



# JAVAGOTCHI



## MANUAL DE USUARIO

### ÍNDICE

1. INTRODUCCIÓN.....	1
2. REQUISITOS DEL SISTEMA.....	1
3. INSTALACIÓN.....	1
4. EJECUTAR JAVAGOTCHI.....	3
5. MENÚ DE USUARIO.....	4
6. MENÚ DE MASCOTAS.....	6
7. INTERACCIÓN CON LAS MASCOTAS.....	9
8. REGISTROS AUTOMÁTICOS.....	10

## 1. INTRODUCCIÓN

JAVAGOTCHI se basa en un **sistema de gestión de mascotas virtuales** que permite a los usuarios (dueños) interactuar con sus mascotas a través de acciones como **jugar, alimentar y bañar**. Este sistema gestionará el **registro, consulta y eliminación de usuarios y mascotas, actualizando las estadísticas** de cada mascota según las interacciones, y guardando la información de cada interacción en fecha y hora exactas, todo almacenado en una base de datos.

## 2. REQUISITOS DEL SISTEMA

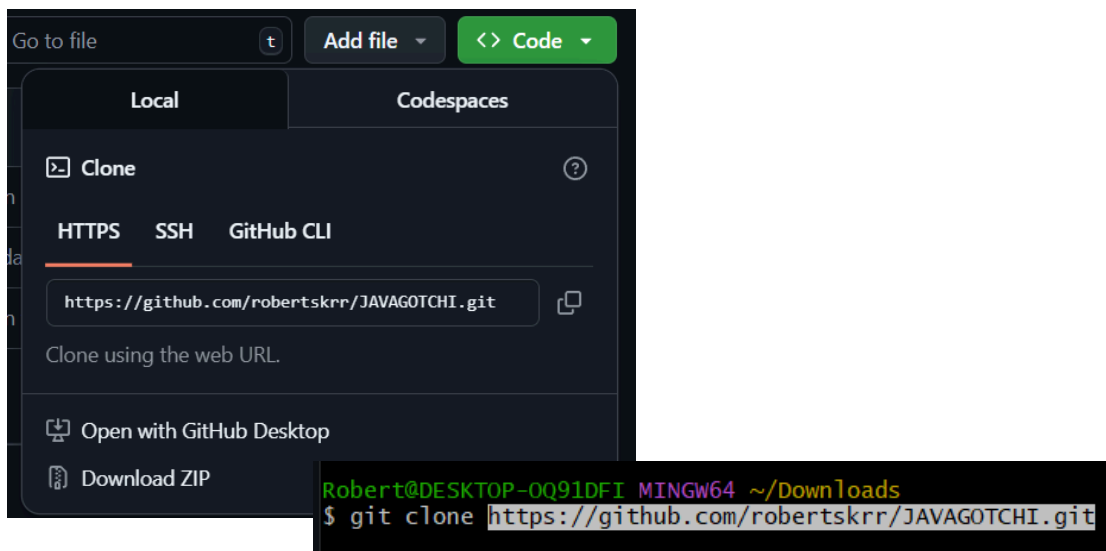
- ★ **JAVA (JDK 21)**
- ★ **mySQL 8.0**
- ★ **JDBC Driver** (incluido en la carpeta */lib* dentro del proyecto)
- ★ **GIT (Opcional)**

## 3. INSTALACIÓN

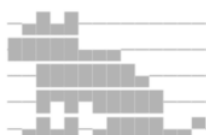
### 3.1. CLONAR REPOSITORIO O DESCARGAR

Desde el **repositorio de Javagotchi** selecciona el apartado **<> CODE** y, **copia el HTTPS o SSH** del repositorio, a continuación **abre Git Bash** en la carpeta donde quieras clonar el repositorio y ejecuta:

```
$ git clone https://github.com/robertskrr/JAVAGOTCHI.git
```



Si no tienes Git instalado puedes seleccionar la opción **Download ZIP** o en el apartado de **Releases** (panel derecho del repositorio) seleccionar la **versión** que quieras descargar.

