

**SUPER AiR Hockey**

# **Proyecto Ingeniería de Aplicaciones 2019**

**Universidad  
Nacional del Sur**

**Lucio Arce  
Manuel Garat  
Luciano Vidili**

Objetivo

**Objetivo**

## Objetivo

# **SUPER AiR Hockey es un juego:**

- De hockey de aire para el sistema operativo Android
- Multijugador
- De Realidad Mixta

# Objetivo

Hockey de aire:



# Objetivo

## **Multijugador:**

- Cada jugador con su dispositivo móvil
- Conectados entre sí
- Máximo dos jugadores

# Objetivo

**Realidad Aumentada (AR):**





# Objetivo

Realidad  
Virtual (VR):



# Objetivo

Realidad Mixta:

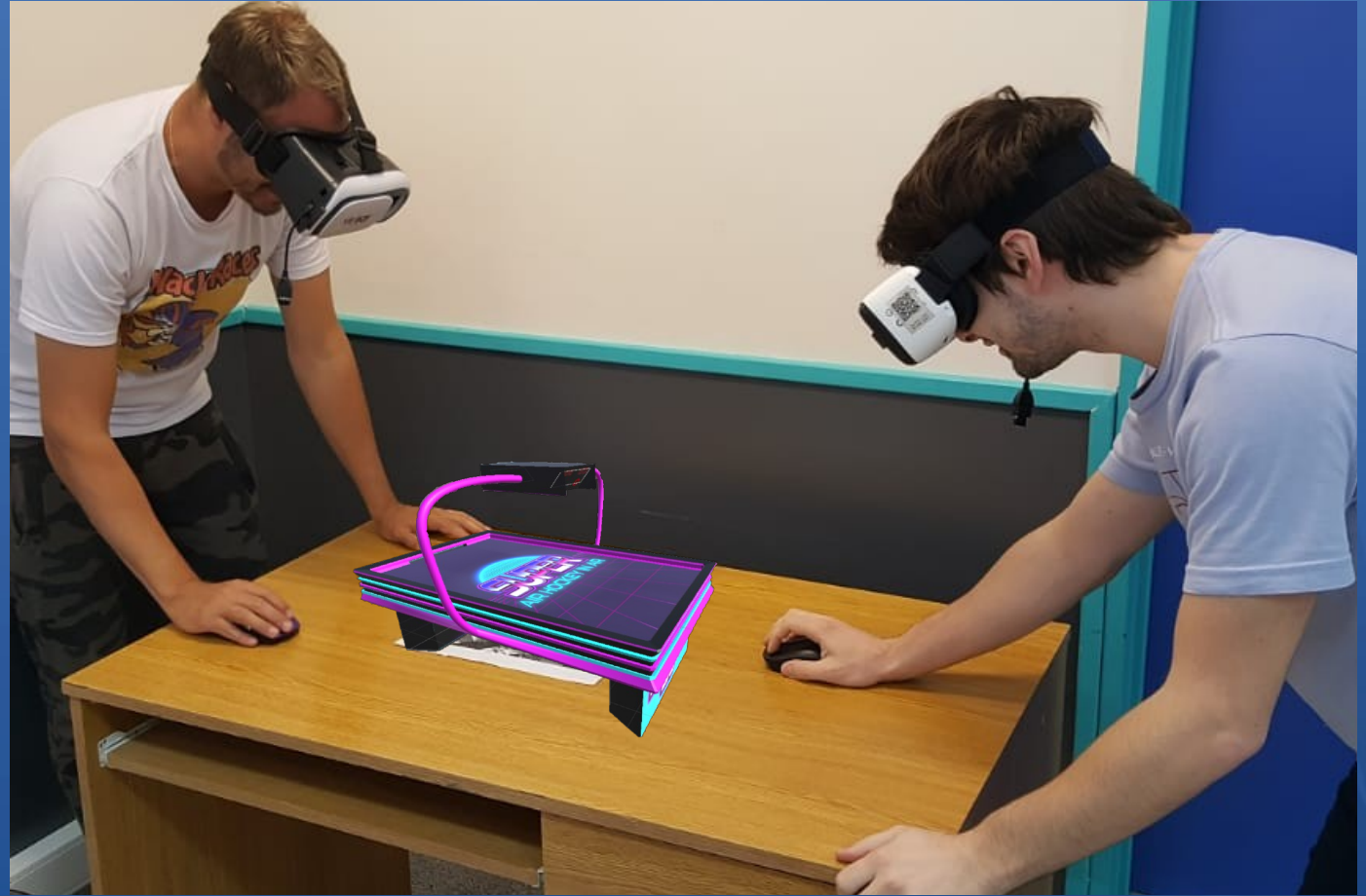
AR + VR





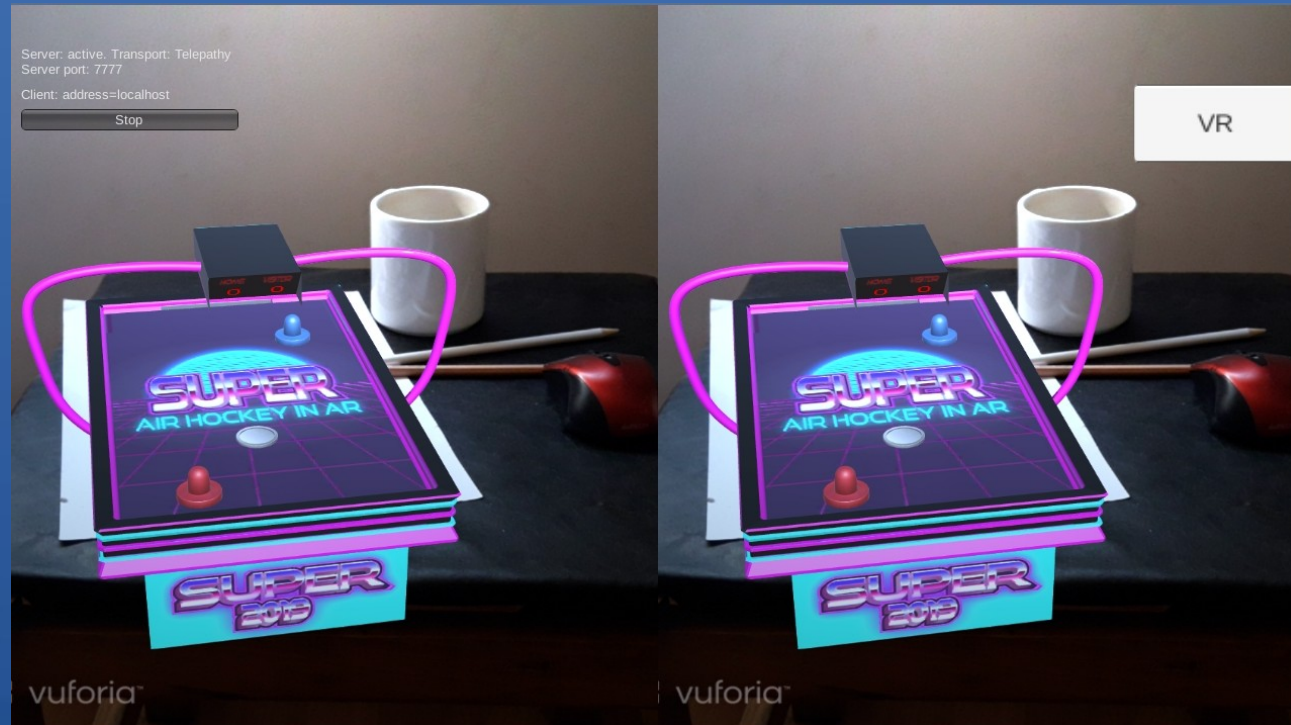
# Objetivo

**SUPER AiR  
Hockey (desde  
fuera):**



# Objetivo

## SUPER AiR Hockey (desde dentro):



Hardware

**Hardware**

# Hardware

## Aplicación:

- Para dispositivos móviles Android, versión 4.4 en adelante
- La cámara es requisito para la AR
- Conectados en la misma red (WI-FI)

# Hardware

## Realidad virtual:

Visores VR BOX o similar





# Hardware

## Movimiento del jugador:

Mouse + USB OTG  
(On-The-Go)



