

# Java Inicial

(20 horas)





# Temario

1. Programación Orientada a Objetos
2. Introducción y Sintaxis Java
3. Sentencias Control Flujo
4. POO en Java
5. Relaciones entre Objetos
6. Polimorfismo, abstracción e interfaces
7. Excepciones
8. **Conceptos avanzados**



# Tema 8

Conceptos Avanzados: Applets



# Objetivos

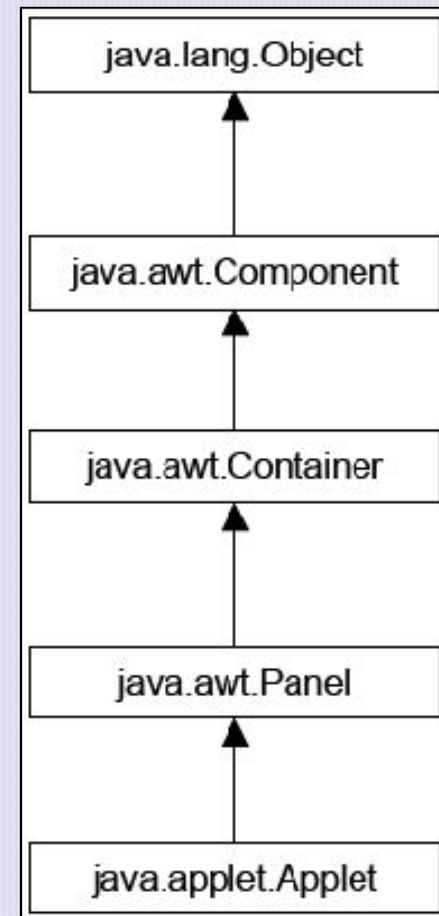
1. Programación Orientada a Objetos
2. Introducción y Sintaxis Java
3. Sentencias Control Flujo
4. POO en Java
5. Relaciones entre Objetos
6. Polimorfismo, abstracción e interfaces
7. Excepciones
8. **Conceptos avanzados**

## ■ Applets

- Introducción
- Ciclo de Vida
- Métodos
- Clase Graphics
- Imágenes
- Parámetros
- Applets / AWT

