

¿Como configurar mi ambiente de JavaFX en Visual Studio Code (vscode)?

Paso 1:

Actualizar o descargar la última versión de VS code

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 nuestro proyecto, acceder a paleta de herramientas de vs code (presionar CTRL + SHIFT + P) e ingresar *'Java: Create java project...'* en modo de trabajo No build tools

Paso 5:

Seleccionar la ubicación en que queremos guardar nuestro proyecto y asignarle un nombre ejemplos: 'Hola_Mundo', 'Project1'

Paso 6:

Instalar las siguientes extensiones de JavaFX para VS code: **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 nos ayudarán a crear interfaces gráficas de usuario.

Paso 7:

Ir a nuevamente a la paleta de herramientas de vs code (presionar CTRL + SHIFT + P) y buscar *'Configure java runtime'*

Paso 8:

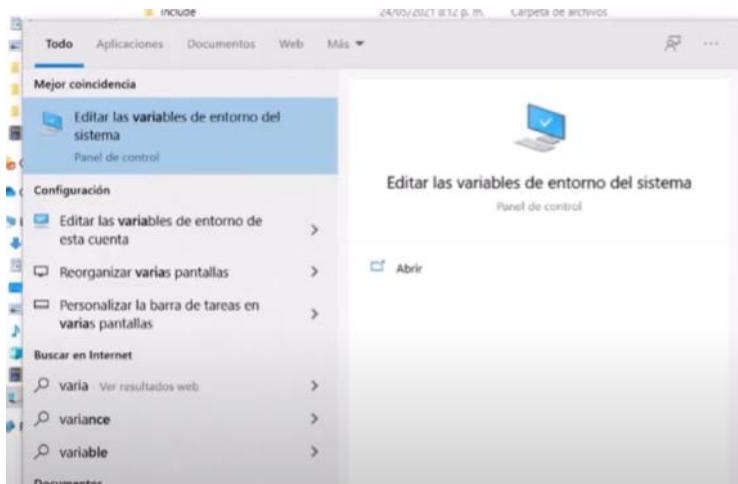
Desde la vista de Configure Java Runtime, accedemos a la pestaña de Project JDKs e instalamos el jdk sugerido (botón azul), esto hará la descarga de un archivo AdoptOpenJDK con extensión .msi, el siguiente paso es abrirlo y seguir los pasos de instalación.

Paso 9:

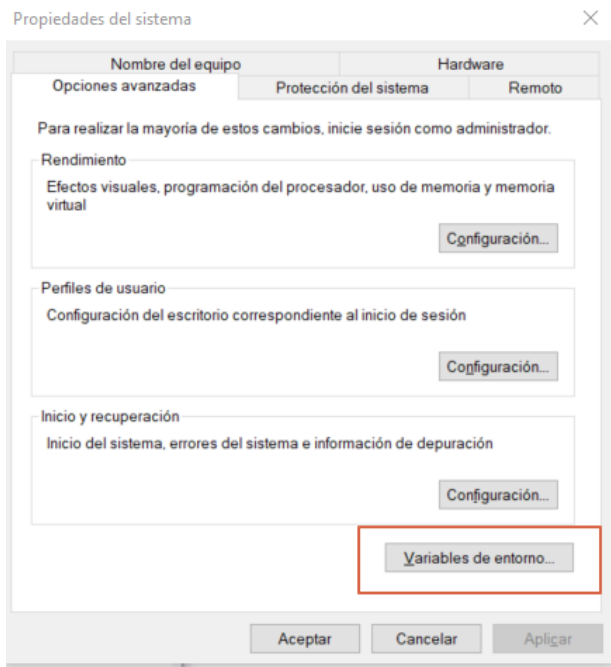
Una vez finalizada la instalación de AdoptOpenJDK buscamos su ubicación en los archivos de programa de nuestro equipo para configurar las variables de entorno. Accedemos a la carpeta AdoptOpenJDK/jdk-11.0.11.9-hotspot y copiamos la ruta de su ubicación.

