

# ¿Como configurar mi ambiente de javafx en Visual Studio Code (Vscode)?

## Paso 1:

Actualizar o descargar la última versión de Vscode

## Paso 2:

Instalar en Vscode la extensión Java Extension Pack

(<https://marketplace.visualstudio.com/items?vscjava.vscode-java-pack>).

Java Extension Pack es una colección de extensiones populares que pueden ayudar a escribir, probar y depurar aplicaciones Java en Visual Studio Code.

## Paso 3:

Reiniciar VS code

## Paso 4:

Para comenzar con el proyecto, acceder a la paleta de herramientas de Vscode (presionar CTRL + SHIFT + P) e ingresar *'Java: Create java project...'* en modo de trabajo No build tools

## Paso 5:

Seleccionar la ubicación en que se quiere guardar el proyecto y asignar un nombre ejemplos: 'Hola\_Mundo', 'Project1'

## Paso 6:

Instalar las siguientes extensiones de javafx para Vscode: **javafx Support** (<https://marketplace.visualstudio.com/items?itemName=shrey150.javafx-support>) y **FXML Viewer** (<https://marketplace.visualstudio.com/items?itemName=lzw-723.fxml-viewer>) estas herramientas ayudarán a crear interfaces gráficas de usuario.

### Paso 7:

Ir a nuevamente a la paleta de herramientas de Vscode (presionar CTRL + SHIFT + P) y buscar

*'Configure java runtime'*

### Paso 8:

Desde la vista de Configure Java Runtime, acceder a la pestaña de Project JDKs e instalar el jdk sugerido (botón azul), esto hará la descarga de un archivo AdoptOpenJDK con extensión

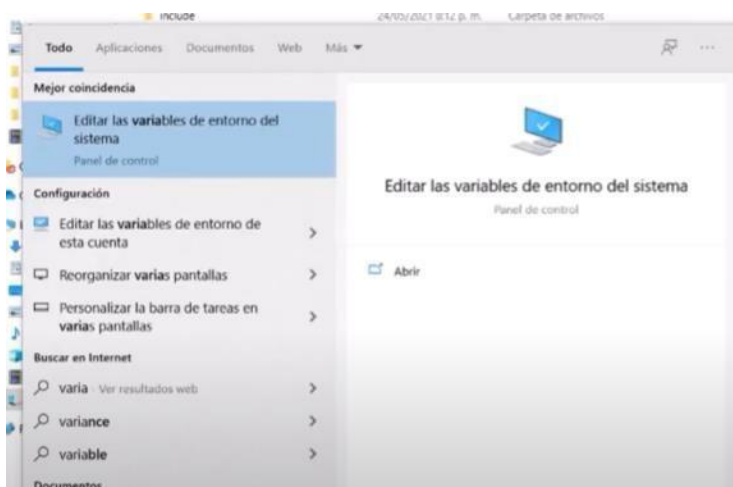
.msi, el siguiente paso es abrirlo y seguir los pasos de instalación.

### Paso 9:

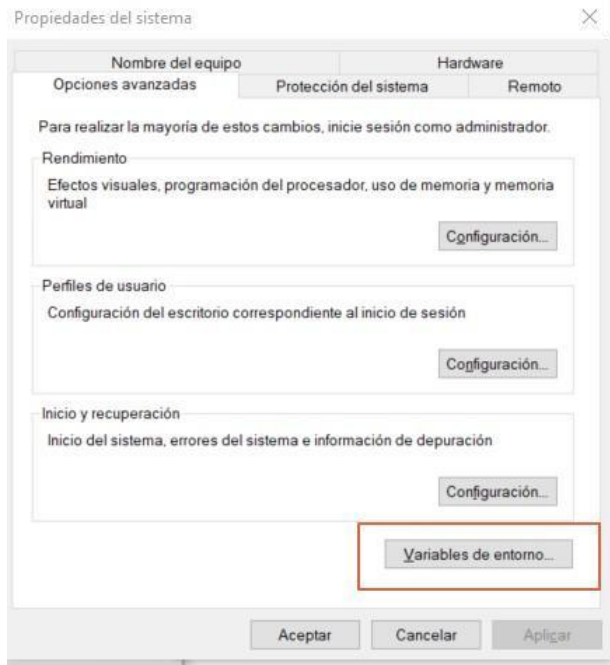
Una vez finalizada la instalación de AdoptOpenJDK buscar su ubicación en los archivos de programa del equipo para configurar las variables de entorno. Acceder a la carpeta AdoptOpenJDK/jdk-11.0.11.9-hotspot y copiar la ruta de su ubicación.

