

DESARROLLO DE APLICACIONES MULTIPLATAFORMA

Desarrollo de Interfaces

Equivalencia en créditos ECTS (BOE): 9

Tema 2: Generación de GUI a partir de documentos XML (Parte 1)

RA: Generar interfaces gráficos de usuario basados en XML utilizando herramientas específicas y adaptando el documento XML generado

Índice

1. El lenguaje XML
2. Componentes en Flash Builder
3. Eventos en Flash Builder



XML (eXtensible Markup Language)

- **XML** es un **lenguaje de marcas** utilizado para almacenar datos en forma legible. Ventajas del XML:
 - **Extensible**: Es posible extender XML con la adición de nuevas etiquetas, pudiendo continuar utilizando sin complicación alguna.
 - Mejora la **compatibilidad entre aplicaciones**, la **comunicación** en distintas **plataformas** y **Bases de Datos**.
 - Aporta **flexibilidad** para estructurar documentos.



Estructura de un documento XML

- La información de un XML se estructura en base a **etiquetas**.
- **<nombre>**Nombre del remitente
</nombre>
- Las etiquetas pueden ir acompañadas de **atributos y valores**:
<libro año="2017">
Texto **</libro>**

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE Edit_Mensaje SYSTEM "Edit_Mensaje.dtd">

<Edit_Mensaje>
  <Mensaje>
    <Remitente>
      <Nombre>Nombre del remitente</Nombre>
      <Mail>Correo del remitente </Mail>
    </Remitente>
    <Destinatario>
      <Nombre>Nombre del destinatario</Nombre>
      <Mail>Correo del destinatario</Mail>
    </Destinatario>
    <Texto>
      <Asunto>
        Este es mi documento con una estructura muy sencilla
        no contiene atributos ni entidades...
      </Asunto>
      <Parrafo>
        Este es mi documento con una estructura muy sencilla
        no contiene atributos ni entidades...
      </Parrafo>
    </Texto>
  </Mensaje>
</Edit_Mensaje>
```

DTD

- Una **definición de tipo de documento** (DTD) es una descripción de estructura y sintaxis de un documento XML. Su función básica es la descripción de la estructura de datos, para usar una estructura común y mantener la consistencia entre todos los documentos que utilicen la misma DTD.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE Edit_Mensaje SYSTEM "Edit_Mensaje.dtd">

<Edit_Mensaje>
  <Mensaje>
    <Remitente>
      <Nombre>Nombre del remitente</Nombre>
      <Mail>Correo del remitente </Mail>
    </Remitente>
    <Destinatario>
      <Nombre>Nombre del destinatario</Nombre>
      <Mail>Correo del destinatario</Mail>
    </Destinatario>
    <Texto>
      <Asunto>
        Este es mi documento con una estructura muy sencilla
        no contiene atributos ni entidades...
      </Asunto>
      <Parrafo>
        Este es mi documento con una estructura muy sencilla
        no contiene atributos ni entidades...
      </Parrafo>
    </Texto>
  </Mensaje>
</Edit_Mensaje>
```

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- Este es el DTD de Edit_Mensaje -->

<!ELEMENT Mensaje (Remitente, Destinatario, Texto)*>
<!ELEMENT Remitente (Nombre, Mail)>
<!ELEMENT Nombre (#PCDATA)>
<!ELEMENT Mail (#PCDATA)>
<!ELEMENT Destinatario (Nombre, Mail)>
<!ELEMENT Nombre (#PCDATA)>
<!ELEMENT Mail (#PCDATA)>
<!ELEMENT Texto (Asunto, Parrafo)>
<!ELEMENT Asunto (#PCDATA)>
<!ELEMENT Parrafo (#PCDATA)>
```

Ejercicio

- A partir del siguiente XML, obtener su DTD:

```
<?xml version="1.0"?>
- <enviNFe xmlns="http://www.portalfiscal.inf.br/nfe" versao="1.10">
  <idLote>71</idLote>
  - <NFe>
    - <infNFe versao="2.00" Id="NFe3508059978">
      <cUF>35</cUF>
      <XXX>aaa</XXX>
      <cNF>518005127</cNF>
      <cNF> </cNF>
    </infNFe>
    - <Emit>
      518005127
      <CNPJ>08133339000152</CNPJ>
      <xNome>Empresa X</xNome>
      <xFant>XXX</xFant>
    - <enderEmit>
      <xLgr>R. XX</xLgr>
      <nro>99</nro>
    </enderEmit>
    <IE>143333334114</IE>
    <IM>35544447</IM>
  </Emit>
</NFe>
</enviNFe>
```

Documentos XML bien formados

- Aquellos que cumplen con todas las definiciones básicas de formato y pueden analizarse correctamente por cualquier analizador sintáctico o **parser** que cumpla con la norma:
 - Los documentos han de seguir una estructura **estrictamente jerárquica** con lo que respecta a las etiquetas que delimitan sus elementos. Las etiquetas deben estar correctamente **anidadas**.
 - Solo pueden tener **un elemento inicial**, es decir, partir de una **raíz**.
 - Los **valores de los atributos en XML** siempre deben estar encerrados entre **comillas**.
 - El XML es **sensible a mayúsculas y minúsculas**.
 - Las construcciones como etiquetas, referencias de entidad y declaraciones se denominan **marcas**.

Partes de un documento XML

- **Prólogo:** Aunque no es obligatorio, los documentos XML pueden empezar con unas líneas que describen la versión XML, el tipo de documento y otras consideraciones.

EJEMPLO: `<?xml version="1.0" encoding="UTF-8"?>`

- **Cuerpo:** Es obligatorio y debe contener solo un elemento raíz, característica indispensable también para que el documento esté bien formado.

`<Caracteristicas>`

`(...)`

`</ Caracteristicas >`

Partes de un documento XML (2)

- **Elementos:** Los elementos XML pueden tener contenido (más elementos, caracteres o ambos), o bien ser elementos vacíos.
- **Atributos:** Los elementos pueden tener atributos, que son una forma de incorporar características o propiedades a los elementos de un documento, deben ir entre comillas. Por ejemplo, un elemento «estudiante» puede tener un atributo «tipo» y un atributo «pelo», con valores «estudia mucho» y «rubio» respectivamente.
`<Estudiante tipo="estudia mucho" pelo="rubio">Juan</Estudiante>`
- **Comentarios:** A modo informativo que han de ser ignorados por el procesador. Los comentarios en XML tienen el siguiente formato:
`<!-- Esto es un comentario --> <!-- Otro comentario -->`

XML en la generación de GUI

- **MXML**: es un lenguaje descriptivo desarrollado inicialmente por Macromedia para la plataforma FLEX de Adobe.
- **XAML**: es un lenguaje de marcado basado en XML desarrollado por Microsoft.
- Diseñar GUIs mediante XML permite realizar **la separación entre las capas de presentación y la capa lógica**.



Ejercicio

- Instalar el IDE Flash Builder
 - Estudiar programación en lenguaje MXML



Flash - Prueba/src/Prueba.mxml - Flash Builder

File Edit Source Navigate Search Project Data Run Window Help

Package Explorer

Prueba.mxml

Outline

- Application
 - Script
 - Declarations
 - RadioButtonGroup radiogroup1
 - Image (assets/homer.jpg)
 - Label (Nombre)
 - Label (Apellidos)
 - Label (Fecha nacimiento)
 - TextInput nombre
 - TextInput tel
 - Label (Nº de teléfono)
 - TextInput
 - Label (Sexo)
 - VGroup
 - RadioButton (Mujer)
 - RadioButton (Hombre)
 - CheckBox (Familia numerosa)
 - CheckBox (Carné Joven)
 - Label (Situación Laboral)
 - ComboBox
 - dataProvider
 - Button (Aceptar)
 - Button (Cancelar)
 - DateField

Source

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
  xmlns:s="library://ns.adobe.com/flex/spark"
  xmlns:mx="library://ns.adobe.com/flex/mx"
  width="679" height="374" minWidth="955" minHeight="588"
  pageTitle="Formulario de Inscripción">

  <fx:Script>
    <![CDATA[
      protected function button1_clickHandler(event:MouseEvent):void
      {
        // TODO Auto-generated method stub

      }
    ]]>
  </fx:Script>

  <fx:Declarations>
    <s:RadioButtonGroup id="radiogroup1"/>
    <!-- Place non-visual elements (e.g., services, value providers) here -->
  </fx:Declarations>
  <s:Image x="56" y="50" width="293" height="250" scaleMode="stretch"
    source="assets/homer.jpg"/>
  <s:Label x="424" y="73" text="Nombre:" textAlign="right"/>
  <s:Label x="418" y="108" text="Apellidos:"/>
  <s:Label x="371" y="142" text="Fecha nacimiento:"/>
  <s:TextInput id="nombre" x="481" y="63"/>
  <s:TextInput id="tel" x="481" y="168" displayAsPassword="true"/>
  <s:Label x="393" y="178" text="Nº de teléfono:"/>
</s:Application>
```

