



**UNIVERSIDAD DE CASTILLA-LA MANCHA  
ESCUELA SUPERIOR DE INFORMÁTICA**

**MÁSTER UNIVERSITARIO EN INGENIERÍA  
INFORMÁTICA**

**RehabiliTEA: Sistema basado en gamificación  
para evaluar la reactividad y conectividad en  
niños con desórdenes en el neurodesarrollo**

**Roberto Plaza Romero**

**Julio, 2020**



**REHABILITEA: SISTEMA BASADO EN GAMIFICACIÓN PARA EVALUAR  
LA REACTIVIDAD Y CONECTIVIDAD EN NIÑOS CON DESÓRDENES EN EL  
NEURODESARROLLO**





**UNIVERSIDAD DE CASTILLA-LA MANCHA  
ESCUELA SUPERIOR DE INFORMÁTICA**

**MÁSTER UNIVERSITARIO EN INGENIERÍA  
INFORMÁTICA**

**RehabiliTEA: Sistema basado en gamificación  
para evaluar la reactividad y conectividad en  
niños con desórdenes en el neurodesarrollo**

**Autor: Roberto Plaza Romero**

**Tutor académico: Dr. David Vallejo Fernández**

**Cotutor académico: Dr. Abel Toledano González**

Julio, 2020



**Roberto Plaza Romero**

Ciudad Real – España

© 2020 Roberto Plaza Romero

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Se permite la copia, distribución y/o modificación de este documento bajo los términos de la Licencia de Documentación Libre GNU, versión 1.3 o cualquier versión posterior publicada por la *Free Software Foundation*; sin secciones invariantes. Una copia de esta licencia esta incluida en el apéndice titulado «GNU Free Documentation License».

Muchos de los nombres usados por las compañías para diferenciar sus productos y servicios son reclamados como marcas registradas. Allí donde estos nombres aparezcan en este documento, y cuando el autor haya sido informado de esas marcas registradas, los nombres estarán escritos en mayúsculas o como nombres propios.



**TRIBUNAL:**

**Presidente:**

**Vocal:**

**Secretario:**

**FECHA DE DEFENSA:**

**CALIFICACIÓN:**

**PRESIDENTE**

**VOCAL**

**SECRETARIO**

Fdo.:

Fdo.:

Fdo.:



# Resumen

El trastorno del espectro autista o TEA es uno de los desórdenes neurológicos más comunes en niños y adolescentes que presentan déficits, fijándose en una prevalencia según datos del INE de 0.33 por cada 1000 niños nacidos en España, lo que se traduce en una gran cantidad de niños con problemas o rasgos asociados a los trastornos del neurodesarrollo y en especial esta patología (INE, 2019).

El estudio de los síntomas físicos de esta patología del neurodesarrollo dista mucho de estar tan avanzado como lo pueda estar en otros trastornos como el trastorno de déficit de atención. Este documento propone una plataforma de videojuegos basada en gamificación como medio para la estimulación, rehabilitación y educación de sus usuarios en un estudio sobre el impacto del TEA en la población infantil.

La unión de técnicas de gamificación y sensores de espectrometría hace posible un estudio en profundidad sobre cómo se desarrolla un individuo con esta patología del neurodesarrollo. RehabiliTEA es entonces, una plataforma de rehabilitación, para niños con enfermedades del neurodesarrollo. Esta plataforma consta de minijuegos para reforzar las capacidades cerebrales de estos niños y, además, incorpora maneras mediante las cuales profesionales terapeutas pueden extraer métricas sobre el desarrollo funcional de los usuarios.



# Abstract

The autism spectrum disorder or ASD is one of the most common neurological disorders among teens and kids with this kind of disorders, 0.33 per 1000 kids born in Spain looking at the data provided by the INE. This translates into a large number of children with problems or traits associated to neurodevelopmental disorders, specially autism spectrum disorder (INE, 2019).

The scientific study of physical symptoms of this neurodevelopmental disorder are not as refined as the studies we can find in other pathologies like attention deficit hyperactivity disorder. This document proposes a gamification based video game platform as a mean to stimulate, rehabilitate and educate its users in a study about the impact of autism spectrum disorder among child population

The union of gamification techniques and spectrometric sensors makes possible an in-depth study on how an individual develops with this neurodevelopment pathology. Rehabili-TEA is, therefore, a rehabilitation platform for kids with neurodevelopmental diseases. This platform is composed of games designed to enhance the cerebral capabilities of its users, and it also allows professional therapists to extract metrics on functional development of the users.



# Agradecimientos

Es de obligado cumplimiento agradecer a mi familia que esto haya sido posible. Mamá, Papá, además del máster dentro de poco tendréis que aflojar dinero para la mudanza a otro país, así que espero que disfrutéis esta dedicatoria porque os va a salir cara. También se os quiere por otras cosas pero eso mejor en privado. Raúl, de ti tampoco me olvido.

Monescillo, Chamorro, sois mis dos más mejores amigos, lo sabéis, ¿Hace falta mencionar el apoyo moral? ¿Lo que me habéis aguantado mientras daba el coñazo con Suiza? ¿Los kilos extra por esas inolvidables noches? Se os quiere y, espero, siempre se os querrá. ¡Afortunadamente ahora tendré más tiempo para demostrarlo!

Hacen falta unas gracias especiales para los tutores de este trabajo. Gracias, de corazón, gracias por haber depositado esfuerzo, ganas y, sobretodo, por haber tenido fe en mi. Sobre todas las cosas, un pupilo le debe sus logros al maestro y espero no decepcionar con los resultados.

Por último, gracias al equipo de máster universitario en ingeniería informática por crear un entorno en el que un estudiante mediocre como el que venía siendo pueda acabar, momentáneamente, su carrera como estudiante por todo lo alto. El entorno es envidiable y me alegra y reconforta que los esfuerzos se vean recompensados.



*Al Atlas que todos llevamos dentro.*



# Índice general

<b>Resumen</b>	<b>V</b>
<b>Abstract</b>	<b>VII</b>
<b>Agradecimientos</b>	<b>IX</b>
<b>Índice general</b>	<b>XIII</b>
<b>Índice de figuras</b>	<b>XV</b>
<b>Listado de acrónimos</b>	<b>XVII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Planteamiento general . . . . .	1
1.2. Marco de trabajo . . . . .	2
1.2.1. Herramientas de desarrollo . . . . .	2
1.2.2. Dispositivos Hardware . . . . .	3
1.3. Competencias a adquirir . . . . .	5
1.4. Estructura del documento . . . . .	6
<b>2. Objetivos</b>	<b>7</b>
2.1. Objetivo general . . . . .	7
2.2. Objetivos específicos . . . . .	7
<b>3. Antecedentes.</b>	<b>9</b>
3.1. Disfunciones neurológicas en la infancia . . . . .	9
3.1.1. Trastorno de déficit de atención e hiperactividad . . . . .	9
3.1.2. Trastorno del espectro autista . . . . .	10
3.2. Gamificación y tratamiento médico . . . . .	11
3.3. Desarrollo para plataformas móviles . . . . .	14

<b>4. Método de trabajo</b>	<b>19</b>
4.1. Metodología de trabajo y desarrollo . . . . .	19
4.2. Planificación y distribución del trabajo . . . . .	20
<b>5. Resultados</b>	<b>21</b>
5.1. Arquitectura general . . . . .	21
5.2. Técnicas de gamificación . . . . .	23
5.3. Esquema de medición de métricas . . . . .	25
5.4. Prototipo de aplicación para dispositivos móviles . . . . .	28
5.4.1. Marco general . . . . .	29
5.4.2. Minijuegos . . . . .	30
5.4.3. Gestión de perfiles . . . . .	32
5.5. Infraestructura y protocolos web . . . . .	33
5.6. Validación y pruebas . . . . .	34
5.6.1. Planteamiento . . . . .	34
5.6.2. Resultados . . . . .	35
<b>6. Conclusiones y líneas de trabajo futuro.</b>	<b>37</b>
6.1. Análisis de resultados . . . . .	37
6.2. Adquisición de competencias y cumplimiento de objetivos . . . . .	37
6.3. Líneas de trabajo futuro . . . . .	38
6.4. Reflexión personal . . . . .	40
<b>A. Infraestructura como código de la aplicación</b>	<b>43</b>
<b>B. Contenidos multimedia utilizados para el desarrollo del proyecto.</b>	<b>45</b>
<b>C. Resultados de la validación.</b>	<b>47</b>
<b>Referencias</b>	<b>49</b>

# Índice de figuras

1.1. Casco «Emotiv Insight».	4
3.1. <i>Hype Cycle</i> de Gartner	12
3.2. Imagen promocional de REHABILITY.	14
3.3. Juegos comúnmente usados para mejorar la atención y memoria a corto plazo.	14
3.4. Logos de los principales motores gráficos.	18
5.1. Diagrama de despliegue de RehabiliTEA.	21
5.2. Modelo de datos del alcance original.	26
5.3. Modelo de datos del prototipo.	27
5.4. Ejemplo de cuadro de mando hecho con Cronograf.	28
5.5. Diagrama de clases simplificado de las principales clases de la aplicación móvil.	29
5.6. Capturas de la aplicación final.	31
B.1. Propuesta visual del prototipo.	45
B.2. Ejemplos de assets utilizados en el juego.	46



# Listado de acrónimos

<b>TEA</b>	Trastorno del Espectro Autista
<b>TDAH</b>	Trastorno de Déficit de Atención e Hiperactividad
<b>CPU</b>	Unidad central de proceso
<b>iOS</b>	iPhone Operating System
<b>IDE</b>	Entorno de desarrollo integrado
<b>SO</b>	Sistema Operativo
<b>API</b>	Interfaz de Programación de Aplicaciones
<b>NDK</b>	Kit de desarrollo nativo
<b>JVM</b>	Máquina Virtual de Java
<b>LTS</b>	long term support
<b>ASD</b>	Autism Spectrum Disorder
<b>CI/CD</b>	integración y entrega continua
<b>HTML</b>	Hypertext Markup Language
<b>IOT</b>	internet de las cosas
<b>ORM</b>	Object Relational Mapper
<b>WSGI</b>	Web Server Gateway Interface
<b>MVC</b>	Modelo, Vista y Controlador
<b>HTTP</b>	Hypertext Transfer Protocol
<b>APK</b>	Android application package
<b>ORM</b>	Object-Relational Mapper
<b>TSDB</b>	Time series database
<b>SQL</b>	Structured Query Language
<b>XML</b>	eXtended Markup Language
<b>JSON</b>	JavaScript Object Notation

## 0. LISTADO DE ACRÓNIMOS

<b>WSGI</b>	Web Server Gateway Interface
<b>REST</b>	Representational state transfer
<b>TICK</b>	Telegraf, InfluxDB, Chronograf y Kapacitor
<b>UCLM</b>	Universidad de Castilla-La Mancha

## Capítulo 1

# Introducción

**R**EHABILITEA es el nombre comercial del proyecto presentado. Este trabajo nace de una colaboración entre diferentes campus dentro de la Universidad de Castilla-La Mancha. La simbiosis entre profesionales de la salud y las ciencias de computación es, afortunadamente, cada vez más común, y los proyectos derivados de esta tienden a generar un elevado interés ya que procuran solventar problemas sociales reconocidos con tecnología vanguardista.

### 1.1 Planteamiento general

El estudio, en términos de salud, de patologías como el Trastorno del Espectro Autista (TEA) se ha hecho, hasta la fecha desde un punto de vista social, qué aspectos sociales influyen en la vida de un paciente de TEA y cómo integrar a este socialmente. Sin embargo, de otras patologías como el TDAH (Trastorno de Déficit de Atención e Hiperactividad) se han realizado estudios sobre los síntomas del paciente suponen a nivel físico (entendiendo a nivel físico como a nivel corporal).

Es sabido a día de hoy que existe una alteración de las ondas cerebrales en un paciente de TDAH cuando este no consigue concentrarse en una tarea[LPP<sup>+</sup>05] [YFYCZ<sup>+</sup>07]. Con el fin de evaluar los síntomas del TEA a nivel cognitivo se empezaron a buscar sensores sensibles a este tipo de **variables cognitivas**. La principal variable cognitiva que quería ser medida es la de la actividad cerebral, es por esto que resulta interesante añadir una vía de estudio como esta a la plataforma.

Este planteamiento requiere de una actividad que sirva de estímulo sensorial a la hora de hacer la medición. Un juego cuya mecánica principal se base en memoria, cálculo numérico, deducción y discriminación de formas y colores supone una actividad ideal para este fin [BFC<sup>+</sup>09], [Dic05], [She13]. La gamificación ha probado ser útil para labores de rehabilitación y apoyo [BLL<sup>+</sup>15], [BMC<sup>+</sup>09], ya que los videojuegos consiguen crear una mayor vínculo de compromiso con el usuario y conseguir que no vea la tarea de rehabilitación como tediosa.

Conseguir un método de rehabilitación de patologías como puedan ser el trastorno de déficit de atención que requieran de menos esfuerzo por el paciente puede ser posible. Con

## 1. INTRODUCCIÓN

el uso de dispositivos como smartphones o tablets no sólo es posible crear una plataforma de tratamiento ubicua. Por otro lado, el estímulo visual que ofrece un videojuego puede ayudar a que una terapia no sea percibida como un ejercicio sino como un pasatiempo.

Análogamente, la proliferación de las técnicas de tratamiento de datos masivos y de aprendizaje automático hace posible obtener conocimiento en base a casi cualquier fuente de datos. Los hábitos de los usuarios de esta plataforma podrían suponer una fuente de datos muy valiosa para profesionales que busquen estudiar la interacción de niños con Trastorno del Espectro Autista con la tecnología, e incluso, para profesionales del aprendizaje automático y la clasificación.

De esta manera, los resultados que se esperan obtener del proyecto son ambiciosos. Por un lado, una plataforma de rehabilitación de enfermedades del neurodesarrollo es un proyecto que abarataría costes sociales y personales en terapias de integración de pacientes con TEA. Por otro lado, también se ofrece valor científico por medio del estudio de las características físicas del Trastorno del Espectro Autista.

Por medio de unir estas visiones se llega a la conclusión de que una aplicación con actividades lúdico-recreativas para niños con Trastorno del Espectro Autista además de poder suplir parte de su formación supone una muy buena oportunidad para estudiar el impacto que tienen los videojuegos en el neurodesarrollo infantil. Aporta, también una nueva vía de estudio, ya que el estudio sobre el impacto de los videojuegos en pacientes de TEA no se ha estudiado formalmente.

Este trabajo fin de máster propone el desarrollo de una plataforma de rehabilitación de Trastorno del Espectro Autista por medio de técnicas de gamificación, la cual, además de labores de rehabilitación y soporte, dispondrá de un módulo de recolección de datos sobre fenómenos cerebrales y hábitos de juego de sus usuarios, teniendo como objetivo su posterior estudio.

## 1.2 Marco de trabajo

Para llevar a cabo el desarrollo de RehabiliTEA se ha utilizada una amplia variedad de herramientas hardware y software. El proyecto usó entornos Linux y Windows, tecnologías cloud, tecnologías de código abierto y marcos de trabajo privativos. En el ámbito hardware la heterogeneidad del proyecto también es notable, ya que se utilizaron dispositivos tanto Android como de plataforma cerrada, aportando desafío y atractivo a la tarea.

### 1.2.1 Herramientas de desarrollo

#### Entorno de desarrollo Linux

Con el fin de agilizar el desarrollo de la sección backend de la aplicación esta ha sido dockerizada. El *workflow* de desarrollo Docker en Linux es más simple ya que la idea de Docker es la de virtualización a nivel sistema operativos de distribuciones con Kernel de Linux. Esta

herramienta junto con «Docker Compose» y el plugin de Python «Gunicorn» va a permitir desplegar una infraestructura *production ready* en la nube de manera muy sencilla.

Las herramientas usadas para el resto del desarrollo han sido: Visual Studio Code y VIM como editores de código, PostgreSQL e InfluxDB como bases de datos, Python como lenguaje con la librería «Flask» para hacer más rápido el desarrollo de aplicaciones web, «Nginx» es un servidor web que podrá ser utilizado para montar un proxy inverso a la aplicación y «Graphana» permitirá realizar gráficos de forma muy sencilla. La distribución Linux elegida ha sido Ubuntu 20.04 y su predecesora, Ubuntu 18.04.

### **Entorno de desarrollo Windows**

La tarea de desarrollo de videojuegos, por otro lado está, a día de hoy, casi monopolizada por el sistema operativo de Microsoft. Los principales motores gráficos comerciales están basados en la API gráfica DirectX, nativa a Windows. «Unity», por suerte dispone de más de un subsistema de renderizado, pero, al igual que Docker, es una herramienta que gana mucho cuando la usamos en el entorno pensado para esta.

