



Manual de Configuración e Instalación

## **Contenido:**

### 1. Requisitos previos

#### 1.1 Instalación de JDK

#### 1.2 Instalación de NetBeans

### 2. JavaFX

### 3. Acerca del juego

### 4. Modelos

### 5. Lógica del juego

### 6. Diagramas

### 7. Clases principales



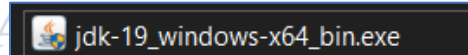
## < 1. Requisitos previos >

Para la correcta utilización del **Juego UNO** se necesita instalar *Java* y después el Entorno de Desarrollo Integrado (IDE - Integrated Development Environment) *NetBeans*. Se utilizaron las versiones 19.0.1 y 16.0, respectivamente, para la implementación de Juego UNO. Se asume que la instalación y ejecución de GH se hará en una computadora con el Sistema Operativo Windows (se utilizó Windows 10, 64 bits).

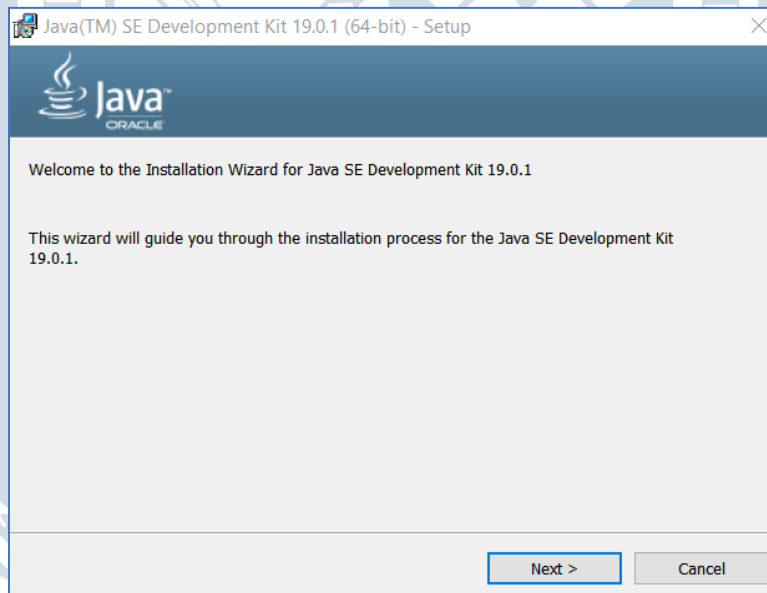
Los archivos de instalación se ofrecen junto con esta documentación y el código de la aplicación.

### 1.1 Instalación de Java Developer Kit (JDK) 19.0.1

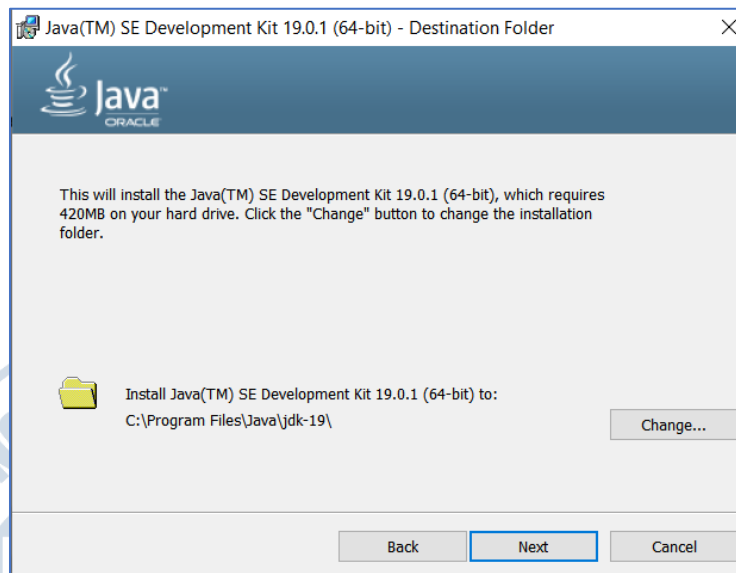
1. Ejecutar el archivo de instalación, haciendo doble click.



2. Hacer click en **Next**, para iniciar el proceso de instalación.



3. Elegir el directorio de instalación. Se sugiere el directorio por default.

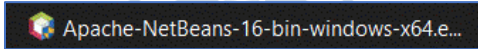


4. El proceso de instalación toma unos minutos, dependiendo de las características del sistema de cómputo. Una vez que se termina dicho proceso, hacer click en **Close**.

