

Proyecto
“Jarvis IDE”

Descripción del proyecto:

1. Párrafo introductorio:

El proyecto trata sobre un IDE en el que por medio de reconocimiento de voz se pueden crear diagramas de flujo con sus respectivas figuras. El IDE contendrá el diagrama de flujo descrito por la persona y una sección en donde podrá ver las funciones que ha creado en ese algoritmo pero en código python. El proyecto se construirá con Java y estará limitado a las instrucciones de crear variables, if, while, prints, inputs, funciones, y acciones libres.

a. Listado de features:

- i. Ver algoritmo creado: el algoritmo se creará al momento que el usuario termine de decir la instrucción vocal.
- ii. Ver funciones: el usuario podrá ver las funciones en el panel superior izquierdo del algoritmo que ha creado.
- iii. Guardar diagrama: el archivo se guardará en el lugar deseado del usuario en su computadora.
- iv. Asignación de parámetros: el usuario podrá ver en el panel inferior izquierdo los parámetros que le va a asignar a su instrucción.

b. Herramientas que se utilizará:

- i. APIs: se hará uso de apis por las cuales se pueda cumplir la funcion de reconocimiento de voz, se harán varias pruebas para decidir cual usaremos en el proyecto final:
 1. Dialog Flow: <https://dialogflow.com/> es un tipo de chatbot que reconoce voz para convertirlo a texto.
 2. Google Cloud: <https://developers.google.com/api-client-library/java/apis/language/v1> epecificamente seria cloud speech to text api que es un api muy completo de google que reconoce hasta 120 idiomas para reconocer la voz y pasarlo a texto.
 3. wit.ai: <https://wit.ai/getting-started> es un tipo de api con cierto nivel de inteligencia artificial en donde aprende el lenguaje humano.
- ii. Librerías: se utilizarán librerías para poder dibujar las figuras del diagrama de flujo:
 1. canvas: <https://docs.oracle.com/javase/7/docs/api/java/awt/Canvas.html> representa un rectangulo blanco en la pantalla con el cual se puede dibujar

2. Graphics:

<https://docs.oracle.com/javase/7/docs/api/java/awt/Graphics.html> es una librería de java que permite dibujar sobre la plataforma que se haya creado

2. Diseño de la experiencia de usuario: (Mockups)

- <https://marvelapp.com/9e7afh1>

3. Catálogo de clases:

a. Clase:

- i. Descripción, Propósito
- ii. Atributos:
 1. Atributo:
 - a. Descripción, Propósito,
 - b. Tipo
- iii. Métodos:
 1. Método:
 - a. Descripción, Propósito, Modificadores de visibilidad
 - b. Firma

b. Algoritmo

- i. Clase principal, contiene la secuencia de las acciones.
- ii. Atributos:
 1. secuencia:
 - a. Lista de acciones, contiene la secuencia de acciones
 - b. Tipo: lista.
- iii. Métodos:
 1. addAccion:
 - a. agrega una accion a la secuencia
 - b. Firma: public void addAccion(accion)
 2. removeAccion:
 - a. elimina una accion de la secuencia
 - b. Firma: public void removeAccion(accion)
 3. guardar:
 - a. guarda el algoritmo creado
 - b. Firma: public

c. Módulo

- i. Clase secundaria, para contener todas las funciones.
- ii. Atributos:
 1. funciones:
 - a. lista de funciones que tiene el programa
 - b. Tipo: Lista

d. Input

- i. Para crear inputs
- ii. Atributos:
 1. texto: instrucción, comando de la instrucción
 2. valorIngresado: captura el texto que el usuario ingreso

iii. Métodos:

1. setTexto: le asigna el valorIngresado al atributo texto.
2. Firma: public void setTexto()

e. Funcion

f. Var

g. If

h. Condición

i. While

i. Clase para crear ciclos while que reciban parámetros.

j. Print

k. Acción

4. Diagrama UML:

