

**HERRAMIENTA SOFTWARE PARA EL ENTRENAMIENTO Y DESARROLLO  
DE LAS HABILIDADES COGNITIVAS: MEMORIA Y ATENCIÓN**

**LADY MARCELA CHAPARRO ARIAS  
DANIEL ORDOÑEZ PRIETO**



**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA**

**2012**

**HERRAMIENTA SOFTWARE PARA EL ENTRENAMIENTO Y DESARROLLO  
DE LAS HABILIDADES COGNITIVAS: MEMORIA Y ATENCIÓN**

**LADY MARCELA CHAPARRO ARIAS  
DANIEL ORDOÑEZ PRIETO**

**Trabajo de grado presentado como requisito para optar al título de Ingeniero  
de Sistemas**

**Directora:  
SONIA CRISTINA GAMBOA SARMIENTO**



**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA**

**2012**

## DEDICATORIA

*Agradecidos con Dios, dedicamos este proyecto a:  
Nuestros padres, quienes nos brindaron su apoyo infinito;  
A nuestros hermanos, por darnos ánimo constante;  
Pero muy especialmente a nuestra querida hija Samantha, por darnos la motivación  
para seguir adelante; que la curiosidad y la alegría que brilla intensamente en su  
mirada, nunca se desvanezca. Te amamos.*

***Marcela y Daniel***

## **AGRADECIMIENTOS**

Los autores del presente proyecto expresan sus agradecimientos a:

La profesora **Sonia Cristina Gamboa Sarmiento**, directora del proyecto, por apoyarnos desde que se le planteó nuestra idea de proyecto, por su buena disposición siempre, por dedicar su tiempo y compartir su conocimiento en lo que contribuía al proyecto.

La **Universidad Industrial de Santander** y a la **Escuela de Ingeniería de Sistemas e Informática**, por la formación profesional recibida.

A nuestras familias por el apoyo incondicional a lo largo de nuestra carrera y especialmente en el tiempo de desarrollo de este proyecto, gracias por la colaboración y el ánimo constante.

## CONTENIDO

	<b>Pág.</b>
1. INTRODUCCIÓN	18
2. OBJETIVOS	20
2.1 OBJETIVO GENERAL	20
2.2 OBJETIVOS ESPECÍFICOS	20
3. MARCO DE REFERENCIA	21
3.1 HABILIDADES COGNITIVAS	21
3.1.1 Memoria	21
3.1.1.1 Memoria de Trabajo	23
3.1.2 Atención	27
3.1.2.1 Atención Selectiva	28
3.2 PROGRAMAS DE ENTRENAMIENTO COGNITIVO	30
3.2.1 Peak Genius	30
3.2.2 Lumosity	31
3.2.3 MIND360 Play Smart	33
3.2.4 Train your Brain with Dr Kawashima	33
3.3 HERRAMIENTAS TECNÓLOGICAS	34
3.3.1 HERRAMIENTAS DE PROGRAMACIÓN	34
3.3.1.1 UNITY 3D	34
3.3.2 HERRAMIENTAS DE DISEÑO	36
3.3.2.1 3DS Max.	36
3.3.2.2 Adobe Photoshop	36
3.3.3 HERRAMIENTAS DE MANEJO DE DATOS	36
3.3.3.1 PHP	36
3.3.3.2 MySQL	37
3.4 DISEÑO VISUAL PARA NIÑOS	38
4. METODOLOGÍA	41

5. HERRAMIENTA DE ENTRENAMIENTO COGNITIVO:	
CUATRO MINI-JUEGOS	46
5.1 JUEGOS DE MEMORIA	46
5.1.1 Secuencia de Frutas	46
5.1.2 SIMON	47
5.1 JUEGOS DE ATENCIÓN.	48
5.2.1 ¿Dónde están las Perlas?	48
5.2.2 Alimenta los Peces	49
6. DESARROLLO DEL PROYECTO	51
6.1 PLANEACIÓN	51
6.2 ESPECIFICACIÓN DE REQUISITOS	51
6.2.1 Descripción del sistema	52
6.2.2 Creación del Product Backlog	53
6.3 DESARROLLO POR SPRINTS: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN	57
6.3.1 SPRINT UNO	57
6.3.1.1 Página Principal	60
6.3.1.2 Sistema de Registro y Login	62
6.3.1.3 Menú Principal de la Aplicación	64
6.3.1.4 Pantalla de Selección de Categoría de Juego	66
6.3.1.5 Sistema de Tiempo	69
6.3.1.6 Resultado final del Sprint Uno	70
6.3.2 SPRINT DOS	71
6.3.2.1 Juego SIMON	74
6.3.2.2 Juego Secuencia de Frutas	76
6.3.2.3 Resultado final del Sprint Dos	78
6.3.3 SPRINT TRES	79
6.3.3.1 Juego ¿Dónde están las Perlas?	80
6.3.3.2 Resultado final del Sprint Tres	83
6.3.4 SPRINT CUATRO	84
6.3.4.1 Juego Alimenta los Peces	85

6.3.4.2 Resultado final del Sprint Cuatro	88
6.3.5 SPRINT CINCO	89
6.3.5.1 Pantalla información fin de partida	92
6.3.5.2 Pantalla de Logros y Trofeos	94
6.3.5.3 Pantalla de Estadísticas	96
6.3.5.4 Resultado final del Sprint Cinco	100
6.3.6 SPRINT SEIS	101
6.3.6.1 Modo Vista Completa	103
6.3.6.2 Pantalla de Tutoriales	104
6.3.6.3 Sistema de mensajes	105
6.3.6.4 Resultado final del Sprint Seis	106
7. PRUEBAS DE FUNCIONALIDAD Y USABILIDAD	109
CONCLUSIONES	114
RECOMENDACIONES	116
REFERENCIAS BIBLIOGRAFICAS	117
ANEXOS	119

## LISTA DE FIGURAS

	<b>Pág.</b>
Figura 1. Etapas de la memoria	22
Figura 2. Modelo Multi-almacén	24
Figura 3. Modelo memoria de trabajo (Baddeley and Hitch, 1974)	24
Figura 4. Componentes del modelo (Baddeley and Hitch, 1974)	26
Figura 5. Bucle Fonológico	26
Figura 6. 'Dichotic listening task'	29
Figura 7. Modelo de Atención Selectiva	29
Figura 8. Representación gráfica del Sprint	43
Figura 9. Visión General del Proceso Scrum	45
Figura 10. Pantalla del juego SECUENCIA DE FRUTAS	47
Figura 11. Pantalla del juego SIMON	48
Figura 12. Pantalla del juego ¿DÓNDE ESTÁN LAS PERLAS?	50
Figura 13. Pantalla del juego ALIMENTA LOS PECES	51
Figura 14. Estructura de una Historia de Usuario	57
Figura 15. Arquitectura Cliente Servidor de INGENIUM	62
Figura 16. Pantalla Principal Sistema de Registro y Login	63
Figura 17. Formulario de Registro	64
Figura 18. Iniciar Sesión	65
Figura 19. Menú Principal	66
Figura 20. Pantalla de Categorías de Juego	68
Figura 21. Pantalla Juegos de Memoria	68
Figura 22. Pantalla Juegos de Atención	69
Figura 23. Prefabricado Reloj	70
Figura 24. Navegación de escenas hasta el Sprint 1	71
Figura 25. Arquitectura Cliente-Servidor del juego SIMON	74

Figura 26. Arquitectura del juego SECUENCIA DE FRUTAS	77
Figura 27. Navegación entre escenas hasta el Sprint 2	78
Figura 28. Arquitectura Juego ¿DÓNDE ESTÁN LAS PERLAS?	81
Figura 29. Navegación entre escenas hasta el Sprint 3	82
Figura 30. Arquitectura del Juego ALIMENTA LOS PECES	86
Figura 31. Navegación entre escenas hasta el Sprint 4	87
Figura 32. Pantalla fin de partida	91
Figura 33. Arquitectura Cliente/servidor Escena Trofeos	92
Figura 34. Pantalla Trofeos	93
Figura 35. Arquitectura Cliente/Servidor Escena Estadísticas	96
Figura 36. Pantalla de Estadísticas: Resultados individual	96
Figura 37. Pantalla de Estadística: Resultado Comparativo	97
Figura 38. Navegación entre escenas hasta el Sprint 5	98
Figura 39. Arquitectura Cliente/servidor de la escena de mensajes	103
Figura 40. Navegación entre escenas: Aplicación completa.	104
Figura 41. Análisis de funcionalidades	109
Figura 42. Análisis de Usabilidad	110

## LISTA DE TABLAS

	<b>Pág.</b>
Tabla 1. Product Backlog o Pila del Producto	55
Tabla 2. Sprint Backlog, Sprint Uno	58
Tabla 3. Historias de Usuario, Sprint Uno	58
Tabla 4. Sprint Backlog, Sprint Dos	71
Tabla 5. Historias de Usuario, Sprint Dos	71
Tabla 6. Sprint Backlog, Sprint Tres	78
Tabla 7. Historias de Usuario, Sprint Tres	79
Tabla 8. Sprint Backlog, Sprint Cuatro	83
Tabla 9. Historias de usuario, Sprint Cuatro	83
Tabla 10. Sprint Backlog, Sprint Cinco	88
Tabla 11. Historias de Usuario, Sprint Cinco	89
Tabla 12. Sprint Backlog, Sprint Seis	99
Tabla 13. Historias de Usuario, Sprint Seis	99

## LISTA DE ANEXOS

	<b>Pág.</b>
ANEXO A. Glosario de Términos de Unity3D	117
ANEXO B. Pantalla de Tutoriales	119
ANEXO C. Imágenes de las pruebas de funcionalidad y usabilidad	123

## RESUMEN

### TÍTULO:

HERRAMIENTA SOFTWARE PARA EL ENTRENAMIENTO Y DESARROLLO DE LAS HABILIDADES COGNITIVAS: MEMORIA Y ATENCIÓN.<sup>\*</sup>

### AUTORES:

Lady Marcela Chaparro Arias  
Daniel Ordóñez Prieto<sup>\*\*</sup>

### PALABRAS CLAVES:

Entrenamiento cognitivo, Habilidades cognitivas, Memoria, Atención, Juegos 3D, Niños, Aprendizaje, Educación.

