

Control de documento

Nombre del proyecto	Healthec
Cierre de iteración	I1 24/02/2023
Generador por	Carlos Antonio Madrigal Trejo
Aprobado por	Carlos Antonio Madrigal Trejo
Alcance de la distribución del documento	Control interno para todo el proyecto.

## Índice

<b>Sobre este documento</b>	<b>3</b>
<b>Resumen de la Iteración</b>	<b>4</b>
Identificación	4
Artefactos y evaluación	4
Riesgos y problemas (Riesgo con su identificador si sucedió o no sucedió, en este sprint no se aplica)	5
<b>Asignación de recursos</b>	<b>5</b>
<b>Anexos</b>	<b>7</b>
Anexo A.	7
Anexo B	8
Anexo C.	10
Anexo D.	13
Anexo E.	14
Anexo F.	16
Anexo G.	17
Anexo H.	23
Anexo I.	25
Anexo J.	26
<b>Glosario de términos</b>	<b>27</b>

### **Sobre este documento**

La calidad se logra por medio de la revisión constante de las actividades que conducen desde la idea al producto. Al momento del cierre de una iteración es buen momento para hacer un alto, y evaluar lo logrado, los problemas encontrados y los retos a enfrentar.

El presente documento marca el final de la iteración I 1, y contiene una evaluación de los artefactos y actividades realizadas durante la misma.

Se recogen también las impresiones y observaciones hechas durante el desarrollo de la iteración, así como el esfuerzo invertido en cada una de las disciplinas involucradas.

## Resumen de la Iteración

### Identificación

Código de la iteración	Fase a la que pertenece	Fecha de inicio	Fecha de cierre	Comentarios
I1	Inicio (Ing. Requisitos)	20/02/23	24/02/23	Se completó correctamente.

### Artefactos y evaluación

Artefacto	Meta (%)	Comentarios
INV-01	100%	Se logró la investigación de “Investigar aplicaciones similares” efectivamente.
INV-02	100%	Se logró la investigación de “Investigar cómo crear diseños” efectivamente.
INV-03	100%	Se logró la investigación de “Investigar cómo utilizar sensores” efectivamente.
INV-04	100%	Se logró la investigación de “Investigar procesos en Android” efectivamente.
INV-05	100%	Se logró la investigación de “Investigar ejercicios de relajación” efectivamente.
INV-06	100%	Se logró la investigación de “Investigar fuentes de información de un sueño saludable” efectivamente.
INV-07	100%	Se logró la investigación de “Implementar bases de datos en apps móviles” efectivamente.
INV-08	100%	Se logró la investigación de “Investigar la implementación de un login en la app” efectivamente.
INV-09	100%	Se logró la investigación de “Investigar cómo acceder a la actividad de una tabla” efectivamente.

INV-10	100%	Se logró explorar Android Studio para conocer sus herramientas efectivamente.
--------	------	-------------------------------------------------------------------------------

Artefacto	Aspecto a evaluar	Evaluación	Comentarios
INV-01	Tabla comparativa	100%	Se cumplió en su totalidad.
INV-02	Tabla comparativa	100%	Se cumplió en su totalidad
INV-03	Tabla comparativa	100%	Se cumplió en su totalidad.
INV-04	Tabla comparativa	100%	Se cumplió en su totalidad
INV-05	Tabla comparativa	100%	Se cumplió en su totalidad
INV-06	Tabla comparativa	100%	Se cumplió en su totalidad.
INV-07	Tabla comparativa	100%	Se cumplió en su totalidad.
INV-08	Tabla comparativa	100%	Se cumplió en su totalidad
INV-09	Tabla comparativa	100%	Se cumplió en su totalidad.
INV-10	Tabla comparativa	100%	Se cumplió en su totalidad.

**Riesgos y problemas (Riesgo con su identificador si sucedió o no sucedió, en este sprint no se aplica)**

#### Asignación de recursos

Rol	Horas-Hombre	Desempeñado por( área)	Observaciones
Investigador	2.5 horas.	Carlos Antonio Madrigal Trejo	Hizo bien su trabajo.