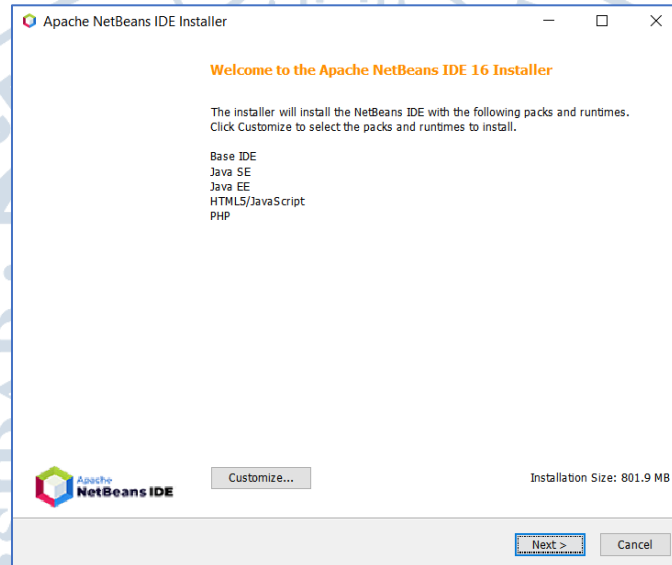


## 1.2 Instalación de NetBeans 16.0

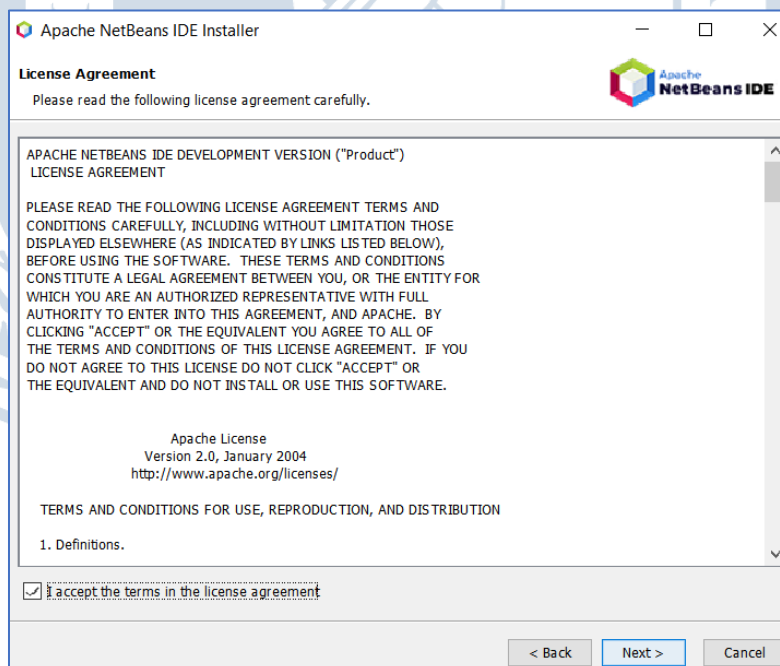
1. Ejecutar el archivo de instalación, haciendo doble click.



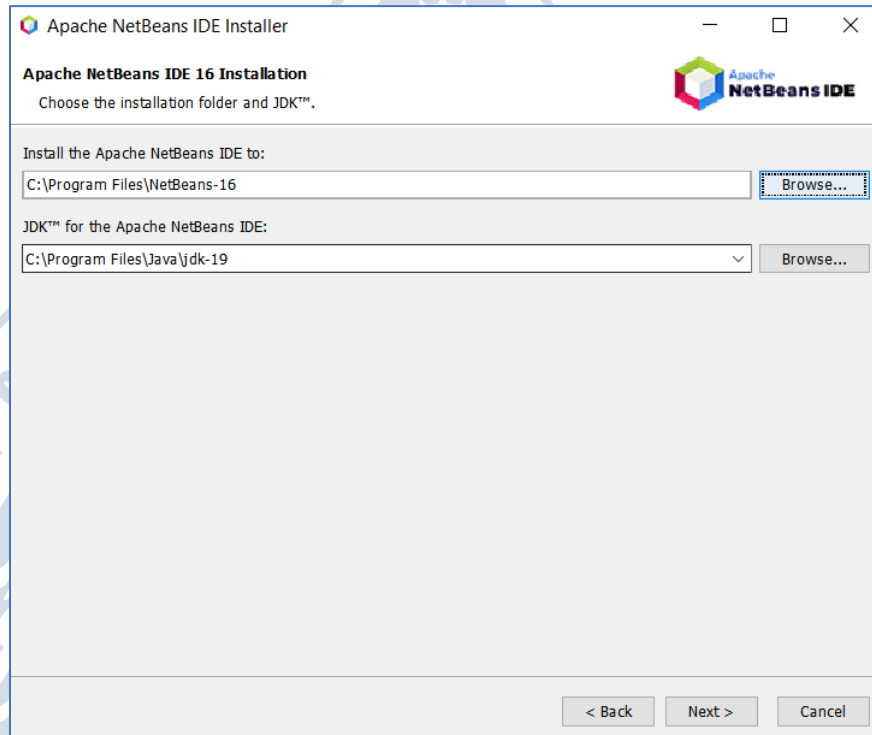
2. Hacer click en **Next**, para iniciar el proceso de instalación.



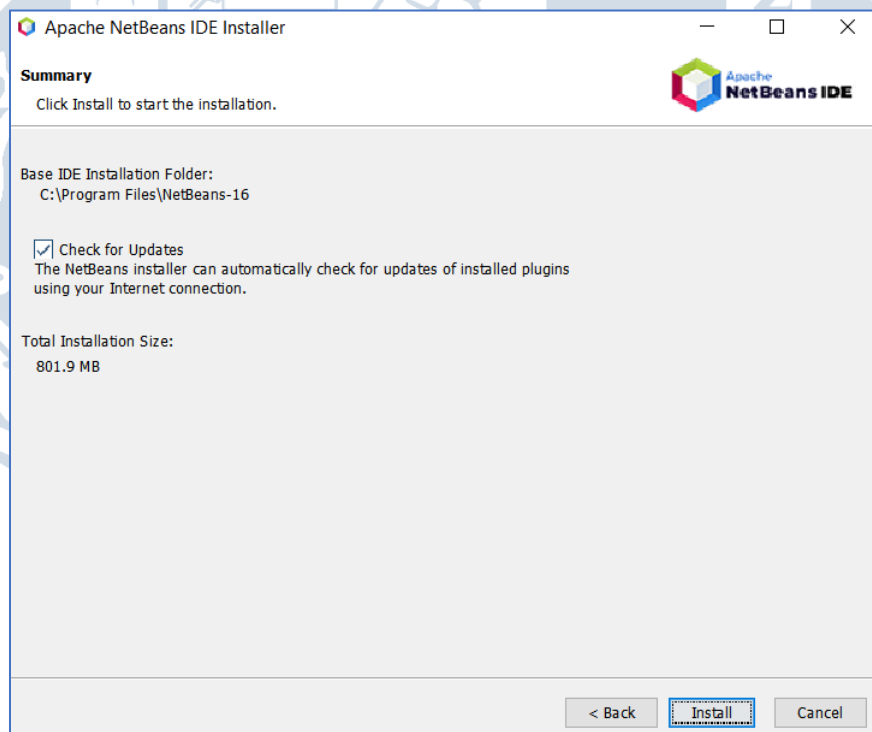
3. Aceptar la licencia y hacer click en **Next**.



