



---

# MANUAL TECNICO

---



JUNE 7, 2021  
GRUPO 7

## Tabla de contenido

|                                      |   |
|--------------------------------------|---|
| Introducción.....                    | 2 |
| Objetivos.....                       | 2 |
| 1. Vocabulario técnico.....          | 3 |
| 2. Requerimientos.....               | 4 |
| 3. Instalación y configuración ..... | 5 |
| 4. Configuración del sistema .....   | 6 |
| Estructura del proyecto.....         | 6 |
| Directorio <i>src</i> .....          | 6 |
| 5. Lógica del programa.....          | 7 |

## **Introducción**

A continuación, son descritos los elementos técnicos e informáticos que componen *Restaurant Manager*. El documento introduce al usuario a la programación orientada a objetos y los aspectos y detalles que componen a la aplicación, guiando al usuario de una manera sencilla y específica, mostrando así las características del programa y su funcionamiento.

## **Objetivos**

Familiarizar al usuario con el término programación orientada a objetos y el uso de la aplicación desarrollada a través de dicho paradigma de programación. Se orienta al usuario con los conocimientos sobre la configuración y el entorno de la aplicación, así como también se presenta la instalación de la aplicación, describiendo los distintos elementos y aspectos relevantes del sistema que lo componen.

## **1. Vocabulario técnico**

- Variable global: variable a la cual se puede acceder desde cualquier ámbito o bloque de código.
- IDE de programación: entorno en el cual se desarrolla el código, es decir, es un editor, compilador, depurador y constructor de código.
- Aplicación de consola: aplicación que se ejecuta dentro de una ventana de línea de comandos.
- CMD: CoMmanD, es un intérprete de comandos incluido en los sistemas operativos de las computadoras
- Java class: se le conocen como plantillas para la creación de objetos dentro de un entorno de programación.
- Objeto: ente abstracto usado en programación que permite separar los diferentes componentes de un programa, simplificando así su elaboración, depuración y posteriores mejoras.
- Programación orientada a objetos: es un paradigma de programación don los objetos se utilizan como metáfora para emular las entidades reales del negocio a modelar.
- Paradigma de programación: consiste en un método para llevar a cabo cómputos y la forma en la que deben estructurarse y organizarse las tareas que debe realizar un programa.

## **2. Requerimientos**

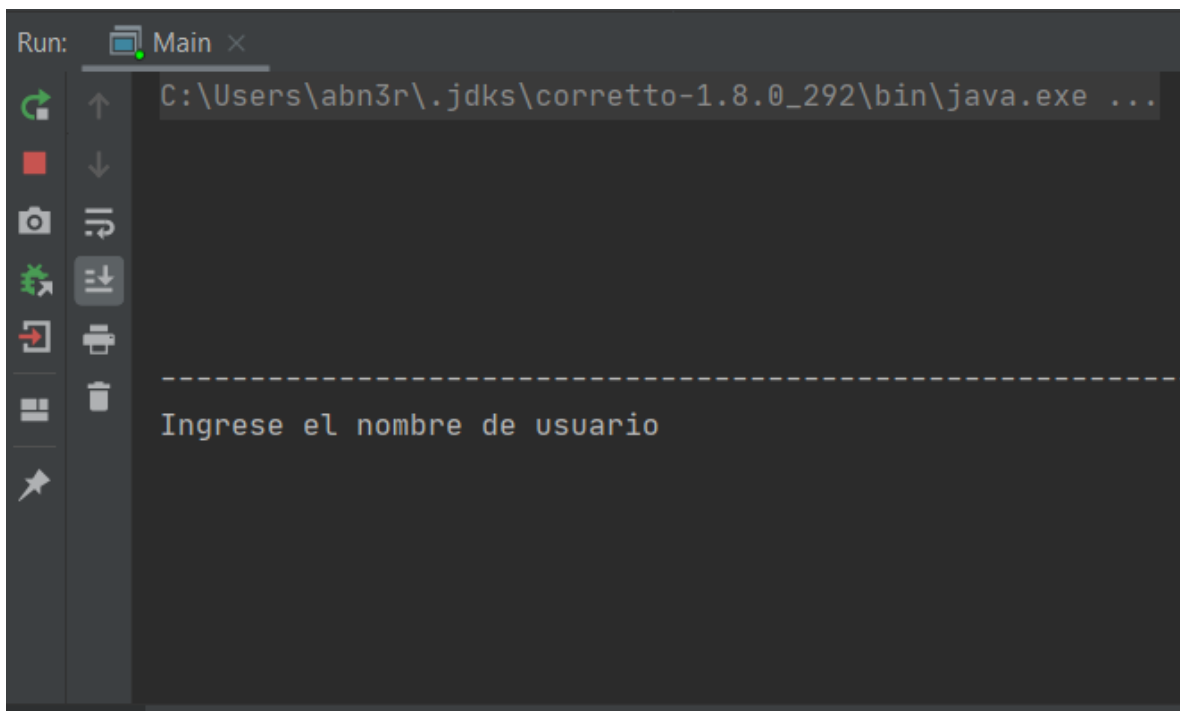
La aplicación puede ser ejecutada en cualquier sistema operativo, sin embargo, al ser una aplicación de consola requiere de los siguientes requerimientos:

- Software de Java versión 1.08
- IDE de Programación (IntelliJ, eclipse, Netbeans, etc)
- Java virtual machine

### 3. Instalación y configuración

La aplicación, al ser de tipo de consola, será ejecutada a través de el *CMD* de la computadora, y para eso bastará de escribir un par de comandos para tener la aplicación funcionando.

1. Descargue la carpeta que contiene la aplicación y guarde la ruta del archivo (C:\Users\user\files)
2. El siguiente paso es abrir el *CMD* de la computadora:
  - 2.1 Utilizar la combinación de teclas *windows+enter*, en el cuadro de texto escribir “cmd” y presionar enter.
3. Una vez ejecutado el *CMD* de la computadora, escribir *java -jar ruta del archivo*.
4. A continuación, la aplicación comenzará a funcionar.



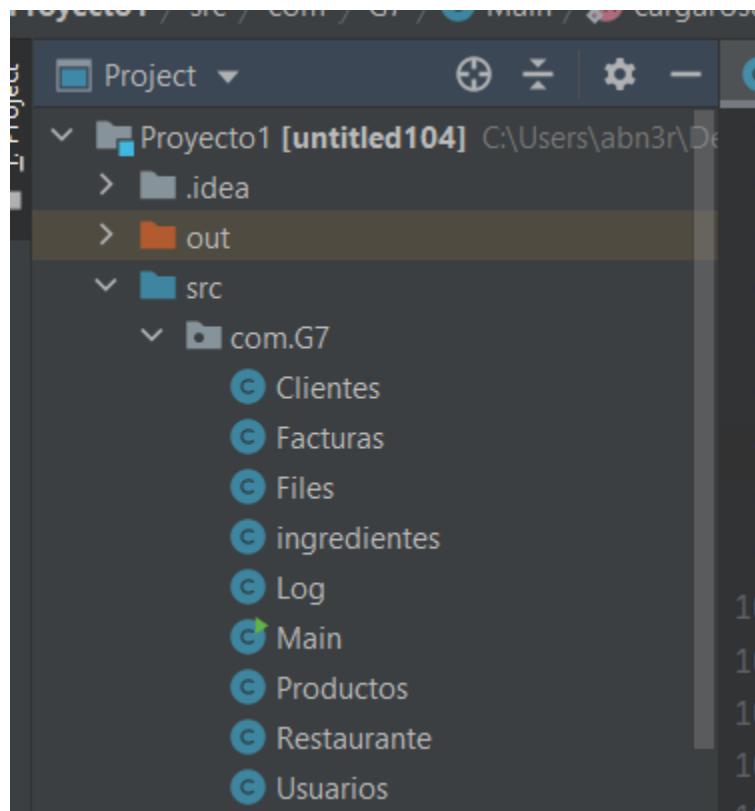
*Nota: para el correcto funcionamiento de la aplicación, es necesario que se cuente con la versión indicada de java.*

#### 4. Configuración del sistema

La aplicación *Restaurant Manager* está desarrollada bajo el lenguaje de programación JAVA, lo cual permite un desarrollo y ejecución en casi cualquier tipo de dispositivo que cuente con la Java virtual machine. A continuación, se describe la estructura de la aplicación.