### DESCRIPCIÓN

Investigaciones muestran que ciertos tipos de ejercicios mentales pueden mejorar la salud y las funciones del cerebro, y potenciar el desarrollo de habilidades cognitivas las cuales están relacionadas con el aprendizaje y el intelecto, siendo posible mejorar de manera progresiva la memoria, la atención y otras áreas cognitivas a través de la resolución de ejercicios mentales, por lo tanto estimular estas habilidades a temprana edad es de vital importancia dado que las habilidades adquiridas en este tiempo son clave en el éxito en la educación superior.

El presente trabajo de grado ha sido desarrollado como medio de entrenamiento de la memoria y la atención a través de una galería de cuatro juegos desarrollados a partir de ejercicios. Estos juegos fueron desarrollados en 3D para una plataforma web. La herramienta es para uso en las aulas o personal, en donde el niño estimule las habilidades cognitivas mencionadas, en la interacción con ambientes virtuales. Tales juegos fueron diseñados a partir de una serie de ejercicios propuestos como resultado de investigaciones en campos como la psicología y la neurología.

Por otra parte en el trabajo con niños, se observa una gran motivación e interés por aprender, pero también por divertirse, pues el juego es parte inherente del hecho de ser niño y constituye en él su actividad central. El juego le aporta una larga serie de experiencias que responden a las necesidades específicas de las etapas del desarrollo. Es por esto que la idea de aplicar el juego en los procesos de aprendizaje resulta bastante útil ya que en este sentido, el aprendizaje se transforma en una experiencia placentera.

---

<sup>\*</sup> Proyecto de Grado.

<sup>\*\*</sup> Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Directora de Proyecto: Sonia Cristina Gamboa Sarmiento.

## ABSTRACT

### TITLE:

SOFTWARE TOOL FOR TRAINING AND DEVELOPMENT OF COGNITIVE SKILLS: MEMORY AND ATTENTION.\*

### AUTHORS:

Lady Marcela Chaparro Arias  
Daniel Ordóñez Prieto\*\*

### KEYWORDS:

Cognitive training, cognitive skills, memory, attention, 3D Games, Kids, Learning, Education.

### DESCRIPTION

Research shows that certain types of exercise can improve mental health and brain function, and promote the development of cognitive skills which are related to learning and intellect, making it possible to progressively improve memory, attention and other cognitive areas through the resolution of mental exercises, therefore stimulating these skills at an early age is vital because the skills acquired at this time are key to success in higher education.

This degree work has been developed as a mean for memory training and attention through a gallery of four games which development stems from practical exercises. These games were developed in 3 D for a web platform. The tool is for use in the classroom or personal, where cognitive skills of a child shall be stimulated, during interaction with virtual environments. These games were designed from a series of exercises as a result of research in fields such as psychology and neuroscience.

Moreover, it is well known that in working with children, there is a great motivation and interest in learning from their part, but also for fun, because the game is an inherent part of being a child and it is their core activity. The game gives a long series of experiences that match the specific needs of developmental stages. That is why the idea of applying the game in the learning process is quite useful since in this sense, learning becomes an enjoyable experience.

---

\* Thesis.

\*\* Faculty physical-mechanical engineering. School systems engineering and information technology. Project Director: Sonia Cristina Gamboa Sarmiento.

## 1. INTRODUCCIÓN

El presente proyecto desarrolla una aplicación en la web que incluye la reingeniería, diseño e implementación de cuatro mini juegos seleccionados de una vasta gama de aplicaciones destinadas al entrenamiento de habilidades cognitivas, producto de investigaciones en campos como la neuropsicología.

Para la selección de estas cuatro aplicaciones se realizó un marco de referencia y estado del arte como un primer paso que dio como resultado la elección de dos mini juegos encargados del entrenamiento de la memoria de trabajo y dos juegos dedicados al entrenamiento de la atención selectiva, puesto que estas habilidades están directamente relacionadas con el aprendizaje y el concepto abstracto de la inteligencia.

Este proyecto integró estos cuatro mini juegos que se encuentran en sistemas separados, acoplándolos y rediseñándolos para que su aspecto visual y gráfico estuviera dedicado a una audiencia infantil, específicamente a niños de básica primaria, ya que, aunque existen muchos programas de entrenamiento cognitivo y aplicaciones de este tipo, por lo general, la población seleccionada para el desarrollo son personas adultas pertenecientes a la mediana y tercera edad. El eje central de este proyecto por tanto, es brindar a la población infantil, una aplicación con la que se pueda realizar un entrenamiento cognitivo desde temprana edad, bajo un diseño que supla sus necesidades especiales como usuarios, donde prime ante todo la usabilidad, brindando al usuario pantallas y ambientes cómodos, intuitivos y fáciles de entender y navegar acompañados de escenarios que estimulan la imaginación.

Dada la extensión y magnitud que el proyecto alcanza, tanto funcional como gráficamente, se acogió una metodología de desarrollo ágil denominada Scrum, la cual representó grandes ventajas pues permitió un desarrollo incremental basado en iteraciones y revisiones, con una porción del software, funcional y sin errores, al final de cada iteración, permitiendo además flexibilidad ante nuevos requerimientos que surgieron, además de que se concentra en el desarrollo más que en la documentación.

Por otra parte, se utilizó para la realización de este proyecto, un motor creado específicamente para el desarrollo de juegos 3d: Unity 3d, que se convirtió en pieza clave para este proyecto, pues además de que permite programar en un lenguaje conocido por los autores, gráficamente permite incorporar y administrar modelos 3d con facilidad.

Este proyecto concluyó con una actividad en donde se probó la funcionalidad y usabilidad de la herramienta con niños de básica primaria del Colegio Fundación UIS.

## **2. OBJETIVOS**

### **2.1 OBJETIVO GENERAL**

Diseñar y desarrollar una herramienta software en la web que sirva de apoyo en el desarrollo y entrenamiento de las habilidades cognitivas: memoria y atención en niños de grados de básica primaria.

### **2.2 OBJETIVOS ESPECÍFICOS**

- Elaborar un estado del arte de las investigaciones que en psicología, neurología y educación han propuesto ejercicios para el desarrollo de la memoria y la atención.
- Seleccionar dos ejercicios para el desarrollo de la memoria y dos para el desarrollo de la atención, a partir del estado del arte propuesto.
- Determinar las funcionalidades de los ejercicios seleccionados que realizará la aplicación web, para dos tipos de usuario: jugador y tutor.
- Desarrollar el diseño de las funcionalidades de los ejercicios propuestos que más satisfaga los criterios de éxito determinados para el proyecto.
- Desarrollar los módulos y las interfaces de la aplicación web que permitan la interacción con los dos tipos de usuario.
- Aplicar y documentar pruebas de funcionalidad y usabilidad de la herramienta.

### 3. MARCO DE REFERENCIA

#### 3.1 HABILIDADES COGNITIVAS

Las habilidades cognitivas están relacionadas directamente con el conocimiento además de cómo éste es adquirido y estructurado a partir de la percepción constituyendo parte importante del aprendizaje y la inteligencia en general, siendo la base del pensamiento, de la acción y de la comunicación, entre las diferentes habilidades cognitivas se encuentran:

- Memoria.
- Atención.
- Lenguaje.
- Habilidades viso-espaciales.
- Velocidad de procesamiento.
- Resolución de problemas.

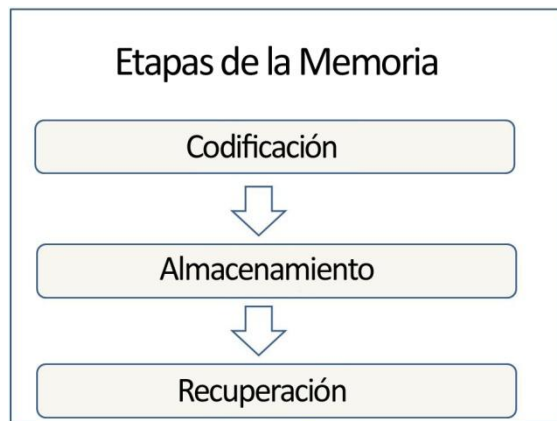
Los procesos y habilidades cognitivas pueden manifestarse de manera artificial o natural y ser conscientes o inconscientes y su estudio puede ser abordado desde diferentes enfoques como la neurología, psicología, sociología y la filosofía, este proyecto se concentrara en las habilidades cognitivas de memoria y atención desde una perspectiva neuropsicológica.

**3.1.1 Memoria.** Prácticamente todas las tareas que realizamos a diario, ya sea en el trabajo, la escuela o en casa, incluyendo la resolución de problemas o simplemente hablar requieren de la memoria; es por esto que el mantenimiento de la memoria es esencial para el eficiente funcionamiento de las actividades diarias.

Los procesos por los cuales nuestra mente nos permite almacenar y recuperar información se dividen en tres etapas clave (ver figura1):

- Codificación.
- Almacenamiento.
- Recuperación.

Figura 1. Etapas de la memoria



Tomado de <http://www.simplypsychology.org/memory.html>

**Codificación:** Cuando la información es adquirida debe ser codificada de manera que pueda ser almacenada de manera correcta, Por ejemplo, una palabra que se lee se puede almacenar cuando se codifica como un sonido o un significado.

Hay tres formas principales en las que la información puede ser codificada:

- Visual (imagen).
- Acústica (sonido).
- Semántica (significado).