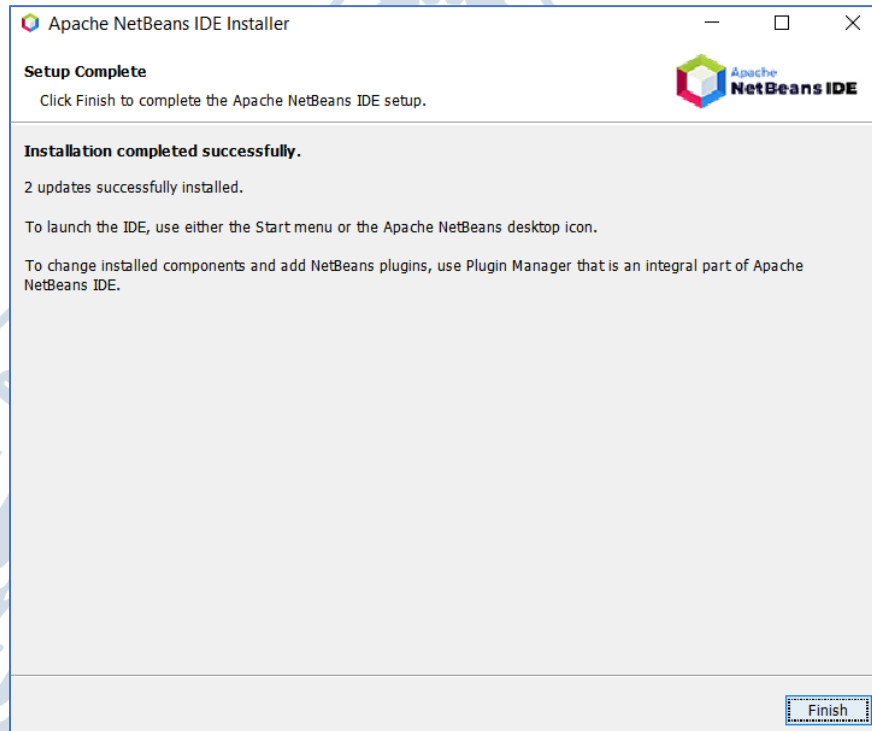
4. Elegir el directorio de instalación. Se sugiere el directorio por default, así como el directorio donde se instaló Java. Hacer click en **Next**.



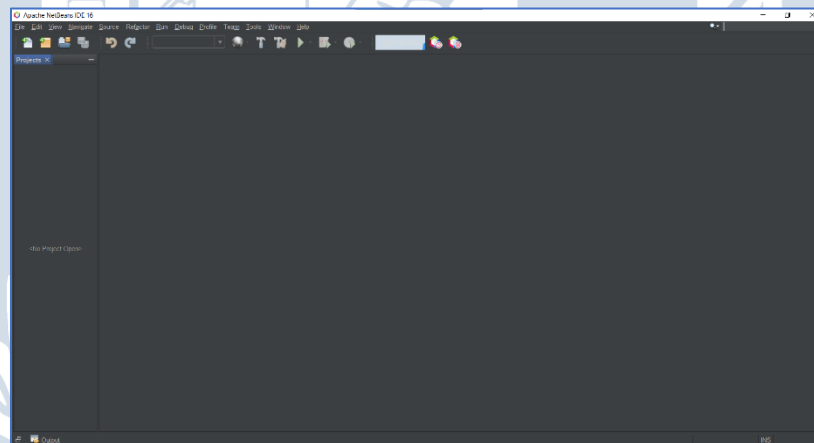
5. Hacer click en **Install**.



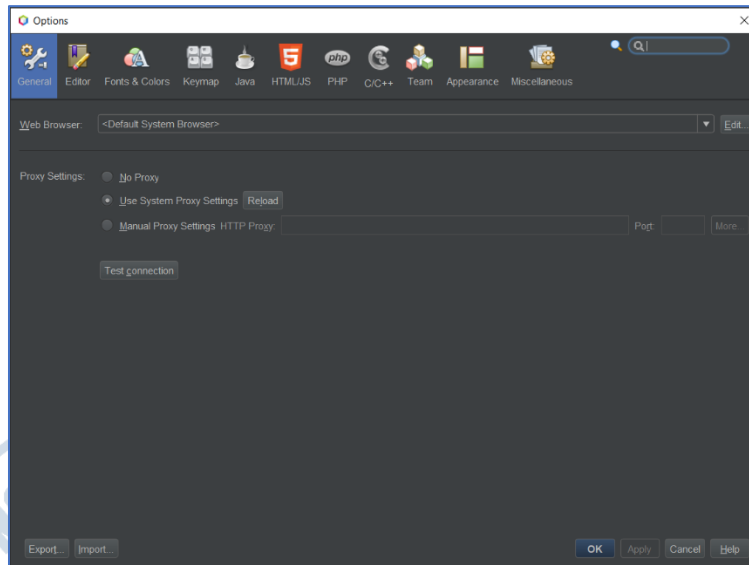
6. El proceso de instalación toma unos minutos, dependiendo de las características del sistema de cómputo. Una vez que se termina dicho proceso, hacer click en **Finish**.



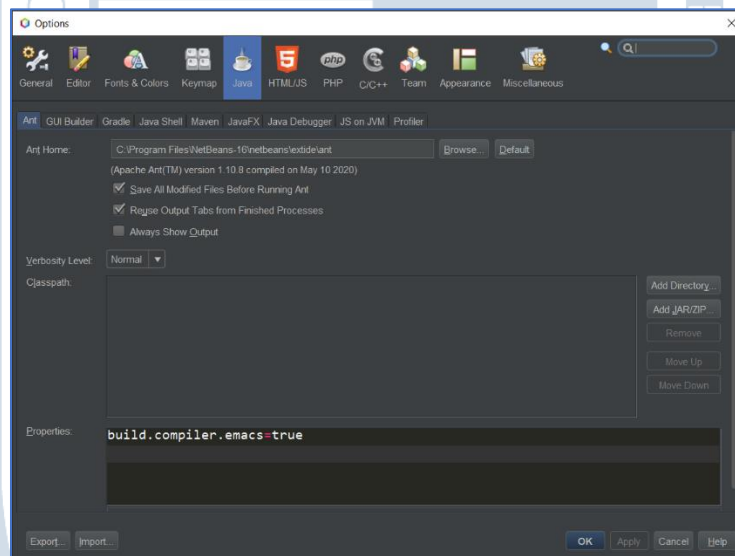
7. Abrir el IDE.