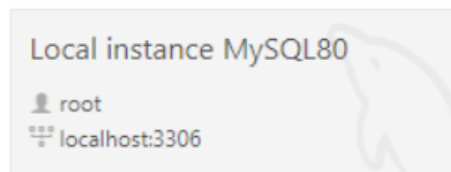


### 3.2. CREACIÓN Y CONFIGURACIÓN DE LA BASE DE DATOS

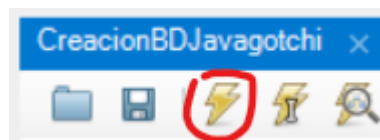
Javagotchi utiliza una **base de datos sql**, así que para poder ejecutar la aplicación sin problemas debemos **crear** la misma y **cargar los datos**, ambos scripts se encuentran en la **carpeta sql**.

1. Abrir mySQL y entrar en tu conexión.

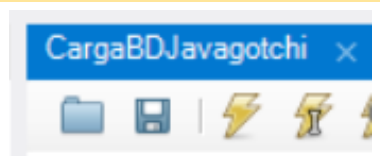
#### MySQL Connections + ⓘ



2. Abrir script sql/CreacionBDJavagotchi.sql y ejecutarlo (pulsa el icono del rayo).



3. Abrir script sql/CargaBDJavagotchi.sql y ejecutarlo.

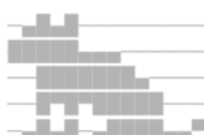


Después de esto, la **base de datos está creada** y, para asegurarnos de que la aplicación funcione perfectamente, debemos **abrir el proyecto Java** con cualquier IDE y ejecutar este **último paso**.

4. Abrir las clases DAO y chequear que los valores de estos campos coinciden con el de la conexión a tu base de datos

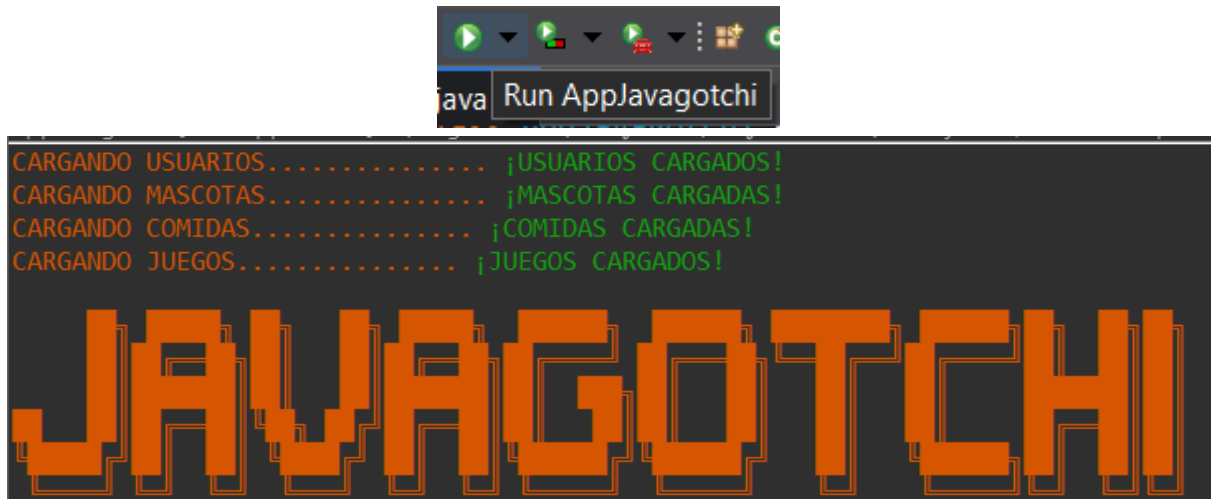
```
private final String USUARIO = "root";  
private final String PASSWORD = "root";  
private final String MAQUINA = "localhost";
```

*\*En el caso de no coincidir cambia estos valores por los tuyos.*

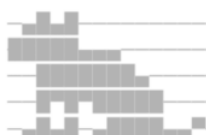


#### 4. EJECUTAR JAVAGOTCHI

Una vez configurada la base de datos, es hora de ejecutar la aplicación. Abre tu IDE preferido, abre el proyecto Java de Javagotchi y ejecuta `Appjavagotchi.java`.



Si al ejecutarlo te sale lo mismo que en la imagen superior, enhorabuena, ya tienes acceso a Javagotchi 😊.



