**EXAMEN TEMA 1**

**INTRODUCCION**

Al principio las primeras interfaces graficas eran simples CLI (Interfaces Command Line).

**Resumen linea temporal:**

1. CLI (Interfaces Command Line)
2. WIMP (Windows, icons, menus…)
3. Simulacion 3D
4. Eventos por voz, vista, AI

**DESARROLLO DE INTERFACES DE USUARIO**

Las diferentes utilidades están contenidas en diferentes módulos.

**Biblioteca /Librería**:

* Conjunto de implementaciones o subprogramas que son utilizados para desarrollar software.
* Contienen codigo y datos de forma modular, que en la ejecutacion de un programa se accede a estos mediante un software denominado enlazador.

**Framework**:

* Los frameworks proporcionan herramientas, bibliotecas y convenciones que facilitan la creación de aplicaciones web robustas y escalables siguiendo el patrón **MVC**:
  + **Modelo**:  
    Operaciones lógicas.
  + **Vista**:  
    Muestra los datos.
  + **Controlador:**  
    Controla el acceso a nuestra app.

**INTERFAZ GRÁFICA**

Conjunto de elementos hardware y software de un ordenador que presentan información al usuario y le permite interactuar con esta.

**Principios generales en una interfaz gráfica:**

1. Sencilla
2. Intuitiva
3. Coherente
4. Clara
5. Predecible
6. Flexible
7. Consistente

**Elementos básicos de una interfaz gráfica:**

* **Widgets:**Objetos visuales del interfaz (ventanas, botones, campos de texto,…).
* **Layout:**Disposicion de los componentes del GUI.
  + Linear.
  + Relative.
  + Frame.
  + Table.
* **Eventos:**
  + Eventos de ventana:

Son los que se generan en respuesta a los cambios de una ventana, frame o dialogo.

* + - WINDOW\_DESTROY (al cerrar)
    - WINDOW\_EXPOSE (al abrir ventana)
    - WINDOW\_ICONIFY (al minimizar)
    - WINDOW\_DEICONIFY (al restaurar la ventana)
    - WINDOW\_MOVED (al mover la ventana)
  + Errores:

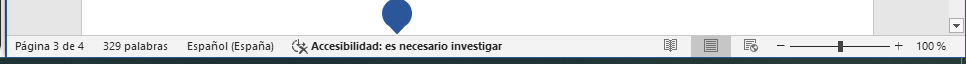
Al producirse un error se suelen detectar y corregir por medio de las excepciones.

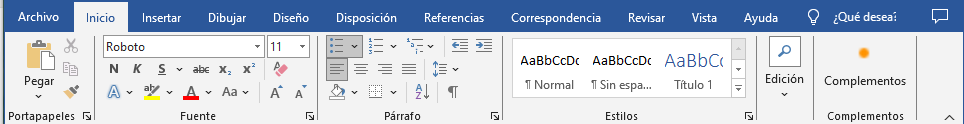
**Elementos tipicos de una interfaz gráfica:**

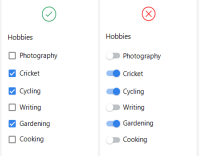
* Barras de progreso:  
  Indican el estado de avance de una tarea o proceso.
* Radio buttons (botones de opcion):  
  Para elegir una unica opcion del conjunto de opciones.

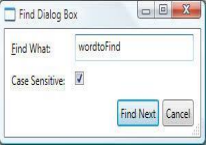
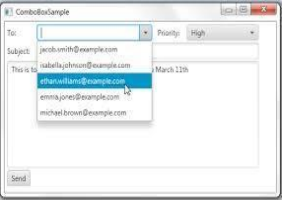


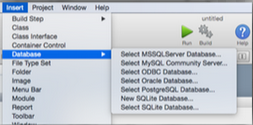
* Barras de desplazamiento:  
  Verticales y horizontales que permiten el desplazamiento en la pagina.
* Statusbar (barra de estado):  
  Muestra información acerca del estado actual de la ventana.