**Almacenamiento:** El almacenamiento involucra diferentes variantes entre las que se incluye, donde se almacena la información, cuánto tiempo dura la memoria

(duración), que cantidad puede ser almacenada (capacidad) y qué tipo de información se almacena. El modo en que se almacena la información afecta la manera como se recupera, manifestando de esta forma dos estados de la memoria: memoria a corto plazo y memoria a largo plazo.

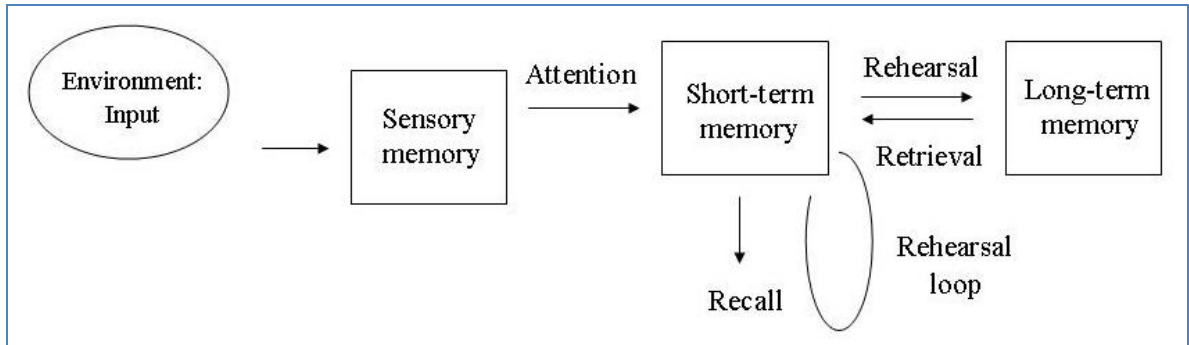
Se piensa que la memoria a corto plazo tiene un límite (la mayoría de los adultos pueden almacenar entre 5 y 9 elementos en su memoria a corto plazo sólo durante un tiempo breve de entre 0-30 segundos) pero la memoria a largo plazo puede durar toda la vida y se considera ilimitada.

**Recuperación:** La recuperación hace referencia a la obtención de la información almacenada, si no se puede recordar algo, probablemente no será recuperado. Cuando se pide recuperar información de la memoria, las diferencias entre memoria a largo y corto plazo quedan claras ya que la primera almacena y recupera secuencialmente mientras que la segunda almacena y recupera por asociación.

**3.1.1.1 Memoria de Trabajo.** Hace referencia al proceso de almacenamiento temporal y manipulación de la información, es la base del rendimiento en muchas otras actividades, incluyendo el razonamiento lógico y la comprensión de lectura.

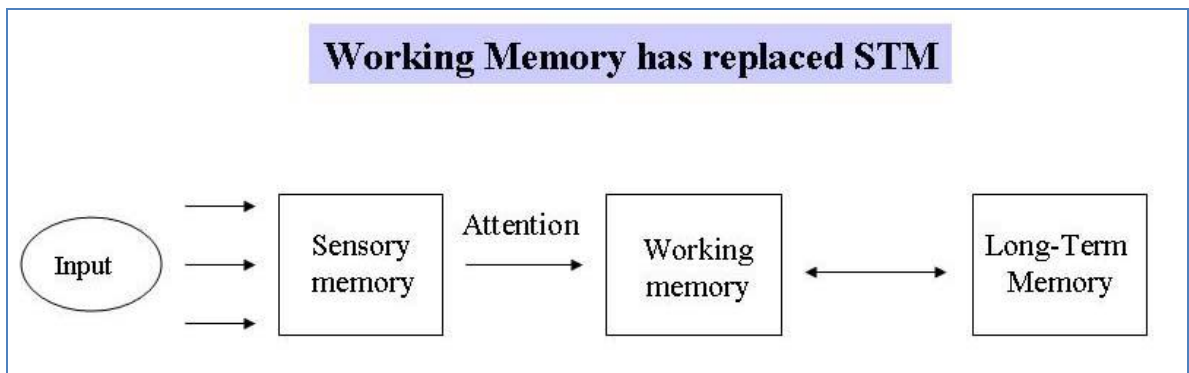
La memoria de trabajo está relacionada con la memoria a corto plazo, es más, el concepto de memoria a corto plazo que fue propuesto por Atkinson's y Shiffrin's (1968) en su modelo de multi-almacén (ver figura2.) es sustituido por el concepto de memoria de trabajo, propuesto por Baddeley y Hitch (1974) argumentando que el primero es muy simple, dado que de acuerdo al modelo de multi-almacén la memoria a corto plazo involucra al simple almacenamiento temporal de información, en contraste con el modelo de la memoria de trabajo que implica una combinación de almacenamiento y manipulación.

Figura 2. Modelo multi-almacén



De esta manera el modelo propuesto por Atkinson's y Shiffrin's (1968) es sustituido por el de Baddeley y Hitch (1974)

Figura 3. Modelo Memoria de Trabajo (Baddeley and Hitch, 1974)



La memoria de trabajo en contradicción con la memoria a corto plazo que es un único sistema que almacena la información, está constituida por diferentes subsistemas que almacenan diferentes tipos de información, y consta de un ejecutivo central<sup>1</sup> que controla y coordina el funcionamiento de los subsistemas: el bucle fonológico y el bloque de bocetos visuo-espacial.

- Ejecutivo central: Dirige todo el sistema y asigna los datos a los subsistemas. También se ocupa de las tareas como la aritmética mental y resolución de problemas.

<sup>1</sup> El ejecutivo central es la abstracción de un proceso que realiza el cerebro.

- bloque de bocetos visuo-espacial (ojo interior): Encargado de almacenar y procesar información de una manera visual o espacial y es usado para la navegación.
- El bucle fonológico (ver figura5): Es la parte de la memoria de trabajo que tiene que ver con el material escrito y hablado. Se puede utilizar por ejemplo para recordar un número de teléfono y Se compone de dos partes:
  - almacén fonológico (oído interno) - relacionado con la percepción del lenguaje, almacena información en forma de lenguaje (es decir, palabras habladas) durante 1-2 segundos, las palabras escritas deben ser convertidas en un código articulado (hablado) antes de que puedan entrar en el almacén fonológico.
  - Proceso de control articulatorio (voz interior) - ligado con la producción del habla. Actúa como una voz interna haciendo circular la información una y otra vez, es así que se puede recordar un número telefónico que se acaba de escuchar. Mientras se siga repitiendo, se puede retener la información en la memoria de trabajo.

Figura 4. Componentes del modelo (Baddeley and Hitch, 1974)

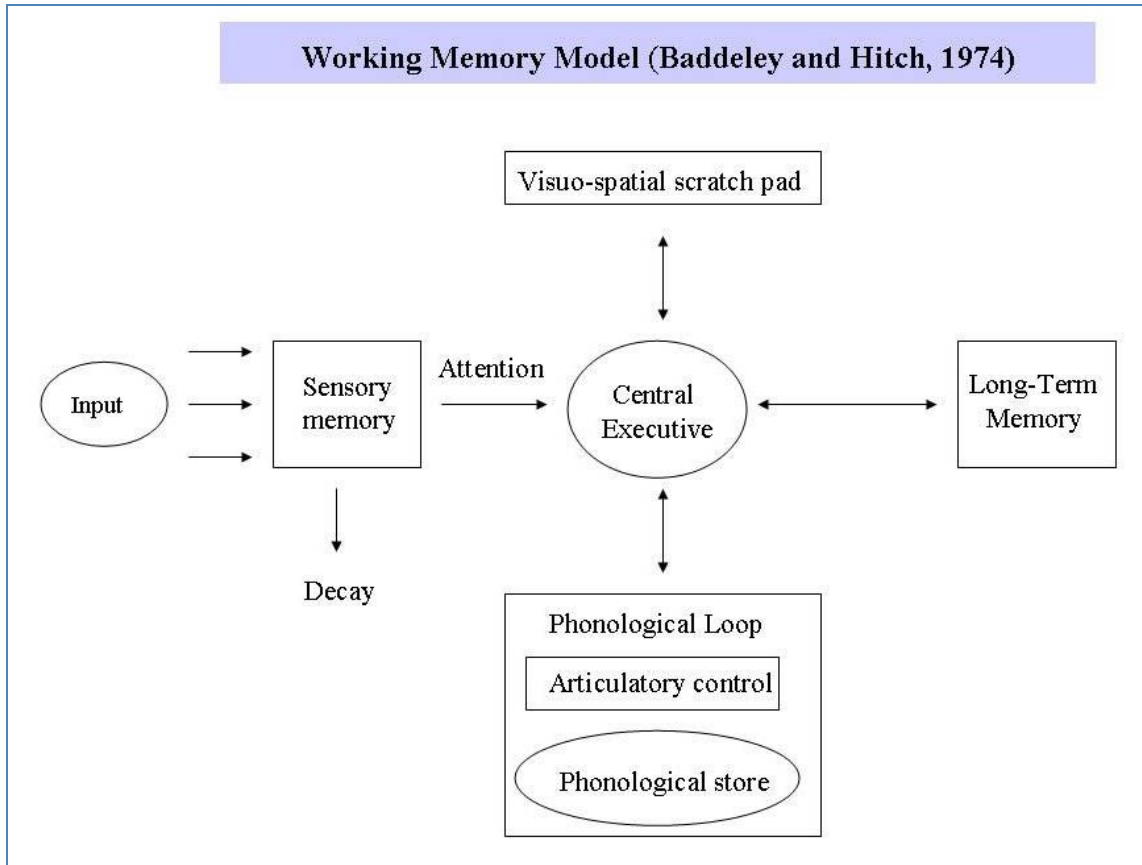
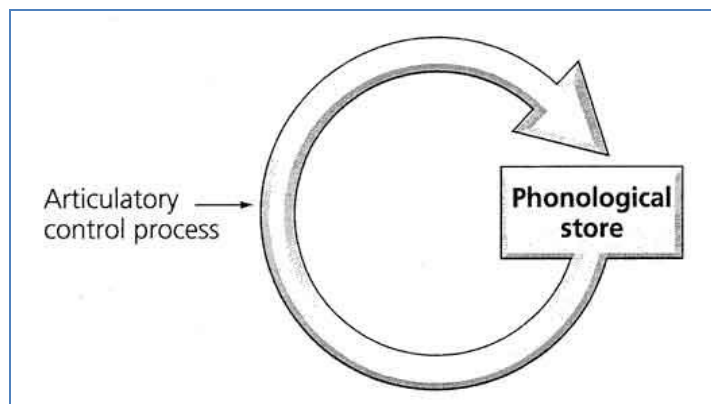


Figura 5. Bucle Fonológico



**3.1.2 Atención.** La atención es la habilidad que permite seleccionar la información importante sin tener en cuenta los datos innecesarios. Esta habilidad permite concentrarse sólo en las tareas pertinentes, y llevarlas a cabo de manera eficiente.

