UNLP. Facultad de Informática.
Algoritmos y Estructuras de Datos
Cursada 2018

Práctica Nº 1-A
Eclipse

1. **El entorno de Desarrollo.** Para realizar las prácticas deberá instalar:

- **Java Development Kit (JDK):** Descargue la Plataforma JAVA (Edición Estándar o Java SE) desde la URL http://www.oracle.com/technetwork/java/javase/downloads/index.html

- **Eclipse:** Descargue desde la URL http://www.eclipse.org/downloads/eclipse-packages/. Seleccione la versión Eclipse IDE for Java Developers y descomprima.

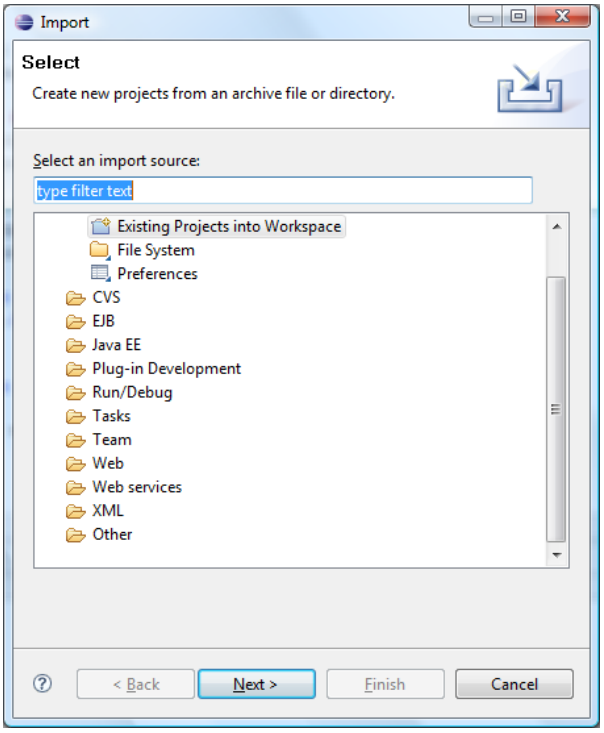
2. Uso de Eclipse

2.1. Importando un Proyecto en Eclipse

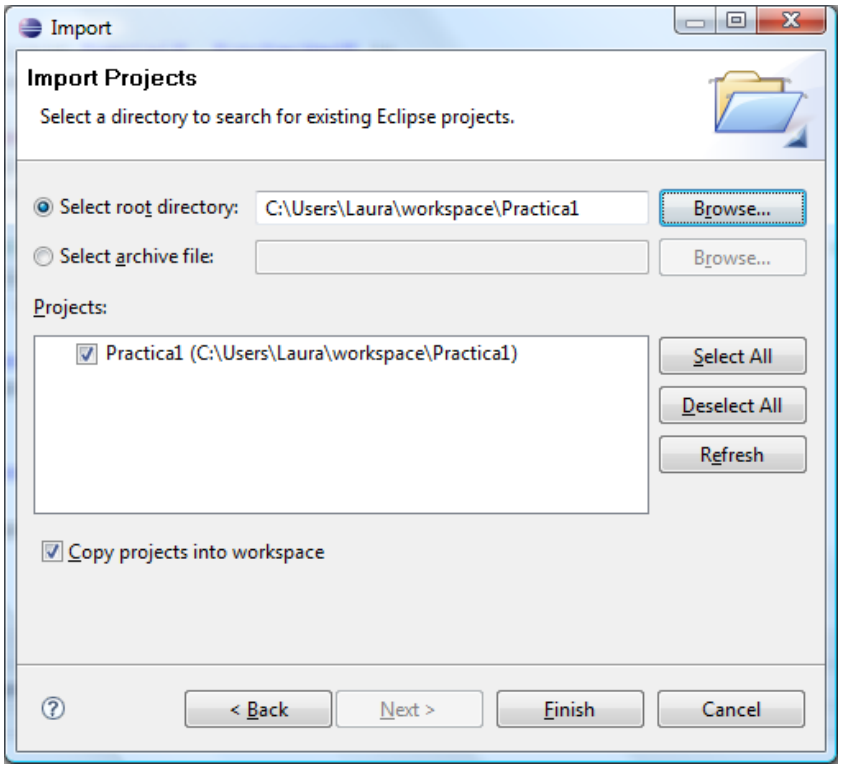
Descargue el archivo **Practica1.zip** y descomprima en algún lugar de su disco local.

