

**INFORME**

**ABP DESARROLLO 2 GUARDERÍA**

**INTEGRANTE:**

* Javier Andres Herrera Manjarrez (85506).
* Carlos Andres Coneo Díaz (91174).
* Andres Fabian Burgos De las salas (85960).

**PRESENTADO A**: ING. John Carlos Arrieta Arrieta.

**INGENIERÍA EN SISTEMAS TERCER SEMESTRE**

**CARTAGENA DE INDIAS D, T Y C**

**08/2/ 2022**

**Términos**

**Programación orientada a objetos:** La programación orientada a objetos (POO) es un paradigma de programación que usa objetos para crear aplicaciones. Está basada en tres pilares fundamentales: herencia, polimorfismo, encapsulación. Su uso se popularizó a principios de la década de 1990.

**Lenguaje de programación:** En informática, se conoce como lenguaje de programación a un programa destinado a la construcción de otros programas informáticos. Su nombre se debe a que comprende un lenguaje formal que está diseñado para organizar algoritmos y procesos lógicos que serán luego llevados a cabo por un ordenador o sistema informático, permitiendo controlar así su comportamiento físico, lógico y su comunicación con el usuario humano**.**

**File:** Este nos permite buscar un archivo dentro del paquete NetBeans o dentro del pc.

**Arraylist:** La clase ArrayList en Java, es una clase que permite almacenar datos en memoria de forma similar a los Arrays, con la ventaja de que el número de elementos que almacena, lo hace de forma dinámica, es decir, que no es necesario declarar su tamaño como pasa con los Arrays.

**Constructor java:** En Java es un método especial dentro de una clase, que se llama automáticamente cada vez que se crea un objeto de esa clase.

Vehículo minivan = new Vehículo ()

**Clase:** Cuando defines una clase, declaras su forma y naturaleza exactas. Una clase se crea al usar la palabra clave “class”

Ej: public class NombreClase{

**Objetivos**

* **General**
* Crear un Aplicativo con interfaz que permita realizar los distintos procesos en una guardería (registro de infantes y sus acudientes, almacenar los menús, registrar el consumo de menús de los niños y dar un total de los gastos realizados por el ).
* **Específicos**
* Crear Archivos .txt para poder almacenar la información y consultarla de manera dinámica.
* Identificar el problema
* Investigar las posibles soluciones.

**Definición de problemas**

Una guardería desea controlar los gastos que cada uno de los niños realiza a través de su asistencia y de las comidas que consume. De cada niño se desea conocer los datos propios de su matrícula en el centro educativo, es decir, el número de matrícula, el nombre, la fecha de nacimiento y la fecha de ingreso en la guardería. Para aquellos niños que se hayan dado de baja, también se desea conocer la fecha de la baja. Diagrama de clases

Los niños sólo pueden ser recogidos en la guardería por un conjunto de personas que suelen ser un familiar del niño o un conocido de sus familiares. De éstos se desea conocer el DNI, el nombre, la dirección y al menos un número de teléfono de contacto. Además, debe de quedar constancia de cuál es la relación entre la persona autorizada y el niño. El coste mensual del niño en la guardería es abonado por una persona, de la que se desea conocer el DNI, el nombre, la dirección, el teléfono, y el número de la cuenta corriente en la que se realizará el cargo. Estas personas también pueden estar autorizadas para recoger al niño. En la guardería aparece un conjunto de menús, compuesto por una serie de platos concretos, cada uno de los cuales presentan unos ingredientes determinados. Cada menú se identifica por un número, mientras que los platos y los ingredientes se caracterizan por su nombre. Un niño puede ser alérgico a diferentes ingredientes, y por tanto no puede consumir los platos en los que aparece este ingrediente. Estas alergias deben de ser controladas para evitar posibles intoxicaciones en los niños.El cargo mensual de un niño se calcula como la suma de un coste fijo mensual y el coste de las comidas realizadas. Este último se obtiene a partir del número de días que el niño ha comido en la guardería, por lo que resulta necesario controlar dicho número. Además, se desea saber el menú que ha consumido cada niño cada día.

* Almacenar la información registrada en un archivo txt

**Análisis del problema**

* Se debe realizar la creación de un algoritmo en Java que nos permita ingresar los datos y almacenarlos en archivos txt.
* Realizar un programa en java que nos permita guardar y gestionar los datos de un jardín infantil datos del actor principal del problema de Infante (numero de matricula, nombre, fecha de nacimiento, fecha de ingreso a la guardería, si se dio de baja ingresar la fecha, alergias de infante ), y tomar los datos de su acudiente(DNI, nombre, dirección, numero de teléfono, relación con el infante) ha si como los de quien costea la mensualidad (DNI, nombre, dirección, número de teléfono, número de cuenta).

Al Igual debería hacer con los menús que consumen los infante(número de menú, nombre de los platos, costo de los platos, ingrediente, costo del menú) y al realizar un consumo se le asignará un menú a un infante y se ingresará en los txt.