Es de gran importancia para el desarrollo de múltiples actividades cognitivas entre las que resalta la memoria y sobre todo la memoria de trabajo, constituyendo el puente entre los estímulos externos y el ejecutivo central (ver figura5), la atención por tanto es la encargada de seleccionar la información que entra al sistema de memoria a modo de cuello de botella o embudo.

**Categorías:**

- **Atención selectiva**  
Capacidad de focalizar voluntaria o involuntariamente la atención en un conjunto reducido de estímulos seleccionados entre una cantidad de candidatos en función de su intensidad y su relevancia.  
El entrenamiento de la atención selectiva, permite aumentar la capacidad de permanecer enfocado.
  
- **Atención dividida**  
Nos encontramos con esta tarea a lo largo de nuestra vida cotidiana. Por ejemplo, mientras conduce un automóvil y escuchar música o habla por teléfono.
  
- **Atención sostenida**  
La atención sostenida se refiere a un estado en el que la atención debe ser mantenida durante un período de tiempo. Esto se requiere en actividades en las que se debe monitorear continuamente una situación, puesto que acontecimientos imprevisibles y poco frecuentes pueden ocurrir. Un ejemplo de la atención sostenida se centraría en una pantalla de radar con el fin de

detectar inmediatamente cualquier bache repentino que podría ser señal de un buque o aeronave que se acerca.

**3.1.2.1 Atención Selectiva.** Los modelos de atención selectiva describen como las limitaciones para prestar atención funcionan a modo de cuello de botella, lo que evita que toda la información pase, siendo sólo una parte de esta seleccionada.

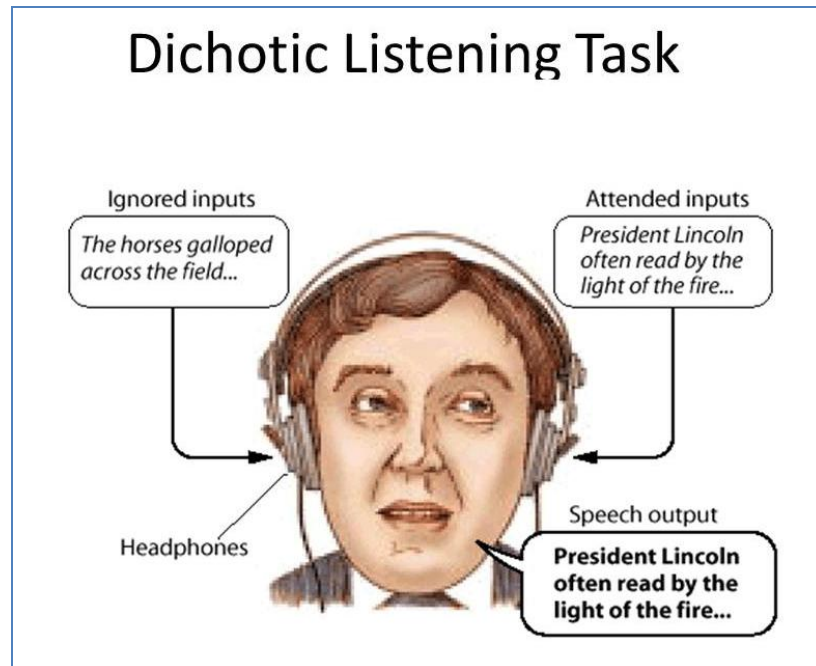
**Modelo Filtro de Broadbent de la atención selectiva:** Broadbent (1958) sostiene que todos los estímulos entran en un buffer sensorial<sup>2</sup> cuyas entradas pueden ser por ejemplo imágenes o sonidos. Dada la capacidad limitada para procesar información, se selecciona una de las entradas que pasará por un filtro para su posterior procesamiento, este filtro está diseñado para evitar que el sistema de procesamiento de información se sobrecargue. Las entradas no seleccionados inicialmente por el filtro permanecen brevemente en el buffer sensorial, y si no se procesan se desintegran rápidamente.

Al realizar un experimento denominado como 'dichotic listening task' el cual consiste en ponerle a una persona unos auriculares, enviarle un mensaje diferente a cada oído y pedirle que solo repita uno de estos mensajes (ver figura6.) Broadbent concluyo que sólo se puede poner atención a un canal a la vez, al concentrarse en un canal, la información del otro canal se perderá parcial o totalmente.

---

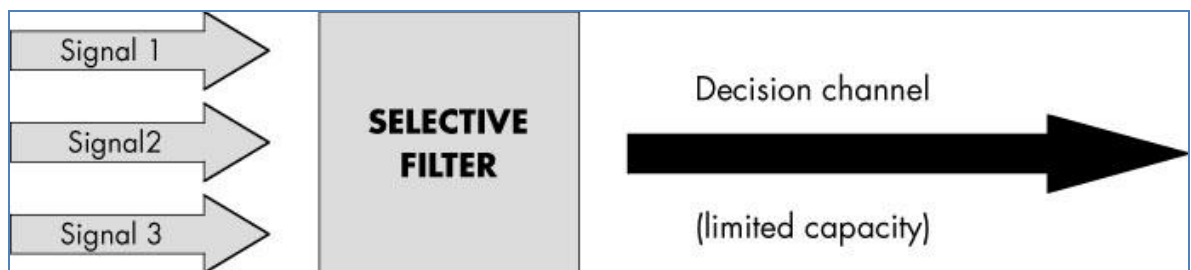
<sup>2</sup> El buffer sensorial es una memoria en la que se almacenan los estímulos sensoriales brevemente.

Figura 6. 'Dichotic listening task'



Por lo tanto la atención selectiva permite a la persona permanecer enfocado en un solo canal ignorando los estímulos de las demás entradas, lo que genera, que se pierdan rápidamente. Con el entrenamiento la capacidad de permanecer atento al canal de entrada deseado, puede aumentar, provocando así que sólo la información deseada pase a ser procesada por otras habilidades como la memoria de trabajo<sup>3</sup>.

Figura 7. Modelo de Atención Selectiva



<sup>3</sup> McLeod, S. A. (2008). Selective Attention. Tomado de <http://www.simplypsychology.org/attention-models.html>

## 3.2 PROGRAMAS DE ENTRENAMIENTO COGNITIVO

Un programa de entrenamiento cognitivo comprende un cumulo de ejercicios y tareas que al realizarlas, implican un esfuerzo mental que ayuda a mantener la mente en forma protegiéndola del deterioro. De este modo, el objetivo de un programa de entrenamiento cognitivo es estimular las diversas funciones o habilidades cognitivas que incluyen no sólo la memoria sino también la atención, el lenguaje y la velocidad de procesamiento.

Existe gran variedad de estos programas dispuestos en diferentes medios, algunos a través de internet, otros en dispositivos portátiles y también en consolas de video juego.

Los programas de entrenamiento se basan en el modelo cognitivo conductual. Es decir, parten de los fundamentos teóricos de la concepción dinámica de la inteligencia, la modificación cognitiva y el aprendizaje.

El cerebro, por su parte reacciona al entrenamiento cognitivo modificando las rutas que conectan a las neuronas. Esto genera efectos en el funcionamiento de los circuitos neurales y en la organización del cerebro, a esto se denomina como neuroplasticidad.

Algunos de estos programas de entrenamiento cognitivo son:

**3.2.1 Peak Genius.** Peak Genius.com es un sitio web dedicado al entrenamiento del cerebro, ejercicios para la mente y el incremento del IQ, en este sitio web se encuentra diferentes tipos de ejercicios y actividades fundamentadas en investigaciones realizadas por su equipo que dan como resultados una amplia gama de artículos que se pueden encontrar y consultar al navegar por la pagina.

La mayoría del material con el que consta este programa de entrenamiento, está pensado como actividades para desarrollar las habilidades cognitivas realizando ejercicios basados en la cotidianidad de las personas (como por ejemplo recordar números telefónicos) y además cuentan con una pequeña galería de juegos la

cual consta de 2 juegos de los cuales uno está dedicado al desarrollo de la memoria asociativa y las habilidades lógicas<sup>4</sup>.

**3.2.2 Lumosity.** Lumosity es un sitio web diseñado para ayudar a mejorar las habilidades cognitivas. De acuerdo al sitio, "La capacidad del cerebro humano para hacer nuevas amistades y adquirir nuevos conocimientos ha sido apreciada por cientos de años. Sin embargo, la capacidad del cerebro para reorganizarse fundamentalmente cuando se enfrenta a nuevos desafíos es un descubrimiento relativamente reciente. En respuesta a estos desafíos, el cerebro se adapta y cambia. Tomando el correcto tipo de ejercicio cerebral, este dará nueva forma a sí mismo para ser más eficiente y eficaz. Este poder de remodelación se conoce como neuroplasticidad. Lumosity ha adaptado esta investigación de vanguardia en la neurociencia y el conocimiento en un accesible y agradable programa de formación, que estimula el cerebro y conduce a una mayor capacidad cognitiva y mejor salud."

El grupo de trabajo de Lumosity afirma que en menos de 10 minutos al día, usted puede mejorar su memoria y atención.

