

JAVA STANDARD EDITION

Oscar Chávez-Bosquez

<http://chavezbosquez.github.io>

Contenido del módulo

1. Interfaces gráficas con Java Swing
2. Diseño de contenedores (*Layout managers*)
3. Componentes Swing
4. Manejo de eventos (*Listeners*)
5. Base de datos relacionales
6. Java Persistence API (JPA)
7. Modelo-Vista-Controlador (MVC)
8. Aplicaciones CRUD

Hoy veremos...

1. *Refresh*
2. Introducción a Swing
3. Diseño de contenedores
4. Componentes Swing

Refresh...

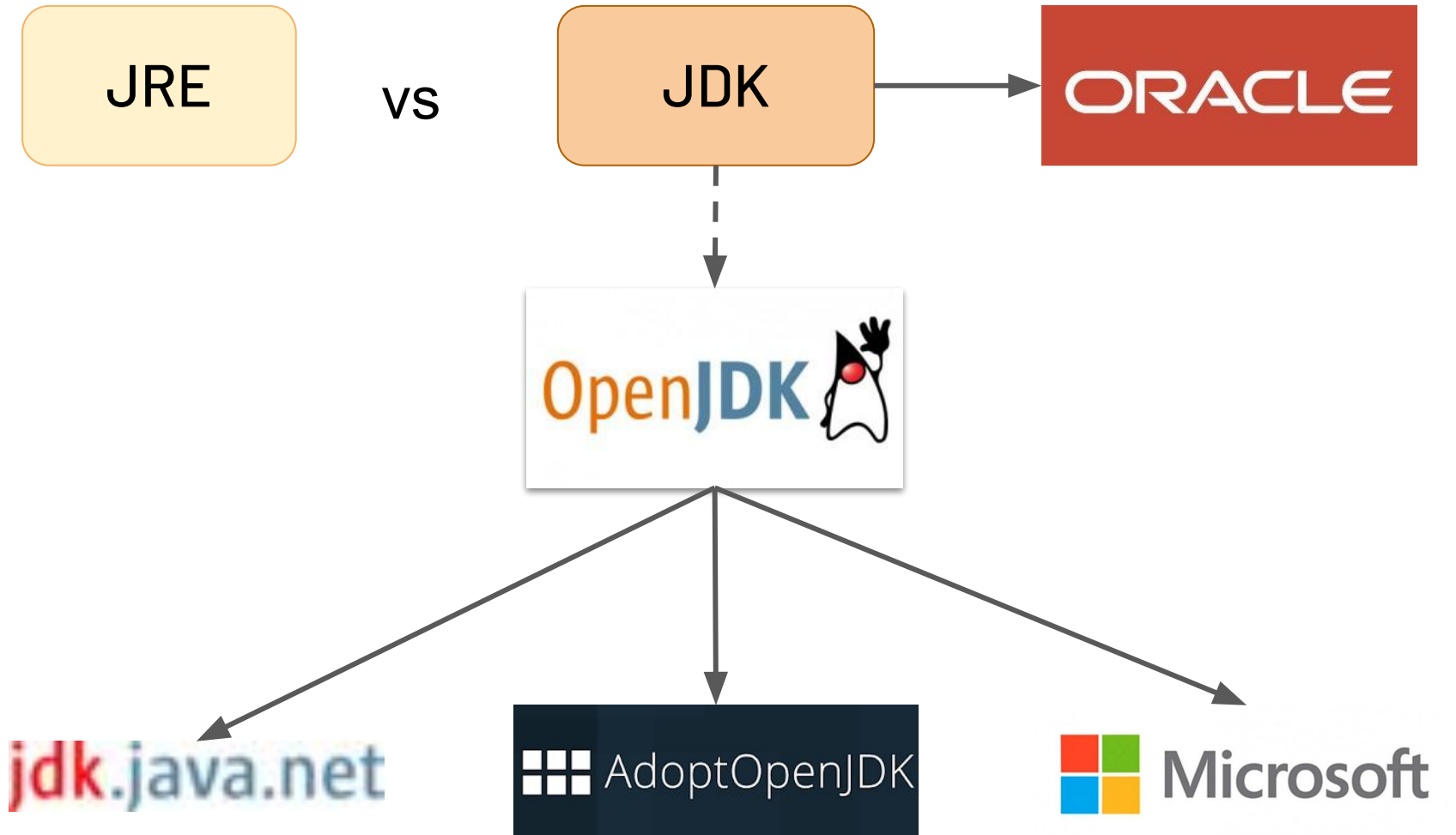
Digital Learning SL: [Test de Java y POO](#)

