Un ***Sistema Informático*** es un ***conjunto de piezas*** que están conectadas entre sí. Este se maneja a través de periféricos. En él hay cuatro ***partes fundamentales***, en orden son: ***Usuario, Aplicaciones, Sistema Operativo y Hardware.***

Una ***Interfaz*** es un ***conjunto de programas o servicios*** que se ejecutan y relacionan los niveles para que el usuario pueda ***acceder al sistema y ejecutar programas***. Estas se encuentran en la ***capa de aplicaciones.***

Una ***Aplicación*** puede clasificarse según : Su finalidad, si es de uso genérico o a medida, si es cliente-servidor…

—------------------------------------------------------------------------------------------------------------------------

Las ***GUI / Interfaces de Usuario Gráficas*** son las que ***usan elementos gráficos*** y permiten mejor interacción ya que son intuitivas.

Las ***CLI / Interfaces de Líneas de Instrucciones*** son las que ***funcionan mediante comandos*** que ejecutan instrucciones. Estas ***se siguen usando en routers y conmutadores***, aunque también es posible usar una tanto en Linux como en Windows.

***Las interfaces han evolucionado a la vez que los sistemas operativos. Windows 3.1*** fué el primero en usar una GUI con tantos elementos y ventanas. Esto hizo que fuera más sencillo y accesible para más gente. Su ***evolución*** se clasifica en ***cinco fases :***

***CLI / Interfaces de Línea de Instrucciones :*** Se empezaron a usar en 1960, cuando aparecieron los lenguajes de órdenes.

***Primeras Interfaces de Usuario :*** Se crearon a partir de 1970, estas se usaron en los primeros ordenadores personales. Estos estaban físicamente conectados a un ordenador central que daba el servicio a los usuarios mediante instrucciones.

***Evolución de las Primeras Interfaces de Usuario :*** Se popularizaron los ordenadores personales, estos usaban las interfaces WIMP (Windows, Icons, Menus and Pointer).

Estas interfaces ya se consideraban de tercera dimensión, consiguiendo superponer ventanas.

***Interfaces Gráficas de Usuario No Basadas en Instrucciones :*** En 1995 la gente ya tenía ordenadores en casa. Estas interfaces seguían basándose en las WIMP, simplemente fueron añadiendo nuevos elementos y funcionalidades. Los elementos multimedia, por ejemplo, llegaron a etiquetarse como la cuarta dimensión.

***Interfaces Gráficas de Usuario Interactivas :*** A día de hoy todo el mundo interactúa con una GUI por lo extendido y fácil que es aprender a usarlas.

***Las Interfaces Actuales se pueden clasificar de cuatro formas :***

***Forma de interactuar con los usuarios :***

***CLI / Interfaces de Línea de Instrucciones. GUI / Interfaces Gráficas de Usuarios.***

***Interfaces de Pantallas Táctiles :*** Se diferencian en que están adaptadas a las necesidades de los dispositivos de entrada y salida táctiles.

***Interfaces de Dispositivos Móviles :*** Sus sistemas operativos e interfaces son muy específicas.

***En Función de la interfaz :***

***Hardware :*** Estas interfaces permiten la interacción entre el usuario y el sistema físicamente, como un lector manual.

***Software :*** La interacción del usuario es mediante un software / sistema operativo.

***Software-Hardware :*** Usan ambas tecnologías, como un lector de supermercado que además muestra la información en pantalla.

***En Función de la Arquitectura de las Aplicaciones y su funcionamiento :***

***Aplicaciones Locales :*** Usan interfaces Winforms, diseñadas para un solo usuario que están hechas con distintos lenguajes de programación y vinculadas a bases de datos instaladas en el dispositivo.

***Aplicaciones Cliente-Servidor :*** Se basan en el funcionamiento del servidor. Los clientes mandan peticiones al servidor y este les envía una respuesta con la información o servicio. También son Winforms y se instalan en el cliente y en el servidor.

***Aplicaciones Web :*** Son un tipo específico de aplicaciones cliente-servidor. Se caracterizan por usar servidores web a través de un navegador.

***En función de la tecnología con la que se desarrolla :***

***Plataformas RichClient :*** IDE’s con elementos para diseñar y desarrollar una interfaz gráfica haciendo uso de bibliotecas.

***Plataformas ThinClient :*** IDE’s que necesitan otras herramientas para el diseño y desarrollo de interfaces gráficas y software.

La ***Ley de Fitts*** hace una estimación del ***tiempo que necesita un usuario para mover el puntero*** de una zona a otra teniendo en cuenta varios factores.

No todas las interfaces contienen los mismos elementos y estos no siempre tienen las mismas funcionalidades. ***Los elementos más comunes son :***

***Ventanas :*** Son el elemento más importante, ofrecen información organizada y sencilla.

***Sus partes son :***

***Marco :*** Divide la venta del resto y permite redimensionarla.

***Cabecera :*** En ella se encuentran los iconos y principales elementos.

***Espacio del contenido :*** Varía en función de la dimensión de la ventana.

***Scroll / Barra de desplazamiento :*** Permite ver el contenido que está abajo.

***Pie :*** Ofrece algún elemento extra de la aplicación.

***Cuadros de diálogo / Ventanas secundarias :*** Aparecen encima de otra ventana pidiendo una interacción al usuario. Puede ser simplemente informativa, como un error al que debemos darle a ok o cerrar, aunque puede ser uno que aceptes o no guardar cambios.

***Asistentes :*** Son otro tipo de ventanas usadas en los instaladores de dispositivos y software.

***Menús :*** Muestran todas las maneras de interactuar con la aplicación, estos suelen estar bajo la cabecera. Se clasifican en tres grupos :

***Menús jerárquicos :*** Se estructuran en forma de árbol, tiene niveles horizontales y verticales.

***Menús contextuales :*** No son visibles en la interfaz, se abren en una ventana flotante que varía su aparición dependiendo de donde se haya clicado con el ratón.

***Scroll / Menús de navegación :*** Son la evolución del jerárquico, simplificando el número de elementos hasta que se activen, con la finalidad de que sea más óptimo.

***Pestañas :*** Muestran un conjunto de datos y funcionalidades de forma agrupada.

***Barra de herramientas :*** Facilitan el acceso a funciones mediante iconos, aunque no aparecen todas como en el Menú

***Iconos :*** Son imágenes que contienen una funcionalidad, archivo, carpeta, etc…

***Entornos de trabajo :*** Es la ubicación principal de información que es dada al usuario.

***Controles :*** También proporcionan funcionalidades al usuario mediante botones

***Hay tres tipos :***

***Botones de radio :*** Se usan en formularios o menús, sólo se puede marcar una opción.

***Botones de confirmación :*** Estos tienen la posibilidad de seleccionar más de uno.

***Botones de relieve :*** Imita a un botón físico con un texto. Puede estar activado o no.

***Tipografía :*** Es la forma en que se muestra el texto que se usa en las interfaces. Normalmente cada sistema operativo tiene una propia.

—------------------------------------------------------------------------------------------------------------------------

*Las interfaces dependen del lenguaje de programación y entorno de desarrollo* que se usen. Ya existen ***herramientas de creación y edición para las GUI***, estos software lo agilizan mucho. Estos tienen ***tres elementos destacables :***

***Elementos programados :*** Facilitan la creación de interfaces más complejas, por lo que es más sencillo de usar.

***Gestión de errores :*** Permiten procesar excepciones, gestionar los accesos y dispositivos.

***Herramientas WYSIWYG :*** “Lo que ves es lo que tienes” muestra el resultado a tiempo real.

La mayoría de las herramientas usan ***la programación visual***, que permite colocar los elementos sin tener que escribir líneas de código. Estas ***tienen cinco secciones :***

Panel de proyectos, Área de trabajo / diseño, Propiedades, Menús y Barra de herramientas.

***Los elementos de estas herramientas son :***

Los ***Componentes principales*** son : Label, Button, PictureBox, ProgressBar, RadioButton, CheckBox, ListBox, TextBox y ComboBox.

Un ***Control*** ***es un componente***, que permite interactuar e intercambiar información.

Un ***Panel*** ***es un control*** que contiene otros controles. (Barra de herramientas)

Un ***Formulario es un control*** parecido a la ventana de la herramienta a la que se pueden añadir más componentes. (Ventanas, cuadros de diálogo…)

La **Ventana de cuadro de herramientas** contiene componentes que se usan en el área de diseño.

Hay ***Contenedores*** que heredan de la clase ContainerControl que funcionan como contenedores de componentes.

Las ***Bibliotecas*** se usan para guardar elementos creados que se vayan a necesitar varias veces, podremos hacerlas nosotros o usar una ya creada.

Un ***Widget*** es un componente visual programado que puede reutilizarse para desarrollar otras aplicaciones.

—------------------------------------------------------------------------------------------------------------------------

***Características de los Controles:***

***Propiedades :*** Características que identifican a cada componente, estos usan la encapsulación, permite que cada uno sea independiente del resto y no afecten sus cambios.

***Métodos :*** Funciones asociadas a un componente que realizan operaciones.

***Eventos :*** Se producen cuando se interactúa con algún componente del programa.

—------------------------------------------------------------------------------------------------------------------------

Un ***Cuadro de diálogo*** es una ***ventana con información, hay tres tipos :***

***Cuadro de diálogo modal :*** Cuando la aplicación no puede recibir eventos hasta que se cierre el mismo, confirmando algo.

***Cuadros de mensaje :*** Es un cuadro de diálogo modal en el que no necesariamente hay que seleccionar una opción, se puede cerrar.

***Cuadro de diálogo no modal :*** Cuando no limita a seguir interactuando con el resto de ventanas y se puede cerrar.

***Cuadros de diálogo comunes :*** Pueden ser modales o no modales, como los del móvil.

El ***Enlace de datos*** es un mecanismo que ***permite enlazar datos con los controles del formulario.*** Existen dos tipos de ***Data Binding:***

***Simple data binding / Acceso a datos simples :*** Asocia un control a un único dato.

***Complex data binding / Acceso a datos complejos :*** Funciona como visor de datos, ya que puede mostrar varios o todos los datos.

***\****Estos usan la clase ***Binding\****