

## **Anuncios**

- **Rogamos inicie su portátil y arranque Forte cada viernes al llegar:**
  - Desde la barra de tareas, vaya a Inicio-> Programas -> Forte for Java CE -> Forte for Java CE.
  - Forte tarda un rato (2 minutos aprox.) en arrancar. Tenga paciencia.
  - Si Forte le está dando problemas al arrancar, por favor, pida la palabra.
- **Rogamos que arranque también Netscape y se registre en la Web del curso 1.00 cada viernes:**
  - [web.mit.edu/1.00/www/](http://web.mit.edu/1.00/www/)
  - Si tiene problemas al acceder a la Web, levante la mano. Dependerá de la tarjeta de red inalámbrica de su portátil.
- **Si lo desea, también puede traer el portátil a las clases de los lunes y miércoles.**

## **1.00 Clase 2 (Práctica)**

**Introducción a *Forte for Java***

## ¿Que es un IDE?

- **Un entorno de desarrollo integrado (IDE) es un entorno en el que el usuario realiza las tareas principales de desarrollo:**
  - Nombrar y crear los archivos para guardar un programa.
  - Escribir el código (en Java u otro lenguaje).
  - Compilarlo (comprobar que sea correcto, generar un binario ejecutable).
  - Depurarlo y realizar pruebas con él.
  - Además de otras muchas tareas: control de versiones, proyectos, generación de código, etc.
- **La edición comunitaria de *Forte for Java* es un entorno (IDE) bastante estándar que utilizaremos en las prácticas, seminarios y ejercicios del curso 1.00.**
- **Una alternativa a estos entornos (IDE) es una interfaz de “línea de comandos” no visual y en la que las herramientas para cada tarea no suelen estar firmemente integradas.**

## ¿Por qué utilizar un IDE?

- **Los estudiantes del curso 1.00 en ediciones anteriores no asimilaron bien las interfaces de línea de comando.**
  - **Tampoco muchos programadores industriales.**
  - **Aunque con una interfaz de línea de comandos es posible hacer lo mismo que con un entorno IDE, en realidad mucha gente no es capaz.**

## ¿Por qué utilizar un IDE?

- ¿Qué facilidades proporciona un IDE?
  - Representación visual de los componentes del programa.
  - Habilidad para navegar fácilmente por los componentes existentes, pudiendo reutilizarlos con facilidad.
  - Acceso rápido a ayuda y documentación para usar las herramientas y librerías existentes, en vez de tener que escribir unas propias.
  - Mejor respuesta y mensajes de error cuando éstos se dan en su programa.
  - Un depurador, cuya función principal no es la de depurar, sino la de leer y verificar el código.
  - La comunicación entre los programadores de un equipo, que comparten una visión común del programa.
- Estas son las cuestiones del proceso, pero constituyen problemas muy reales en la investigación y la industria.

## Componentes de Forte

- Observe la ventana de Forte de su portátil.
- La interfaz de usuario del entorno IDE *Forte for Java*, está formada por los siguientes componentes:

Ventana principal  
(*Main window*)  
Explorador (*Explorer*)  
Editor de código fuente  
(*Source editor*)  
Hojas de propiedades  
(*Property sheets*)  
Editor de formulario  
(*Form editor*)

Configuraciones del proyecto,  
opciones de compilación y depuración  
Inspector de componentes  
Escribir programas  
Ventana de ejecución, etc.  
Navegador Web

- Los componentes más importantes por el momento son: la ventana principal, el explorador y el editor de código fuente.

# Ventana principal

- La ventana principal aparece por primera vez cuando se lanza el entorno IDE *Forte for Java* y permanece abierta mientras éste se ejecuta.



- La ventana principal presenta una barra de herramientas para las opciones del menú. Esta barra constituye un método rápido y fácil para la ejecución de comandos.
- En la esquina inferior izquierda de la ventana principal hay 5 pestañas, una para cada “Modo” o espacio de trabajo. Estos espacios de trabajo ayudan a organizar las ventanas en conjuntos lógicos.

# Explorador



- El explorador ofrece una visión integrada de todos los objetos y archivos del entorno IDE *Forte for Java*.
- El explorador constituye el punto de partida para muchas funciones de programación, como son crear y ejecutar una aplicación. Le ayuda a trabajar con objetos, a organizar su trabajo, a modificar propiedades de los objetos y a conectar varias fuentes de datos.
- El explorador posee cuatro pestañas: *Filesystems* (sistemas de archivos), *Project* (proyecto), *Javadoc* y *Runtime* (tiempo de ejecución).