Eclipse organiza el código fuente en proyectos con el propósito de facilitar el desarrollo de aplicaciones, sin embargo, esto no es un requerimiento del lenguaje Java.

a. Desde la barra de menú, seleccione **File -> Import …**
b. Seleccione **Existing Projects into Workspace** y presione **Next**.



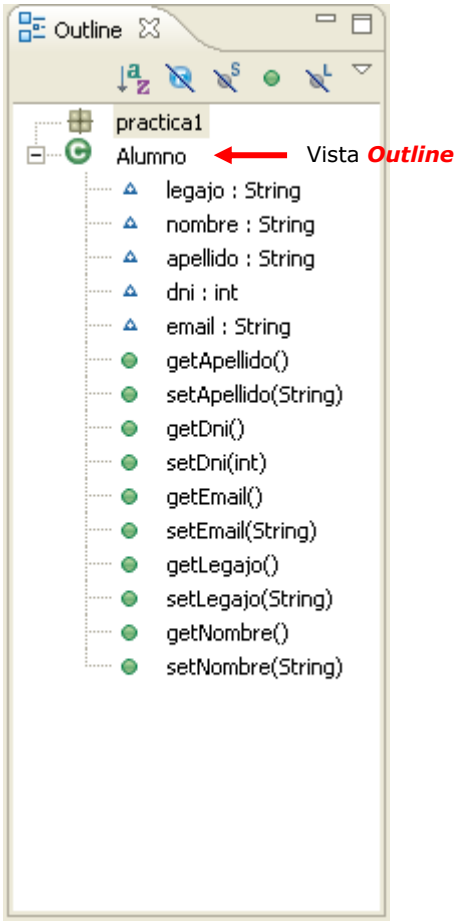
c. Luego seleccione la carpeta **Practica1** que descomprimió en su disco. Presione el botón **Select All** para importar todos los recursos de ese archivo a su proyecto como muestra la figura y presione **Finish**.



NOTA: *Eclipse utiliza el concepto de vistas para permitirnos explorar y modificar los programas, examinar resultados, depurar código, etc.* ***Las perspectivas son organizaciones visuales de una selección de esas vistas****. Existen perspectivas preestablecidas y también es posible crear nuevas personalizadas.*
Note que una vez importado el proyecto, Eclipse muestra por defecto la perspectiva ***Java****.*

2.2. Explorando paquetes y editando código en el entorno

a. En la vista ***Package Explorer*** expanda el paquete practica1 para inspeccionar su estructura. Cada paquete contiene uno o más archivos Java.
b. Busque la clase **Alumno.java** y haga doble click para abrirlo en el editor java. Observe el código fuente: estructura de la clase, palabras claves, etc.
c. Seleccione la vista ***Outline***. Observe como muestra el paquete al cual pertenece la clase, los paquetes que importa, las declaraciones, miembros, métodos.



