del pedido al que pertenece esta prenda: 1
Indica los quilates de la Joya: 25
Indica el peso de la joya: 8
```

Persistencia a los pedidos:

```
_id: 4
_tipoPrenda: "Calzado"
_precioXmayor: 67
_precioPublico: 86
_fechaCompra: 2001-01-31T23:00:00.000+00:00
_material: "Cuero"
_paisFabric: "España"
_pedi: 2
_suela: "Goma"
_unidadesEnmercado: 23
_calidad: "Alto"
```

```
_id: 5
_tipoPrenda: "Abrigo"
_precioXmayor: 45
_precioPublico: 70
_fechaCompra: 2000-12-31T23:00:00.000+00:00
_material: "Algodon"
_paisFabric: "España"
_pedi: 3
_manga: "Corto"
_cremallera: false
_cuello: "Corto"
```

Creación de objeto pedido Normal:

```
1.- Crear nuevo abrigo en memoria
2.- Crear nuevo calzado en memoria
3.- Crear nueva joya en memoria
4.- Crear nuevo pedido normal en memoria
5.- Crear nuevo pedido expres en memoria
6.- SALVAR PRENDAS EN BD
7.- SALVAR PEDIDOS EN BD
8.- Mostrar pedidos listos
9.- Hacer persistentes pedidos ya enviados
10.- Mostrar precio total de todo el pedido
11.- Cambiar una prenda de pedido
12.- Mostrar el beneficio de cada pedido y el beneficio total
13.- Borrar un pedido y todas sus prendas de la BD
14.- Hacer persistentes los mejores rendimientos
0.- Salir
opción: : 4
Id del pedido: 3
Precio Base del pedido: 2
Dias que tarda en llegar aproximadamente: 6
Compañia de reparto: A
Fecha de Envio: 6
Pais desde el que sale el envio: Canada
Estado del producto:
Porcentaje de incremento al pedido: 2
Impuesto aplicado al pedido: 3
```

Creación de objeto pedido express:

```
1.- Crear nuevo abrigo en memoria
2.- Crear nuevo calzado en memoria
3.- Crear nueva joya en memoria
4.- Crear nuevo pedido normal en memoria
5.- Crear nuevo pedido expres en memoria
6.- SALVAR PRENDAS EN BD
7.- SALVAR PEDIDOS EN BD
8.- Mostrar pedidos listos
9.- Hacer persistentes pedidos ya enviados
10.- Mostrar precio total de todo el pedido
11.- Cambiar una prenda de pedido
12.- Mostrar el beneficio de cada pedido y el beneficio total
13.- Borrar un pedido y todas sus prendas de la BD
14.- Hacer persistentes los mejores rendimientos
0.- Salir
opción: : 5
Id del pedido: 4
Precio Base del pedido: 4
Dias que tarda en llegar aproximadamente: 3
Compañia de reparto: B
Fecha de Envio: 6
Pais desde el que sale el envio: España
Estado del producto: 1
Material del paquete: Carton
Volumen del paquete: 25
Proteccion del paquete: 1
```

Persistencia a los pedidos:


```
_id: 3
_tipoPedido: "Normal"
_precioBase: 2
_diasAprox: 6
_compañia: "A"
_fechaEnvio: 2001-05-31T22:00:00.000+00:00
_paisSalida: "Canada"
_estado: false
_incremento: 2
_impuesto: 3
```

