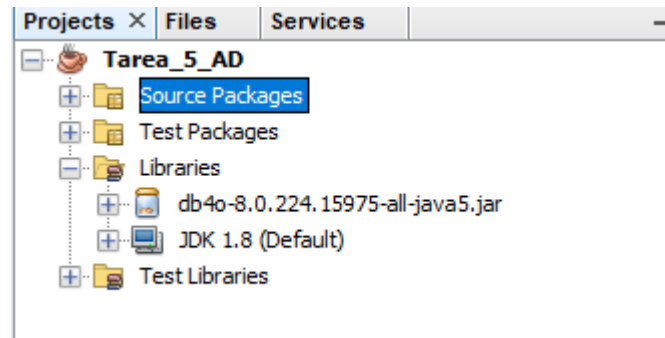


EJERCICIO 1

Se nos pide crear una base de datos llamada “DBJefeHijo” en DB4O e introducir unos datos en la misma para poder trabajar con ella.

Lo primero que haré será crear un nuevo proyecto para la tarea en NetBeans y dentro de la carpeta librerías voy a incorporar el paquete de la base de datos embebida “DB4o” que se nos ha facilitado a través del foro ya que la página oficial ya no proporciona la descarga por estar un tanto obsoleta:



Dentro del proyecto crearé varias clases que nos servirán para poder trabajar con la base de datos:

- **Clase Jefe:** Contendrá los atributos de jefe junto con los métodos. Las instancias que hagamos de esta clase junto con los datos contenidos serán los que vamos a almacenar en la base de datos:

```
public class Jefe {

    private String nombre;
    private int anioEmpresa;
    private int edad;
    private Hijo hijo;

    public Jefe(String nombre, int anioEmpresa, int edad, Hijo hijo) {
        this.nombre = nombre;
        this.anioEmpresa = anioEmpresa;
        this.edad = edad;
        this.hijo = hijo;
    }

    public Jefe() {
    }
}
```

- **Clase Hijo:** Contiene los atributos del hijo que serán incluidos como datos dentro de la clase jefe:

```
public class Hijo {

    String nombre;
    int edad;

    public Hijo(String nombre, int edad) {
        this.nombre = nombre;
        this.edad = edad;
    }

    public Hijo() {
    }
}
```

- **Clase *BDJefeHijo*:** Dentro de esta clase crearé un método para generar la conexión con la base de datos y el almacenamiento de los datos que propone la tarea:

```
public class BDJefeHijo {

    public ObjectContainer baseDatos;

    public BDJefeHijo() {

    }

    /**
     * Método que crea la base de datos y añade los objetos de la tarea
     */
    public void crearBBDD() {
        //Este código anterior lo ponemos por si la base de datos ya existiera y quisiéramos empezar desde el principio
        File fichero = new File("BDJefeHijo.db4o");
        fichero.delete();

        //Creamos la base de datos embebida dentro del proyecto
        this.baseDatos = Db4oEmbedded.openFile("BDJefeHijo.db4o");
        //Añadimos los datos de la tarea
        baseDatos.store(new Jefe("Ángel", 5, 53, new Hijo("Gustavo", 7)));
        baseDatos.store(new Jefe("Nieves", 3, 45, new Hijo("Iván", 3)));
        baseDatos.store(new Jefe("Jesús", 3, 5, new Hijo("Noelia", 3)));
        baseDatos.store(new Jefe("Dolores", 5, 63, new Hijo("Sergio", 7)));
        baseDatos.store(new Jefe("Vicki", 3, 5, null));
        baseDatos.store(new Jefe("Fátima", 5, 63, new Hijo("Lidia", 27)));
        baseDatos.store(new Jefe("Juan Luis", 3, 5, null));
        baseDatos.store(new Jefe("Elena", 1, 42, new Hijo("David", 19)));
        baseDatos.store(new Jefe("Miguel", 20, 45, new Hijo("Paula", 3)));
        baseDatos.store(new Jefe("Jesús", 19, 44, new Hijo("Rubén", 12)));
    }
}
```