## Explorador, 2ª parte

- La pestaña *Filesystems* muestra los sistemas de archivos (carpetas / directorios y sus archivos) que están montados actualmente en el IDE.
  - Existe un comando *File-> Mount* para montar nuevos directorios.
- La pestaña *Project* muestra los archivos y directorios desde el proyecto actual.
  - Los proyectos se utilizan para controlar los componentes relacionados de Java que juntos forman un programa.
- La pestaña *Javadoc* contiene los directorios de la documentación de la API en formato *Javadoc*.
  - *Javadoc* es la documentación en línea de Java.
- La pestaña *Runtime* contiene una lista de procesos que están ejecutándose actualmente e información de depuración.
  - No lo utilizaremos mucho en el curso 1.00.

## Sistema de archivos

- Un sistema de archivos es una jerarquía de archivos como la que se guarda en una unidad local o de red, o en un fichero archivo como JAR, ZIP o WAR.
- Para desarrollar aplicaciones en el entorno IDE, es necesario determinar los sistemas de archivos con los que se desea trabajar, y montarlos en el IDE. Los sistemas de archivos montados pueden visualizarse en la pestaña *FileSystems* del explorador.
- Montar un sistema de ficheros permite que el IDE tenga acceso a dicho sistema.
  - Avanzado: por defecto, cuando se monta un sistema de archivos, éste se añade a su ruta de clase para compilar, ejecutar y depurar su aplicación en el IDE.

# Editor de código fuente

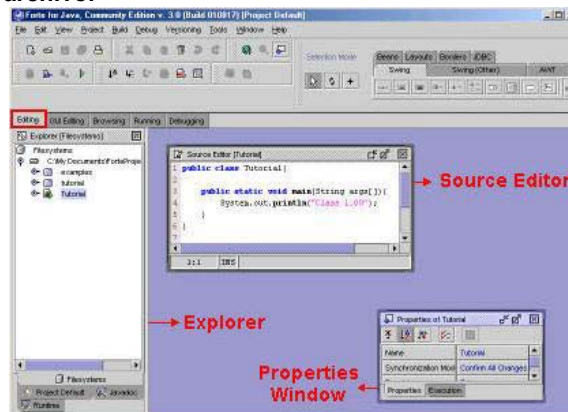
- Usted observa el código Java y lo modifica en el editor de código fuente. Es fácil visualizar y depurar el código en esta ventana porque el color del editor codifica el código según la sintaxis. Por ejemplo, subraya todas las palabras clave por defecto en azul.



```
1 public class Welcome{
2
3     public static void main(String args){
4         System.out.println("Welcome to 1.00");
5     }
6 }
7
```

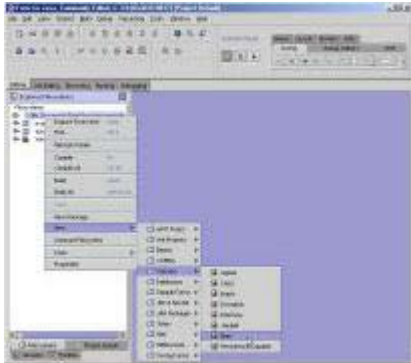
# Modo Editor

- El IDE *Forte forJava* muestra el espacio de trabajo de edición cuando el IDE se arranca. El espacio de trabajo de edición incluye lo siguiente:
  - Explorador (*Explorer*).
  - Ventana de propiedades (*Properties Window*).
  - Editor de código fuente (*Source editor*), que aparece sólo si se abre un archivo.



## Cómo utilizar el modo Editor

### Cómo crear un archivo:



- Haga un clic con el botón derecho sobre el directorio raíz de *FileSystems* en la vista del explorador:
- Seleccione *New->Classes->Main* (Nuevo-> Clases -> *Main*) (observe el dibujo). Se abrirá un asistente de plantilla.
- Escriba “*Welcome*” (Bienvenida) en el cuadro de texto y haga clic en *Finish* (terminar). Este será el nombre del archivo y el nombre de la clase del archivo que esté creando.

## Cómo utilizar el modo Editor, 2ª parte.

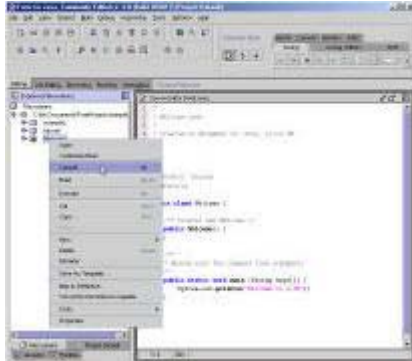
- Una vez que se ha creado el archivo, se abrirá el editor de código fuente.
- Queremos escribir la aplicación “*Welcome*” para que muestre un mensaje por pantalla.
- Es necesario que modifique el método *main* para que aparezca impreso “Bienvenido al curso 1.00”, además de otras cosas. Su método *main* debería ser exactamente como el siguiente:

```
public static void main(String[] args) {  
    System.out.println("Bienvenido al curso 1.00");  
    int estudiantes = 240;  
    int licenciados = 35;  
    double pctLic= (double) licenciados/ estudiantes;  
    System.out.println("Porcentaje de Licenciados: " + pctLic);  
    System.exit(0);  
}
```