**Organización de idea**

Se debe realizar la creación de un algoritmo el cual debemos hacer usos de librerías y para así poder registrar, consultar y mostrar los datos de los infantes, acudientes, menús, gastos, además almacenar estos datos para así tener toda la información requerida.

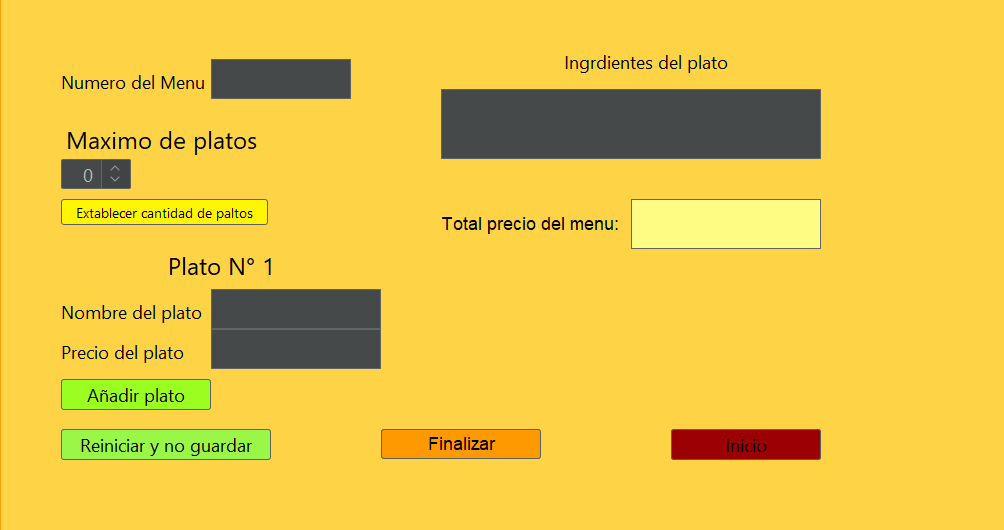
**Justificación**

Los motivos que nos llevaron a la realización de este proyecto con la metodología de aprendizaje basado en problemas, es como se mencionó anteriormente aprender nuevas habilidades con la investigación realizada respecto al problema planteado.

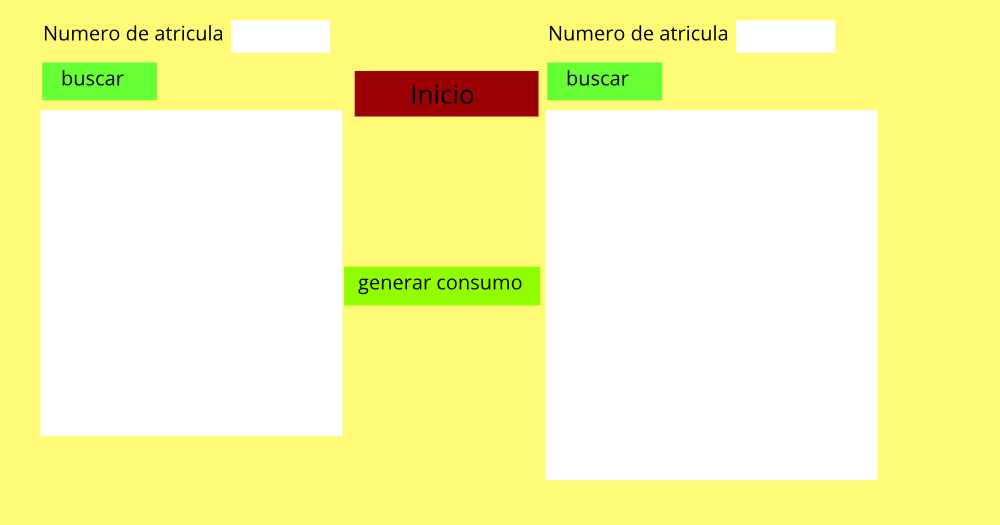
**Mockus - Prototipos de IU**













**Programa en java**

**Clases:**

* Acudiente
* ListasYmetodos
* Matricula
* Menu
* PersonaCosteaMensualidad
* consumoDeInfante