- **Clase *Menu_Principal*:** Esta clase contendrá el método main() que mostrará un menú al usuario con las distintas opciones que nos propone la tarea:

```
public class Menu_Principal {

    /**
     * Método principal o main
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        //Declaración de variables y objetos
        int op;
        boolean correcta=true;
        Scanner teclado = new Scanner(System.in);
        Controlador_BBDD controlBBDD = new Controlador_BBDD();
        //Instanciamos la clase de la base de datos
        BDJefeHijo bdJefeHijo = new BDJefeHijo();
        bdJefeHijo.crearBBDD();

        //Muestra por pantalla el menú de opciones y llamada a los métodos
        //necesarios según opción seleccionada del usuario
        do{
            try{
                System.out.println("-----");
                System.out.println("MENÚ DE OPCIONES EJERCICIO 1");
                System.out.println("-----");
                System.out.println("1.- Visualizar los jefes con más de 55 años");
                System.out.println("2.- Incrementar la edad de Miguel incrementando su edad un año más");
                System.out.println("3.- Borrar los jefes que llevan más de 6 años en la empresa");
                System.out.println("4.- Visualizar todos los jefes que no hayan sido borrados");
                System.out.println("5.- Salir");
                System.out.println("Introduzca la opción deseada: ");
                op = teclado.nextInt();
            }
        }
    }
}
```

- **Clase *Controlador_BBDD*:** Nos servirá de enlace entre la base de datos que hemos creado y el menú principal que sirve como interfaz al usuario. Contendrá los métodos para las distintas peticiones que nos propone la tarea:

```
public class Controlador_BBDD {

    public Controlador_BBDD() {

    }

    /**
     * Método que consulta a la bbdd los jefes mayores de 55 años
     *
     * @param db ObjectContainer de la bbdd
     */
    public void consultarJefes55(ObjectContainer db) {

    }

    /**
     * Método para incrementar en 1 la edad del jefe llamado Miguel
     *
     * @param db ObjectContainer de la bbdd
     */
    public void actualizarEdadMiguel(ObjectContainer db) {

    }

    /**
     * Método para eliminar todos los jefes de la base de datos que lleven más 6
     * años en la empresa
     *
     * @param db ObjectContainer de la bbdd
     */
    public void borrarMasAnios(ObjectContainer db) {

    }

    /**
     * Método para consultar todos los jefes que contiene la base de datos
     *
     * @param db ObjectContainer de la bbdd
     */
    public void consultarTodosJefes(ObjectContainer db) {

    }

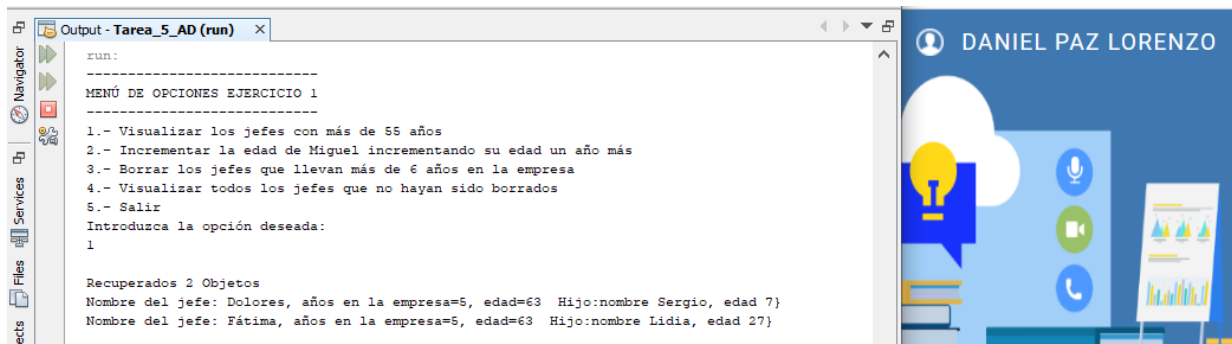
    /**
     * Método para mostrar los resultados obtenidos de la consulta de la base de
     * datos
     *
     * @param resul ObjectSet de la base de datos DB4o
     */
    public void mostrarBBDD(ObjectSet resul) {

    }

}
```