El servicio de Lumosity es gratis por 7 días. Lumosity también proporciona retroalimentación y seguimiento detallados de mejora en su nivel.

Es popular porque hace divertido el aprendizaje. Hay una sección escolar en Lumosity que está específicamente dirigida a estudiantes, la cual está demostrada que mejora los resultados en las pruebas estandarizadas a estudiantes. Otras secciones están destinadas específicamente para ciertas áreas, tales como las personas con trastorno por déficit de atención con hiperactividad (TDAH), lesiones cerebrales traumáticas o con necesidades especiales.

Sus usuarios han afirmado que después de usar el programa de entrenamiento adquieren los siguientes beneficios:

---

<sup>4</sup> Para mayor información ver <http://www.peakgenius.com/braingames.html>

- Se piensa más claro y más rápido
- Mejora la memoria para los nombres, números, direcciones, etc.
- Incremento de la alerta y la conciencia
- Estado de ánimo elevado
- Mejor concentración en el trabajo o mientras se conduce

El programa de entrenamiento cerebral de Lumosity es el resultado de una síntesis de los trabajos cumulativos de investigadores en todo el mundo en los campos de la neurociencia, la psicología, y la ciencia cognitiva. Sus investigaciones han formado la base para crear los juegos y los ejercicios cerebrales para el entrenamiento de las habilidades cognitivas como:

Memoria:

Los ejercicios de memoria se centran en los tipos de memoria que son importantes para la vida diaria tales como la memoria de trabajo, la memoria espacial y la memoria de nombres y caras.

Atención:

La capacidad de concentrarse en información pertinente y a la vez ignorar información irrelevante, se mejora por medio de juegos que requieren el uso de la atención visual, la atención selectiva y un cambio de enfoque.

Velocidad de Procesamiento:

La velocidad de procesamiento determina la rapidez con la cual una persona puede pensar, percibir información sensorial, o manejar otros procesos cognitivos tales como entender el lenguaje<sup>5</sup>.

---

<sup>5</sup> Para mayor información ver <http://www.lumosity.com/>

**3.2.3 MIND360 Play Smart.** MIND360 es un sitio web el cual se encuentra actualmente es un versión beta sin embargo ya cuentan con una buena galería de juegos los cuales se encuentran divididos según las habilidades cognitivas a las cuales se dedican encontrándose entre estas la memoria, la atención, funciones ejecutivas, la lógica, y la percepción visual.

Este entrenamiento cognitivo ofrece al usuario la promesa de encontrar una interfaz fácil de usar, gráficos cautivadores y juegos diseñados para animar. Se Indica que 10 minutos al día es todo lo que se necesita para garantizar una aptitud mental optima<sup>6</sup>.

Se ofrece además programas de entrenamiento personales en el que se garantiza un rápido progreso en el aumento de las habilidades cognitivas y un programa en el cual se tiene acceso a un entrenador personal y “feedback” permanente garantizando a los usuarios que se beneficien plenamente. Además de contar con un foro en el que se tiene de acceso a Psicoanalistas.

El personal detrás de los juegos cuenta con un impresionante equipo de psicólogos cognitivos que trabajan junto a expertos desarrolladores de juegos en línea para crear actividades únicas. El resultado es un entorno interactivo de juegos en línea intelectualmente estimulante, tanto en formas novedosas como placenteras.

**3.2.4 Train your Brain with Dr Kawashima.** Desarrollado exclusivamente para PC, consta con una amplia de ejercicios distribuidos en cuatro categorías:

- Memoria
- Lógica
- Visual
- Procesamiento matemático

---

<sup>6</sup> Para mayor información ver <http://www.mind360.com>

Que en total suman 30 mini aplicaciones dedicadas al entrenamiento de habilidades cognitivas, el programa ofrece al usuario la opción de realizar una prueba diaria la cual está constituida por un grupo juegos de las diferentes categorías, esta prueba es utilizada para monitorear el progreso del usuario y dura alrededor de diez minutos, sin embargo el jugador puede utilizar los juegos cuantas veces quiera.

Cabe destacar de este sistema la gran calidad de su apartado grafico, el fácil acceso a las ayudas, el fácil entendimiento de la interfaz y la navegación que hace posible que sea utilizado por todo tipo de público y el detalle seguimiento que se brinda por parte de las estadísticas, además de contar con el respaldo y la experiencia del doctor kawashima, el cual es reconocido por su larga trayectoria en el campo del entrenamiento cognitivo desde un enfoque neurológico siendo la imagen de un buen numero de aplicaciones de este tipo, entre las que destacan las publicaciones de Nintendo, “Brain age” para su consola portable nintendo DS.

### **3.3 HERRAMIENTAS TECNOLÓGICAS**

En este apartado se mencionan y describen las herramientas utilizadas durante la realización de este proyecto.

#### **3.3.1 HERRAMIENTAS DE PROGRAMACIÓN**

**3.3.1.1 UNITY 3D.** Es un motor para el desarrollo de juegos principalmente en 3D, sin embargo, dadas sus características es posible crear juegos 2D y es utilizado además, para realizar aplicaciones que requieren un alto contenido gráfico.

Unity 3d consta de una amplia variedad de herramientas que permiten que el desarrollo de video juegos sea una experiencia ágil, proporcionando soporte para la simulación de colisiones, física, sombra, luces, animaciones, entre otras además de permitir la conectividad con base de datos, manejo de archivos XML y

el soporte multi-jugador. Una de las características que llama más la atención de Unity 3d es la posibilidad de crear juegos para múltiples plataformas tales como:

- Web Browser
- Mac y PC
- Dispositivos IOS
- Dispositivos Android
- Nintendo wii
- Play Station3
- Xbox 360

Su API permite al desarrollador elegir entre tres lenguajes de programación, C#, JavaScript, y Boo que es un dialecto de Phytion, dando la posibilidad de que los scripts construidos en estos se comuniquen indistintamente.

El desarrollo que permite unity es descrito como programación gráfica, lo cual causa que sea muy fácil ubicar y controlar los elementos que se dispongan en cualquier aplicación.

En este proyecto, Unity 3d fue la herramienta principal de desarrollo debido a que está diseñada para ofrecer un entorno de desarrollo ágil, además de tener la opción de utilizar el lenguaje de programación C# en el cual los autores tienen experiencia. De acuerdo con esto, es importante introducir una serie conceptos básicos de este motor, utilizados durante el desarrollo de este proyecto. Ver anexo A.

Otro punto a favor para que se eligiera Unity 3d como herramienta de desarrollo, es que además de ser un motor poderoso para hacer video juegos, cuenta con una licencia libre; la versión libre utilizada en este proyecto es la 3.5.2.

### **3.3.2 HERRAMIENTAS DE DISEÑO**

**3.3.2.1 3DS Max.** Es una herramienta de modelado, animación y renderización en 3D que permiten a los artistas y los diseñadores dedicar más energía a la creatividad en lugar de a las dificultades técnicas. Ofrece herramientas especializadas a los desarrolladores de juegos, creadores de efectos visuales, diseñadores de gráficos de movimiento y otros profesionales de la creatividad que trabajan en el diseño de medios.

En este Proyecto todos los elementos y modelos 3D fueron elaborados en 3DS max en su versión 2012 debido a su detallada documentación y facilidad de uso que sumados a la compatibilidad con Unity 3D hacen de esta la herramienta ideal.

**3.3.2.2 Adobe Photoshop.** Programa destinado para la edición, retoque fotográfico y pintura a base de imágenes de mapa de bits. Su nombre en español significa literalmente "tienda de Fotos" pero puede interpretarse como "taller de foto". Su capacidad de retoque y modificación de fotografías le ha dado el rubro de ser el programa de edición de imágenes más famoso del mundo.

Dada la experiencia previa de los autores de este proyecto en el uso de Adobe Photoshop este fue seleccionado como software para la edición y creación de las texturas usadas en los modelos 3d, los fondos y los elementos de la GUI.

### **3.3.3 HERRAMIENTAS DE MANEJO DE DATOS**

**3.3.3.1 PHP.** Es un lenguaje de programación de uso general de script del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor

web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. PHP puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

En este proyecto el uso de php fue necesario para comunicara a la aplicación de unity 3d con la base de datos. De esta manera una serie de archivos php sirven como intermediario para consultar y actualizar la base de datos en el servidor.

**3.3.3.2 MySQL.** MySQL es la base de datos open source más popular y, posiblemente, mejor del mundo. Su continuo desarrollo hace de MySQL un competidor cada vez más directo de gigantes en la materia de las bases de datos. MySQL es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales.

**Características:**

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Condición de open source de MySQL hace que la utilización sea gratuita y se puede modificar con total libertad
- Se puede descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones
- Es una de las herramientas más utilizadas por los programadores orientados a Internet.

- Infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación.
- MYSQL, es considerado como el más rápido de Internet
- Gran rapidez y facilidad de uso
- Infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación.
- Fácil instalación y configuración.

### **3.4 DISEÑO VISUAL PARA NIÑOS**

Los niños de edad escolar pueden ser considerados como nuevos lectores dado que apenas están aprendiendo a lidiar con el idioma por lo tanto a la hora de diseñar interfaces para estos usuarios se debe evitar el uso de recursos que no se entiendan, las imágenes utilizadas deben ser de fácil comprensión evitándose el uso de imágenes que puedan tergiversar su sentido en los escenarios.

La comprensión de una imagen depende de la cantidad de detalles significativos y reconocibles que incluya, pues son estos detalles los que permiten reconocerlas, aunque el exceso de los mismos también podría interferir negativamente en su comprensión, el uso de símbolos, iconos donde su empleo es el fruto del acuerdo de una cultura o época dada, será asimilado después de un acercamiento didáctico hacia ese tipo de símbolos.