8. Habilitar los módulos. Hacer click en el menú **Tools/Options**.



9. Hacer click en **Java**, y después en **Ok**.



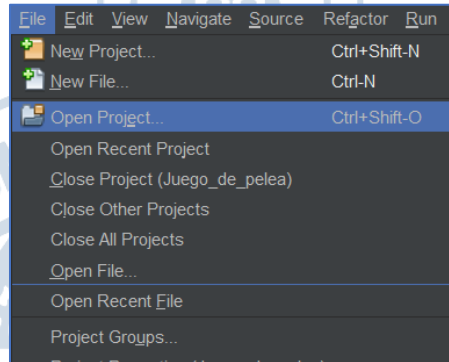
10. NetBeans está listo para importar JuegoUNO.



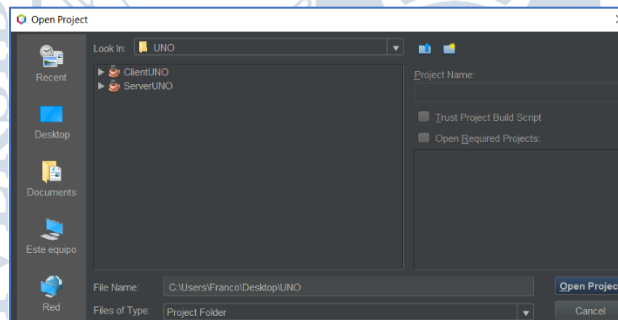
## < 2. JavaFX >

Primero importaremos JuegoUNO al IDE, y después haremos las configuraciones necesarias para que pueda ser ejecutado.

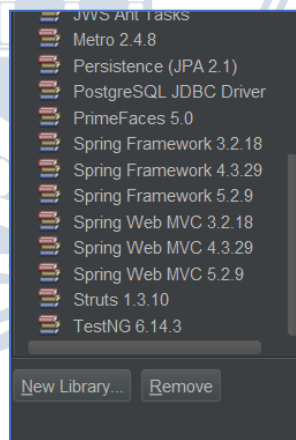
1. Importar el proyecto. Hacer click en el menú **File/Open Project**.



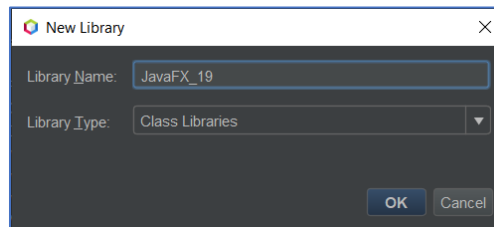
2. Seleccionar el directorio donde se encuentra el proyecto. NetBeans lo identificará con un ícono en particular. Hacer click en **Open Project**.



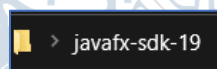
3. Crear una nueva biblioteca, y llamarla **JavaFX\_19**. Seleccionar el menú **Tools/Libraries**, hacer click en **New Library**.



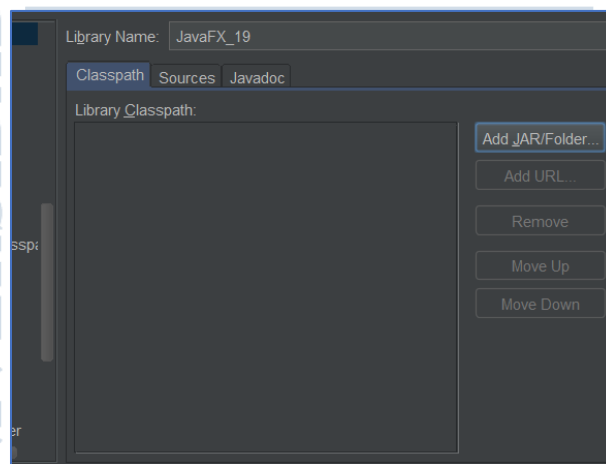
4. Nombrar JavaFX\_19 a la nueva biblioteca.



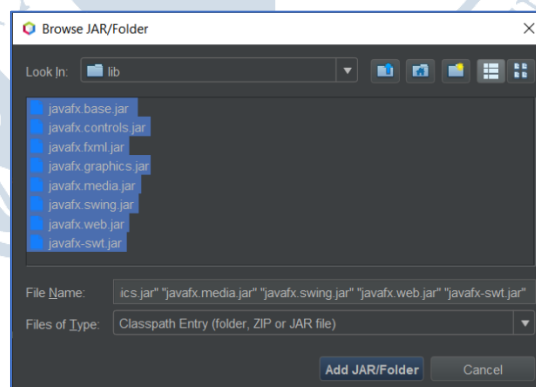
5. Añadiremos los archivos jar que se encuentran en la carpeta:



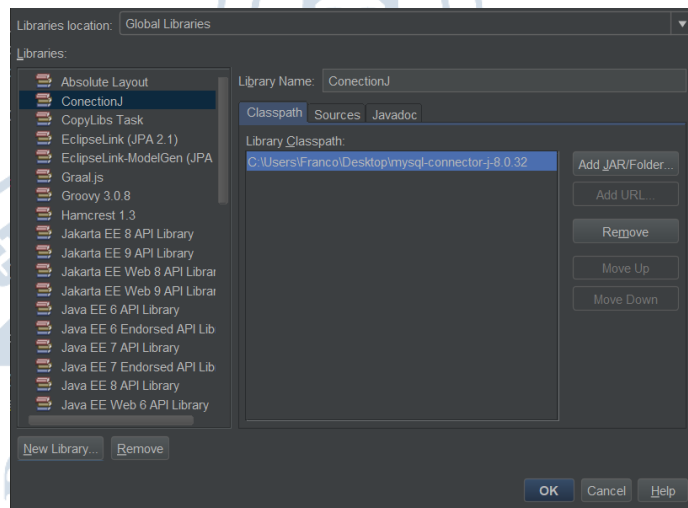
6. Hacer click en **Add JAR/Folder...**



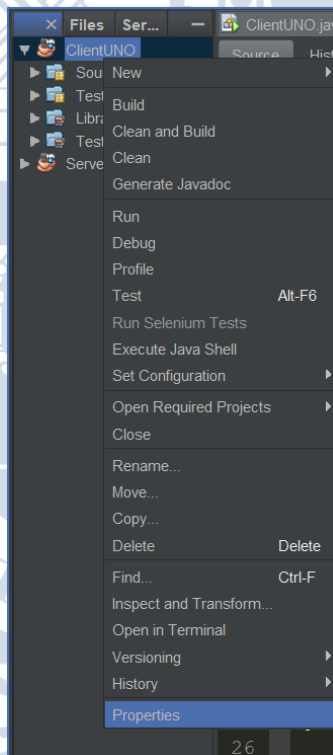
7. Dentro del directorio descrita en el punto 5, buscar dentro del directorio **lib**. Seleccionar todos los archivos .jar Hacer click en **Add JAR/Folder**.



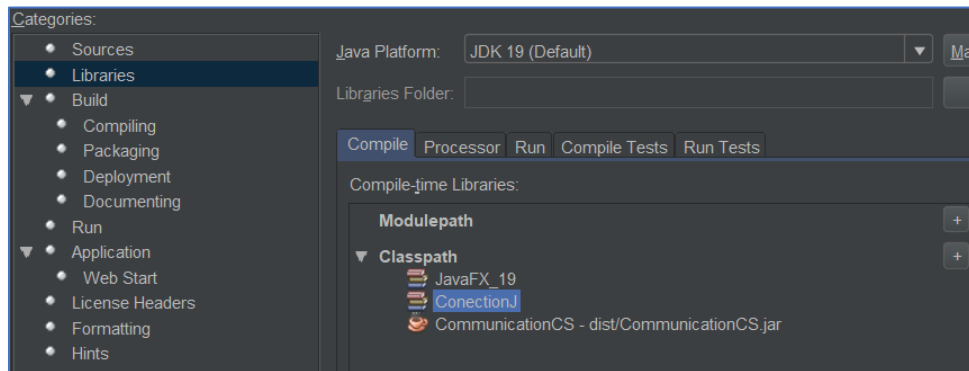
8. Repetir los pasos del punto 3 al 7 para crear otra librería llamada **ConnectionJ** con el archivo:



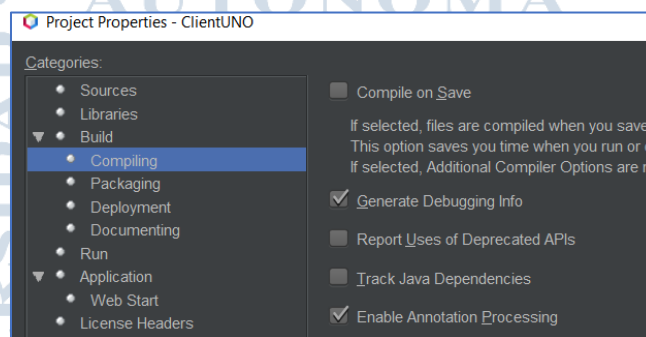
8. En el proyecto **ClientUNO**, hacer click con el botón derecho y seleccionar **Properties**.



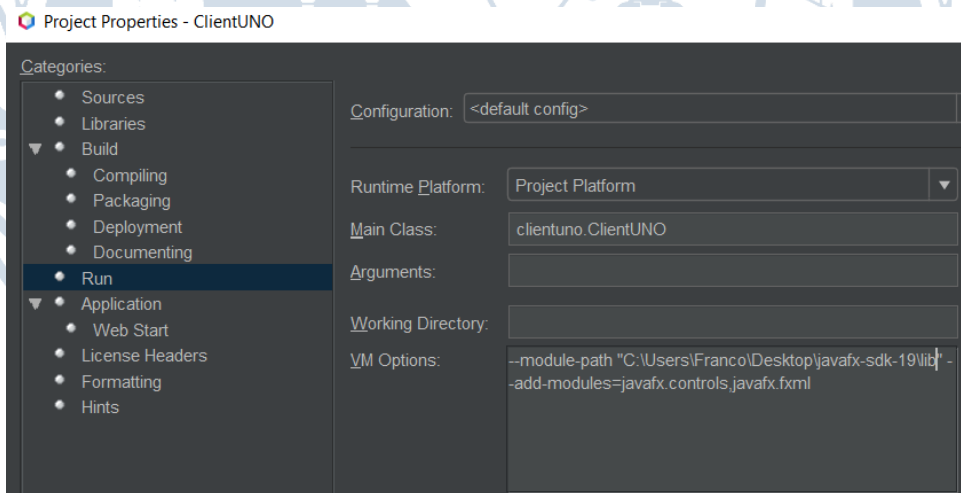
9. Seleccionar **Libraries**, asegurarse que la plataforma **JDK 19** esté seleccionada y añadir en **Classpath** la biblioteca **JavaFX\_19** y **ConnectionJ** creadas anteriormente.