Unity ofrece dos interfaces de programación, una en el lenguaje C# y otra por medio de un runtime de JavaScript llamado «UnityScript». Como editor de código se ha usado Visual Studio Code antes que Visual Studio por razones de precio y usabilidad. Análogamente se ha utilizado Git, con su aplicación de escritorio «Github Desktop» para realizar la gestión de versiones dentro de un repositorio alojado en la forja de repositorios «Github».

## **1.2.2 Dispositivos Hardware**

### **Computador de desarrollo**

Para desarrollar este proyecto se ha usado un portátil de la marca Medion. El modelo en cuestión es el Erazer p6689. Los marcos de trabajo sobre los que se ha hablado se han instalado en este dispositivo por medio del uso de particiones. A nivel comercial es considerado una gama baja de portátil de juegos, es decir, tendría componentes algo más potentes aunque menos autonomía.

Hoy por hoy, a excepción del disco en estado sólido, esta configuración es muy común y podría considerarse el hardware mínimo para una máquina de desarrollo de videojuegos. No obstante, al tratarse de un desarrollo para plataformas móviles la exigencia el requisito de potencia de computación gráfica no es tan grande. De cualquier manera es recomendable aprovisionar una máquina de desarrollo de videojuegos con una GPU moderna pensada para desarrollo gráfico (las gamas RTX y GTX de Nvidia y la RX de AMD), ya que así se ve asegurada la compatibilidad con controladores y tecnologías más recientes.

## 1. INTRODUCCIÓN

### Terminal de desarrollo

Con el fin de construir, desplegar y depurar la aplicación Android generada a lo largo del proyecto se ha hecho uso de un terminal personal. El nombre completo del dispositivo es «Redmi Note 8 Pro», del fabricante «Xiaomi». La aplicación se compila como un paquete Android, es decir, cualquier dispositivo con ese sistema operativo, sea tablet, terminal u otro tipo de dispositivo podrá lanzar la aplicación, siempre que este dispositivo disponga de pantalla táctil.

Prácticamente cualquier terminal Android que haya salido en el último año, tal vez incluso móviles anteriores, puede servir para esta tarea. Con un coste de unos 100 dólares o 95 euros podemos encontrar dispositivos de gama entrante capaces de correr la aplicación final, razón por la cual el precio de un terminal de desarrollo debería estimarse a la baja.

### Casco de espectrometría

El dispositivo del cual nace este proyecto fue el «Insight», de la marca «Emotiv». Emotiv es una empresa de bioinformática especializada en el desarrollo de sensores de la actividad cerebral. Insight es el modelo de casco de espectrometría más comedido de la marca. Su precio, a día de hoy ronda los 265 euros ó 300 dólares americanos.

El Emotiv Insight cuenta con cinco sensores de polímero semiseco, AF3, AF4, T7, T8 y Pz capaces de medir los canales las ondas alfa, beta, theta y gamma del encefalograma. Estas señales se toman con una frecuencia de 128Hz con una resolución de 14bits. Análogamente, el casco dispone de otros sensores como giroscopios y sensores de movimiento con nueve grados de libertad. Funciona con una batería recargable por USB con una autonomía de unas nueve horas y se puede comunicar con el PC por medio de protocolo Bluetooth y USB.

La manera en la que funciona este dispositivo es la siguiente. El driver, Emotiv App, es un programa tipo demonio con el cual es posible sincronizar cualquier sensor de Emotiv. Este controlador tiene, también, una ventana en la que es posible comprobar el estado del dispositivo, consultar la batería restante y conectividad de los sensores, entre otros valores.



Figura 1.1: Casco «Emotiv Insight».

El punto fuerte de este dispositivo es que permite a un desarrollador acceder a él por medio de una API<sup>1</sup> programable. De esta manera, el terapeuta podrá elegir qué información quiere ver y cómo la quiere ver. Esta flexibilidad en cuanto a los datos permite un estudio menos generalista y más pormenorizado de los datos

### 1.3 Competencias a adquirir

#### **[CE4] Capacidad para modelar, diseñar, definir la arquitectura, implantar, gestionar, operar, administrar y mantener aplicaciones, redes, sistemas, servicios y contenidos informáticos**

A lo largo del proyecto será necesario desarrollar una plataforma de gestión de perfiles. Para ello serán necesarios conocimientos sobre diseño de infraestructuras TI, también será necesario llevar a cabo las labores de despliegue y mantenimiento durante tres meses de experimento. Esta competencia engloba todo lo que un administrador de sistemas hace y será necesario administrar el sistema web de RehabiliTEA.

Se usará arquitectura en base a microservicios, un enfoque puntero el cual requiere de una especial atención a la hora de planificar, desplegar y mantener la aplicación. De hecho, la propia elección de este paradigma de infraestructura puede considerarse una tarea de diseño de servicios, tarea relacionada con esta competencia.

#### **[CE14] Capacidad para conceptualizar, diseñar, desarrollar y evaluar la interacción persona-ordenador de productos, sistemas, aplicaciones y servicios informáticos.**

El más ambicioso objetivo que se presenta en este documento pasa por el desarrollo y puesta en producción de una aplicación para niños con Trastorno del Espectro Autista (TEA). Esto supone un escenario donde lograr una interacción persona-ordenador adecuada y usable no es sólo una opción. Además, los resultados del proyecto permitirán evaluar de distinta forma cómo los usuarios interactúan con el sistema.

Al ser la competencia probablemente más relevante del proyecto se ha desarrollado, también, un proceso de validación de la interacción de usuarios con el sistema. De esta manera los operadores del sistema podrán decidir si la interacción del sistema es adecuada o no para estos y sus pacientes.

#### **[CE15] Capacidad para la creación y explotación de entornos virtuales, y para la creación, gestión y distribución de contenidos multimedia.**

La aplicación que se ha mencionado previamente consistirá en una colección de minijuegos con el fines de rehabilitación y apoyo, de esta manera se consiguen alinear la CE14 y la CE15. Dentro de este juego hay gran cantidad de contenido multimedia, sobretodo imágenes y audio, que son presentados en un entorno virtual. Esta puede ser la competencia más

<sup>1</sup>Especificación completa en <https://emotiv.gitbook.io/cortex-api/>

## 1. INTRODUCCIÓN

directa que ofrece el trabajo ya que la mayor parte del tiempo de desarrollo ha sido frente a una herramienta de creación de entornos virtuales.

### **1.4 Estructura del documento**

#### **Capítulo 2: Objetivos**

A lo largo del capítulo dos se presentan los objetivos generales y específicos del proyecto que se documenta en este escrito.

#### **Capítulo 3: Antecedentes**

El capítulo de antecedentes explica el estado del arte en el concepto de gamificación, desarrollo de videojuegos multiplataforma, y el diagnóstico e impacto social disfunciones neurológicas en la infancia.

#### **Capítulo 4: Método de trabajo**

Este capítulo explica qué metodologías han hecho posible el desarrollo del proyecto. Además explica cómo varios conceptos de la cultura DevOps están presentes en el mundo de desarrollo de videojuegos.

#### **Capítulo 5: Resultados**

En este capítulo se presenta el producto resultante del proyecto. Se hace especial énfasis en aspectos más técnicos como patrones de diseño utilizados, modelo de datos de las bases de datos usadas o la justificación del uso de las tecnologías frente a otras alternativas.

#### **Capítulo 6: Conclusiones y líneas de trabajo futuro**

Es un capítulo subjetivo donde se plantean varias líneas de trabajo futuro y se valoran las implicaciones personales y profesionales de este proyecto.

## Capítulo 2

# Objetivos

Los objetivos de este proyecto son amplios, haciendo el trabajo más interesante y dinámico. La integración de hardware y software va a jugar una parte importante en el desarrollo del proyecto. La plataforma «RehabiliTEA» consta de varios módulos que solventan diferentes problemas y funcionan de manera casi independiente.

Para conseguir un diseño modular, coherente y versátil ha sido necesario hacer un estudio exhaustivo de los objetivos y tener claro el diseño a lo largo de todo el proceso de desarrollo. Si bien, el objetivo general abarca mucho como para entender la dirección del proyecto, los objetivos específicos hacen más clara y comprensible la arquitectura de este trabajo.

### 2.1 Objetivo general

El objetivo de este proyecto es medir la evolución de un grupo de niños con Trastorno del Espectro Autista (TEA) antes y después de usar una plataforma de gamificación para la rehabilitación. Esta evolución se medirá por medio de la intensidad de las ondas delta, theta, alpha, beta y gamma. El estudio se realizará midiendo el antes y el después de 90 días de uso de la plataforma.

Para llevar a cabo este objetivo será necesario, en primer grado, crear la plataforma. Dicha plataforma deberá desplegarse en un medio accesible a gran parte de la población, teléfonos y tablets son buen candidato, pero la elección del medio forma parte de la solución y la estructura final. Desgraciadamente, debido a la irrupción de una pandemia este año, el estudio queda como trabajo futuro.

### 2.2 Objetivos específicos

En la descripción del objetivo general se ha hablado de una plataforma de rehabilitación de patologías como TEA (Trastorno del Espectro Autista) y de la integración de un dispositivo de medición de ondas cerebrales con el mismo. No obstante, se pueden fraccionar todavía más los objetivos específicos, ya que una plataforma de esta envergadura requiere de sub-proyectos diferentes con sub-objetivos diferentes.

## 2. OBJETIVOS

### **Desarrollar una colección de juegos de rehabilitación.**

Si no el principal objetivo, el más vistoso, es el desarrollo de una aplicación de juegos de desarrollo. Se espera obtener resultados de mejoría de pacientes por medio del uso de técnicas de gamificación para el apoyo y la rehabilitación de pacientes. El fin de estos juegos es el de ofrecer un reto de concentración, comprensión lectora, razonamiento matemático y memoria a corto plazo.

El objetivo es llegar, en la primera versión, a un mínimo de cinco mini-juegos de, como mucho, cinco minutos de duración. Al ser Android la plataforma mayoritaria de dispositivos para juegos será, también, objetivo llegar a ella, bien sea por una aplicación nativa o por medio de aplicaciones web soportadas por este sistema operativo. Esta colección debe registrar la progresión de cada usuario y guardarla para que cuando la sesión acabe poder avanzar desde el punto en el que el usuario lo dejó.

### **Desarrollar una infraestructura de gestión de perfiles**

Con el fin de poder añadir persistencia a la aplicación en los aspectos previamente mencionados, otro subobjetivo es el desarrollo y despliegue de una infraestructura que permita gestionar ese tipo de información. Desde un punto de vista de desarrollo web podemos entender que el juego es el *Frontend* de una aplicación, haciendo necesario llevar a cabo el desarrollo del *Backend*.

Este *backend* debe comunicarse con la colección de juegos, y almacenar, además del progreso, los eventos que el usuario produzca durante el juego. Idealmente ha de desplegarse en una plataforma de alta disponibilidad como puedan ser servicios cloud tales como AWS o MS Azure, para garantizar que cualquier usuario a cualquier hora pueda acceder al contenido de la aplicación y no se pierdan datos.

### **Integrar el espectrómetro en el sistema.**

Se ha mencionado que el objetivo principal es el estudio de las ondas cerebrales de un grupo de niños con TEA por medio de un espectrómetro, paradójicamente, es "la guinda del pastel" de este proyecto, ya que es necesario desarrollar toda la infraestructura hardware y software hasta llegar al punto donde la recolección de estos datos puedan darse.

El dispositivo de espectrometría en este caso es un casco de Emotiv, en concreto el modelo «Insight». El subobjetivo final es el de integrar este dispositivo con los minijuegos y el *backend* para que las sesiones de recolección de datos sean más sencillas y productivas. La información del espectrómetro debe etiquetarse con el perfil de usuario de juego para poder realizar estudios *a posteriori*.

## Capítulo 3

# Antecedentes.

**D**URANTE presente capítulo se hablará del concepto de gamificación y el estado actual del mismo en el mercado. También se tratará el impacto de la gamificación dentro en contextos de apoyo, rehabilitación y desarrollo. Por último, se discurrirá sobre lo que implica la masificación de dispositivos móviles en el contexto de gamificación y juegos de apoyo, así como herramientas para el desarrollo de los mismos.

### **3.1 Disfunciones neurológicas en la infancia**

Los trastornos neurológicos son aquellas patologías que afectan al sistema nervioso central, causando un desempeño anormal de las funciones cerebrales. Estas condiciones afectan a las facetas emocionales, de autocontrol, de memoria y en la habilidad de aprendizaje del paciente durante toda su vida. A día de hoy, están recogidas más de 600 enfermedades de este tipo.

Nombres tan sonados como son la enfermedad de Parkinson y el mal de Alzheimer pertenecen a esta rama de patologías, pero el abanico de condiciones es mucho mayor, incluyendo desórdenes motores, de espectro, de aprendizaje, de comunicación y de hiperactividad, así como discapacidades intelectuales misceláneas. Este documento se centra en los trastornos de déficit de atención y del espectro autista.

La razón para prestar una atención especial a estas dos patologías es la preocupación y concienciación social respecto a su impacto juvenil, amén de su elevado número de casos entre la población. Es posible ver, hoy más que nunca, asociaciones de personas y campañas de sensibilización y desestigmatización de estas condiciones, especialmente en entornos infantiles y juveniles. A continuación se definirán ambas disfunciones siguiendo el libro [A<sup>+</sup>13], una de las principales autoridades en el estudio de trastornos mentales.

#### **3.1.1 Trastorno de déficit de atención e hiperactividad**

El trastorno de déficit de atención e hiperactividad, ADHD, por sus siglas en inglés afecta, de forma aproximada, a un 5 % de la población en edad escolar según el DSM-V [A<sup>+</sup>13]. Un error común de esta patología es confundir la personalidad más nerviosa y activa de los niños con él, aún siendo tres veces más propensos a este tipo de condiciones que los adultos,

### 3. ANTECEDENTES.

el periodo que dura la excitación debe ser muy prolongado para poder confirmar la aparición de la patología.

Esta patología se caracteriza por un patrón de falta de atención o por episodios de impulsividad hiperactiva que no son propios al carácter del individuo. Síntomas del ADHD pueden ser la incapacidad de organizar una agenda y seguirla con el fin de desarrollar una rutina estable y normalizada así como la pérdida de interés y concentración frecuente cuando se realiza una actividad o, en caso de adultos, una casi incontrolable sensación de inquietud.

Las causas de este desorden no están muy claras, pero parecen estar estrechamente relacionadas con la genética. Es también mucho más común en varones que en mujeres, con un ratio 2:1 en caso de niños y 1.6:1 en periodo de adulto. Un peso anormalmente bajo durante el nacimiento triplica las posibilidades de contraer ADHD, asimismo, un parto prematuro aumenta, también la probabilidad de suceso de este trastorno en torno a un 20-30 %.

Afortunadamente el ADHD ha sido objeto de estudio, descubriendo métodos y técnicas para que los pacientes, tanto niños como adultos pueden aprender a controlar sus síntomas y evitar que el trastorno se haga más notable a lo largo de la vida del paciente. En caso de no ser tratado, este trastorno puede degenerar en trastornos de conducta o trastornos antisociales. No obstante, en rangos generales, los pacientes de ADHD sufren de una mayor dificultad en entornos escolares y laborales.

#### 3.1.2 Trastorno del espectro autista

El Trastorno del Espectro Autista (TEA) o ASD (Autism Spectrum Disorder), es otra patología neurológica caracterizada por un déficit de comunicación e interacción social persistente o patrones de conducta, interés o comportamiento repetitivos. Estos síntomas suelen presentarse antes de los tres años, comúnmente como consecuencia de la prematuridad. Un paciente de Trastorno del Espectro Autista (TEA) presenta, como norma general, dificultades a la hora de interpretar mensajes en una comunicación social.

El TEA forma parte de los trastornos generalizados de desarrollo y sus síntomas pueden presentarse a partir de los 12 meses de vida, pero la edad media de aparición de estos síntomas puede variar en función del tipo de trastorno. El manual de diagnóstico y estadístico de trastornos mentales [A<sup>+</sup>13], ya citado, identifica cinco grupos o tipos dentro de los trastornos generalizados del Desarrollo.

- **Trastorno del Espectro Autista (TEA):** Es un trastorno neurológico que, generalmente, dura toda la vida y daña las capacidades sociales y comunicativas de los pacientes. El que este grupo esté categorizado se debe a que en anteriores ediciones del DSM-V todos los trastornos generalizados del desarrollo estaban clasificados como autismo.
- **Síndrome de Rett:** La particularidad de este trastorno es que, mientras el resto de síndromes son cuatro veces más frecuentes en hombres que en mujeres, el síndrome de

Rett se presenta, casi con total exclusividad, en niñas. También se diferencia del resto por su carácter regresivo, además de impedir el desarrollo destruye las habilidades de comunicación, motoras y sociales de la niña, además de afectar al desarrollo muscular.