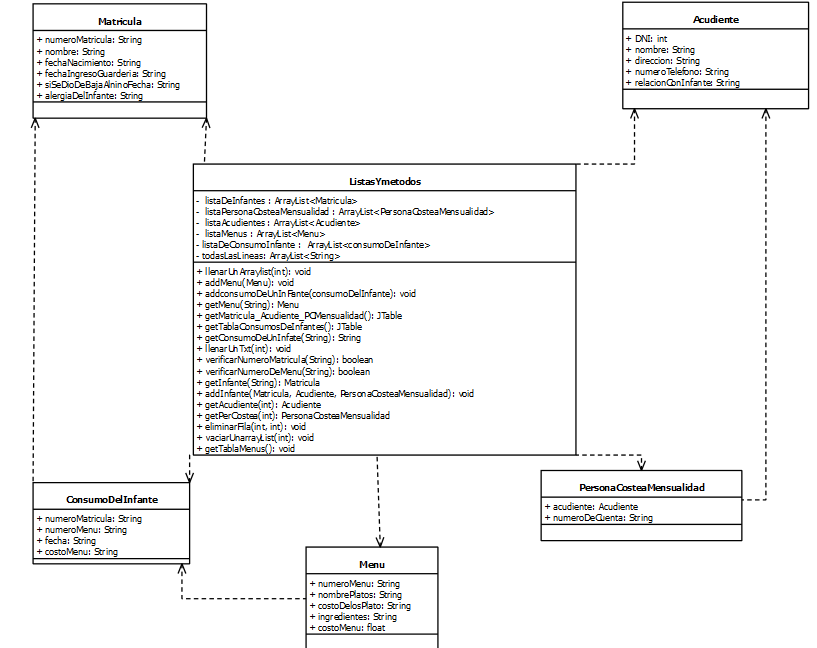
**Funciones:**

* llenarUnArraylist (in int): void
* addInfante(in Matricula,in, in Acudiente, in PersonaCosteaMensualidad): void
* getMatricula\_Acudiente\_PCMensualidad():JTable
* getTablaConsumosDeInfantes():JTable
* verificarNumeroMatricula():boolean
* verificarNumeroDeMenu():boolean
* eliminarFila(in int, in int):void
* addInfate(in Matricula, in Acudiente, in PersonaCosteaMensualidad ):void
* addMenu(in Menu): void
* addConsumoDeUnInfante(in consumoDeInfante):void
* getInfante( in String): Matricula
* getAcudiente(in int):Acudiente
* getperCostea(in int): PersonaCosteaMensualidad
* getMenu (in String):Menu
* vaciarUnarrayList(in int):void
* getConsumoDeUnInfate(in String):String
* getTablaMenus():void