## Características del editor

- **Finalización dinámica del código.**
  - Escriba los caracteres iniciales y pulse la tecla ctrl-barra espaciadora para ver las finalizaciones.
- **Emparejamiento de corchetes.**
  - Sitúe el cursor antes / después de { } ( ) para comprobar el emparejamiento { } ( ).
- **Actualización automática del explorador, etc.**
- **Correspondencia de palabras, macros de teclado, etc.**
  - Revise la ayuda de Forte o pregunte a su ayudante técnico.

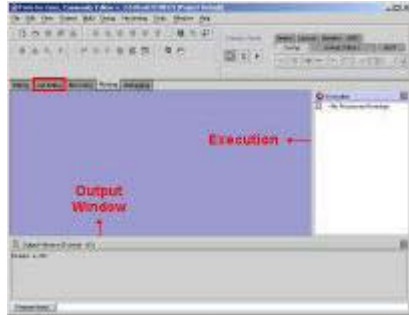
## Cómo compilar su programa



- Haga clic con el botón derecho sobre el archivo "Welcome" en la vista del explorador.
  - Seleccione "Compile" para compilar el archivo. Aparecerá una ventana de salida.
  - Si no hubiese errores en su programa, debería obtener el mensaje "Finished Welcome" (fin de la compilación de "Welcome").
- 
- Si no obtiene este mensaje, es que había un error en su programa. (Nota: Forte ofrece herramientas útiles para depurar errores. (Obtendrá más información sobre estas herramientas en el seminario).
  - Asegúrese de que ha introducido el código correctamente. Vuelva a compilar el programa. Si aún así sigue obteniendo el error, pida la palabra, por favor.

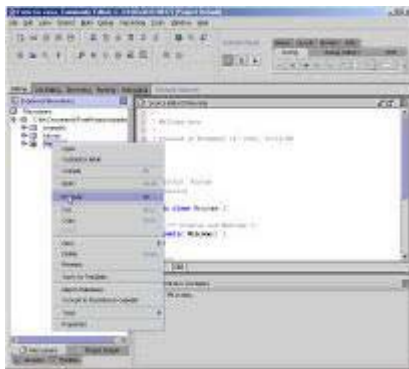
# Modo Ejecución

- Se utiliza el modo Ejecución (*Running workspace*) para ejecutar una aplicación. El espacio de trabajo de ejecución incluye lo siguiente:
  - Ventana de ejecución (*Execution Window*).
  - Ventana de salida (*Output Window*).



- Cuando se ejecuta un programa, el IDE *Forte for Java* cambia automáticamente al modo Ejecución. Si no existen errores de ejecución, se lanza la aplicación.
- Si existen errores de ejecución, el IDE los muestra en la ventana de salida.

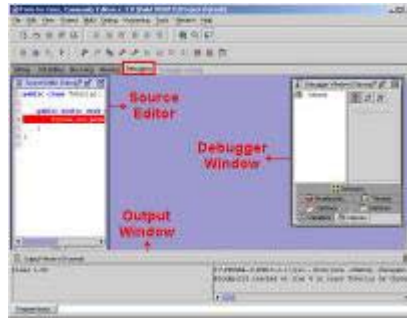
# Modo Ejecución



- Vuelva al espacio de trabajo de edición seleccionando la pestaña "*Editor*" del menú principal.
  - Haga clic con el botón derecho sobre el archivo *Welcome* de la vista del explorador y seleccione "*Execute*" (ejecutar).
- Si su programa se ha compilado con éxito anteriormente, el IDE pasará al modo Ejecución y en la ventana de salida aparecerá "Bienvenido al curso 1.00".

## Modo Depuración

- Para depurar una aplicación, se utiliza el modo Depuración (*Debugging workspace*). Este espacio de trabajo incluye lo siguiente:
  - Ventana del depurador (*Debugger Window*).
  - Ventana de salida (*Output Window*).
  - Editor de código fuente (*Source Editor*).