```
_id: 4
_tipoPedido: "Expres"
_precioBase: 4
_diasAprox: 3
_compañia: "B"
_fechaEnvio: 2001-05-31T22:00:00.000+00:00
_paisSalida: "España"
_estado: true
_material: "Carton"
_volumen: 25
_proteccion: true
```

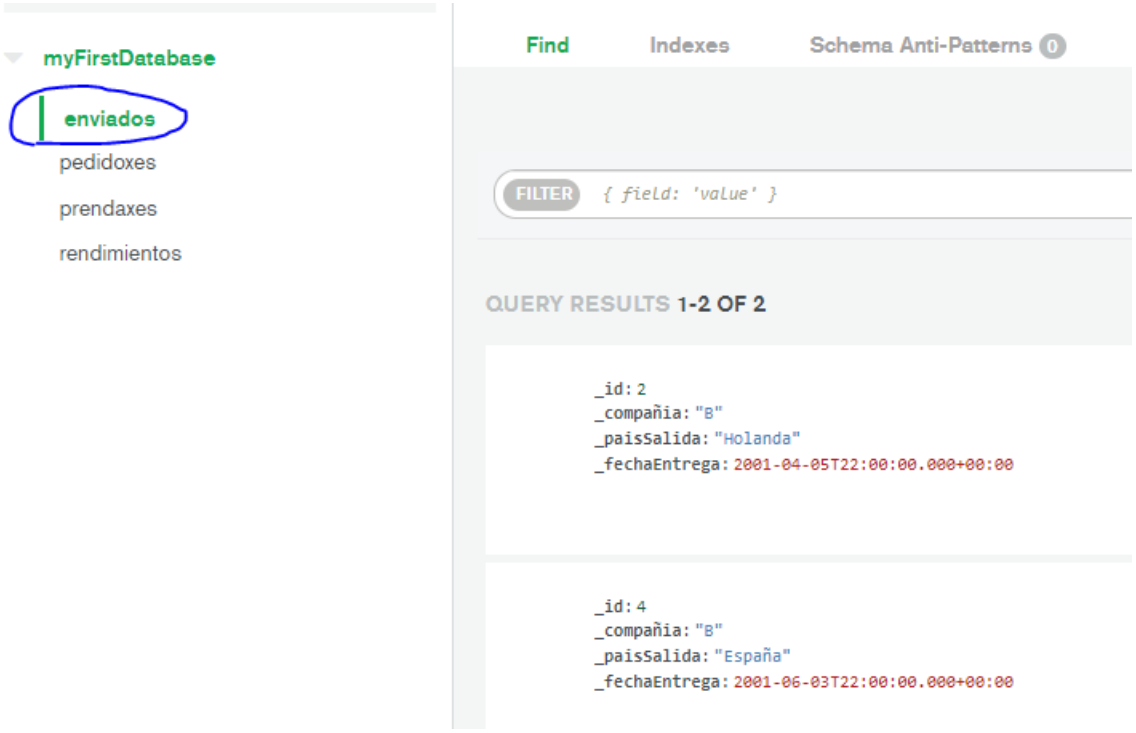
1.2. Asociación de pedidos con prendas (Opción 8):

```
Normal {
  _id: 1,
  _precioBase: 2,
  _diasAprox: 12,
  '_compañia': 'A',
  _fechaEnvio: 2001-04-30T22:00:00.000Z,
  _paisSalida: 'España',
  _estado: false,
  _arrayPrendas: [
    Abrigo {
      _id: 1,
      _precioXmayor: 12,
      _precioPublico: 23,
      _fechaCompra: 2000-12-31T23:00:00.000Z,
      _material: 'Cuero',
      _paisFabric: 'España',
      _pedi: 1,
      _manga: 'Corto',
      _cremallera: false,
      _cuello: 'Largo'
    },
    Calzado {
      _id: 2,
      _precioXmayor: 34,
      _precioPublico: 54,
      _fechaCompra: 2000-12-31T23:00:00.000Z,
      _material: 'Piel',
      _paisFabric: 'Italia',
      _pedi: 1,
      _suela: 'Goma',
      _unidadesEnmercado: 54,
      _calidad: 'Alto'
    },
    Joya {
      _id: 3,
      _precioXmayor: 78,
      _precioPublico: 100,
      _fechaCompra: 2001-03-31T22:00:00.000Z,
      _material: 'Plata',
      _paisFabric: '4',
      _pedi: 1,
      _quilates: 22,
      _peso: 4
    },
    Joya {
      _id: 7,
      _precioXmayor: 89,
      _precioPublico: 120,
      _fechaCompra: 2001-08-31T22:00:00.000Z,
```

```
Expres {
  _id: 2,
  _precioBase: 2,
  _diasAprox: 5,
  '_compañia': 'B',
  _fechaEnvio: 2001-03-31T22:00:00.000Z,
  _paisSalida: 'Holanda',
  _estado: true,
  _arrayPrendas: [
    Calzado {
      _id: 4,
      _precioXmayor: 67,
      _precioPublico: 86,
      _fechaCompra: 2001-01-31T23:00:00.000Z,
      _material: 'Cuero',
      _paisFabric: 'España',
      _pedi: 2,
      _suela: 'Goma',
      _unidadesEnmercado: 23,
      _calidad: 'Alto'
    }
  ],
  _material: 'Carton',
  _volumen: 15,
  _proteccion: true
}
```