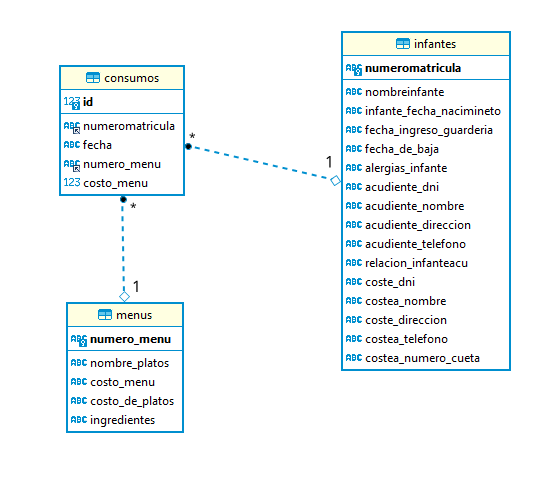
**Archivos:**

* Consumo.txt
* Infante.txt
* Menus.txt

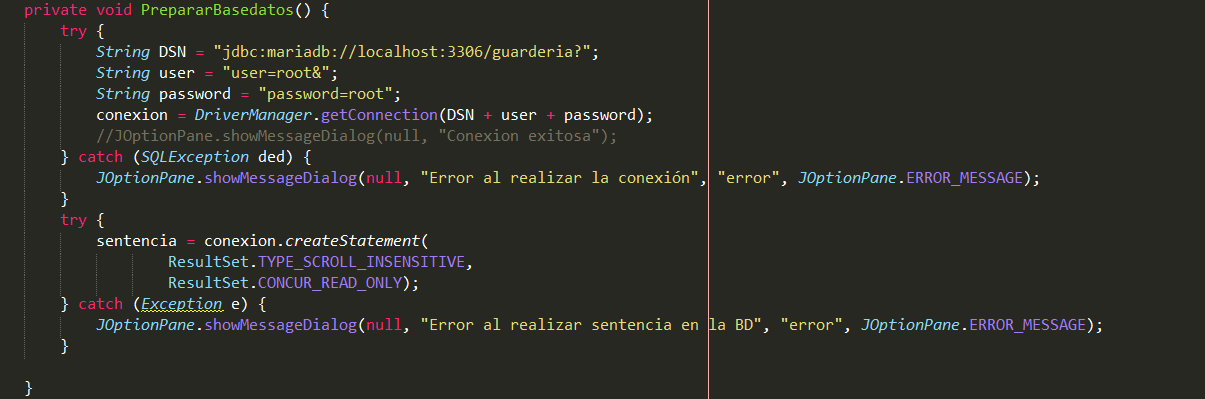
**UML**



**Modelo Relacional de BD**



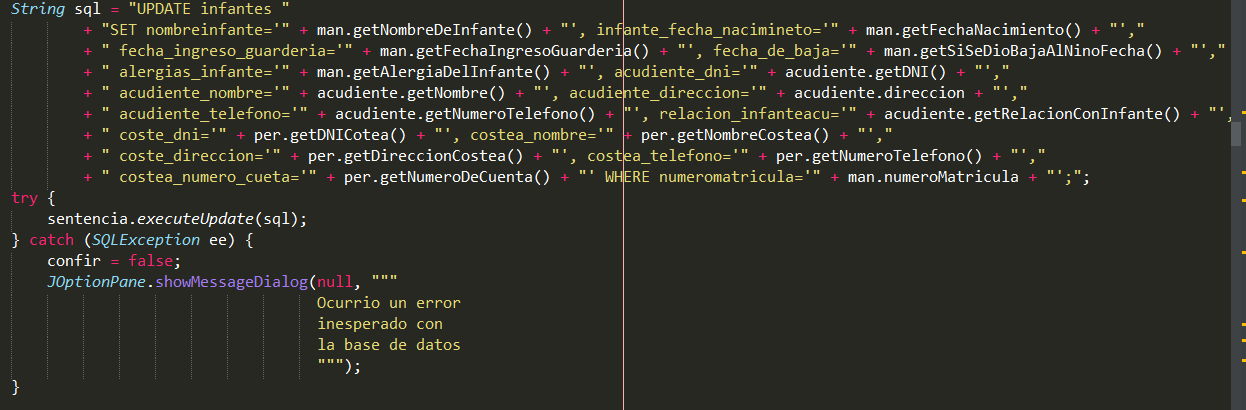
**Fragmentos de código de cómo se conecta java con la bd**