Design

FB

ADOBE FLASH BUILDER 4.6 PREMIUM

Create

- New Flex Project...
- New Flex Mobile Project...
- New ActionScript Project...
- New ActionScript Mobile Project...
- New MXML Application...
- New MXML Component...
- New ActionScript File...
- New ActionScript Class...

Recent Files

No files have been opened yet

Resources

- Build a mobile app in five minutes
- Flex Test Drive
- Flex in a Week
- Documentation: Using Flash Builder
- Flash Builder Tips and Tricks

Getting Started With Flash Builder

Articles and Tutorials

- Building iOS Applications
- Data Visualization App
- What's new in Flash Builder 4.5
- Build your first mobile Flex application
- Packaging mobile applications
- One Developer Platform, Many Devices
- Create your first Flex and Facebook application

Flex SDK & Flash Builder News

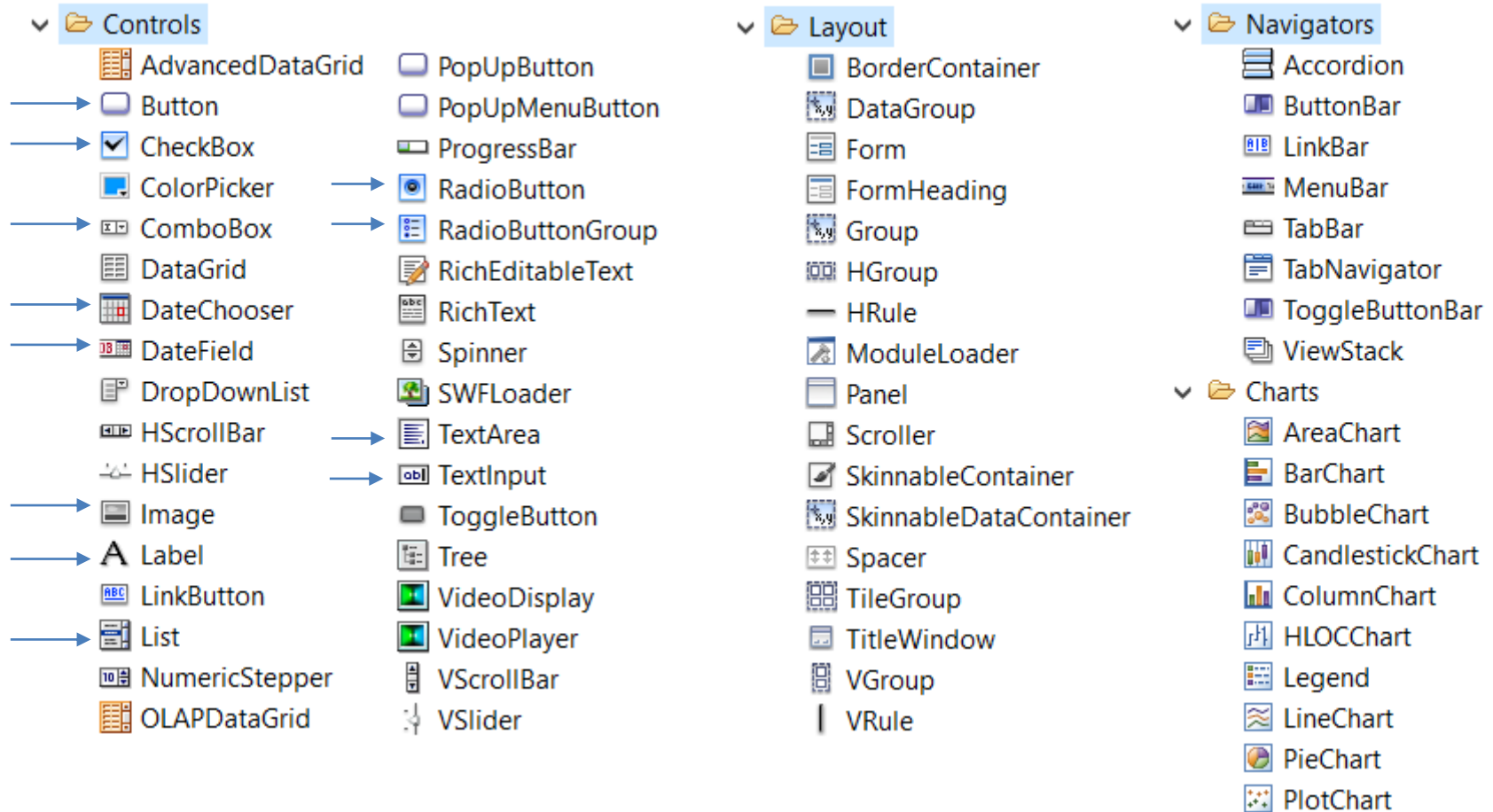
- Flash-Based Machinarium Game is Now #1 iPad App**
The Flex SDK and Flash Builder teams are extremely excited to advise that the previously-announced June updates...
- Cool Stuff with the Flash Platform**
This week features a number of showcase mobile applications, many from Adobe evangelists...
- Flex: where we are headed**
It has been about three months since we shipped Flex 4.5 SDK and a corresponding update to Flash Builder...