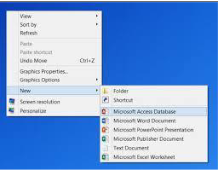
* Toolbar (barra de herramientas):  
  Agrupación de iconos.
* Titlebar (barra de titulo):
* Buttons (botones):  
  Son tipos de widget que permiten al usuario comenzar un evento.
* Checkbox (casillas verificacion):  
  Permite al usuario marcar multiples selecciones.

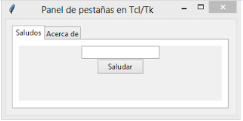
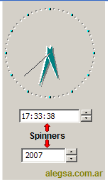
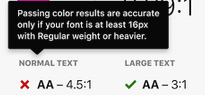
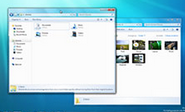
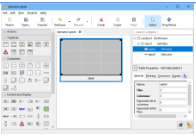


* Dialogbox (cuadro de dialogo):  
  Ventana para mostrar informacion.
* Textarea (cuadro de texto):  
  Cuadro donde se puede escribir texto.
* Iconos del sistema:  
  Usados para identificar archivos, etc…
* List (lista):  
  Relacion de datos ordenados según su tipo.
* Lista desplegable:  
  Lista de opciones que se muestra al usuario cuando escribe en un cuadro de texto.
* Menú:  
  Herramienta gráfica que nos muestra una lista de opciones.

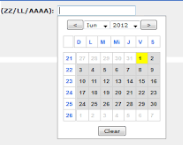


* Menú contextual:

Tipo de menú que se adapta al contexto desde donde se accede (usando click derecho).

* Panel (panel de pestañas):  
  Permite cambiar entre distintos documentos o secciones.
* Slider:  
  Elemento que permite seleccionar un valor moviendo un indicador.
* Spinner:  
  Elemento que permite al usuario ajustar un valor dentro de un cuadro de texto con dos flechas en direcciones opuestas.
* Tooltip:  
  Elemento del GUI (Interfaz Grafica) que se usa para mostrar una descripcion cuando el usuario deja encima el cursor.
* Ventana:  
  Parte delimitada de la pantalla en un SO grafico y que contiene elementos afines a ella.
* Contenedor:  
  Elemento que sirve para almacenar diferentes widgets y manejarlos de forma conjunta.
* Label (etiqueta):  
  Mensajes cortos de carácter informativo acerca de otros widgets.
* Textbox (cajas de texto):  
  Permite la introduccion de datos que luego se van a procesar.



* Calendar (calendario):  
  Permite introducir /examinar fechas.

**SVG**

Formato de gráficos vectoriales bidimensionales tanto estaticos como animados.

Una imagen vectorial es una imagen formada por objetos geometricos dependientes, definidos por atributos matemáticos (forma, posicion,…).

Hay tres tipos de objetos graficos:

1. Elementos geometricos vectoriales (curvas, rectas y areas limitadas por ellos).
2. Imágenes de mapas digitales.
3. Texto.

**Acceso a datos**

**API:**

Conjuntos de **herramientas, definiciones y protocolos** que se usa para diseñar e integrar software de aplicaciones.

Permite que su producto o servicio se comunique con otros productos y servicios, en nuestro caso con una base de datos, sin la necesidad de saber cómo se implementan.

Ejemplos de API para BBDD:

* JDBC (java database connectivity):  
  Ejecucion de operaciones sobre bases de datos desde el lenguaje Java, utilizando el dialecto SQL del modelo de la BD que se utilice.
* ODBC:  
  Estandar de acceso a bases de datos

**Usabilidad**

Es la medida de la calidad de la experiencia que tiene un usuario cuando interactua con un producto o sistema.

