

PRÁCTICA PMDM04

**Diseño. Gestores de colocación.
Internacionalización y
localización**

27/02/2024

CÉSAR BOUZAS SOTO

Contenido

1	¿Qué es "ConstraintLayout"? ¿Para que sirve? Pon un ejemplo.	2
2	¿Qué es la compatibilidad de pantalla? ¿Cómo se gestiona? Pon un ejemplo.	3
3	Pon Ejemplos de tipos de Recursos.	4
4	Explica cómo se gestiona a internalización das aplicación	5

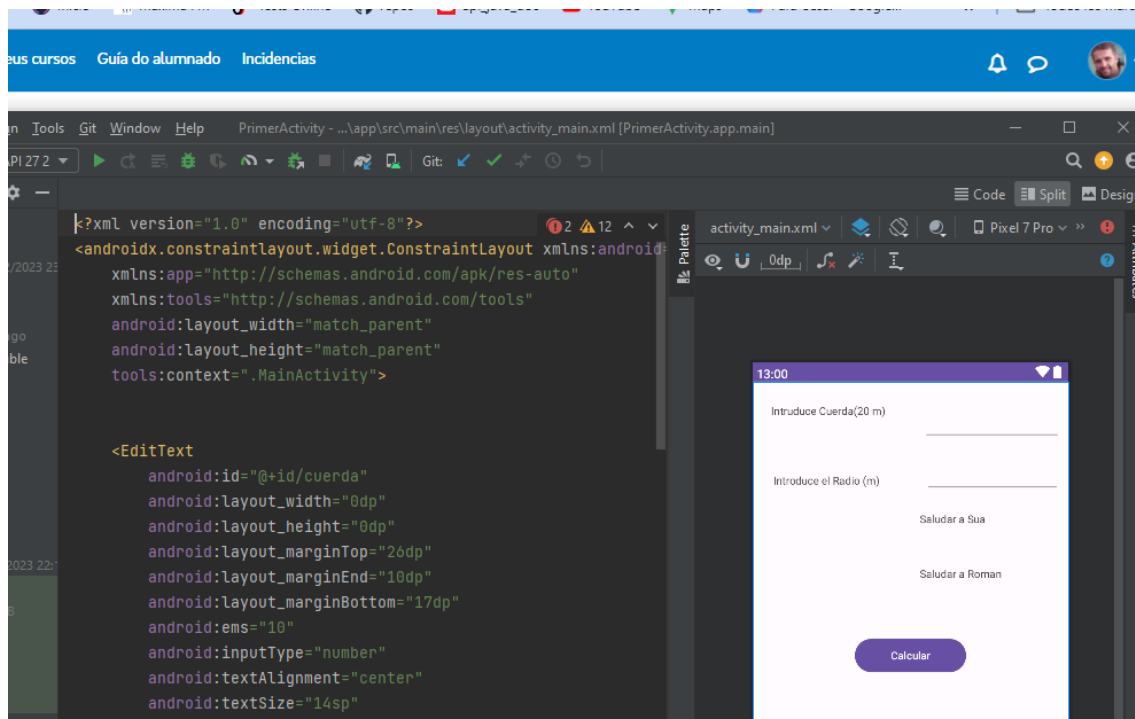
1 ¿Qué es "ConstraintLayout"? ¿Para que sirve? Pon un ejemplo.

El "ConstraintLayout" es un componente esencial en el desarrollo de aplicaciones para dispositivos Android. En esencia, se trata de un sistema de diseño flexible y poderoso que permite a los desarrolladores crear interfaces de usuario dinámicas y adaptables. Su utilidad radica en la capacidad de definir relaciones entre los elementos de la interfaz de usuario de manera intuitiva, lo que facilita la creación de diseños complejos y responsivos.

Por ejemplo, considera una aplicación de redes sociales donde se muestra una lista de publicaciones. Con ConstraintLayout, puedes especificar que cada elemento de la lista (como una imagen de perfil, nombre de usuario y contenido de la publicación) esté posicionado de manera relativa entre sí y al borde de la pantalla. Esto garantiza que, sin importar el tamaño de la pantalla del dispositivo o la cantidad de contenido, la interfaz se ajuste y se vea bien.