##### Estructura del proyecto

El desarrollo de la aplicación presenta consigo la siguiente estructura de directorios y clases:



##### Directorio *src*

Dentro de este directorio se encuentran los recursos del programa, así como también el código fuente de la aplicación y las distintas clases que componen a la aplicación.

## 5. Lógica del programa

En cuanto a el desarrollo de la aplicación, el proyecto fue realizado utilizando la programación orientada a objetos, por lo que varias clases fueron ocupadas para el correcto funcionamiento de la aplicación, esto con el objetivo de acercarse tanto como fuera posible a la realidad que se trata de conceptualizar, siendo este el caso de el manejo de un restaurante.

Entre los detalles que destacan, se tiene presente la declaración de todas las variables globales, las cuales son cruciales en el desarrollo del código para una ejecución más limpia.

```
public class Main {  
    public static Scanner consola = new Scanner(System.in);  
    public static ArrayList<Usuarios> usuariosArr;  
    public static ArrayList<Productos> productosArr;  
    public static ArrayList<Clientes> clientesArr;  
    public static ArrayList<Facturas> facturasArr;  
    public static ArrayList<Restaurante> restauranteArr;  
    public static String usuarioLog;
```

Además de esto, la estructura que presentan las distintas clases se ve de la siguiente manera:

```
4  
5 public class Clientes implements Serializable {  
6     public int id;  
7     public String name;  
8     public String address;  
9     public int phone;  
10    public String nit;  
11  
12    public Clientes(int id, String name, String address, int phone, String nit){  
13        this.id = id;  
14        this.name = name;  
15        this.address = address;  
16        this.phone = phone;  
17        this.nit = nit;  
18    }  
19  
20  
21  
22    public int getId() { return id; }  
23  
24    public void setId(int id) { this.id = id; }  
25  
26    public String getName() { return name; }  
27  
28    public void setName(String name) { this.name = name; }  
29  
30    public String getAddress() { return address; }  
31  
32    public void setAddress(String address) { this.address = address; }  
33  
34    public int getPhone() { return phone; }  
35
```

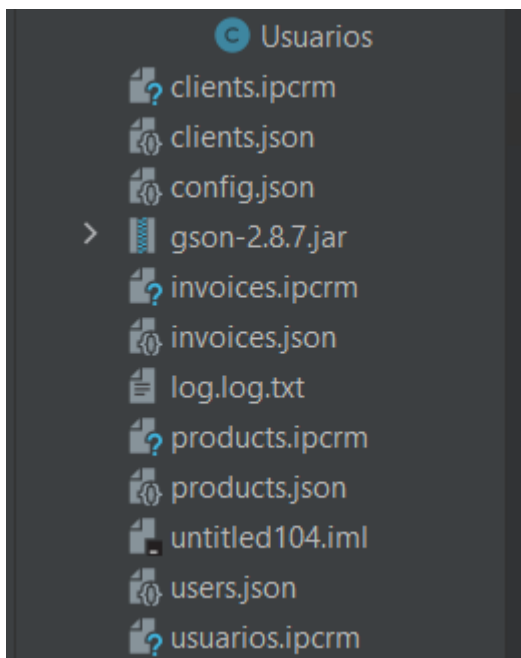
Contando con los distintos identificadores, así como “constructores”, *setters and getters*, los cuales son herramientas que ayudan a la conceptualización y correcto funcionamiento al programar utilizando el paradigma de la programación orientada a objetos



Continuando con la lógica que presenta la aplicación, se encuentra la forma en la que los datos serán cargados, y es que se utiliza un formato de archivo *Json*, el cual permite un manejo más sencillo de cantidades grandes de información almacenada.

```
{ } users (1).json X
C: > Users > abn3r > Downloads > { } users (1).json > ...
1  [
2      {
3          "username": "alexizzarevalo",
4          "password": "helloworld"
5      }
6  ]
```

*Ejemplo archivos Json.*



*Archivos utilizados por la aplicación.*

```
Ingrese el nombre de usuario
alexizzarevalo
Ingrese la contraseña
helloworld
El nombre de usuario y contraseña son válidos, Bienvenido.
-----
Seleccione una opcion:
01. Informacion del restaurante.
02. Usuarios.
03. Productos.
04. Clientes.
05. Facturas.
06. Guardar Cambios.
07. Cerrar.
|
```

*autenticación y menú de la aplicación.*