- **Síndrome de Altas capacidades:** Este trastorno está caracterizado por la dificultad de interacción social, rutinas obsesivas, y patrones de conducta peculiares. Los Ásperger tienen gran capacidad para tareas de memoria, matemáticas y ciencia. Los síntomas de este síndrome suelen durar toda la vida pero pueden ser disminuidos con detección precoz y tratamiento.
- **Síndrome de Heller:** El trastorno de Heller es parecido al síndrome de Rett en sus síntomas, no obstante, se manifiesta un tanto más tarde, a partir de los dos años, y afecta más a varones que a mujeres. El niño pierde, en un periodo de hasta ocho años, las habilidades sociales, comunicativas y motoras. Además de esta vuelta atrás, los niños empiezan a desarrollar síntomas propios del TEA a lo largo de los siguientes años.
- **No especificado:** Existe una clasificación llamada «Autismo atípico» que comúnmente es confundida con los trastornos del desarrollo no especificados. A esta categoría pertenecen aquellos trastornos que no pertenezcan a los síndromes anteriormente expuestos y, además, afecten a áreas más allá de las habilidades lingüísticas, las relaciones sociales y las conductas e intereses. Esta categoría es un ejemplo de «cajón desastre», lo que no se consigue clasificar se destina aquí.

Hoy por hoy se desconocen las causas de estos trastornos, aunque la genética y las complicaciones en el parto han mostrado estar relacionadas con este tipo de patologías.

### 3.2 Gamificación y tratamiento médico

Definir gamificación es una tarea complicada puesto que, dependiendo del contexto y la visión del concepto, este puede cambiar. Es posible encontrar dos acepciones de gamificación en [HKS14]. La definición más *ad hoc* del término tiene que ver con la inclusión de mecánicas propias de un videojuego en todo tipo de aplicaciones con el fin de lograr una interacción con más motivación por parte del usuario.

Por otro lado, en un contexto de explosión de la era digital como es el actual, la gamificación puede entenderse como un elemento transformador al que entidades de todo tipo acabarán acudiendo. El escepticismo frente a este tipo de pensamiento existe y la principal oposición a la gamificación es tal porque consideran que en el *Hype cycle* de Garter [O’L08], la gamificación ocupa lugar en la escalada al pico de expectativas irreales en lugar de ser una medida productiva.

En contraposición a estos detractores, Garter, la misma revista que introdujo el modelo de ciclo de *hype*, considera que la gamificación será aceptada por el público mayoritario en un

### 3. ANTECEDENTES.

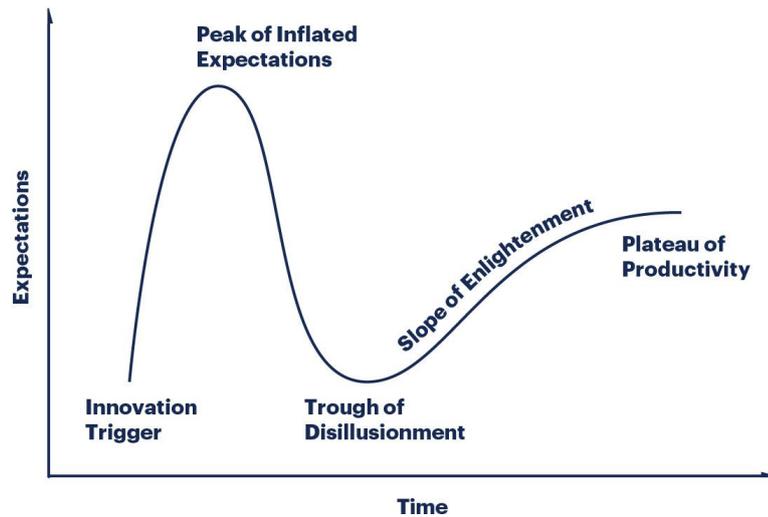


Figura 3.1: *Hype Cycle* de Gartner

periodo de dos a cinco años [Smi19]. También se posiciona a la gamificación como parte de la sección *Climbing the Slope* dentro del ciclo de *hype* de las tecnologías para trabajadores de cara al público.

Si no a raíz del concepto de gamificación, paralelamente a él surgía el término juegos serios, o *Serious Games*. Al igual que con la gamificación, la acepción varía con el contexto, pero la mayoría de definiciones pasan por definir estos juegos serios como juegos cuyo propósito u objetivo principal no es el lúdico.

Algunos autores como [AD<sup>+</sup>11] o [BAAH13] clasifican estos juegos según el propósito de los mismos. Otros ejemplos como [LEES14] hacen un análisis más exhaustivo de estos tipos, no obstante, la idea general de como el mercado ve los juegos serios es más cercana a las dos primeras. A grandes rasgos, los *Serious Games* se relacionan, sobretodo, con el *eLearning*, pero pueden usarse, también, a modo de anuncio, en este caso se les suele llamar *advergames* o, en última instancia con juegos destinados a mantener a sus usuarios saludables, estos últimos reciben el nombre de *exergames*.

En [DAJR11] se explica como es posible que el videojuego más *tradicional* puede haber surgido como evolución de programas de aprendizaje que bien podrían considerarse juegos serios. Sea como fuere, en 2002 empezaron a ser populares este tipo de prácticas, aunque no fueran los pioneros. A día de hoy el valor de mercado de los juegos serios se estima en miles de millones de dólares y las expectativas de mercado crecen a cada año.

Es posible encontrar ejemplos de uso de videojuegos con fines de apoyo y rehabilitación en [RMTMPMR14]. En este caso, el uso de videojuegos para tratar trastornos de déficit de atención es adecuado, ya que usuario no percibe que este esfuerzo cognitivo como algo negativo. Estos juegos también ayudan a acabar con la adicción a videojuegos que muchos pacientes de ADHD sufren.

Estudios como [PPRH17] o [SEZ<sup>+</sup>16] confirman que la utilización de aplicaciones informáticas impacta de manera muy positiva en pacientes con déficit de atención. Estas aplicaciones mejoran la situación de los pacientes, llegando a valorar la idea de eliminar los fármacos para el tratamiento de esta condición, debido a que el tratamiento por medio de juegos produce efectos secundarios menos invasivos que los químicos.

Otras patologías como el Trastorno del Espectro Autista (TEA) también pueden ser tratadas con medios digitales. Ejemplos de aplicaciones para el tratamiento del TEA pueden ser encontrados en [VSD<sup>+</sup>19], [PWV<sup>+</sup>19], [BML<sup>+</sup>19] y [YBK<sup>+</sup>19]. Muchos de ellos divergen en el aspecto del trastorno a tratar, lo cual muestra el amplio abanico de mejoras que este tipo de aplicaciones pueden proporcionar.

En primer lugar, ha habido aplicaciones de realidad aumentada, apoyadas en el uso de *wearables*, las cuales han mostrado ser efectivas para apoyar a pacientes de TEA a entablar lazos afectivos. Estos sistemas usan dispositivos ofrecen un refuerzo sensorial en forma de emoticono, color o sonido en función de la expresión facial de la persona, ayudando al paciente a diferenciar estados de ánimo y facilitando la socialización.

Por otro lado, la ubicuidad de los sensores y dispositivos móviles a día de hoy hacen posible la recolección de información de forma masiva. Aunque este conocimiento no implique tratamiento en el sentido más estricto del término, todos estos grandes volúmenes de datos pueden ser estudiados con técnicas modernas de analítica de datos y encontrar mejores soluciones de tratamiento.

Ejemplos de estas técnicas de estudio de la información pueden ser la de intentar encontrar patrones comunes entre los diferentes tipos de TEA, como se propone en [LLBC19]. No obstante, la idea de heterogeneizar aún más los grupos podría, también, llegar a ser interesante, puesto que se podrían personalizar aún más las interacciones, actividades y mecánicas de grupo en terapias dirigidas a estos grupos.

No obstante, aunque solo se haya hablado de gamificación para patologías neurológicas, la rehabilitación física es, también, un hito de la gamificación. El sistema de rehabilitación basada en gamificación más conocido puede ser «REHABILITY». REHABILITY es una plataforma de juegos de rehabilitación para supervivientes de infartos cerebrales y pacientes de Parkinson o esclerosis múltiple.

Este ejemplo plantea una solución para el desempeño de ejercicio modera desde cualquier parte. Probablemente el factor más importante de REHABILITY sea que es un sistema comercial. Esto convierte a REHABILITY el primer sistema cuya viabilidad económica ha sido probada, mostrando que hay mercado para este tipo de desarrollos y soluciones.

Otros sistemas de rehabilitación física basados en gamificación no radican en el movimiento sino en el estímulo que un videojuego es capaz de provocar. Estudios como [BMC<sup>+</sup>09] o [BMC<sup>+</sup>10] muestran como aplicaciones con tecnologías modernas como la realidad au-

### 3. ANTECEDENTES.



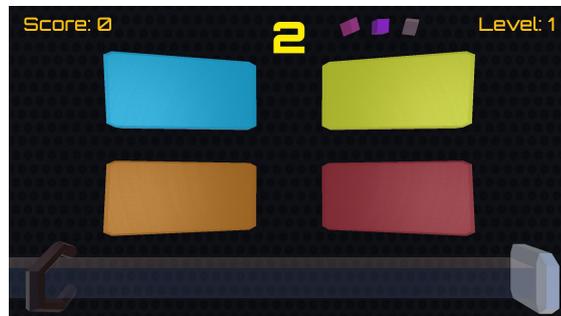
Figura 3.2: Imagen promocional de REHABILITY.

mentada e interacción corporal consiguen motivar a pacientes en rehabilitación de una forma mucho más efectiva que la rehabilitación tradicional.

Por desgracia, la memoria a corto plazo no es uno de los temas más socorridos en cuando a tratamiento de TEA. Se ha podido ver que el principal uso de gamificación en este aspecto es para facilitar la interacción social. No obstante existen sistemas de entrenamiento de memoria a corto plazo, aunque la mayoría son colecciones de juegos de memoria clásicos como «Simon» o «Simón dice» y «Memory».



(a) Memory



(b) Simon dice.

Figura 3.3: Juegos comúnmente usados para mejorar la atención y memoria a corto plazo.

### 3.3 Desarrollo para plataformas móviles

Hoy por hoy se puede decir que acceder a dispositivos móviles es más fácil que nunca. La inclusión de los *smartphones* ha supuesto un antes y un después en lo que a tecnología respecta. Estudios como [Kak14] citan a lgunos datos sobre la penetración de esta tecnología en la vida diaria. El índice de penetración de esta tecnología en España en 2013 era del 55 %, 20 puntos superior a la misma cifra en 2011.

Medios más actuales como el artículo [Che20] estiman que más del 32 % de la población del mundo tiene o ha tenido acceso a un *smartphone*. La proliferación de esta tecnología es

tal, muy probablemente, debido a la integración con internet y la posibilidad que abre para estar conectado a la red de redes en cualquier parte a cualquier hora.

No obstante, desde un punto de vista más cercano a la explotación de tecnologías, la carga de sensores de estos dispositivos resulta muy interesante. Si bien los sensores dentro de los teléfonos inteligentes no son los más precisos que se puedan encontrar, la ubicuidad de los terminales y la conectividad de los mismos hace que el volumen de mediciones no solo pueda dispararse.

El desarrollo para estos dispositivos puede variar según el fabricante. La principal razón de esto es la amplia variedad de procesadores (y arquitecturas) móviles. En la actualidad la familia Snapdragon de Qualcomm y la familia Helio de MediaTek han conseguido estandarizar, de un modo laxo, la oferta de CPU para móviles. Sin embargo, la oferta de sistemas operativos apenas ha variado desde que estos dispositivos se popularizaran.

### **Fabricantes y pilas de desarrollo**

Se le atribuye a la compañía «Apple» la proliferación de los teléfonos inteligentes cuando lanzó al mercado el producto conocido como «iPhone» en 2007. Todos los terminales de esta compañía utilizan un sistema operativo conocido como iOS (iPhone Operating System). La manera en la que Apple resuelve su pila de desarrollo tiende a ser con un único lenguaje, el cual es portable a todos los dispositivos de la marca. Este *lenguaje único* ha cambiado a lo largo de los años, el más antiguo siendo «Objective-C», hoy deprecado en favor de «Swift».

En la página web «stackshare», el perfil oficial de Apple, <https://stackshare.io/apple/apple>, lista todas sus tecnologías de desarrollo, utilidades, y herramientas de integración y entrega continua. Esta pila de desarrollo se resume en los lenguajes anteriormente mencionados, el IDE Xcode y el entorno de construcción «Cocoa Touch».

El principal y, prácticamente, único competidor de iOS en cuanto a sistemas operativos para terminales móviles es Android. No obstante, definir Android como competidor de iOS es no hacer justicia a todo lo que es y lo que abarca, ya que es el sistema operativo más extendido del mundo. Se estima que en 2017 casi el 38 % de los equipos con conexión a internet operaban bajo Android[Sta19].

La historia de Android se remonta a 2003 con la fundación de «Android Inc.». En 2005 esta entidad sería comprada por Google, con quien se presentaría el proyecto al gran público en 2007 y lanzarían su primer dispositivo comercial en 2008. Este primer terminal se llama «T-Mobile G1» o «HTC Dream» y fue comercializado por la empresa «HTC», actual partner de «Valve» en el desarrollo de dispositivos de realidad virtual.

A nivel técnico Android es un SO basado en el kernel de Linux. Las primeras versiones del mismo estaban basadas en la virtualización, pero esta característica desapareció en la

### 3. ANTECEDENTES.

versión 6.0, «Marshmallow», en 2015, haciendo el SO completamente nativo. A día de hoy, Android es mantenido por Google y un consorcio de las principales empresas de telefonía móvil llamado «Open Handset Alliance».

La API que ofrece Android para sus desarrolladores está basada en Java, lenguaje conocido por ser portable aún después de ser compilado. En 2009 fue lanzada la primera versión del Kit de desarrollo nativo (NDK) de Android, proporcionando, entre otras herramientas, un compilador de C++ para Android. Por último, en 2016, la versión 1.0 de «Kotlin» tuvo su lanzamiento. Este lenguaje se integra con la Máquina Virtual de Java (JVM) y puede compilar a otros lenguajes como JavaScript o ensamblador. Pese a aceptar estos lenguajes, y alguno más, toda aplicación de Android necesita de código Java.

El principal entorno de desarrollo para Android es el Entorno de desarrollo integrado (IDE) «Android Studio». Este IDE es distribuido de forma gratuita para aplicaciones no comerciales y funciona en las plataformas Windows, Linux, MacOS y ChromeOS. Junto a Android Studio se instalarán tanto librerías como herramientas de construcción para desarrollar cualquier tipo de aplicación Android.

#### **Motores de juegos**

En lo que compete a juegos, Android no es desconocido, la explosión de demanda de videojuegos de los últimos años se debe, en gran parte, a las plataformas móviles. Tanto Android como iOS soportan la versión para sistemas empujados de OpenGL, OpenGL ES, de hecho, Android soporta, inclusive, todas las versiones de Vulkan. Para desarrollar aplicaciones gráficas interactivas, estas librerías y el conjunto de llamadas al sistema es suficiente, no obstante se suele añadir una capa más de abstracción.

Esta capa de abstracción que une herramientas y entornos de desarrollo con el fin de desarrollar aplicaciones gráficas en tiempo real las llamamos motores gráficos o motores de juego. Un error común es confundir el motor de juegos con el editor de mapas, ya que, a día de hoy, la mayoría de motores comerciales incluyen un editor de mapas consigo. El motor del juego es más, aún, toda la infraestructura y subsistemas software que soportan el desarrollo y despliegue del juego.

En la actualidad, es posible encontrar ejemplos de motores, tanto comerciales, como libres, que llegan a un nivel de calidad y madurez muy alto. Estos motores disponen de grandes equipos de desarrollo que centran su actividad laboral en, tan sólo, actualizar y corregir el motor. A continuación se hablará sobre cuatro motores gráficos los cuales, a la fecha de redacción de este documento, pueden ser instalados de forma gratuita y soportan despliegue en dispositivos móviles.

