|  |  |
| --- | --- |
| **Alumno** | **Jose Manuel Almagro Dominguez** |
| **Asignatura** | **Entornos de desarrollo** |
| **Curso** | **1º DAM** |
| **Año** | **2020-2021** |
| **Título de la práctica** | **Lenguajes y paradigmas de programación** |

# 1. Código espaghetti

## 1.1 Significado

Código espaghetti es una forma despectiva de llamar a los programas de computación que tienen una estructura de control de flujo compleja.

El control de flujo es la forma grafica de representar un algoritmo o proceso

## 1.2 Motivo

Se le denomina por este nombre porque el control de flujo de este tipo de programas de computación se asemejan a un plato de spaghetti.

Ya que se asocia con lenguajes básicos y muy antiguos, donde el flujo era controlado por sentencias de control como goto, y se utilizaban números de línea.

*GOTO es una instuccion muy antigua utilizada en BASIC y otros lenguajes, que se basaba en transferir el control a un punto determinado del control, es decir un salto incondicional dentro de la secuencia de ejecución del código*

# 2. Charles Babbage y Ada Lovelace

## 2.1 Charles Babbage

Charles es un matemático británico nacido en 1791 y que fallecio en 1871 . A este se le considera el padre de la computacion, diseño y desarrollo la maquina analítica, solo la diseñó. Hizo varios intentos de consturir su maquina hasta , inspirada en el telar de Joseph Marie . siendo este progamable mediante tablas perforadas.

## 2.2 Ada Lovelace

Lady Ada Loveace era matemática, es a la que se le considera como la primera programadora del mundo, nació en 1815 y falleció en 1852 . Ella estudió la maquina analítica de Charles Babbage y describe las operaciones aritméticas que realizaban las tablas perforadas. Creando el primer algoritmo informático

# 3. Tabla comparativa de lenguajes de programación

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tabla Comparativa | | | | |
| Lenguaje | Características | Fortalezas | Debilidades | Opinión |
| Python | Es un código abierto | Ahorra tiempo y recursos y fácil de entender |  | No lo he usado |
| Java | Lenguaje orientado a objeto | Es simple, es distribuido, seguro y portable |  | No lo he usado |
| JavaScript | Lenguaje orientado a objeto | Es rápido y simple. Se carga directamente desde el servidor |  | Me parece complciado pero porque no lo he practicado |
| C# | Lenguaje orientado a componentes, creado por microsoft | Es sencillo y es soportado por unity |  | No lo he usado |
| PHP | Lenguaje basado para creación de webs | Facil, se le puede implementar jacilmente utilidades de javascript y sql |  | Es fácil de usar, con muchas posibilidades ya que con css puedes dar cualquier tipo de formato |
| C/C++ | Extensión de C# utilizado para bdd,navegadores,compiladores y videojuegos | Versatilidad a la hora de hacer programas | Es mas complejo | Dificíl de usar pero es muy útil |
| R | Lenguaje pensado para análisis estadístico | Es sencillo |  | No lo he usado |
| SQL | Lenguaje basado a las base de datos | Sencillo de implementar, rápido | Depende de otros lenguajes para sacarle el mayor partido | Es mi lenguaje favorito , no puedo ser objetivo con el |

Información recogida de [AQUI](https://www.yeeply.com/blog/lenguajes-de-programacion-mas-usados/)

# 4. Full-Stack Developer

Un Full Stack Devoleper es un programador que ha demostrado su experiencia, que sabe manejarse en el back end y en el front end. Es decir, saber manejar la parte de software que interactúa con los usuarios y la parte que procesa los datos.

Los datos de búsqueda de empleo me dan problemas al subirlos al archivo asique, se las enviare junto al trabajo.

# 5. Paradigmas de programación

Un paradigma de programación es el modelo a seguir que nos define los pasos a seguir para el desarrollo de nuestro programa. Llevando acabo asi el programa con la metodología que usa el programador y siguiendo los requerimientos del cliente.

No es recomendable seguir usando el método tradicional porque el cliente no seguirá viendo el progreso de su programa, y partiendo que el cliente no suele tener claro lo que quiere, cada reunión con el para mostrar la evolución del programa es crucial para ver si el cliente consigue lo que quiere. También después de segmentar el trabajo es mas fácil centrarse en un solo bloque por lo que para el programador es más fácil centrarse.