

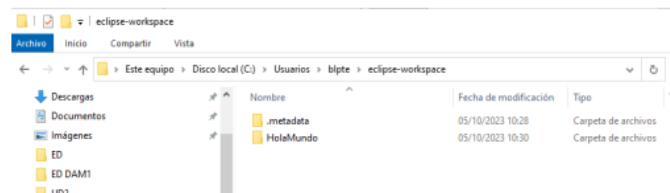
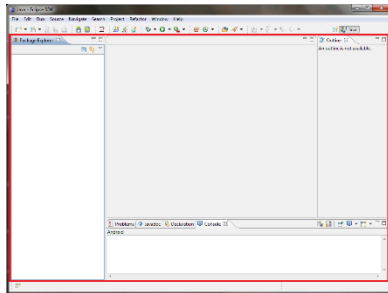
RESUMEN UD2 IDE'S

- **Pasos para la creación de un programa ejecutable**
 - Escritura del código fuente
 - Transformación del código fuente en código objeto mediante un compilador o intérprete
 - Obtención del código ejecutable
- **Un entorno de desarrollo** es una aplicación informática que permite realizar los anteriores pasos de manera conjunta. También se conocen con el nombre de **IDE (Integrated Development Environment)**.
- **Componentes incorporados en la mayoría de los entornos (los dos primeros componentes son obligatorios).**
 - **Editor de código:** nos permite escribir el código fuente. Coloración de la sintaxis (reconoce, resalta y cambia colores de las variables, cadenas de caracteres, palabras reservadas, instrucciones e inicio y fin de corchetes) así es más visual y se detectan antes los errores.
 - **Compilador:** Traduce el código fuente escrito en un lenguaje de alto nivel a un programa que tras enlazarlo con las librerías, se convierte en lenguaje máquina. El proceso se llama compilación.
 - **Intérprete:** se diferencian del compilador en que van realizando la traducción a medida que se va ejecutando la instrucción. Son más lentos que los compilados pero más flexibles.
 - **Depurador:** Se encarga de depurar y limpiar los errores del código fuente de un programa informático. Permite analizar instrucción a instrucción la ejecución de un programa.
 - **Constructor de interfaz gráfica:** utiliza un editor de arrastrar y soltar para facilitar la creación de interfaces gráficas. Permite colocar botones, listas, menús y demás elementos. Algunos IDE incorporan estas herramientas con el plugin correspondiente como veremos con Eclipse.
 - **Control de versiones:** esta herramienta permite controlar los cambios que se realizan sobre las aplicaciones, así se obtienen revisiones y versiones de las aplicaciones en un momento dado de su desarrollo.
- **Entornos de desarrollo de código abierto y gratuitos para la creación y ejecución de programas en Java:** Netbeans, Eclipse, Visual Studio Code...
- **Plugins:**
 - **Los módulos (plugins)** permiten realizar tareas no disponibles en la versión del IDE instalada.
 - El módulo **WindowBuilder** permite crear interfaces gráficas de usuario (GUI) en Eclipse.
 - Para instalarlos en **Eclipse** ir a **Eclipse MarketPlace**. En **Netbeans** está en **Tools>Plugins**

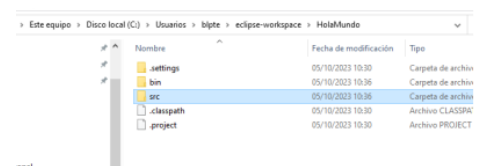
RESUMEN UD2 IDE'S

Eclipse:

Espacio de trabajo: en la aplicación y en la carpeta eclipse-workspace (los cambios en uno de los dos afectan al otro) La carpeta eclipse-workspace contiene una carpeta llamada “.metadata”, es la primera carpeta que crea Eclipse, y es donde se guardan todas las configuraciones, es una carpeta cuyo contenido se actualiza automáticamente por Eclipse, con lo que no se toca bajo ningún concepto.



En la carpeta llamada “**src**” que viene de “Source Code” -código fuente- es donde se almacenará ordenado todo el código que escribamos en un futuro.



“**JRE System Library**”, Aquí están todas las librerías que trae por defecto Java. “System” no significa sistema como tal, sino el paquete que contiene todas las librerías por defecto.

