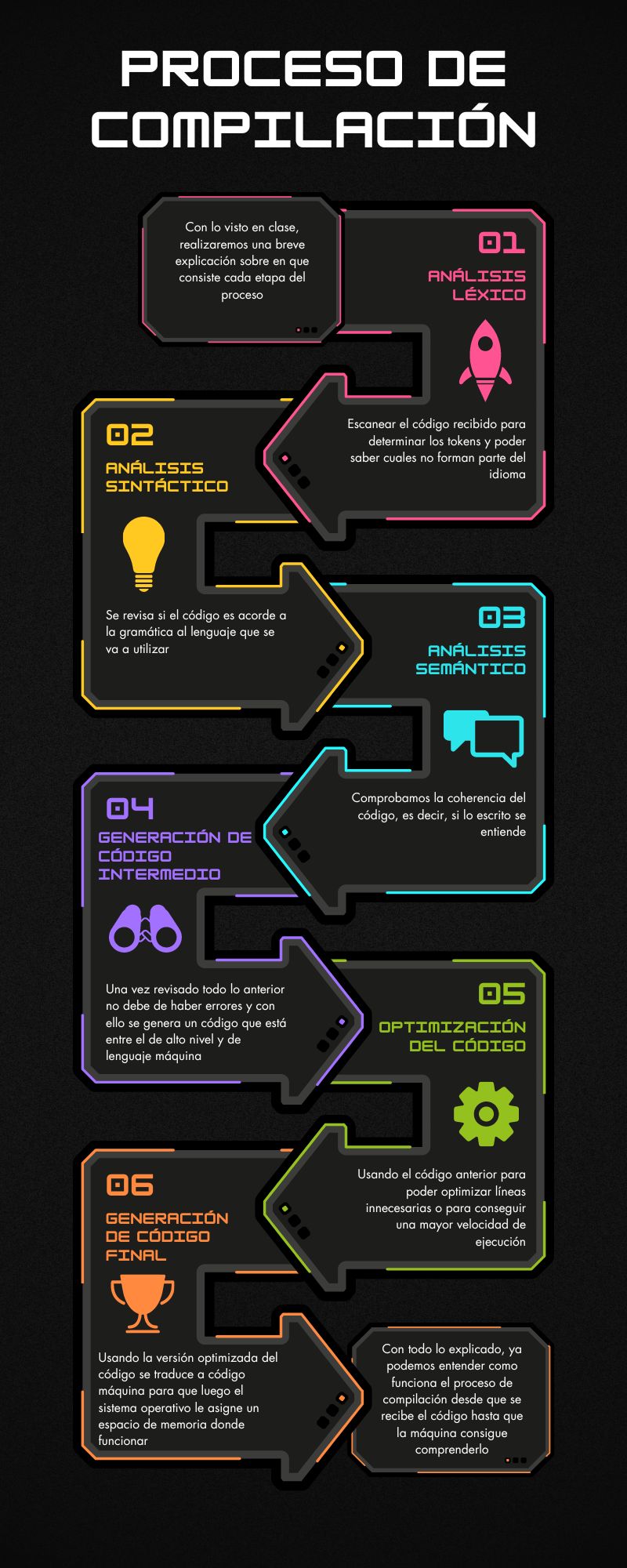
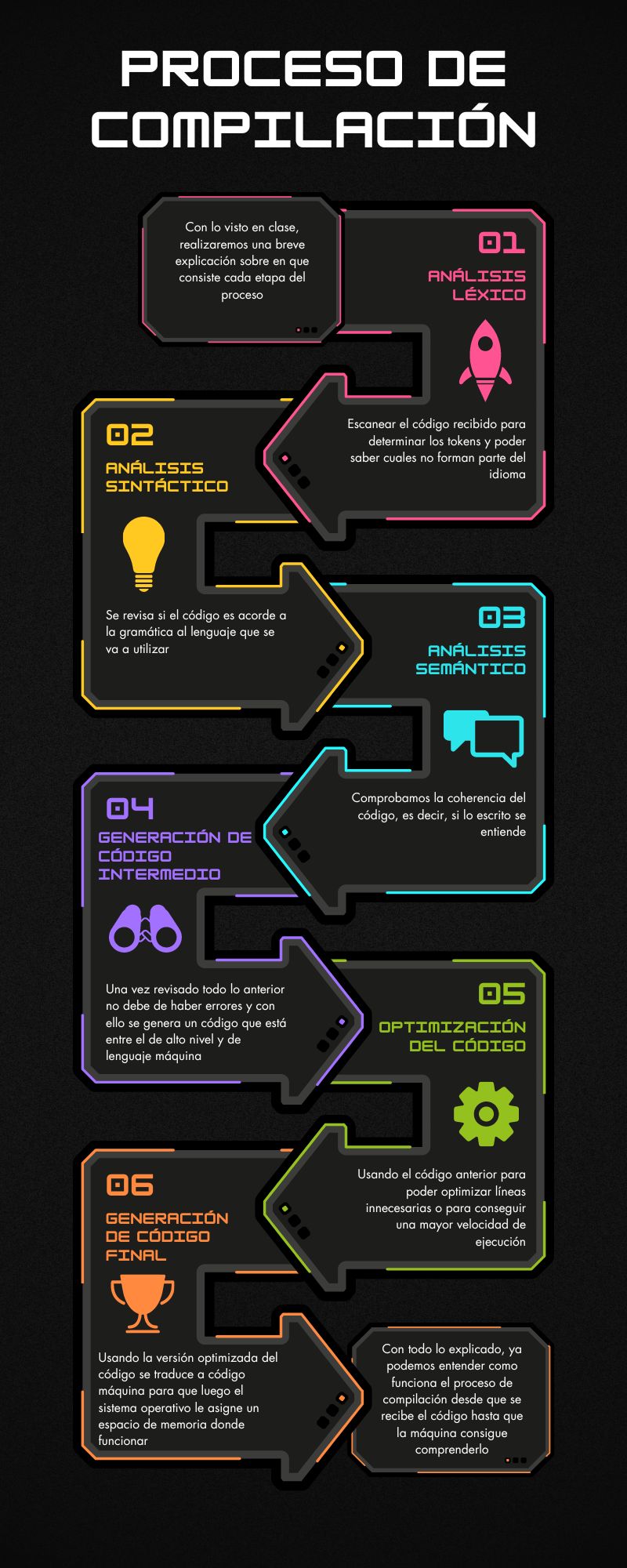
**Actividad de Investigación Autónoma UD1**