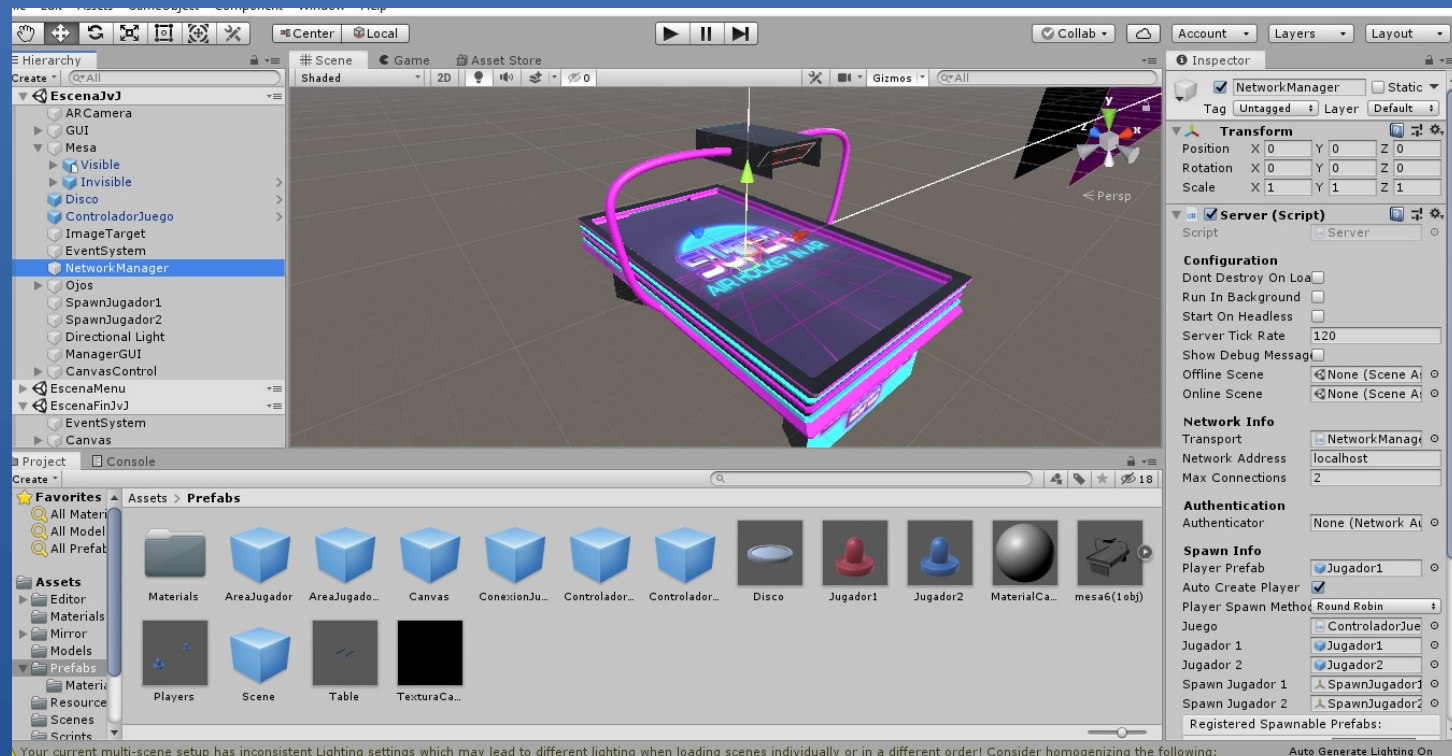
Tecnologías

**Tecnologías**

# Tecnologías

## Entorno:

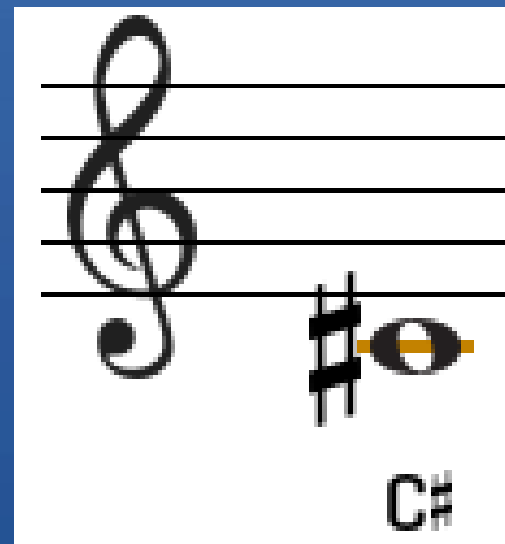
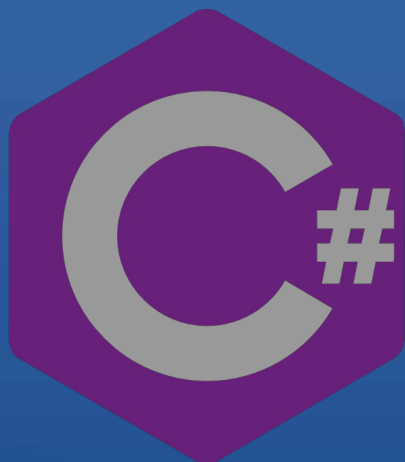
## Unity