## Cocos2d

Cocos2d es la librería de gráficos para móviles más extendida del mundo. Dispone de un editor de niveles llamado «Cocos Creator» y dispone de Interfaz de Programación de Aplicaciones (API) para los lenguajes: C++, Lua, JavaScript, TypeScript, C#, Python y Swift. La librería Cocos es parecida a Ogre3d a nivel de desarrollo, no obstante, la dirección es casi opuesta. Esto se debe a que Ogre, aspira a ofrecer herramientas de renderizado realista mientras que Cocos se centra en desarrollar subsistemas de juego.

Pese a no ser tan reconocible como otros motores, Cocos2d dispone de una pila de juegos bastante aceptable en su elenco, más aún, teniendo en cuenta que la versión que soportaba plataformas móviles de este motor salió en 2011. Lo más resaltable de estos juegos es que gran parte de los juegos hechos con Cocos han llegado a ser recomendados en la *Android Store*, ejemplos de estos juegos son: *Don't tap the white tile*, *Clash of Lords 2* y *Family Guy: The Quest For Stuff*.

## Godot Engine

Godot es la gran sorpresa de los motores de la era moderna. Fue lanzado al mercado en 2014 y, a día de hoy, es el tercer motor gráfico más popular, superando a motores como CryEngine o Source. Este *sorpasso* se debe, principalmente a que Godot se enfoca en la productividad del desarrollador y plataformas móviles en lugar de sistema de gameplay muy complejo, como Source, o, un motor de renderizado muy realista, como CryEngine.

Este motor es libre y su código fuente puede encontrarse en github. El editor de niveles está integrado en el motor y, para los desarrolladores, Godot ofrece *bindings* para programar con C++, C#, GDScript (Godot runtime para JavaScript) y VisualScript. Este último lenguaje es el más interesante de todos, ya que la programación visual es muy socorrida en desarrollo de videojuegos, ya que perfiles no técnicos pueden entenderla.

## Unreal Engine 4

Si bien, Unreal Engine 4 salió al mercado el mismo año que Godot, 2014, es el motor con más historia de todos los que se mencionan en este documento. Unreal Engine '1' no existe como motor, Unreal fue un juego lanzado en 1998, fue en 2002 donde se lanzaría como programa *standalone* Unreal Engine 2. En 2006, Unreal Engine 3 sale al mercado y en 2010 sale «Unreal Development Kit», la versión *freeware* del mismo y, por último, en 2014 la versión 4. A día de hoy el código del motor es libre, pero la licencia de publicación no, y es mantenido por Epic y es actualizado de forma regular.

Unreal Engine 4 es, actualmente, el motor más popular y más solicitado en entornos laborales. No obstante, su curva de aprendizaje es la más pronunciada de todos los motores. Los desarrolladores pueden programar utilizando C++ y *Blueprints*. Blueprints es otro lenguaje de programación visual, similar al de Godot. Unreal Engine tiene integrado el editor de ni-

### 3. ANTECEDENTES.

veles y un IDE de Blueprints, asimismo, trabaje en tándem con Visual Studio para depurar y perfilar (*profiling*) nuestros juegos.

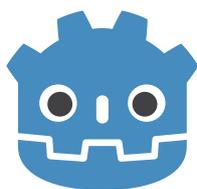
Este motor se centra en ofrecer herramientas de renderizado realista en tiempo real, fue pues, uno de los primeros motores en soportar Ray Tracing. La cantidad de juegos comerciales que es posible nombrar hechos en este motor es gigantesca, no obstante, en plataformas móviles esta cifra no es tan grande. Aún así es posible encontrar *ports* para Android de juegos reconocidos como *Mortal Kombat* o *Life is Strange*.

#### Unity

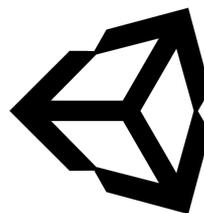
Unity, si bien, no es el motor más demandado, es, sin duda, el mejor valorado de todos los motores que se han nombrado. A día de hoy su comunidad es numerosa, activa y multidisciplinar. Originalmente, Unity, nace de un juego para web fallido, pero fue capaz de reinventarse como un motor para usuarios de Mac. En 2006 tendríamos la primera versión de Unity, y en 2016 este motor pasaría de lanzar versiones mayores a un proceso de actualización continua con versiones long term support (LTS) y versiones menores.

La naturaleza de Unity es la de «democratizar» el desarrollo de videojuegos, es decir, ofrecer herramientas sencillas y potentes para desarrollos con la mayor ubicuidad posible. El motor se caracteriza por soportar despliegues en prácticamente cualquier dispositivo con pantalla y la programación se hace por medio de C# y UnityScript, un entorno de runtime de JavaScript para Unity, similar al de Godot.

Es posible encontrar ejemplos de juegos comerciales para todas las plataformas existentes, de hecho, se estima que la mitad de los juegos que se hacen hoy por hoy utilizan Unity. Juegos como *Angry Birds*, *Pokémon Go* y *Hearthstone: Heroes of Warcraft* han contribuido a dotar al motor de gran renombre así como convertirse en auténticos éxitos comerciales.



(a) Godot



(b) Unity



(c) Unreal Engine



(d) Cocos.

Figura 3.4: Logos de los principales motores gráficos.

## Capítulo 4

# Método de trabajo

LA metodología y planificación de un proyecto software pueden llegar a ser tan o más importantes como el desarrollo del mismo. En un plan de estudios que espera formar futuros directivos está especialmente merecida la atención a la metodología y a la planificación de proyectos.

Se ha seguido el concepto de **Kaizen** (cambio para bien) [MW12] a la hora de trabajar con el fin de acercar la cultura DevOps[EGHS16] al mundo del videojuego. Conceptos propios de DevOps como CI/CD (integración y entrega continua) pueden ser aplicados a muchos proyectos de desarrollo de videojuegos, pero es importante saber cuando son necesarios y cuando van a aportar valor.

### 4.1 Metodología de trabajo y desarrollo

La metodología utilizada está basada en el manifiesto ágil y podría considerarse una versión reducida de Scrum[SB02]. Lo que podría considerarse Scrum "tradicional" se plantea demasiado complejo para un equipo de un sólo desarrollador, ya que la recolección de los artefactos Scrum y la documentación de todos los Backlogs de cada sprint entorpecerían mucho la tarea de desarrollo.

En contraposición, se ha propuesto un modelo de desarrollo basado en Sprints semanales, en los cuales se presentan las novedades del producto para someterlas a revisión. Una vez al mes, tras esta reunión se ha hecho una prueba de validación, en este proceso se ha provisto al cliente, en este caso evaluador, una versión del sistema construida como en despliegue. El cliente tiene instrucciones de intentar romper el sistema una vez acaba la evaluación de los requisitos que considera importantes.

La ingeniería del caos [BBDR<sup>+</sup>16] es un concepto ligado a DevOps [DPvH18] especialmente importante cuando se trabaja con marcos de trabajo con un alcance tan grande como es un motor de juegos<sup>1</sup>. Es virtualmente imposible que un desarrollador conozca todas las vulnerabilidades de un motor, es por esto que se ha prestado bastante atención a este ámbito, ya que es crítico encontrar las vulnerabilidades antes del paso a producción o despliegue.

---

<sup>1</sup><https://www.perforce.com/blog/kw/unreal-engine-and-static-analysis>

## 4. MÉTODO DE TRABAJO

Se ha hecho la planificación con tan sólo dos entornos en mente, uno de desarrollo y otro de producción/despliegue. Este último entorno en inglés suele denominarse *shipment*, debido a que hace tiempo era necesario producir los discos y enviarlos. No obstante, este proceso de *shipment* en la era digital puede resolverse por medio de plataformas de distribución como «Google Play» o «App Store»

Esto se relaciona con la idea de un despliegue único, es decir, tan sólo se lanzará la aplicación a producción una vez, a menos que se descubra una falta de seguridad grave. Es por esto que se ha prescindido de cualquier tipo de pipeline de DevOps, no es necesaria la automatización de algo que sólo se va a hacer una vez.

Por último, para asignar la prioridad de las tareas se empezó usando el método «MoS-CoW» [Wat09]; tendrá, debe tener, podría tener, no tendrá; pero, pronto mutó a una clasificación binaria. Dicho de otra manera, una característica es descartada o aceptada en el acto. Las características aceptadas tienen todas la misma prioridad salvo en caso de dependencias.

### 4.2 Planificación y distribución del trabajo

#### Noviembre 2019 - Enero 2020

La primera parte del proyecto estuvo destinada a investigar sobre los dispositivos y la integración de estos con una posible plataforma. El trabajo de esta parte está vinculado a las tareas de investigación antes que desarrollo, recopilando ideas sobre cómo debería ser el sistema final.

#### Enero - Marzo, 2020

En enero llega el casco, sin embargo, el principal esfuerzo es el de crear el backend de la aplicación. En esta etapa se desarrolla un primer prototipo de infraestructura web y se despliega en Azure para estimar los costes.

#### Marzo - Mayo, 2020

Los siguientes meses estuvieron dedicados al desarrollo de juegos que estarían en el prototipo final. También se realizaron pruebas de integración con el sensor de espectrometría, añadiéndolo a la infraestructura web.

#### Mayo - Septiembre, 2020

Los últimos meses consistieron en preparar el despliegue de la aplicación para poder ponerla en marcha a principios del curso escolar. El trabajo de esta parte consistió en desplegar la infraestructura web completa, exportar la colección de juegos a la plataforma final y, preparar perfiles para los usuarios.

## Resultados

LA arquitectura de la solución para este proyecto se plantea un tanto compleja. Es posible encontrar hasta cuatro dispositivos trabajando a la vez en una sesión de recolección de datos. La estructura en base a microservicios, pese a solucionar problemas de compatibilidad y escalabilidad, tiende a necesitar gran cantidad de artefactos para resolver cualquier problema.

El fruto de este trabajo puede encontrarse en dos repositorios distintos, uno alberga la aplicación móvil, <https://github.com/RoberPlaza/RehabiliTEA.App>, y otro la infraestructura web, <https://github.com/RoberPlaza/RehabiliTEA.WebService>.

### 5.1 Arquitectura general

A lo largo del segundo capítulo se presentaron tres subobjetivos claramente diferenciados, un backend en base REST para la gestión de usuarios, un frontend, en este caso en formato juego que use el backend y un sistema de recolección de datos con el sensor. También se ha hablado de la API del sensor, recordemos, implementa json-rpc sobre websocket en el puerto 6868 de un dispositivo con el demonio de emotiv instalado.

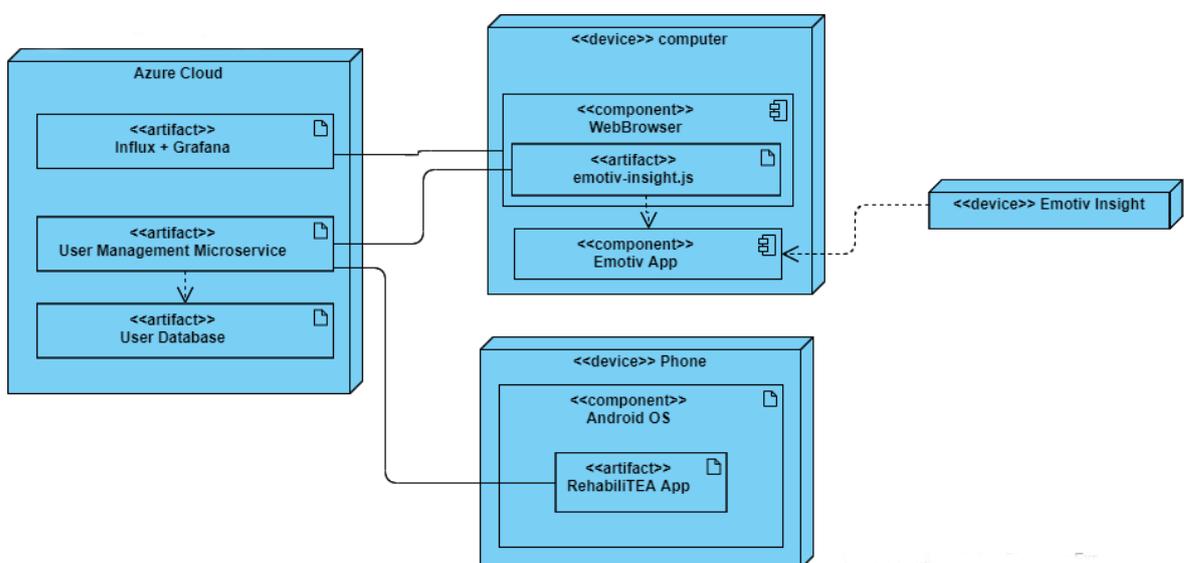


Figura 5.1: Diagrama de despliegue de RehabiliTEA.

### **Recolección de datos**

Debido a la flexibilidad de esta interfaz seleccionar las herramientas adecuadas es una dificultad añadida. En el caso del estereotipo expuesto se ha utilizado el lenguaje JavaScript. Un servidor web puede servir contenido HTML con JavaScript embebido en él, haciendo que el usuario que quiera recolectar datos no necesite instalar más programas en su ordenador. También se consigue de esta manera aumentar la portabilidad del programa de recolección de datos.

Este programa en formato web de recolección de datos puede servirse como parte del backend web de la aplicación y sólo será usado por computadores con Emotiv App instalado y el casco sincronizado. Las llamadas de JavaScript utilizadas forman todas parte de la librería estándar con el fin de lograr la máxima portabilidad posible. No obstante, se ha desarrollado y depurado usando el runtime de JavaScript V8, del proyecto Chromium.

Tanto el servidor que gestión de usuarios (junto con su base de datos) como el tándem Grafana e InfluxDB forman parte de una infraestructura desplegada en Azure. Esta infraestructura puede encontrarse en forma de código en el archivo «docker-compose.yml» del repositorio. El despliegue de esta infraestructura ha sido orquestado con Docker Compose, ya que Azure cuenta con un servicio para el despliegue de este tipo de infraestructuras y es un sistema de orquestación sencillo, pero suficientemente potente para este escenario.

InfluxDB es una base de datos pensada para series temporales, es usada para almacenar la información generada en las sesiones con el casco. Grafana, por otro lado, es una herramienta de visualización de datos en formato web. Grafana es capaz de conectar con diferentes tipos de bases de datos y generar gráficos a tiempo real con la información en ellas. Se usan principalmente para monitorización de hardware y servicios, pero también tienen uso en entornos de IOT o en base a sensores.

El fichero JavaScript que recolecta información del casco es el que hace las inserciones en la base de datos de InfluxDB. Estas inserciones tienen un ID de usuario asignado, no obstante, es necesario que el operador introduzca este id, ya que estos dos sistemas no tienen interacción entre ellos. Separar esta información, aunque aparentemente innecesario, es útil puesto que se delimita el alcance de cada microservicio.

### **Gestión de usuarios**

El sistema de gestión de usuarios es un proyecto web de Python el cual, ofrece, por medio de una API REST, acceso a una base de datos PostgreSQL. Esto se ha hecho usando la librería «Flask» y «SQLAlchemy». La primera de estas librerías, Flask, es un microframework de desarrollo web WSGI, por su parte, SQLAlchemy es un ORM (Object-Relational Mapper). Ambas librerías son interesantes juntas ya que SQLAlchemy permite a Flask implementar de forma mucho más sencilla un patrón arquitectural MVC.

Para llegar a producción con estas herramientas se ha montado un proxy inverso usando el plugin «Gunicorn» con el servidor web «Nginx». A diferencia de otros marcos de trabajo, la especificación Web Server Gateway Interface (WSGI) requiere de un servidor web para poder servir la aplicación desarrollada. En esta caso se ha optado por Nginx puesto que es la opción más ligera y sencilla, con el plugin de Gunicorn es posible hacer que un servidor Nginx atienda la petición HTTP genérica y redirija a la aplicación python sólo la información que necesita. También puede actuar como balanceador de carga si se quisiera lanzar varias instancias de la aplicación bajo un mismo endpoint.

### **Aplicación Android**

Por último, la aplicación RehabiliTEA para plataformas Android consiste en un Android application package (APK) con la plataforma. Hace uso del servicio de gestión de usuarios para consultar y guardar la información sobre el usuario cada vez que se lanza el juego y se desbloquea un nivel de dificultad. Es una aplicación desarrollada utilizando el motor «Unity», con el lenguaje de programación «C#». Para motivos de depuración se ha construido la aplicación también para Windows ya que la herramienta de construcción del motor permite llevar a cabo esta tarea con tan sólo un click.

No se han usado plataformas de distribución como puedan ser la Play Store o la App Store para el estudio preliminar y las sesiones de recolección de datos. Esto se debe a que el proceso de aprovisionamiento es necesario que se haga de forma manual para configurar el perfil. Una configuración automática del mismo podría ser mas práctica, pero podríamos estar recolectando más información de la necesaria tanto a nivel tanto técnico como ético.