**Los diez principios de usabilidad heuristica:**

1. Visibilidad del estado de alarma.
2. Correspondencia entre el sistema y el mundo real.
3. Control de usuario y libertad.
4. Prevencion de errores.
5. Coherencia y estandares.
6. Reconocimiento en vez de recordar.
7. Flexibilidad y eficiencia de uso.
8. Diseño estetico y minimalista.
9. Ayuda y documentacion.
10. Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de errores.

**IPO (interaccion persona ordenador):**

Disciplina dedicada al diseño, evaluación e implementación de sistemas informáticos interactivos para el uso humano.

El prototipado dinamico es considerada la mejor tecnica de desarrollo y consiste en:

Diseñar, testar y volver a diseñar (así sucesivamente).

Los sentidos mas importantes en IPO son:

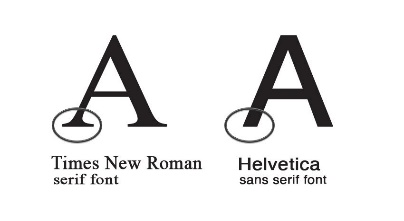
* Vista.
* Oido.
* Tacto.

**El color.**

**Las fuentes:**

La recomendación general es no utilizar más de 2 fuentes (y en algunos casos con 1 es suficiente).

* La regla del 8:  
  Consiste en que si tu fuente básica es de 10px, si necesitas otros tamaños aumentar en 18px, 26px, etc…



* Fuentes serif:
* Interlineado:  
  Recomendable utilizar alrededor de 1,5.
* Ancho de linea:  
  Entre 50 y 70 caracteres por linea es el ideal.



**Los iconos:**

Las indicaciones de un boton deben ser claras.

**El Layout /disposición:**La disposicion de los diferentes widgets es lo más importante en las interfaces de usuario.

* Menos es más.
* Los widgets deben de ser familiares (ej. icono de una impresora antigua estandar, mejor que una nueva creada por ti).

**Medidas de la usabilidad**

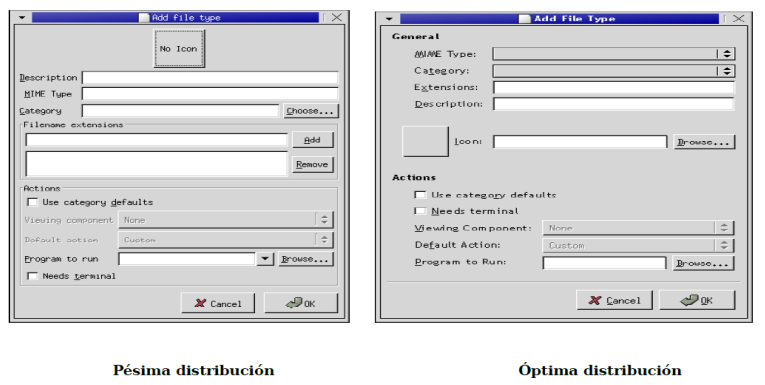
Medidas para determinar el grado y calidad de una interfaz gráfica:

* **Metricas de la eficiencia:**
  + Hacen referencia al tiempo media que es necesario invertir en cada tarea, hay diferentes parametros:
    - Tiempo invertido en primer intento.
    - Tiempo completar tarea comparado con el que necesitaria un experto.
    - Tiempo en subsanar errores cometidos.
* **Metricas de la efectividad:**
  + Miden el porcentaje de éxito (por encima de 75 es aceptable).
* **Encuestas de satisfaccion.**

Metodos para medir el grado de usabilidad:

* User testing.
* Usability inspection.

Pautas de diseño:

* Estructura de la informacion y las tareas del usuario en la app (posicion y jerarquia de los elementos visuales).
* Punto focal en la ventana (ubicación de los elementos prioritarios).
* etc…

El cuadro de Dialogo de la izquierda, presenta las **etiquetas sin alinear**. Si miramos fijamente se notará la dificultad para hacer un “scaneo” o revisión rápida de la pantalla. Algunos consejos para tener en cuenta son:

* Cuando los cuadros de texto tengan la misma longitud, se recomienda alineación izquierda.
* Si la mayoría de las etiquetas o labels difieren en longitud, se recomienda alineación derecha.
* Mantener la consistencia de los componentes de la ventana en términos de alineación y tamaño. En resumen, evitar que los ojos del usuario estén dando saltos de un sitio a otro.