# Tecnologías

## Lenguaje de programación:

Scripting con C# para comportamiento no establecido por Unity

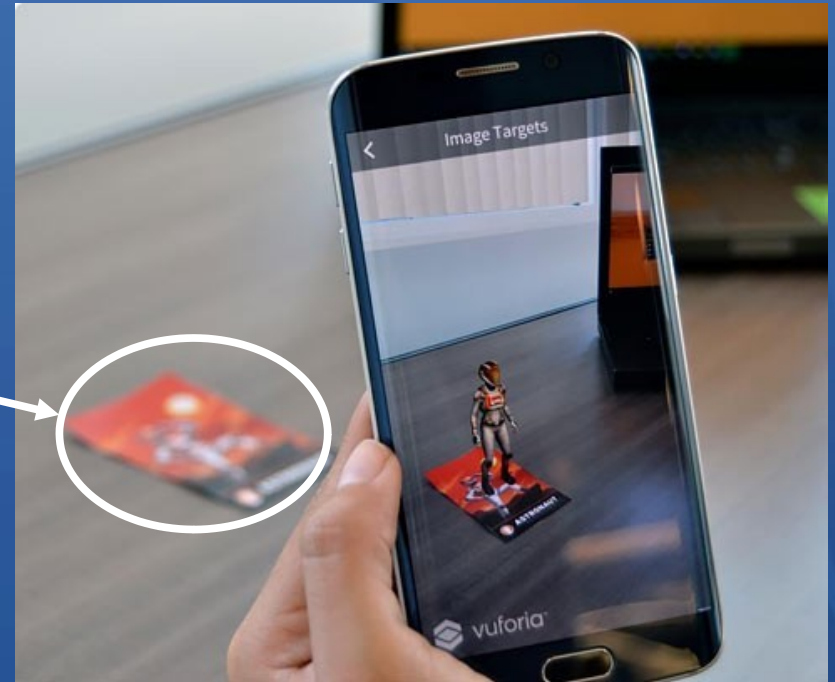


# Tecnologías

## Realidad aumentada:

- Módulo Vuforia para Unity
- Marcador

vuforia® engine™

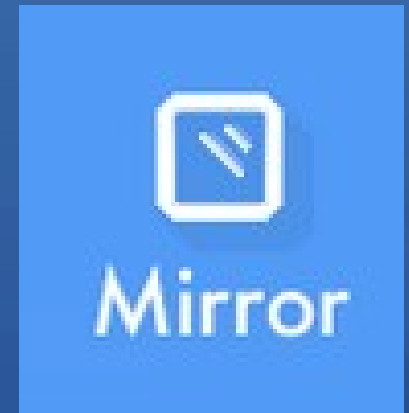




## Networking:

### API Mirror para Unity:

- Evolución del networking de Unity
- Confiable (utilizado en MMOs)
- 
- FOSS



Diseño