Se ha hablado de la posibilidad de que una vez acabado el experimento la aplicación pueda quedarse alojada en la Play Store de Android para ser usada por cualquier persona que desee. Para esta versión la aplicación sería *standalone*, prescindiendo del resto de arquitectura web y de los servicios ligados a la misma.

## **5.2 Técnicas de gamificación**

Una parte importante del proyecto es la de crear una serie de actividades atractivas para el usuario con el fin de camuflar que se están realizando ejercicios de concentración y cálculo. Las técnicas utilizadas para crear un vínculo con el usuario son variadas, pero casi todas tienen que ver con ofrecer un nivel de estímulos sensoriales mayor al que se pueda conseguir con actividades no digitales.

### **Estímulo visual constante**

Todos los minijuegos disponen de un arte conceptual minimalista pero reconocible. La mayoría se basan en reconocimientos de formas y figuras, para ello es necesario que estas formas y figuras se muestren como interactivas. Los colores son brillantes para captar la aten-

## 5. RESULTADOS

ción del jugador lo máximo posible, las figuras son simples para hacerlas más reconocibles y, en general, se busca que el juego esté lo menos vacío posible.

Dicho de otra forma, el diseño del arte conceptual espera que este juego sea divertido tan sólo de ver. También se confía en que este arte minimalista ayude al jugador a entender cómo interactuar con el juego, presentando el menor número de interfaces posibles con el fin de no distraer demasiado al jugador de la tarea a realizar.

### **Respuesta inmediata a las interacciones**

En relación con el punto previo, con el fin de resultar más atractiva al usuario, la aplicación debe dejar claro al usuario cuando este ha realizado una interacción con el juego. En este caso todas las interacciones tienen una respuesta instantánea y muchas de ellas se sirven de algún tipo de respuesta sonora, además de la visual, para estimular al jugador y hacerle entender que está interactuando con el juego.

Algunos minijuegos gozan de respuesta extra en función de la naturaleza del mismo. Por ejemplo, al explotar una bomba se la escuche explotando o que un piano tenga sonidos con sus notas. Esta técnica ayuda a que el juego se sienta más real y, en algunos casos, es posible responder a ciertas interacciones con estímulos más o menos potentes que los que se recibirían si el ejercicio no estuviese digitalizado, haciendo posible la alteración del ciclo de recompensa.

### **Dificultad ajustable**

Uno de los mayores problemas en el diseño de videojuegos hoy por hoy, no sólo en los juegos serios, es el diseño de dificultad. Una actividad muy fácil puede causar desinterés a los pocos minutos debido a la falta de desafío, generando una sensación de falta de propósito. Por otro lado, una actividad excesivamente compleja puede hacer que el usuario se sienta mal consigo mismo y frustrarle hasta hacer que desista, siendo esta una alternativa incluso peor.

El diseño original estaba pensado con una configuración muy dinámica de dificultades, pero el prototipo consta de minijuegos con tres niveles de dificultad. Estos niveles son los clásicos fácil, medio y difícil. Una vez el usuario termina un nivel sin cometer un número dado de fallos desbloquea el siguiente nivel de dificultad. Este número de fallos depende del minijuego y, a grandes rasgos, es bastante permisivo.

Esta técnica consigue alargar el posible interés del usuario por medio de ofrecer más contenido a modo de nuevos desafíos, potencialmente alargando la vida útil del producto hasta tres veces el valor original. También puede ser interesante el factor de progreso y recompensa de la aplicación. Un usuario puede notar cómo se va haciendo mejor en una tarea si en un intento no consigue pasar de dificultad pero, al volver después de cierto tiempo, consigue pasar la pantalla.

### **Actividades atractivas, motivantes y gratificantes**

Aunque caiga en un plano subjetivo, que el conjunto de actividades sea atractivo para el usuario es siempre importante. Normalmente se buscan actividades coloquiales, para que causar el menor rechazo posible al usuario, pero esta actividad también ha de proponer un desafío, debe hacer al usuario aprender algo o entrenar algún ámbito personal. Si la actividad es muy coloquial el jugador se dedicará a repetir lo que ya sabe, si es muy ajena opondrá resistencia.

En este caso el diseño de los juegos puede recordar a actividades cotidianas o juegos clásicos, pero las actividades son suficientemente complejas como para ofrecer un estímulo al jugador. Los objetivos de estas actividades, por otro lado, también están relacionados con los usuarios y sus vidas y al ofrecer cantidad de estímulos positivos cuando algo se hace bien se espera que el usuario quiera volver al día siguiente.

Por último cuando un usuario desempeña por debajo de lo esperado en una actividad nunca se le dice de forma explícita que ha fallado. En lugar de un mensaje abiertamente negativos o de error, tanto sonoros como visuales, al usuario se le anima a volver a intentarlo, tanto en la sesión actual como en futuras sesiones. Se espera que esto consiga no desanimar al usuario cuando no puede alcanzar las exigencias del minijuego.

## **5.3 Esquema de medición de métricas**

El sistema de recolección de métricas del proyecto está soportado sobre una infraestructura web que tiene como servidor principal un servicio de la librería «Flask» de Python. La base de datos es un microservicio dockerizado y desplegado con Docker-Compose la especificación completa puede encontrarse en el repositorio <https://github.com/RoberPlaza/RehabiliTEA.WS> con Licencia Pública General.

### **Base de datos de usuarios**

De todo el proyecto esta parte es en la que más se ha reducido el alcance, ya que como podemos ver en la imagen 5.2, la idea original era muy ambiciosa. Un vistazo en profundidad al diagrama revela que el modelo es muy flexible y admite situaciones tales como que diferentes juegos tengan diferentes posibles dificultades, cada una con una asignación de configuraciones completamente diferente.

Esto es debido a que la especificación de requisitos original iba a permitir que un terapeuta cambiase los parámetros de cada dificultad de cada juego para cada usuario. El concepto de «nivel» o «asignación de dificultad» vino después para poder simplificar el diseño de la base de datos. Este modelo de datos es suficientemente bueno como para llevar a cabo esta primera especificación de requisitos, el problema llegó a la hora de implementar controladores para este modelo en el juego.

## 5. RESULTADOS

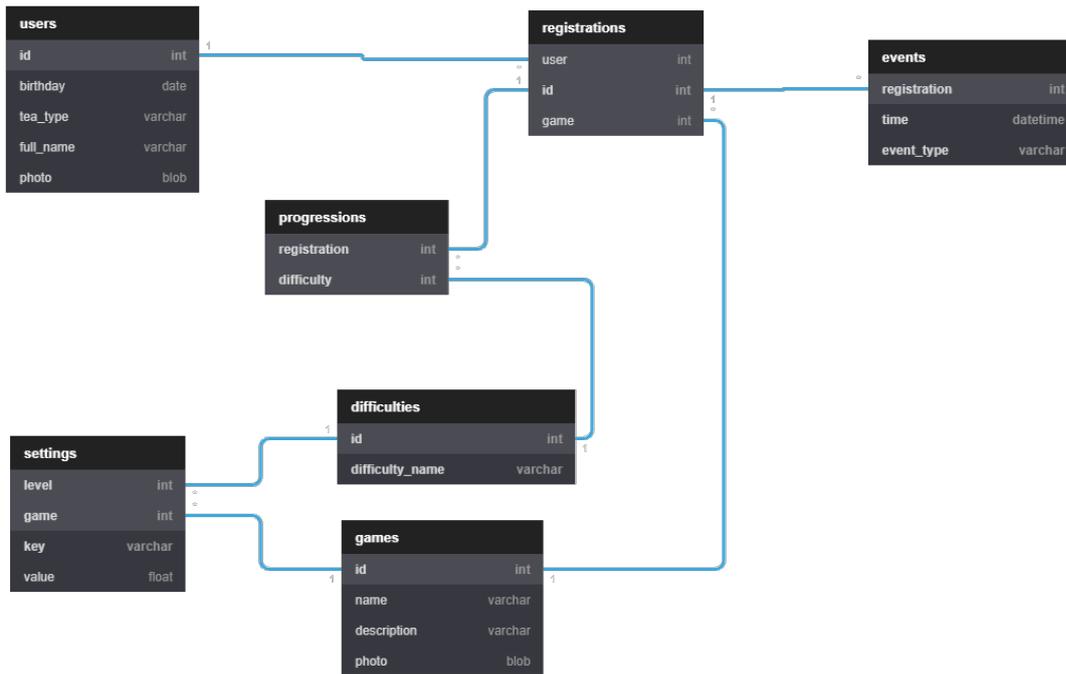


Figura 5.2: Modelo de datos del alcance original.

Además de la dificultad y carga de trabajo extra que supone el modelo, hay un inconveniente añadido, y es que la seguridad y ética del modelo de datos. Guardar información referente a la enfermedad de una persona en una base de datos de usuarios de videojuego puede acarrear problemas en forma de medidas de seguridad extra. Además del costo que puede suponer implantar estas medidas, encontramos que no es necesaria la mayoría de la información para que el juego funcione a nivel técnico.

En su lugar se ha optado por asignar un identificador a cada dispositivo. Este identificador es la columna «id» de la tabla «usuarios». Este id podrá contrastarse con información almacenada y recopilada en medios no digitales, o por lo menos, medios a los que no se pueda acceder a través de una dirección IP pública. En este caso se han usado formularios de Windows alojados en servidores europeos.

En su lugar se ha optado por el modelo esquema de base de datos descrito en la figura 5.3. Este modelo es mucho más simple y permite mucho menos, pero la tabla de eventos, probablemente la más importante sigue igual. En la versión de producción ambos modelos están más desnormalizados, y es que aunque se haya utilizado una librería de ORM para facilitar la interacción con la base de datos, poblar esta puede volverse una tarea compleja. También, la flexibilidad extra que ofrece el hecho de que las tablas de relación tengan identificadores propios puede ayudar a la mantenibilidad en este caso de desarrollador único.

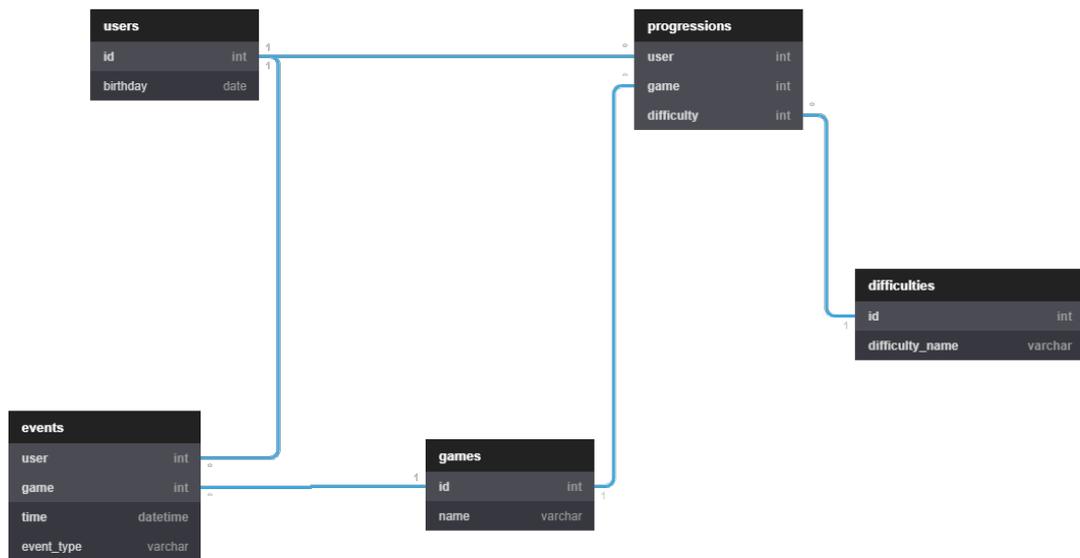


Figura 5.3: Modelo de datos del prototipo.

Pese a estar simplificado, el esquema guarda la información sobre los eventos que se han producido de la misma forma. Dejar un registro de cuando un usuario, ha hecho una interacción a una hora en un minijuego es la forma más en bruto de guardar los datos. Esto abre las puertas a que un desarrollador hábil pueda programar e indagar sobre la extracción de indicadores de muy diversa índole en esta colección de datos.

### Series temporales y espectrometría

Por otro lado, las lecturas del casco de espectrometría son guardadas en una base de datos específica para series temporales. Las bases de datos de series temporales o Time series database (TSDB), son bases de datos optimizadas para la inserción, búsqueda y almacenamiento de datos indexados por tiempo. De forma común se utilizan para monitorizar el consumo de recursos de sistemas, pero tienen aplicación en otros contextos como el de los mercados o la evolución de las lecturas de sensores.

Se decidió usar tecnología en lugar de ampliar la base de datos genérica por dos motivos. El primero de estos es que el caso de uso se adecuaba a la tecnología, las medidas registradas por el casco pueden ser consideradas como las medidas de un sensor más tradicional. No obstante, que una tecnología soporte un caso de uso a veces no justifica completamente el uso de esta tecnología, es por esto que la gran cantidad de herramientas de analíticas y visualización de series temporales declinan la balanza.

Herramientas como «Grafana» o «Chronograf» usan de las particularidades de este tipo de bases de datos para generar información visual de forma eficaz y sencilla. En particular,

## 5. RESULTADOS

Chronograf, es de los creadores de InfluxDB y funcionan de forma muy cohesionada. Por medio de una consulta básica, Chronograf es capaz de transformar una serie temporal en diferentes tipos de gráficos actualizados en tiempo real.

La mayoría de las bases de datos de series temporales no cumplen, ni buscan cumplir, el estándar SQL. Este es el caso de InfluxDB, que no implementa conceptos como uniones o claves foráneas, en su lugar, InfluxDB ofrece crear medidas o «**measurements**», equivalentes a tablas de SQL. El concepto de «tags» es el de variables estáticas, sirven para distinguir unas medidas de otras, están desconectadas del tiempo y se consideran estáticas para cada entidad.

Por último, los campos o «fields» hacen referencia a aquellas variables que son dependientes o están relacionadas a una marca temporal. En el caso que a este proyecto ocupa, el formato de los datos de entrada es el siguiente:

- **Measures:** Una única medida llamada «brain\_activity»
- **Tags:** El id del paciente de cada medición
- **Fields:** Las lecturas de las ondas alpha, beta, gamma t theta de los sensores AF3, T7, Pz, T8 y AF4.

El proceso de recolección empieza cuando el usuario que asiste al niño durante una sesión de este tipo pulsa el botón «recolectar» en una página web para ello. En ese momento un script de JavaScript se conecta con el driver de los cascos y a cada nueva medida actualizara la base de datos y genera gráficos en tiempo real.



Figura 5.4: Ejemplo de cuadro de mando hecho con Cronograf.

### 5.4 Prototipo de aplicación para dispositivos móviles

Se pueden encontrar varias versiones dentro de la rama de etiquetas del repositorio, <https://github.com/RoberPlaza/RehabiliTEA.App/tags>. Estas versiones corresponden a los lanza-

mientos de evaluación a lo largo del proyecto. Hay ejecutables tanto para Windows como para Android en formato aplicación.

### 5.4.1 Marco general

La aplicación para móviles se divide por minijuegos. Cada minijuego es una actividad que toma entre dos y cinco minutos para ser completada. Cada jugador tiene un máximo de 30 minutos al día para jugar, sin embargo, podrá jugar más, ya que la aplicación no se va a bloquear hasta que se acabe uno de estos minijuegos. La razón para aplicar esta restricción es evitar contaminar la muestra del estudio con tiempos dispares.

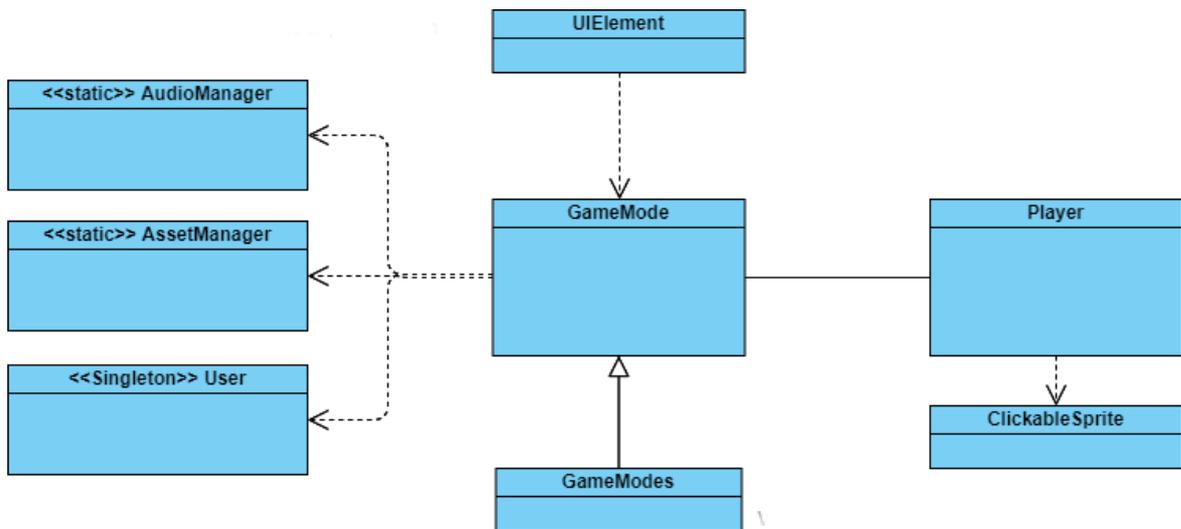


Figura 5.5: Diagrama de clases simplificado de las principales clases de la aplicación móvil.