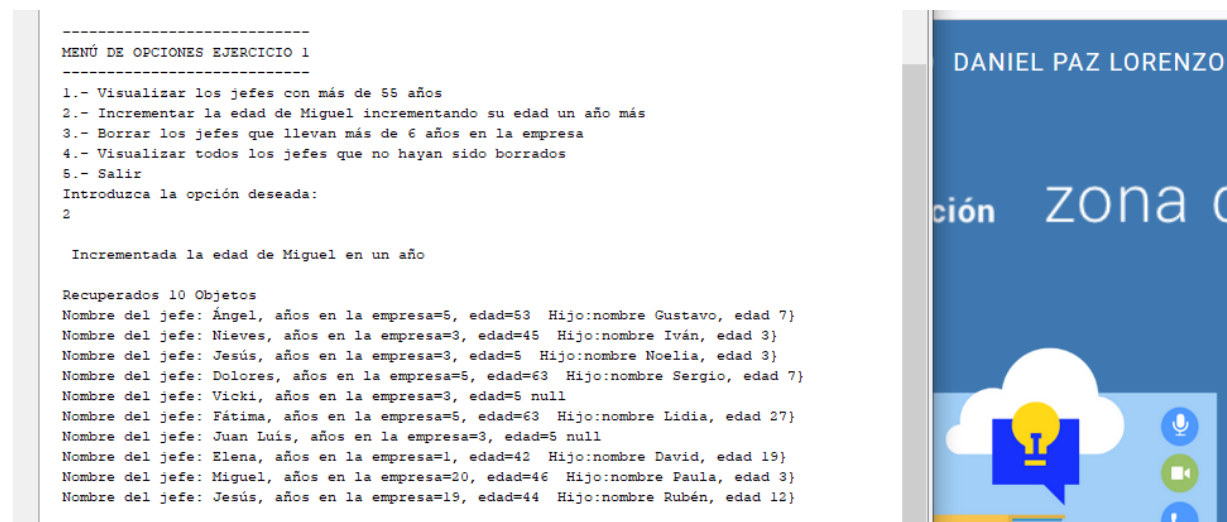
FASE DE PRUEBAS

Una vez tenemos creado ya todo el proyecto tenemos que probarlo, así que lo ejecutamos y seleccionamos la **opción 1**:



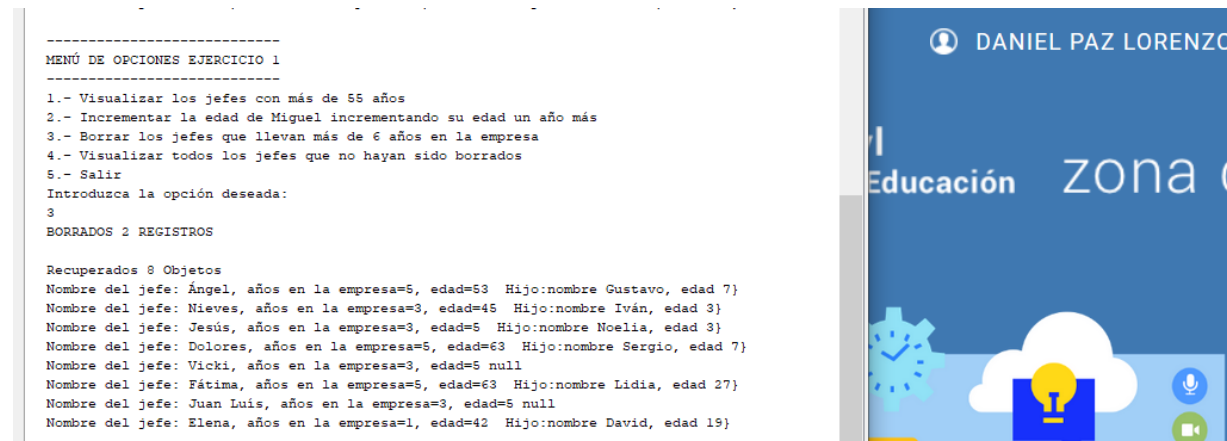
Nos muestra el contenido de los 2 objetos que cumplen la condición

