

1. DISEÑO DE INTERFACES

1.1. Interfaz de usuario (UI).

1.2. Experiencia de usuario (UX).

interfaz gráfica alguna vez en nuestras vidas nos hemos abrumado al usar algo o en otros casos tan solo lo hemos dejado de lado al no entenderlo siendo esta una debilidad de muchas de las aplicaciones que llevan grandes estudios de mercado o de diseño de interfaz ya que descuidan al usuario y terminan desarrollando software es con estéticas impresionantes pero cuando los usuarios comienzan a usarlas se sienten ineptos y no vuelven a usar el producto y el desarrollo de una interfaz eficaz y agradable pensada en el usuario para que las personas se sientan cómodas al momento de buscar información haciendo que las aplicaciones desarrolladas cumplan su función principal tratando de resolver está y muchas más problemáticas asociadas a la experiencia del usuario con la aplicación en estos tiempos el desarrollo de productos no tiene que ser sólo funcional sino que tiene que conectarse con el usuario haciendo que este prefiera el producto lo adopte y se sienta cómodo utilizandole



Introducción

En estos tiempos el desarrollo de productos no tiene que ser solo funcional, sino que tiene que conectarse con el usuario



antecedentes la UX o experiencia de usuario aunque es un término y una disciplina relativamente nuevos para la mayoría de la gente la experiencia de usuario es una disciplina con más de un siglo de antigüedad muchos de los métodos que hoy conocemos y usamos en la experiencia de usuario tienen sus raíces en los primeros estudios de ergonomía y factores humanos que comenzaron cerca del comienzo del siglo 20 y tuvieron una fuerte influencia hasta la segunda guerra mundial de la misma manera los distintos campos de la psicología experimental especialmente aquellos anclados en las teorías del aprendizaje y las ramas cognitivas y conductuales de la psicología jugaron un papel preponderante sentando las bases de lo que hoy aceptamos comúnmente como métodos de investigación con usuarios



UX

La Experiencia de Usuario es una disciplina con más de un siglo de antigüedad



a mediados de la segunda revolución industrial cada vez se incorporaban más máquinas y herramientas para hacer tareas de producción industrial en cuanto mayor era el uso de estas máquinas más rápido se trabajaba y más aumentaba la producción por este motivo era importante optimizar la factibilidad de uso y la eficiencia de las máquinas para así mejorar la producción de bienes y servicios con esto acababa de nacer una parte del actual y a partir de 1990 la experiencia del usuario crece de forma exponencial y empezamos a oír conceptos como el de usabilidad diseño de interacción y arquitectura de información todo esto nos trae hasta la actualidad en la que la experiencia de usuario es una característica fundamental para lograr que un producto o un servicio tenga éxito ya que de ello depende en gran medida la percepción positiva o negativa de las organizaciones



Antecedentes

La A partir de 1990 la experiencia del usuario crece de forma exponencial y empezamos a oír conceptos como el de usabilidad, diseño de interacción y arquitectura de información.



hoy en día nos referimos a nosotros mismos o vemos en ofertas de empleo cosas como x visual designer o x international designers o x servicios design air cuando simplemente se refiere a diseñar una web o ave sin tener en cuenta el resto de la experiencia la experiencia de usuario es todo la manera en la que se experimenta el mundo la vida un servicio en sí una web o haga pero no de forma aislada sino como un conjunto por lo que se recomienda utilizar las ocho reglas doradas del diseño de interfaces las cuales son la base de la evaluación heurística de sistemas computacionales modernos las mismas se resumen en uno esfuérzate por la consistencia 2 permite que los usuarios frecuentes utilicen accesos directos o atajos 3 ofrece respuestas a las acciones del usuario 4 diseña textos de diálogo para cerrar procesos 5 ofrece un sencillo manejo de errores 6 permite la facilidad de regresar sobre sus propios pasos 7 soporta el locus interno de control 8 reduce la carga de memoria a corto plazo



Antecedentes

- ☐ Esfuérzate por la consistencia.
- ☐ Permite que los usuarios frecuentes utilicen accesos directos o atajos.
- ☐ Ofrece respuestas a las acciones del usuario
- ☐ Diseña textos de diálogo para cerrar procesos
- ☐ Ofrece un sencillo manejo de errores.
- ☐ Permite la facilidad de regresar sobre sus propios pasos
- ☐ Soporta el locus interno de control.
- ☐ Reduce la carga de memoria a corto plazo.



para el desarrollo de aplicaciones móviles es importante definir y responder las siguientes interrogantes uno quién es su audiencia saber a qué audiencia va a enfocarse puede ayudarle a saber cómo son las personas que utilizan la aplicación dónde están con qué frecuencia y durante cuánto tiempo utilizan una aplicación y cuáles son los dispositivos más populares entre sus usuarios los datos pueden ayudarle a descubrir nuevos mercados y a mejorar el rendimiento de la aplicación 2 cuál es el propósito de su aplicación el propósito de una app es facilitarnos en cualquier momento la resolución de una tarea determinada o ayudarnos en operaciones y gestiones del día a día 3 cuál es el problema que se trata de resolver la definición del problema que se trata de resolver consiste en realizar un estudio a la problemática en cuestión independientemente del contexto en que se encuentre qué tipo de contenido se podrá incorporar el contenido a incorporar será todo aquel que se adapte a las necesidades del usuario este contenido se encuentra intrínsecamente relacionado con el problema que se trata de resolver



Desarrollo de Apps



Audiencia.
Propósito.
Problema a resolver.
Contenido a incorporar.



definiendo la interfaz de usuario simplicidad un usuario tiene que interactuar con una interfaz de aplicación en la forma más sencilla posible para diseñar la interfaz hay que tener al usuario en mente y desarrollarla de forma eficiente clara y sencilla cada detalle es importante y esto marca la diferencia respecto a otras apps la aplicación móvil debe ser fácil de usar siempre es de ayuda si se reducen los procesos y se logra realizar una determinada acción con pocas pulsaciones o gestos las líneas guías para un buen diseño de interfaz de usuario son uno centrarse en las tareas principales dos realizar uso fácil y obvio 3 utilizar la interfaz de usuario de forma coherente 4 utilizar animación sutil para comunicarse



Simplicidad

- Centrarse en las tareas principales.
- Realizar uso fácil y obvio.
- Utilizar la UI de forma coherente.
- Utilizar animación sutil para comunicarse.



consistencia como norma general las aplicaciones móviles deben mantener un mismo diseño en toda la aplicación misma gama de colores misma tipografía mismo tamaño de la fuente y misma estructura que el diseño sea homogéneo en todas las pantallas de la app ayuda a que el usuario reconozca el producto y tenga una buena experiencia de usuario esta consistencia también la tenemos que ver entre lo que vemos en la página de la aplicación del market app store y google play y la misma app el diseño del icono y las screenshot tienen que ser consistentes con el diseño interno de nuestra aplicación móvil



Consistencia

- Diseño sea homogéneo .
- Consistentes con el diseño interno.
- Un mismo diseño en toda la aplicación.



navegación intuitiva las interacciones que vaya a realizar el usuario deben ser muy sencillas hay que tener en cuenta las zonas donde se produce el mayor número de interacciones en los diferentes dispositivos móviles por ejemplo un smartphone se utilizan normalmente con una sola mano es por ello que los botones que más se utilizan deberían estar en la parte inferior de la pantalla para poder acceder mejor a ellos tampoco podemos olvidarnos de los gestos el uso de gestos asimilados por los usuarios hay que aprovecharlos tocar arrastrar deslizar mantener pulsado no es recomendable crear gestos exclusivos para la app creyendo que esto le dará un toque diferente posiblemente esto provocará que el usuario se pierda piense que no funcione y termine por no utilizar la aplicación un aspecto importante a considerar es que es complicado predecir cómo el usuario va a utilizar la aplicación móvil por lo que mantener el diseño simple e intuitivo ayuda en gran medida



Navegación Intuitiva

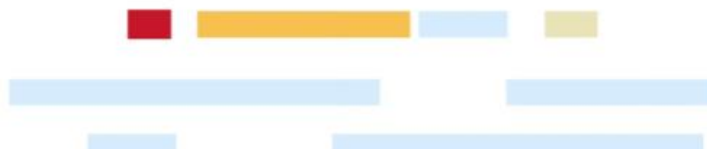
Un smartphone se utiliza normalmente con una sola mano.



definiendo la interacción las aplicaciones móviles se basan en la programación orientada a eventos es decir el flujo de la aplicación se determina por eventos del sistema y acciones del usuario el usuario realiza acciones en la interfaz que desencadenan acontecimientos que dan lugar a la ejecución lógica y manipulación de la aplicación



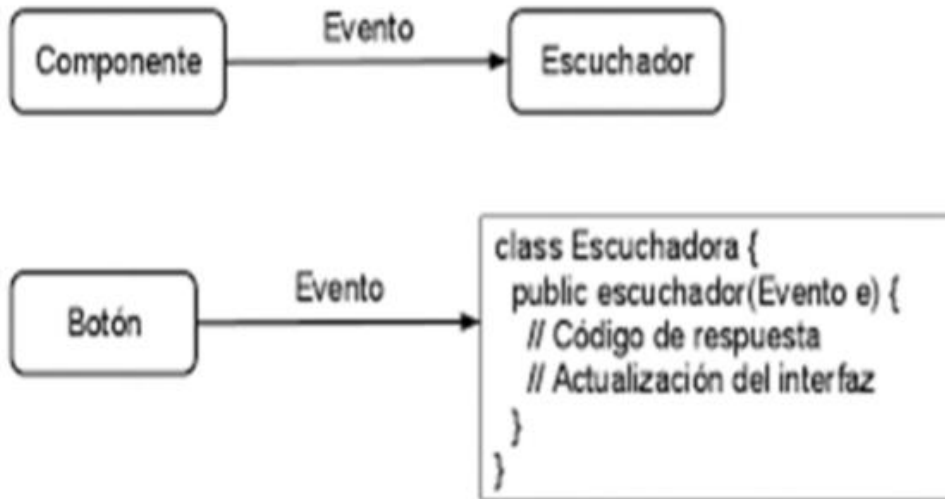
DEFINIENDO LA INTERACCIÓN



la respuesta de la aplicación para la acción del usuario se refleja luego devuelta en la interfaz debido a que el usuario en lugar del desarrollador tiene el control de ciertas partes del código de la aplicación cuando se ejecutan se debe identificar qué acciones puede realizar un usuario y lo que ocurre en respuesta a esas acciones



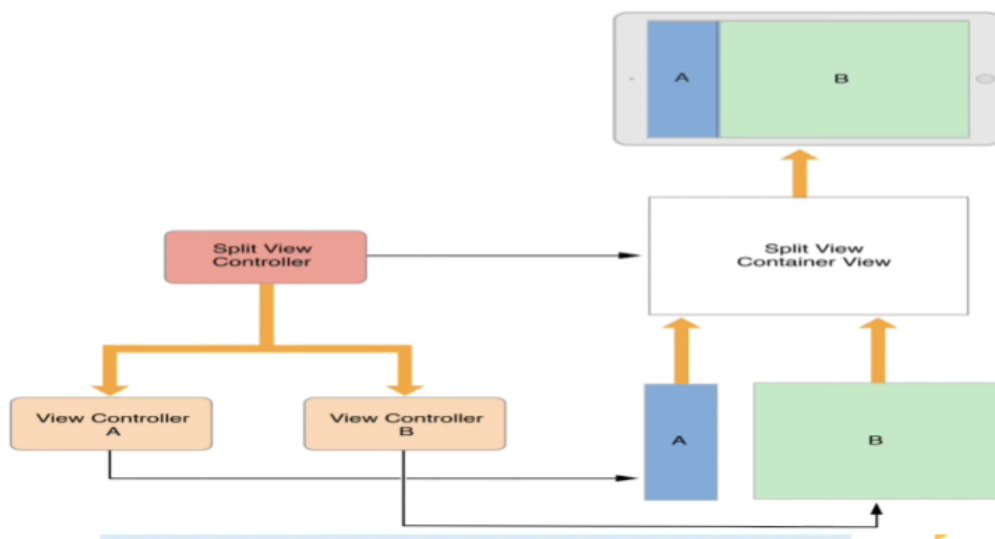
Respuesta de Aplicación



el programador define gran parte de la lógica de control de eventos en los controladores de vistas view controller los controladores son objetos que apoyan a sus vistas al responder las acciones del usuario y mostrar en sus respectivos elementos el contenido luego de construir una jerarquía de vistas se debe tener el control de los elementos visuales y responder a las entradas del usuario.



Controladores de vistas



1. DISEÑO DE INTERFACES

1.3. Gamificación

Ver pdf Gamification (Ludificación)

1. DISEÑO DE INTERFACES APLICACIONES

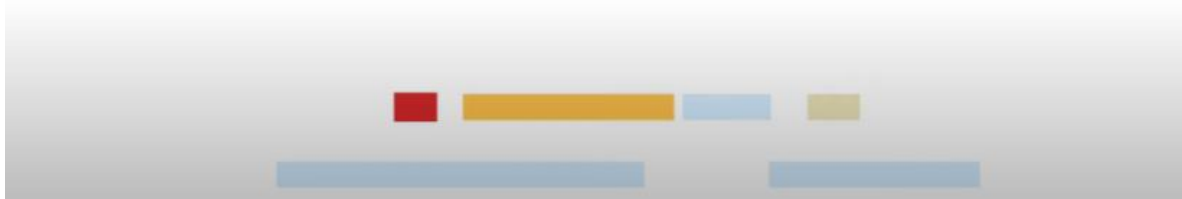
1.4 Prototipos de interfaz de usuario

1.4.1. Implementación de una App.

La implementación de una app patrones de diseño un patrón de diseño resuelve problemas comunes de ingeniería de software los patrones son diseños abstractos no código se utilizan para ayudar a definir la estructura de modelos de datos y su interacción con el resto de la aplicación cuando se adopta un patrón de diseño es para adaptar un modelo general a las necesidades específicas no importa qué tipo de aplicación se está creando es bueno conocer los patrones de diseño fundamentales utilizados en los frameworks la comprensión de los patrones de diseño le ayuda a utilizar los frameworks de manera más eficaz y permite escribir aplicaciones que son más reutilizables más extensibles y más fáciles de cambiar



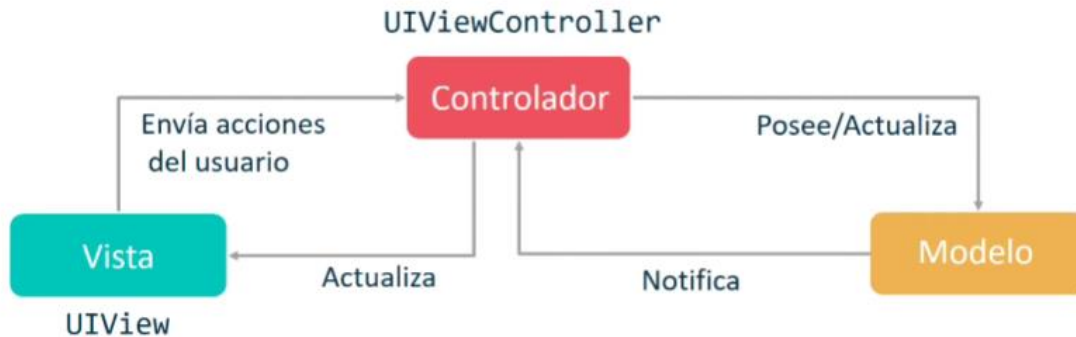
PATRONES DE DISEÑO



El patrón MVC es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de su presentación y el módulo encargado de gestionar los eventos y las comunicaciones. El modelo vista controlador MVC es fundamental para un buen diseño para cualquier aplicación de Android y iOS. MVC asigna los objetos en una aplicación para uno de los tres roles: modelo, vista o controlador. Los modelos son los componentes de datos de la aplicación en ellos reside una presentación de cada elemento que constituye una parte del problema que se pretende resolver. Las vistas muestran la interfaz de usuario y constituyen el contenido de una aplicación.



Patrón MVC

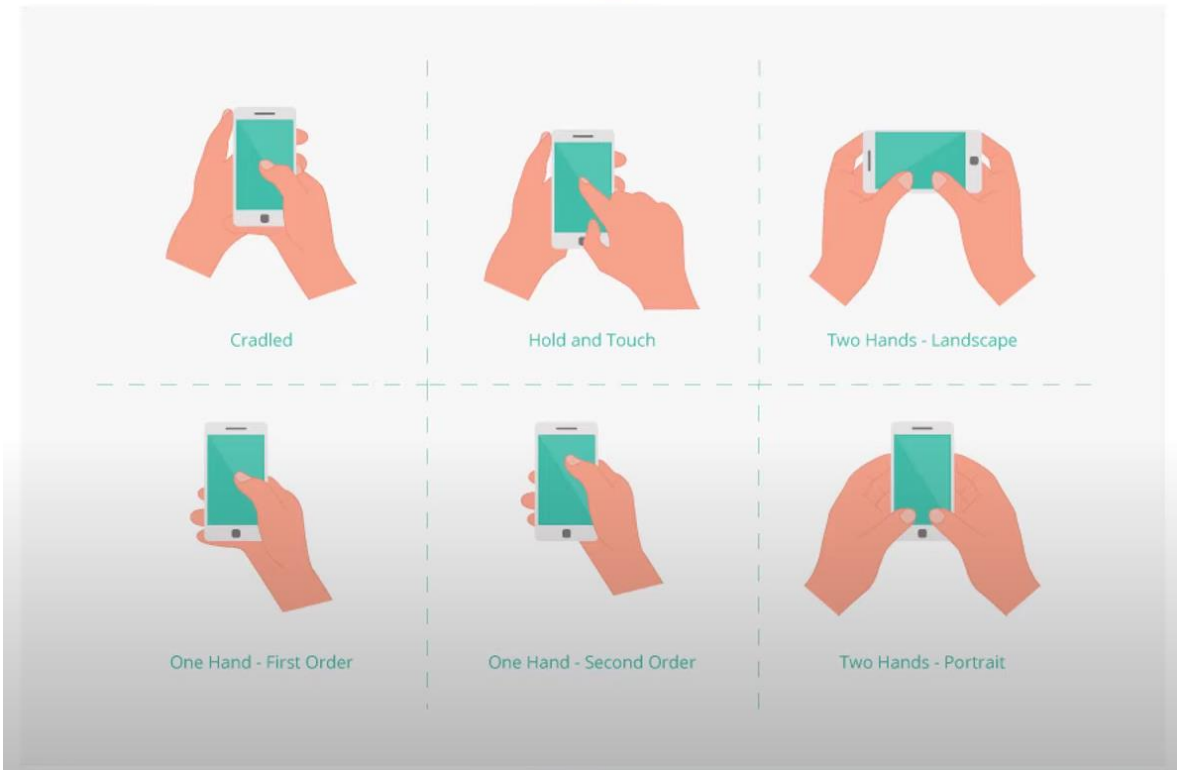


los controladores gestionan las vistas y envían información a ellas para ser posteriormente mostradas al usuario de manera genérica los componentes de mvc se podrían definir de la siguiente manera el modelo es la representación de la información con la cual el sistema opera por lo tanto gestiona todos los accesos a dicha información tantas consultas como actualizaciones implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación lógica de negocio envía a la vista aquella parte de la información que en cada momento se le solicita para que sea mostrada las peticiones de acceso o manipulación de información llegan al modelo a través del controlador el controlador responde a eventos usualmente acciones del usuario e invoca peticiones al modelo cuando se hace alguna solicitud sobre la información por ejemplo editar un documento o un registro en una base de datos también puede enviar comandos a su vista asociada si se solicita un cambio en la forma en que se presenta el modelo por ejemplo desplazamiento o scroll por una screen o por los diferentes registros de una base de datos por tanto se podría decir que el controlador hace de intermediario entre la vista y el modelo la vista presenta el modelo información y lógica de negocio en un formato adecuado para interactuar usualmente la interfaz de usuario por tanto requiere de dicho modelo la información que debe representar como salida



patrón target action target action es un diseño conceptualmente simple en el que un objeto envía un mensaje a otro objeto cuando se produce un evento específico el mensaje action es un selector que se define en el código fuente y el target objetivo en el objeto que recibe el mensaje es un objeto capaz de llevar a cabo la acción un view controla el objeto que envía el mensaje de acción es por lo general un control como un botón control deslizante o cambiar que puede desencadenar un evento en respuesta a la interacción del usuario como tocar arrastrar o cambiar de valor patrón

Patrón Target Action



delegation delegation es un patrón simple y de gran alcance en el que un objeto en una aplicación actúa en nombre de o en coordinación con otro objeto el objeto delegar mantiene una referencia a otro objeto delegado y en el momento apropiado el objeto delegar envía un mensaje al delegado el mensaje informa al delegado de un evento que el objeto delegar está a punto de manejar o simplemente ha manejado el delegado puede responder al mensaje mediante la actualización de la partición o estado de sí mismo o de otros objetos en la aplicación y en algunos casos se devolverá un valor que afecta a cómo se maneja un evento inminente



Patrón Delegation



desarrollo de apps para android

DESARROLLO DE APPS PARA ANDROID

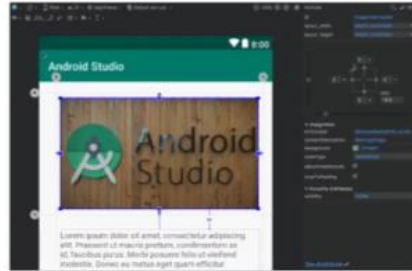
android estudio el desarrollo de aplicaciones android se realiza con un grupo de herramientas que son suministradas en el sdk software development kit el sdk de android incluye un conjunto de herramientas de desarrollo comprende un depurador de código biblioteca un simulador de teléfono basado en un e-mail documentación y ejemplos de código y tutoriales las plataformas de desarrollo soportadas incluyen en eeuu linux mac os x 10.5.8 o posterior y windows xp o posterior la plataforma integral de desarrollo y de integrated development environment soportada oficialmente es android estudio junto con el complemento a dt android development tools plugin además los programadores pueden usar un editor de texto para escribir ficheros java y xml y utilizar comandos en una terminal se necesitan los paquetes jdk java del velamen kit y apache ans para crear y depurar aplicaciones así como controlar dispositivos android que estén conectados es decir reiniciar los instalar aplicaciones en remoto etcétera



Android Studio

El SDK de Android, incluye un conjunto de herramientas de desarrollo

Las plataformas de desarrollo soportadas incluyen GNU/Linux, Mac OS X 10.5.8 o posterior, y Windows XP o posterior.

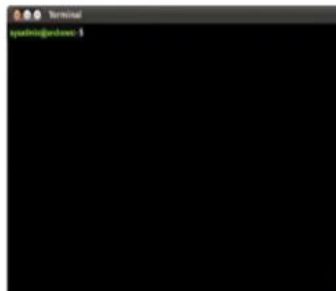


una aplicación android está compuesta por un conjunto de ficheros empaquetados en formato punto apk y guardada en el directorio data fleca app del sistema operativo android este directorio necesita permisos de superusuario root por razones de seguridad un paquete apk incluye ficheros de x 5 ejecutables dalvik un código intermedio compilado recursos etcétera la utilización de este grupo de herramientas puede ser de dos formas utilizando un entorno de desarrollo integrado ide en combinación con un plugin llamado adt android development tools herramientas de desarrollo para android y desde la línea de comandos



Android Studio

- Utilizando un Entorno de Desarrollo Integrado (IDE) en combinación con un plugin llamado ADT (Android Development Tools, Herramientas de Desarrollo para Android)
- Desde la línea de comandos




principales características que incluye android estudio soporte para programar aplicaciones para android sistema operativo para dispositivos corporales como por ejemplo un reloj herramientas lint que detecta código no compatibles entre arquitecturas diferentes o código confuso que no es capaz de controlar el compilador y para detectar problemas de rendimiento usabilidad y compatibilidad de versiones broward para optimizar y reducir el código del proyecto al exportar a apk muy útil para dispositivos de gama baja con limitaciones de memoria interna integración de la herramienta grado encargada de gestionar y automatizar la construcción de proyectos como pueden ser las tareas de testing compilación o empaquetado posibilita el control de versiones accediendo a un repositorio desde el que poder descargar mercurial kit githu o subversión vista previa en diferentes dispositivos y resoluciones integración con google cloud platform para el acceso a los diferentes servicios que proporciona google en la nube



Características Android Studio

- ☐ Soporte.
- ☐ Herramientas Lint.
- ☐ ProGuard.
- ☐ Integración de la herramientas Gradle.
- ☐ Posibilita el control de versiones.
- ☐ Integración con Google Cloud Platform.



pasos básicos para desarrollar aplicaciones android una instalación en esta etapa se instala el entorno de desarrollo completo incluyendo el eddie y el sdk de android y se crean ave de android virtual de baix dispositivos virtuales android 2 la conceptualización el resultado de esta etapa es una idea de aplicación que tiene en cuenta las necesidades y problemas de los usuarios la idea responde a una investigación preliminar y a la posterior comprobación de la viabilidad del concepto 3 definición en este paso del proceso se describe con detalle a los usuarios para quienes se diseñará la aplicación usando metodologías como personas y viajes del usuario también aquí se sientan las bases de la funcionalidad lo cual determinará el alcance del proyecto y la complejidad de diseño y programación de la app 4 diseño en la etapa de diseño se llevan a un plano tangible los conceptos y definiciones anteriores primero en forma de wireframe que permiten crear los primeros prototipos para ser probados con usuarios y posteriormente en un diseño visual acabado que será provisto al desarrollador en forma de archivos separados y pantallas modelo para la programación del código

Pasos para Desarrollar Aplicaciones Android

- ☐ Instalación.
- ☐ Conceptualización.
- ☐ Definición.
- ☐ Diseño.



5 desarrollo en esta etapa se crea y desarrolla el proyecto android creando el código fuente de la aplicación y añadiendo todos los archivos fuentes que puedan necesitar como imágenes o demás recursos 6 depuración y pruebas en esta etapa en primer lugar se genera un paquete de depuración punto apk que contiene el proyecto desarrollado en la etapa de desarrollo este paquete se puede instalar y arrancar en cualquier emulador o teléfono android a continuación se depura la aplicación usando un depurador jw vp y las herramientas de book del sdk android 7 publicación en esta última etapa se configura y se construye la aplicación para generarse una versión real y una versión de entrega para distribuir entre los usuarios cambia tus pensamientos y cambiarás tu mundo hasta la próxima.

Pasos para Desarrollar Aplicaciones Android

- ☐ Desarrollo.
- ☐ Depuración y pruebas.
- ☐ Publicación.




1. DISEÑO DE INTERFACES

APLICACIONES

1.4 Prototipos de interfaz de usuario

1.4.2. Conceptos básicos.

y componentes básicos activity representa una ventana pantalla de la aplicación y podemos decir que normalmente una aplicación móvil está compuesta por varias actividades a cada actividad se le asigna una ventana en la cual dibujar su interfaz gráfica una actividad debe ser una subclase directa o indirectamente de la clase activity si queremos iniciar otra activity declaramos un intento cuyo contexto es la actividad actual y especificamos la clase de la actividad que deseamos iniciar como lo podemos ver en la pantalla




Activity

```
import android.app.Activity;

public class MainActivity extends Activity{

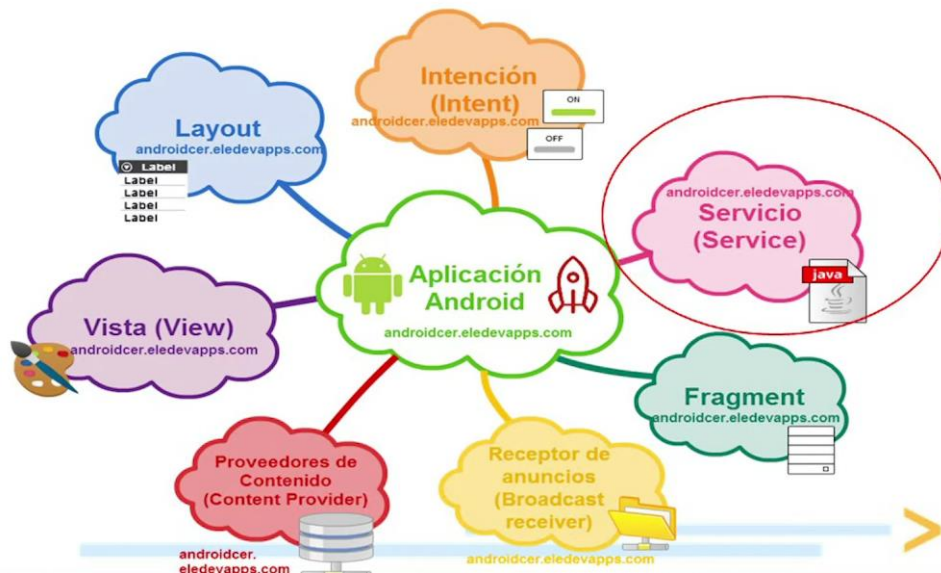
}

Intent intent = new Intent(this, SecondActivity.class);
startActivity(intent);
```



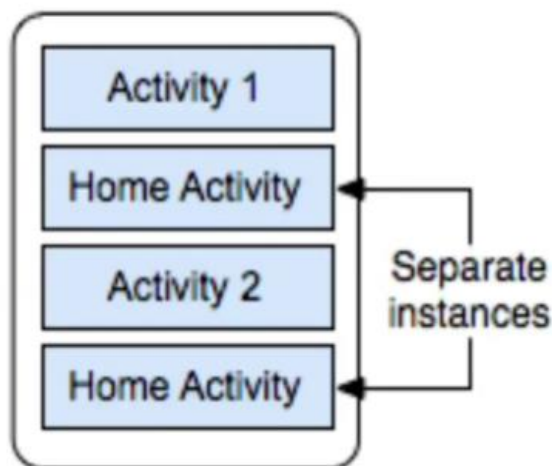
services los servicios services son componentes sin interfaz gráfica que se ejecutan en segundo plano pueden realizar cualquier tipo de acciones por ejemplo actualizar datos lanzar notificaciones o incluso mostrar elementos visuales si se necesita en algún momento la interacción con el usuario broadcast recibe un broadcast louise ivers es un componente destinado a detectar y reaccionar ante determinados mensajes o eventos globales generados por el sistema escuchan eventos producidos desde el sistema o por otras aplicaciones por ejemplo batería baja sms recibido tarjeta sd insertada etcétera contemplo raiders es el mecanismo que implementa android para compartir datos entre aplicaciones no existe área común de datos ya que cada aplicación almacena los datos por separado

Service



las tareas o estás son grupos de actividades que en conjunto realizan un trabajo determinado toda una tarea puede pasar el background cuando el usuario presiona la tecla de home utiliza el task manager o abre una notificación una actividad puede ser instancia da muchas veces a través de tareas distintas

Task



intent son mensajes asíncronos que permiten la comunicación entre aplicaciones y notificaciones al sistema operativo una interna es un objeto de mensajería que puede usar para solicitar una acción de otro componente de una app las actividades se inician por medio de intensa que describe la

actividad que queremos iniciar existen dos formas básicas de iniciar una actividad explícitamente actividades dentro de nuestra aplicación implícitamente describimos al sistema operativo la acción que queremos realizar y android lanza la actividad que pueda procesar esa acción si existe más de una actividad que cumpla los requisitos el usuario escoge cuál utilizar

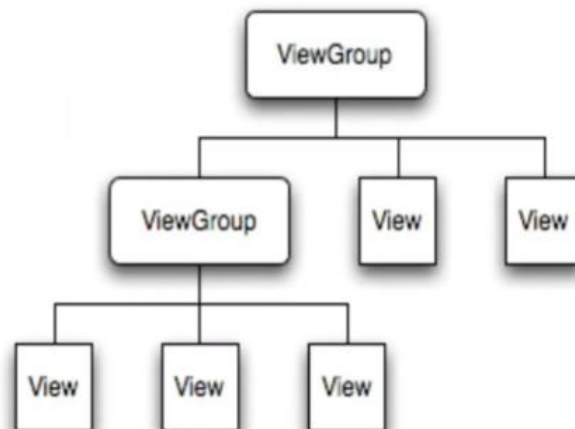
< Intent

Constante	Componente Target	Accion que realiza
ACTION_CALL	activity	Inciar una llamada
ACTION_EDIT	activity	Desplegar data para que sea editada por el usuario
ACTION_MAIN	activity	Iniciar la primera actividad de una tarea que no recibe parametros ni devuelve salidas
ACTION_SYNC	activity	Sincronizar data con un servidor que se encuentra en el dispositivo
ACTION_BATTERY_LOW	broadcast receiver	Adevertencia de que la bateria esta en un nivel bajo
ACTION_HEADSET_PLUG	broadcast receiver	Un set de manos libres o audifonos ha sido conectado o desconectado del dispositivo.
ACTION_SCREEN_ON	broadcast receiver	La pantalla ha sido encendida
ACTION_TIMEZONE_CHANGED	broadcast receiver	La zona horaria ha cambiado, por el usuario o la red.

componentes UI elementos de interfaz de usuario en android la interfaz gráfica es construida a partir de los elementos virus y bill groups virus son las unidades básicas de una interfaz gráfica es la clase base para los controles groups proveen el medio para agrupar virus y ordenarlas un view group sigue siendo un view excepto que posee la capacidad de contener más bills la interfaz gráfica es organizada en un árbol de elementos la wii es dibujada desde la base del árbol hacia abajo

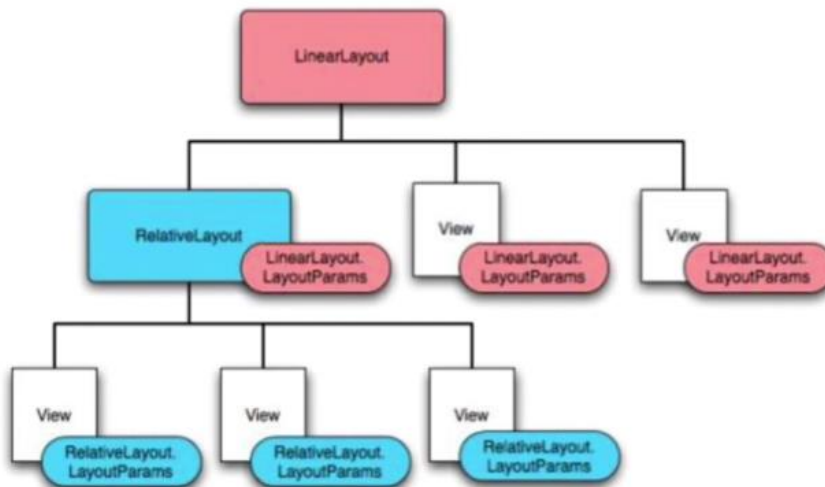
< Elementos

- ☐ Views
- ☐ ViwGroups



layout los layout son elementos no visuales destinados a controlar la distribución posición y dimensiones de los controles que se insertan en su interior parámetros del layout los parámetros del layout son elementos en los cuales definimos como los virus deben de colocarse y dimensionarse dentro del bio group en el cual residen

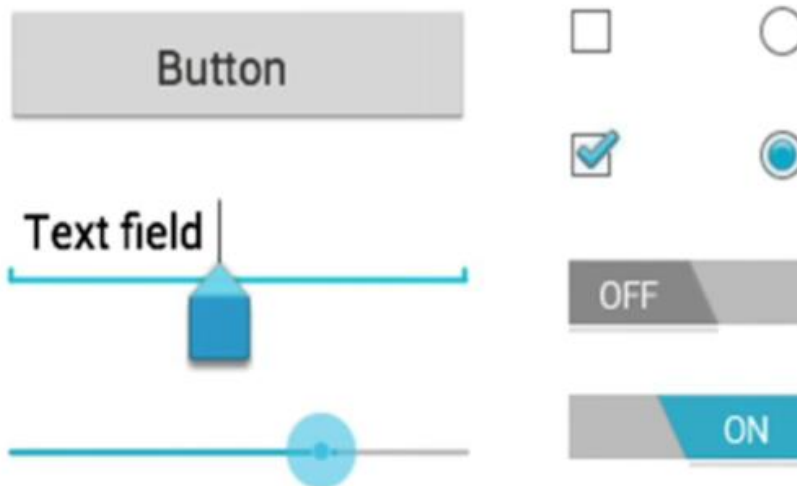
< Layout



controles de entrada los controles de entrada son los componentes interactivos de la interfaz de usuario de tu android ofrece una variedad de controles que puedes usar en tu wii como botones campos de texto barras de búsqueda casillas de verificación botones de zoom botones de activación o desactivación y muchos más



Controles de Entrada



controles comunes control botón botón que el usuario puede presionar o en el que puede hacer clic para realizar una acción las clases relacionadas en la clase duro control campo de texto campo de texto editable puede usar el widget autocomplete test view para crear un widget de entrada de texto que proporcione sugerencias para autocompletar las clases relacionadas con el ipd ex y autocomplete text vibe control casilla de verificación conmutador de selección y de selección que el usuario puede activar o desactivar se debe de usar la casilla de verificación cuando presentamos a los usuarios un grupo de opciones seleccionables que no sean mutuamente exclusivas la clase relacionada es check box control botón de selección similar a las casillas de verificación excepto porque sólo se puede seleccionar una opción en el grupo las clases relacionadas son ready group y radio button y control botón para activar o desactivar botón de activación y desactivación con un indicador luminoso la clase relacionada es paul baran control control del número es una lista desplegable que permite a los usuarios seleccionar un valor de un conjunto la clase relacionada es spinner control selectores es un cuadro de diálogo para que los usuarios seleccionen un valor individual para un conjunto con los botones de arriba y abajo o mediante un gesto de deslizamiento usa un widget de peter kohl para ingresar los valores de fecha mes día año o un widget time picker para ingresar valores de hora hora minuto a mpm a los que se les dará formato automáticamente de acuerdo con la configuración regional del usuario las clases relacionadas son take picker y time picker

</ Controles Comunes

- | | |
|--|------------------------------------|
| <input type="checkbox"/> Botón. | ➤ Button. |
| <input type="checkbox"/> Campo de Texto | ➤ EditText, AutoComplete Text View |
| <input type="checkbox"/> Casilla de verificación | ➤ CheckBox |
| <input type="checkbox"/> Botón de Selección. | ➤ RadioGroup, RadioButton. |

agregar un control de entrada a to you es tan simple como agregar un elemento xml a tu diseño xml por ejemplo en pantalla podemos ver el diseño como un campo de texto y un botón

</ Agregar Control UI

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal">
    <EditText android:id="@+id/edit_message"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="@string/edit_message" />
    <Button android:id="@+id/button_send"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_send"
        android:onClick="sendMessage" />
</LinearLayout>
```

1. DISEÑO DE INTERFACES

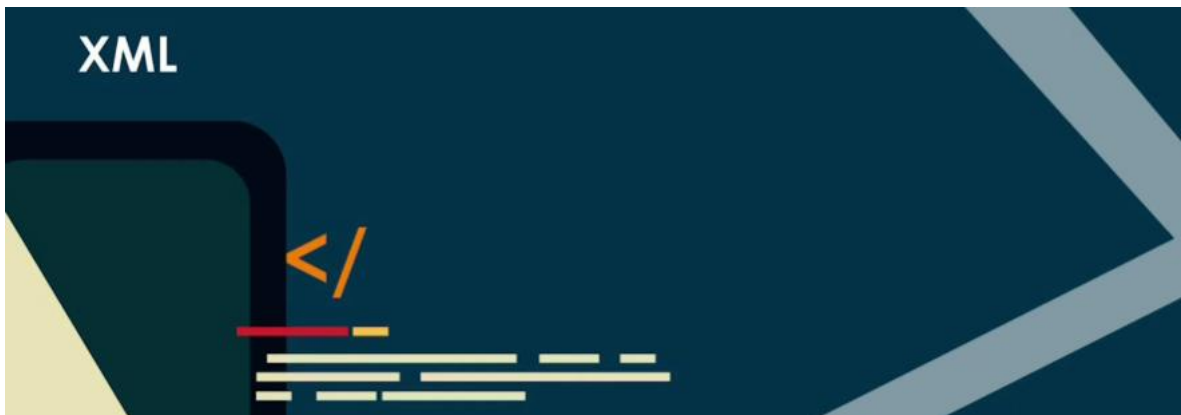
APLICACIONES

1.4.3 Diseño de Pantallas

1.4.3.1. Interfaz de Usuario en Android (XML).

1.4.3.2. Interfaz de usuario en iOS

y xml el lenguaje de marcas extensible es un metalenguaje extensible de etiquetas desarrollado por el world wide web consortium w3c es una simplificación y adaptación del sml y permite definir la gramática de lenguajes específicos xml es el lenguaje en donde el diseñador crear sus propias etiquetas y a partir de las cuales puede desarrollar nuevos estándares de lenguajes de marcado a través de lo que denomina el dt de xml no ha nacido solo para su aplicación en internet sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas se puede usar en bases de datos editores de texto hojas de cálculo y casi cualquier cosa imaginable xml es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura fiable y fácil



existen dos tipos de documentos xml estos son uno documento bien formado dos documento bien construido estos deben cumplir como una estructura correcta todas las etiquetas se cierran todos los atributos deben de estar entrecomillados no debe haber etiquetas traslapadas se deben respetar mayúsculas y minúsculas

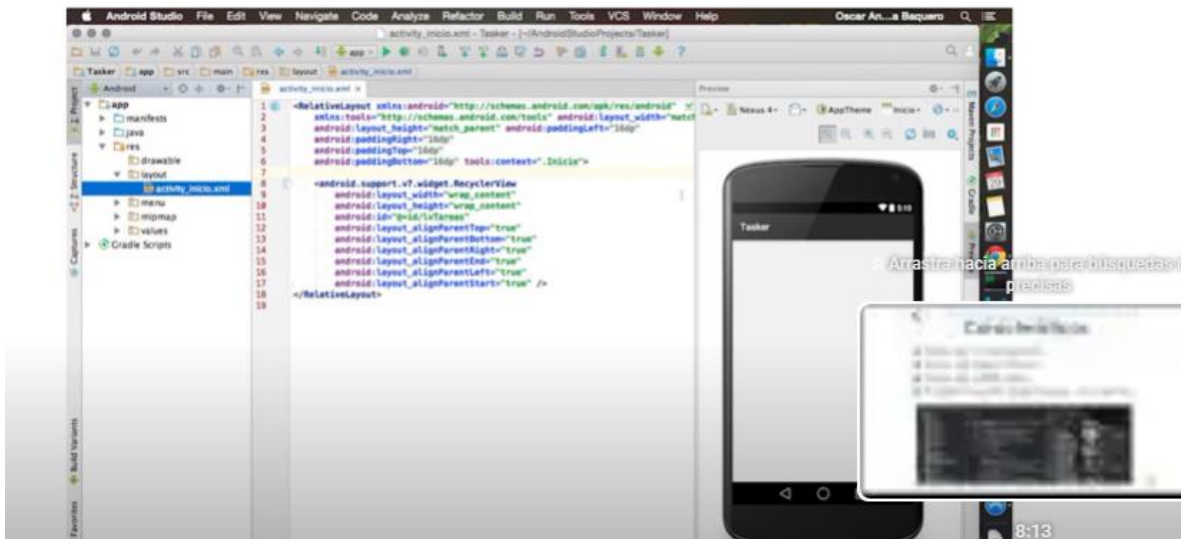
< XML

Existen dos tipos de documentos XML.

- Documentos bien formados.
- Documentos bien contruidos.

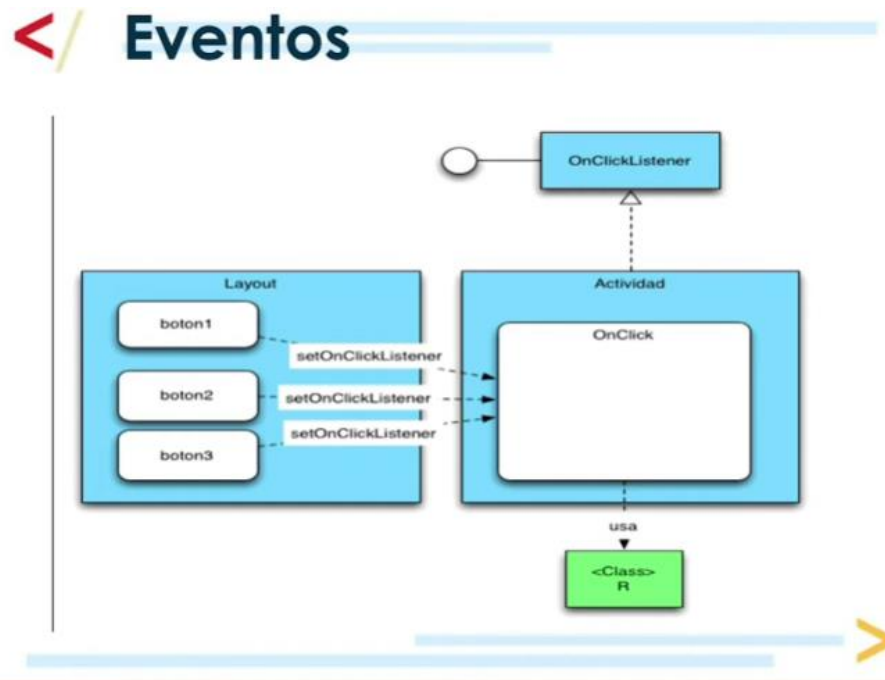
xml en android al usar vocabulario xml de android puedes crear rápidamente diseños de eeuu y de los elementos de pantalla que contienen de la misma manera que creas páginas web en html con una serie de elementos anidados cada archivo de diseño debe contener exactamente un elemento raíz que debe ser un objeto view group una vez que hayas definido el elemento raíz puedes agregar widgets u objetos de diseño adicionales como elementos secundarios para crear gradualmente una jerarquía de vistas que defina tu diseño

< XML en Android



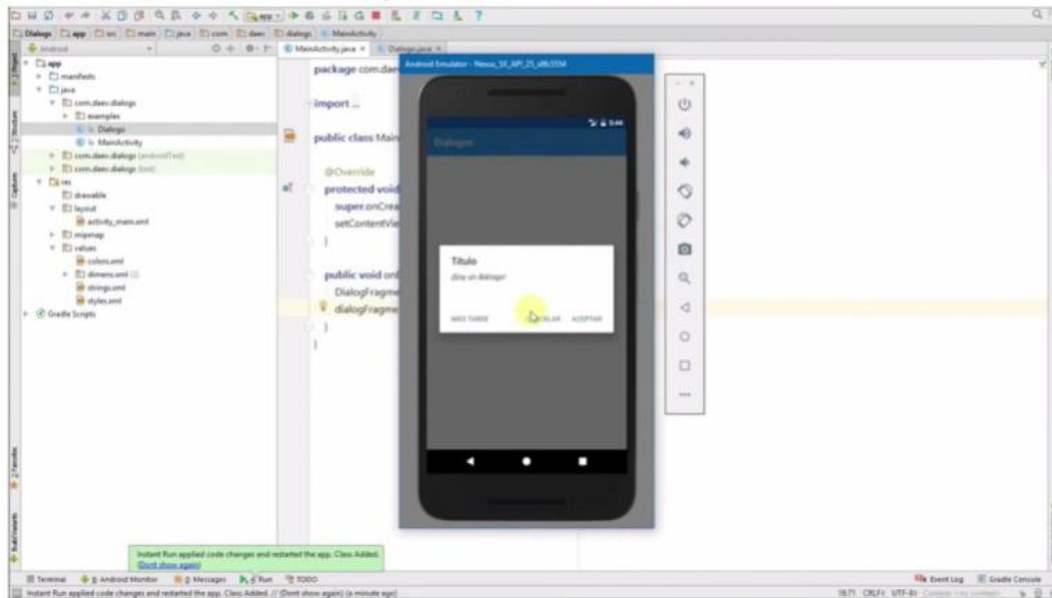
eventos cuando el usuario interacciona con la interfaz y por ejemplo pulsa un botón se generan eventos la aplicación debe especificar qué acciones se deben llevar a cabo cuando se producen

determinados eventos a los objetos que están pendientes de los eventos que se producen se los llama listeners para cada componente que pueda generar un evento como un botón o un campo de texto es posible especificar a qué objeto se le notificará los eventos que produzca la clase view define interfaces que deben cumplir los diferentes tipos de listeners cada una de estas interfaces declara un método que es llamado cuando se recibe el evento al implementar dicho método el programador puede especificar cómo se gestiona el evento veamos algunos de ellos view.OnClickListener para procesar notificaciones de click view.OnLongClickListener para procesar las notificaciones de clic mantenido view.OnFocusChangeListener para procesar la adquisición o pérdida del foco aunque listener para procesar las pulsaciones de teclas físicas del dispositivo sólo se reciben estas notificaciones si el elemento tiene el foco mientras se realiza la pulsación bill and touch edition or cuando el usuario toca pulsando o librando un elemento view.OnContextMenuListener cuando se crea un menú contextual porque el usuario ha mantenido una pulsación larga



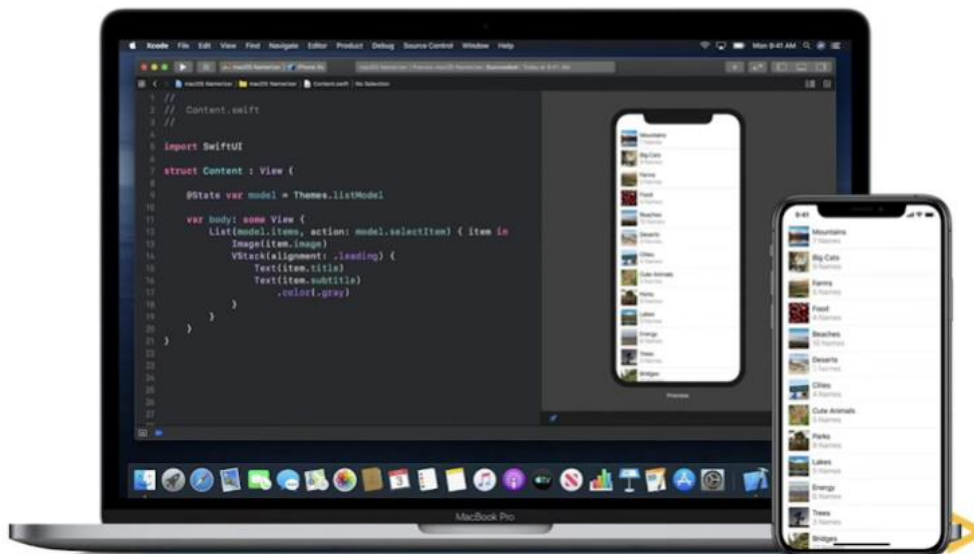
diálogos un diálogo es una pequeña ventana al frente de una actividad para abrir un diálogo este debe ser llamado desde una actividad por el método `showDialog` el entero que se pasa como parámetro al método es un número que identifica al diálogo este método provoca que android genere un callback en `onCreateChild` el cual es una implementación definida por el programador para cargar los diálogos para cerrar el diálogo llamamos al método `dismiss` desde el diálogo o por medio de `dismiss` para cerrarlo desde cualquier punto de la actividad en pantalla puede ver ejemplos de una actividad que posee dos diálogos y la implementación del `Calvari` con `Great Island`

< Diálogos



desarrollo de apps para IOS xcode al desarrollar aplicaciones se utiliza el kit de desarrollo de software y ios sdk y éx con el entorno de desarrollo integrado de apoyo y de éxito ofrece todo lo necesario para crear grandes aplicaciones para el iphone ipod touch y ipad es que trabaja conjuntamente con interface build ver una herencia de next una herramienta gráfica para la creación de interfaces de usuario incluye un editor de código fuente un editor de interfaz gráfica de usuario y muchas otras características escoe utiliza una sola ventana llamada ventana de trabajo que presenta la mayoría de las herramientas que necesita para desarrollar aplicaciones el sdk de ellos amplía el conjunto de herramientas de xcode para incluir las herramientas los compiladores y los marcos necesarios específicamente para ellos

< Xcode



principales características de xcode entre las características más apreciadas de xcode está la tecnología para distribuir el proceso de construcción a partir de código fuente entre varios ordenadores utilizando Bonjour y de analizar los detalles de su proyecto para identificar los errores tanto en la sintaxis y la lógica incluso puede ayudar a arreglar el código entre las principales características tenemos interface wílder interface wílder está integrado como un lienzo de diseño de la idea de xcode esto permite arrastrar componentes gráficos y visualizarlos en perspectivas a cómo quedará en la aplicación final el asistente muestra los archivos relacionados con lo que se está editando como el encabezado la superclase o un controlador apple LLVM es un compilador que está integrado con el líder con soporte para objetivos c&c más más barra de herramientas en esta sección presenta todo lo necesario para seleccionar diferentes dispositivos y simuladores instalar ejecutar código también divide en dos pantallas el área de edición de código y ocultar o mostrar el área de utilidades depuración

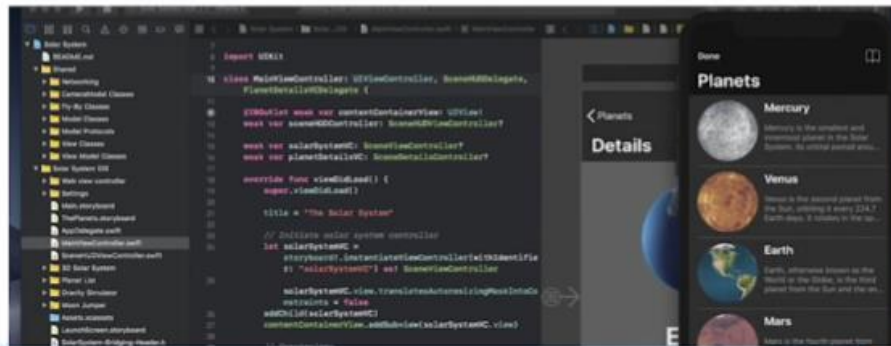


navegación área de navegación aquí se presentan todos los ficheros recursos y carpetas que están en el proyecto en esta sección podremos hacer búsqueda de código indexadas comprobar avisos y fallos en el código y ver dónde colocar los puntos de ruptura en el código entre otras cosas área de depuración en esta sección se ven todos los avisos del sistema también será el lugar en donde se observa el valor de variables en tiempo de ejecución o simplemente avisos y valores de variables introducidos área de utilidades en el área de utilidades estarán todas las propiedades de los diferentes objetos implementados en la interfaz gráfica también se pueden ver y modificar diferentes propiedades de las clases donde el desarrollador se encuentre esta sección ofrece otra funcionalidad que será de gran utilidad si seleccionamos un método de cualquier librería nativa en esta sección aparecerá toda la información acerca del método seleccionado programación orientada a eventos al igual que las aplicaciones android las aplicaciones de ellos se basan en la programación orientada a eventos es decir el flujo de la aplicación se determina por eventos eventos del sistema y las acciones del usuario se debe identificar exactamente qué acciones puede realizar un usuario y lo que ocurre en respuesta a esas acciones.



Características

- ☐ Área de navegación.
- ☐ Área de depuración.
- ☐ Área de utilidades.
- ☐ Programación Orientadas a eventos.



1. DISEÑO DE INTERFACES

APLICACIONES

1.4.3 Diseño de Pantallas

1.4.3.3. Diseño de la interfaz gráfica.

diseño de la interfaz gráfica hacer una aplicación de uso simple para la app store no es una tarea fácil por lo que el uso estándar de componentes y el uso de buenas prácticas es fundamental se deben tener en cuenta algunos aspectos importantes para el diseño de aplicaciones en ellos algunos aspectos importantes son **diferencia** la interfaz de usuario ayuda a la gente a entender e interactuar con el contenido pero nunca compete con el **claridad** el texto es legible en todos los tamaños los iconos son precisos y lúcidos los adornos son sutiles y apropiados y con un enfoque más preciso en la funcionalidad que motiva al diseño **profundidad** capas visuales y movimiento realista impartan vitalidad y aumentan el placer y comprensión de las personas

</ Aspectos Importantes

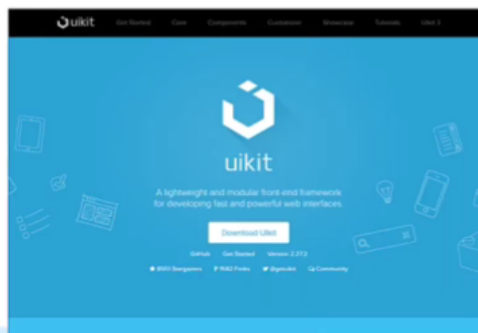
- Deferencia.
- Claridad.
- Profundidad.



casi todas las aplicaciones de ellos utilizan al menos algunos de los componentes de interfaz de usuario definidas por el marco wikis y conocer los nombres funciones y capacidades de estos componentes básicos ayuda a tomar decisiones informadas al diseñar la interfaz de usuario de su aplicación toma en cuenta que muchos de los conceptos que son aplicados para el desarrollo de interfaces de usuario para android pueden ser aplicados para el desarrollo de aplicaciones en ayers por lo que se recomienda revisar estos conceptos

</ Aspectos Importantes

Conocer los nombres, funciones y capacidades de estos componentes básicos ayuda a tomar decisiones informadas al diseñar la interfaz de usuario de su aplicación.



componentes UI en ellos la interfaz gráfica es construida a partir de diversos elementos actions sheets una hoja de acción es un estilo específico de alerta que aparece en respuesta a un control o acción y presenta un conjunto de dos o más opciones relacionadas con el contexto actual use una

hoja de acción para permitir que las personas inicien tareas o soliciten confirmación antes de realizar una operación potencialmente destructiva en pantallas más pequeñas una hoja de acción se desliza hacia arriba desde la parte inferior de la pantalla en pantallas más grandes una hoja de acción aparece de una vez como un pop over



Action Sheets

Una hoja de acción es un estilo específico de alerta que aparece en respuesta a un control o acción, y presenta un conjunto de dos o más opciones relacionadas con el contexto actual.



activity views una actividad es una tarea como copiar favorito o buscar que es útil en el contexto actual una vez iniciada una actividad se puede realizar una tarea de inmediato o solicitar más información antes de continuar las actividades se gestionan mediante una vista de actividad que aparece como una hoja o ventana emergente según el dispositivo y la orientación use actividades para dar a las personas acceso a servicios personalizados o tareas que su aplicación puede realizar



Activity Views

Una actividad es una tarea, como Copiar, Favorito o Buscar, que es útil en el contexto actual.

alerts las alertas transmiten información importante relacionada con el estado de su aplicación o dispositivo y a menudo solicitan comentarios una alerta consta de un título un mensaje opcional uno o más botones y campos de texto opcionales para recopilar información además de estos elementos configurables la apariencia visual de una alerta es estática y no se puede personalizar



Alerts

Una alerta consta de un título, un mensaje opcional, uno o más botones y campos de texto opcionales para recopilar información.



collections una colección gestiona un conjunto ordenado de contenido como un conjunto de fotos y lo presenta en un diseño personalizable y muy visual debido a que una colección no aplica a un formato estrictamente lineal es particularmente adecuada para mostrar elementos que varían en tamaño en términos generales las colecciones son ideales para mostrar contenido basado en imágenes los fondos y otras vistas decorativas pueden implementarse opcionalmente para distinguir visualmente subconjuntos de elementos



Collections

En términos generales, las colecciones son ideales para mostrar contenido basado en imágenes.

images views una vista de imagen muestra una sola imagen o una secuencia animada de imágenes sobre un fondo transparente u opaco dentro de una vista de imagen las imágenes se pueden estirar escalar ajustar o fijar a una ubicación específica las vistas de imagen no son interactivas por defecto



Image Views

Dentro de una vista de imagen, las imágenes se pueden estirar, escalar, ajustar o fijar a una ubicación específica.

pages es un controlador de vista de página proporciona una forma de implementar navegación lineal entre páginas de contenido como en un documento libro bloc de notas o calendario un controlador de vista de página utiliza uno de los dos estilos para administrar las transiciones entre páginas durante la navegación desplazamiento o desplazamiento de página una transición de desplazamiento no tiene una apariencia específica las páginas se desplazan fluidamente de una a la siguiente una transición de giro de página hace que las páginas se enrollan al deslizar por la pantalla girando como páginas en un libro físico



Pages

Un controlador de vista de página proporciona una forma de implementar navegación lineal entre páginas de contenido, como en un documento, libro, bloc de notas o calendario.

popovers una ventana emergente es una vista transitoria que aparece sobre otro contenido en pantalla cuando toca un control o un área por lo general un pop over incluye una flecha que apunta a la ubicación de la que surgió pop auvers puede ser no modal o modal un pop over no modal se descarta tocando otra parte de la pantalla o un botón en el pop o ver un pop over modal se descarta tocando un botón cancelar u otro botón en el pop over



Popovers

- ☐ Vista transitoria.
- ☐ incluye una flecha que apunta a la ubicación de la que surgió.
- ☐ puede ser no modal o modal.

scroll views una vista de desplazamiento permite a los usuarios explorar contenido como texto en un documento o una colección de imágenes que es más grande que el área visible a medida que las personas deslizan desplazan arrastran tocan y pellizcan una vista desplazamiento sigue el gesto revelando o ampliando el contenido de una manera que se siente natural una vista de desplazamiento en sí no tiene apariencia pero muestra indicadores de desplazamiento transitorios a medida que las personas interactúan con ella una vista de desplazamiento también se puede configurar para operar en modo de paginación donde el desplazamiento revela una página de contenido completamente nueva en lugar de moverse por la página actual



Scroll Views

Una vista de desplazamiento permite a los usuarios explorar contenido, como texto en un documento o una colección de imágenes, que es más grande que el área visible.

- ☐ Sigue el gesto.
- ☐ No tiene apariencia.
- ☐ Se puede configurar.

split views una vista dividida gestiona la presentación de dos paneles de contenido en paralelo con contenido persistente en el panel primario e información relacionada en el panel secundario cada panel puede contener cualquier variedad de elementos incluidas barras de navegación barras de herramientas barras de pestañas tablas colecciones imágenes mapas y vistas personalizadas las vistas divididas a menudo se usan con contenido filtra blé aparece una lista de categorías de filtro en el panel primario y los resultados filtrados para la categoría seleccionada se muestran en el panel secundario



Split Views

Una vista dividida gestiona la presentación de dos paneles de contenido en paralelo, con contenido persistente en el panel primario e información relacionada en el panel secundario.

tables una tabla presenta los datos como una lista de desplazamiento de una sola columna de filas que se pueden dividir en secciones o grupos text views una vista de texto muestra contenido de texto con estilo de varias líneas las vistas de texto pueden tener cualquier altura y permitir el desplazamiento cuando el contenido se extiende fuera de la vista por defecto el contenido dentro de una vista de texto está alineado a la izquierda y usa la fuente del sistema en negro si una vista de texto es editable aparece un teclado cuando toca dentro de la vista web views una vista web carga y muestra contenido web como html incrustado y sitios web directamente dentro de su aplicación el correo utiliza una vista web por ejemplo para mostrar contenido html en los mensajes



Tables, Text Views, Web Views

Tables: Lista de desplazamiento de una sola columna de filas.

Text Views: Pueden tener cualquier altura y permitir el desplazamiento cuando el contenido se extiende fuera de la vista.

Web Views: Muestra contenido web como HTML .

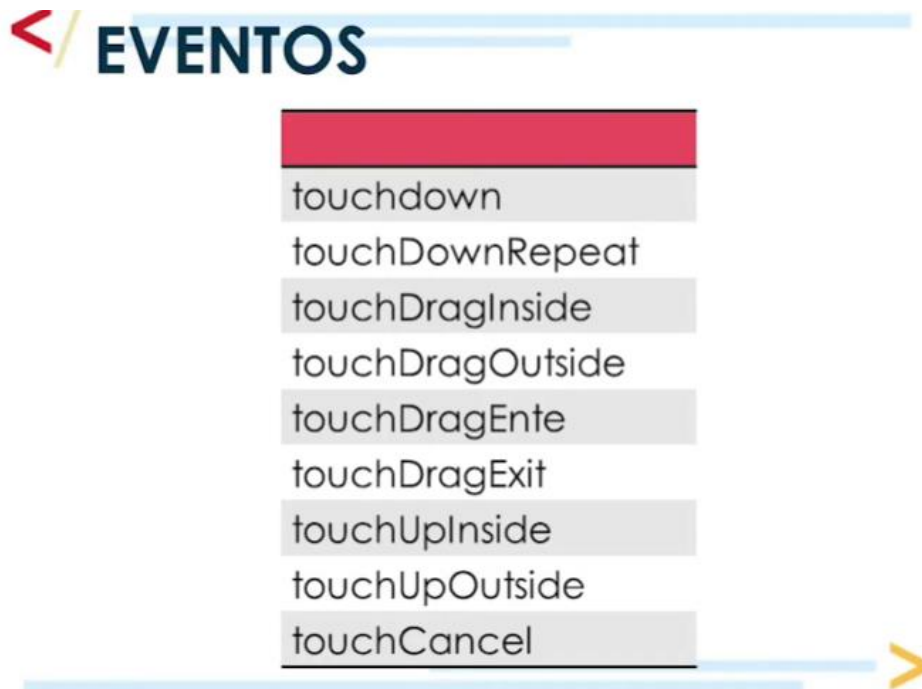


APLICACIONES

1.4.3 Diseño de Pantallas

1.4.3.4. Eventos en los dispositivos.

eventos cuando el usuario interacciona con la interfaz y por ejemplo pulsa un botón se generan eventos en ellos tenemos por ejemplo los siguientes eventos que son manejados por la clase y hay control point event la cual describe los eventos posibles para los controles veamos algunos eventos touch down evento de toque deslizamiento hacia abajo en el control touch down repeat evento de aterrizaje repetido en el control para este evento el valor del método y whiteout tap count es mayor que 1 touch rack inside evento donde se arrastra un dedo dentro de los límites del control touch drag outside evento donde se arrastra un dedo justo fuera de los límites del control touch drag enter evento donde se arrastra un dedo dentro de los límites del control touch drag exit evento donde se arrastra un dedo desde un control hacia afuera de sus límites a inset evento de retoque en el control donde el dedo está dentro de los límites del control touch outside evento de retoque en el control donde el dedo está afuera de los límites del control touch cancel evento del sistema que cancela los toques actuales para el control



value changed punto que arrastrando o manipulando un control haciendo que emita una serie de valores diferentes por aimar y action trigger una acción semántica activada por botones ereading de picking un toque para iniciar una sesión de edición en un white & field ereading change un toque haciendo un cambio de edición en un white & field air y tintín en un toque que finaliza una sesión de edición en un white ex field objetivo sí audio tics y es un lenguaje orientado a objetos elegantes con el cual se pueden desarrollar aplicaciones de años hay que entender este lenguaje respecto a su estructura y sintaxis con el fin de utilizarlo para la mayoría de los entornos de aplicaciones aunque se pueden utilizar otros lenguajes de programación como sí sí plas plas os web para el desarrollo no se puede construir una aplicación para ellos sin objetivo

</ EVENTOS



valueChanged
primaryActionTriggered
editingDidBegin
editingChanged
editingDidEnd

</ OBJECTIVE-C

Es un lenguaje orientado a objetos.
No se puede construir una aplicación para iOS sin Objective-C



Objective-C

objective-C es un lenguaje bastante árido y con muchas diferencias frente a lenguajes de propósito más general como si sharp o java por eso muchos programadores que se meten en la programación para mac o para iphone ipad encuentran su principal barrera en comprender bien y utilizar objetos y una de las primeras cosas que llaman la atención es que objetiv si es un lenguaje compilado pero también es al mismo tiempo un lenguaje enlazado esto quiere decir básicamente que el resultado del compilador no es el programa final sino que existe una segunda fase que lleva a cabo el enlazador

</ OBJECTIVE-C



- Lenguaje bastante árido
- Es un lenguaje compilado.
- Lenguaje enlazado

relación entre compilador y enlazador el enlazador es el responsable de recoger el código compilado y de juntarlo con otro código compilado el de las librerías que el programa utilice sean de sistema o de terceros este concepto sorprende si se tiene conocimiento de si sharp o java en estos lenguajes el concepto de enlazado no existe si quieres utilizar una librería un ensamble o bien un fichero punto heart te limitas a referenciar lo desde tu proyecto automáticamente sin que tengas que hacer nada más todas las clases definidas en esta librería las tienes disponibles en tiempo de ejecución basta con que el ordenador de destino tenga las librerías instaladas para que tu programa funcione en estos lenguajes esto es posible porque el código compilado ya sean ensamblados de punto net o archivos punto class de java expone metadatos que informan de qué tipos existen qué métodos tienen y en y todo lo que el compilador necesita para poder compilar el código

</ ENLAZADOR

- ✓ El **enlazador** es el responsable de recoger el código compilado y de "juntarlo" con otro código compilado.
- ✓ El concepto de enlazado no existe.
- ✓ En tiempo de ejecución basta con que el ordenador de destino tenga las librerías instaladas para que tu programa funcione.

Objective C cuando se usa una librería ya sea del sistema o de terceros el compilador no la necesita para nada tampoco puede hacer nada con ella el código compilado no expone meta datos que permitan leerlo para saber qué tipos hay definidos y qué métodos tienen así que para poder utilizar

la librería se debe suministrar el compilador uno o varios archivos de código fuente que definan la clase y todos sus métodos tan solo la firma de los métodos no la implementación dichos archivos reciben el nombre de archivos de cabecera y tienen o suelen tener la extensión punto h y por supuesto no los creas tú sino que los crea quien haya hecho la librería así que cuando se usa una librería en objective C se tendrá el código compilado y uno o varios archivos de cabecera que contienen la definición de las clases y métodos de esa librería el compilador tan solo utiliza la información de los ficheros de cabecera y hace el enlazador quien recoge el código compilado generado por el compilador los binarios de las librerías y lo enlaza todo en el programa final



COMPILADOR

- ❖ El código compilado no expone metadatos.
- ❖ Se debe suministrar al compilador uno, o varios, archivos de código fuente que definan la clase y todos sus métodos



en resumen podemos decir el compilador tan solo tiene en cuenta los archivos de cabecera por lo tanto si una clase no está definida en los archivos de cabecera es como si no existiese no podrá usarse porque no existirá para el compilador eso elimina la necesidad de tener una visibilidad para las clases como es el caso de interna al clase en sí sharp el enlazador es un paso posterior al compilador eso significa que un código puede compilar bien es decir que el código objetivo si es válido y que todos los headers existen pero tener errores del inglésador significa que alguna de las

librerías no se encuentra o no contiene realmente las clases o métodos que define el archivo de



RESUMEN

El compilador solo tiene en cuenta los archivos de cabecera.

Si una clase no está definida en los archivos de cabecera no podrá usarse porque no existirá para el compilador