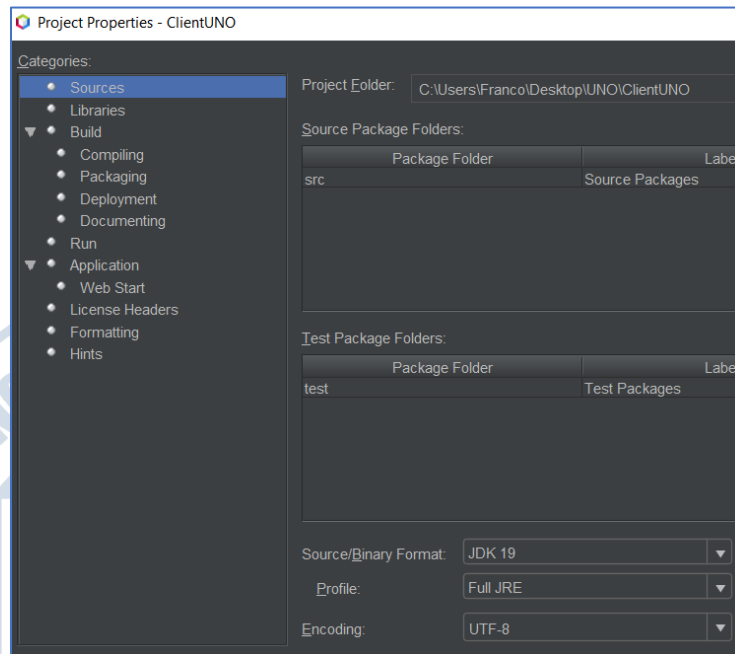
10. Nuevamente en las **Properties** del proyecto, en la sección **Build/Compiling**, asegurarse que la opción **Compile on Save (...)** no está marcada.



11. En **Properties/Run** del proyecto, asegurarse que la ruta donde están los archivos .jar añadidos en el punto 7 estén en el comando.



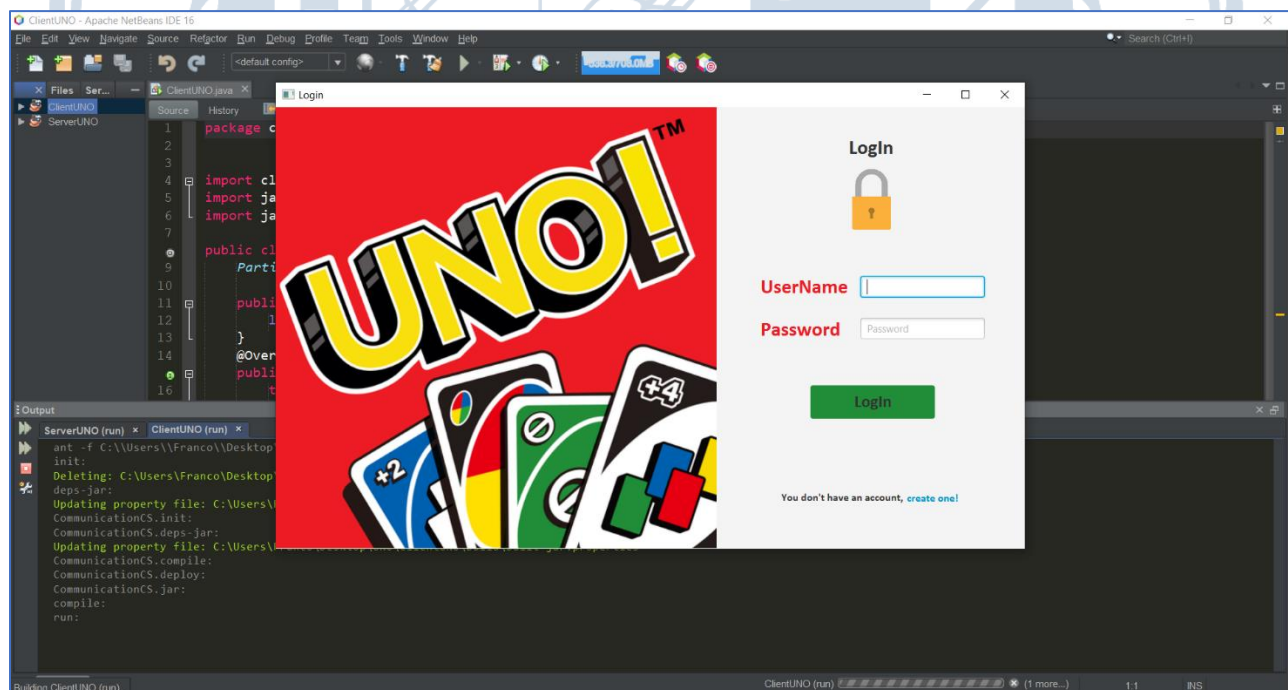
12. Por último, revisar en **Properties/Sources** que **JDK 19** esté seleccionada en **Source/Binary Format**.



13. Repetir los puntos del 8 al 12 ahora con el proyecto **ServerUNO**

13. Grabar todos los cambios y ejecutar el proyecto.

14. **JuegoUNO** deberá estar en ejecución.



### < 3. Acerca del juego >

Este proyecto fue creado mediante el software javaFX, con la ayuda de la aplicación Scene Builder.