Opción 2:



Muestra mensaje que ha incrementado correctamente y muestra todos los objetos para poder visualizarlos

Opción 3:



Borra los jefes y muestra los datos actualizados

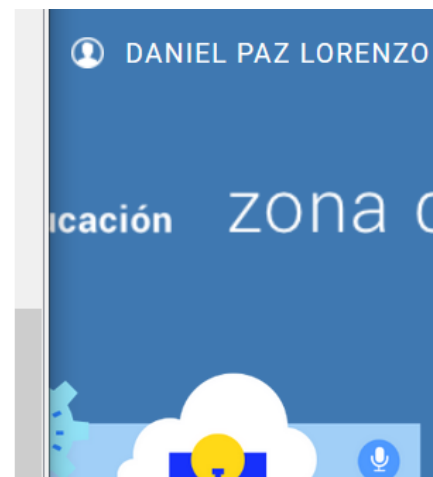
Opción 4:

```

-----
MENÚ DE OPCIONES EJERCICIO 1
-----
1.- Visualizar los jefes con más de 55 años
2.- Incrementar la edad de Miguel incrementando su edad un año más
3.- Borrar los jefes que llevan más de 6 años en la empresa
4.- Visualizar todos los jefes que no hayan sido borrados
5.- Salir
Introduzca la opción deseada:
4

Recuperados 8 Objetos
Nombre del jefe: Ángel, años en la empresa=5, edad=53 Hijo:nombre Gustavo, edad 7}
Nombre del jefe: Nieves, años en la empresa=3, edad=45 Hijo:nombre Iván, edad 3}
Nombre del jefe: Jesús, años en la empresa=3, edad=5 Hijo:nombre Noelia, edad 3}
Nombre del jefe: Dolores, años en la empresa=5, edad=63 Hijo:nombre Sergio, edad 7}
Nombre del jefe: Vicki, años en la empresa=3, edad=5 null
Nombre del jefe: Fátima, años en la empresa=5, edad=63 Hijo:nombre Lidia, edad 27}
Nombre del jefe: Juan Luis, años en la empresa=3, edad=5 null
Nombre del jefe: Elena, años en la empresa=1, edad=42 Hijo:nombre David, edad 19}

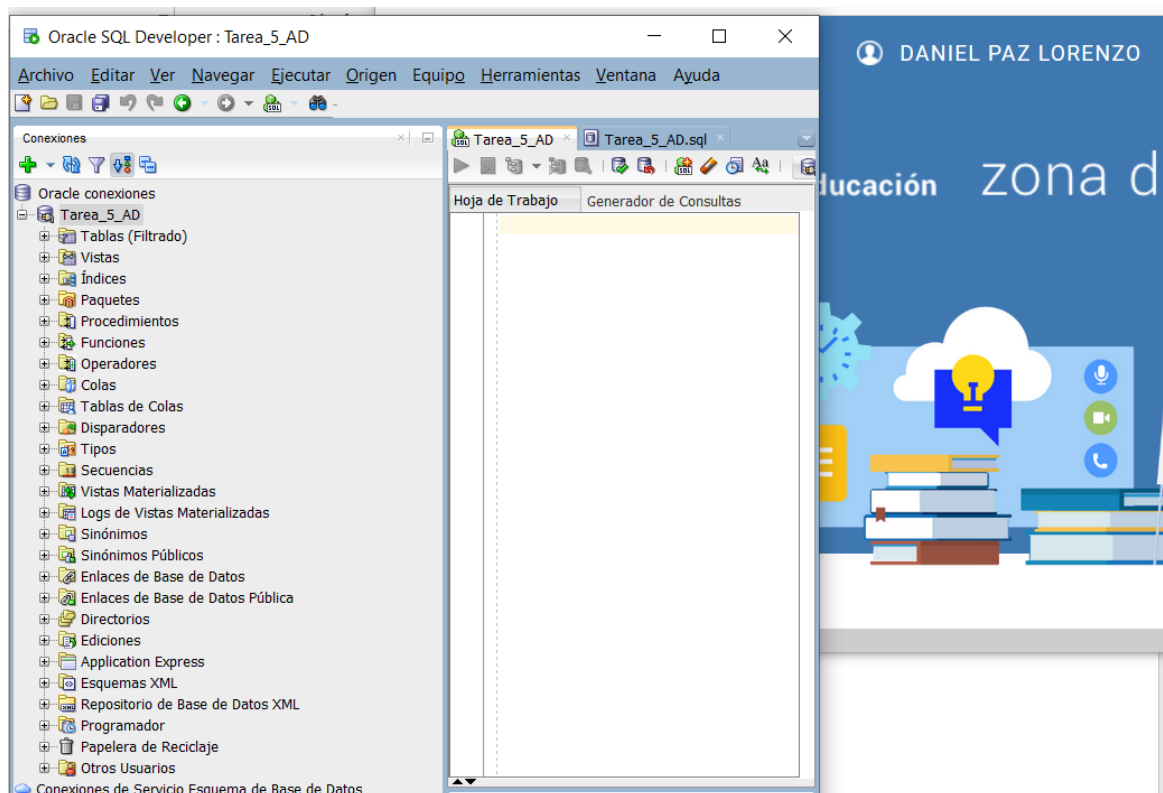
```