.java extensión de un programa escrito en java

Window Builder-Eclipse WindowBuilder es un diseñador de GUI bidireccional de Java potente y fácil de usar. Está compuesto por **SWT Designer** y **Swing Designer** y hace que sea muy fácil crear aplicaciones Java GUI sin perder mucho tiempo escribiendo código.

Mediante el diseñador visual **WYSIWYG**(What You See Is What You Get) y las herramientas de diseño se pueden crear formas simples para ventanas complejas generándose el código de Java de forma automática. Se pueden agregar fácilmente controles mediante arrastrar y soltar, controladores de eventos a sus controles, cambiar propiedades de los controles utilizando un editor de propiedades, etc.

Se distinguen varios bloques:

Structure, se ven de manera jerárquica los componentes que se han agregado a la ventana.

Palette, se muestran los elementos de tipo contenedor que se pueden añadir a la ventana. Los layouts, es decir la distribución de elementos de la ventana, o los componentes, tipo etiquetas, cajas de texto, listas desplegables, etc. que se pueden elegir.

Properties, se muestran las propiedades del elemento o componente seleccionado, título, tipo de letra, tamaño, color, etc. Estas propiedades se pueden cambiar.

La ventana o formulario, que es donde se van añadiendo los elementos.

Librería Swing

Swing es una biblioteca (librería) de interfaces gráficas de usuario (GUI) para Java. Viene incluida con el entorno de desarrollo de Java (JDK). Extiende a otra librería gráfica más antigua llamada AWT. Presenta componentes ligeros, que pueden tomar diferente aspecto y comportamiento.

RESUMEN UD2 IDE'S

Los **componentes** son los elementos básicos de la programación con Swing. Todo lo que se ve en un GUI de Java es un componente. Los componentes se colocan en otros elementos llamados contenedores que sirven para agrupar componentes.

Los componentes de Swing tienen nombres que comienzan con **J**: **JButton**, **JLabel**, **TextField**, **JComboBox**, **JCheckBox**. (Ver lista anexa)

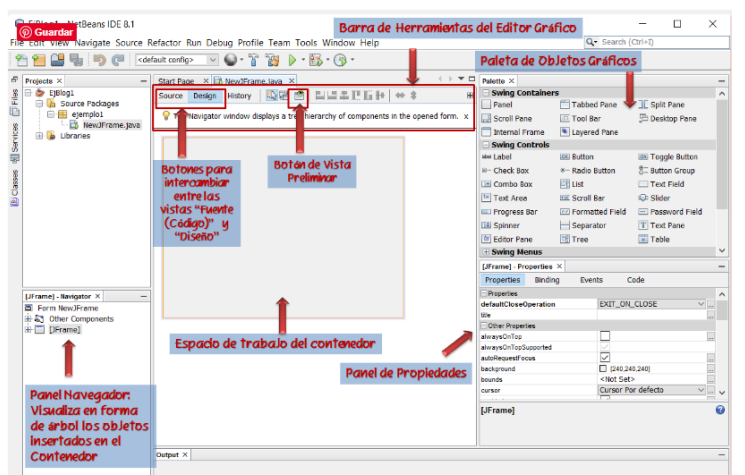
Podemos modificar las **propiedades** de los componentes como ser tamaño de texto y color (Font), título de la ventana (Title), texto a mostrar en un botón (Text) y la más importante la propiedad **Variable** que le asigna un nombre único a cada componente.

Un evento es un suceso que ocurre como consecuencia de la interacción del usuario con la interfaz gráfica. Ejemplos:

- Pulsación de un JButton.
- Cambio del contenido en un JTextField.
- Deslizamiento de una JProgressBar.
- Activación de un JCheckBox.
- Movimiento de la ventana.

getText() es un método que permite acceder (obtener) a la propiedad Text de un componente.

setText(..) es un método que permite setear (asignar) un valor a la propiedad Text de un componente.



También entra el Anexo: Resumen controles Swing

RESUMEN UD2 IDE'S

Para la parte práctica

- *Saber como crear un proyecto en Java tanto en Eclipse como en Netbeans
- *Saber como instalar un plugin tanto en Eclipse como en Netbeans
- * Saber como guardar y entregar un proyecto
- * Conocer los componentes de Swing para hacer una interfaz sencilla como las que hemos hecho en las prácticas de clase.