#### Reglas:

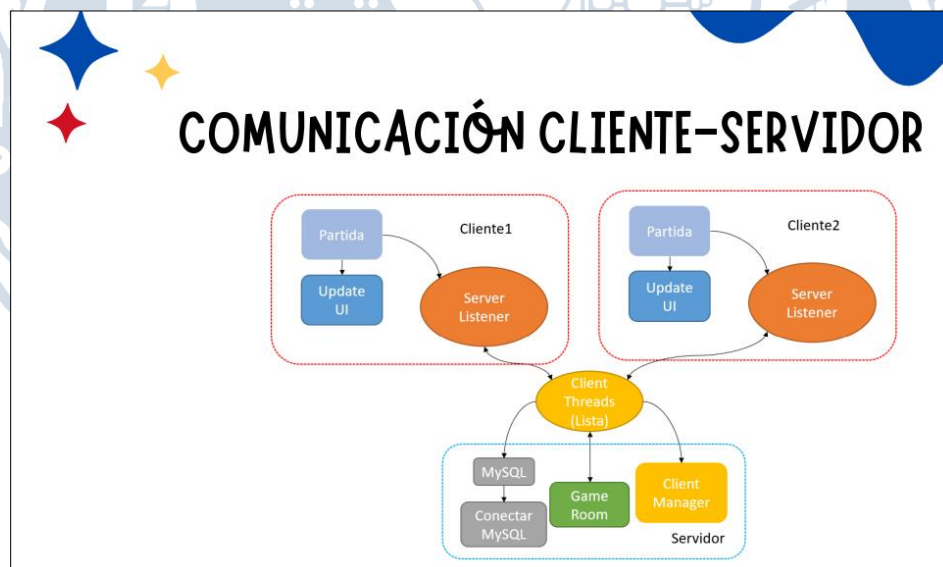
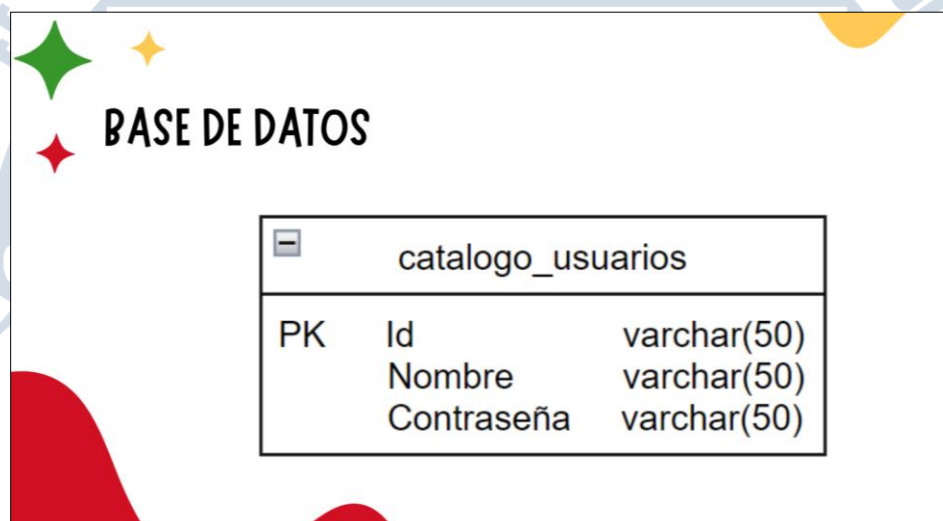
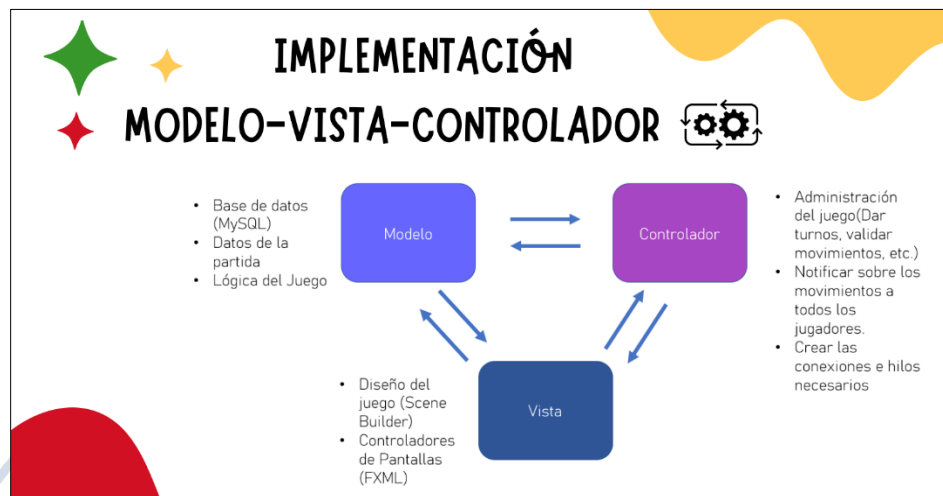
- El UNO es un juego de cartas y tiene unas reglas sencillas de aprender. Las reglas del UNO fundamentales para entender el juego es que pueden participar varios jugadores, es decir, de 2 hasta 4.
- Cada uno de ellos tiene el objetivo de quitarse las cartas que le tocan en el reparto inicial antes que el resto.
- Para conseguirlo, cada jugador va dejando sus cartas sobre la mesa.
- El juego finalizara en el momento en el que un jugador se quede sin cartas.

#### Información de la interfaz:





## < 4. Modelos >



## < 5. Lógica del juego >

### 1. Información Compartida Con Jugadores:

- Turnos Específicos
- Turno actual
- Bandera: tienes un castigo
- 7 cartas a cada usuario
- Carta mesa

\* CASO BASE: Deshabilita botones.

### 2. Turno de usuario:

- Caso 1: No es tu turno
  - Caso Bas
- Caso 2: Es su turno
  - Caso 2.1: No tiene castigo
  - Se habilita:
    - ✓ Cartas jugador
    - ✓ Boton tomar carta
  - Caso 2.2 : tienes efecto
    - ✓ Manda mensaje al servidor
    - ✓ Caso Base

### 3. Respuesta del servidor:

Jugador presiona "tomar carta":

- Da una carta de la comida y se la envía

Jugador presiona carta de su mazo

- Caso 1: No es compatible con la carta de la mesa
  - Informa al jugador
- Caso 2: Es compatible con la carta de la mesa
  - Carta Mesa= carta
  - Caso 2.1: El jugador ya no tiene cartas
    - ✓ Informa de la victoria
    - ✓ Se acaba el juego
  - Caso 2.2: Carta es especial
    - Siguiente usuario-> bandera = true
    - Cambio color -> habilita botones (Caso Extra)
    - Reversa -> aumento = aumento\*-1
- Aumenta turno, actualiza información

Jugador tiene un castigo acumulado:

- Switch
  - +4-> envían 4 cartas
  - +2 -> envían 2 cartas
  - Pasar->Caso Base
- Aumenta turno