## 5. MENÚ DE USUARIO

Una vez accedes a la aplicación, se presenta el **menú principal** con las siguientes **opciones**:

```
JAVAGOTCHI

** ELIGE UNA OPCIÓN **
1. 🐱 Crear usuario 🐱
2. 🐱 Listar usuarios 🐱
3. 🔍 Buscar usuario 🔍
4. ⚙️ Modificar tu usuario ⚙️
5. 🗑️ Eliminar tu usuario 🗑️
6. 🐾 Menú de mascotas 🐾
7. 🚪 SALIR 🚪
* OPCIÓN:
```

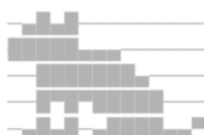
**1. Crear usuario** → Al completar los datos solicitados ,**crea un usuario** (si el username no está utilizado ya por otro usuario) y **lo registra** en la **base de datos**.

```
🐱 CREACIÓN DE USUARIO 🐱
* Username: pruebaManual
* Contraseña: manual123
* Nombre completo: prueba de manual
* Email: manual@manual.com
* Fecha de nacimiento (yyyy-MM-dd): 2000-12-12
* Ciudad: Sevilla
CREANDO USUARIO 'pruebaManual'..... ¡USUARIO CREADO CON ÉXITO!
```

**2. Listar usuarios** → Dependiendo del **tipo de orden** que **elijas** te devuelve los **datos de los usuarios** registrados en forma de **lista** (exceptuando la contraseña obviamente).

```
🐱 USUARIOS DE JAVAGOTCHI 🐱
1. Ordenados por username.
2. Ordenados por fecha de registro.
* ELIGE EL ORDEN: 2
🐱 USUARIOS ORDENADOS POR FECHA DE REGISTRO 🐱

=====
* USERNAME: usuario03 * NOMBRE: Carlos Ruiz * EMAIL: carlos3@correo.com * FECHA NACIMIENTO: 1995-08-09 * CIUDAD: Valencia * FECHA REGISTRO: 2019-01-14
=====
* USERNAME: usuario06 * NOMBRE: Sofia Sánchez * EMAIL: sofia6@gmail.com * FECHA NACIMIENTO: 1993-01-20 * CIUDAD: Granada * FECHA REGISTRO: 2020-11-17
=====
* USERNAME: usuario07 * NOMBRE: Miguel Torres * EMAIL: miguel7@hotmail.com * FECHA NACIMIENTO: 1999-04-17 * CIUDAD: Alicante * FECHA REGISTRO: 2021-03-28
=====
* USERNAME: usuario02 * NOMBRE: Laura Gómez * EMAIL: laura2@hotmail.com * FECHA NACIMIENTO: 1988-11-22 * CIUDAD: Sevilla * FECHA REGISTRO: 2022-12-13
=====
* USERNAME: robertskrr * NOMBRE: Robert Esquerre * EMAIL: robertskrr@gmail.com * FECHA NACIMIENTO: 2000-12-03 * CIUDAD: Sevilla * FECHA REGISTRO: 2023-04-24
=====
```



- 3. Buscar usuario** → Según el **username** a buscar, la app **accede a la base de datos** para comprobar si hay alguna coincidencia y, en caso positivo, te muestra la información de ese usuario.

```

🔍 BÚSQUEDA DE USUARIO 🔍
★ Introduce el username a buscar: pruebaManual
★ !USUARIO ENCONTRADO! ★
!! USUARIO PRUEBAMANUAL !!
◆ Nombre: prueba de manual
◆ Email: manual@manual.com
◆ Fecha Nacimiento: 2000-12-12
◆ Ciudad: Sevilla
◆ Fecha Registro: 2025-05-21

```

- 4. Modificar tu usuario** → Al introducir tu username y contraseña, accedes a la **modificación de todos los datos** de tu usuario (excepto fecha de registro), también **puedes presionar ENTER** en el registro que no quieras modificar.

```

🔒 MODIFICACIÓN DE USUARIO 🔒
★ Introduce el username del usuario a modificar: pruebaManual
★ Introduce la contraseña actual para poder modificar: manual123
✅ USUARIO AUTENTICADO ✅
★ INTRODUCE LOS DATOS A MODIFICAR (O INTRO SI NO DESEA MODIFICAR EL CAMPO) ★
★ Nuevo username: manualNew
★ Nueva contraseña:
★ Nuevo nombre completo: manual nuevo
★ Nuevo email:
★ Nueva fecha de nacimiento (yyyy-MM-dd):
★ Nueva ciudad: Ciudad Manual
MODIFICANDO USUARIO..... ¡USUARIO MODIFICADO CON ÉXITO!