```
Normal {
  _id: 3,
  _precioBase: 2,
  _diasAprox: 6,
  '_compañia': 'A',
  _fechaEnvio: 2001-05-31T22:00:00.000Z,
  _paisSalida: 'Canada',
  _estado: false,
  _arrayPrendas: [
    Abrigo {
      _id: 5,
      _precioXmayor: 45,
      _precioPublico: 70,
      _fechaCompra: 2000-12-31T23:00:00.000Z,
      _material: 'Algodon',
      _paisFabric: 'España',
      _pedi: 3,
      _manga: 'Corto',
      _cremallera: false,
      _cuello: 'Corto'
    },
    Calzado {
      _id: 6,
      _precioXmayor: 56,
      _precioPublico: 67,
      _fechaCompra: 2001-03-31T22:00:00.000Z,
      _material: 'Cuero',
      _paisFabric: 'China',
      _pedi: 3,
      _suela: 'No goma',
      _unidadesEnmercado: 89,
      _calidad: 'Bajo'
    }
  ],
  _incremento: 2,
  _impuesto: 3
}
```

1.3. Persistencia a los pedidos ya enviados (Opción 9):



1.4. Precio total de todos los pedidos (Opción 10)

En esta función se tiene en cuenta tanto el precio de la prenda como el precio del pedido ya que esta enfocada a que el cliente conozca el precio del pedido que hace.

```
0.- Salir
opción: : 10
El precio final del pedido con id: 1
es igual a: 923.4499999999999 €
El precio final del pedido con id: 2
es igual a: 285.07 €
El precio final del pedido con id: 3
es igual a: 182.445 €
El precio final del pedido con id: 4
es igual a: 276.21999999999997 €
```

1.5. Cambiar una prenda de pedido (Opción 11)

```
Indica el identificador del pedido donde meteras la prenda: 2
Indica el identificador de la prenda a modificar: 3
El pedido modificado quedara de la siguiente manera:
Joya {
  _id: 3,
  _precioXmayor: 78,
  _precioPublico: 100,
  _fechaCompra: 2001-03-31T22:00:00.000Z,
  _material: 'Plata',
  _paisFabric: '4',
  _pedi: 2,
  _quilates: 22,
  _peso: 4
}
Modificado Correctamente
```

Cambio en BD:

```
_id: 3
_tipoPrenda: "Joya"
_precioXmayor: 78
_precioPublico: 100
_fechaCompra: 2001-03-31T22:00:00.000+00:00
_material: "Plata"
_paisFabric: "4"
_pedi: 2
_quilates: 22
_peso: 4
```

1.6. Beneficio de cada pedido y beneficio total (Opción 12):

Calculamos el precio de venta al público, a ello le restamos el precio al por mayor y obtenemos el beneficio que obtiene la tienda

```
14.- Hacer persistentes los mejores rendimientos
0.- Salir
opción: : 12
El beneficio del pedido con id: 1
es igual a: 219.178 €
El beneficio del pedido con id: 2
es igual a: 236.9946 €
El beneficio del pedido con id: 3
es igual a: 68.84099999999998 €
El beneficio del pedido con id: 4
es igual a: 176.25319999999996 €
El beneficio total de todos los pedidos es: 701.2668 €
```

1.7. Borrar un pedido y todas sus prendas (Opción 13):

```
Indica el identificador del pedido que quieres borrar: 3
ATENCION: Todas las prendas asociadas a dicho pedido tambien se eliminaran
La prenda con id 5 se ha eliminado
Abrigo {
  _id: 5,
  _precioXmayor: 45,
  _precioPublico: 70,
  _fechaCompra: 2000-12-31T23:00:00.000Z,
  _material: 'Algodon',
  _paisFabric: 'España',
  _pedi: 3,
  _manga: 'Corto',
  _cremallera: false,
  _cuello: 'Corto'
}
La prenda con id 6 se ha eliminado
Calzado {
  _id: 6,
  _precioXmayor: 56,
  _precioPublico: 67,
  _fechaCompra: 2001-03-31T22:00:00.000Z,
  _material: 'Cuero',
  _paisFabric: 'China',
  _pedi: 3,
  _suela: 'No goma',
  _unidadesEnmercado: 89,
  _calidad: 'Bajo'
}
```

1.8. Hacer persistentes las prendas con mejor rendimiento (Opción 14)