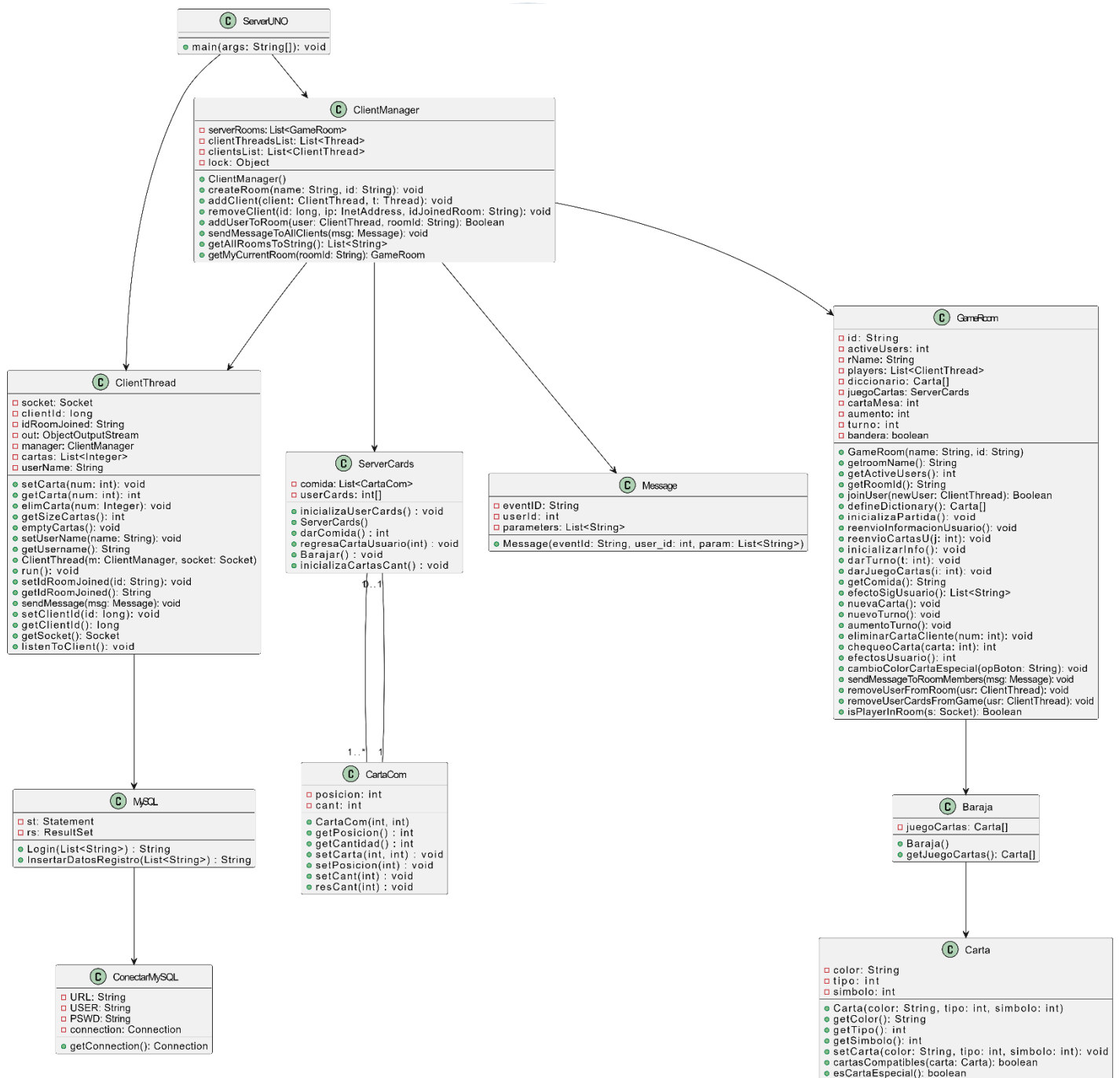


## < 6. Diagramas >

### 1. Cliente:



## 2. Servidor:



## < 7 Clases principales >

### Cliente:

#### **Partida:**

- Esta clase controla toda la aplicación.
- Mantiene los datos de usuario y conexión.
- Tiene métodos para actualizar la interfaz (Están referenciados en UpdateUI)
- En general, la clase Partida actúa como un controlador central que coordina la interacción entre los controladores, la interfaz de usuario y la comunicación con el servidor para el juego UNO.

#### **ServerConnection:**

- Facilita la comunicación con el servidor en el juego UNO al establecer la conexión, enviar mensajes al servidor y recibir mensajes del servidor a través de un hilo de escucha dedicado.

#### **ServerListener:**

- Es responsable de recibir y procesar los mensajes del servidor en el juego UNO. Su objetivo principal es mantener la comunicación bidireccional entre el cliente y el servidor, actualizando la interfaz de usuario según las respuestas y eventos del servidor.

#### **StageData:**

- Sirve para almacenar los datos de la escena antes de que se destruya
- Almacena el socket (conexión al servidor)
- En general sirve como un contenedor para almacenar y acceder a los datos relevantes de la escena actual del juego UNO. Al guardar estos datos, se asegura que la información importante no se pierda cuando se destruye la escena y se proporciona un medio para restaurar el estado del juego cuando sea necesario.

#### **UpdateUI:**

- Permite ejecutar acciones en el hilo de la interfaz de usuario desde un hilo de fondo, garantizando que la interfaz se actualice correctamente y permanezca receptiva.

### Servidor:

#### **ClientThread:**

- Es responsable de manejar las solicitudes individuales de los clientes en el servidor, comunicándose con ellos a través de sockets y realizando acciones según los mensajes recibidos.

#### **ClientManager:**

- Es responsable de administrar los clientes y las salas de juego en el servidor, controlando la conexión, desconexión y comunicación con los clientes, así como la creación y gestión de salas de juego.

## MySQL

- Se encarga de realizar operaciones relacionadas con una base de datos MySQL.
- Proporciona métodos para autenticar usuarios y agregar nuevos usuarios a una base de datos MySQL.