**Reglas de uso de diferentes herramientas**

**Barras de menús:**

Proporciona acceso a submenus desplegables, siempre está accesible desde el teclado /ratón.

* Debe contener como minimo el item “Ayuda”.
* Nunca tener items inactivos y usar palabras compuestas.



Los menus en cascada:

* No tener submenus con menos de tres opciones.
* No puede haber mas de dos niveles de jerarquia.

Los menus contextuales es un tipo de menu desplegable que se muestra dependiendo del contexto cuando se presione el **boton derecho del raton**.

**TextBox (cuadros de texto):**

Usado para ingresar una o mas lineas de texto plano.

* Justificar a la derecha el contenido exclusivamente numerico.
* Ajustar el tamaño del textbox con el probable tamaño de los datos de entrada.
* Proporcionar un texto explicativo para aquellos que necesiten que los datos de entrada tengan X formato.

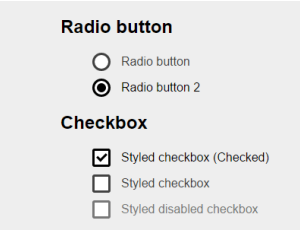
**Command Button (botones de comando):**

**Radio Button:**

* Son botones que excluyen al otro y al menos deben de existir 2.
* Max 8.
* Alineacion vertical preferiblemente.

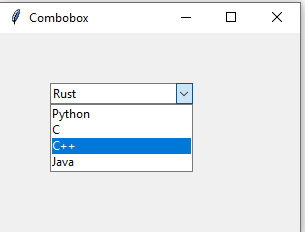
**CheckBox:**

* Multiples opciones posibles.
* Max 8.
* Alineacion vertical preferiblemente.



**Combo box:**

Lista desplegable con multiples opciones donde la selección es unica.

* Se recomienda min 8 opciones.

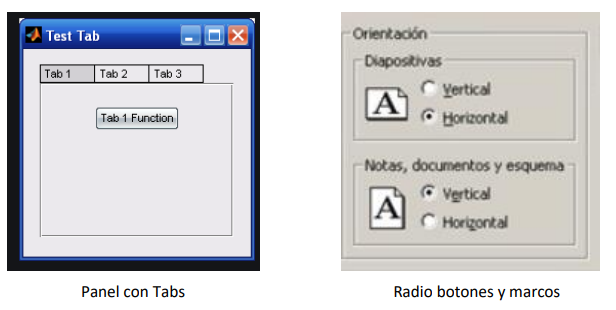
**List Box:**

Posibilidad de elegir uno o varios valores dentro de un conjunto.

**Tabs:**

**Control frames (marcos y separadores):**

Un marco es un cuadro con titulo que puede dibujarse alrededor de un grupo de objetos o controles para organizarlos como grupos funcionales.

Un separador es una linea simple.

**Informes**

Permiten al usuario acceder a la informacion de una forma estructurada.

Se suelen generarse a partir de un diseño previo o plantilla en el cual se vuelcan la informacion procedente de una base de datos.

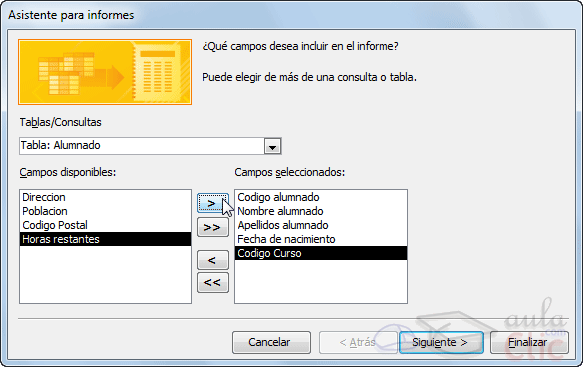
Hay dos grandes grupos según el punto de vista de diseño e inclusion en el origen de los datos:

1. El informe incrustado /integrado:  
   El proyecto crea directamente dicho informe, ej. ACCESS de Oficce (apenas se usan).
2. No incrustado /integrado:  
   Externo al proyecto, ej. Crystal Reports o ReportLabs.