1. **Crea una infografía o mural donde se represente el proceso de compilación que hemos estudiado.**

****

****

1. **Crea una nube de palabras de conceptos de todos los lenguajes de programación que conozcas. Incluye al menos ocho.**

****

1. **Menciona dos lenguajes de cada uno, según los paradigmas de programación.**

* Imperativo → C, Python
* Declarativo → SQL, Prolog
* Procedimental → C#, C++
* Orientado a Objetos → Java, C#
* Funcional → Haskell, Scala
* Lógico → Prolog, Mercury

1. **¿Cómo interpretar la siguiente frase? “El software es un elemento que no se estropea, pero que si se degrada”.**

En cuanto a que no se estropea es cierto ya que no es un elemento físico, si no que existe dentro de nuestro sistema y como tal necesita mantenimiento para seguir funcionando pero no pasa igual con la memoria RAM, procesadores, etc … que sí pueden fallar y dejar de funcionar por lo tanto se estropean. Lo que lo hace diferente es que el software al no estropear igual que los componentes físicos lo que si le ocurre es que con el paso del tiempo deja de ser útil ya que conforme pasa el tiempo van a ir sacando versiones mejoradas del que ya poseemos y por ello este se *degrada* deja de estar a la última y se vuelve obsoleto para su funcionamiento en programas más modernos.

1. **INVESTIGA. ¿Qué dos tipos de licencia existen de software? Explícalas brevemente.**

* **Libre o de Código Abierto**

Son licencias gratuitas, es decir que al no pagar por ellas tenemos acceso libre incluso al código fuente para poder modificarlo y personalizar a nuestro gusto. Como ejemplo tenemos a **FireFox, Chrome, CentOs**

* **Propietario**

Al contrario, estas licencias son de pago y por ello su uso es restringido a quien lo compre, además no son de código abierto, con lo que su personalización está limitada a lo que el creador deje al usuario. Todo esto viene seguido de un acuerdo de licencia que una vez que se compra se está aceptando para poder usarlo.