```

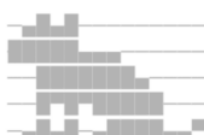
- 5. Eliminar tu usuario** → Al introducir tu username y contraseña, accedes a la opción de **borrar tu usuario**, para hacerlo correctamente la aplicación te pide una **confirmación** introduciendo por **segunda vez tu contraseña**.

```

🔒 ELIMINACIÓN DE USUARIO 🔒
★ Introduce tu username: manualNew
★ Introduce tu contraseña para poder eliminar: manual123
¿ESTÁS SEGURO DE ELIMINAR TU USUARIO? ESTO IMPLICA TAMBIÉN BORRAR A TUS MASCOTAS.
★ Introduce la contraseña nuevamente para confirmar la eliminación: manual123
ELIMINANDO USUARIO..... 🗑️ ¡USUARIO ELIMINADO CON ÉXITO! 🗑️

```

- 6. Acceso al menú de mascotas** → Al **autenticarte** con tu username y contraseña accederás a tu menú de mascotas, el cual te explicaremos en el **siguiente punto**.





- 2. Listar mascotas** → Dependiendo del **tipo de orden** que elijas te devuelve los **datos de las mascotas** registrados con tu usuario en forma de **lista**.

```

🏠 MASCOTAS DE ROBERTSKRR 🏠
  1. Ordenadas por nombre.
  2. Ordenadas por tipo.
  3. Ordenadas por fecha de creación.
* ELIGE EL ORDEN: 1
🏠 MASCOTAS ORDENADAS POR NOMBRE 🏠
=====
* ID: 34 * NOMBRE: Flauta * TIPO: Pajaro * COLOR: CIAN * SEXO: MACHO * FECHA CREACIÓN: 2024-02-12
=====
* ID: 35 * NOMBRE: Qwerty * TIPO: Perro * COLOR: ROJO * SEXO: MACHO * FECHA CREACIÓN: 2024-01-01
=====
* ID: 32 * NOMBRE: Rufo * TIPO: Perro * COLOR: AMARILLO * SEXO: MACHO * FECHA CREACIÓN: 2024-03-11
=====
* ID: 33 * NOMBRE: Yiyi * TIPO: Gato * COLOR: AZUL * SEXO: HEMBRA * FECHA CREACIÓN: 2024-03-23
=====

```

- 3. Buscar mascota** → Según el **nombre de la mascota** a buscar, la app **accede a la base de datos** para comprobar si hay alguna coincidencia y, en caso positivo, te muestra la información de esa mascota. Si hay dos mascotas del mismo usuario que se llamen igual te muestran ambas.

```

🔍 BÚSQUEDA DE MASCOTA 🔍
* Introduce el nombre a buscar: Rufo
* ¡MASCOTA ENCONTRADA! *
=====
* ID: 32 * NOMBRE: Rufo * TIPO: Perro * COLOR: AMARILLO * SEXO: MACHO * FECHA CREACIÓN: 2024-03-11
=====

```

- 4. Crear mascota** → Te pide el **nombre, tipo y sexo** de la mascota, su **color** se genera **aleatoriamente** y el resto de sus datos se asignan de forma automática.

```

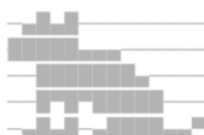
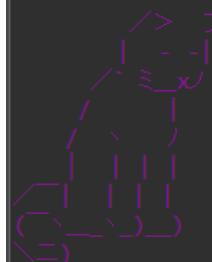
🐾 CREACIÓN DE MASCOTA 🐾
* Nombre de la mascota: manual
* Tipo de mascota (1. Perro, 2. Gato, 3. Pájaro): 2
* Sexo de la mascota (1. Macho, 2. Hembra): 2
CREANDO MASCOTA 'manual'..... ¡MASCOTA CREADA CON ÉXITO!