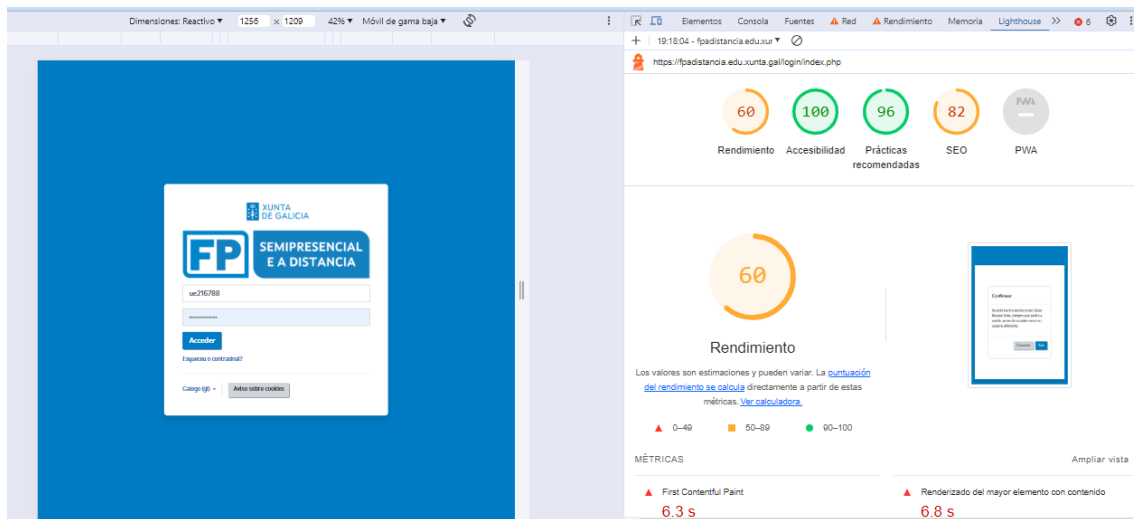
En un nivel más técnico, ConstraintLayout utiliza restricciones para definir la posición y el tamaño de los elementos de la interfaz de usuario. Estas restricciones establecen relaciones espaciales entre los elementos, como alineación, margen y centrado, lo que permite un diseño más preciso y fácil de mantener.

En resumen, ConstraintLayout es una herramienta fundamental para el desarrollo de aplicaciones Android, ya que simplifica la creación de interfaces de usuario adaptables y atractivas, lo que mejora la experiencia del usuario final.



2 ¿Qué es la compatibilidad de pantalla? ¿Cómo se gestiona? Pon un ejemplo.

La compatibilidad de pantalla en Android se refiere a la capacidad de una aplicación para adaptarse y funcionar correctamente en una variedad de dispositivos con diferentes tamaños, resoluciones y densidades de pantalla. Se gestiona en Android utilizando diversas técnicas y recursos proporcionados por el sistema operativo y las herramientas de desarrollo.



Una de las principales herramientas para gestionar la compatibilidad de pantalla en Android es el sistema de recursos. Android proporciona diferentes carpetas de recursos (como drawable, layout, values, etc.) que permiten especificar diferentes versiones de archivos de recursos para diferentes configuraciones de dispositivos. Por ejemplo, puedes proporcionar diferentes tamaños de imágenes para diferentes densidades de pantalla, y Android seleccionará automáticamente la imagen adecuada según el dispositivo en el que se ejecute la aplicación.

Además, se utilizan unidades de medida relativas en el diseño de la interfaz de usuario, como dp (píxeles independientes de la densidad) en lugar de píxeles absolutos, lo que permite que los elementos de la interfaz de usuario se escalen correctamente en diferentes tamaños de pantalla y densidades de píxeles.

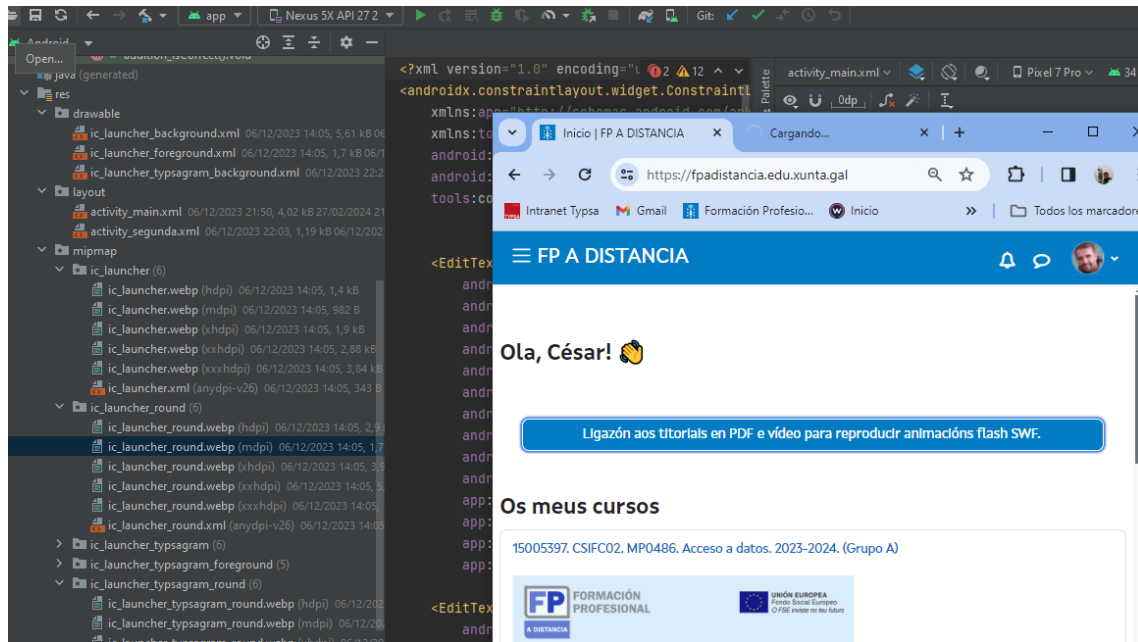
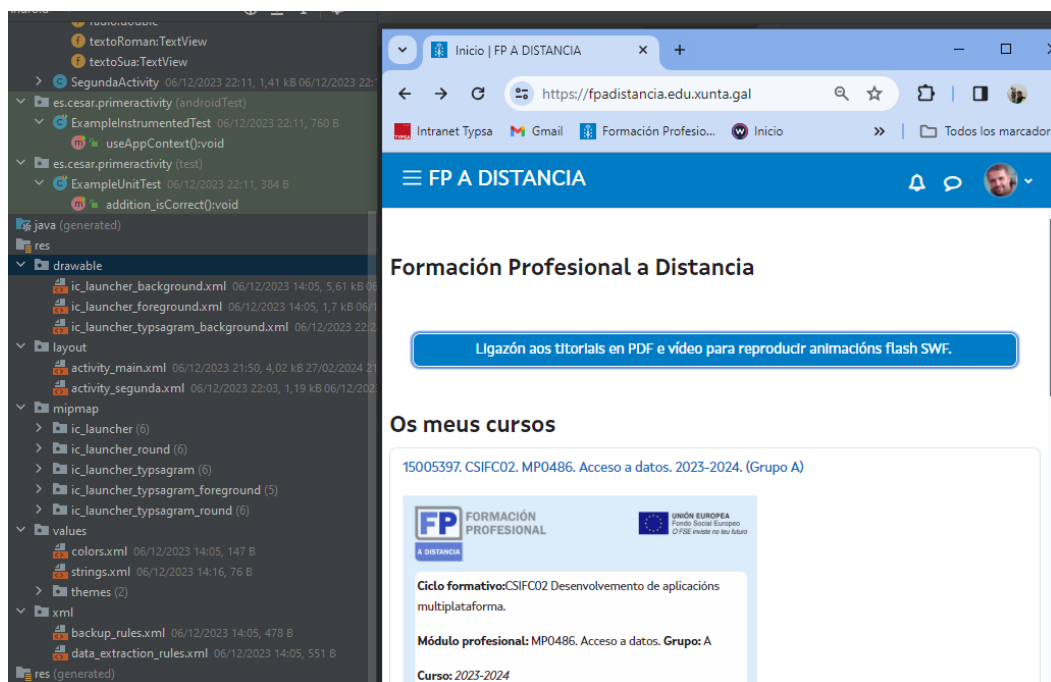


Ilustración 1 imagenes con distintas densidades en el desplegable izquierdo

3 Pon Ejemplos de tipos de Recursos.

En Android, los recursos son elementos que se utilizan para proporcionar datos estáticos, como imágenes, cadenas de texto, estilos, dimensiones, colores, archivos de audio, diseños de interfaz de usuario, etc. Estos recursos se utilizan para separar los datos de la lógica de la aplicación, lo que facilita la personalización, la internacionalización y la gestión de cambios en la interfaz de usuario.



Imágenes: Incluyen íconos, fondos, imágenes de botones, etc.

Dibujables: Archivos XML que definen formas y estados de los gráficos, como formas geométricas o estados de botones.

Strings XML: Almacenan cadenas de texto utilizadas en la aplicación, lo que facilita la internacionalización y la localización.

Styles XML: Definen atributos de estilo que se pueden aplicar a las vistas para mantener la consistencia visual en la aplicación.

Themes XML: Define el aspecto global de la aplicación, incluidos colores de fondo, estilos de texto predeterminados, etc.

Dimensiones XML: Almacenan valores de tamaño que se pueden reutilizar en toda la aplicación, lo que facilita la consistencia en el diseño.

Colors XML: Almacenan valores de color que se pueden reutilizar en toda la aplicación, lo que facilita la consistencia en el diseño y permite cambios rápidos y globales en la paleta de colores.

Menus XML: Definen la estructura y el contenido de los menús en la aplicación.

Animaciones XML: Definen animaciones que se pueden aplicar a las vistas para crear efectos visuales dinámicos.

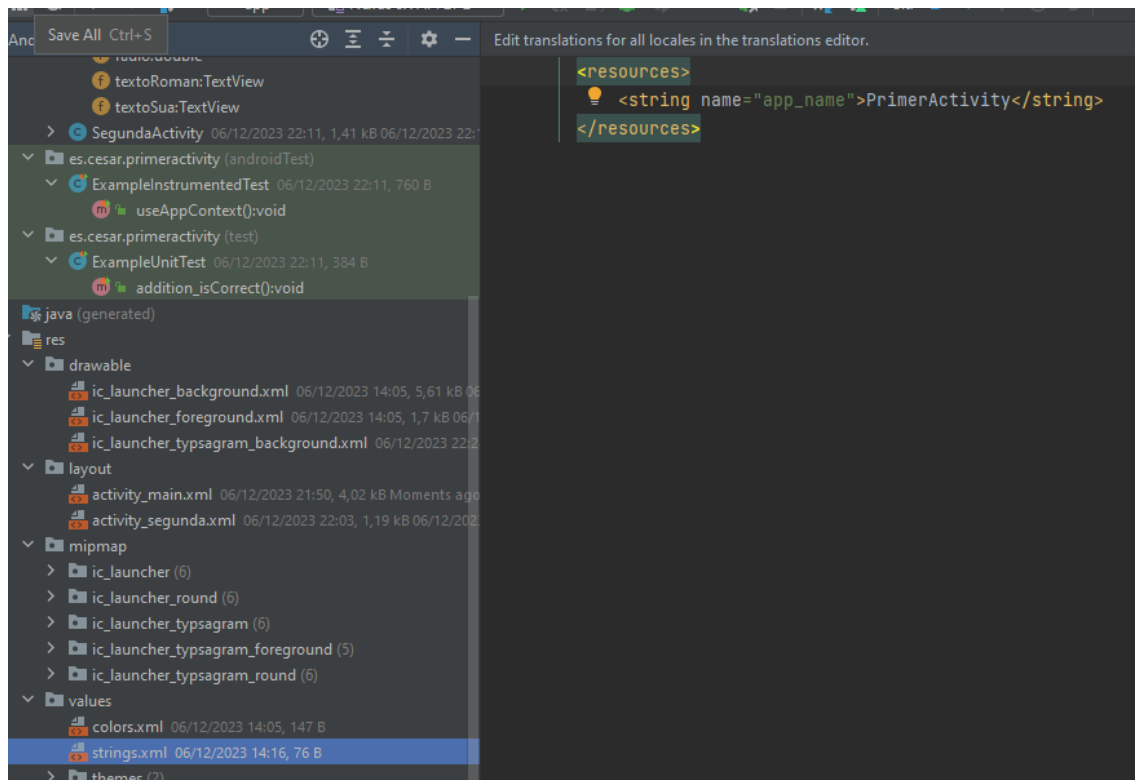
Archivos de Audio/Video: Almacenan archivos multimedia utilizados en la aplicación, como efectos de sonido o clips de vídeo.

Estos recursos se organizan en directorios específicos dentro del proyecto de Android y se accede a ellos mediante identificadores generados automáticamente por el sistema de compilación de Android. Esto hace que sea fácil referenciar y utilizar estos recursos en el código de la aplicación.

4 Explica cómo se gestiona la internacionalización de la aplicación

La internacionalización en Android se refiere al proceso de adaptar una aplicación para que pueda funcionar en diferentes idiomas y regiones del mundo. Algunos pasos clave para gestionar la internacionalización en Android:

Recursos de cadenas: En lugar de codificar cadenas de texto directamente en el código fuente, tienen que ser almacenarlas en archivos XML separados. Android proporciona un sistema de recursos que permite definir diferentes versiones de estos archivos para diferentes idiomas. Por ejemplo, un archivo `strings.xml` para inglés (`values/strings.xml`) y otro para español (`values-es/strings.xml`).



Recursos de imágenes: Si la aplicación incluye imágenes con texto incrustado, como botones o iconos, se deben tener distintas versiones de estas imágenes adaptadas para diferentes idiomas si contienen texto. Esto asegura que el texto sea legible y relevante para los usuarios de diferentes idiomas.

Formatos de fecha y hora: Las clases de formato de fecha y hora de Android (`DateFormat`, `SimpleDateFormat`, etc.) en lugar de formatos estáticos. Esto permite que la aplicación formatee fechas y horas de acuerdo con la configuración regional del dispositivo.

Soporte para diferentes unidades de medida y formatos numéricos: Algunos países utilizan sistemas de medidas diferentes o formatos numéricos distintos. Android proporciona clases como `NumberFormat` para formatear números de acuerdo con la configuración regional del dispositivo.