## ■ Applets

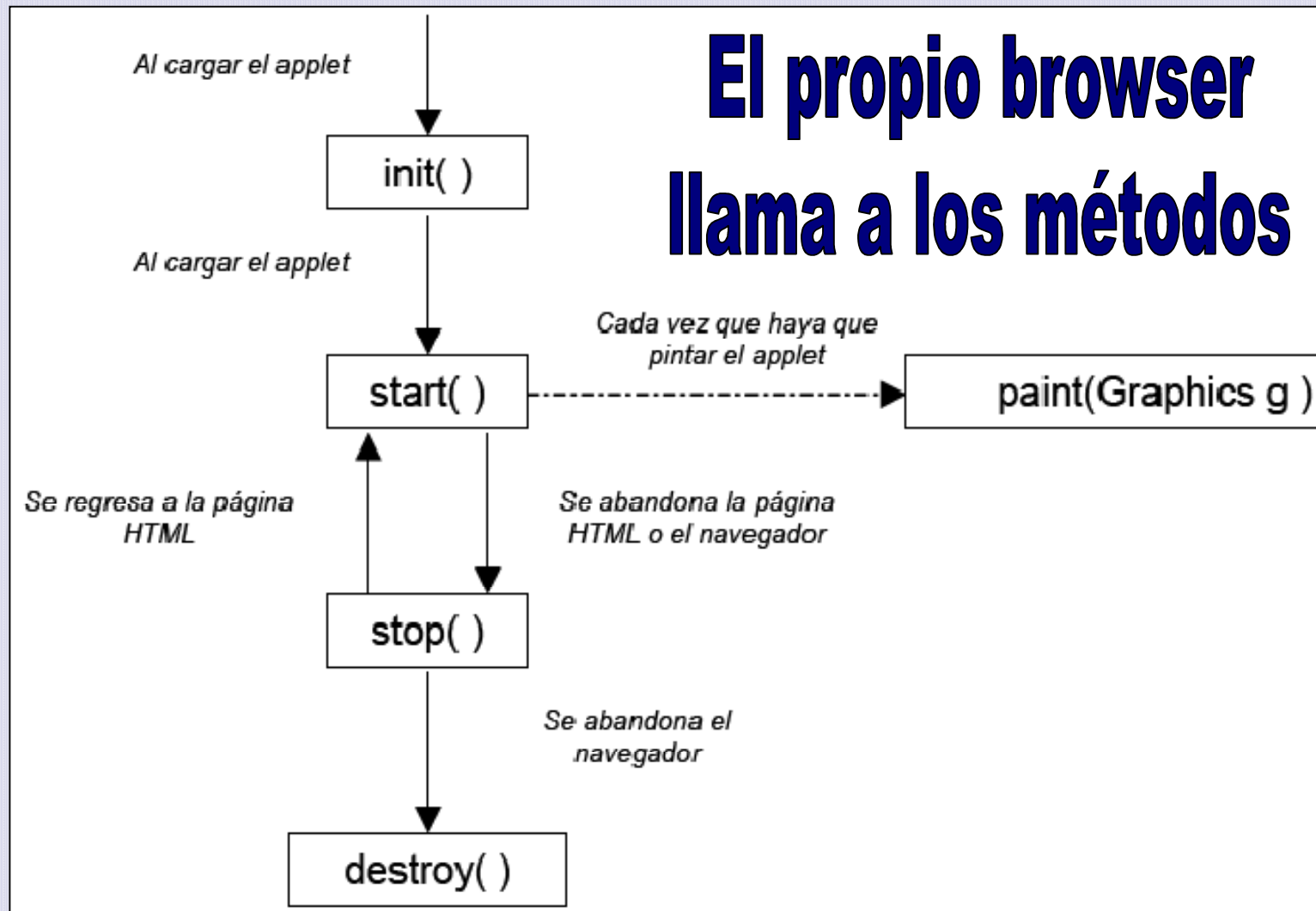
- Pequeña aplicación que se ejecutará en un browser al cargar una página HTML
- El fichero HTML contendrá etiquetas con lanzarán al Applet
- El Applet vendrá en un .class que será descargado desde el servidor Web.
- Es un contenedor AWT



## ■ Applets como contenedor

- Se puede aplicar todo lo aprendido con AWT
- Los Applets no tienen ventana propia
  - Se ejecutan en la propia ventana del browser
- **No** tienen método main
- Restricciones por seguridad
  - Sólo pueden leer/escribir ficheros almacenados en el servidor donde reside el Applet
  - Sólo pueden leer cierta información del usuario.
- Puede contener sonido e imágenes.

## ■ Ciclo de vida de un Applet



## ■ Métodos para pintar un Applet

### □ *paint(Graphics g)*

- Llamado la primera vez que se pinta el Applet
- g = Contexto para dibujar.
  - Lo crea el browser