Una parte del código del juego ha sido reutilizada para todos los minijuegos. Pese a, posiblemente, complicar el diagrama de clases, se puede ahorrar mucho código por medio de crear clases con las tareas básicas que todos los juegos deben llevar a cabo. El principal ejemplo de esta filosofía es la clase «GameMode». Un modo de juego atiende a un patrón **Manager**, una versión del Singleton adaptada al desarrollo de juegos.

Un Manager es una clase de la que sólo debe haber una instancia. Se diferencia del Singleton en que este último usa de instancia estática y el resto de entidades pueden acceder a ella. Un manager, por otro lado, puede, a nivel código, instanciarse más de una vez. Esto puede ser problemático en un desarrollo común, pero trabajando con un editor de mapas es muy sencillo evitar instancias duplicadas. Desde un punto de vista más técnico, la ventaja que ofrece un Manager sobre un Singleton es que el acceso universal a un Manager no está garantizado y esto podía llegar a ser un gran problema en el caso del Sigleton.

Otro ejemplo de patrón puede encontrarse en la clase «AudioManager». En este caso se está usando una inyección de dependencia de tipo setter. A una fuente de audio se le cambia el clip de audio que reproduce para poder usar un punto fijo en el mapa como altavoz. También podemos encontrar un caso de patrón **Prototype** en los prefabs de Unity. No obstante, los

## 5. RESULTADOS

prefabs de tanto el gestor de contenido como el gestor de audio atienden más a una versión avanzada de un Singleton.

Por último, un patrón avanzado dentro del código es el **Observer**, implementado por medio de **delegados multicast**. La inclusión de delegados multicast en lenguajes de alto nivel es muy útil, ya que previenen que el desarrollador necesite escribir la jerarquía de clases de un patrón Observer, eliminando así el principal inconveniente de este patrón.

### 5.4.2 Minijuegos

Este proyecto presenta tres minijuegos. Al tratarse de un prototipo no se ha dedicado un tiempo o esfuerzo mayor en esta parte ya que el concepto de «Scene» o «Escena» de Unity permite el desarrollo e integración de nuevos minijuegos de forma escalable siempre y cuando ofrezcamos la posibilidad al usuario de cargar nuevas escenas.

#### Memory

Memory es una actividad en la cual el jugador es capaz de voltear dos cartas a la vez, si las dos cartas tienen el mismo símbolo estas se quedan bocarriba, si no se volverán a girar, el objetivo es darle la vuelta a todas las cartas. Es un juego clásico implementado como parte de esta colección de minijuegos de rehabilitación.

El usuario interactúa con las cartas por medio de clicks o tocar con el dedo en la pantalla y se ha prescindido de interfaz 2D porque el estado de las cartas aporta al jugador la información suficiente para que este entienda el estado de juego. Es, de la colección, el juego más complejo de implementar.

Además de crear un modo de juego y un gestor de spawn de objetos, lo común, ha sido necesario crear una generalización de la clase «ClickableSprite». Esta clase se encarga de girar la carta una vez que ha sido seleccionada, además de consultar la colisión. También se ha tenido que crear una clase derivada de el jugador, para que el rayo que el usuario lanza pueda generar un evento con una de estas cartas.

#### Burbujas

El juego más sencillo en cuanto a mecánicas y más corto en cuanto a tiempo es el de las burbujas. En este minijuego el usuario debe explotar las burbujas en orden numérico, empezando desde '0' y acabando en '4', '6' o '9' dependiendo de la dificultad. Estas burbujas aparecerán de forma aleatoria por la pantalla. En la dificultad más alta pueden aparecer elementos distractorios como formas y/o letras.

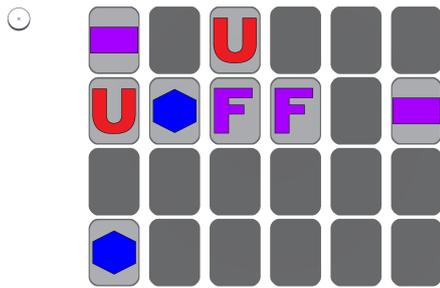
De la misma manera que en Memory no es necesario implementar interfaz ya que el propio 'tablero' muestra información suficiente al jugador. Para desarrollar este minijuego se han creado sus respectivos modo de juego y gestor de spawn. El usuario interactúa con el juego por medio de clicks con el ratón o taps en la pantalla usando el canal de eventos estándar.



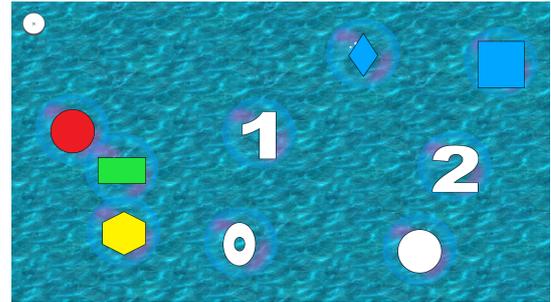
(a) Menú principal



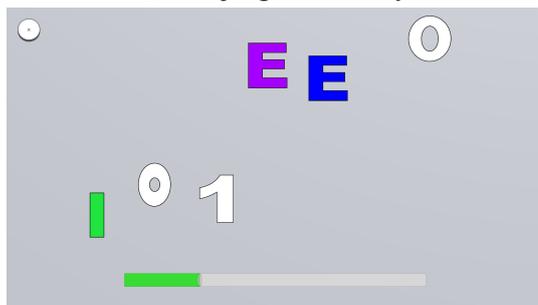
(b) Minijuegos de RehabiliTEA



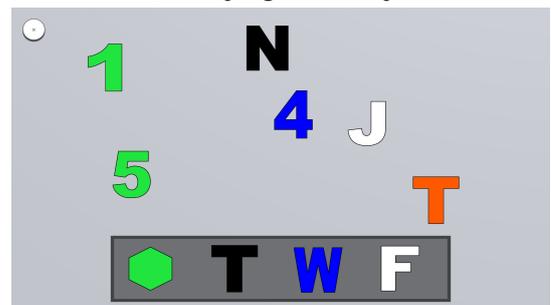
(c) Minijuego «Memory»



(d) Minijuego «Burbujas»



(e) Minijuego «Donde están»



(f) Minijuego «Donde están», menú



(g) Minijuego «Donde están», mensaje positivo



(h) Minijuego «Donde están», mensaje de apoyo



(i) Mensaje cuando el usuario pasa el minijuego.



(j) Mensaje cuando el usuario falla el minijuego.

Figura 5.6: Capturas de la aplicación final.

### ¿Donde están?

El último juego trata sobre memoria visual. Al usuario se le presentarán un número de figuras en pantalla, de las cuales una desaparecerá a los pocos segundos. Al usuario se le presentan cuatro alternativas de cuál ha sido la figura que ha desaparecido y ha de elegir una. El número de figuras así como el tiempo que permanecen todas en pantalla está ligado a la dificultad con la que el usuario está jugando.

Este es el único juego que cuenta con un elemento de interfaz en todo momento, consta de una barra de progreso para indicar al jugador cuánto tiempo queda para que desaparezca la figura de la ronda. Además del gestor de interfaz hizo falta codificar el modo de juego y el gestor de spawns. La dificultad también afecta al tipo de figuras que pueden aparecer, en fácil sólo aparecerán formas geométricas, en normal se añade a la lista de posibles sprites la colección de números y letras, en difícil también podrán aparecer dibujos.

### 5.4.3 Gestión de perfiles

Se ha hablado en la sección sobre la que se almacenaban datos sobre que la idea original de la gestión de perfiles era la de crear dificultades completamente ajustables para cada usuario. Por medio de relacionar *muchos a muchos* los minijuegos y las dificultades tenemos el concepto de nivel, el cual puede guardarse para cada usuario. Por otro lado, un terapeuta podría, en cualquier momento modificar los ajustes de cada nivel para adecuarlos en función de los resultados.

En primer lugar, que cada minijuego tiene configuración propia. Desde un punto de vista de un lenguaje menos estricto esto no es problema, puesto que formatos como XML (eXtended Markup Language) o JSON (JavaScript Object Notation) pueden cargarse en memoria y representar cualquier estructura de datos. Lamentablemente, C# es un lenguaje de tipos fuertes y estáticos, es decir, no permite cargar el contenido de uno de estos ficheros en memoria como un objeto nativo.

La solución pasa por crear una clase de configuración para cada modo de juego. Unity, por suerte, dispone de un constructor de objetos en base a JSON. Por ejemplo, si una clase «Class» tiene los atributos *int attrib1*; e *int attrib2*; esta puede ser instanciada con un JSON tal que `{ "attrib1": 1, "attrib2": 2 }`. Desgraciadamente esto no sólo complica el diseño, sino que también afectaría notablemente a los tiempos de desarrollo, la mantenibilidad y la depuración del código.

El principal inconveniente es que si en alguna actividad de añade o substraer alguna clave de configuración de algún minijuego es necesario, una vez acabadas las labores de desarrollo y depuración distribuir la versión actualizada de la aplicación a todos los usuarios. Esto conlleva añadir un sistema de coherencia de versiones y actualización forzosa.

## 5.5 Infraestructura y protocolos web

El diagrama de despliegue de la figura 5.1 se traduce en la especificación de Docker-Compose dentro del fichero raíz del repositorio del servicio Web. Dicho fichero consta de cinco microservicios los cuales podemos desplegar usando un único comando y definen todos los servicios de RehabiliTEA más allá de la aplicación Android.

Además de los servicios también se especifican tres volúmenes. Estos volúmenes sirven para dotar a los contenedores de de persistencia. Estos volúmenes son discos virtuales que crea el driver de Docker, estos volúmenes pueden ser tumbados manualmente o serán cerrados una vez el driver deje de funcionar. Esto quiere decir que con cada reinicio del equipo los datos se pierden, pero es posible tumbar y levantar la arquitectura sin perder los datos.

### Servicio Web

El servicio web consta de los servicios «rehabilitea» y «nginx». El primero de estos es una API REST desarrollada con la librería de Python «Flask». Esta API ofrece la información sobre usuarios y juegos en los *endpoints* «`http://rehabilitea/user/<id_usuario>`» y «`http://rehabilitea/game/<id_juego>`». También puede recibir información sobre eventos en «`http://rehabilitea/event`», enviando un JSON con los campos «user», «game» y «type».

Análogamente, en el *endpoint* raíz o índice el servidor sirve el programa JavaScript para conectarse al casco y enviar los datos generados a la base de datos. De forma ideal esto debería ser un servicio a parte, pero, al tratarse de una sola página, es posible reducir el número de servicios y la complejidad del prototipo por medio de unir los dos servicios.

El microservicio de nginx entra dentro de esta categoría, puesto que está configurado para redirigir el tráfico al servicio. Es por esto que podemos ver que en el orquestador la aplicación de rehabilitea se lanza con gunicorn y cuatro trabajadores, es decir, hay cuatro instancias del servicio sobre las que el servidor nginx redirigirá y balanceará las peticiones. El orquestador expone el puerto 5000 del servicio para poder comunicar el servidor nginx con la aplicación y redirige el puerto 80 de nginx para permitir al usuario acceder a esta.

### Base de datos PostgreSQL

La base de datos utilizada ha sido una PostgreSQL. Con su nombre es posible deducir que cumple el estándar SQL, sin embargo, también lo amplía con atributos como herencia de tablas. El puerto por defecto de este servicio es el 5432. En el fichero de producción dicho puerto no está redirigido, esto se debe a que, en producción, sólo la aplicación web deberá interactuar con la base de datos, pero en desarrollo puede ser interesante utilizar «pgAdmin» para consultar el estado interno de la base de datos.

La imagen docker de Postgres crea un usuario con contraseña, un administrador con contraseña y una base de datos por defecto usando las variables de entorno. La manera que se ha

## 5. RESULTADOS

tenido para lidiar con el entorno de desarrollo y producción es por medio de crear los ficheros «dev.env» y «prod.env» además de un «docker-compose.override.yml». Los entornos de producción y la sobrecarga del docker-compose no están subidos al repositorio.

Los protocolos utilizados por Postgres en este caso son irrelevantes, puesto que la librería ORM «Flask-SQLAlchemy» resuelve toda la comunicación con la base de datos de forma automática. En otras condiciones será necesario usar un lenguaje de programación con librerías para comunicación con Postgres o el plugin «pgsql-http».

### **InfluxDB y Chronograf**

La parte final de la infraestructura consta del tándem InfluxDB + Chronograf. Ambos softwares son desarrollados por InfluxData y forman parte de la pila TICK (Telegraf, InfluxDB, Chronograf y Kapacitor). Telegraf y Kapacitor son programas que sirven para recolectar datos de la máquina en la que se está ejecutando el servicio, en el caso de Telegraf y procesar datos en stream y lotes en el caso de Kapacitor.

InfluxDB es, como se ha comentado anteriormente, una base de datos de series temporales, es decir, no es SQL *compliant*. A cambio, Influx ofrece un algoritmo de compresión de datos más óptimo y tiempos de búsqueda menores. Chronograf, por otro lado es una herramienta de visualización de datos, permite generar gráficos, vistas y exportar datos a csv, entre otras utilidades, sin necesidad de programar o entender Influx.

Esta base de datos dispone de su propio lenguaje de consulta, InfluxQL, similar al SQL estándar. No obstante, Chronograf ofrece un IDE para generar la consulta sin necesidad de programar. Para insertar datos en InfluxDB es posible usar las librerías oficiales según el lenguaje de programación o mediante un Post HTTP en el cual se codifiquen los datos como datos en crudo o en formato binario.

## **5.6 Validación y pruebas**

Se han planificado dos procesos de validación idénticos. La diferencia es que uno de ellos se ha llevado a cabo con el fin de ofrecer *feedback* sobre la opinión de diferentes terapeutas sobre este prototipo, y el otro se realizará con la versión final del sistema antes de que este sea desplegado. De esta manera los encuestados podrán ver si sus sugerencias han sido escuchadas.

### **5.6.1 Planteamiento**

La validación se hace por medio de un cuestionario de la plataforma «Microsoft Forms». Se ha usado este servicio y no otro con el fin de utilizar las herramientas que ofrece la universidad. El cuestionario está dirigido a profesionales terapeutas de atención temprana y habilitación funcional de dentro y fuera de la Universidad de Castilla-La Mancha. Análogamente el uso de MS Forms podrá ser utilizado para lidiar con los datos de carácter personal.

En primer lugar, se ha grabado un vídeo mostrando las partes más relevantes del sistema, dicho vídeo puede encontrarse en <https://youtu.be/TpXF4eUyXbM>. En la descripción del cuestionario, los encuestados podrán ver una descripción del sistema, sus objetivos y la declaración de privacidad. Dicho cuestionario consta de 10 preguntas sobre los aspectos de usabilidad y adecuación funcional de la parte visible del sistema.

El cuestionario ha sido diseñado por Abel Toledano González, cotutor de este proyecto y doctor en el ámbito de la psicología. El cuestionario sigue el estándar de Sistema de Escalas de Usabilidad, también conocido como EUS o SUS<sup>1</sup>. Un test de este formato consta de 10 preguntas. Dichas preguntas se plantean a modo de declaraciones y el usuario responde cómo de acuerdo está con esas declaraciones en una escala de uno a cinco, siendo uno «Desacuerdo total» y cinco «Acuerdo total».