Para ser realmente clara cada imagen debe representar tan solo una idea, una sola situación en contraste, las imágenes que intentan transmitir varios mensajes, se tornan confusas, además de que las imágenes sean claras y fácilmente reconocibles deben ser llamativas, originales, contrastantes y fantasiosas, todo esto implica conocer el nivel de relación y manejo por parte de los destinatarios.

Un diseño consistente y llamativo, puede brindar presentaciones más rápidas, reduce los errores de supuestos y aumenta el éxito en la identificación de ítems relevantes.

Se busca diseñar interfaces que sean predecibles, comprensibles y controlables, en donde se expongan las reglas claras y concisas, estructuradas de manera concreta, en oraciones cortas, en tono amigable, tipografía comprensible, tamaño grande. Los niños de edad escolar responden de manera positiva a estímulos auditivos y visuales, por lo tanto es aconsejable que hayan estímulos permanentes cuando hacen bien las labores o para mostrar los errores.

Diseñar señales visuales y/o sonoras que refuercen la navegación o los ejercicios en las diferentes escenas, también presentar las diferentes secciones de la interfaz organizadamente, clasificadas y jerarquizadas por grupos de tareas para ayudar a desarrollar las habilidades de agrupar, diferenciar, y de relacionar que los niños entre 7 y 12 años de edad mental empiezan a asimilar en la escuela. Y se aconseja también ponerle nombre a cada menú o grupo y a sus componentes<sup>7</sup>.

Los colores y las formas son elementos importantes para los niños. El uso de colores vivos, saturados, llama más la atención de un niño que de un adulto que prefiere colores más calmos y tranquilos. También las formas pueden ser factores muy atractivos para captar a un joven espectador.

Las metáforas de navegación geográfica funcionan: El objetivo del uso de la metáfora en el diseño es hacer familiar y comprensible lo desconocido. Además el uso de metáforas visuales (ya sean geográficas o de otro tipo) es una acertada

---

<sup>7</sup> ESCOBAR, G. G. (1999). *LUDOMÁTICA: DISEÑO GRÁFICO PARA AMBIENTES EDUCATIVOS LÚDICOS, CREATIVOS E INTERACTIVOS*. SANTAFÉ DE BOGOTÁ, DC.

decisión de diseño para niños, ya que éstos poseen menos conocimientos, habilidades y capacidad para la lectura.

## 4. METODOLOGÍA

A la hora de desarrollar un producto software se hace necesario utilizar una serie de técnicas, herramientas, procedimientos y documentación que permitan elaborar eficientemente el producto; ante esta necesidad, surgen las metodologías de desarrollo de software. La metodología conlleva a una selección de técnicas concretas acerca del procedimiento para realizar las distintas etapas del desarrollo de software como la planeación, recolección y análisis de requerimientos, diseño, implementación, pruebas de funcionamiento, documentación y mantenimiento.

Los requisitos de un software a otro, son tan variados y cambiantes, que han dado lugar a que exista una gran variedad de metodologías para la creación de software. Estas se pueden clasificar en dos grandes grupos: Metodologías pesadas y Metodologías ágiles; las primeras se centran en la definición detallada de los procesos y tareas, herramientas a utilizar, y requieren una extensa documentación; las segundas están orientadas a la interacción con el cliente y el desarrollo incremental del software, mostrando versiones parcialmente funcionales del software al cliente en intervalos cortos de tiempo, permitiéndole a éste, sugerir cambios en el producto según se va desarrollando.

Para el desarrollo de este proyecto se optó por utilizar una metodología ágil, pues se considera que en el desarrollo de un producto para niños, es apropiado interactuar con los clientes más potenciales del producto: profesores, padres y niños. La metodología escogida es una de las más notables dentro de las ágiles y se conoce como SCRUM.

Scrum se consideró apropiada para este proyecto porque emplea un desarrollo incremental basado en iteraciones y revisiones, con una porción del software, funcional y sin errores, al final de cada iteración, lo que nos facilitó visualizar el

proyecto día a día; un desarrollo donde los requerimientos y soluciones evolucionan a la par, lo que nos permitió flexibilidad y adaptación ante nuevos requerimientos que podían surgir. Además de esto, Scrum maneja principios ventajosos como la simplicidad, refiriéndose a realizar sólo los artefactos necesarios, aportando productividad y calidad al producto y sin necesidad de una documentación extensa.

Otro principio fundamental de SCRUM es la motivación, compromiso y responsabilidad del equipo por autogestión.

SCRUM define tres roles: El Product Owner (dueño del producto), el Scrum Master (o facilitador) y el Scrum Team (equipo de desarrollo). El product Owner representa a todos los interesados en el producto final, entre ellos están los niños que son los usuarios finales, entidades educativas, profesores y padres de familia que son los clientes. El Scrum Master no es el líder del equipo, porque el equipo se auto-organiza, pero es quien se asegura que las reglas se cumplan. El Scrum Team es el equipo cuya responsabilidad es entregar el producto y realizar el trabajo de análisis, diseño, desarrollo, pruebas y documentación, este equipo lo conforma los autores del presente proyecto. Dentro de SCRUM también se definen roles auxiliares que se involucran parcialmente con el proceso de desarrollo.

Los roles definidos por Scrum son los necesarios para desarrollar un producto tecnológico, pero un producto dirigido a niños requiere de la colaboración de otras disciplinas. Es así que el desarrollo de software para niños es un trabajo interdisciplinar en el que tienen cabida niños, especialistas en educación, diseñadores gráficos, y por supuesto diseñadores de software, que en SCRUM es el Scrum Team. Los especialistas en educación y diseñadores gráficos son roles auxiliares en este proyecto.

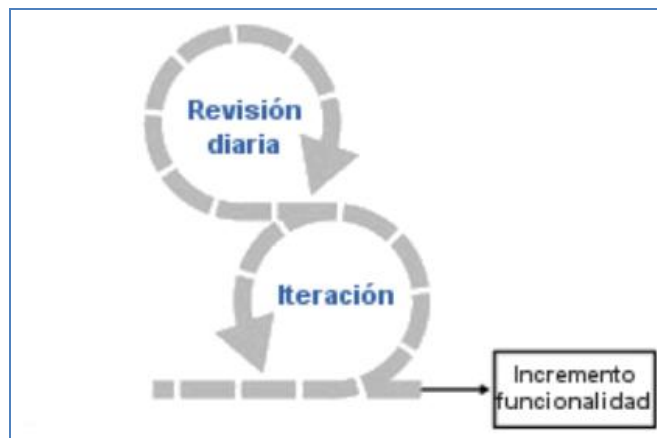
Es importante mencionar que durante el desarrollo de este proyecto se contó con la opinión de niños acerca de qué cosas querían ver en el software y que les

motivaba. Se contó además con la colaboración de un profesional y un estudiante de Diseño Gráfico de la UDI, y con la colaboración de la directora de este proyecto Sonia Gamboa como especialista en Educación.

**Estructura Central de Scrum.** La estructura central de SCRUM se basa en iteraciones incrementales de desarrollo denominadas “Sprint”. Cada Sprint debe tener una duración de 15 a 30 días y SCRUM gestiona su evolución a través de reuniones breves diarias en las que el equipo revisa el trabajo realizado el día anterior y el previsto para el día siguiente. Además de las reuniones diarias durante el sprint, al iniciar cada sprint, el equipo revisa el trabajo pendiente del proyecto y selecciona la parte que terminará como un incremento de funcionalidad incorporado al software al terminar el sprint. La Figura 8 representa este proceso.

Al finalizar cada iteración se lleva a cabo una revisión con todas las personas implicadas en el proyecto y se puede inspeccionar y evaluar la parte del producto operativa que se incorpora al software.

**Figura 8. Representación gráfica del Sprint**



## Elementos del desarrollo Scrum

### Las reuniones:

- Planificación de sprint: hace referencia a la jornada de trabajo previa al inicio de cada sprint en la que se determina cuál va a ser el trabajo y los objetivos que se deben cumplir en esa iteración.
- Reunión diaria: breve revisión del equipo del trabajo realizado hasta la fecha y la previsión para el día siguiente.
- Revisión de sprint: Análisis y revisión del incremento generado.

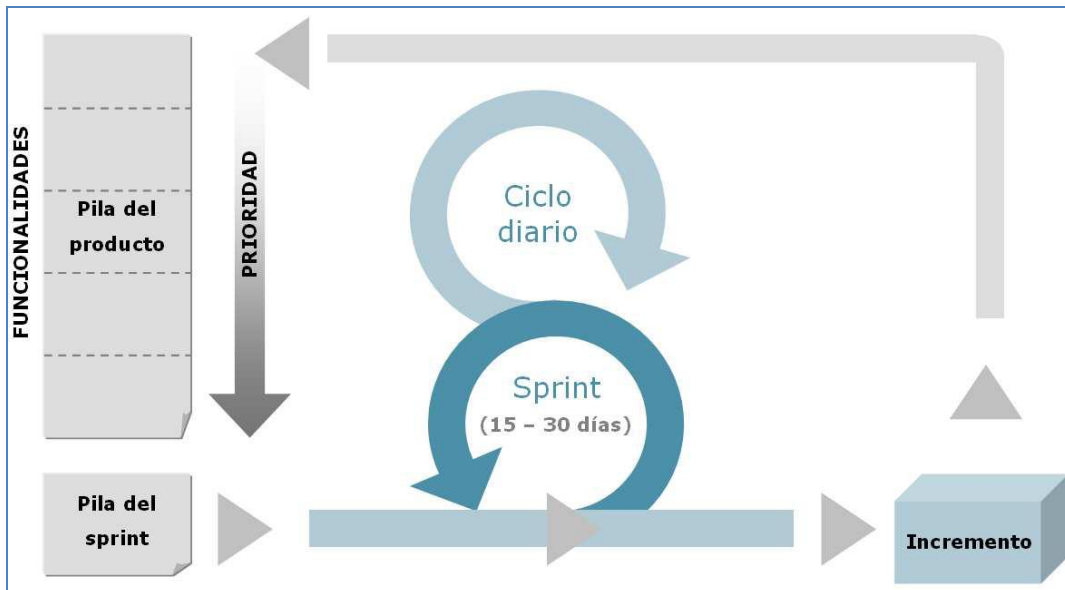
### Los elementos:

- Product Backlog: hace referencia a la lista de requisitos de usuario que se origina con la visión inicial del producto y va creciendo y evolucionando durante el desarrollo. Para la creación del product backlog es generalmente usado un formato de trabajo llamado Historia de Usuario<sup>8</sup>, sin que haya mayores problemas en utilizar Casos de Uso.
- Sprint Backlog: lista de los trabajos que debe realizar el equipo durante el sprint para generar el incremento previsto.
- Incremento: resultado de cada sprint.
- Burn Down Chart: es una gráfica que muestra el avance del proyecto.