AIR Native Extensions

- Vibration**
Use the vibration feature of an Android or iOS mobile device.
- Network Info**
Retrieves info about the network interface on an iOS device.
- Local Notifications**

SHOW ADVANCED TIPS

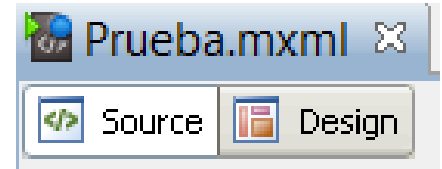
Componentes en Flash Builder



Ventanas de Código y de Diseño

- Flash Builder permite trabajar, a la vez con:

- Ventana de código (Source)
- Ventana de diseño (Design)




- En la ventana de diseño se encuentra el elemento Application, que se corresponde con la ventana principal de la ejecución.
- En la ventana de código se encuentra la implementación de la GUI en MXML.
- La ejecución del aplicativo se realizará en navegador Web.**
Para cambiar el navegador: Window → Preferences → General → Web Browser

Ejemplo de GUI en Flash Builder



Nombre:


Apellidos:

Fecha nacimiento: 

Nº de teléfono:

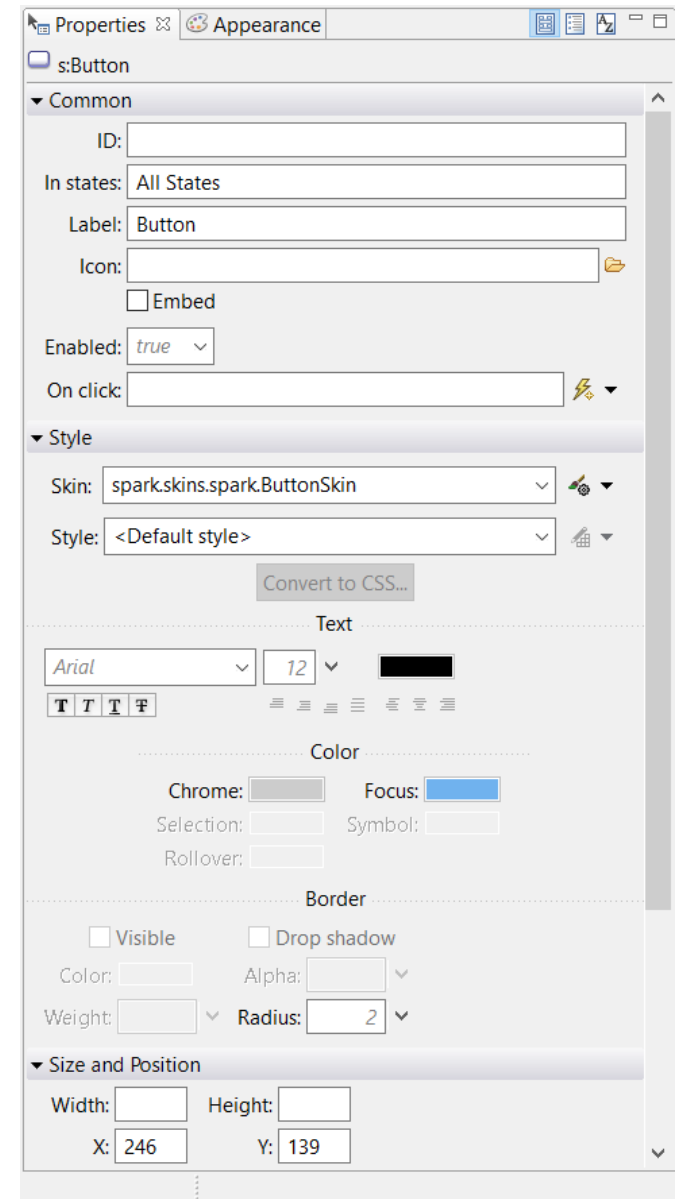
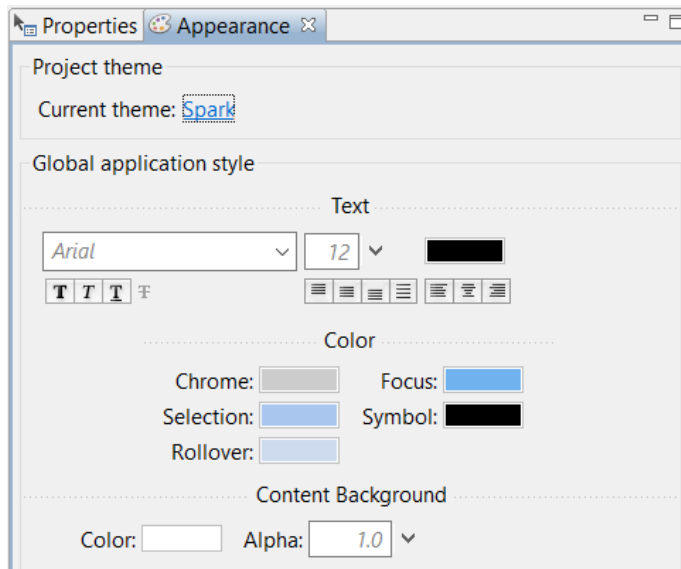
Sexo: ☒ Mujer ☐ Hombre

☐ Familia numerosa ☐ Carné Joven

Situación Laboral: 

Ejercicio