```

```

*** MASCOTA DE 'ROBERTSKRR' ***
* NOMBRE: MANUAL
* TIPO: Gato
* HEMBRA
* EDAD: 0 AÑOS
* NUTRICIÓN: ██████████
* LIMPIEZA: ██████████
* FELICIDAD: ██████████

```



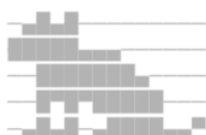


- 5. Eliminar mascota** → Te muestra tus mascotas y, al seleccionar el ID de la mascota que quieres borrar, la aplicación te pide una **confirmación** introduciendo por **segunda vez tu contraseña**.

```

🐾 ELIMINACIÓN DE MASCOTA 🐾
* Mostrando a tus mascotas...
=====
* ID: 35 * NOMBRE: Qwerty * TIPO: Perro * COLOR: ROJO * SEXO: MACHO * FECHA CREACIÓN: 2024-01-01
=====
* ID: 34 * NOMBRE: Flauta * TIPO: Pajaro * COLOR: CIAN * SEXO: MACHO * FECHA CREACIÓN: 2024-02-12
=====
* ID: 32 * NOMBRE: Rufo * TIPO: Perro * COLOR: AMARILLO * SEXO: MACHO * FECHA CREACIÓN: 2024-03-11
=====
* ID: 33 * NOMBRE: Yiyi * TIPO: Gato * COLOR: AZUL * SEXO: HEMBRA * FECHA CREACIÓN: 2024-03-23
=====
* ID: 49 * NOMBRE: manual * TIPO: Gato * COLOR: MORADO * SEXO: HEMBRA * FECHA CREACIÓN: 2025-05-21
=====
* ID de la mascota a eliminar (o -1 para volver atrás): 49
!!¿ESTÁS SEGURO DE ELIMINAR A TU MASCOTA?!!
=====
* ID: 49 * NOMBRE: manual * TIPO: Gato * COLOR: MORADO * SEXO: HEMBRA * FECHA CREACIÓN: 2025-05-21
=====
* Introduce tu contraseña para confirmar la eliminación: 12345
ELIMINANDO MASCOTA..... 🐾 ¡MASCOTA ELIMINADA CON ÉXITO! 🐾

```





## 8. REGISTROS AUTOMÁTICOS

Cada interacción con las mascotas de cada usuario **se registra de forma automática** en fecha/hora en la **base de datos**. Para consultar estos datos sigue los siguientes **pasos**:

1. Abre un script de MySQL y selecciona la base de datos 'Javagotchi'.



2. Copia y pega la siguiente sentencia para cada registro:

- a. Registro de alimentación de la mascota:

```
SELECT *
FROM REGISTRO_ALIMENTACION
WHERE id_mascota = <id de la mascota a comprobar>;
```

- Ejemplo de salida:

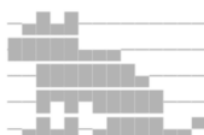
	cod_comida	id_mascota	fecha_hora
▶	C001	32	2025-03-29 16:00:00
	C001	32	2025-04-06 14:20:00
	C001	32	2025-05-18 15:04:25
	C001	32	2025-05-21 14:20:06
	C002	32	2025-04-01 19:30:00
	C002	32	2025-05-18 15:04:42
	C003	32	2025-04-03 08:30:00

- b. Registro de juego de la mascota:

```
SELECT *
FROM REGISTRO_JUEGO
WHERE id_mascota = <id de la mascota a comprobar>;
```

- Ejemplo de salida:

	cod_juego	id_mascota	fecha_hora
▶	J001	32	2025-03-28 09:45:00
	J001	32	2025-05-18 15:04:05
	J001	32	2025-05-21 14:18:29
	J002	32	2025-04-04 11:30:00
	J003	32	2025-04-02 12:50:00
	J003	32	2025-05-21 14:33:04



**c. Registro de limpieza de la mascota:**

```
SELECT *  
FROM REGISTRO_LIMPIEZA  
WHERE id_mascota = <id de la mascota a comprobar>;
```

- Ejemplo de salida:

	id_mascota	fecha_hora	felicidad_aportada
▶	32	2025-03-28 11:30:00	2
	32	2025-04-02 13:00:00	2
	32	2025-04-04 12:30:00	2
	32	2025-05-18 15:04:31	3
	32	2025-05-21 14:20:23	2