Como ejemplo podemos mencionar a **Microsoft, Adobe,** etc…

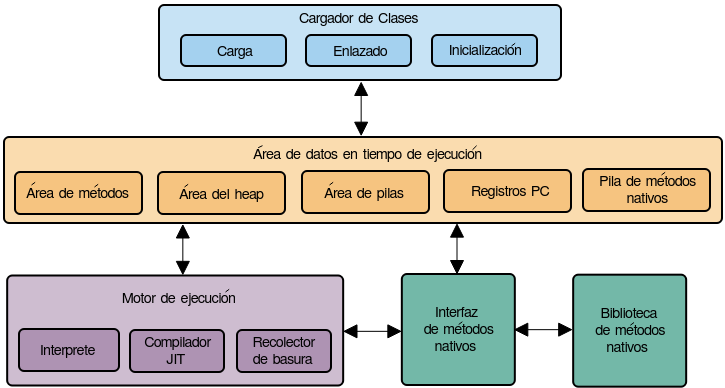
1. **INVESTIGA. Acerca de la máquina virtual de Java. Céntrate en sus características, componentes y funcionamiento. Incluye un esquema de funcionamiento.**

Es un programa que permite ejecutar otros programas escritos en java. Se puede añadir tanto en sistemas operativos como en cualquiera de los navegadores que usamos. La JVM es capaz de interpretar y ejecutar instrucciones en el lenguaje Java, actúa como una capa intermedia que interpreta y ejecuta el código compilado en la máquina física.

Con esto podremos ejecutar nuestros programas y códigos desde nuestro navegador.

Básicamente, la JVM toma el bytecode y lo convierte en instrucciones de lenguaje máquina que pueden ser comprendidas por la CPU del sistema​​.

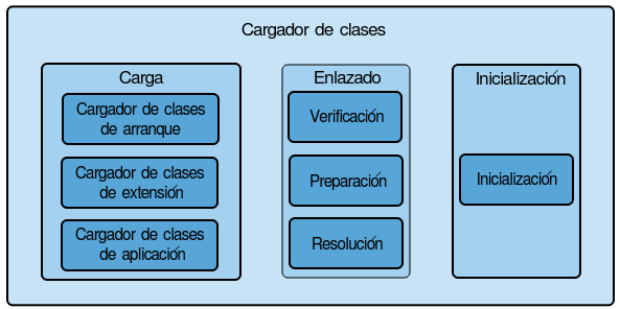
**Componentes y funcionamiento →** consta de 3 partes

****

1. **Cargador de clases**

Luego de almacenar el código en un ***fichero.java***, se obtiene **bytecode** almacenado en un ***fichero .class***. La primera clase que se carga en memoria es el método **main() .**

Este proceso tiene tres fases: carga, enlace, e inicialización (initialization). Que explicaremos a continuación.



* **Carga:**

Se encarga de encontrar, verificar y preparar las clases en tiempo de ejecución, lo que incluye las clases definidas por el usuario y las bibliotecas estándar de Java.

Existen 3 cargadores de clase:

* **Cargador de clases de arranque (Bootstrap Class Loader)**
* **Cargador de clases de extensión (Extension Class Loader)**
* **Cargador de clases de aplicación (Application Class Loader)**
* **Enlazado:** tiene varios pasos a realizar

Verificación: como indica el nombre se encarga de la corrección del archivo.class

Preparación: la JVM asigna memoria para los campos de una clase o interfaz, se inician con valor por defecto

* **Inicialización**

Consiste en ejecutar el método de inicialización de una clase o interfaz. Es el paso final del proceso de carga de clases.

1. **Área de datos en tiempo de ejecución**

Compuesta de 5 elementos:



* **Área de métodos**

Todos los datos a nivel de clase son almacenados en esta área.

* **Área del montículo**

Todos los objetos y sus correspondientes variables de instancia son almacenados en esta zona.

* **Area de pilas**

Las variables locales, resultados y llamadas a métodos se almacenan en esta zona de memoria conocida como *pila.*

* **Registros de contador de programa (PC)**

Cada hilo tiene su propio registro → PC que guarda la dirección de la instrucción de la JVM ejecutándose en ese momento. Una vez ejecutada dicha instrucción, el registro PC es actualizado con la dirección de la próxima instrucción.

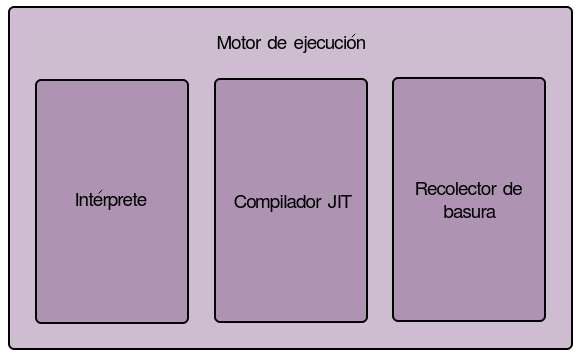
* **Pilas para métodos nativos**

La JVM puede hacer uso de pilas para que soporten métodos escritos en lenguajes diferentes a Java.

1. **Motor de ejecución**

Una vez que el ***bytecode*** se ha cargado en memoria y la información necesaria está disponible en el ***área de datos de tiempo de ejecución,*** el siguiente paso es ejecutar el programa. El motor de ejecución es quien se encarga de este proceso.