## Modo Depuración

- Vuelva al modo Editor seleccionando la pestaña “*Editor*” del menú principal.
- Seleccione el archivo “*Welcome*” en la vista del explorador y vaya a *Debug* (depurar) > *Step Into* (entrar).
- El IDE cambiará a modo Depuración. La línea de código que se ejecute a continuación estará en azul. Por ahora, la primera línea de código debería ser la destacada en azul.
- Para ejecutar el código línea a línea, seleccione una y otra vez *Debug* -> *Step Into* (o utilice la tecla rápida “*F7*”). Si se abre un cuadro de diálogo que dice: “*Select an action upon Step Into a method without a source*”, elija “*Step Out*” (salir).
- El depurador ofrece herramientas útiles para controlar el valor de las variables y para añadir puntos de parada. Obtendrá más información sobre estas herramientas en el seminario.

## Modo Navegación

- El modo Navegación (*Browsing workspace*) le ayuda a visualizar los paquetes, objetos y miembros de una aplicación. Este espacio de trabajo incluye lo siguiente:
  - Navegador de objetos (*Object Browser*).
  - Ventana de propiedades (*Properties Window*).



## Proyectos en Forte

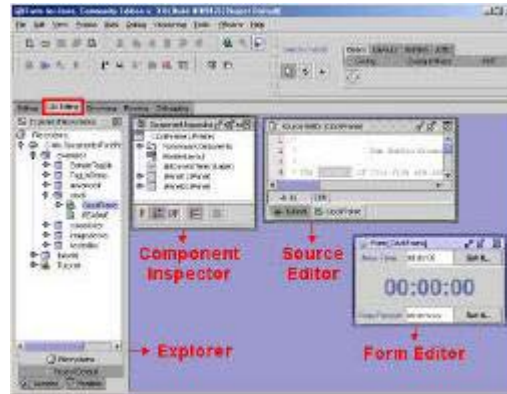
- El IDE *Forte for Java* le ayuda a organizar y administrar los archivos, mediante el concepto de proyecto.
- Un proyecto contiene todos los archivos necesarios para ejecutar una aplicación.
- Se pueden administrar fácilmente todos los archivos de una aplicación o *applet* porque están en un proyecto.
- Por ejemplo, cuando se compila un proyecto, se incluyen todos los archivos del lenguaje Java (obtendrá más información en la parte de proyectos que se verá en el seminario).

## Modo Edición de una GUI

- El modo Edición de una GUI (*GUI Editing workspace*) es el área principal donde se puede desarrollar una interfaz gráfica de usuario. Este espacio de trabajo incluye lo siguiente:

- Explorador.
- Editor de código fuente.
- Inspector de componentes.
- Editor de formularios.

- No trataremos esto en el curso 1.00, pero puede utilizarlo en los boletines de problemas 6-10.



## Resumen de un entorno IDE

- El IDE Forte se ocupa del desarrollo central del software.
  - Se utilizan también otras herramientas:
    - Lenguaje de modelado unificado (UML), para los requisitos y la estructura global de las aplicaciones.
    - Modelado de datos para bases de datos.
    - Herramientas de prueba automatizadas, controladores de pruebas.
    - Etc.
- Referencias:
  - Centro Web de aprendizaje de Sun (<https://tmn.sun.com>)
  - Edición comunitaria de *Forte for Java*, v. 3.0, AYUDA.

## Ejercicio 1

- Utilizando Forte, cree un archivo *Lecture2*.
- Escriba un método *main()* en la clase *Lecture2* para:
  - Declarar como *double* *x* e *y*, para definir la longitud y ancho de un rectángulo.
    - Inicialícelos en su programa como valores positivos.
  - Calcule el perímetro y el área del rectángulo.
    - Declare dos variables más de tipo *double* para almacenar los resultados.
  - Imprima el perímetro y el área.
- Compile su programa.
- Repase su programa:
  - Pásele el depurador al programa.

## Ejercicio 2 (Opcional)

- Utilizando Forte, cree un archivo *Quadratic*.
- Escriba un método *main()* en la clase *Quadratic* para:
  - Declarar e inicializar una variable *x* de tipo *double*.
  - Calcular el valor de la ecuación cuadrática:  $3x^2 - x - 2$ 
    - Declare una variable de tipo *double* para almacenar el resultado.
  - Imprima el valor de la ecuación cuadrática. La salida debería ser algo así:
    - Para  $x = 4.0$  el valor de la ecuación es 42.0.
- Compile y ejecute su programa.

## **Ejercicio 2 (Avanzado)**

- **Introducción a los “Tipos de datos”:**
  - **Ejecute el programa con varios valores para  $x$ . Utilice valores con comas decimales, valores grandes, valores pequeños, valores negativos y cero.**
  - **Avanzado: resuelva la ecuación con lápiz y papel (utilice la fórmula cuadrática). Ésta debería dar como resultado cero para  $x = 1.0$  y para  $x = -2/3$ . Pruebe estos valores para  $x$ . ¿Son exactamente correctos los resultados? (Veremos más sobre este tema en la próxima clase).**