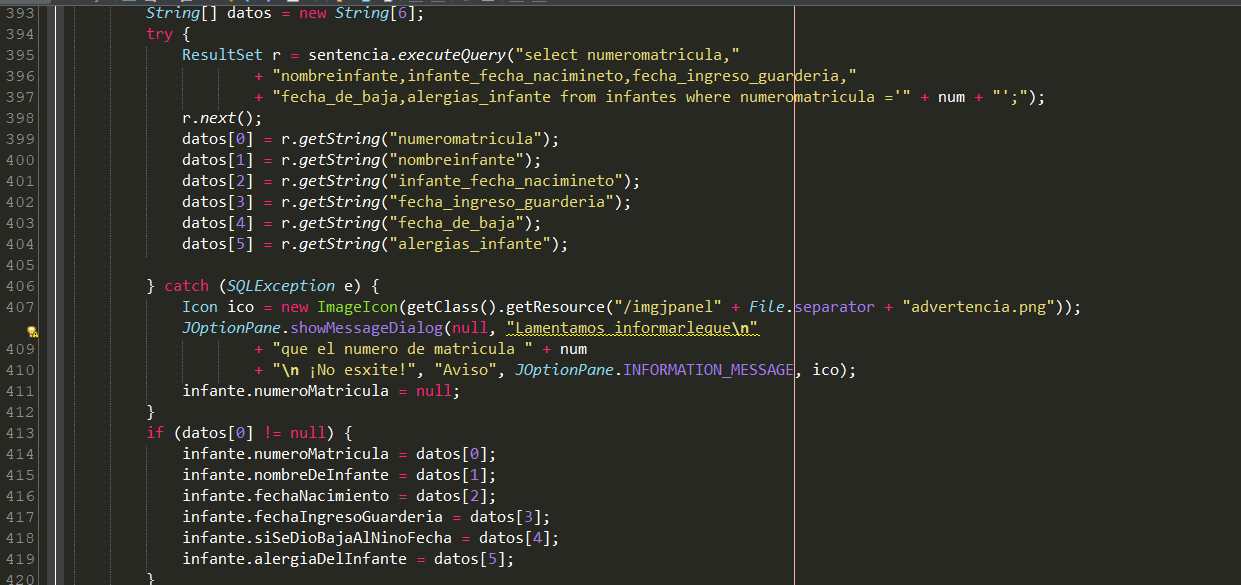
**Fragmento de código para guardar**



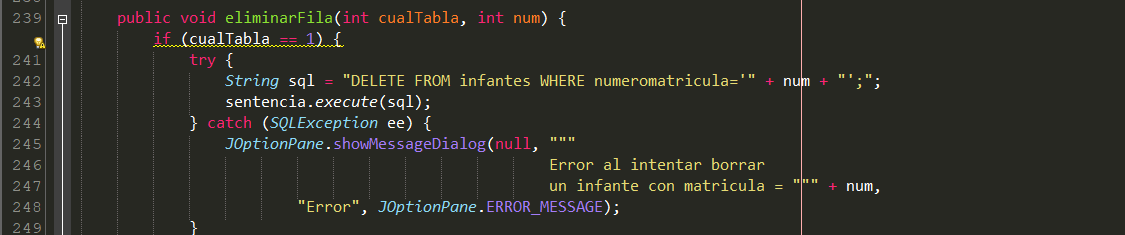
**Fragmento de código para actualizar**



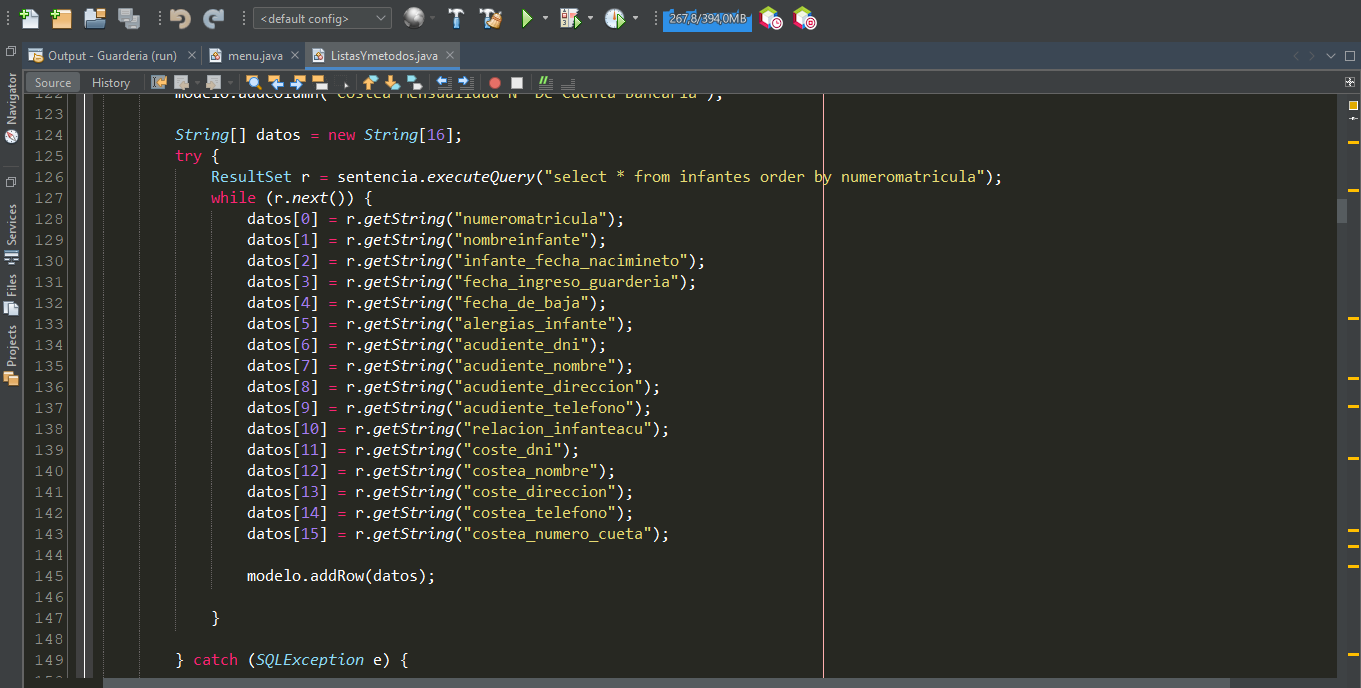
**Fragmento de código para buscar**



**Fragmento de código para eliminar**



**Fragmento de código para listar**



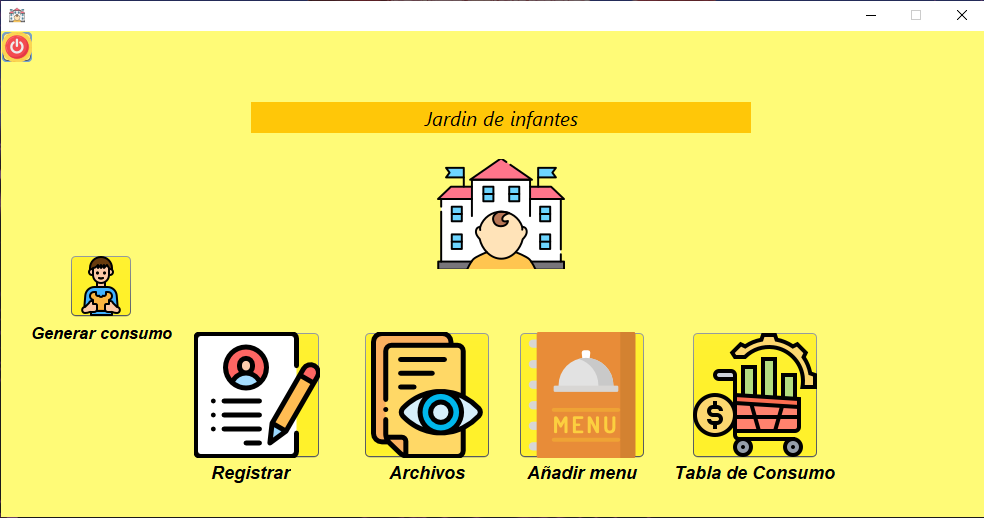
**Obtención de información**

**Software de aplicación**: En informática, se entiende por software de aplicación, programas de aplicaciones o en algunos casos aplicaciones, al conjunto de los programas informáticos generalmente instalados en el sistema por el usuario, y diseñados para llevar a cabo un objetivo determinado y concreto, de tipo lúdico, instrumental, comunicativo, informativo, etc.