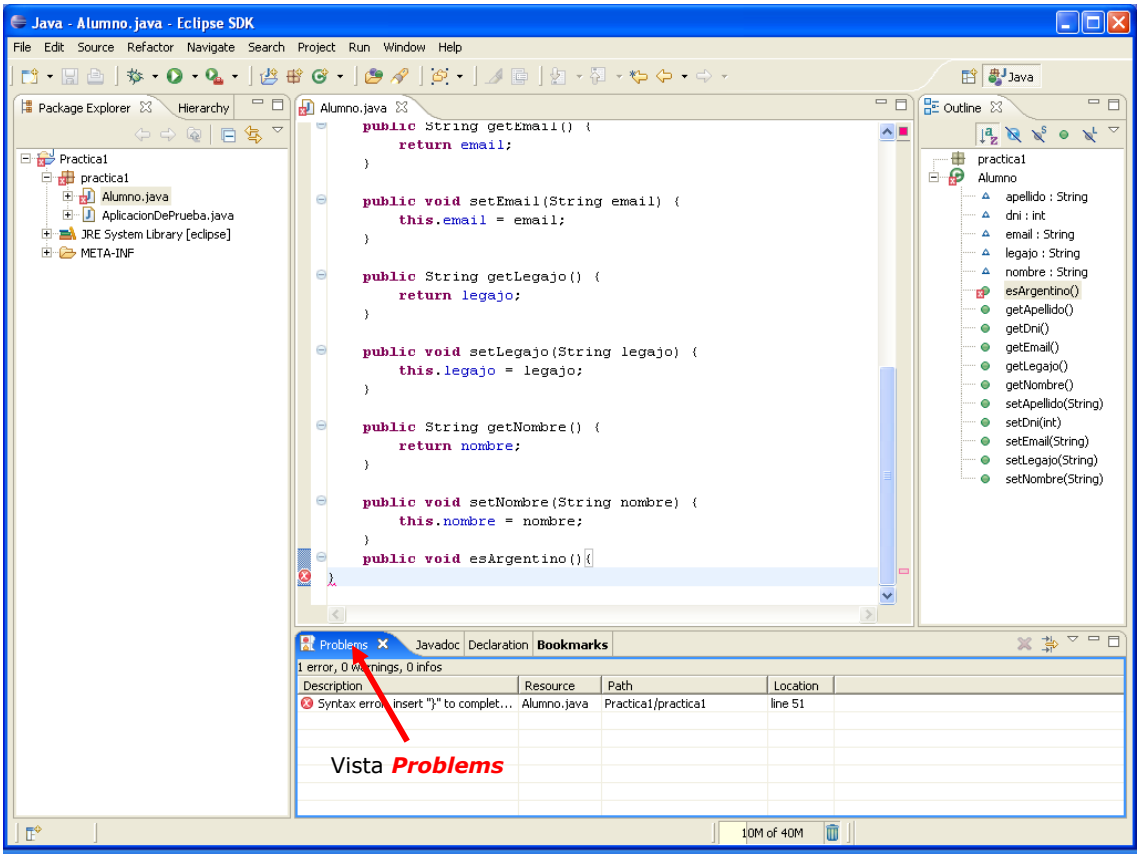
d. El código fuente puede editarse completamente o visualizando sólo un método determinado. Seleccione en el editor la clase **Alumno.java** y seleccione en la vista ***Outline*** un método determinado, haciendo doble clic sobre el mismo se accede directamente a la definición del mismo. Presione el botón ***Sort***, para ver ordenado alfabéticamente los elementos de la vista ***Outline***.
e. Cierre el archivo fuente **Alumno.java** y si hizo cambios no los salve.

2.3. Usando el asistente de código

a. En la perspectiva Java, y desde la vista ***Package Explorer*** abra el archivo **Alumno.java.**
b. Asegúrese de que el botón **Sort** (de la vista ***Outline***), esté presionado para que muestre la vista ordenada alfabéticamente.
c. Agregue en el código fuente, al final de la clase el siguiente código:

```
public void esArgentino(){
```

Observe como automáticamente el código es agregado en orden alfabético
d. Presione el botón **Save**. Eclipse compila automáticamente cuando se salvan los fuentes. Los errores son mostrados en la vista ***Problems*** y en el editor con el símbolo: .