Sin embargo, antes de ejecutar el programa, hay que traducir el bytecode a instrucciones del lenguaje máquina haciendo uso de **JIT**.



* **Intérprete**

Se encarga de leer y ejecutar las instrucciones del bytecode línea a línea. Cómo está leyendo línea a línea este proceso se vuelve extremadamente lento

* **Compilador JIT (Just In Time)**

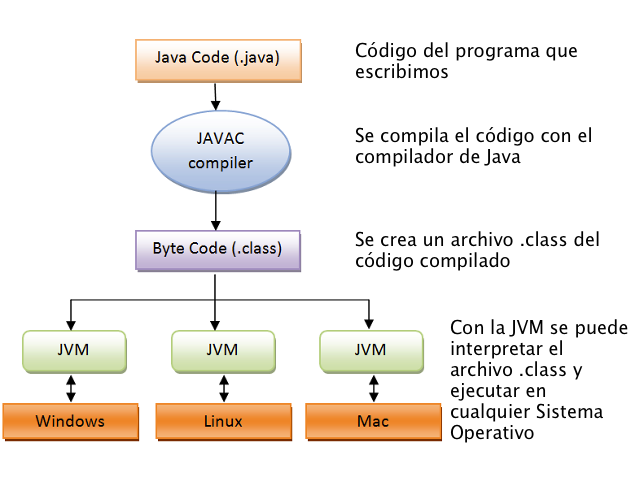
Con él se consigue solventar el problema del intérprete mejorando el rendimiento compilando códigos de bytes en código de máquina nativo en tiempo de ejecución.

* **Recolector de basura:**

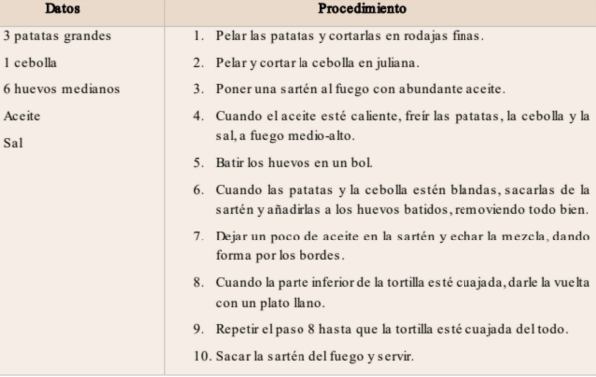
Es el proceso de recuperar automáticamente, en tiempo de ejecución, la memoria ocupada por objetos que ya no van a ser utilizados

Con esto se consigue que la JVM sea eficiente gestionando la memoria ya que la va liberando para dejar espacio a otras cosas que sí sean más importantes

**Esquema del funcionamiento:**

****

1. **Crea un programa en pseudocódigo para realizar esta receta.**

****

**INICIO**

**// Declarar variables**

**Patata\_grande = 3**

**cebolla = 1**

**Huevo\_mediano = 6**

**aceite**

**Sal**

**Leer Patatas\_grandes**

**PelaryCortar → usar patatas\_grandes**

**PelaryCortar → usar cebolla**

**Freír → coger sartén con patatas\_grandes, cebolla y sal**

**Batir → usar huevos**

**SI freír → patatas y cebollas blandas**

**ENTONCES añadir → batir (huevos)**

**SI freír → parte inferior cuajada**

**ENTONCES darlavuelta (a el conjunto de freír)**

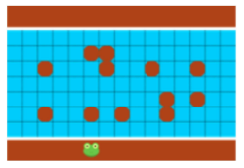
**REPETIR darlavuelta hasta → freír listo**