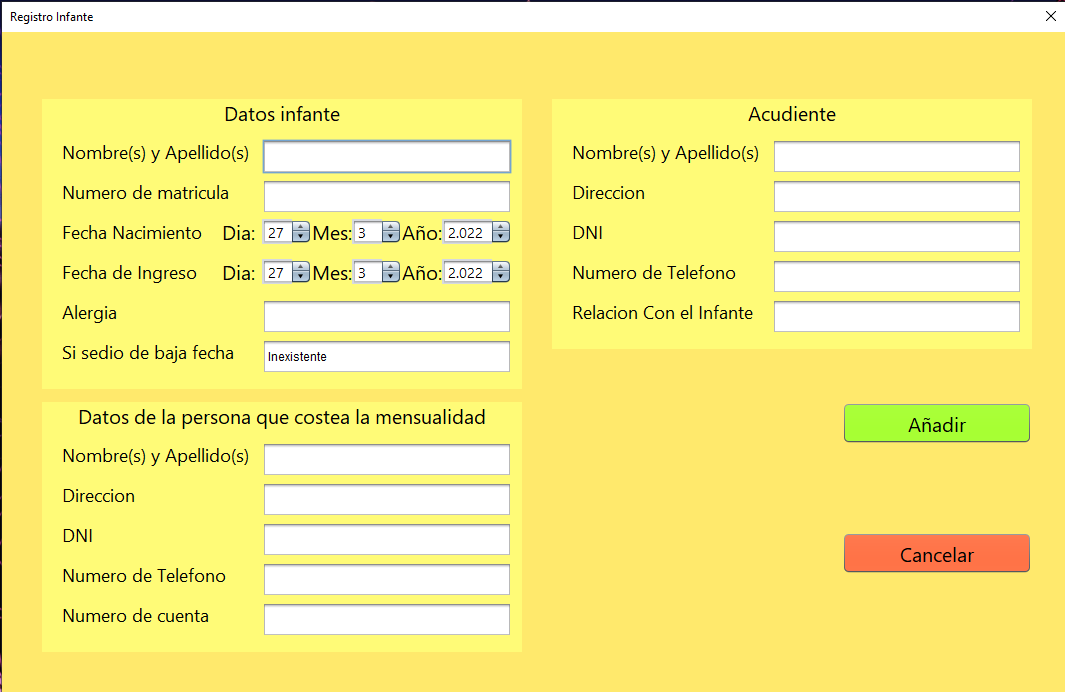
**Librería:** son trozos de código hechas por terceros. Esto nos facilita mucho la programación y hace que nuestro programa sea más sencillo de hacer y luego de entender.

**Manual del usuario o persona encargada**

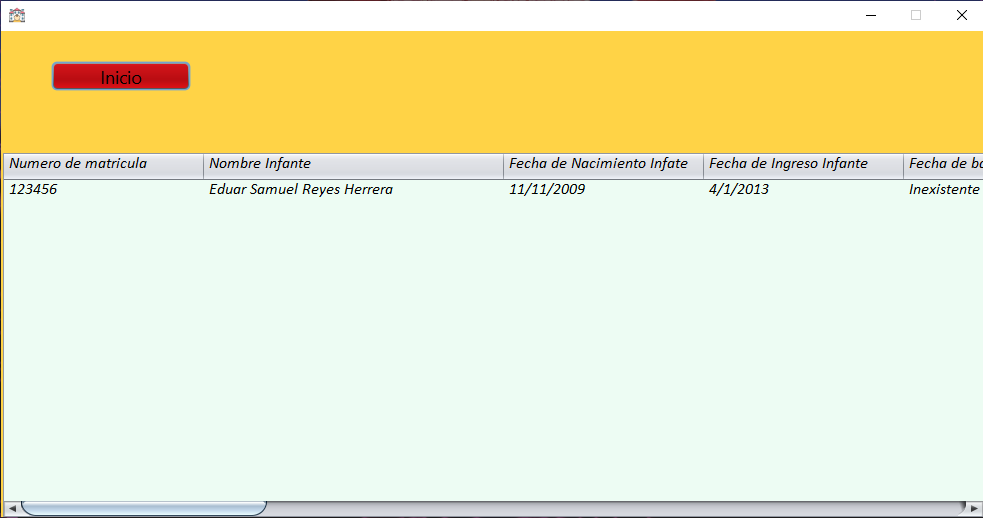
* Al iniciar el programa verán la siguiente interface con cinco botos cada uno con su funcionalidad.