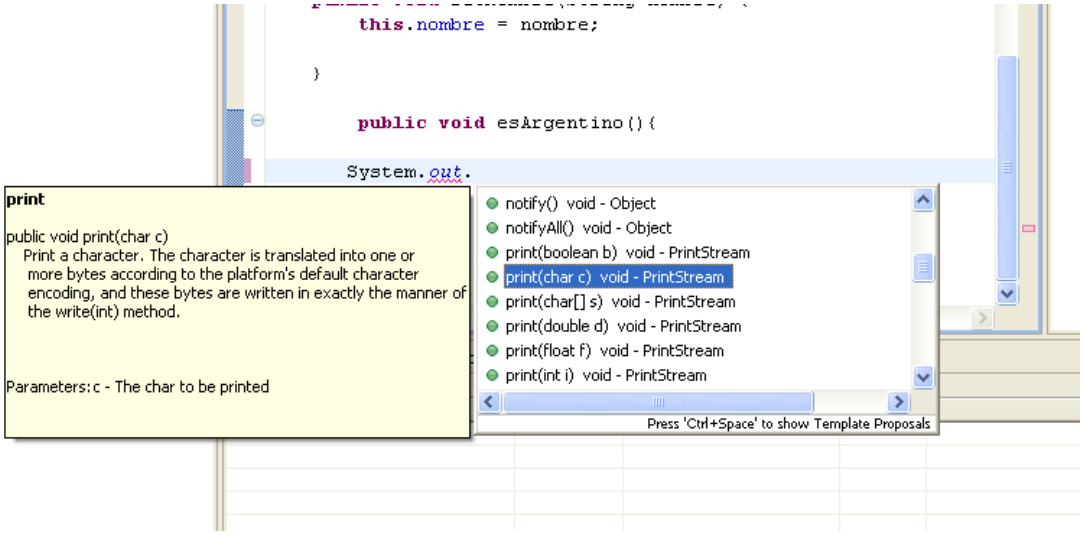


2.4. Usando el asistente de contenido

a. Continúe trabajando con la clase **Alumno**. En la vista ***Outline***, seleccione el método `esArgentino()`.Agregue las siguientes líneas en el método seleccionado:

```
System.out.
```

b. Con el cursor al final de System.out. presione CTRL+Barra espaciadora para activar el asistente de código.



c. Seleccione el método print(char c) y presione ENTER. Después de que el código es insertado complételo así:

```
System.out.print('A');
```

También se puede activar este asistente poniendo la referencia del objeto o una clase y "."

d. Salve la clase.

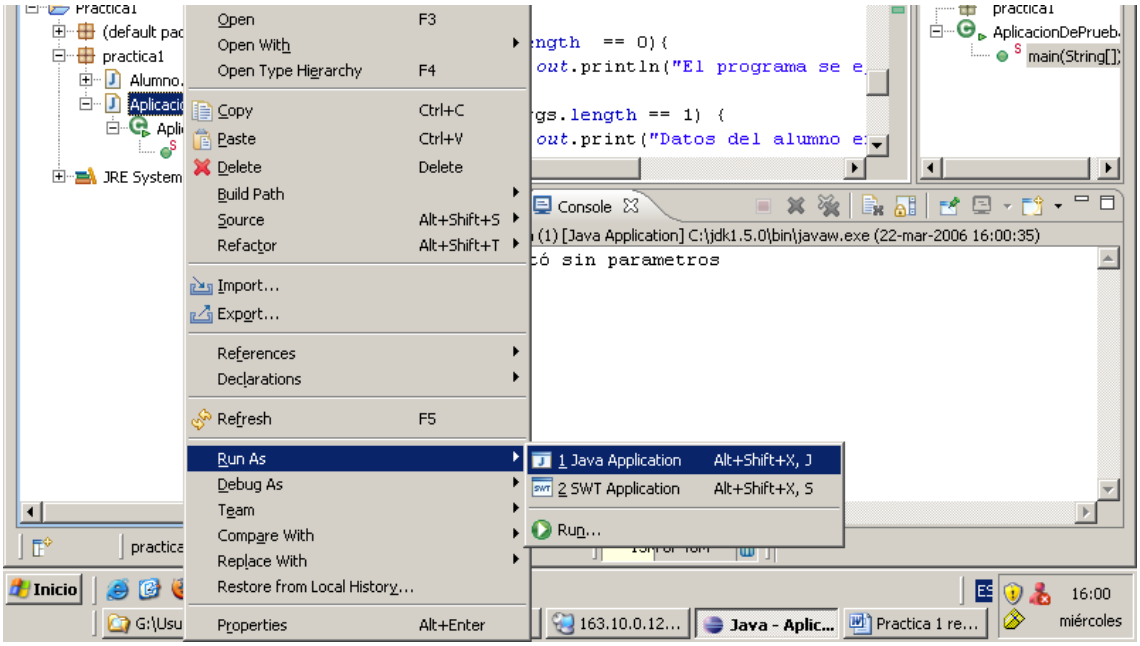
2.5. Ejecutando un programa Java

Para ejecutar un programa Java es necesario tener una clase que contenga un método llamado main cuya declaración sea:

public static void main(String[] args) *{}*

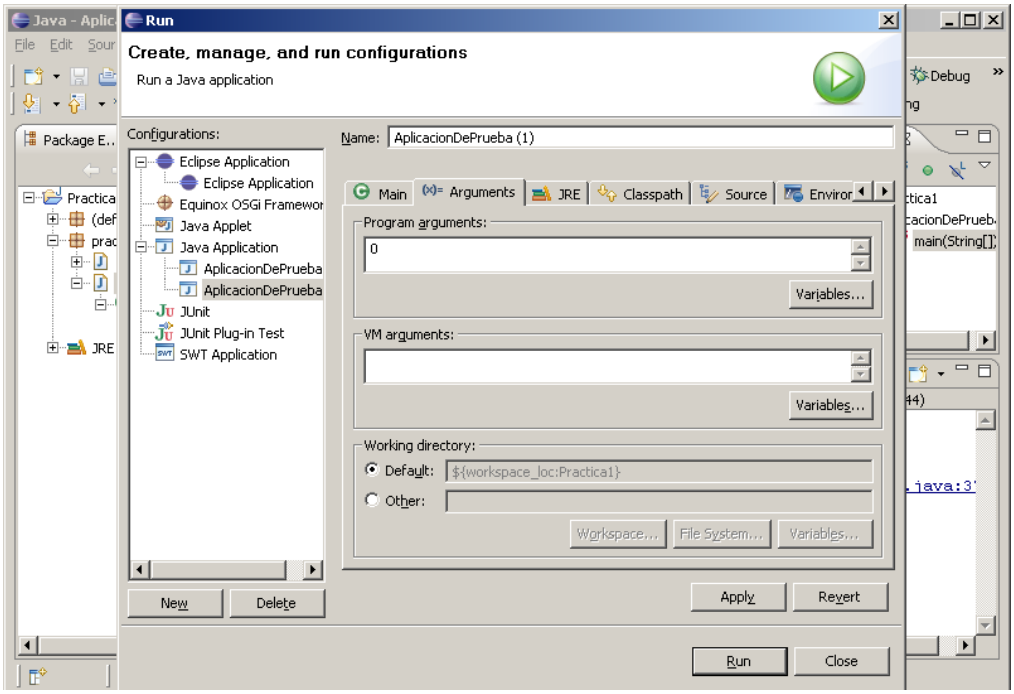
En esta aplicación ejemplo la clase que posee este método es AplicacionDePrueba.

a. Ejecute la aplicación seleccionando la clase AplicacionDePrueba en la vista Package Explorer y con el botón derecho seleccione **Run As** -> **Java Application**
La aplicación crea 2 alumnos e imprime los datos personales del alumno especificado por un número de orden. Si no se especifica dicho número imprime "`El programa se ejecutó sin parámetros`".