**Escribir “Tortilla lista”**

**FIN**

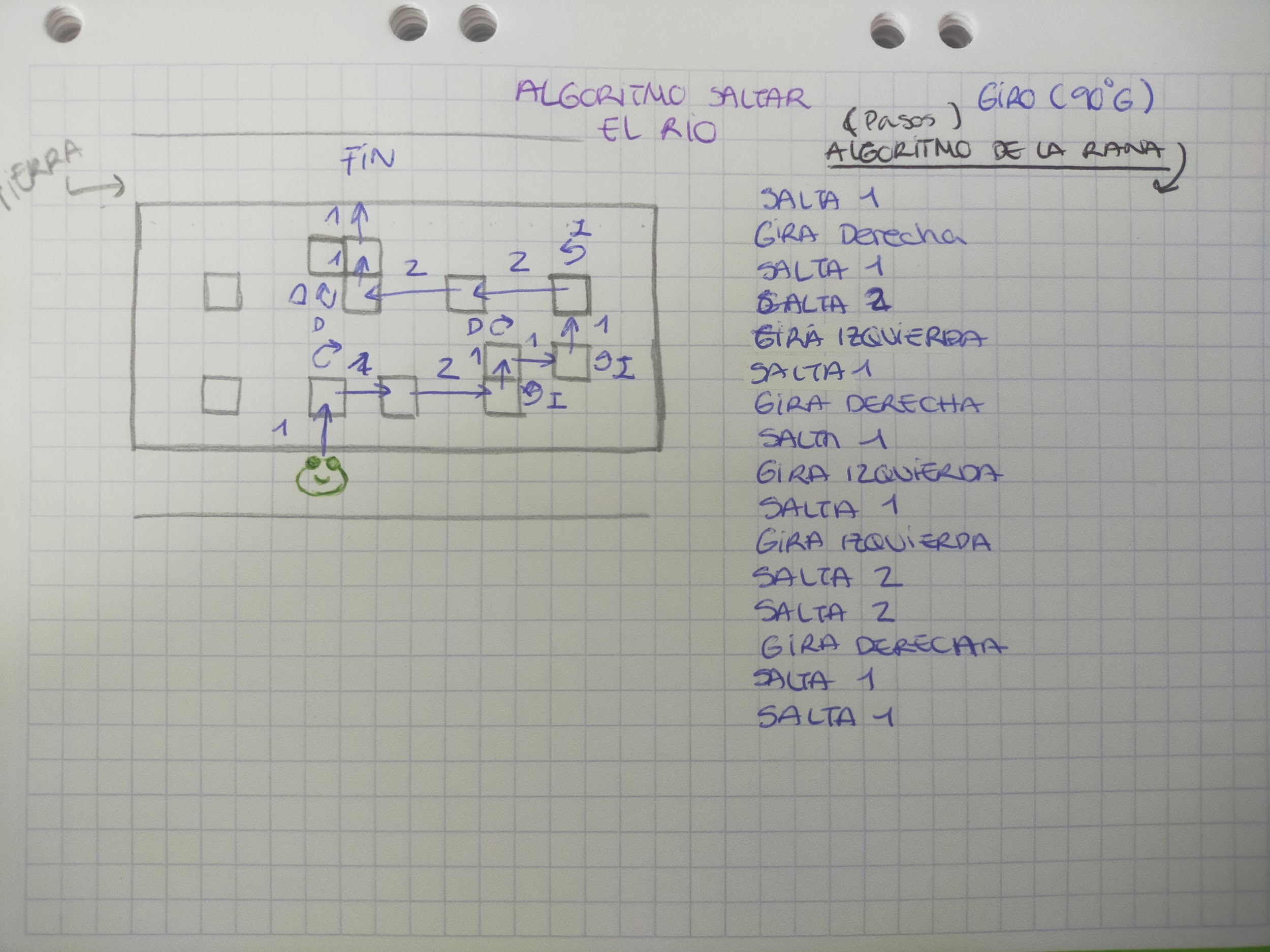
1. **Crea el algoritmo para saltar el río. Una rana se encuentra en la orilla de un río y quiere cruzar a la orilla opuesta saltando sobre las piedras. Suponiendo que las operaciones que puede realizar la rana son las siguientes:**

* **Saltar 1, 2 o 3 posiciones.**
* **Girar 90º a la izquierda o derecha**

****

**En un papel, indica el algoritmo o la serie de pasos que debería seguir para llegar al otro extremo. Cuando hayas terminado, descubre la respuesta y comprueba tu resultado.**

Para la solución al problema anterior, en una hoja lo he resuelto indicando los pasos de la rana y los saltos, giros y casillas que avanza



1. **Cuál es la verdadera.**
2. **El HW se compone de los programas, las imágenes, los iconos y los ficheros de configuración. Se complementa con el SW para componer lo que sería un sistema informático. SW refiere a lo tangible y HW a lo intangible. →** Falso
3. **El SW se compone de los programas, imágenes, los iconos y los ficheros de configuración. Se complementa con el HW para componer lo que sería un sistema informático. SW refiere a lo intangible y HW a lo tangible. →** Verdadero
4. **El SW se compone de los programas, imágenes, los iconos y los ficheros de configuración. Se complementa con el HW para componer lo que sería un sistema informático. SW refiere a lo tangible y HW a lo intangible. →** Falso
5. **El HW se compone de los programas, imágenes, los iconos y los ficheros de configuración. El HW se complementa con el SW para componer lo que sería un sistema informático. SW refiere a lo intangible y HW a lo tangible →** Verdadero