Cibertest: [Programación en Java Básico](#)

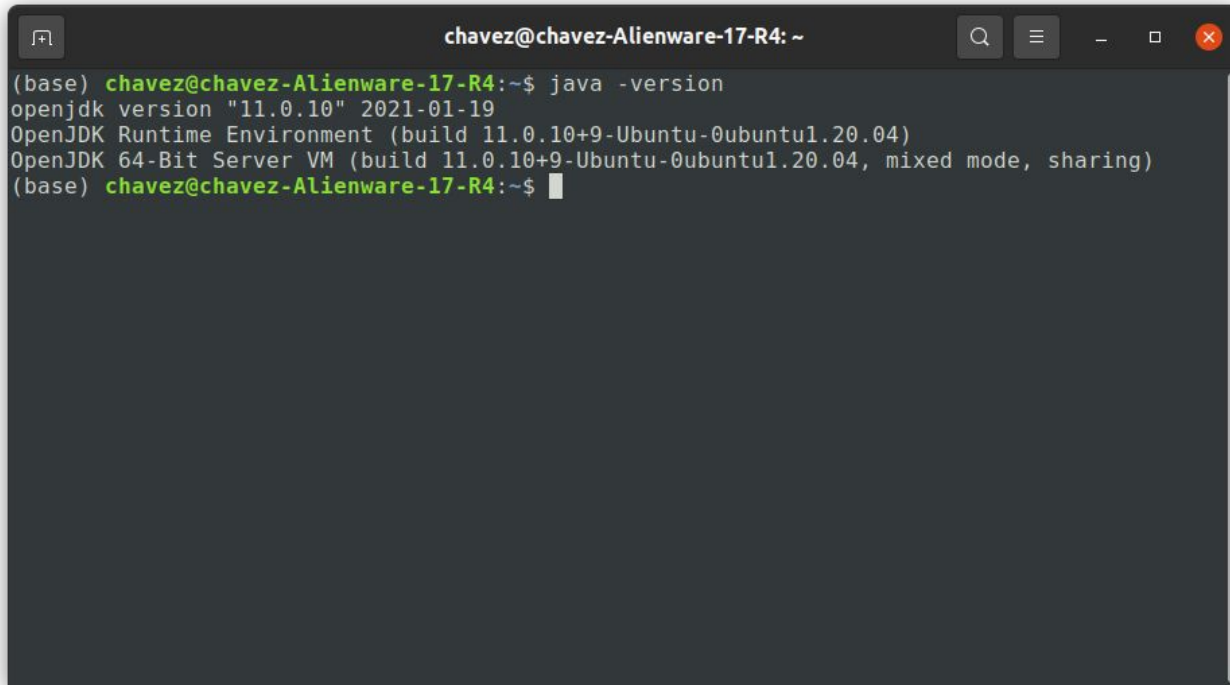
Daypo: [Test de Java, nivel básico](#)

[`https://chavezbosquez.github.io/java.html`](https://chavezbosquez.github.io/java.html)

Which Java SDK Should You Use?



Versión de Java

A terminal window with a dark background and light text. The title bar at the top reads 'chavez@chavez-Alienware-17-R4: ~'. The terminal shows the command 'java -version' being executed, followed by its output: 'openjdk version "11.0.10" 2021-01-19', 'OpenJDK Runtime Environment (build 11.0.10+9-Ubuntu-0ubuntu1.20.04)', and 'OpenJDK 64-Bit Server VM (build 11.0.10+9-Ubuntu-0ubuntu1.20.04, mixed mode, sharing)'. The prompt '(base) chavez@chavez-Alienware-17-R4:~\$' is visible at the bottom.

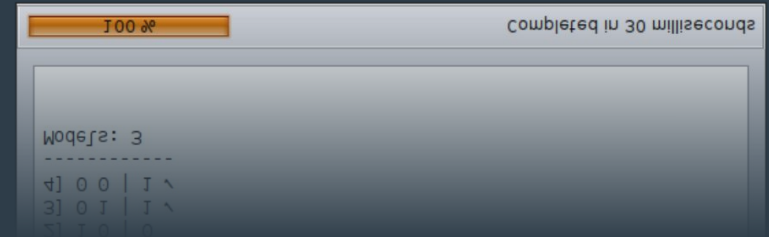
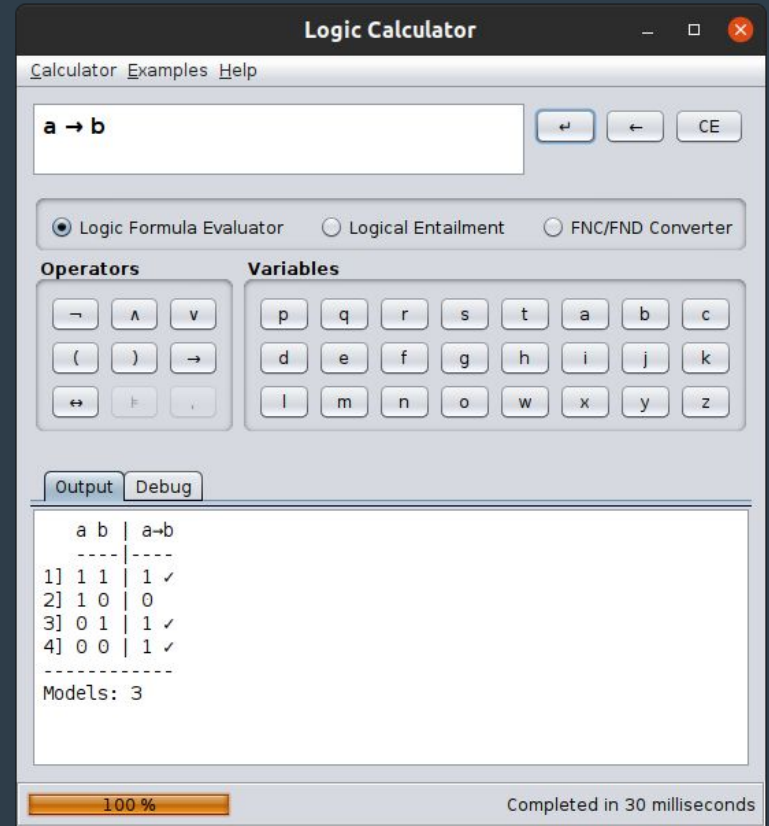
```
(base) chavez@chavez-Alienware-17-R4:~$ java -version
openjdk version "11.0.10" 2021-01-19
OpenJDK Runtime Environment (build 11.0.10+9-Ubuntu-0ubuntu1.20.04)
OpenJDK 64-Bit Server VM (build 11.0.10+9-Ubuntu-0ubuntu1.20.04, mixed mode, sharing)
(base) chavez@chavez-Alienware-17-R4:~$
```

Best Java IDE 2021

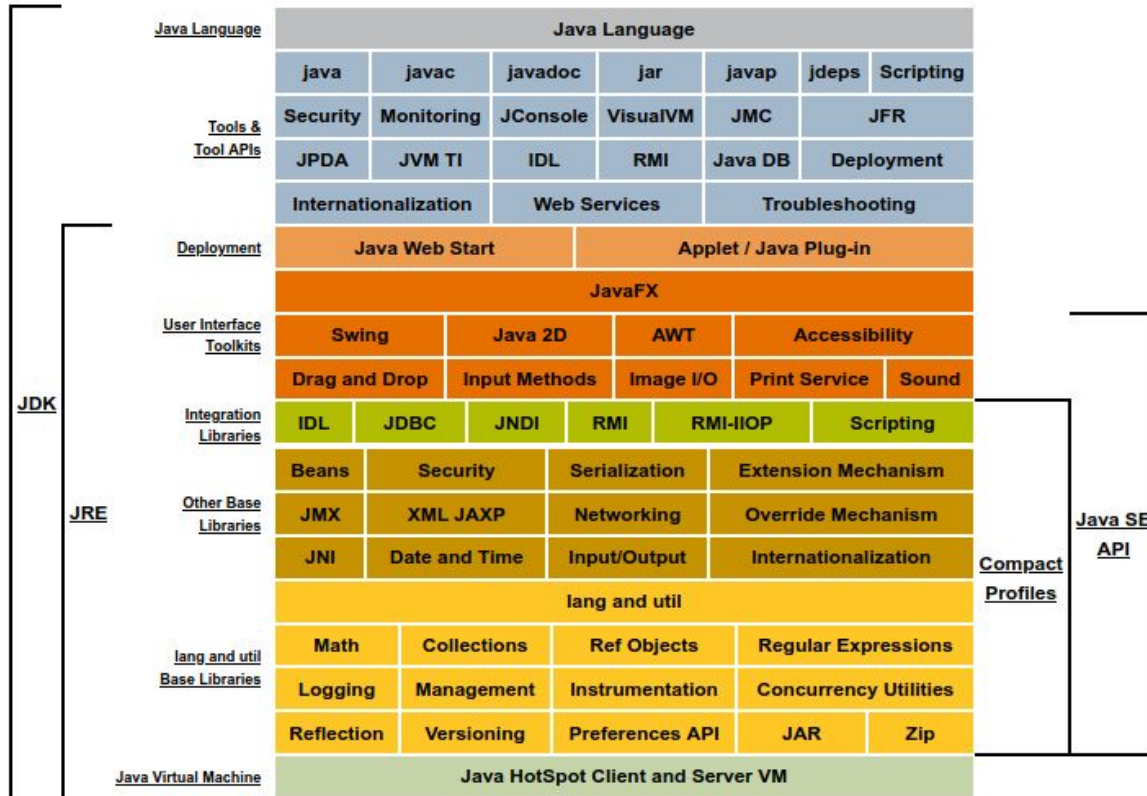
Best Java IDEs



Introducción a Swing



Swing: API de Java para desarrollo de UI



API = Application Programming Interface

UI = User Interface

Swing: API de Java para desarrollo de UI

OVERVIEW PACKAGE CLASS USE **TREE** DEPRECATED INDEX HELP

Java™ Platform
Standard Ed. 8

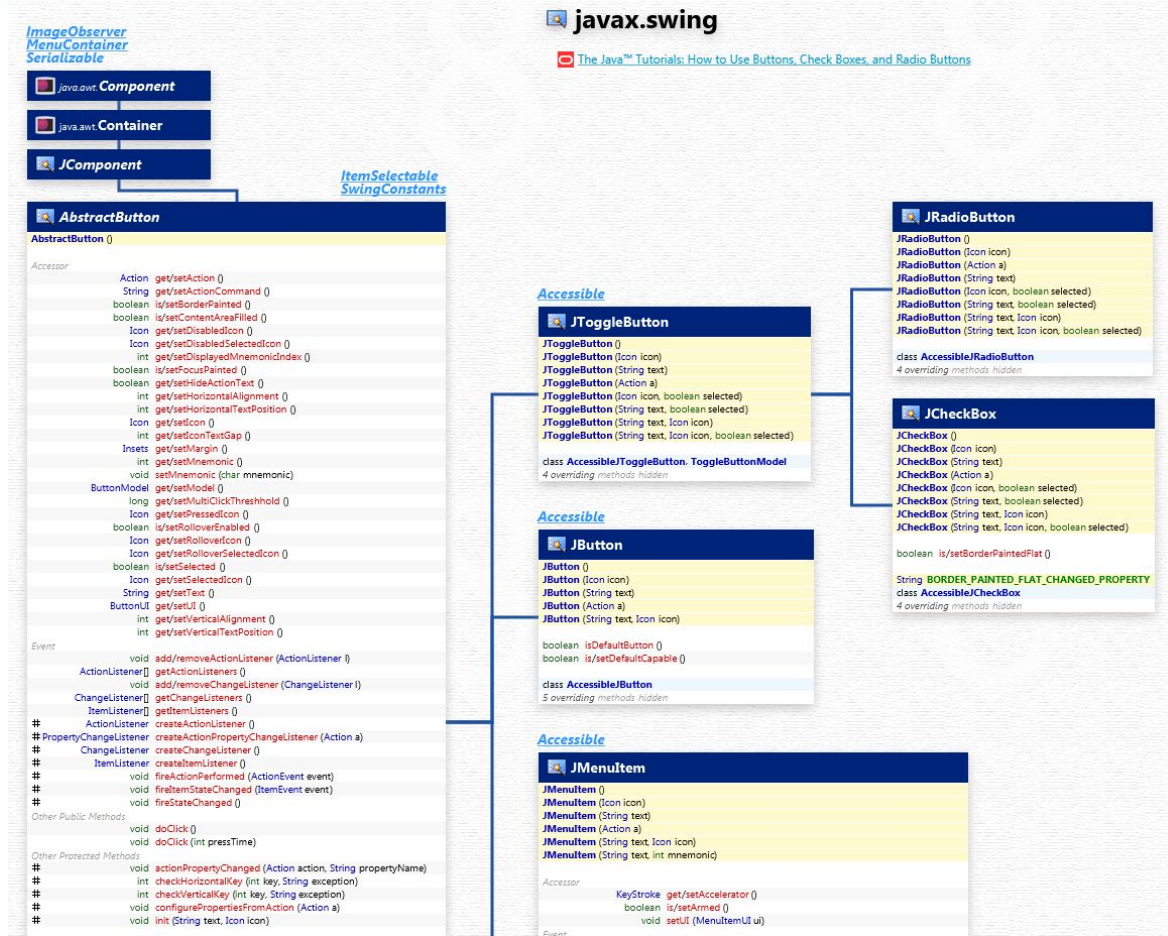
PREV NEXT FRAMES NO FRAMES ALL CLASSES

Hierarchy For All Packages

Package Hierarchies:

java.applet, java.awt, java.awt.color, java.awt.datatransfer, java.awt.dnd, java.awt.event, java.awt.font, java.awt.geom, java.awt.im, java.awt.im.spi, java.awt.image, java.awt.image.renderable, java.awt.print, java.beans, java.beans.beancontext, java.io, java.lang, java.lang.annotation, java.lang.instrument, java.lang.invoke, java.lang.management, java.lang.ref, java.lang.reflect, java.math, java.net, java.nio, java.nio.channels, java.nio.channels.spi, java.nio.charset, java.nio.charset.spi, java.nio.file, java.nio.file.attribute, java.nio.file.spi, java.rmi, java.rmi.activation, java.rmi.dgc, java.rmi.registry, java.rmi.server, java.security, java.security.acl, java.security.cert, java.security.interfaces, java.security.spec, java.sql, java.text, java.text.spi, java.time, java.time.chrono, java.time.format, java.time.temporal, java.time.zone, java.util, java.util.concurrent, java.util.concurrent.atomic, java.util.concurrent.locks, java.util.function, java.util.jar, java.util.logging, java.util.prefs, java.util.regex, java.util.spi, java.util.stream, java.util.zip, javax.accessibility, javax.activation, javax.activity, javax.annotation, javax.annotation.processing, javax.crypto, javax.crypto.interfaces, javax.crypto.spec, javax.imageio, javax.imageio.event, javax.imageio.metadata, javax.imageio.plugins.bmp, javax.imageio.plugins.jpeg, javax.imageio.spi, javax.imageio.stream, javax.jws, javax.jws.soap, javax.lang.model, javax.lang.model.element, javax.lang.model.type, javax.lang.model.util, javax.management, javax.management.loading, javax.management.modelmbean, javax.management.monitor, javax.management.openmbean, javax.management.relation, javax.management.remote, javax.management.remote.rmi, javax.management.timer, javax.naming, javax.naming.directory, javax.naming.event, javax.naming ldap, javax.naming.spi, javax.net, javax.net.ssl, javax.print, javax.print.attribute, javax.print.attribute.standard, javax.print.event, javax.rmi, javax.rmi.CORBA, javax.rmi.ssl, javax.script, javax.security.auth, javax.security.auth.callback, javax.security.auth.kerberos, javax.security.auth.login, javax.security.auth.spi, javax.security.auth.x500, javax.security.cert, javax.security.sasl, javax.sound.midi, javax.sound.midi.spi, javax.sound.sampled, javax.sound.sampled.spi, javax.sql, javax.sql.rowset, javax.sql.rowset.serial, javax.sql.rowset.spi, javax.swing, javax.swing.border, javax.swing.colorchooser, javax.swing.event, javax.swing.filechooser, javax.swing.plaf, javax.swing.plaf.basic, javax.swing.plaf.metal, javax.swing.plaf.multi, javax.swing.plaf.nimbus, javax.swing.plaf.synth, javax.swing.table, javax.swing.text, javax.swing.text.html, javax.swing.text.html.parser, javax.swing.text.rtf, javax.swing.tree, javax.swing.undo, javax.tools, javax.transaction, javax.transaction.xa, javax.xml, javax.xml.bind, javax.xml.bind.annotation, javax.xml.bind.annotation.adapters, javax.xml.bind.attachment, javax.xml.bind.helpers, javax.xml.bind.util, javax.xml.crypto, javax.xml.crypto.dom, javax.xml.crypto.dsig, javax.xml.crypto.dsig.dom, javax.xml.crypto.dsig.keyinfo, javax.xml.crypto.dsig.spec, javax.xml.datatype, javax.xml.namespace, javax.xml.parsers, javax.xml.soap, javax.xml.stream, javax.xml.stream.events, javax.xml.stream.util, javax.xml.transform, javax.xml.transform.dom, javax.xml.transform.sax, javax.xml.transform.stax, javax.xml.transform.stream, javax.xml.validation, javax.xml.ws, javax.xml.ws.handler, javax.xml.ws.handler.soap, javax.xml.ws.http, javax.xml.ws.soap, javax.xml.ws.spi, javax.xml.ws.spi.http, javax.xml.ws.wsaddressing, javax.xml.xpath, org.ietf.jgss, org.omg.CORBA, org.omg.CORBA_2_3, org.omg.CORBA_2_3.portable, org.omg.CORBA.DynAnyPackage, org.omg.CORBA.ORBPackage, org.omg.CORBA.portable, org.omg.CORBA.TypeCodePackage, org.omg.CosNaming, org.omg.CosNaming.NamingContextExtPackage, org.omg.CosNaming.NamingContextPackage, org.omg.Dynamic, org.omg.DynamicAny, org.omg.DynamicAny.DynAnyFactoryPackage, org.omg.DynamicAny.DynAnyPackage, org.omg.IOP, org.omg.IOP.CodecFactoryPackage, org.omg.IOP.CodecPackage, org.omg.Messaging, org.omg.PortableInterceptor, org.omg.PortableInterceptor.ORBInitInfoPackage, org.omg.PortableServer, org.omg.PortableServer.CurrentPackage, org.omg.PortableServer.POAManagerPackage, org.omg.PortableServer.POAPackage, org.omg.PortableServer.portable, org.omg.PortableServer.ServantLocatorPackage, org.omg.SendingContext, org.omg.stub.java.rmi, org.w3c.dom, org.w3c.dom.bootstrap, org.w3c.dom.events, org.w3c.dom.ls, org.w3c.dom.views, org.xml.sax, org.xml.sax.ext, org.xml.sax.helpers

Swing: API de Java para desarrollo de UI



Swing: API de Java para desarrollo de UI

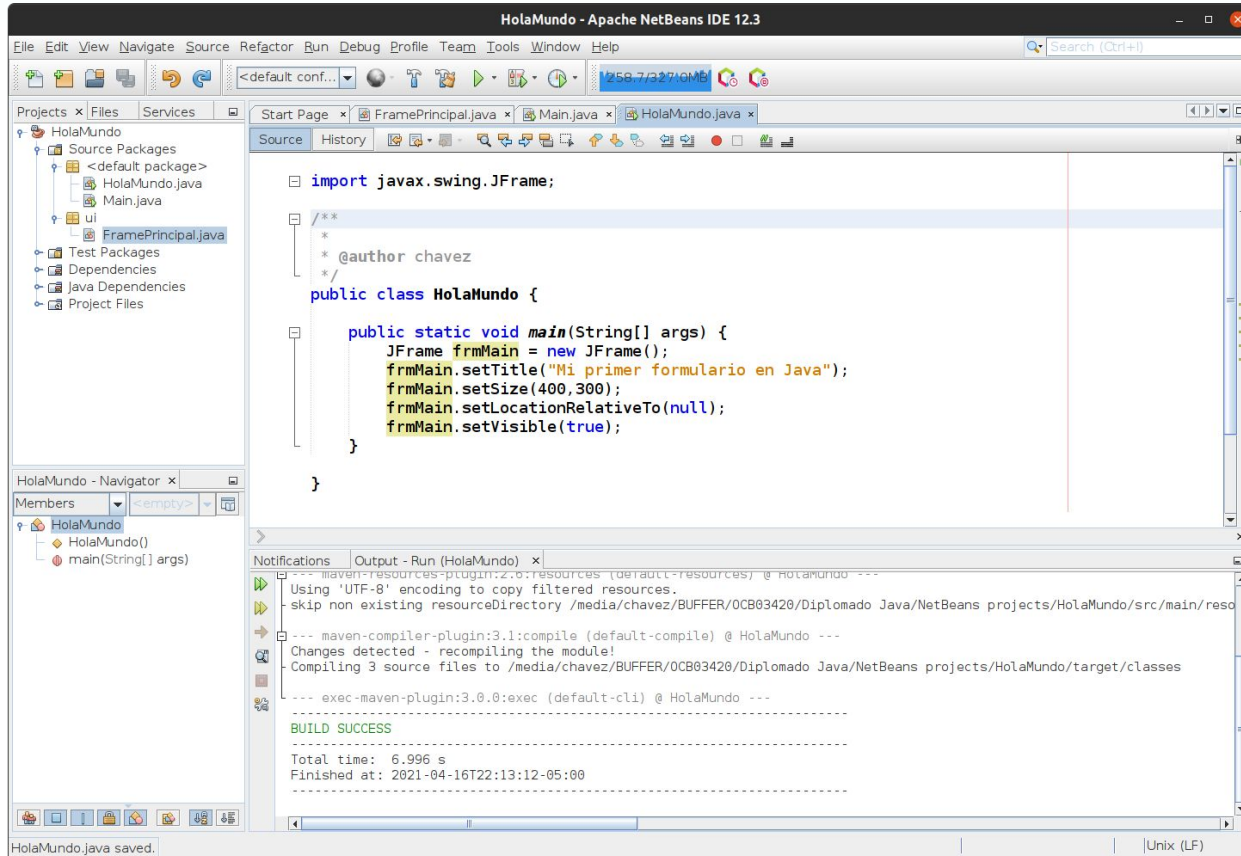
- Las interfaces gráficas ofrecen al usuario ventanas, cuadros de diálogo, barras de herramientas, botones, listas desplegables y muchos otros elementos con los que ya estamos muy acostumbrados a tratar.
- Las aplicaciones son conducidas por eventos y se desarrollan haciendo uso de las clases de la API.

Programación Orientada a Eventos

Swing: API de Java para desarrollo de UI

- Proporciona una biblioteca de clases e interfaces para el desarrollo de interfaces gráficas de usuario .
- Interfaces gráficas que tienen una apariencia y se comportan de forma semejante en todas las plataformas en las que se ejecuten.
- La estructura básica de la biblioteca gira en torno a componentes y contenedores:
 - Los contenedores contienen componentes y son componentes a su vez.
 - Los eventos pueden tratarse tanto en contenedores como en componentes.

NetBeans IDE



Diseño de contenedores

Sistema de Evaluación Antropométrica y Alimentaria

Paciente Consulta Ayuda

Nuevo Paciente Nueva Consulta Ayuda

Nuevo paciente

Nombre(s)

Apellido(s)

Domicilio

Ciudad Estado

Ocupación

Fecha de Nacimiento

Estado civil

Género ☒ Femenino ☐ Masculino

Estatura (cm)

Peso (kg)

Actividad física (Porcentaje adicional al GEB: 0 %)

Actividad física (Porcentaje adicional al GEB: 0 %)

Peso (kg)

Contenedores

Contenedores de alto nivel:

- **JFrame**: Habitualmente se emplea para crear la ventana principal de una aplicación en Swing.
- **JDialog**: Ventanas de interacción con el usuario.

Contenedores intermedios:

- **JPanel**: Sirve para agrupar otros componentes.
- **JScrollPane**: Panel que incluye barras de desplazamiento.

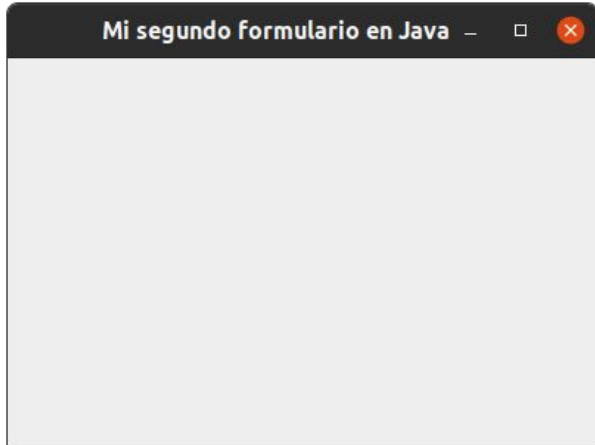
iHola Mundo!

```
public class HolaMundo {  
  
    public static void main(String[] args) {  
        JFrame frmMain = new JFrame();  
        frmMain.setTitle("Mi primer formulario en Java");  
        frmMain.setSize(400,300);  
        frmMain.setLocationRelativeTo(null);  
        frmMain.setVisible(true);  
    }  
}
```



iHola Mundo! 2.0

```
public class FramePrincipal extends JFrame {  
  
    public FramePrincipal() {  
        this.setTitle("Mi segundo formulario en Java");  
        this.setSize(400,300);  
        this.setLocationRelativeTo(null);  
    }  
  
}
```



```
public class Main {  
  
    public static void main(String[] args) {  
        FramePrincipal frmPrincipal = new FramePrincipal();  
        frmPrincipal.setVisible(true);  
    }  
  
}
```

Métodos de la clase `JFrame`

- `setTitle("Título de la ventana");`
- `setSize(400,300);`
- `setLocationRelativeTo(null);`
- `setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);`
- `setAlwaysOnTop(true);`
- `setUndecorated(true);`
- `setVisible(true);`
- `Container fondo = frame.getContentPane();`

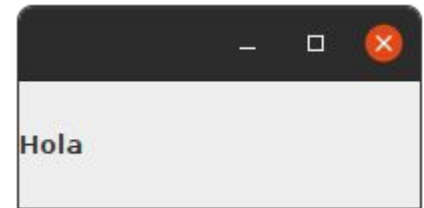
Añadiendo componentes

```
public class FramePrincipal extends JFrame {  
  
    public FramePrincipal() {  
        this.setSize(200, 100);  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.crearUI();  
    }  
  
    private void crearUI() {  
        JButton btn = new JButton("Presióname");  
        this.add(btn);  
    }  
}
```



Añadiendo componentes

```
public class FramePrincipal extends JFrame {  
  
    public FramePrincipal() {  
        this.setSize(200,100);  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.crearUI();  
    }  
  
    private void crearUI() {  
        JButton btn = new JButton("Presióname");  
        this.add(btn);  
        JLabel lbl = new JLabel("Hola");  
        this.add(lbl);  
    }  
  
}
```

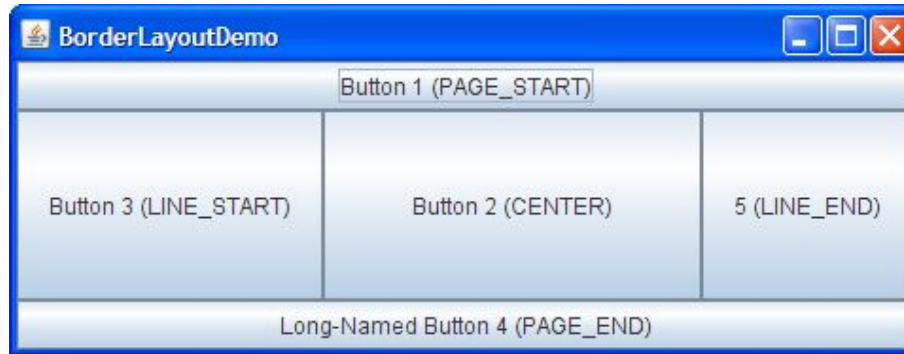


Contenedores

- Los componentes se agregan al contenedor con el método `add()`.
- El efecto de `add()` depende del esquema de disposición (*layout*) del contenedor.
- Existen diversos esquemas de disposición:
 - `BorderLayout`,
 - `FlowLayout`,
 - `GridLayout`,
 - ...
- <https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>

Contenedores

- Los contenedores requieren objetos **LayoutManager** (administradores de disposición).
- En el caso de **JFrame** y **JDialog** por default usan **BorderLayout**.



Añadiendo componentes

```
public class FramePrincipal extends JFrame {  
  
    public FramePrincipal() {  
        this.setSize(200,100);  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.crearUI();  
    }  
  
    private void crearUI() {  
        JButton btn = new JButton("Presióname");  
        this.add(btn, BorderLayout.CENTER);  
        JLabel lbl = new JLabel("Hola");  
        this.add(lbl, BorderLayout.PAGE_START);  
    }  
  
}
```



Tarea

- Desarrollar la siguiente UI:



1. `JTextField txt = new JTextField(10);`
2. `JCheckBox chk = new JCheckBox("Check");`
3. `JRadioButton rdo = new JRadioButton("Ok");`

Eventos, componentes, JFrame y JDialog