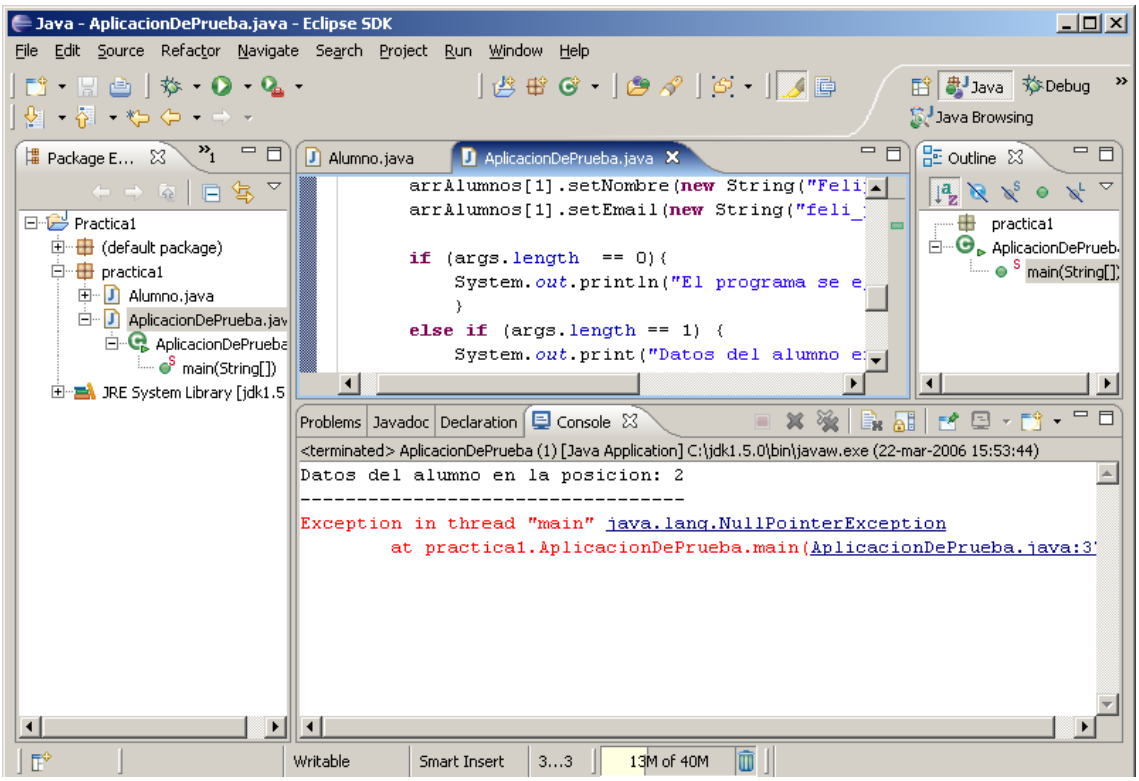


Java utiliza una vista llamada Console para dejar mensajes. Esta aplicación hace uso de ella para obtener una salida de forma sencilla.

b. Ejecute la aplicación pasándole como parámetro el número de orden igual a 0 ó 1 Para esto seleccione el menú **Run** -> **Run...**, luego seleccione la pestaña **Arguments** y en **Program Arguments** especifique 0 ó 1. El programa debería imprimir los datos personales del alumno seleccionado.



c. Pruebe ejecutar el programa con un número de orden mayor a 2. Esto debería dar un error de ejecución.



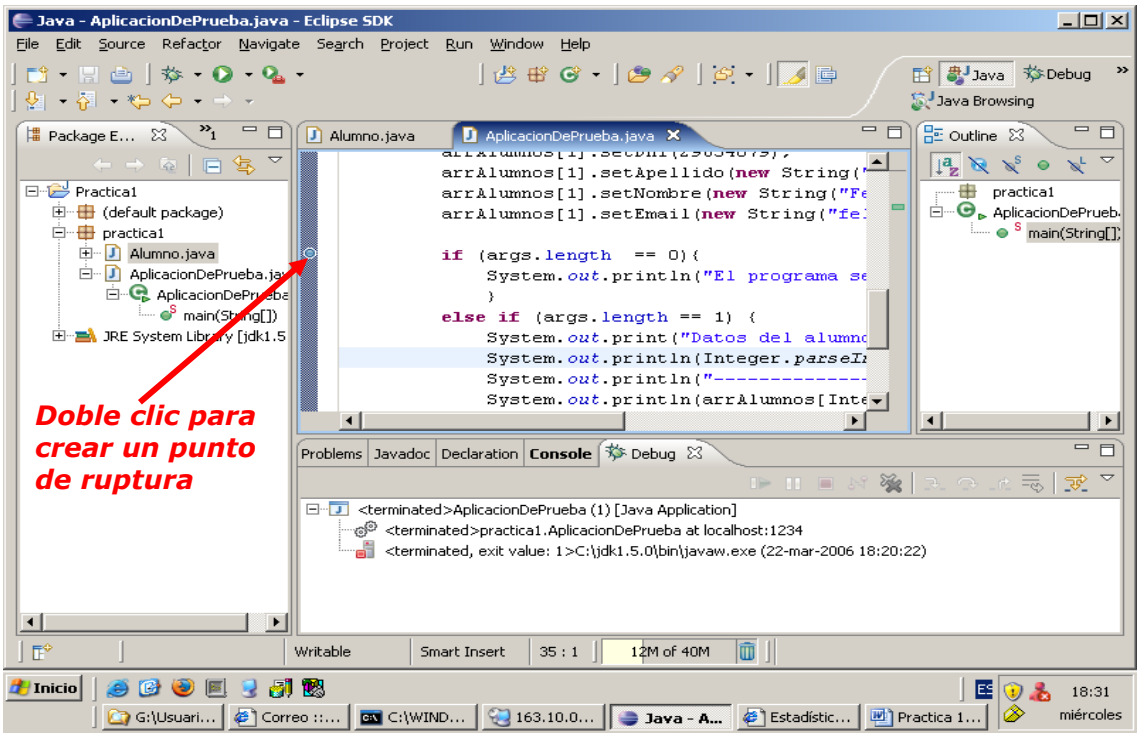
2.6. Usando el Debug

Eclipse provee una perspectiva llamada Debug para encontrar errores que puedan producirse durante la ejecución de un programa. Para esto se pueden establecer puntos de ruptura (Breakpoints) para que a partir de ahí se pueda correr el programa paso a paso y pudiendo examinar el estado de las variables y el flujo de ejecución que sigue el mismo.