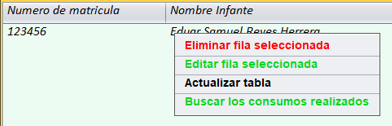
* Al presionar el botón registrar se le mostrará una venta emergente donde deberá ingresar todos los datos solicitados en la interface sin excepción



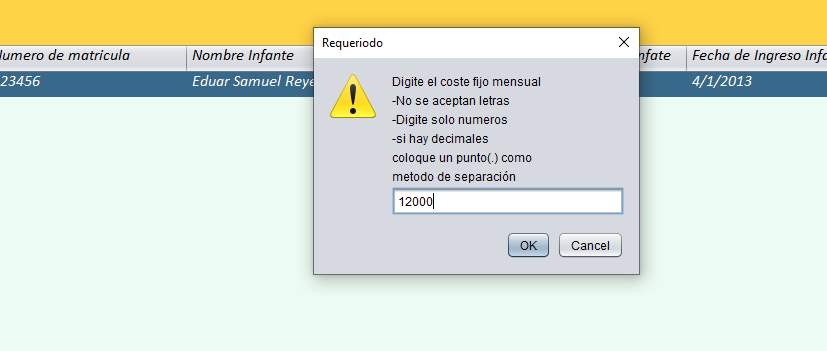
* Al registrar todos los datos ya preverlos registrados en la tabla que a parecerá al presionar el botón archivos.



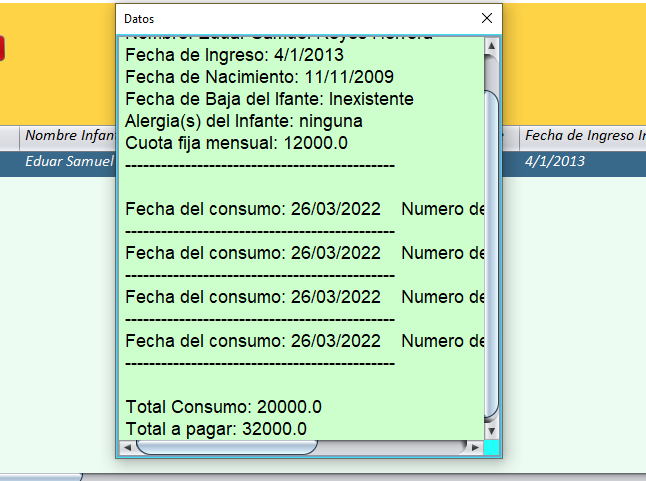
* Al presionar el clic derecho le parecerá un menú con distintas opciones, las dos primeras opciones son fáciles de intuir



* Al presionar buscar consumos realizados se muestra un menú emergente



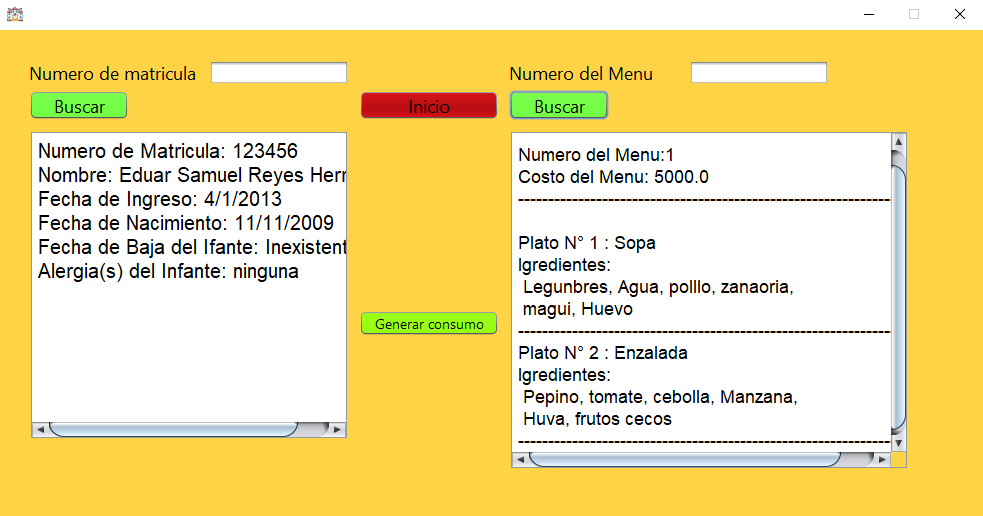
* Al presionar ok en el menú emergente se le mostrara una ventana de dialogo donde estarán los datos correspondientes a ese infante junto con los consumos que realizado y el total apagar



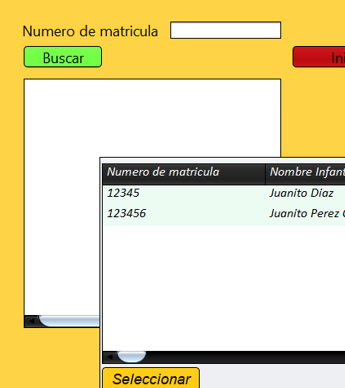
* Al presionar el botón añadir menú se mostrara un panel emergente con una pequeña instrucción indicando que debe establecer el número de plato primero pero eso será en la siguiente interface.