Paso 10:

No dirigimos al panel de control de nuestro equipo y a la opción *'Editar las variables de entorno del sistema'*



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



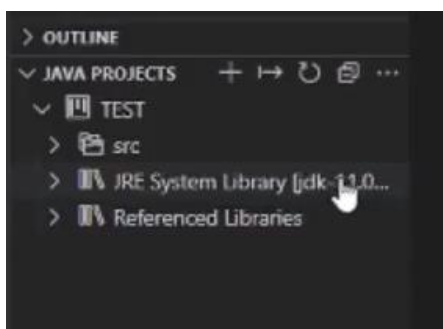
En variables de entorno crearemos una nueva con **Nombre: JAVA_HOME** y **valor: ubicación de AdoptOpenJDK copiada en paso anterior** y guardamos.

Paso 11:

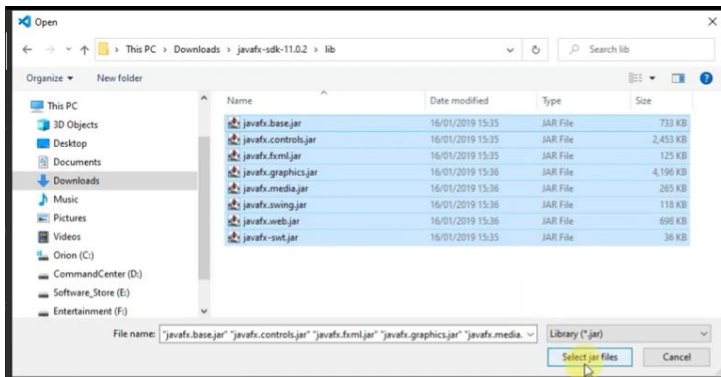
Instalar **JAVAFX SDK** <https://gluonhq.com/products/JavaFX/> su ultima versión. Descomprimir carpeta descargada y guardarla donde sea de preferencia. Necesitaremos configurar este sdk en vs code para correr nuestros programas con JavaFX o cualquier otra aplicación JavaFX.

Paso 12:

Volver a nuestro 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 nuestro vscode gracias a las extensiones instaladas detectará esto y habilitará una nueva opción llamada 'JAVA PROJECTS' que contendrá una carpeta con el nombre de nuestro proyecto, desplegamos este menú y accedemos a la subcarpeta Referenced Libraries. A esa carpeta arrastraremos o cargaremos los ítems del sdk que descargamos en el paso anterior (las que se encuentran en el directorio 'lib')



Una vez carguemos todos estos archivos .jar a nuestro proyecto, aparecerán nuevos archivos a nuestro proyecto que harán parte de su configuración. Un ejemplo es el archivo settings.json el cual contiene los archivos .jar de JavaFX a los que haremos referencia.

Paso 13:

Para trabajar, necesitaremos que nuestro editor de código vscode cuente con la sintaxis de aplicaciones JavaFX, para esto iremos la opción 'Run' en el menú de vscode y seleccionaremos la opción Add Configuration...

Dentro de este archivo, añadiremos un nuevo argumento de configuración llamado "vmArgs" y tendrá como valor la ruta en nuestro equipo a la carpeta **lib** de **java-fx-sdk** descargada en el paso 11. Al final, nuestro archivo debe verse parecido a la siguiente foto:

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "type": "java",
      "name": "Launch Current File",
      "request": "launch",
      "mainClass": "${file}"
    },
  ],
}
```

```

        "type": "java",
        "name": "Launch App",
        "request": "launch",
        "vmArgs": "--module-path /Users/user/Downloads/JavaFX-
sdk-11.0.2/lib --add-
modules JavaFX.controls,JavaFX.fxml",
        "mainClass": "App",
        "projectName": "Test_afab5ca9"
    }
]

```

Paso 14:

Pegamos en nuestro archivo App.js el siguiente ejemplo para validar que todo funcione y presionamos en el botón Run Java, esto abrirá una venta de JavaFX de nuestra 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?

Piense en JavaFX como un grupo de paquetes que le permite a uno crear grandiosas aplicaciones de escritorio e Internet.

JavaFX te 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>