- Diseñar una GUI en Flash Builder haciendo uso de 10 elementos, y estudiar las diferentes **propiedades** y **apariencia** de cada componente.



Componentes en Flash Builder

- Application:
 - TITULO: pageTitle="MI PRIMER FLASH"
 - ANCHO: minWidth="955"
 - ALTO: minHeight="600"
- Button:
 - TEXTO: label="PULSA"
 - REFERENCIAR: id="boton1"
 - DESACTIVAR: enabled="false"
- TextInput:
 - CONTENIDO: text="Hola"
 - TEXTO PREDEFINIDO: prompt="Escriba usuario"
 - Nº MÁXIMO CARACTERES: maxChars="13"
 - PASSWORD: displayAsPassword="true"
 - DESACTIVAR: editable="false"



Componentes en Flash Builder (2)

- TextArea:
 - EDITABLE: editable="true"
 - ALINEACION TEXTO: textAlign="center"
- CheckBox:
 - SELECCIONADO: selected="true"
 - TEXTO: label: "Acepta los términos"
- ComboBox:
 - INTRODUCIR DATOS:

```
<s:ArrayList>  
    <fx:String>Agustin</fx:String>  
    <fx:String>Damian</fx:String>  
    <fx:String>Alfredo</fx:String>  
    <fx:String>Daniel</fx:String>  
</s:ArrayList>
```
 - ELEMENTO SELECCIONADO: selectedIndex="3"
- Label:
 - TEXTO: text="Label"

Componentes en Flash Builder (3)