**Diseño**

# Diseño

## Conectividad (opción 1):



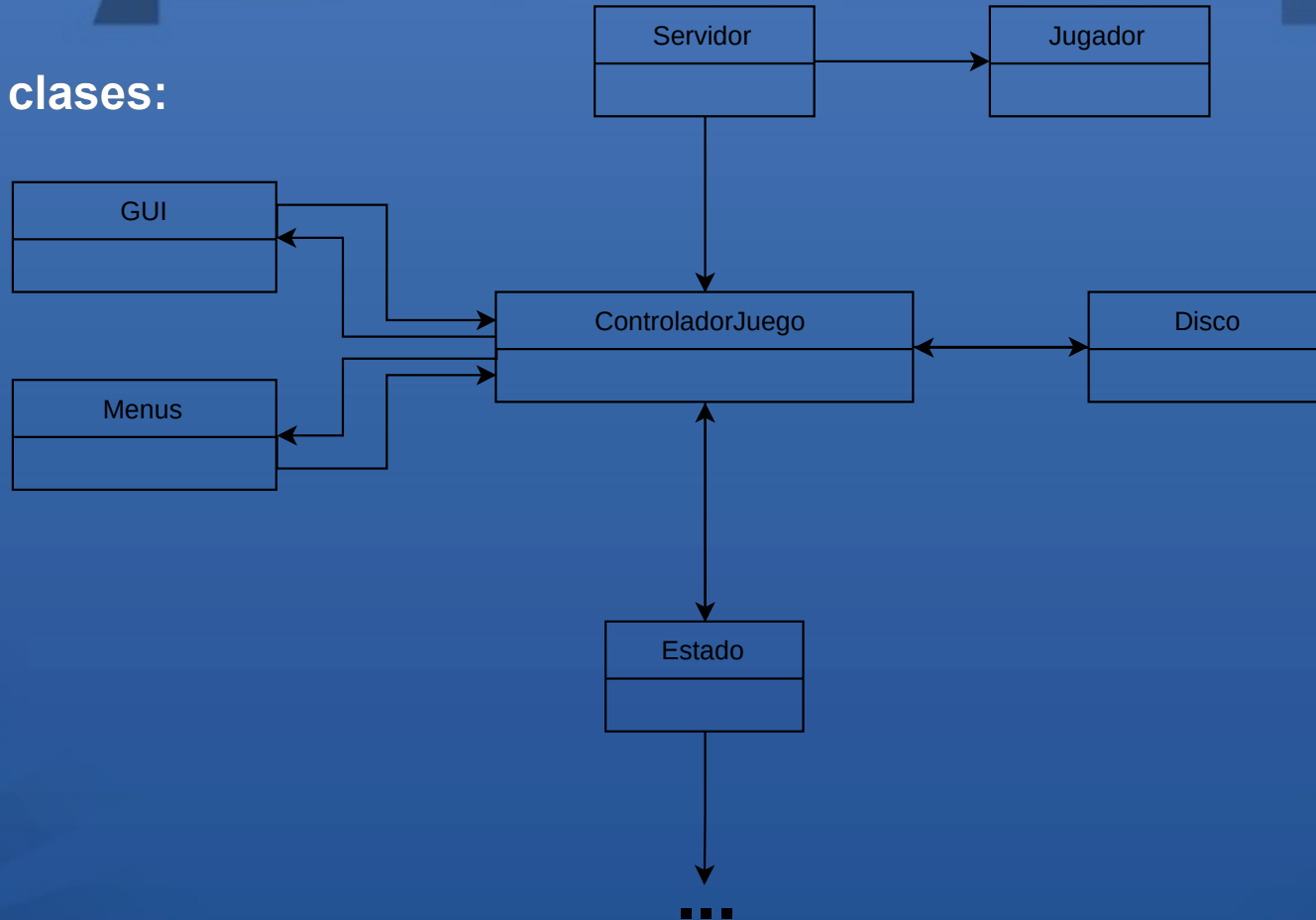
# Diseño

## Conectividad (opción 2):



# Diseño

## Diagrama de clases:





# Servidor

- Genera los mazos (jugadores) al conectarse
- Notifica al Controlador de Juego
- Cuando hay dos jugadores, se inicia la partida

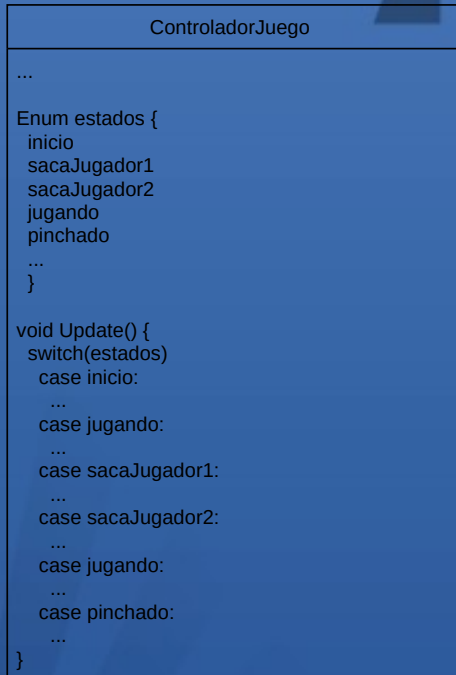
# Controlador del Juego:

- Organizado en estados:
  - Inicio
  - Jugando
  - ...