Estas 10 preguntas pueden variarse en función de la aplicación, no obstante, se han usado las 10 preguntas estándar del modelo de evaluación. Abel, por su parte, el perfil más cercano a los profesionales que operarán el sistema, considera este conjunto adecuado para comprobar la usabilidad del sistema. Las preguntas son las siguientes:

- ¿Usaría esta aplicación frecuentemente?
- ¿Encuentra esta aplicación innecesariamente compleja?
- ¿Cree que la aplicación es fácil de usar?
- ¿Cree que necesitaría ayuda de una persona con conocimientos técnicos para usar esta aplicación?
- ¿Cree que las funciones de esta aplicación están bien integradas?
- ¿Cree que la aplicación es inconsistente?
- ¿Cree que la mayoría de las personas aprendería a usar esta aplicación fácilmente?
- ¿Encuentra la aplicación difícil de usar?
- ¿Se siente confiado al usar esta aplicación?
- ¿Necesitaría aprender muchas cosas antes de ser capaz de usar esta aplicación?

### 5.6.2 Resultados

Este cuestionario estuvo abierto durante un periodo temporal de tres días y se recibieron un total de 37 respuestas. Pese a que las nociones generales son positivas se percibe que la principal preocupación de los operadores es la necesidad de un técnico para asistirles en su uso. Esto puede deberse principalmente a la falta de contacto que estos profesionales tienen con herramientas de monitorización, pero ha servido para considerar la opción de redactar un manual de usuario en la versión final.

<sup>1</sup><https://uxpanol.com/teoria/sistema-de-escalas-de-usabilidad-que-es-y-para-que-sirve/>

## 5. RESULTADOS

De todas las métricas, tal vez la moda arroje como resultado que las funciones de la aplicación no puedan estar del todo bien definidas o, en su defecto, se deba invertir esfuerzo extra en ella, ya que es la única declaración positiva que no obtiene la puntuación máxima como valor más repetido. No obstante, observando los mínimos, es posible deducir que la confianza y aceptación de la mayoría de expertos en la aplicación es, cuanto menos neutral.

Después de aplicar el algoritmo de medición se obtiene un 83.24 como nota media de los factores de usabilidad. No obstante, al ser la muestra bastante reducida se ha aplicado, también, el algoritmo de medición a la respuesta de cada usuario. Esto deja ver que los mayores detractores de la aplicación la puntúan con, como mínimo un 57.5, por otro lado la nota más repetida es 100, el máximo, por lo cual podemos concluir que los resultados del proceso son positivos.

<b>Declaración</b>	<b>Media</b>	<b>Máximo</b>	<b>Mínimo</b>	<b>Moda</b>
Creo que usaría esta aplicación frecuentemente	4.19	5	2	5
Encuentro esta aplicación innecesariamente compleja	1.65	5	1	1
Creo que la aplicación es fácil de usar	4.46	5	2	5
Creo que necesitaría ayuda de una persona con conocimientos técnicos para usar esta aplicación	1.89	5	1	1
Las funciones de esta aplicación están bien integradas	4.27	5	2	4
Creo que la aplicación es muy inconsistente	1.62	3	1	1
Imagino que la mayoría de las personas aprendería a usar esta aplicación fácilmente	4.40	5	3	5
Encuentro que la aplicación es muy difícil de usar	1.49	5	1	1
Me siento confiado al usar esta aplicación	4.32	5	3	5
Necesitaré aprender muchas cosas antes de ser capaz de usar esta aplicación	1.70	5	1	1

Cuadro 5.1: Resultados de la encuesta

## Capítulo 6

# Conclusiones y líneas de trabajo futuro.

**D**ESPUÉS de acabar el prototipo queda preguntarse qué ha sido del proyecto, echar la vista atrás y extraer conclusiones. Al ser este un capítulo más personal que técnico hablaré en primera persona en lugar de en impersonal a la hora de evaluar lo que ha supuesto este proyecto para mí.

### 6.1 Análisis de resultados

Para presentar lanzará el servicio en local conectando con el endpoint *localhost* puesto que el proyecto no será desplegado hasta después de el curso académico 2020-2021 haya comenzado.

La aplicación cuenta en, primer lugar, cuenta con actividades pensadas para estimular muy diversas habilidades de los usuarios. Tenemos una actividad para mejorar la memoria a medio y largo plazo, otra que consigue mejorar los reflejos y el reconocimiento de patrones y colores y, una última, que consigue poner a prueba la memoria visual de cualquier jugador o usuario que la pruebe.

Por otro lado, la aplicación web queda un poco desaprovechada. Es necesaria para almacenar el progreso de cada usuario y para ofrecer una experiencia más personalizada con la aplicación. Esto podría haberse hecho usando ficheros dentro de cada dispositivo, pero se hubiese perdido la posibilidad de escalar más el sistema. La idea original era que un terapeuta pudiera cambiar la dificultad de una aplicación a placer para cada usuario y no se ha podido hacer por el alcance tan amplio que propone.

La parte positiva de este componente radica en el inmenso potencial que tiene para recolección de información. El registro de eventos que se espera conseguir de esta aplicación puede contener una cantidad de información enorme que, combinada con los datos extraídos del casco, pueden aportar mucho a la investigación de enfermedades del neurodesarrollo infantil. Esta base de datos sería, además, pionera en su temática.

### 6.2 Adquisición de competencias y cumplimiento de objetivos

A lo largo de este proyecto he tenido que interactuar con profesionales de diferentes profesiones, aprendiendo a comunicarme, cómo lidiar con y la manera de pesar de cada uno.

## 6. CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURO.

Esta experiencia, esta posibilidad de interactuar con gente tan dispar laboralmente es, sin duda, el conocimiento más importante que soy capaz de extraer de este periodo. Muchas veces se olvida el factor humano propio de un proyecto y es importante recordar que lidiar con personas también es trabajo para un ingeniero.

A un nivel más técnico el proyecto usa tecnologías del estado del arte con técnicas del estado del arte para resolver problemas del estado del arte. Poco o nada más se puede pedir a un proyecto. Durante proyecto he aprendido a desarrollar interfaces y aplicaciones gráficas que una base de usuarios de tamaño considerable acepta y valida, atendiendo a las necesidades de un cliente.

También he tenido posibilidad de desarrollar, desplegar y mantener un sistema web basado en microservicios con las mismas garantías que una empresa competente y madura lo haría con sus productos. El uso de tecnologías como Docker, Docker Compose, librerías ORM, y servicios cloud se puede acercar bastante a las labores de un ingeniero de software DevOps puede llevar a cabo en una empresa en su día a día.

Las competencias específicas CE04, CE14 y CE15 han sido ejercitadas por la naturaleza propia del proyecto. El diseño de una infraestructura de microservicios debería satisfacer la CE04, además, esta se ve potenciada con tecnologías y métodos no estudiados en clase. Por otro lado cuando desarrollamos un videojuego estamos activamente entrenando las competencias CE14 y CE15 ya que, por definición un videojuego es un espacio 3D interactivo.

Por último, aunque he trabajado anteriormente en proyectos software, la gestión y seguimiento de este proyecto han sido un material docente especialmente útil. He llegado a comprender en profundidad que es lo que importa a la hora de evaluar el desarrollo de un proyecto software. La organización y desarrollo de un proyecto software también han sido materiales de trabajo interesantes, ya que, al ser el único responsable del código del proyecto he aprendido mucho a la hora de estimar fechas de entrega y tiempos de desarrollo de característica.

### 6.3 Líneas de trabajo futuro

Cuando pensamos en ampliar un sistema de minijuegos la primera idea que nos viene a la cabeza es la de incluir más minijuegos. El alcance original del proyecto era de cinco minijuegos, pero actualmente es posible llegar a 10 en la primera versión. Dentro del portfolio del proyecto hay pensadas más de 20 actividades. El desarrollo de nuevos minijuegos puede incrementar la vida útil del software y ser objeto de estudio para un trabajo fin de grado o máster sin problema.

Más allá de obviedades superficiales, el estudio original ha acabado relegado a una línea de trabajo futuro. En este estudio a una muestra de niños con TEA se les medirían, con ayuda de un profesional, las ondas cerebrales la primera vez que prueban la aplicación. Después se

les dejaría usarla libremente durante 90 días no más de 30 minutos diarios. Tras 90 días de uso se volvería a medir la actividad cerebral de cada niño con ayuda de un profesional del sector.

Una vez el sistema llegue a una versión preparada para el despliegue, la primera nueva característica que debería tener es su propia pipeline de DevOps. He ha hablado de cómo se prescindió de una ya que, en principio, sólo va a haber una versión de despliegue. Sin embargo, si el sistema siguiera funcionando una de estas pipelines sería un activo muy valioso para el proyecto, ya que permitiría trabajar con mayor comodidad a posibles nuevos desarrolladores.

Tal vez la parte más rudimentaria del sistema sea el script de conexión con el casco de espectrometría. Un proyecto para ofrecer una interfaz de usuario avanzada para esta tarea no se plantea descabellada. En lo personal creo que una arquitectura *serverless* con un programa hecho en Rust o C++ con QtCreator compilado a WebAssembly como objetivo puede ser la opción más vanguardista y sencilla para el usuario.

La opción de continuación tal vez más directa es la que se ha dejado ver en contadas partes del documento, hacer un estudio de los datos generados con técnicas de aprendizaje automático. Como ya se ha dicho, en estos datos puede haber enterrada muchísima información y muchísimo conocimiento y estamos en un periodo donde hay técnicas para extraerlo y personal formado con anhelos de llevar a cabo esta tarea.

En relación con la línea previa, más mediciones, más indicadores, una muestra de estudio más amplia puede beneficiar el estudio. Guardar la información como eventos ha sido un acierto por lo simple que es almacenar la información y transformarla, pero siempre hay lugar para más mediciones y casos. Aumentar la envergadura del sistema hasta que el volumen de datos sea un problema, además de abrir nuevas líneas de estudio, puede ayudar a la labor del científico de datos.

Además de las opciones de Análisis de datos, la idea original de este trabajo es la medición de síntomas físicos de pacientes de enfermedades del neurodesarrollo. Es decir, la generación de literatura científica en el ámbito de salud por medio de interpretar los resultados es, además de una línea de trabajo que se ha abierto, la principal motivación de este proyecto.

Durante algunas de las reuniones de seguimiento del proyecto se habló de la posibilidad de ampliar el alcance de la aplicación en cuanto a público objetivo. Actualmente la aplicación está preparada para atender las necesidades de estímulos de pacientes de TEA, en especial de síndrome de altas capacidades. Con la cantidad de recursos visuales e infraestructura de código y web implementada sería posible llegar a muchas más enfermedades de neurodesarrollo como el Trastorno de Déficit de Atención e Hiperactividad o, incluso, para mejorar el desarrollo cognitivo en niños hasta los siete años.

Por último, dejar claro que todo producto software puede ser mejorado. Mejores interfaces, código más óptimo, mejores recursos visuales y mejores paradigmas de interacción, son ejemplos de objetivos que deberían ser continuos y tenidos en cuenta en cada iteración. A esta colección de mejoras continuas puede añadirse un estudio de usuarios por plataforma para construir las versiones en plataformas más propensas a llegar a nuestro público.

## 6.4 Reflexión personal

Es menester comenzar diciendo que el proyecto aún no ha acabado puesto que está este en parte financiado por una beca de departamento que no cumple hasta Septiembre. Esto no viene a reforzar la idea de que este proyecto no representa un sistema sino un prototipo. No obstante, la sensación general del software presentado se me plantea, en el peor de los casos, adecuada. La sensación general del proyecto es muy buena y la satisfacción tras ver los resultados del prototipo es alta.

La capacidad de tener contenido como dibujos y audio desarrollados sólo para un juego del que estoy a cargo es una sensación muy buena. La responsabilidad de que salga bien, el orgullo cuando empieza a tomar forma y la emoción de un despliegue a producción son sentimientos que se van a quedar conmigo mucho tiempo y que me han influenciado como profesional.

Espero, también, algún día poder realizar personalmente un estudio de estos datos, ya que el diario de laboratorio resultante, desde el diseño hasta los resultados, puede ser muy interesante. Si esto se llevase a cabo existiría un manual directo sobre cómo implementar y gestionar un estudio de minería de datos para labores de investigación, un logro del cual hacer gala.

En última instancia, este proyecto ha puesto a prueba todo lo que buscaba, lo he conseguido enfocar de la manera que quería para ser el profesional que espero ser. Aúna, también, los aspectos que más disfruto del amplio mundo de la informática, desarrollo de videojuegos, cultura DevOps, herramientas de desarrollo modernas y responsabilidad sobre un sistema que sobre el que puedo afirmar orgulloso, «es mío».

Me gustaría acabar la parte evaluable de este documento diciendo que este ha sido el proyecto adecuado en el momento adecuado, he crecido como profesional, y lo que es más importante, como persona gracias a él y por ello sólo me queda dar gracias a todo el que lo ha hecho posible.

# ANEXOS



## Anexo A

# Infraestructura como código de la aplicación

```
services:
  rehabilitea:
    build: rehabilitea/
    command: gunicorn --bind 0.0.0.0:5000 manage:app --workers=4
    expose: [ "5000" ]
    env_file: ["rehabilitea/dev.env"]
    depends_on: ["database", "influxdb"]

  database:
    image: postgres:12-alpine
    volumes: ["postgres_data:/var/lib/postgresql/data/"]
    env_file: ["postgres/dev.env"]
    ports: ["5432:5432"]

  nginx:
    build: nginx/
    ports: ["80:80"]
    depends_on: ["rehabilitea"]

  influxdb:
    image: influxdb:1.8-alpine
    env_file: influxdb/dev.env
    volumes: ["influx_data:/var/lib/influxdb"]
    ports: ["8086:8086"]

  chronograf:
    image: chronograf:1.6-alpine
    volumes: ["chronograf_data:/var/lib/chronograf"]
    ports: ["8888:8888"]
    links: ["influxdb"]
    depends_on: ["influxdb"]

volumes:
  postgres_data:
  influx_data:
  chronograf_data:
```

De todos los ficheros del proyecto este sea probablemente el más importante. En la figura 5.1 se presenta un posible despliegue de los elementos hardware y software de la aplicación, este documento presenta la infraestructura final. Es posible que el lector encuentre más sencilla la implementación de la infraestructura como código, motivo suficiente como para justificar la inclusión de este anexo.

## Anexo B

# Contenidos multimedia utilizados para el desarrollo del proyecto.

Este anexo adjunta algunos de los contenidos multimedia utilizados a lo largo del desarrollo. En primera instancia es posible ver una muestra variada de los sprites utilizados. En el segundo bloque se muestra la primera prueba de concepto para la interfaz de la aplicación de juegos para el móvil.



Figura B.1: Propuesta visual del prototipo.

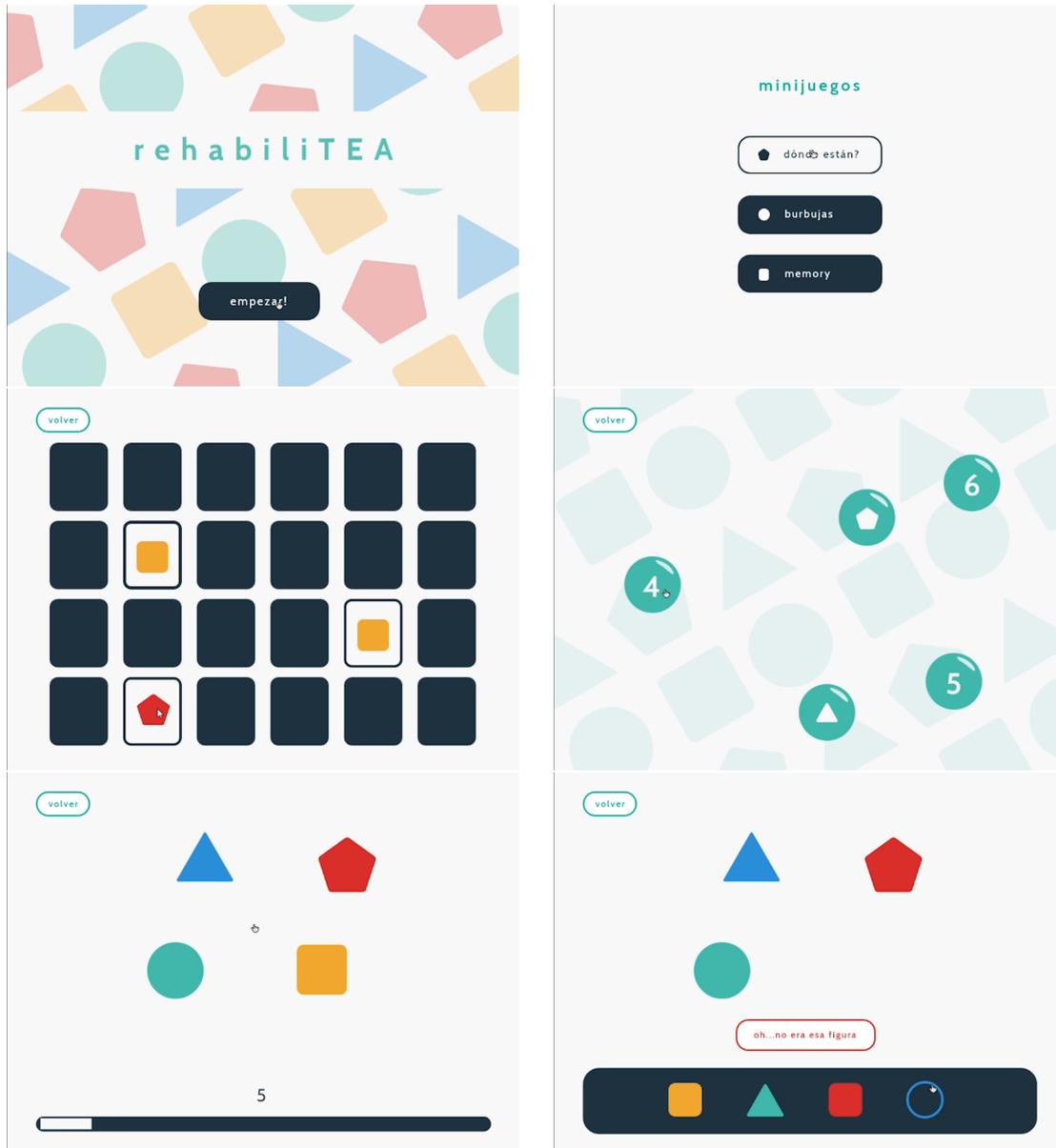


Figura B.2: Ejemplos de assets utilizados en el juego.

