

UT1 - Introducción al diseño de interfaces

1.	El diseño de una interfaz. El diseño gráfico	1
1.1.	¿Qué busca la interfaz de usuario?	2
1.2.	Planteamiento y diseño de una interfaz	2
2.	El color	3
3.	El sistema RGB	3
3.1.	Matiz, saturación y brillo	4
3.2.	La importancia de los colores	5
4.	Uso de colores en Android Studio	6
4.1.	Uso de los colores	8
4.2.	Selección de colores en las APP	10
5.	Bibliografía y webgrafía	10

1. El diseño de una interfaz. El diseño gráfico

El diseño gráfico consiste en programar, proyectar y realizar comunicaciones visuales de aplicaciones u otro tipo de herramientas software que generalmente serán transmitidos por medios industriales. A día de hoy, este área de desarrollo tiene a sus propios profesionales, denominados **diseñadores gráficos**. Sus funciones:

- Función estética
- Función publicitaria
- Función comunicativa

Se distinguen 4 grupos de elementos en el diseño de interfaces:

- Elementos conceptuales
- Elementos visuales
- Elementos de relación
- Elementos prácticos

Por lo tanto, la interfaz será un conjunto de elementos gráficos y un diseño de su distribución que permite una mejor presentación y navegación en la aplicación. Si no existen ambos factores unidos, el resultado final no será óptimo, pudiendo tirar por tierra todo el buen trabajo que lleva en el backend. Ejemplo:



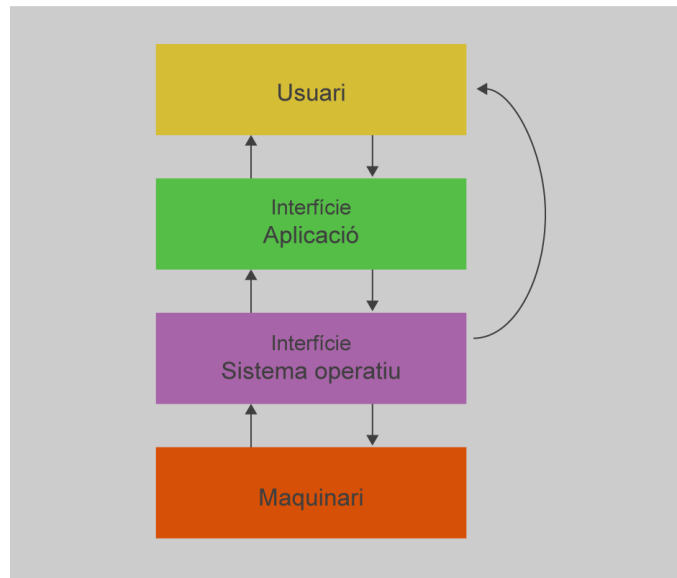
Como se puede apreciar, la figura, las dos imágenes son la misma aplicación. En la izquierda tenemos un mal diseño, ya que los elementos no aparecen diferenciados, no se puede leer bien el texto, no se aprecian correctamente las imágenes, etc. Mientras que en la imagen de la derecha, la navegación del usuario es mucho más intuitiva, proporcionando un mayor grado de satisfacción al usarla.

Actividad 1.1: Busca alguna aplicación que, al igual que el ejemplo anterior, se pongan de manifiesto las diferencias entre una aplicación de buen diseño y otra con un mal diseño.

- ¿Qué te hace clasificarlas dentro de estas dos categorías?
- ¿Qué partes consideras que son más importantes en el diseño?

1.1. ¿Qué busca la interfaz de usuario?

Para responder a esta pregunta debemos recordar la siguiente figura, la cual muestra la relación que existe entre los diferentes elementos que intervienen en un equipo:



Con ello, podemos entender que la interfaz de usuario busca establecer una relación entre el usuario que utiliza la aplicación y el hardware del equipo mediante el uso del S.O.

Las interfaces de usuario son muchas y variadas, tantas como hardware que debemos utilizar, como por ejemplo la siguiente imagen.



1.2. Planteamiento y diseño de una interfaz

En la construcción de una aplicación debemos tener en cuenta diferentes fases, desde la definición de los objetivos que persigue nuestro proyecto hasta el diseño visual definitivo, pasando por las especificaciones funcionales.

Las fases del diseño pueden verse en la siguiente pila:



Los elementos clave que debemos tener en cuenta para crear nuestra aplicación son:

- Elementos que van a formar parte de nuestra aplicación
- Extensión de la aplicación
- patrones de diseño que van a utilizarse para estandarizar el diseño de interfaces
- Aspectos técnicos de usabilidad

2. El color

El ojo humano solo es capaz de percibir los denominados **colores aditivos**; a través de la combinación de éstos es posible obtener el resto de los colores, estos son:

- Azul (B)
- Rojo (R)
- Verde (G)

3. El sistema RGB

De la misma forma, un ordenador es capaz de obtener la representación de todos los colores utilizando el sistema RGB, indicando la proporción de cada uno de ellos dentro de dicha combinación, dando lugar a toda la paleta de colores conocida.

Para representar cada color de forma que pueda ser traducido por el ordenador se utilizan 8 bits para codificar cada uno de los colores aditivos, es decir, se establece la proporción de cada color que va a formar parte de los 3. La escala monocromática de un color tendrá 256 (2^8) valores.

A la hora de representar cada uno de los colores, es posible utilizar tanto el sistema hexadecimal (donde codificamos con 4 bits cada uno de los dígitos de la cifra) como el decimal (de 0 a 255).

El número de combinación de colores se calcula multiplicando el número máximo de grados en la escala monocromática de cada color: $256 \times 256 \times 256$, obteniendo 16777256 colores.

El color estaría formado por:

Rojo = 255	1111 1111	FF
Verde = 255	1111 1111	FF
Azul = 0	1111 1111	00

En hexadecimal quedaría expresado: #FFFF00



Círculo Cromático

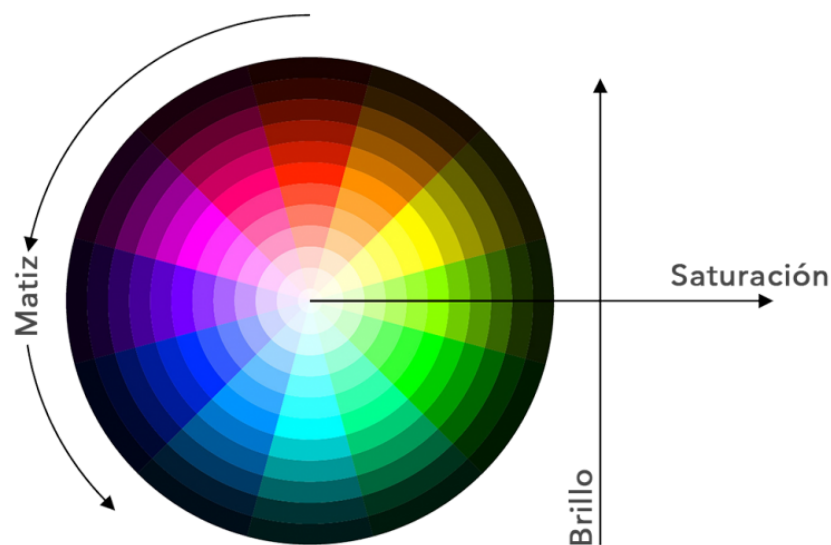
3.1. Matiz, saturación y brillo

Además del grado en la escala monocromática de cada uno de los colores RGB, los colores representan 3 propiedades que permiten distinguirse unos de otros:

- El **matiz**. Atributo que permite distinguir un color de otro. Los 3 matices primarios son los colores aditivos, verde, rojo y azul; el resto de los colores se obtiene

mezclando estos 3. El matiz permite definir 2 colores como complementarios cuando están uno frente al otro en el círculo cromático.

- La **saturación**. Este atributo define la intensidad de un color. Pueden relacionarse con el ancho de banda de luz que estamos visualizando, por tanto, queda condicionado por el nivel de gris presente en un color: cuando mayor sea el nivel de gris, menos saturado será un color, y será menos intenso.
- El **brillo**. Atributo que define la cantidad de luz de un color. Representa lo oscuro (si se le añade negro) o claro (si se le añade blanco) que es un color respecto de su patrón, es decir, respecto del color puro sin modificar el brillo. En una composición de colores en diseño gráfico, cuanto más brillante sea un color, más cerca parece estar.



Estas propiedades nos permiten definir los colores como cromáticos, complementarios o cercanos, así como definir el contraste de color.

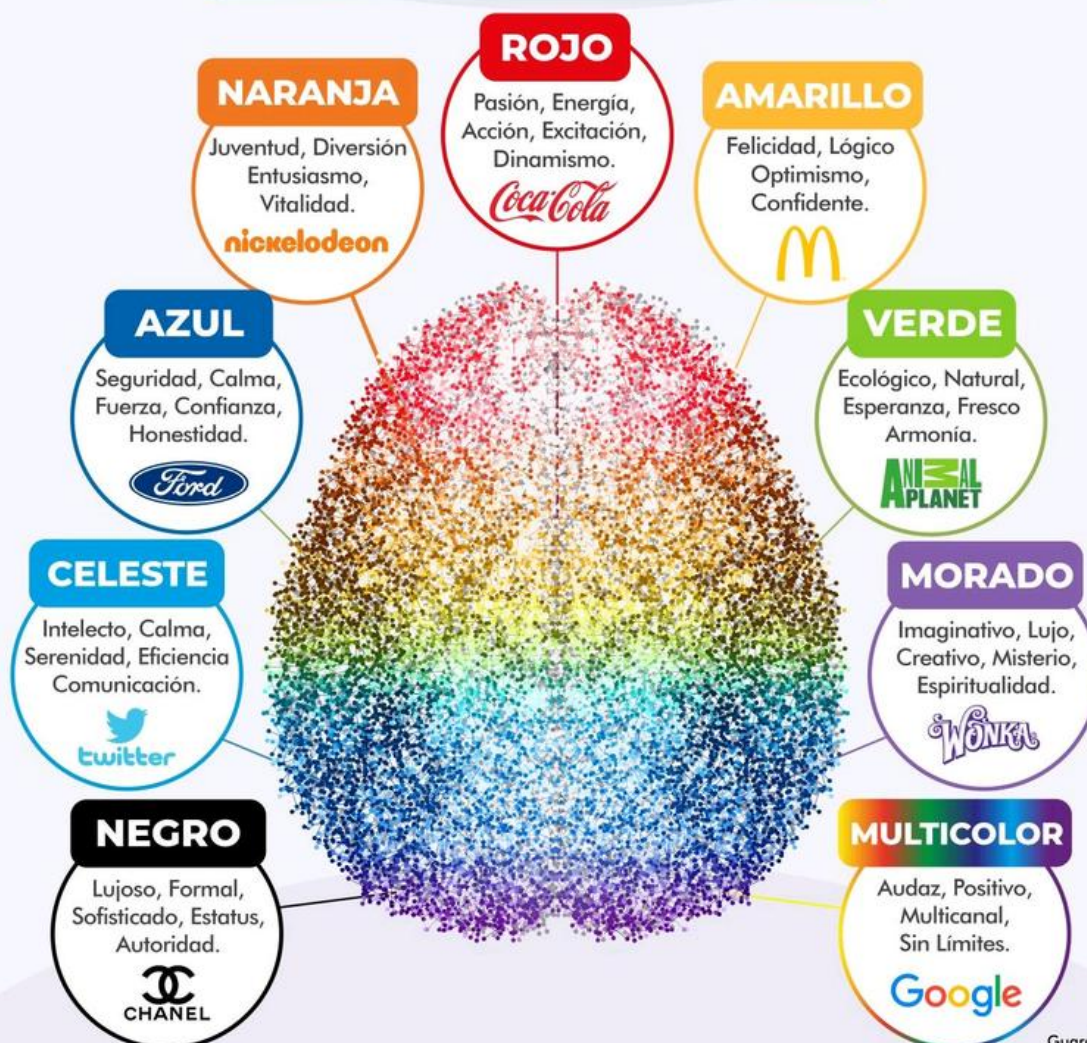
Actividad 1.2: Busca los valores que codifican los siguientes colores y exprésalos en hexadecimal: magenta, white, purple, black y coral.

3.2. La importancia de los colores

Como se puede observar en la imagen siguiente, cada color nos evoca una emoción y una acción, por lo tanto, a la hora de configurar vuestra App y vuestro logo sería recomendable elegir unos colores acorde con el fin de vuestra App:

PSICOLOGÍA DEL COLOR EN MARKETING Y PUBLICIDAD

Algunas emociones y acciones que representa cada color.



Elia Guardiola

Guarda
Comparte
Comenta

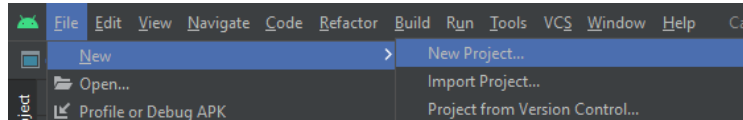
4. Uso de colores en Android Studio

Llegados a este punto, vamos a ver cómo se pueden utilizar los colores en Android. Se ha elegido Android Studio para realizar este módulo debido a un acuerdo entre los miembros del equipo docente y basándose en el entorno socioeconómico de Castellón. No obstante, se ha de tener en cuenta que los conceptos de diseño de interfaces que se aprendan aquí podrán ser desarrollados en cualquier aplicación y entorno de desarrollo.

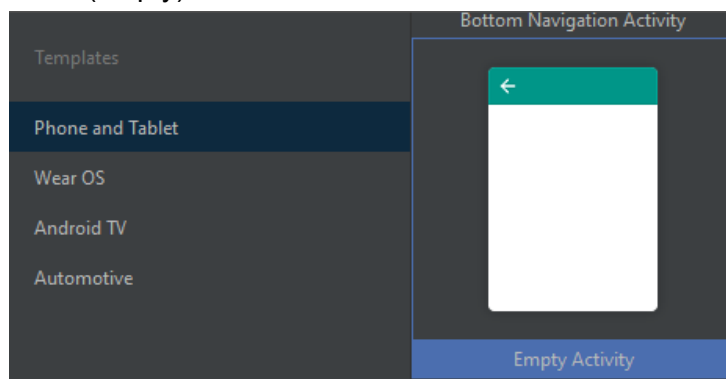
android studio



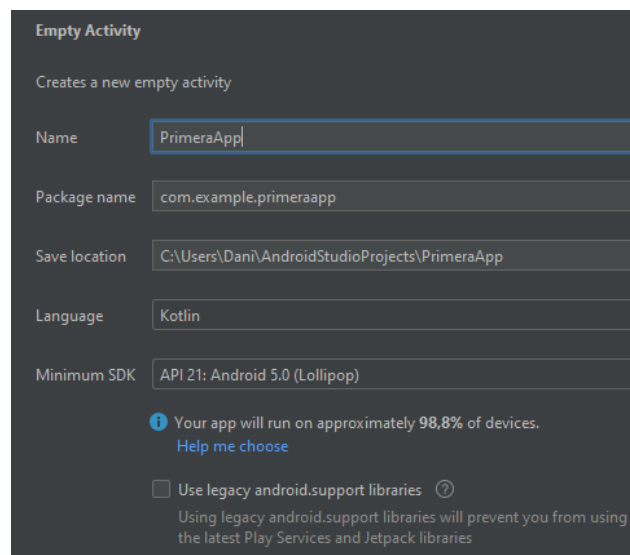
Una vez hayáis instalado Android Studio, procederemos a iniciar un nuevo proyecto, para ello, nos dirigiremos a su correspondiente botón o link.



Al ser las primeras veces que iniciamos android, es recomendable que creéis un proyecto con una actividad vacía (empty):



Y por último seleccionar el nombre de la app, la ubicación de los archivos en vuestro PC, el lenguaje de programación a utilizar, en este caso Kotlin y la mínima SDK que será compatible con tu aplicación.



La primera vez que creáis un proyecto tardará un poco, el resto irán mucho más rápido.

A partir de aquí, nos vamos a centrar en los elementos del proyecto que dependen del diseño de interfaces, dejando la parte de la programación para el módulo correspondiente.

La carpeta **res** se genera de manera automática y contiene todos los recursos que se van a utilizar en el proyecto de la aplicación (iconos, sonido, diseños, etc.). Los recursos por defecto que nos da la aplicación son 4:

4.1. Uso de los colores

El uso de los colores dentro de Android Studio sigue los pasos de la explicación sobre los códigos de colores en hexadecimal y decimal del principio del tema.

Cuando se crea un nuevo proyecto se asignan colores por defecto, estos colores se pueden encontrar en la carpeta **res/values/color**.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>

    <color name="gray">#E5E5E5</color>
    <color name="midGray">#8A8A8A</color>
    <color name="darkGray">#303030</color>
    <color name="blue">#007EFF</color>
</resources>
```

Los colores nuevos que se inserten en el proyecto los añadiremos en este archivo. Para definir un nuevo proyecto seguiremos la estructura del XML:

<COLOR name="nombreDelColor">#CODIGO_COLOR</COLOR>

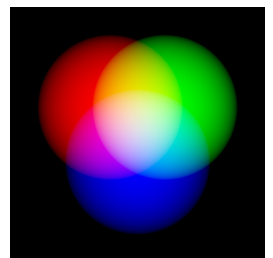
Donde **nombreDelColor**, es el nombre que se le quiera asignar y que gracias a ese nombre, podrá ser usado en cualquier parte del proyecto mediante esa referencia. El **#CODIGO_COLOR** es el código numeral del color asignado.

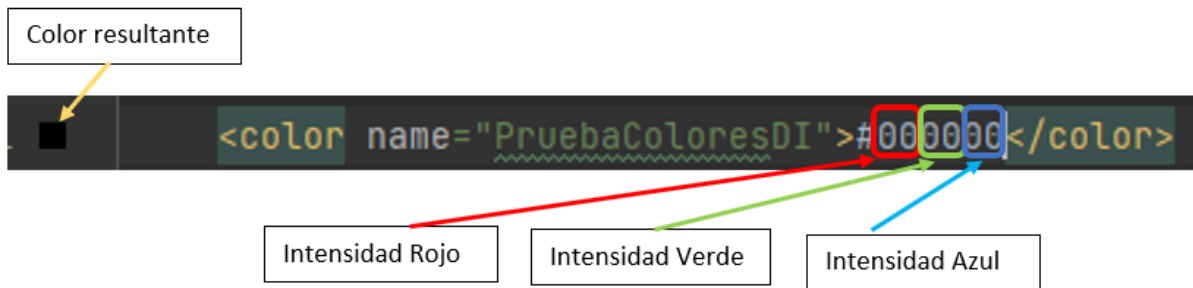
El color es un resultado de la mezcla de tres colores primarios, por tanto, los códigos que se representan mediante la combinación de RGB. Hay que tener en cuenta que

Actividad 1.3: Añade los siguientes colores al proyecto nuevo: magenta, white, purple, black y coral. Estos colores podrías utilizarlos en el futuro.

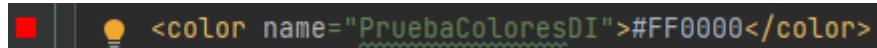
Una vez realizada la actividad anterior es fácil ver que los colores se manejan como si fueran 3 luces independientes, las cuales modificando la intensidad de cada color (RGB) la fusión resultante de los 3 dará un nuevo color.

Podemos expresar el máximo de cada color como FF y el mínimo como 00. por ejemplo:

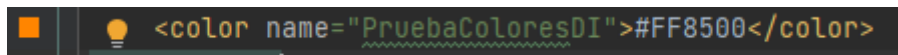




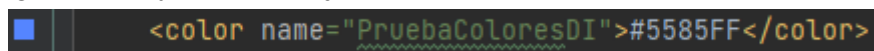
Como se puede apreciar, tenemos la secuencia RGB con toda intensidad de los 3 colores a 0, por tanto el resultado que da es 0. Poniendo al máximo el rojo:



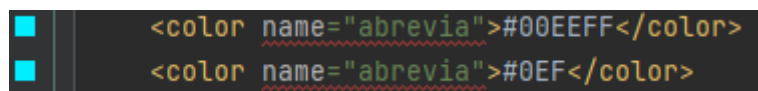
Añadiendo luz verde:



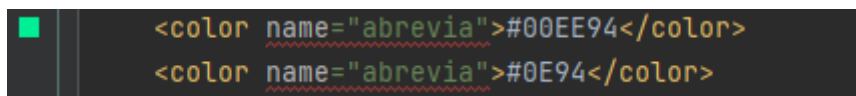
Añadiendo algo de azul y quitando rojo:



Es posible realizar una abreviación en el código hexadecimal de los colores, siempre que cada pareja tenga la misma letra:



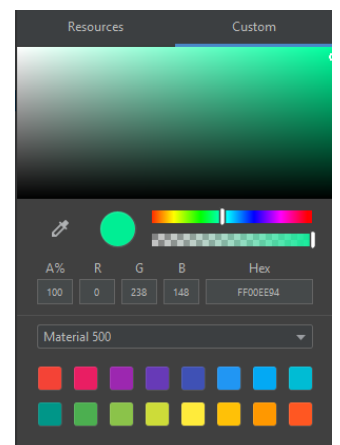
Pero siempre deben repetirse los 3 pares, no es posible esto:



Una de las partes más interesantes del color en android studio es que si clicamos en el color resultante nos aparecerá la paleta de colores, por lo que podemos seleccionarlos de manera visual.

El último punto a tratar sobre el color es la pareja de cifras hexadecimales que tienen al principio todos los colores por defecto y que no hemos añadido en los ejemplos anteriores.

Los dos primeros dígitos hacen referencia al grado de transparencia, también llamado canal Alfa. Siendo el valor máximo (FF) nada transparente y 00 totalmente transparente.



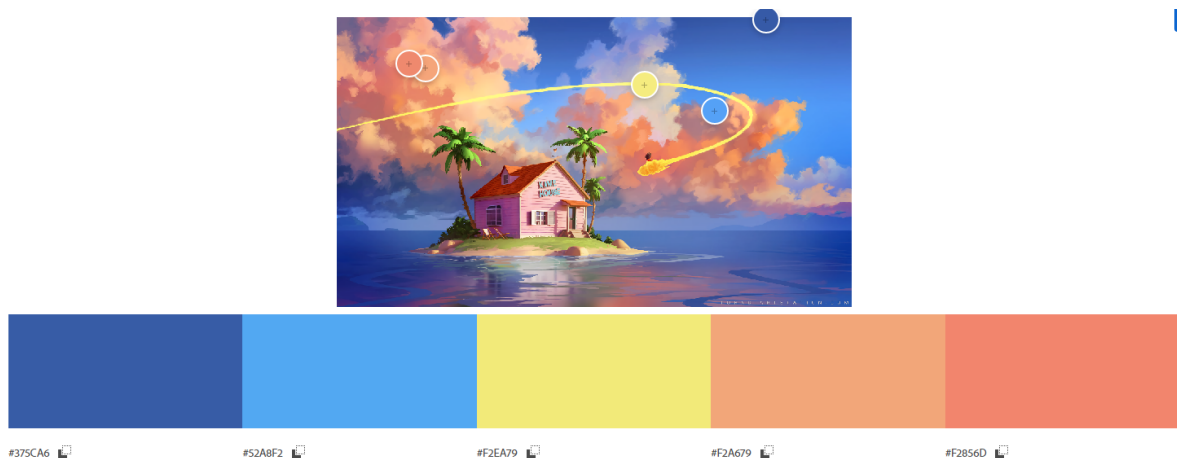
4.2. Selección de colores en las APP

Como se ha visto anteriormente, la elección de los colores no es trivial y puede ayudar a una APP sea más utilizada. Existen herramientas en Internet que nos ayudan a realizar una buena selección de color. Algunas de ellas:

- Adobe colors: <https://color.adobe.com/es/create/color-wheel>
- Colors es una web que te permite generar esquemas de colores compatibles.
 - <https://coolors.co/>
- Page flows: Web con múltiples patrones de diseño de interfaces para aplicaciones tanto web como para móviles
 - <https://pageflows.com/>
- Mobbin: Web con patrones de diseño de interfaces para web y móviles.
 - <https://mobbin.com/browse/ios/apps>

Actividad 1.4: Utiliza la herramienta de adobe colors para obtener los colores principales de una imagen que selecciones tú mismo y que te gustaría obtener los colores para obtener un tema en tu APP. Una vez hecho esto, añade los colores a tu proyecto.

Realiza esta actividad con al menos 3 imágenes. Se deben subir a la actividad de aules un documento con una captura de pantalla dónde se vea la imagen y los colores obtenidos y otra captura con los colores en el proyecto. Ejemplo de imagen:



5. Bibliografía y webgrafía

- García-Miguel, D; García-Miguel, B; Fernández, A. Diseño de interfaces. Editorial síntesi. 2021
- [Disseny d'interfícies al IOC.](#)
- [Programación de Aplicaciones para Android con Kotlin 2022. Plataforma Udemy.](#)