Investigador	2.5 horas.	Fernando Pérez Romero	Desarrolló bastante bien su documentación.
Investigador	2 horas.	Omar Adrián Tapia Guzmán	Realizó un buen trabajo.
Investigador	2.5 horas.	Rubén Dario Vidaña	Realizó una buena documentación.
Investigador	2 horas.	Carlos Daniel López Romo	Hizo bien su trabajo.

## Anexos

### Anexo A.

#### INV-01 Investigar aplicaciones similares

Características	Precio	Ejercicios de respiración	Rastreo del sueño	Diario	Recordatorio de citas médicas	Notificaciones de motivación	Información saludable	Podcasts	Recetas diarias	Rastreo del humor
<b>Healthec</b>	Gratis									
<b>Headspace</b>	\$100/mes									
<b>IBreath</b>	Gratis									
<b>Moodkit</b>	\$150									

Actualmente en el mercado existen muchas opciones viables que se asemejan a la aplicación a desarrollar, sin embargo al comparar se puede observar herramientas que podrían marcar la diferencia entre las demás, es por eso que en base al análisis podemos enfocarnos en darle un mejor desarrollo a aquellas que son únicas en el mercado y las que son semejantes darles un giro nuevo para hacerla más atractiva al usuario final.

## Anexo B

### INV-02 Investigar cómo crear diseños

#### TABLA

Número de paso	Instrucción
Paso 1	Abre el archivo "activity_main.xml" en el editor visual de diseño.
Paso 2	Selecciona el dispositivo virtual para el que deseas diseñar tu aplicación.
Paso 3	Selecciona los elementos de diseño que desees agregar desde la barra de herramientas de diseño, como botones, cuadros de texto, imágenes, etc.
Paso 4	Arrastra y suelta los elementos de diseño en el área de diseño.
Paso 5	Configura las propiedades de los elementos de diseño, como el tamaño, la posición, el texto y el color, desde la ventana Propiedades.
Paso 6	Utiliza las guías y los ajustes de diseño para asegurarte de que tu diseño se ajuste a las directrices de diseño de Android y sea coherente con la apariencia de otras aplicaciones.
Paso 7	Prueba tu diseño en diferentes tamaños de pantalla para asegurarte de que se vea bien en diferentes dispositivos.
Paso 8	Agrega lógica y funcionalidad a tu diseño utilizando Java o Kotlin, como eventos de clic, animaciones y transiciones.

En conclusión, crear diseños en Android Studio es un proceso sencillo pero que requiere atención y cuidado en los detalles para lograr una interfaz de usuario atractiva y funcional. La plataforma ofrece diversas herramientas para diseñar interfaces gráficas de manera fácil y efectiva.

Para crear diseños en Android Studio se pueden utilizar dos enfoques: el editor visual de diseño y el lenguaje de marcado XML. El editor visual de diseño es la opción más recomendada para principiantes, ya que permite arrastrar y soltar elementos de la interfaz de usuario para crear un diseño. En cambio, el lenguaje de marcado XML proporciona un control más preciso sobre el diseño, pero puede requerir más conocimientos técnicos.





Crear diseños en Android Studio requiere una buena planificación y atención a los detalles, pero puede lograrse fácilmente utilizando las herramientas y directrices proporcionadas por la plataforma.

## Anexo C.

### INV-03 Investigar cómo utilizar sensores

Sensor	Datos de eventos de los sensores	Descripción	Unidades de medida
<a href="#">TYPE_ACCELEROMETER</a>	SensorEvent.values[0]	Indica la fuerza de aceleración en el eje x (incluida la gravedad).	m/s <sup>2</sup>
	SensorEvent.values[1]	Indica la fuerza de aceleración en el eje Y (incluida la gravedad).	
	SensorEvent.values[2]	Indica la fuerza de aceleración en el eje z (incluida la gravedad).	
<a href="#">TYPE_ACCELEROMETER_UNCALIBRATED</a>	SensorEvent.values[0]	Indica la aceleración medida en el eje X sin compensación de sesgo.	m/s <sup>2</sup>
	SensorEvent.values[1]	Indica la aceleración medida en el eje Y sin compensación de sesgo.	
	SensorEvent.values[2]	Indica la aceleración medida en el eje Z sin compensación de sesgo.	
	SensorEvent.values[3]	Indica la aceleración medida en el eje X con compensación de sesgo estimada.	
	SensorEvent.values[4]	Indica la aceleración medida en el eje Y con compensación de sesgo estimada.	
	SensorEvent.values[5]	Indica la aceleración medida en el eje Z con compensación de sesgo estimada.	
<a href="#">TYPE_GRAVITY</a>	SensorEvent.values[0]	Indica la fuerza de gravedad en el eje X.	m/s <sup>2</sup>

	SensorEvent.values[1]	Indica la fuerza de gravedad en el eje y.	
	SensorEvent.values[2]	Indica la fuerza de gravedad en el eje z.	
<a href="#">TYPE_GYROSCOPE</a>	SensorEvent.values[0]	Indica la rotación alrededor del eje x.	rad/s
	SensorEvent.values[1]	Indica la rotación alrededor del eje y.	
	SensorEvent.values[2]	Indica la rotación alrededor del eje z.	
<a href="#">TYPE_GYROSCOPE_UNCALIBRATED</a>	SensorEvent.values[0]	Indica la velocidad de rotación (sin compensación de variación) alrededor del eje x.	rad/s
	SensorEvent.values[1]	Indica la velocidad de rotación (sin compensación de variación) alrededor del eje y.	
	SensorEvent.values[2]	Indica la velocidad de rotación (sin compensación de variación) alrededor del eje z.	
	SensorEvent.values[3]	Indica la variación estimada alrededor del eje x.	
	SensorEvent.values[4]	Indica la variación estimada alrededor del eje y.	
	SensorEvent.values[5]	Indica la variación estimada alrededor del eje z.	
<a href="#">TYPE_LINEAR_ACCELERATION</a>	SensorEvent.values[0]	Indica la fuerza de aceleración en el eje x (sin incluir la gravedad).	m/s <sup>2</sup>
	SensorEvent.values[1]	Indica la fuerza de aceleración en el eje y (sin incluir la gravedad).	
	SensorEvent.values[2]	Indica la fuerza de aceleración en el eje z (sin incluir la gravedad).	

<a href="#">TYPE_ROTATION_VECTOR</a>	SensorEvent.values[0]	Indica el componente vectorial de rotación en el eje x ( $x * \sin(\theta/2)$ ).	Sin unidades
	SensorEvent.values[1]	Indica el componente vectorial de rotación en el eje y ( $y * \sin(\theta/2)$ ).	
	SensorEvent.values[2]	Indica el componente vectorial de rotación en el eje z ( $z * \sin(\theta/2)$ ).	
	SensorEvent.values[3]	Indica el componente escalar del vector de rotación ( $(\cos(\theta/2)).1$	
<a href="#">TYPE_SIGNIFICANT_MOTION</a>	N/D	N/D	N/D
<a href="#">TYPE_STEP_COUNTER</a>	SensorEvent.values[0]	Indica la cantidad de pasos que dio el usuario desde el último reinicio mientras estaba activado el sensor.	Pasos
<a href="#">TYPE_STEP_DETECTOR</a>	N/D	N/D	N/D

### Conclusiones

La plataforma de Android Studio nos provee de diferentes sensores que podrían ser útiles en el desarrollo de las herramientas que aplicaremos, también pudimos diferenciar entre las aplicaciones de cada uno y así identificar el que sería más útil para nuestra aplicación.

## Anexo D.

### INV-04 Investigar procesos en Android

Tipo	Descripción
onCreate()	Se llama a este método cuando se crea la actividad. Aquí se realiza la inicialización de la actividad, como la creación de la interfaz de usuario (UI) y la asignación de valores a las variables.
onStart()	Se llama a este método cuando la actividad se hace visible para el usuario. En este punto, la actividad está en primer plano y el usuario puede interactuar con ella.
onResume()	Se llama a este método cuando la actividad está en primer plano y el usuario puede interactuar con ella. Este método se utiliza para reanudar cualquier operación que se haya suspendido durante el ciclo de vida de la actividad.
onPause()	Se llama a este método cuando la actividad pierde el enfoque pero aún es visible para el usuario. En este punto, la actividad está en segundo plano y puede ser interrumpida por otra actividad.
onStop()	Se llama a este método cuando la actividad ya no es visible para el usuario. En este punto, la actividad se encuentra en segundo plano y está en uso.
onDestroy()	Se llama a este método cuando se destruye la actividad. Este método se utiliza para liberar cualquier recurso que haya sido asignado a la actividad, como memoria o archivos.
onRestart()	Se llama a este método cuando se inicia una actividad después de haber estado detenida.
onSaveInstanceState()	Se llama a este método cuando la actividad está a punto de ser destruida, para que se pueda guardar los datos de la actividad.

Estos métodos nos ayudaron a entender cómo funciona un proceso en android y como usarlos a mi favor.

## Anexo E.

INV-05 Investigar ejercicios de relajación

TABLA

Ejercicio	Descripción	Duración
Respiración abdominal	Siéntate con la espalda recta y las manos en el abdomen. Inhala profundamente por la nariz y siente cómo se expande tu vientre. Exhala lentamente por la boca y siente cómo se contrae tu vientre. Repite durante 5-10 minutos.	5-10 minutos
Estiramiento de cuello	Siéntate con la espalda recta y los pies apoyados en el suelo. Gira lentamente la cabeza hacia un lado y sostén durante 5-10 segundos. Vuelve al centro y repite hacia el otro lado. Realiza 5 repeticiones en cada dirección.	1 minuto
Tensión-relajación muscular progresiva	Siéntate o acuéstate en una posición cómoda. Comienza tensando los músculos de los pies y sostén durante 5 segundos. Luego relaja los músculos y siente la diferencia de tensión. Repite este proceso con diferentes grupos musculares del cuerpo, trabajando hacia arriba hasta llegar a la cabeza.	10-15 minutos

Tensión-relajación muscular progresiva	Siéntate o acuéstate en una posición cómoda. Comienza tensando los músculos de los pies y sostén durante 5 segundos. Luego relaja los músculos y siente la diferencia de tensión. Repite este proceso con diferentes grupos musculares del cuerpo, trabajando hacia arriba hasta llegar a la cabeza.	10-15 minutos
Visualización guiada	Cierra los ojos y visualiza un lugar tranquilo y relajante, como una playa o un bosque. Imagina que estás allí y toma nota de los sonidos, los olores y las sensaciones. Permanece en este lugar durante 5-10 minutos.	5-10 minutos
Yoga	Realiza una secuencia de posturas de yoga suaves y fluidas, como la postura del niño o la postura del perro boca abajo. Sigue una guía de yoga en línea o una aplicación para asegurarte de que las posturas se realicen correctamente.	10-15 minutos

Podemos concluir que para que la app consiga lo que se puso como meta, el usuario final deberá estar comprometido a realizar los ejercicios ya que estos requerirán tiempo para poder lograr su cometido, se buscaron las maneras de relajación más usadas y funcionales, pero aun así se pueden agregar más.

## Anexo F.

### INV-06 Investigar fuentes de información de un sueño saludable

Sueño saludable	Información respaldada	Actualiza información de manera constante	Es página web	Autor	Link
Guía para sueño saludable	si	si	si	National heart and blood institute	<a href="https://www.nhlbi.nih.gov/files/docs/public/sleep/In_Brief_YG_to_Sleep_Spanish_Final.pdf">https://www.nhlbi.nih.gov/files/docs/public/sleep/In_Brief_YG_to_Sleep_Spanish_Final.pdf</a>
Salud del sueño	si	no	si	Doctora Alicia López de Ocáriz	<a href="https://cinfasalud.cinfa.com/p/salud-del-sueno/">https://cinfasalud.cinfa.com/p/salud-del-sueno/</a>
Dormir bien	si	no	si	Centros para el Control y la Prevención de Enfermedades	<a href="https://www.cdc.gov/spanish/cancer/survivors/healthy-living-guides/physical-health/sleep.htm">https://www.cdc.gov/spanish/cancer/survivors/healthy-living-guides/physical-health/sleep.htm</a>
El sueño y su salud	si	si	si	Linda J. Vorvick	<a href="https://medlineplus.gov/spanish/ency/patientinstructions/000871.htm">https://medlineplus.gov/spanish/ency/patientinstructions/000871.htm</a>
Sueño saludable, soluciones para dormir bien	si	no	no	Doctor William Douglas	<a href="https://books.google.com.mx/books?id=bDs0EAAAQBAJ&amp;pg=PT8&amp;dq=sue%C3%B1o+saludable&amp;hl=es&amp;sa=X&amp;ved=2ahUKEwj6340a79AhVpEUQIHbkYBHM6AF6BAGHEAl#v=onepage&amp;q=sue%C3%B1o%20saludable&amp;f=false">https://books.google.com.mx/books?id=bDs0EAAAQBAJ&amp;pg=PT8&amp;dq=sue%C3%B1o+saludable&amp;hl=es&amp;sa=X&amp;ved=2ahUKEwj6340a79AhVpEUQIHbkYBHM6AF6BAGHEAl#v=onepage&amp;q=sue%C3%B1o%20saludable&amp;f=false</a>

Las diferentes fuentes consultadas nos muestran distinta información acerca del sueño saludable, alguna de estas serán de utilidad para el desarrollo de la aplicación.



## Anexo G.

### INV-07 Implementar bases de datos en apps móviles

SQLiteOpenHelper (hereda de Object).

Constructores Públicos	
Métodos	Descripción
SQLiteOpenHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version)	Crear un objeto de ayuda para crear, abrir y / o administrar una base de datos.
SQLiteOpenHelper(Context, String name, SQLiteDatabase.CursorFactory factory, int version, DatabaseErrorHandler errorHandler)	Crear un objeto de ayuda para crear, abrir y / o administrar una base de datos.

MÉTODOS PÚBLICOS	
VALOR QUE DEVUELVE	MÉTODOS
synchronized void	close()
	Cierre cualquier objeto de base de datos abierta.
String	getDatabaseName()
	Devuelva el nombre de la base de datos SQLite que se abre.

SQLiteDatabase	getReadableDatabase()
	Crear y / o abrir una base de datos.
SQLiteDatabase	getWritableDatabase()
	Crear y / o abrir una base de datos que se utilizará para lectura y escritura.
void	onConfigure(SQLiteDatabase db)
	Se llama cuando se está configurando la conexión de base de datos, para permitir las características tales como el registro de escritura anticipada o soporte de clave externa.
abstract void	onCreate(SQLiteDatabase db)
	Se llama cuando se crea la base de datos por primera vez.
void	onDowngrade(SQLiteDatabase db, int oldVersion, int newVersion)
	Se llama cuando la base de datos necesita ser degradada.

void	onOpen(SQLiteDatabase db)
	Se llama cuando la base de datos ha sido abierta.
abstract void	onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
	Se llama cuando la base de datos necesita ser actualizada.
void	setWriteAheadLoggingEnabled(boolean enabled)
	Activa o desactiva el uso del registro de escritura anticipada para la base de datos.

SQLiteDatabase (hereda de **SQLiteClosable**): contiene los métodos para crear, eliminar, ejecutar comandos SQL y realizar otras tareas de administración de base de datos común.

No define constructores públicos.

MÉTODOS PÚBLICOS	
VALOR QUE DEVUELVE	MÉTODOS
int	delete(String table, String whereClause, String[] whereArgs)
	Método de conveniencia para eliminar filas en la base de datos.

<b>static SQLiteDatabase</b>	<b>create(SQLiteDatabase.CursorFactory factory)</b>
	Crea una base de datos SQLite respaldada en memoria.
<b>void</b>	<b>execSQL(String sql)</b>
	Ejecuta una sola sentencia SQL que no es una sentencia SELECT o cualquier otra sentencia SQL que devuelva datos.
<b>void</b>	<b>setVersion(int version)</b>
	Establece la versión de base de datos
<b>int</b>	<b>getVersion()</b>
	Obtiene la versión de base de datos.
<b>long</b>	<b>insert(String table, String nullColumnHack, ContentValues values)</b>
	Método de conveniencia para insertar una fila en la base de datos.
<b>boolean</b>	<b>isOpen()</b>

	Devuelve verdadero si la base de datos está actualmente abierta.
<b>boolean</b>	<b>isReadOnly()</b>
	Devuelve verdadero si se abre la base de datos como sólo lectura.
<b>int</b>	<b>update(String table, ContentValues values, String whereClause, String[] whereArgs)</b>
	Método de conveniencia para la actualización de las filas de la base de datos.
<b>Cursor</b>	<b>query(String table, String[] columns, String selection, String[] selectionArgs, String groupBy, String having, String orderBy, String limit)</b>
	Consulta la tabla dada, devolviendo un cursor sobre el conjunto de resultados.
<b>void</b>	<b>setTransactionSuccessful()</b>
	Marca la transacción actual como un éxito.

SQLiteQuery (hereda de **SQLiteProgram**): Representa una consulta que lee las filas resultantes dentro de SQLiteQuery.

Esta clase no es segura para subprocesos. No define constructores públicos.

MÉTODOS PÚBLICOS	
VALOR QUE DEVUELVE	MÉTODOS
toString()	toString()
	Devuelve una cadena que contiene una descripción concisa y legible.

Conclusión: Esta package **android.database.sqlite** con sus clases y métodos es el que nos ayudará a conectar una base de datos a la app, y esta puede funcionar correctamente con la aplicación de android studio donde es donde llevaremos a cabo el proyecto.

## Anexo H.

### INV-08 Investigar la implementación de un login en la app

Número de paso	Instrucción
Paso 1	Crear un nuevo proyecto de Android Studio.
Paso 2	En el archivo de diseño "activity_main.xml", añadir los elementos necesarios para el formulario de login, como dos EditText para el correo electrónico y la contraseña, y un botón para enviar la información.
Paso 3	Crear una nueva actividad para el dashboard o pantalla principal de la aplicación, en la que se mostrarán los datos y funcionalidades de la misma.
Paso 4	Crear una clase Java que extienda de AppCompatActivity y que se encargue de gestionar el proceso de login, llamada "LoginActivity".
Paso 5	En el método onCreate() de LoginActivity, añadir el código necesario para referenciar los elementos del formulario de login, y para definir un Listener para el botón de enviar la información.
Paso 6	En el Listener del botón de enviar la información, añadir el código necesario para validar los datos introducidos por el usuario, y para comprobar si el correo electrónico y la contraseña son correctos. Si son correctos, se debe lanzar la actividad del dashboard o pantalla principal, y si no, mostrar un mensaje de error.
Paso 7	Guardar la información del usuario en la sesión, utilizando SharedPreferences o alguna librería de manejo de sesiones como JWT.
Paso 8	Añadir el código necesario en la actividad del dashboard o pantalla principal, para recuperar la información del usuario de la sesión y mostrarla en la interfaz de usuario.

## Conclusiones

En conclusión, implementar un login en una aplicación de Android Studio es un proceso que implica varios pasos importantes, como la creación de la interfaz de usuario, la validación de

los datos de entrada, el almacenamiento seguro de las credenciales y la gestión de la sesión de usuario.

Es importante asegurarse de que la implementación del login sea segura, utilizando medidas adicionales como el uso de HTTPS, el almacenamiento seguro de las contraseñas utilizando funciones hash y salting, y la implementación de medidas de protección contra ataques de fuerza bruta y otros tipos de ataques.



## Anexo I.

INV-09 Investigar cómo acceder a la actividad de una tabla.

Tipos	Métodos	Descripción
static Clock	currentGnssTimeClock()	Devuelve un clock que comienza el 1 de enero de 1970 00:00:00.0 UTC, sincronizado mediante el proveedor de ubicación del dispositivo.
static Clock	currentNetworkTimeClock()	Devuelve un clock que comienza el 1 de enero de 1970 00:00:00.0 UTC, sincronizado mediante una fuente de red remota fuera del dispositivo.
static long	currentThreadTimeMillis()	Devuelve los milisegundos que se ejecutan en el subproceso actual.
static long	elapsedRealTime()	Devuelve milisegundos desde el arranque, incluido el tiempo de actividad
static boolean	setCurrentTimeMillis(long millis)	Establece un número dado de milisegundos (de uptimeMillis) antes de regresar.
static void	sleep(long ms)	Espera un número dado de milisegundos (de uptimeMillis) antes de regresar.
static long	uptimeMillis()	Devuelve milisegundos desde el inicio. sin contar el tiempo pasado en suspensión profunda.

Estos métodos nos ayudarán a medir el tiempo que el usuario pasa activo en su celular android lo cual resultará muy útil para lo que se pretende hacer.

## Anexo J.

### INV-10 Explorar Android Studio para conocer sus herramientas

Herramientas de Android Studio	Uso en proyecto	Utilidad (Alta, Media, Baja)	Descripción
Project	Si	Alta	Muestra los archivos del proyecto
Resource manager	Si	Alta	Muestra los recursos en uso
Structure	Si	Alta	Despliega la estructura del proyecto
Bookmarks	No	Baja	Anotaciones acerca del proyecto
Build variants	Si	Media	Controlador de versiones en la construcción del proyecto
Notifications	Si	Media	Notificaciones del proyecto
Device manager	Si	Media	Dispositivos físicos y virtuales enlazados
Gradle	Si	Alta	Constructor del proyecto
Emulator	Si	Alta	Dispositivos para emular
Device file explorer	Si	Media	Acceso al almacenamiento del dispositivo emulado o físicamente conectado
Debug	Si	Alta	Herramienta de depuración

En conclusión el IDE de Android Studio cuenta con múltiples herramientas, que usaremos en nuestro proyecto.

## Glosario de términos

**Interfaz:** En el contexto de la programación, una interfaz es un conjunto de métodos y propiedades que se pueden implementar en una clase para permitir la comunicación con otras clases y componentes. Una interfaz define un contrato que debe cumplirse para que las clases puedan comunicarse entre sí de manera efectiva.

**Hash:** En el contexto de la seguridad informática, un hash es un valor numérico que se genera a partir de un conjunto de datos utilizando un algoritmo de hash. Los algoritmos de hash son funciones criptográficas que toman una entrada de datos y producen una salida de longitud fija que es única para esa entrada. Los hashes se utilizan para verificar la integridad de los datos, proteger las contraseñas y garantizar la autenticidad de los mensajes.

**Salting:** El salting (en español, "salteado") es una técnica de seguridad utilizada en la criptografía de contraseñas. Consiste en agregar una cadena aleatoria de datos a la contraseña antes de realizar el hash. El salting hace que las contraseñas sean más difíciles de romper mediante ataques de fuerza bruta o de diccionario, ya que el hash resultante será diferente para cada usuario. El proceso de salting y hashing juntos se conoce como "salting and hashing" o "hashing with salt".

**Métodos:** En programación, los métodos son bloques de código que realizan una tarea específica. Los métodos pueden recibir parámetros y devolver valores, lo que les permite interactuar con otros componentes del programa. Los métodos se utilizan para organizar y modularizar el código, lo que facilita el mantenimiento y la reutilización del mismo.

**Base de datos:** Una base de datos es un sistema de almacenamiento y recuperación de información. Las bases de datos se utilizan comúnmente en aplicaciones informáticas para almacenar información estructurada y permitir la recuperación y manipulación de los datos de manera eficiente.

**Sensor:** Un sensor es un componente de hardware que detecta y responde a cambios en su entorno. En el contexto de la programación móvil, los sensores se utilizan comúnmente en los dispositivos móviles para detectar la ubicación, el movimiento, la luz, la proximidad y otros aspectos del entorno del dispositivo. Los datos del sensor se pueden utilizar para crear experiencias interactivas en tiempo real y para mejorar la precisión de las aplicaciones.