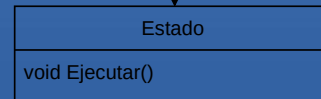
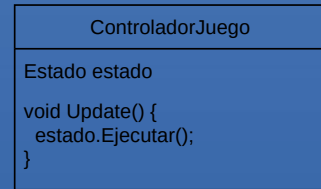
# Controlador del Juego:

- Se utiliza el patrón de diseño STATE. Ventajas:
  - Se facilita agregar estados nuevos
  - Menos código en el Controlador

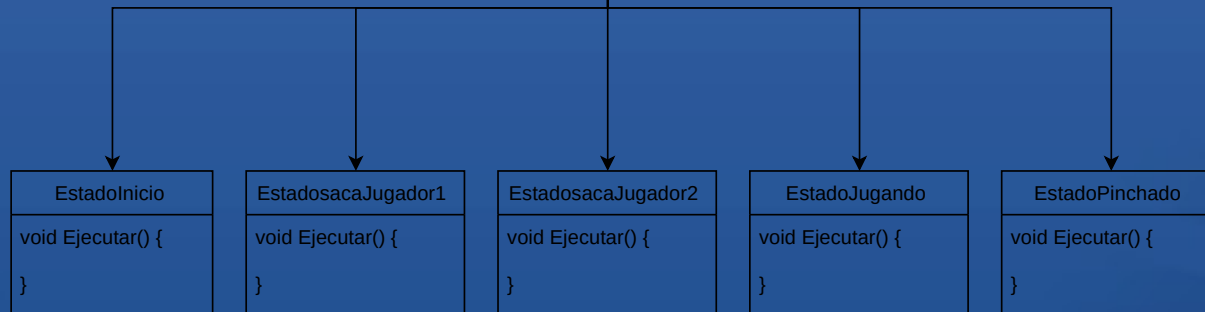
# Diseño



Antes de refactorizar



Después de refactorizar



## Disco:

- Posición calculada en el **servidor**
- Servidor → Clientes



# Diseño

## Jugador:

- Posición calculada en cada **cliente**
- Cliente → Servidor → Otro Cliente

## Código:

- Se respetan las reglas de clean code
- Se respetan los principios **SOLID**

# Diseño

## Notas:

- Mesa diseñada en Blender
- Modo no realidad mixta, con un botón

Desafíos

**Desafíos**

# Desafíos

## Ninguna experiencia con las tecnologías:

- Unity: cuesta aprehender la relación entre código y entorno
- C#: adaptación rápida al lenguaje
- AR: Vuforia simplifica considerablemente este aspecto
- VR: falta de soporte para VRBox con Vuforia

## Desafíos

### Inconvenientes menores:

- Refactorización para implementar patrones de diseño
- Diseñar un juego atractivo y cómodo para quien juega lleva tiempo
- Complicaciones para paralelizar el trabajo entre todos los integrantes

## Desafíos

**Inconveniente mayor:**

Sí... uno:





**Networking**

## Desafíos

# Networking:

- Poca documentación útil de Mirror
- Un esquema de conexión correcto y robusto es inherentemente complicado de diseñar e implementar
- Ninguna experiencia previa con este tipo de programación

Extensibilidad

# Extensibilidad

# Extensibilidad

- Mejorar la interacción con menús
- Funcionamiento fuera de una red local (internet)
- Físicas de golpes disco-mazo mejorables
- Efectos de sonido y/o música

Conclusiones

**Conclusiones**

# Conclusiones

- Pudimos lograr los objetivos:
  - Un juego para Android, multijugador y en realidad mixta
  - Bien diseñado → patrones de diseño
- Sin tener experiencia con las tecnologías utilizadas, los resultados obtenidos son destacables

Fin

**Gracias por su atención**

**¿Preguntas?**