### □ *repaint( )*

- Indica que el contenido ha cambiado

### □ *update()*

- Borra por completo el anterior contenido

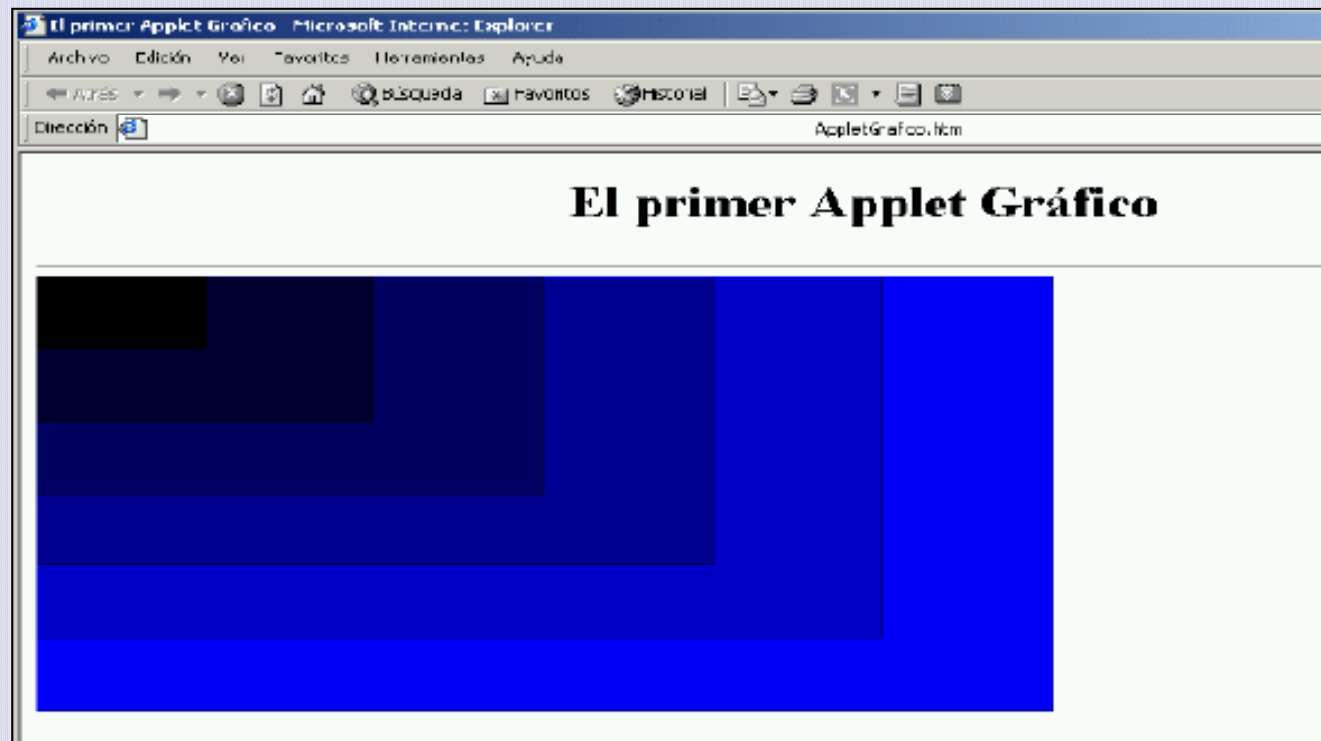
## ■ Mi primer Applet



```
<APPLET CODE="PrimerApplet.class" WIDTH=600 HEIGHT=300>  
</APPLET>  
  
</BODY>  
</HTML>
```

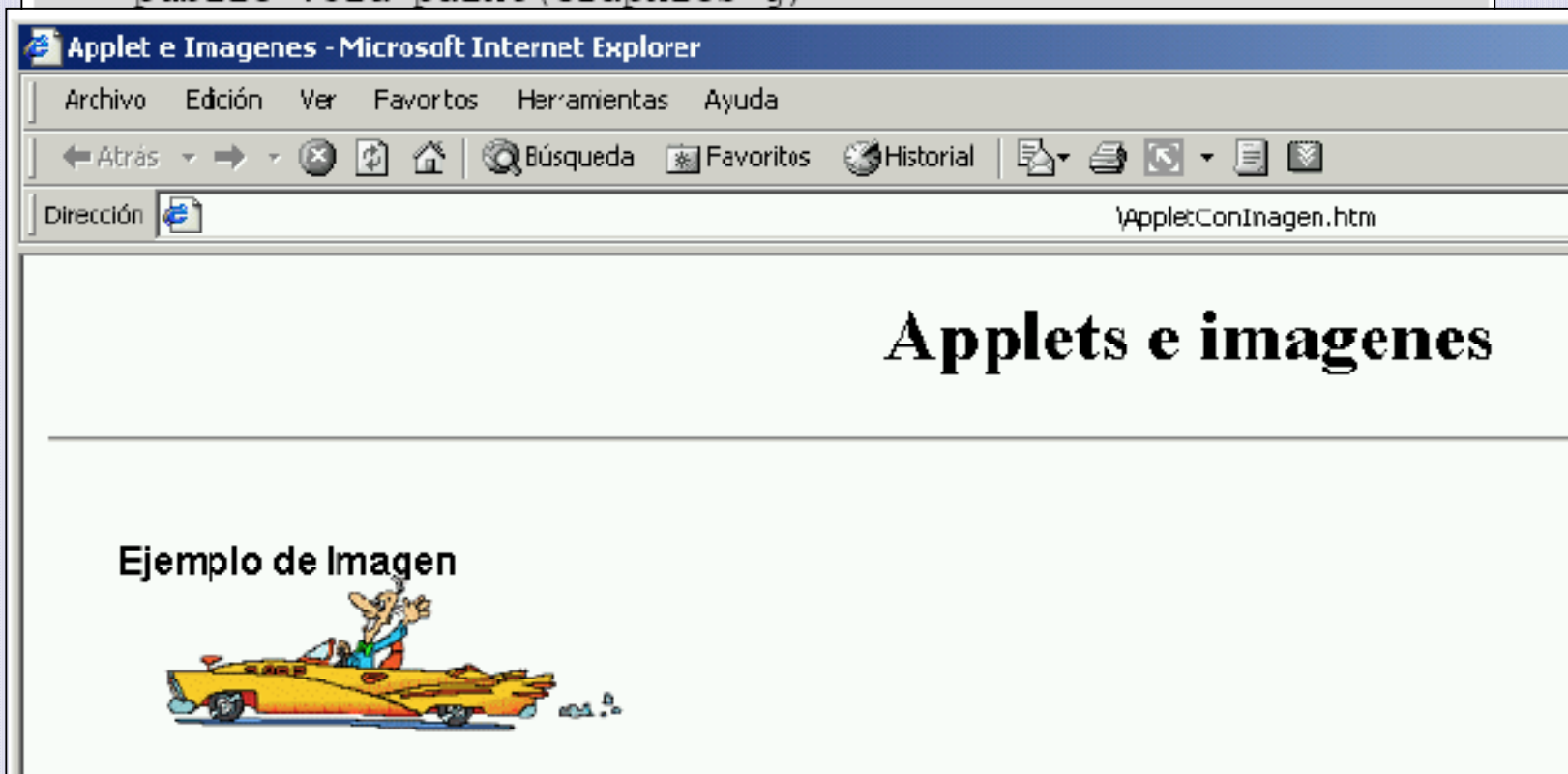
## ■ La clase *Graphics*

- Realiza la mayoría de tareas gráficas.
  - Dibujar áreas rectangulares
  - Representar imágenes
  - Establecer los colores de las figuras



## ■ Ejemplo cargar una imagen

```
import java.awt.*;  
import java.applet.Applet;  
  
public class AppletConImagen extends Applet  
{  
    public void paint(Graphics g)
```



## ■ Applets con parámetros

- Permiten personalizar los applets al igual que las variables/atributos de clase.
- Código para recoger el valor del parámetro:
  - **String getParameter(String nombreParametro)**
  - Hacerlo dentro del método init()
- Código HTML para establecer el parámetro:

```
...  
<APPLET CODE="Codigo.class">  
    <PARAM NAME="nombreParametro1" VALUE="valor1">  
    ...  
    <PARAM NAME=" nombreParametroN" VALUE="valorN">  
</APPLET>  
...
```

## ■ Ejemplo parámetros Applets



- Ejercicio práctico: Applets / AWT
  - Implementar Applet que pinte los datos introducidos en un formulario AWT

**Applet y AWT**

Nombre:

Apellidos:

Sexo  
 Masculino  
 Femenino

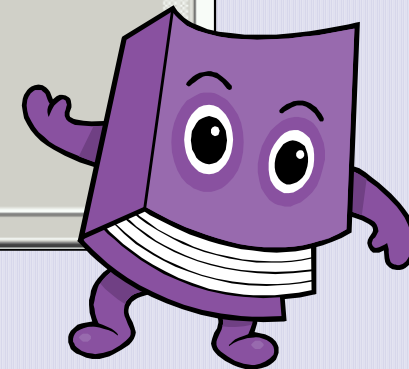
Fecha de Nacimiento:

**Datos introducidos**

Nombre: Enrique  
Apellidos: González Crespo  
Sexo: Masculino  
Año de nacimiento: 1971

Java Applet Window

■ .... Suerte!!!



## ■ Solución : Applet

```
import java.awt.* ;
import java.awt.event.*;          import java.applet.Applet;
public class AppletAWT extends Applet{
    private TextField txtNombre,txtApellidos;
    private CheckboxGroup sexo;
    private Checkbox masculino, femenino;
    private Choice lstNacimiento;
    private Button btnAceptar, btnRestaurar;
    public void init(){
        this.setName("AppletDatosPersonales");
        this.setLayout(new GridLayout(5,1,5,5));
        txtNombre = new TextField(25);
        Panel pNombre = new Panel();
        pNombre.setLayout(new FlowLayout(FlowLayout.LEFT,5,5));
        pNombre.add(new Label("Nombre"));
        pNombre.add(txtNombre);
        txtApellidos = new TextField(50);
        Panel pApellidos = new Panel();
        pApellidos.setLayout(new FlowLayout(FlowLayout.LEFT,5,5));
        pApellidos.add(new Label("Apellidos"));
        pApellidos.add(txtApellidos);
        sexo = new CheckboxGroup();
        masculino = new Checkbox("Masculino",sexo,true);
        femenino = new Checkbox("Femenino",sexo,false);    ...
    }
}
```

## ■ Solución: Eventos

```

btrAceptar = new Button("Aceptar");
btrA private void restaurar() {
n     txtNombre.setText("");
     txtApellidos.setText("");
     masculino.setState(true);
     lstNacimiento.select(0);
     }
} private void mostrarDatos() {
btrR     final Dialog dlg = new Dialog(
btrR         new Frame(), "Datos introducidos");
n     TextArea datos = new TextArea("Nombre: " + txtNombre.getText()
n         + "\n" + "Apellidos: " + txtApellidos.getText()
         + "\n" + "SEXO: " + sexo.getSelectedCheckbox().getLabel()
         + "\n" + "Año de nacimiento "
         + lstNacimiento.getSelectedItem());
     datos.setEditable(false);
     dlg.add(datos);
     dlg.setModal(true);
     dlg.addWindowListener(
         new WindowAdapter() {
             public void windowClosing(WindowEvent e) {
                 dlg.dispose();
             }
         });
     dlg.pack();
     dlg.show();
     }
}

```

## ■ Solución : Página HTML

```
<HTML>
  <HEAD>
    <TITLE>Applet y AWT</TITLE>
  </HEAD>
  <BODY>
    <H1 ALIGN="CENTER">Applet y AWT</H1>
    <HR>
    <APPLET CODE="AppletAWT.class" WIDTH=600 HEIGHT=300>
  </APPLET>
  </BODY>
</HTML>
```



# Conclusiones

1. Programación Orientada a Objetos
2. Introducción y Sintaxis Java
3. Sentencias Control Flujo
4. POO en Java
5. Relaciones entre Objetos
6. Polimorfismo, abstracción e interfaces
7. Excepciones
8. **Conceptos avanzados**

## ■ Applets

- Introducción
- Ciclo de Vida
- Métodos
- Clase Graphics
- Imágenes
- Parámetros
- Applets / AWT