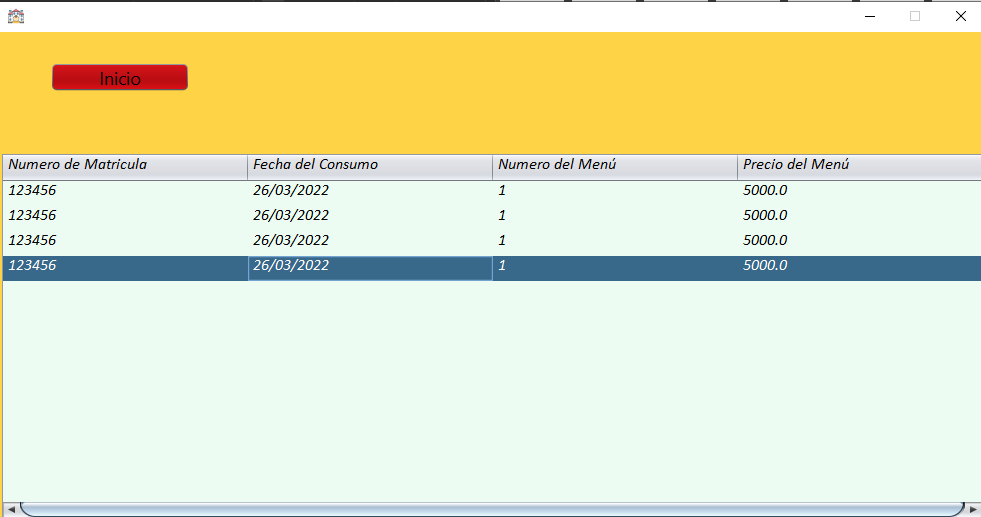
* Después de realizado el formulario ya podrá asignarle este menú a un consumo que realice para un infante, esto lo podrá hacer al presionar el botón generar consumo que está en la interface inicial.



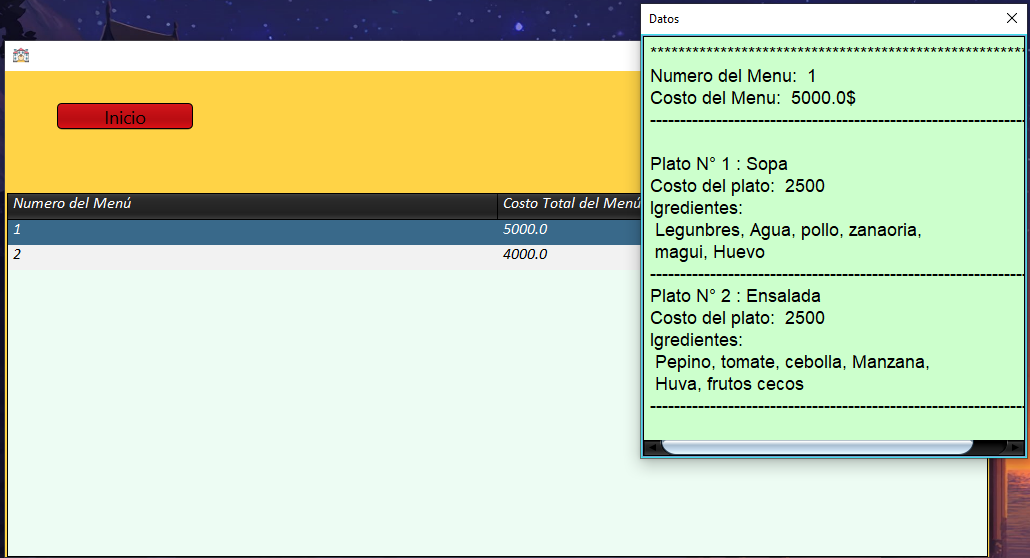
* Al presionar clic derecho sobre el campo el que se muestran los datos de los infantes



* A consecuencia ya podrá ver el consumo registrado en la tabla consumo Al igual que los menús registrados.







**Conclusión**

Después de ver los términos de Software, Librerías, manejo de Archivos; Podemos inferir que aplicando todas estas tecnologías podremos solventar el problema.

**Bibliografía**

(Raffino., 2020) **Lenguaje de programación.**

(Raffino M. E., 2020) **Software de Aplicación.**

(Sanguino, s.f.) **Librería.**

[**https://docs.oracle.com/javase/7/docs/api/java/io/File.html**](https://docs.oracle.com/javase/7/docs/api/java/io/File.html)

(Programación Orientada a Objetos en Java, 2103) **Programación Orientada a Objetos en Java.**