### Paso 10:

Dirigirse al panel de control del equipo y a la opción 'Editar las variables de entorno del sistema'.



En propiedades del sistema acceder a la opción *'Variables de entorno'*



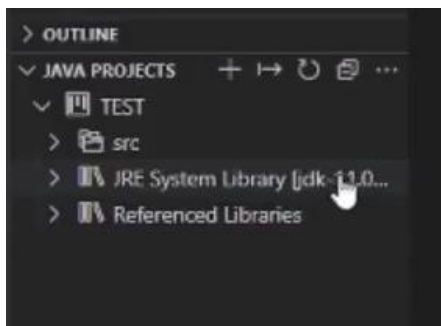
En variables de entorno crear una nueva con **Nombre: JAVA\_HOME** y **valor: ubicación de AdoptOpenJDK copiada en paso anterior** y guardar.

#### Paso 11:

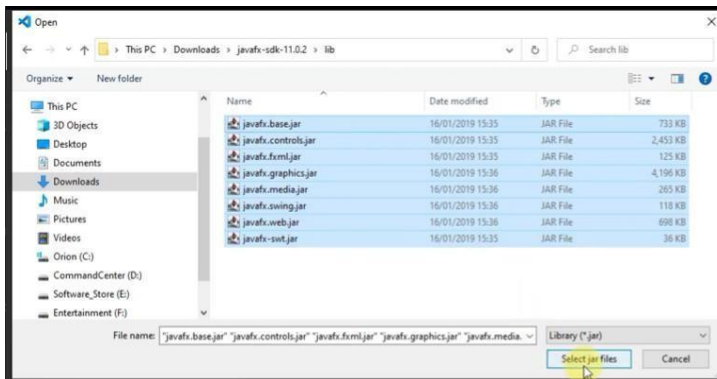
Instalar **javafx SDK** <https://gluonhq.com/products/javafx/> su última versión. Descomprimir carpeta descargada y guardarla donde sea de preferencia. Se necesitará configurar este sdk en Vscode para correr los programas con javafx o cualquier otra aplicación javafx.

#### Paso 12:

Volver al proyecto Java creado y para configurar javafx abrir el archivo App.java y expandir todos los árboles de archivos que se encuentran del lado izquierdo en Vscode.



Al abrir cualquier archivo Java, Vscode, gracias a las extensiones instaladas, detectará esto y habilitará una nueva opción llamada 'JAVA PROJECTS' que contendrá una carpeta con el nombre del proyecto, desplegar este menú y acceder a la subcarpeta Referenced Libraries. A esa carpeta arrastrar o cargar los ítems del sdk que se descargaron en el paso anterior (las que se encuentran en el directorio 'lib')



Una vez cargados todos estos archivos .jar al proyecto, aparecerán nuevos archivos a este que harán parte de su configuración. Un ejemplo es el archivo settings.json el cual contiene los archivos .jar de javafx a los que se hará referencia.

### Paso 13:

Para trabajar, se necesitará que el editor de código Vscode cuente con la sintaxis de aplicaciones javafx, para esto ir a la opción 'Run' en el menú de Vscode y seleccionar la opción Add Configuration.

Dentro de este archivo, añadir un nuevo argumento de configuración llamado "vmArgs" y tendrá como valor la ruta en el equipo a la carpeta **lib** de **java-fx-sdk** descargada en el paso 11. Al final, el archivo debe verse parecido al siguiente, tenga en cuenta que solo debe añadir la línea resaltada:

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "type": "java",
      "name": "Launch Current File",
      "request": "launch",
      "mainClass": "${file}"
    },
    {
      "type": "java",
      "name": "Launch App",
      "request": "launch",
      "mainClass": "App",
      "vmArgs": "--module-path *RUTA EN DONDE SE ENCUENTRAN LIBS JAVAFOX "C:/.."* --add-modules
      javafx.controls,javafx.fxml",
      "projectName": "CP_16_4fdb85f8"
    }
  ]
}
```

Para este paso tener en cuenta:

- La ruta en el parámetro VmArgs del archivo de configuración NO pueden tener espacios, ni signos especiales, Ñ, tildes o Comilla . Cualquiera de estos símbolos hace que java sea incapaz de procesar la ruta.
- El parámetro VmArgs con la ruta de las librerías javafx debe ser añadido en aquellas clases que lo requieran . Ejemplo: Si necesito hacer el llamado de alguna libreria javafx desde mi App.java debo añadir el VmArgs con la ruta de las librerías en la configuración que tenga por *"mainClass": "App"*
- En caso de reutilizar el launch.json de otro proyecto, verificar que el projectName de las configuraciones correspondan con el nombre del proyecto en el que se están pegando y dónde desea hacer uso de las librerías.

**Paso 14:**

Pegar en el archivo App.js el siguiente ejemplo para validar que todo funcione y presionar en el botón Run Java, esto abrirá una ventana de javafx de la prueba.