Muestra los objetos no borrados de la base de datos

EJERCICIO 2:

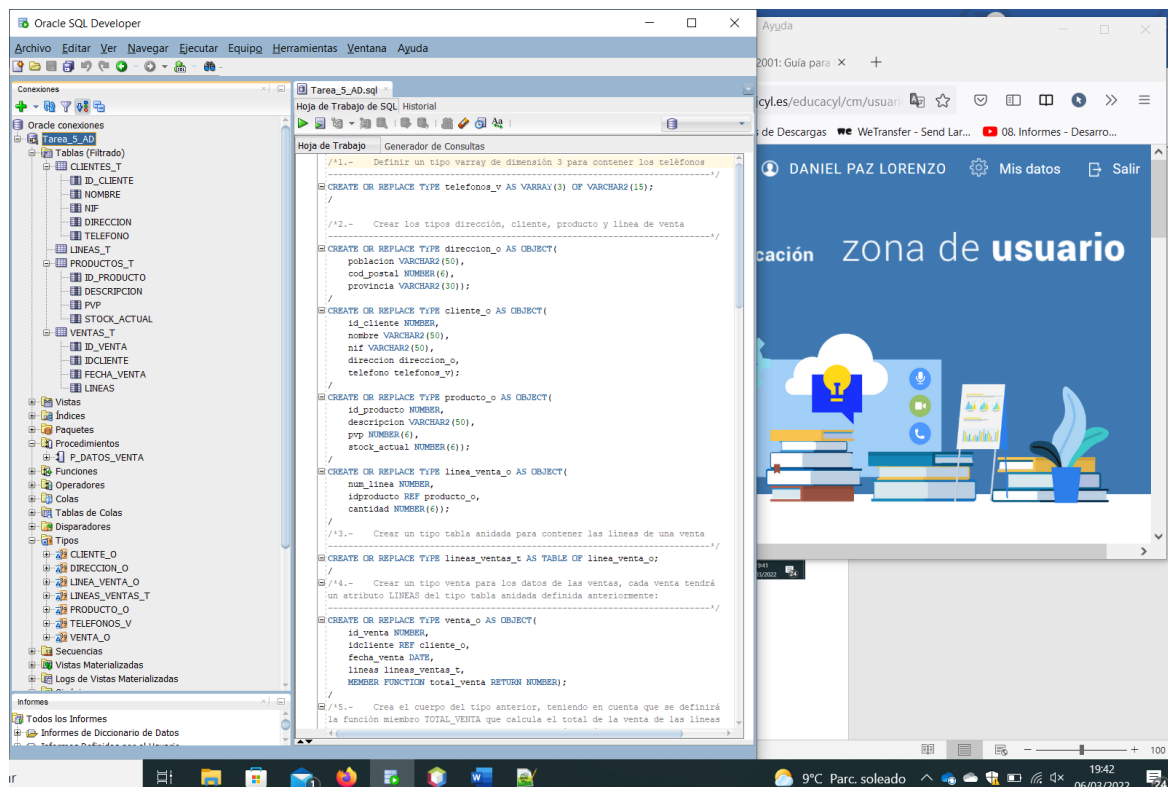
Se nos pide que dado un modelo relacional con distintas tablas las transformemos en una base de datos objeto-relacional creando nuevos tipos, tablas que contendrán objetos y tablas que contendrán otras tablas anidadas. Para ello vamos a crear un nuevo espacio de trabajo dentro del gestor de base de datos de Oracle que tengo instalado en el ordenador. Para poder trabajar más cómodamente con dicho espacio de trabajo conectaremos la base de datos con la aplicación “SQLDeveloper”:



Una vez conectada ya podemos trabajar con la nueva base de datos que queremos crear

Adjunto en la tarea he añadido un script que contiene los 12 puntos que nos pide la tarea para crear la base de datos objeto-relacional. El script tiene extensión “.sql” que permite ser reconocida y ejecutada por el sistema gestor de la base de datos. Durante la tarea y para poder ir probando que el resultado es el deseado yo he ejecutado cada línea de script por separado pero la he configurado para poder ejecutar el script completo dentro de un espacio de trabajo vacío.

El resultado final de la ejecución sería la creación de los nuevos tipos, tablas y un procedimiento:




En el menú de conexiones de la izquierda vemos que se han creado correctamente los nuevos tipos y tablas creadas

Se nos pide insertar datos de prueba y luego a través del procedimiento creado visualizar los resultados. Para probar llamamos a este método con las 2 ventas que hemos insertado, mostrando las siguientes salidas:

Id_venta = 1

```
/*Llamamos al procedimiento que hemos creado*/  
BEGIN  
    p_datos_venta(1);  
END;  
/
```

Salida de DBMS

 | Tamaño de Buffer: 20000

Tarea_5_AD x

CLIENTE: Juan
ID VENTA: 1 | Fecha: 05/03/22
POBLACION: Puebla de Sanabria, c.p.:49007, Provincia:Zamora

LINEA	PRODUCTO	PRECIO	CANTIDAD	TOTAL
-------	----------	--------	----------	-------


1	Harina	2 €	1	2 €
2	Rasqueta	5 €	2	10 €

Precio total:

Id_venta = 2

```
/*Llamamos al procedimiento que hemos creado*/  
BEGIN  
    p_datos_venta(2);  
END;  
/
```

Salida de DBMS

 | Tamaño de Buffer: 20000

Tarea_5_AD x

CLIENTE: Juan
ID VENTA: 2 | Fecha: 06/03/22
POBLACION: Puebla de Sanabria, c.p.:49007, Provincia:Zamora

LINEA	PRODUCTO	PRECIO	CANTIDAD	TOTAL
-------	----------	--------	----------	-------

1	Harina	2 €	2	4 €
2	Amasadora	450 €	1	450 €
2	Banneton	8 €	5	40 €

Precio total: