



IES AUGUSTO GONZALES DE LINARES  
CFGS DAM2º

# Documentación e Investigación Sobre desarrollo de aplicaciones móviles

PROGRAMACIÓN MULTIMEDIA Y  
DISPOSITIVOS MÓVILES

Daniel Espinosa García  
2023/2024



## Índice

RA1.....	2
c) Se han identificado las tecnologías de desarrollo de aplicaciones para dispositivos móviles. .....	2
d) Se han instalado, configurado y utilizado entornos de trabajo para el desarrollo de aplicaciones para dispositivos móvil .....	3
e) Se han identificado configuraciones que clasifican los dispositivos móviles en base a sus características.....	3
g) Se ha analizado la estructura de aplicaciones existentes para dispositivos móviles identificando las clases utilizadas. ....	4
h) Se han realizado modificaciones sobre aplicaciones existentes.....	5
i) Se han utilizado emuladores para comprobar el funcionamiento de las aplicaciones.....	5
RA2.....	6
a) Se ha generado la estructura de clases necesaria para la aplicación. ....	6
b) Se han analizado y utilizado las clases que modelan ventanas, menús, alertas y controles para el desarrollo de aplicaciones gráficas sencillas. ....	7
c) Se han utilizado las clases necesarias para la conexión y comunicación con dispositivos inalámbricos. ....	7
d) Se han utilizado las clases necesarias para el intercambio de mensajes de texto y multimedia. ....	8
<b>Unity:</b> Para realizar las comunicaciones entre usuarios necesitaríamos tener un servidor que gestione las comunicaciones y utilizando las clases UnityEngine.Networking y UnityEngine.Networking.NetworkSystem hacer una conexión a un servidor, utilizando un servicio de cliente-servidor. ....	8
e) Se han utilizado las clases necesarias para establecer conexiones y comunicaciones HTTP y HTTPS. ....	8
f) Se han utilizado las clases necesarias para establecer conexiones con almacenes de datos garantizando la persistencia. ....	9
g) Se han realizado pruebas de interacción usuario-aplicación para optimizar las aplicaciones desarrolladas a partir de emuladores. ....	9
h) Se han empaquetado y desplegado las aplicaciones desarrolladas en dispositivos móviles reales. ....	10
i) Se han documentado los procesos necesarios para el desarrollo de las aplicaciones. ....	10
Tabla Ilustraciones.....	11
Bibliografía.....	12

## RA1

c) Se han identificado las tecnologías de desarrollo de aplicaciones para dispositivos móviles.

Como podemos observar a lo largo del tiempo han existido múltiples sistemas operativos para los cuales se desarrollaron diferentes tecnologías para desarrollar aplicaciones.

En la actualidad el mercado está dominado por los sistemas operativos de Android y iOS.

Estos dos sistemas operativos (SO) utilizan tecnologías distintas para la programación de aplicaciones en el lenguaje nativo de cada SO

**Swift:** Para desarrollo en iOS, se utiliza este lenguaje de programación ofrecido por Apple para el desarrollo de aplicaciones que funcionen en sus plataformas: iOS, macOS, tvOS, watchOS y iPadOS.

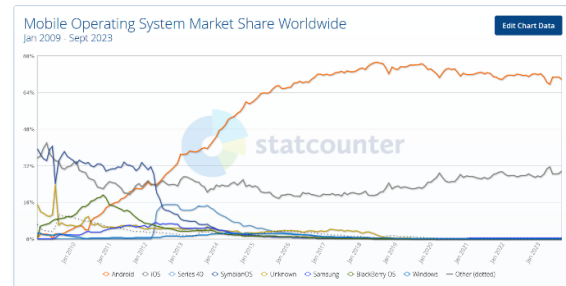


Ilustración 1 <https://gs.statcounter.com/os-market-share/mobile/worldwide>



Ilustración 2 Logo Swift



Ilustración 3 Logo Kotlin

**Kotlin:** Para desarrollo en Android, este lenguaje permite crear apps nativas de alta calidad y gran rendimiento en todas las plataformas soportadas por Android: smartphones, televisores, relojes, etc.

La principal desventaja de estos dos sistemas es que el desarrollo de aplicaciones en los lenguajes nativos no permite portar las aplicaciones entre sistemas operativos por lo cual se requiere duplicar el trabajo.

Para evitar esta duplicidad de trabajo se desarrollaron otras tecnologías para poder portar a ambos SO

**Ionic:** En primer lugar, tenemos ionic, el framework probablemente más popular que nos permite desarrollar aplicaciones multiplataforma. Nos ayuda a construir apps con HTML, CSS3 y Javascript empleando todo el potencial de las tecnologías de desarrollo web.

**Flutter:** Seguidamente, encontramos Flutter, una herramienta desarrollada por Google para el desarrollo de aplicaciones multiplataforma. Está basada en Dart y aunque se presentó en 2015, no se lanzó hasta 2018. Una gran ventaja frente a otras soluciones es que la aplicación desarrollada puede funcionar hasta en 4 plataformas: iOS, Android, Web y Desktop.

**React Native:** Es un framework creado por Facebook en 2015 para el desarrollo de apps multiplataforma.

Al igual que Ionic, también utiliza Javascript, pero lo hace bajo el pretexto de orquestar controles de interfaz de usuario (UI) específicos de la plataforma.

**Unity:** por último, pero no menos importante Unity permite también el desarrollo de aplicaciones para dispositivos móviles además de para PC.

d) Se han instalado, configurado y utilizado entornos de trabajo para el desarrollo de aplicaciones para dispositivos móviles.

Para desarrollar aplicaciones móviles instalamos Unity y Android Studio ambos IDE permiten la creación de apps.

Algunos de los IDEs más populares son:

**Android Studio:** es una popular herramienta de programación recomendada por Google. Viene con características de alto nivel para facilitar el trabajo de los desarrolladores en el desarrollo de aplicaciones Android. Una de las características esenciales son sus dispositivos virtuales.

**Xcode:** es un IDE de uso común para los sistemas operativos macOS. Utilizando este IDE, los desarrolladores pueden desarrollar software o aplicaciones en Mac que pueden utilizarse en iOS, iPadOS, macOS, tvOS y watchOS

**Unity:** es un motor de videojuego multiplataforma creado por Unity Technologies. Unity está disponible como plataforma de desarrollo para Microsoft Windows, Mac OS, Linux. La plataforma de desarrollo tiene soporte de compilación con diferentes tipos de plataformas entre ellas móviles.

Para realizar la instalación de Unity es necesario instalar primero Unity hub y luego realizar la instalación de Unity seleccionando la versión, en este caso estamos 2022.3.9f1. Desde Unity hub podemos luego instalar módulos.

e) Se han identificado configuraciones que clasifican los dispositivos móviles en base a sus características.

Se podrían clasificar en dos grandes grupos, aunque seguramente se podrían clasificar de muchas otras maneras. Estos dos grupos serían los siguientes:

- **Propósito general:** Incluye dispositivos como computadoras, tablets, relojes y teléfonos inteligentes. Se caracterizan por su capacidad para realizar una amplia gama de actividades y sintetizar funciones. Son el medio de trabajo para muchas personas, permitiendo realizar actividades que generan alta productividad.
  - Teléfonos inteligentes o Smartphone. Podemos decir que son como teléfonos móviles convencionales pero contruidos sobre una estructura informática móvil, que les proporciona mayor capacidad de almacenamiento y de procesamiento. Se asemejan bastante a un ordenador. Son usados a menudo tanto para el entretenimiento como herramientas de trabajo.

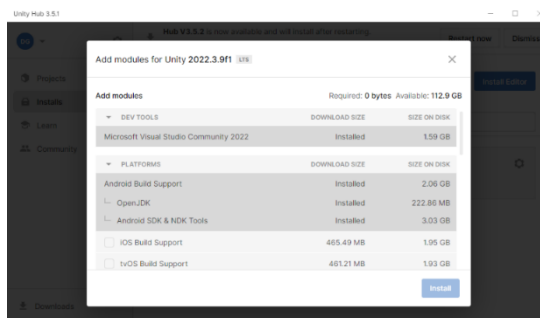


Ilustración 4 configuración de Unity Hub para permitir la exportación a SO Android

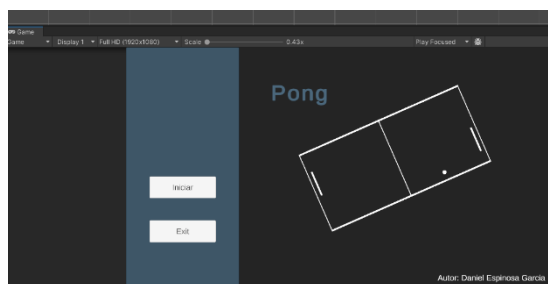


Ilustración 5 Imagen del desarrollo del Pong como primer proyecto de clases

- **Tabletas.** Son ordenadores portátiles personales integrados en una pantalla táctil, con acceso a Internet y capacidad para ejecutar aplicaciones instaladas sobre un sistema operativo. Su manejo es fácil e intuitivo.
  - **Relojes inteligentes.** Son relojes de pulsera que poseen funcionalidades como acceso a Internet, recibir llamadas o enviar mensajes.
  - **Ordenadores portátiles.** Son ordenadores completos que, debido a su tamaño reducido y compacto, permiten su transporte y gracias a su batería, su uso mientras no esté conectado el cargador a la red eléctrica.
- **Propósito de entretenimiento:** Este grupo incluye dispositivos diseñados para proporcionar diversión a las personas, como aquellos que permiten jugar videojuegos como propósito principal.
    - **Nintendo Switch:** es una consola de videojuegos desarrollada por Nintendo. Fue lanzada mundialmente el 3 de marzo de 2017.
    - **Steam Deck:** es un PC de juegos portátil desarrollado por Valve Corporación. Lanzado el 25 de febrero de 2022, Steam Deck se puede usar como PC portátil o de sobremesa usando un monitor externo. El dispositivo está diseñada para soportar gran parte de la biblioteca de Steam utilizando SteamOS, un sistema operativo que estriba en un kernel Debian1. Esto permite el soporte de un gran número de juegos de PC de Windows a través de la capa de compatibilidad Proton utilizando Linux. El sistema es una plataforma abierta, permitiendo a los usuarios instalar otros juegos compatibles con el dispositivo o incluso otros sistemas operativos.

g) Se ha analizado la estructura de aplicaciones existentes para dispositivos móviles identificando las clases utilizadas.

Para el desarrollo del pong utilizo 6 Scripts.

**Script GameManager** que es el que se encargara de gestionar todo el desarrollo del juego desde el inicio. Utilizando el resto de Los scripts para su funcionamiento.

Controla las puntuaciones de los goles. y el tiempo de espera para relanzar la bola después de que se marca el gol.

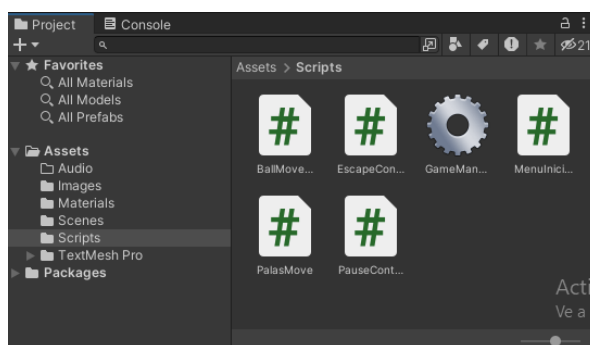


Ilustración 6 Scriptst utilizados en el Pong

**Script MenuInicio** para gestionar el menú de Inicio. El cual se encargará de llamar a la Escena del Juego y También permitirá salir / cerrar el juego.

El resto de Los scripts se encargan del funcionamiento interno del Juego.

**Script PauseControl** que se ocupa de detener y reanudar el juego utilizando la función Time.timeScale que detiene el paso de tiempo en el Juego deteniendo así la bola y las palas.

**Script EscapeControl** permite salir de la escena del juego a la escena del menú principal.

**Script PalasMove** Da funcionalidad a las palas para que se puedan mover cada una independientemente hacia arriba y abajo, teniendo un límite de posición.

**Script BallMovement** este además del GameManager es de los más complejos ya que controla los coliders con las palas aumentando la velocidad tanto de la bola como las palas con cada impacto a estas, limitando su velocidad. Esta monitorizando si el trigger de las porterías es activado para que el GameManager aumente el marcador y relance la bola desde el centro utilizando un lanzamiento aleatorio.

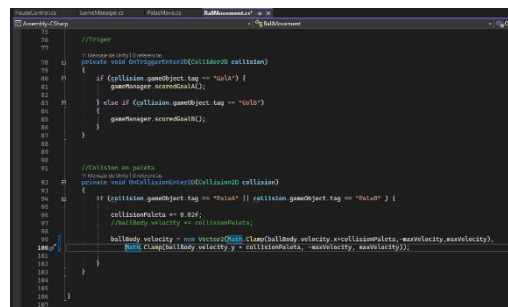


Ilustración 7 Script BallMovement

h) Se han realizado modificaciones sobre aplicaciones existentes.

Para realizar el Pong no me apoye en la documentación entregada en Moodle, lo realice con la ayuda de compañeros para entender el funcionamiento de Unity, como se conectan los Scripts con los objetos y como se controla el juego desde el GameManager.

Despues de revisar un poco la documentación entregada veo que tengo modificado las siguientes cosas:

- Una escena de menú de inicio que me parece que no está incluida en el documento y añadido el botón de Iniciar y el botón de Exit para salir del juego.
- La bola del juego parte con una velocidad inicial y va acelerando a medida que golpea con las palas, añadiendo un límite a la velocidad que puede alcanzar.

Tampoco invertí mucho más tiempo en este proyecto ya que decidimos entre varios compañeros realizar un proyecto más complejo y grande entre varios.



Ilustración 8 Script para controlar la aceleración de la bola.

i) Se han utilizado emuladores para comprobar el funcionamiento de las aplicaciones.

**Yuzu:** Este emulador de libre y de código abierto emula a la Nintendo Switch, fue lanzado en 2018, está escrito en C++ y puede ser ejecutado tanto en Windows como Linux. Al ser un software libre y de código abierto utiliza la licencia GPLv2. En un principio solo podía utilizar programas de prueba y homebrew. Pero con el tiempo ya permite correr sin problemas una gran cantidad de juegos.



**BlueStacks:** es un emulador de Android para Windows y Mac permitiendo replicar las características de un dispositivo móvil en un PC. Lo cual nos permite hacer pruebas de manera segura de software, apps, mucho más facil y segura que en un dispositivo real. Ademas permite simular distintas configuraciones de dispositivos.

**Unity:** Este software de desarrollo de aplicaciones trae de manera integrada un emulador de dispositivos móviles permitiendo ajustar las características de los mismos para comprobar la ejecución de las apps en ellos. Permitiendo de manera mucho más rápida y fácil probar las diferentes aplicaciones sin necesidad de salir del entorno.

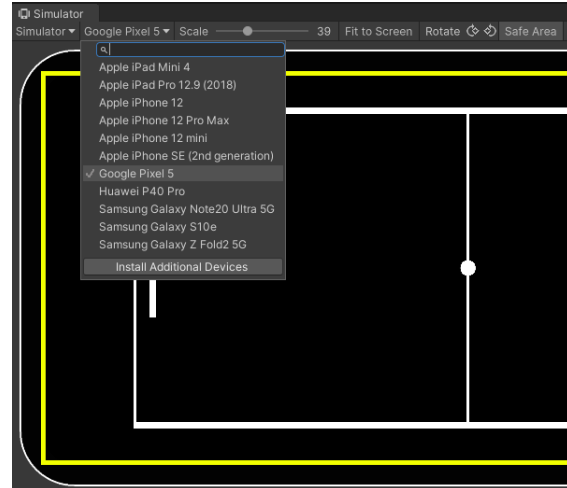


Ilustración 9 Unity Emulación de dispositivo móvil

## RA2

a) Se ha generado la estructura de clases necesaria para la aplicación.

En el siguiente esquema se muestra el funcionamiento de los Scripts del Juego Pong y como se relacionan entre ellos desde el Inicio del Programa hasta su cierre.

El funcionamiento de cada Script ya se realizó en un apartado anterior. RA1 punto g.

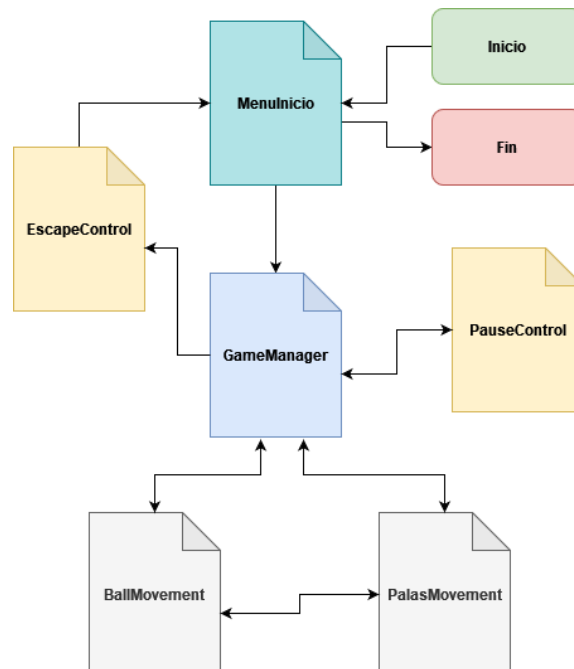


Ilustración 10 Diagrama de como utiliza el Juego Pong los Scripts en la ejecución de la aplicación

b) Se han analizado y utilizado las clases que modelan ventanas, menús, alertas y controles para el desarrollo de aplicaciones gráficas sencillas.

Para realizar el menú de inicio del juego incorpore otra escena en la cual se añade un canvas para añadir a este los Botones de Iniciar y Exit. Los cuales cambian de color al presionarlos.

Se añade dos cuadros de texto para incorporar el Título del Juego y el autor de este.

Este utiliza el Script MenuInicio para lanzar la escena del juego.

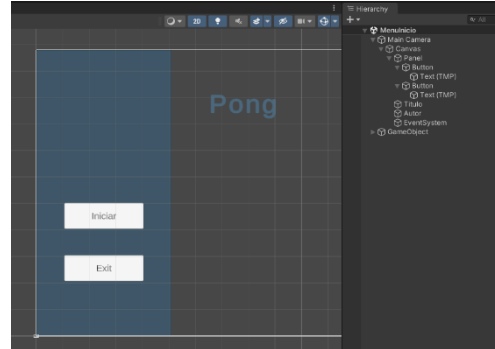


Ilustración 11 Canvas del Menú de Inicio del juego Pong

c) Se han utilizado las clases necesarias para la conexión y comunicación con dispositivos inalámbricos.

Se pueden realizar mediante Wifi o Bluetooth

**WIFI:** Para realizar conexiones con otro dispositivo con Android estudio es necesario utilizar la clase WifiP2pManager. Esta clase contiene metodos para establecer la conexión y permitir la comunicación entre los dos dispositivos, al tratarse de una conexión P2P.

Es necesario ademas añadir ciertos permisos a la aplicación para que la conexión funcione correctamente.

Tambien es posible configurar la búsqueda de servicios a los que se puede tener acceso.

En el siguiente link existe una guía de cómo realizarlo:

<https://developer.android.com/training/connect-devices-wirelessly?hl=es-419>

**BLUETOOTH:** De un modo parecido que con el wifi es necesario realizar una configuración previa de la aplicación para su configuración. Estableciendo permisos.

Las Clases para el funcionamiento de una conexión que usaremos serán bluetoothManager, bluetoothAdapter, BluetoothDevice, BluetoothSocket que permitirán mediante sus metodos y propiedades gestionar las conexiones, la creación de un canal que se utilizara para envío de información entre los dispositivos y la búsqueda de dispositivos.

En la Api de Android tenemos más información al respecto:

<https://developer.android.com/guide/topics/connectivity/bluetooth?hl=es-419>



d) Se han utilizado las clases necesarias para el intercambio de mensajes de texto y multimedia.

**Unity:** Para realizar las comunicaciones entre usuarios necesitaríamos tener un servidor que gestione las comunicaciones y utilizando las clases UnityEngine.Networking y UnityEngine.Networking.NetworkSystem hacer una conexión a un servidor, utilizando un servicio de cliente-servidor.

A partir de aquí tocaría crear un servicio en el server que se encargue de gestionar la comunicación entre los dispositivos. Para más información consultar en el link.

[Unity - Manual: NetworkConnection \(unity3d.com\)](http://unity3d.com/Manual/NetworkConnection)

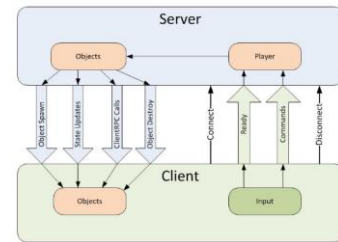


Ilustración 12 Ejemplo de conexión cliente servidor Unity

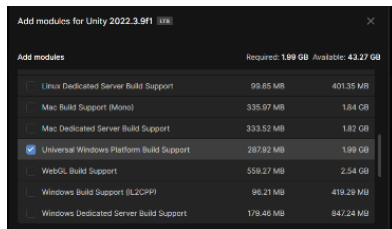


Ilustración 13 Instalación Modulo UWP

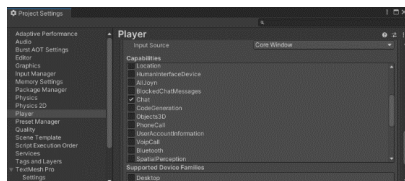


Ilustración 14 Modulo de Chat activado

Unity permite el intercambio de SMS y MMS, para ello es necesario instalar el modulo Universal Windows Platform permitiendo o bloqueando su uso.

Para llegar a ello seguimos la siguiente ruta Edit > Preferences > Project Settings > Player > Capabilities

[Unity - Manual: UWP Player settings \(unity3d.com\)](http://unity3d.com/Manual/UWPPlayerSettings)

**Android Studio:** para se puede realizar de la siguiente manera:

**SMS:** utilizando la clase android.telephony.VisualVoicemailSms permite gestionar el envío de mensajes.

<https://developer.android.com/reference/android/telephony/VisualVoicemailSms?hl=en>

**MMS:** utilizando la clase android.telephony.VisualVoicemailMms permite gestionar el envío de mensajes multimedia.

<https://developer.android.com/reference/android/provider/Telephony.Mms?hl=en>

e) Se han utilizado las clases necesarias para establecer conexiones y comunicaciones HTTP y HTTPS.

Revisando la documentación de Unity encontramos la Clase "UnityEngine.Networking" la cual permite gestionar las conexiones y comunicaciones HTTP y HTTPS.

Esta clase implementa metodos para el acceso y el envío de información. Además de propiedades para la configuración de la conexión.

Tenemos por ejemplo el método `UnityWebRequest.Post` que permite enviar información de un formulario a un servidor http.

En el siguiente link tenemos acceso a toda la información de la Clase `UnityEngine.Networking`

<https://docs.unity3d.com/ScriptReference/Networking.UnityWebRequest.html>

f) Se han utilizado las clases necesarias para establecer conexiones con almacenes de datos garantizando la persistencia.

Unity no tiene como tal una clase específica para gestionar las conexiones a BBDD, por ende, es necesario apoyarnos en C# .NET, utilizando la clase `Microsoft.Data.SqlClient` que permite crear la configuración necesaria para conectarnos a una BBDD y trabajar con ella.

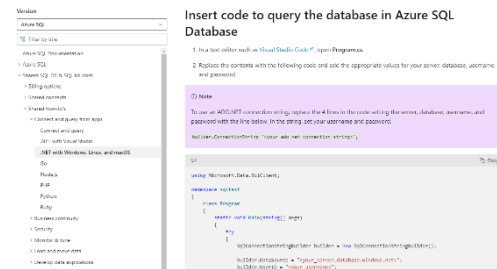


Ilustración 15 Pagina Web Learn.Microsoft.com con la explicación del código para conectarnos a BBDD

De manera local sí que podemos guardar información mediante Unity, podemos utilizar la clase `PlayerPrefs` que nos permite almacenar las preferencias del jugador entre las sesiones del juego. Pero esta información no está cifrada por lo cual se puede modificar fácilmente.

Otra forma de poder hacerlo de manera local sería creando desde C# flujos para escribir archivos serializados y guardar la información.

g) Se han realizado pruebas de interacción usuario-aplicación para optimizar las aplicaciones desarrolladas a partir de emuladores.

A medida que fui desarrollando el Juego fui probándolo para realizar ajustes e implementar funcionalidad y comprobar que todo funcionaba correctamente.

- Se comprobaron que los marcadores funcionan correctamente.
- Que la bola rebota en el entorno y las palas.
- El pulsado de la barra espaciadora Pausa el juego
- El pulsado de Botón de Escape te devuelve a la escena principal
- Se ajustó la velocidad de inicio de la bola ya que era demasiado baja.
- Se ajustó la velocidad máxima de la bola ya que aceleraba demasiado.
- Se añadió una pausa de 1 segundo despues de marcar un gol para volver a lanzar la bola.



Ilustración 16 Pruebas del Pong utilizando el Emulador de Unity

h) Se han empaquetado y desplegado las aplicaciones desarrolladas en dispositivos móviles reales.

Después de realizar los ajustes necesarios para poder crear el build instaló el juego en el dispositivo y comprueba que funciona. Me quedo pendiente ajustar el canvas y modificar los controles.

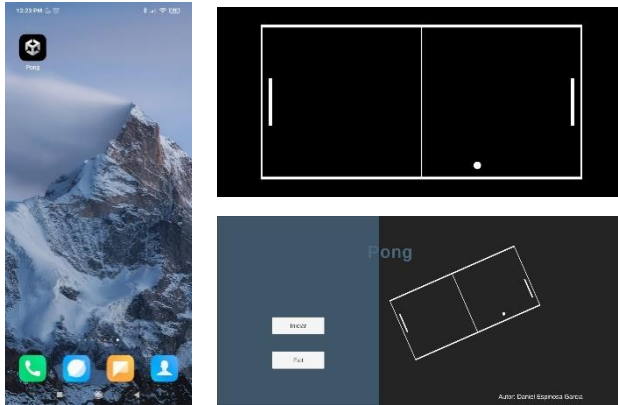


Ilustración 18 Instalación y prueba del Pong en Android

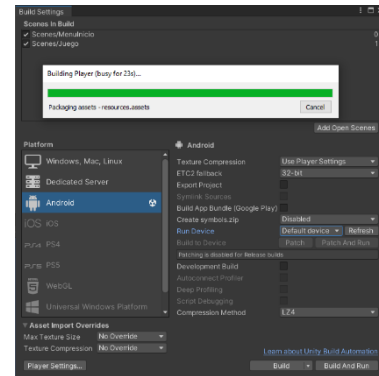


Ilustración 17 Build de APK

i) Se han documentado los procesos necesarios para el desarrollo de las aplicaciones.

Para el desarrollo del siguiente juego hemos empezado con la creación del GDD en el cual tenemos algunos dibujos sencillos de cómo se desarrollarán las escenas del juego así como un documento de texto en el cual se explican las mecánicas del mismo y el concepto general del juego.

En este caso realizaremos un Rogue-Like en el cual dados de 6 caras representarán las armas que utiliza el personaje y sus habilidades de clase a medida que avanza por el mapa generado proceduralmente y enfrenta enemigos, al derrotarlos obtendrá oro el cual podrá emplear para mejorar los dados. Al morir se calculará la experiencia obtenida y esto desbloqueará nuevos dados con los cuales volver a empezar la partida.

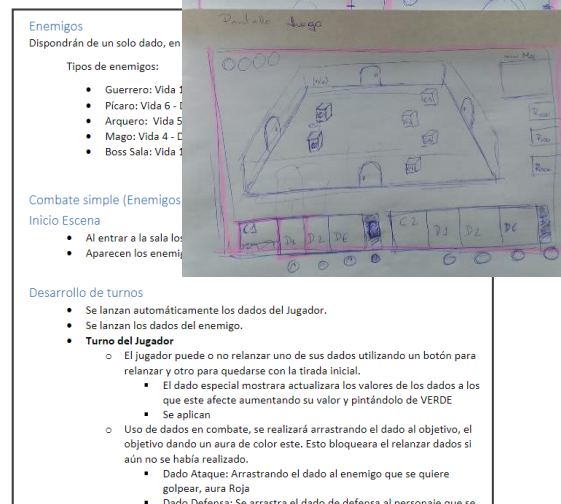


Ilustración 19 GDD del juego que estamos empezando a realizar

## Tabla Ilustraciones

ILUSTRACIÓN 1 <a href="https://gs.statcounter.com/os-market-share/mobile/worldwide">HTTPS://GS.STATCOUNTER.COM/OS-MARKET-SHARE/MOBILE/WORLDWIDE</a>	2
ILUSTRACIÓN 2 LOGO SWIFT	2
ILUSTRACIÓN 3 LOGO KOTLIN	2
ILUSTRACIÓN 4 CONFIGURACIÓN DE UNITY HUB PARA PERMITIR LA EXPORTACIÓN A SO ANDROID	3
ILUSTRACIÓN 5 IMAGEN DEL DESARROLLO DEL PONG COMO PRIMER PROYECTO DE CLASES	3
ILUSTRACIÓN 6 SCRIPT UTILIZADOS EN EL PONG	4
ILUSTRACIÓN 7 SCRIPT BALLMOVEMENT	5
ILUSTRACIÓN 8 SCRIPT PARA CONTROLAR LA ACELERACIÓN DE LA BOLA.	5
ILUSTRACIÓN 9 UNITY EMULACIÓN DE DISPOSITIVO MÓVIL	6
ILUSTRACIÓN 10 DIAGRAMA DE COMO UTILIZA EL JUEGO PONG LOS SCRIPTS EN LA EJECUCIÓN DE LA APLICACIÓN	6
ILUSTRACIÓN 11 CANVAS DEL MENÚ DE INICIO DEL JUEGO PONG	7
ILUSTRACIÓN 12 EJEMPLO DE CONEXIÓN CLIENTE SERVIDOR UNITY	8
ILUSTRACIÓN 13 INSTALACIÓN MODULO UWP	8
ILUSTRACIÓN 14 MODULO DE CHAT ACTIVADO	8
ILUSTRACIÓN 15 PAGINA WEB LEARN.MICROSOFT.COM CON LA EXPLICACIÓN DEL CÓDIGO PARA CONECTARNOS A BBDD	9
ILUSTRACIÓN 16 PRUEBAS DEL PONG UTILIZANDO EL EMULADOR DE UNITY	9
ILUSTRACIÓN 17 BUILD DE APK	10
ILUSTRACIÓN 18 INSTALACIÓN Y PRUEBA DEL PONG EN ANDROID	10
ILUSTRACIÓN 19 GDD DEL JUEGO QUE ESTAMOS EMPEZANDO A REALIZAR	10

## Bibliografía

gs.statcounter.com, 2023. *gs.statcounter.com*. [En línea]

Available at: <https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-200901-202309>

<https://developer.android.com>, 2023. *https://developer.android.com*. [En línea]

Available at: <https://developer.android.com/guide/topics/connectivity/bluetooth?hl=es-419>

<https://developer.android.com>, s.f. *https://developer.android.com*. [En línea]

Available at: <https://developer.android.com/training/connect-devices-wirelessly?hl=es-419>

<https://docs.unity3d.com/>, s.f. *https://docs.unity3d.com/*. [En línea]

Available at: <https://docs.unity3d.com/ScriptReference/PlayerPrefs.html>

<https://docs.unity3d.com>, 2022. *https://docs.unity3d.com*. [En línea]

Available at: <https://docs.unity3d.com/ScriptReference/Networking.UnityWebRequest.html>

<https://es.wikipedia.org>, s.f. *https://es.wikipedia.org*. [En línea]

Available at: [https://es.wikipedia.org/wiki/Unity\\_\(motor\\_de\\_videojuego\)](https://es.wikipedia.org/wiki/Unity_(motor_de_videojuego))

<https://geekflare.com>, 2023. *https://geekflare.com*. [En línea]

Available at: <https://geekflare.com/es/best-ide-for-mobile-app-development/>

<https://learn.microsoft.com>, 2023. *https://learn.microsoft.com*. [En línea]

Available at: <https://learn.microsoft.com/en-us/azure/azure-sql/database/connect-query-dotnet-core?view=azuresql>

<https://www.bluestacks.com>, s.f. *https://www.bluestacks.com*. [En línea]

Available at: <https://www.bluestacks.com/es/index.html>

<https://www.clasificacionde.org>, s.f. *https://www.clasificacionde.org*. [En línea]

Available at: <https://www.clasificacionde.org/clasificacion-de-dispositivos-moviles/>

<https://www.protecciondatos.org>, 2019. *https://www.protecciondatos.org*. [En línea]

Available at: <https://www.protecciondatos.org/tipos-de-dispositivos-moviles/>

<https://yuzu-emu.org>, s.f. *https://yuzu-emu.org*. [En línea]

Available at: <https://yuzu-emu.org/wiki/home/>

[www.armadilloamarillo.com](http://www.armadilloamarillo.com), 2022. *www.armadilloamarillo.com*. [En línea]

Available at: <https://www.armadilloamarillo.com/blog/mejores-tecnologias-para-desarrollo-de-aplicaciones-moviles-2022/>