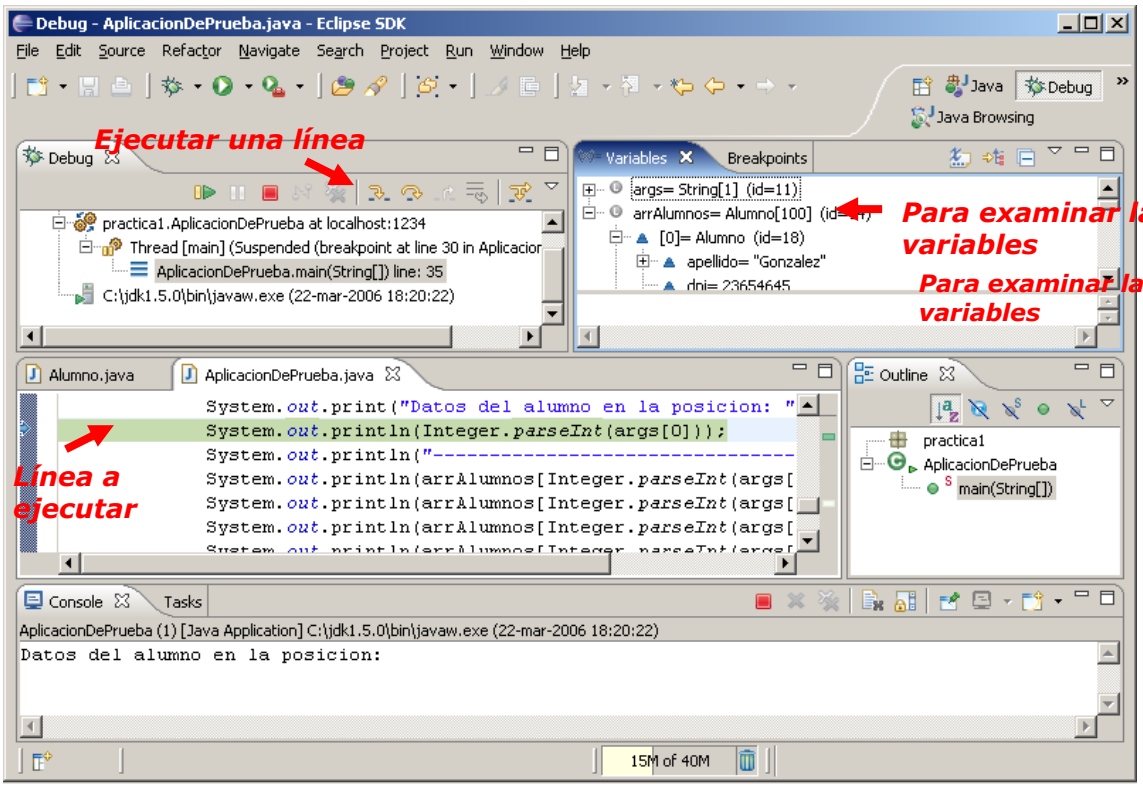
a. En el código de la clase AplicacionDePrueba coloque un punto de ruptura en la línea que dice:

```
if (args.length  == 0){
```

haciendo doble click en el margen izquierdo sobre esa línea.



b. Ejecute el debug seleccionando el menú **Run** -> **Debug...** deberá abrir la perspectiva Debug. Examine el contenido de las variables que contienen los datos de los alumnos y ejecute paso por paso el programa.



2.7. Reemplazo de un método desde el *Local History*

a. Continúe trabajando con la clase **Alumno.java**. En la vista ***Outline***, seleccione el método `esArgentino()` que acaba de crear, edítelo y agregue una }.Luego abra el menú contextual y seleccione **Delete** y confirme.
b. Agregue nuevamente un método al final de la clase con el siguiente código:

```
public void esArgentino(){
        String nacionalidad = "Argentino";
        System.out.println("Si soy " + nacionalidad);
}
```

c. Salve la clase.
d. En la vista ***Outline***, seleccione el método `esArgentino()` que acaba de crear, abra el menú contextual y seleccione **Replace with -> Element from Local History...**.
e. Se abrirá una ventana con las versiones anteriores del método. Cada vez que se elige una versión en el panel superior, los paneles inferiores muestran la versión actual comparándola con la seleccionada en el panel superior.
f. Acepte presionando el botón **Replace**. El código será reemplazado.