## Anexo C

# Resultados de la validación.

Si bien, poner los resultados en crudo de la validación en el capítulo de resultados podría haber sido factible, debido a las limitaciones de espacio se optó por redactar la interpretación de los mismos. Si el lector busca los resultados en crudo del proceso de validación los puede encontrar en la imagen de la siguiente página.

Se ha prescindido de los nombres y los correos electrónicos para preservar la privacidad de los encuestados. Por defecto MS Forms exporta los resultados a un fichero «xlsx», cargable en el programa «MS Excel». La última columna es el resultado de aplicar el algoritmo para calcular la nota<sup>1</sup>, es decir, es una columna sintética, añadida después de generar los resultados.

---

<sup>1</sup><https://uxpanol.com/teoria/sistema-de-escalas-de-usabilidad-que-es-y-para-que-sirve/>

ID	Start time	Completion t	Creo que	Encuentr	Creo que	Creo que	Creo que	Las funci	Creo que	Imagino c	Encuentr	Me sient	Necesita	Si desea	Nota		
1	7/5/20 16:33:03	7/5/20 16:34:19	5	1	5	1	5	1	5	5	1	5	1	5	1	100	
2	7/5/20 17:12:12	7/5/20 17:15:44	4	1	4	1	5	1	1	4	2	5	1	4	1	90	
3	7/5/20 17:11:30	7/5/20 17:47:22	2	3	5	1	4	4	2	5	5	4	1	4	1	70	
4	7/5/20 18:03:55	7/5/20 18:05:33	5	1	2	2	5	1	1	5	1	5	1	5	1	90	
5	7/5/20 18:21:00	7/5/20 18:22:48	5	5	5	5	5	5	1	5	1	5	5	1	Me parece s	80	
6	7/5/20 18:39:27	7/5/20 18:40:26	3	2	4	3	3	1	1	3	1	3	3	3	3	65	
7	7/5/20 18:57:16	7/5/20 19:00:10	5	1	5	4	4	4	1	4	1	4	4	5	La unión de	75	
8	7/5/20 20:06:38	7/5/20 20:08:07	4	2	4	2	4	1	1	4	1	4	2	2	2	80	
9	7/5/20 20:28:21	7/5/20 20:31:59	5	1	5	2	5	1	1	5	1	5	1	5	1	97.5	
10	7/5/20 21:16:45	7/5/20 21:17:27	4	2	4	4	4	4	3	4	2	3	2	2	2	65	
11	7/5/20 21:44:54	7/5/20 21:55:02	4	1	5	1	5	1	1	5	1	5	1	5	1	97.5	
12	7/5/20 22:08:18	7/5/20 22:12:13	4	2	4	3	4	4	2	4	2	4	4	2	2	72.5	
13	7/5/20 22:08:28	7/5/20 22:21:12	5	1	5	1	4	4	2	5	1	4	4	1	1	92.5	
14	7/5/20 22:31:44	7/5/20 22:36:07	4	2	5	2	4	4	2	4	1	4	4	2	2	80	
15	7/5/20 22:42:38	7/5/20 22:44:13	5	1	5	1	4	4	2	3	1	5	1	5	1	90	
16	7/5/20 23:45:39	7/5/20 23:54:19	4	1	4	1	4	4	1	4	1	4	1	4	1	87.5	
17	7/6/20 0:38:24	7/6/20 0:48:46	5	1	5	1	5	1	1	5	1	5	1	5	1	100	
18	7/6/20 2:34:14	7/6/20 2:38:33	4	3	5	2	4	4	2	4	2	4	4	4	4	70	
19	7/6/20 9:42:45	7/6/20 9:52:27	3	1	4	1	4	4	3	4	1	3	1	3	1	77.5	
20	7/6/20 10:29:29	7/6/20 10:35:13	3	1	4	1	4	4	2	4	1	4	1	4	1	82.5	
21	7/6/20 10:44:07	7/6/20 10:52:39	4	3	4	4	4	4	2	4	3	4	4	3	3	62.5	
22	7/6/20 11:42:59	7/6/20 11:52:10	5	1	5	2	5	5	1	5	1	5	1	5	1	97.5	
23	7/6/20 11:57:53	7/6/20 12:00:26	5	1	5	1	5	1	1	5	1	5	1	5	1	100	
24	7/6/20 12:03:38	7/6/20 12:15:06	3	1	5	1	4	4	2	5	1	4	4	2	Creo que al	85	
25	7/6/20 12:14:36	7/6/20 12:24:04	4	1	5	1	4	4	3	5	1	5	1	5	1	90	
26	7/6/20 13:27:26	7/6/20 13:37:24	5	1	4	4	4	4	1	4	1	4	4	5	4	Ciertamente	77.5
27	7/6/20 13:34:46	7/6/20 13:40:23	2	3	3	1	3	3	3	3	1	3	3	3	3	57.5	
28	7/6/20 14:54:02	7/6/20 15:02:58	5	2	5	1	5	1	1	5	1	5	1	5	1	97.5	
29	7/6/20 16:17:28	7/6/20 16:26:09	5	3	5	1	5	1	1	5	1	5	1	5	1	Me parece h	100
30	7/6/20 16:37:25	7/6/20 16:46:11	3	1	4	2	2	3	3	5	1	4	4	1	1	Primeros felii	70
31	7/6/20 19:47:52	7/6/20 19:57:17	5	1	3	4	5	1	1	3	3	3	5	4	4	70	
32	7/6/20 22:49:58	7/6/20 22:50:45	5	1	5	1	5	1	1	5	1	5	1	5	1	Esta genial!!	100
33	7/6/20 22:53:51	7/6/20 22:55:22	4	3	3	3	3	3	2	4	3	3	3	3	3	57.5	
34	7/6/20 22:52:56	7/6/20 22:59:54	3	1	5	1	5	1	1	5	5	5	1	5	1	85	
35	7/6/20 23:07:50	7/6/20 23:09:19	5	1	5	1	5	1	1	5	1	5	1	5	1	Muy interes	100
36	7/6/20 23:07:44	7/6/20 23:11:56	5	2	5	2	4	4	2	5	1	4	4	1	1	87.5	
37	7/6/20 23:18:37	7/6/20 23:25:42	4	2	5	1	4	4	3	4	1	3	1	3	1	80	

# Referencias

- [A<sup>+</sup>13] American Psychiatric Association et al. *Diagnostic and statistical manual of mental disorders (DSM-5®)*. American Psychiatric Pub, 2013.
- [AD<sup>+</sup>11] Julian Alvarez, Damien Djaouti, et al. An introduction to Serious game Definitions and concepts. *Serious Games & Simulation for Risks Management*, 11(1):11–15, 2011.
- [BAAH13] Fran C Blumberg, Debby E Almonte, Jared S Anthony, y Naoko Hashimoto. Serious games: What are they? What do they do? Why should we play them. *The Oxford handbook of media psychology*, páginas 334–351, 2013.
- [BBDR<sup>+</sup>16] Ali Basiri, Niosha Behnam, Ruud De Rooij, Lorin Hochstein, Luke Kosewski, Justin Reynolds, y Casey Rosenthal. Chaos engineering. *IEEE Software*, 33(3):35–41, 2016.
- [BFC<sup>+</sup>09] Jeanne H Brockmyer, Christine M Fox, Kathleen A Curtiss, Evan McBroom, Kimberly M Burkhart, y Jacquelyn N Pidruzny. The development of the Game Engagement Questionnaire: A measure of engagement in video game-playing. *Journal of Experimental Social Psychology*, 45(4):624–634, 2009.
- [BLL<sup>+</sup>15] Kelly J Bower, Julie Louie, Yoseph Landesrocha, Paul Seedy, Alexandra Gorelik, y Julie Bernhardt. Clinical feasibility of interactive motion-controlled games for stroke rehabilitation. *Journal of neuroengineering and rehabilitation*, 12(1):63, 2015.
- [BMC<sup>+</sup>09] James William Burke, MDJ McNeill, Darryl K Charles, Philip J Morrow, Jacqui H Crosbie, y Suzanne M McDonough. Optimising engagement for stroke rehabilitation using serious games. *The Visual Computer*, 25(12):1085, 2009.
- [BMC<sup>+</sup>10] James William Burke, MDJ McNeill, Darryl K Charles, Philip J Morrow, Jacqui H Crosbie, y SM McDonough. Augmented reality games for upper-limb stroke rehabilitation. En *2010 Second International Conference on*

*Games and Virtual Worlds for Serious Applications*, páginas 75–78. IEEE, 2010.

- [BML<sup>+</sup>19] Abigail Bangerter, Nikolay V Manyakov, David Lewin, Matthew Boice, Andrew Skalkin, Shyla Jagannatha, Meenakshi Chatterjee, Geraldine Dawson, Matthew S Goodwin, Robert Hendren, et al. Caregiver Daily Reporting of Symptoms in Autism Spectrum Disorder: Observational Study Using Web and Mobile Apps. *JMIR mental health*, 6(3):e11365, 2019.
- [Che20] Chi-Ying Chen. Smartphone addiction: Psychological and social factors predict the use and abuse of a social mobile application. *Information, Communication & Society*, 23(3):454–467, 2020.
- [DAJR11] Damien Djaouti, Julian Alvarez, Jean-Pierre Jessel, y Olivier Rampnoux. Origins of serious games. En *Serious games and edutainment applications*, páginas 25–43. Springer, 2011.
- [Dic05] Michele D Dickey. Engaging by design: How engagement strategies in popular computer and video games can inform instructional design. *Educational technology research and development*, 53(2):67–83, 2005.
- [DPvH18] Thomas F Düllmann, Christina Paule, y André van Hoorn. Exploiting devops practices for dependable and secure continuous delivery pipelines. En *2018 IEEE/ACM 4th International Workshop on Rapid Continuous Software Engineering (RCoSE)*, páginas 27–30. IEEE, 2018.
- [EGHS16] Christof Ebert, Gorka Gallardo, Josune Hernantes, y Nicolas Serrano. DevOps. *Ieee Software*, 33(3):94–100, 2016.
- [HKS14] Juho Hamari, Jonna Koivisto, y Harri Sarsa. Does gamification work?—a literature review of empirical studies on gamification. En *2014 47th Hawaii international conference on system sciences*, páginas 3025–3034. Ieee, 2014.
- [Kak14] Masao Kakihara. Grasping a Global View of Smartphone Diffusion: An Analysis from a Global Smartphone Study. En *ICMB*, página 11, 2014.
- [LEES14] Fedwa Laamarti, Mohamad Eid, y Abdulmotaleb El Saddik. An overview of serious games. *International Journal of Computer Games Technology*, 2014, 2014.
- [LLBC19] Michael V Lombardo, Meng-Chuan Lai, y Simon Baron-Cohen. Big data approaches to decomposing heterogeneity across the autism spectrum. *Molecular psychiatry*, 24(10):1435–1450, 2019.

- [LPP<sup>+</sup>05] Mario Liotti, Steven R Pliszka, Ricardo Perez, Delia Kothmann, y Marty G Woldorff. Abnormal brain activity related to performance monitoring and error detection in children with ADHD. *Cortex*, 41(3):377–388, 2005.
- [MW12] Kyleigh Morissette y Irwin Wisozk. Gemba Kaizen: A commonsense approach to a continuous improvement strategy. 2012.
- [O’L08] Daniel E O’Leary. Gartner’s hype cycle and information system research issues. *International Journal of Accounting Information Systems*, 9(4):240–252, 2008.
- [PPRH17] Lauren Powell, Jack Parker, Naomi Robertson, y Valerie Harpin. Attention deficit hyperactivity disorder: is there an app for that? suitability assessment of apps for children and young people with ADHD. *JMIR mHealth and uHealth*, 5(10):e145, 2017.
- [PWV<sup>+</sup>19] Dave Parsons, Nathan J Wilson, Sharmila Vaz, Hoe Lee, y Reinie Cordier. Appropriateness of the TOBY Application, an iPad Intervention for Children with Autism Spectrum Disorder: A Thematic Approach. *Journal of autism and developmental disorders*, 49(10):4053–4066, 2019.
- [RMTPMR14] Gonzalo Ruiz-Manrique, Kazuhiro Tajima-Pozo, y Francisco Montañes-Rada. Case Report: “ADHD Trainer”: the mobile application that enhances cognitive skills in ADHD patients. *F1000Research*, 3, 2014.
- [SB02] Ken Schwaber y Mike Beedle. *Agile software development with Scrum*, volume 1. Prentice Hall Upper Saddle River, 2002.
- [SEZ<sup>+</sup>16] Sabrina Schuck, Natasha Emmerson, Hadar Ziv, Penelope Collins, Sara Arastoo, Mark Warschauer, Francis Crinella, y Kimberley Lakes. Designing an iPad app to monitor and improve classroom behavior for children with ADHD: iSelfControl feasibility and pilot studies. *PloS one*, 11(10), 2016.
- [She13] John L Sherry. The challenge of audience reception: A developmental model for educational game engagement. *New directions for child and adolescent development*, 2013(139):11–20, 2013.
- [Smi19] Rob Smith. Hype Cycle for Frontline Worker Technologies, 2019. 2019.
- [Sta19] StatCounter Global Stats. Mobile operating system market share worldwide. *Dostopno prek <https://gs.statcounter.com/os-market-share/mobile/worldwide>*, 2019.

- [VSD<sup>+</sup>19] Catalin Voss, Jesse Schwartz, Jena Daniels, Aaron Kline, Nick Haber, Peter Washington, Qandeel Tariq, Thomas N Robinson, Manisha Desai, Jennifer M Phillips, et al. Effect of wearable digital intervention for improving socialization in children with autism spectrum disorder: A randomized clinical trial. *JAMA pediatrics*, 173(5):446–454, 2019.
- [Wat09] Kelly Waters. Prioritization using moscow. *Agile Planning*, 12:31, 2009.
- [YBK<sup>+</sup>19] Benjamin E Yerys, Jennifer R Bertollo, Lauren Kenworthy, Geraldine Dawson, Elysa J Marco, Robert T Schultz, y Linmarie Sikich. Brief report: pilot study of a novel interactive digital treatment to improve cognitive control in children with autism spectrum disorder and co-occurring ADHD symptoms. *Journal of autism and developmental disorders*, 49(4):1727–1737, 2019.
- [YFYCZ<sup>+</sup>07] Zang Yu-Feng, He Yong, Zhu Chao-Zhe, Cao Qing-Jiu, Sui Man-Qiu, Liang Meng, Tian Li-Xia, Jiang Tian-Zi, y Wang Yu-Feng. Altered baseline brain activity in children with ADHD revealed by resting-state functional MRI. *Brain and Development*, 29(2):83–91, 2007.

Este documento fue editado y tipografiado con  $\text{\LaTeX}$  empleando la clase **esi-tfg** (versión 0.20181017) que se puede encontrar en:  
[https://bitbucket.org/esi\\_atc/esi-tfg](https://bitbucket.org/esi_atc/esi-tfg)

[respeta esta atribución al autor]