- RadioButton:
 - TEXTO: label="Hombre"
 - SELECCIONAR: selected="true"
 - GRUPOS: groupName="r1"
- DateChooser:
 - MOSTRAR DIA ACTUAL: showToday="true"
 - MOSTRAR MES: displayedMonth="0" (enero), "11" (diciembre)
 - MOSTRAR AÑO: displayedYear="1997"
- DateField:
 - FORMATO FECHA: formatString="DD/MM/YYYY"
- Image
 - FUENTE: source="@Embed('file:///C:/Users/CFGGS/Desktop/KOALA.jpg')"
 - ESCALADO:
 - scaleMode="letterbox" → Redimensiona según imagen original
 - scaleMode="stretch" → Ajusta al contenido
 - scaleMode="zoom" → Aplica zoom a la imagen
 - SUAVIZAR: smooth="true"
 - CALIDAD SUAVIZADO: smoothingQuality="high"

Componentes en Flash Builder (4)

- List:

- INTRODUCIR DATOS:

```
<s:ArrayList>
```

```
<fx:String>Agustin</fx:String>
```

```
<fx:String>Damian</fx:String>
```

```
<fx:String>Alfredo</fx:String>
```

```
<fx:String>Daniel</fx:String>
```

```
</s:ArrayList>
```

- ELEMENTO SELECCIONADO: selectedIndex="3"

Eventos en Flash Builder

- Acción que el usuario realiza sobre un componente.

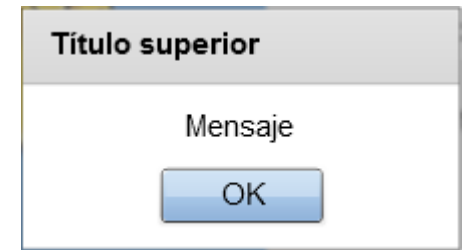
Componentes	Eventos utilizados
Button	clickHandler
TextInput	changeHandler
TextArea	changeHandler
CheckBox	clickHandler
ComboBox	changeHandler

Componentes	Eventos utilizados
RadioButton	clickHandler
DateChooser	changeHandler
DateField	changeHandler
Image	mouseOverHandler mouseOutHandler clickHandler
List	clickHandler

Cuadros de diálogo

- Mensaje simple:

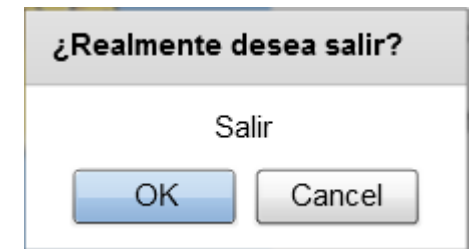
```
import mx.controls.Alert;  
Alert.show('Mensaje', 'Título superior', mx.controls.Alert.OK);
```



- Mensaje interrogativo:

```
Alert.show('Salir', '¿Realmente desea salir?', Alert.OK |  
Alert.CANCEL, this, exit_app);
```

```
public function exit_app(event:CloseEvent):void  
{  
    if (event.detail == Alert.OK)  
        //Implementar  
}
```

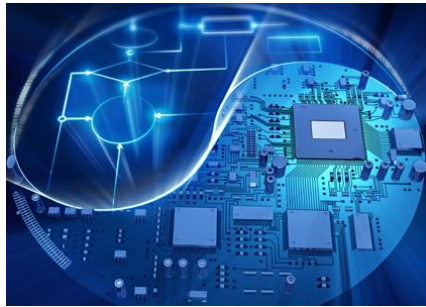


Cuadros de diálogo

- Mensaje con icono:

Ejercicio

- Tras el diseño de la GUI realizado en la diapositiva 15, añadir a cada componente posible un evento.
- Diseñar la GUI de la diapositiva 14 y añadir un evento a cada componente posible.
- Probar a incorporar en uno de los eventos un cuadro de diálogo (mensaje, opciones, etc.).



DESARROLLO DE APLICACIONES MULTIPLATAFORMA

Desarrollo de Interfaces

Equivalencia en créditos ECTS (BOE): 9

Bibliografía:

- Desarrollo de Interfaces. GFGS. Autor: FERRER MARTÍNEZ, JUAN. Editorial: RaMa
- Desarrollo de Interfaces. Autor: VICENTE CARRO, JUAN LUIS. Editorial: Paraninfo
- FLASH BUILDER: <https://helpx.adobe.com/support/flash-builder.html>
- Apuntes proporcionados por el profesor (Moodle)