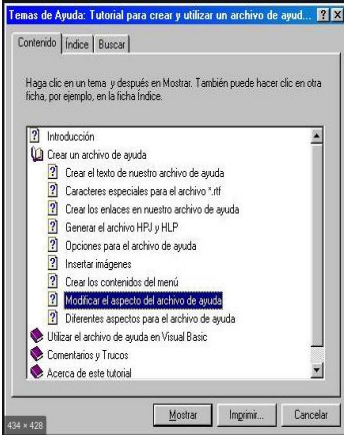
Son mas sencillos y seguros, pero requieren mayor trabajo en su diseño.

**Estructura general de un informe**

1. Report Header:  
   Cabecera del informe (logo empresa, fechas, nombre del informe…).
2. Page Header:  
   Cabecera de pagina, contiene anotaciones generales (si es una factura, num de factura…).
3. Details:  
   Campos del origen de datos.
4. Report Footer:  
   Totales generales, promedios,…
5. Page Footer:  
   Usada para imprimir la paginacion, los totales por pagina.



**Documentacion de programas**

**Winhelp (.hlp)** son archivos con extensión **hlp** de ayuda de Windows, donde el nombre de la extensión proviene de Help. Al hacer doble clic sobre un archivo con extensión hlp éste se abrirá con la **aplicación Ayuda de Windows** gracias al programa **winhlp32.exe**. Fueron los primeros.

**.chm** es el archivo de Ayuda de HTML Compilado.

**Distribucion de aplicaciones**

El empaquetado de aplicaciones de escritorio tambien conocido como application bundle, proporciona las aplicaciones en forma de paquetes.

Estos pueden estar contenidos por:

* programas ejecutables de la aplicación.
* bibliotecas necesarias de las que depende.
* otros tipos de ficheros (imagenes, bbdd, ficheros de audio, …).

Las ventajas de bundle, es que evitas el problema de las dependencias, ya que se empaqueta todo junto y contiene todos los ficheros necesarios para ejecutarse.

**Linux utiliza:**

* rpm
* deb

**Windows utiliza:**

* msi

**Java utiliza:**

* jar

**Pruebas de software**

**Caja negra:**

Se basan en la especificacion del programa a ser probado independientemente de su diseño interno (“de como lo haga”).

El corportamiento solo puede ser determinado estudiando sus entradas y salidas obtenidas a partir de ellas.

**Caja blanca:**

Este metodo se centra en como diseñar los casos de prueba atendiendo al comportamiento interno y a la estructura del programa.

El objetivo de la técnica es diseñar casos de prueba para que se ejecuten, al menos una vez, todas las sentencias del programa, y todas las condiciones tanto en su vertiente verdadera como falsa.

**Pruebas del sistema**

Se prueba el sistema para verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las funciones adecuadas.

Para la generacion de casos de prueba se utilizan tecnicas de caja negra:

**Pruebas Alfa:**

* Realizadas en entorno del desarrollador.

**Pruebas Beta:**

* Realizadas en entorno del cliente.

**Pruebas de regresion**

Cualquier tipo de pruebas que intentan descubrir las causas de los nuevos errores o bugs, carencias de funcionalidad, o divergencias funcionales con respecto al comportamiento esperado del software.

**Pruebas manuales y automatizadas**

**Datos sueltos**

**Snap**:

Es un nuevo concepto introducido con Ubuntu, nueva forma de instalar apps.

**Pip:**

Sistema de gestion de paquetes para instalar y administrar paquetes de python.