```
import
javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import
javafx.scene.layout.StackPane;
import javafx.stage.Stage;

public class App extends Application {
    public static void main(String[] args)
    {
        launch(args);
    }

    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("Hello
World!"); Button btn = new Button();
        btn.setText("Say 'Hello World'");
        btn.setOnAction(new EventHandler<ActionEvent>() {

            @Override
            public void handle(ActionEvent event)
            { System.out.println("Hello
World!");
            }
        });

        StackPane root = new StackPane();
        root.getChildren().add(btn);
        primaryStage.setScene(new Scene(root, 300, 250));
        primaryStage.show();
    }
}
```



## Conceptos claves

### ¿Qué es javafx?

Pensar en javafx como un grupo de paquetes que permite crear grandiosas aplicaciones de escritorio e Internet.

javafx permite crear aplicaciones GUI, pero con menos programación y con más efectos visuales a tu disposición.

### ¿Qué es FXML?

FXML es un formato de archivo que utiliza javafx para crear el diseño de las pantallas.

### ¿Qué es un SDK?

Un kit de desarrollo de software (SDK). Es un conjunto de herramientas que ofrece generalmente el fabricante de una plataforma de hardware, un sistema operativo (SO) o un lenguaje de programación. Los **SDK** permiten que los desarrolladores **de** software creen aplicaciones para esa plataforma, ese sistema o ese lenguaje **de** programación específicos.

### ¿Qué es un JDK?

Java Development Kit es un software que provee herramientas de desarrollo para la creación de programas en Java.

### ¿Qué es un archivo .jar?

Un archivo JAR es un tipo de archivo que permite ejecutar aplicaciones y herramientas escritas en el lenguaje Java

### ¿Qué es una librería?

Una librería Java se puede entender como un conjunto de clases que facilitan operaciones y tareas ofreciendo al programador funcionalidad ya implementada y lista para ser usada través de una Interfaz de Programación de Aplicaciones, comúnmente abreviada como API (por el anglicismo Application Programming Interface).

## Ejemplos básicos

### Contenedor

```
Scene scene = new Scene(root, 500, 300);
primaryStage.setScene(scene);
primaryStage.show();
```

### Botón:

```
Button boton = new Button("JavaFX");
boton.setDefaultButton(true);
// Tamaño del boton
boton.setPrefSize(100, 50);
// Tamaño del boton
// Posicion dle boton
boton.setLayoutX(105);
boton.setLayoutY(110);

// Se agrega el boton
root.getChildren().add(boton);
```

```
boton.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent event) {
        //accion que se debe ejecutar al hacer click en boton
    }
})
```

### Label:

```
//Declaración de nuevo texto /Label
Text text = new Text();
text.setText("Welcome to Portal"); //Setear ontenido del texto
text.setFont(Font.font("Calibri", FontWeight.BOLD, FontPosture.REGULAR, 20))
; //Definir estilos del texto
```

### Formulario:

```
//Formulario, label + textfield o input
```



```

Label    name    =    new
Label("UserName"); Label pass =
new Label("Password"); TextField
tf1 = new TextField();
PasswordField tf2 = new PasswordField();

```

#### Tabla o Grid:

```

GridPane root = new GridPane(); //Declaramos el panel con cuadrícula
root.addRow(1, text); //Añadimos fila a cuadrícula root.addRow(id, elm)
root.setVgap(10); //Altura de los espacios verticales entre filas.
root.addRow(2, name, tf1);
root.setVgap(10);

```

#### Menu:

```

MenuBar menubar = new MenuBar();
Menu FileMenu = new Menu("File");
MenuItem filemenu1 = new MenuItem("Clear Messages");
MenuItem filemenu2 = new MenuItem("Clear Forms");
FileMenu.getItems().addAll(filemenu1, filemenu2);
menubar.getMenus().addAll(FileMenu);

```

#### DOCUMENTACIÓN javafx CLASSES:

<https://docs.oracle.com/javase/8/javafx/api/index.html>