---

<sup>8</sup> La Historia de Usuario es una descripción breve de una funcionalidad del software.

Figura 9. Visión General del Proceso Scrum



## 5. HERRAMIENTA DE ENTRENAMIENTO COGNITIVO: CUATRO MINI-JUEGOS

### 5.1 JUEGOS DE MEMORIA

**5.1.1 Secuencia de Frutas.** Este juego está basado en uno de los mini-juegos del programa de entrenamiento cerebral: "*Train your Brain with Dr. Kawashima*" de la categoría de memoria, llamado "*Picture Sequence*". "*Train your Brain with Dr. Kawashima*" fue publicado en 2009 por Chimera Entertainment para PC y Mac del neurocientífico japonés Ryuta Kawashima conocido por su serie de video juegos de entrenamiento cerebral.

Secuencia de Frutas muestra en la pantalla una serie de imágenes -cada una de las cuales es una fruta- por un lapso de tiempo hasta que son retiradas automáticamente; en este tiempo el jugador debe memorizar la secuencia de imágenes que se le mostró; posteriormente se le mostrará al jugador las imágenes que se desplegaron en la pantalla en un orden arbitrario, pero ahora cada una de las imágenes es un botón que el jugador presionará para intentar repetir la secuencia que le fue presentada antes; esto se deberá hacer en un tiempo asignado. Si el jugador se equivoca en un elemento de la secuencia se considera que perdió esa ronda y se le mostrará otra secuencia; de la misma manera si el jugador acierta la secuencia, se le mostrará otra, pero esta vez acumulará los puntos que ganó en la ronda reciente.

El nivel de dificultad debe aumentar instando al jugador a memorizar más imágenes cada vez que logre acertar dos rondas seguidas. De modo similar, el nivel de dificultad disminuirá cuando el jugador falla en dos rondas seguidas.

Figura 10. Pantalla del Juego SECUENCIA DE FRUTAS



**5.1.2 SIMON.** Este juego está basado en el juego electrónico clásico SIMON® de habilidad de memoria creado por Ralph Baer en 1978, ahora propiedad de la compañía internacional Hasbro dedicada a la fabricación de juegos y juguetes. SIMON® consta de cuatro botones cada uno de ellos de un color diferente (rojo, azul, verde y amarillo) y que emite un sonido propio. El jugador debe pulsar los botones en la misma secuencia que el juego indique.

En este proyecto, al iniciar el juego se le muestra al jugador cuatro botones de distintos colores dispuestos en forma circular; cuando se presiona, cada botón emite un sonido particular y se ilumina. El juego genera una secuencia aleatoria de botones que comienza con un único botón para que el jugador la repita, y a medida que él vaya acertando, ésta se irá incrementando con un nuevo botón aleatorio de manera que cada vez se hará más difícil de replicar. El jugador pierde cuando comete un fallo, acumulando los puntos obtenidos hasta ese momento; de

modo que acumulará más puntos en la medida en que consiga repetir correctamente la secuencia un mayor número de veces.

La figura 11 muestra la pantalla del juego SIMON. En ella puede apreciarse el modelo 3d del prefabricado hongo cuatro veces actuando como botones. En la parte superior izquierda se encuentra el botón para volver al menú anterior. En la parte inferior derecha se encuentran dos botones. El superior lleva al tutorial que explica cómo se juega, aunque en esta fase de desarrollo todavía no estaba programado. El botón inferior tiene la funcionalidad de pantalla completa, tampoco programada hasta este punto.

Figura 11. Pantalla del Juego SIMON



## 5.1 JUEGOS DE ATENCIÓN.

**5.2.1 ¿Dónde están las Perlas?** “¿Dónde están las Perlas?” es un juego basado en el mini-juego de atención: *¿Where are the eggs?* del video juego “*Train your Brain with Dr. Kawashima*”.

“¿Dónde están las Perlas?” presenta en la pantalla unos botones representados por almejas las cuales tienen una posición inicial y algunas de ellas contienen una perla en su interior. El jugador tendrá unos segundos para retener las posiciones de las almejas con perla pues seguidamente las almejas se cerrarán y cambiarán de posición varias veces entre ellas; el jugador deberá estar atento a las almejas que tienen perla para que acabada la animación haga clic a la(s) almeja(s) que él crea que contengan la perla en su interior; si logra acertar, acumulará puntos y continuará a la siguiente ronda que será de mayor dificultad si acierta dos rondas consecutivas. Si no logra acertar la ronda, continuará a otra ronda de igual dificultad y no acumulará puntos. Si no logra acertar dos rondas consecutivas, continuará a una ronda de menor dificultad.

Figura 12. Pantalla del Juego ¿DÓNDE ESTÁN LAS PERLAS?



**5.2.2 Alimenta los Peces.** Este juego está basado en el mini-juego “Playing Koi” del programa de entrenamiento cerebral Lumosity, disponible en su sitio Lumosity.com junto con otros juegos para entrenar y mejorar la memoria, la atención y la velocidad de procesamiento que según el sitio, son creados por científicos especializados en la materia. Lumosity es el primer programa general de entrenamiento cerebral de Lumos Labs, empresa de investigación y desarrollo en neurociencia cognitiva.

Al iniciar “Alimenta los Peces”, se muestra en la pantalla una cantidad de peces en su hábitat que nadan siguiendo caminos aleatorios. El jugador dispone de una provisión de comida exacta para alimentar a cada pez solo una vez. El juego impide al jugador alimentar a un siguiente pez de manera inmediata, en cambio el jugador debe esperar unos segundos para hacerlo sin perder de vista los peces

que aún no ha alimentado. Si el jugador acierta, el número de peces aumenta en uno para la siguiente ronda.

Figura 13. Pantalla del Juego ALIMENTA LOS PECES



## **6. DESARROLLO DEL PROYECTO**

Este capítulo describe el proceso de desarrollo de la herramienta desde la fase de planeación, abordando Scrum por completo.

Como se ha mencionado anteriormente, el producto desarrollado en este proyecto es un programa computarizado de entrenamiento cognitivo, compuesto de cuatro ejercicios o juegos en el que el usuario principal puede llevar un control sobre su evolución en los procesos de memorizar y estar atento. Se describe el sistema más a fondo en el apartado “Descripción del Sistema” de este capítulo.

### **6.1 PLANEACIÓN**

Antes de abordar el proceso Scrum, se realizó una fase de planeación en la que se hizo un estado del arte sobre aplicativos que incluye algunos de los diferentes aplicativos existentes para la web y para escritorio, e incluso electrónicos, similares al que se desarrolla con este proyecto. Se hizo también una consulta teórica de la memoria y la atención como habilidades y procesos cognitivos y de las zonas del cerebro que se activan al ejecutar estos procesos. Así mismo, se consulto acerca del cerebro y de la capacidad que tiene para mejorar dichas habilidades por medio de entrenamiento cognitivo. Por último, se investigó sobre software, diseño y ambientes virtuales para niños.

### **6.2 ESPECIFICACIÓN DE REQUISITOS**

Para dar inicio a esta fase, se hizo una identificación de cuáles debían ser la características del software a realizar, teniendo como referencia algunas aplicaciones similares ya existentes y en general apoyándonos en la investigación

hecha en la fase de planeación. Partiendo de estas investigaciones y de las funcionalidades que los autores querían como innovadoras para el software, se realizó en su momento una descripción general del sistema, una visión de lo que se quería de éste y que además sirvió para sacar los requisitos funcionales y no funcionales.

**6.2.1 Descripción del sistema.** Se quiso desarrollar una herramienta que sirva de apoyo al entrenamiento cognitivo a través de cuatro aplicaciones basadas en varios programas de entrenamiento cerebral que son el resultado de una síntesis de los trabajos de investigadores en todo el mundo en los campos de la neurociencia, la psicología, y la ciencia cognitiva. La herramienta estará publicada en la web, y permitirá el registro de nuevos usuarios, así, cada usuario tendrá una cuenta a fin de que pueda acceder a información suya.

De las cuatro aplicaciones (mini-juegos), dos estarán dedicadas al entrenamiento de la memoria y dos a la atención. La herramienta se concibe como un micro-mundo virtual en donde las cuatro aplicaciones están contenidas en una mayor, la cual permita estar en pantalla completa.

El diseño de los mini-juegos es pensado para un público en particular: niños de básica primaria, aunque podrán ser usados por cualquier público.

Se deben diferenciar claramente las categorías de juegos (memoria y atención) por medio de un menú donde se pueda acceder a ellos. Cada uno de los cuatro mini-juegos será explicado mediante un tutorial que muestra al jugador cómo se juega, el cual podrá ser accedido por el jugador en cualquier momento dentro del juego y que además se desplegará automáticamente en el primer acceso. Adicionalmente cada mini-juego provee realimentación y seguimiento detallado de la mejora del jugador en esa habilidad, expresado en trofeos, logros y puntos, los cuales son visibles una vez finaliza el juego y se almacenan para ser consultados posteriormente por el jugador en cualquier momento; si el jugador juega varias

veces durante el día el mismo juego, sólo se almacenará la puntuación más alta obtenida durante el día.

Cada juego tendrá la capacidad de adaptar su nivel de complejidad al rendimiento del jugador, lo que implica que al principio el nivel de complejidad será bajo e irá aumentando en la medida en que el jugador muestra evolución y se hace más fuerte en esa habilidad.

Hay dos tipos de usuarios en el sistema, tutor y jugador. El tutor es un usuario que tiene conocimientos en el área cognitiva y/o en el entrenamiento cognitivo y cuya labor en el sistema es acompañar el entrenamiento de los jugadores; por tal motivo, se podrá asociar un tutor a un jugador y existirá un módulo visible por el tutor en el que esté contenida la información del progreso de los jugadores asociados, en forma de diagramas para cada uno de los mini-juegos; en el sistema, un tutor puede acompañar a varios jugadores, pero cada jugador puede ser acompañado sólo por un tutor. El jugador es el usuario final del sistema para el cual se desarrollan los juegos, y como se vislumbra que estos usuarios en su mayoría serán niños de básica primaria, gráficamente deberá ser atractivo y agradable para ellos. Para este usuario deberá existir un módulo en el que pueda ver su desempeño de manera lúdica, de modo que contribuya a su entretenimiento y estímulo, su desempeño se verá caracterizado por trofeos, puntajes y logros obtenidos para cada uno de los mini-juegos.

Al ser una herramienta en la web, además de permitirle al tutor llevar el control del progreso del jugador de manera individual, también le permitirá comparar y situar su desempeño respecto a los demás jugadores.

**6.2.2 Creación del Product Backlog.** De la descripción del sistema y de los cuatro mini-juegos surgen los requisitos para elaborar el product backlog, que como se mencionó en el capítulo anterior, es una lista priorizada de objetivos/requisitos del software que según Scrum es propiedad del cliente y es creada por él. Esto resulta un tanto utópico ya que es dificultoso encontrar un

cliente que le pueda o quiera ofrecer el tiempo y dedicación que requiere una gestión ágil de un proyecto; de hecho este proyecto no tiene un cliente definido. Sin embargo, para bien de este proyecto, Scrum define la figura del Proxy, quien se empapa del dominio y de lo que quiere o quisiera el cliente final para elaborar la lista; la figura del proxy fue desempeñada por los desarrolladores y autores de este proyecto.

Teniendo en cuenta que un PB<sup>9</sup> puede sufrir modificaciones tanto de prioridad como alcance durante todo el proyecto, se construyó un primer PB que comprende la descripción de las funcionalidades esperadas a alto nivel y que en su momento nos permitió adquirir una primera visión del proyecto.

El product backlog está descrito en la tabla 1, y contiene los requisitos listados con su identificación, nombre, prioridad, esfuerzo requerido y estado. Además se puede identificar claramente a qué Sprint fue asignado para su realización. Las tareas que implica la lista de requisitos se realizaron distribuidas en 6 sprint o iteraciones, que resultaron en un incremento de funcionalidad que se iba incorporando al software. Para este proyecto se decidió que cada Sprint tuviera una duración de 20 días.

**Tabla 1. Product Backlog o Pila del Producto**

**PRODUCT BACKLOG**

ID	NOMBRE	PRIORIDAD	ESFUERZO (Días)	ESTADO
<b>Sprint 1</b>				
1	Página Principal	100	3	Hecho
2	Sistema de Registro y Login	90	6	Hecho
3	Menú Principal de la Aplicación	87	4	Hecho

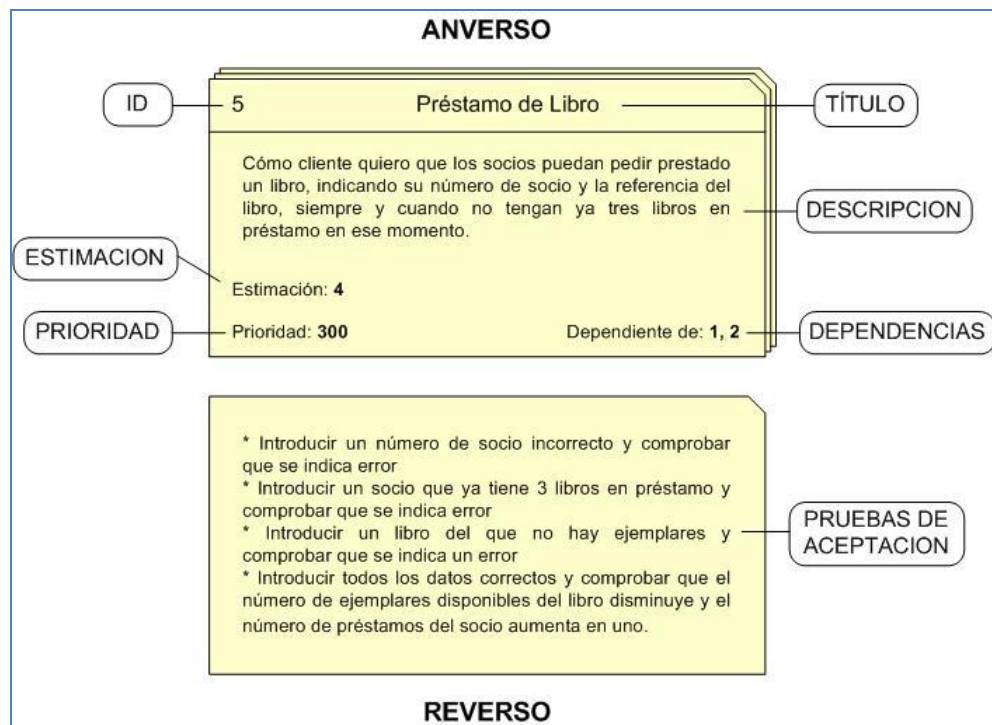
<sup>9</sup> Product Backlog o Pila del Producto.

4	Pantalla de Selección de categoría de juego	86	4	Hecho
5	Sistema de Tiempo	80	3	Hecho
<b>Sprint 2</b>				
6	Juego SIMON	79	10	Hecho
7	Juego SECUENCIA DE FRUTAS	78	10	Hecho
<b>Sprint 3</b>				
8	Juego ¿DÓNDE ESTÁN LAS PERLAS?	77	20	Hecho
<b>Sprint 4</b>				
9	Juego ALIMENTA LOS PECES	76	20	Hecho
<b>Sprint 5</b>				
10	Pantalla información fin de partida	70	6	Hecho
11	Pantalla de Logros y Trofeos	60	7	Hecho
12	Pantalla de Estadísticas	50	7	Hecho
<b>Sprint 6</b>				
13	Modo Vista Completa	40	4	Hecho
14	Pantallas de Tutorial	39	8	Hecho
15	Implementar Sistema de Mensajes	35	8	Hecho

Para ilustrar los requisitos de un software son comúnmente utilizados los diagramas de casos de uso, sin embargo en Scrum se suelen expresar en forma de historias de usuario, pues estas últimas además de que traen más agilidad al proyecto, pueden describir requerimientos funcionales y no funcionales, lo que resulta ventajoso respecto a los casos de uso, que no permiten determinar los requisitos no funcionales.

Cada historia es un requerimiento de negocios visto desde el punto de vista de un usuario. Se escriben con el siguiente formato: “como... quiero... para...”. Un ejemplo en nuestro sistema es: “Como usuario jugador quiero que los juegos de memoria y atención estén debidamente separados y sean accesibles desde un menú para acceder fácilmente a ellos”. La Figura 14 muestra la estructura de una historia de usuario con un ejemplo.

**Figura 14. Estructura de una Historia de Usuario.**



En este proyecto, cada requisito del product backlog es ilustrado por una historia de usuario con su estimación, prioridad, y cuyo título es el mismo requerimiento o nombre en el PB. Los criterios o pruebas de aceptación son pruebas que el código debe superar para dar como finalizada la implementación del requerimiento; estos criterios son comúnmente un consenso entre los usuarios y los desarrolladores, así que algunos de estos criterios son establecidos por los desarrolladores y autores de este proyecto, y otros por quienes serían tutores y jugadores en el sistema.

## 6.3 DESARROLLO POR SPRINTS: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN

### 6.3.1 Sprint Uno

#### Sprint Backlog

Tabla 2. Sprint Backlog, del Sprint Uno.

ID	NOMBRE	PRIORIDAD	ESFUERZO (Días)	ESTADO
<b>Sprint 1</b>				
1	Página Principal	100	3	Hecho
2	Sistema de Registro y Login	90	6	Hecho
3	Menú Principal de la Aplicación	87	4	Hecho
4	Pantalla de Selección categoría de juego	86	4	Hecho
5	Sistema de Tiempo	80	3	Hecho

#### Historias de Usuario: Análisis del Sprint

Tabla 3. Historias de Usuario, Sprint Uno.

TITULO	DESCRIPCIÓN	CRITERIOS DE ACEPTACIÓN
Página Principal	Como usuario quiero poder acceder a la aplicación desde cualquier PC a través de internet	La aplicación debe estar contenida en una página web.  Si el navegador no tiene instalados los plug-ins necesarios, debe darse la opción de instalarlos.  La página principal debe contener

		las opciones para crear una cuenta y para iniciar sesión.
Sistema de Registro y Login	Como usuario quiero poder registrarme para tener una cuenta y acceder al sistema con mi nombre de usuario y contraseña.	<p>El formulario de registro debe ser claro y breve para los dos tipos de usuario.</p> <p>El sistema debe validar el correo electrónico en el registro y comprobar que el nombre de usuario que se ingrese no exista.</p> <p>Cuando el usuario pulse “aceptar” en el formulario de registro, debe registrarse en el sistema y desplegarse la pantalla de iniciar sesión.</p> <p>Al iniciar sesión, el sistema debe revisar y validar que la contraseña sea correcta para ese usuario y que además el usuario ya haya activado la cuenta.</p>
Menú Principal de la Aplicación	Como usuario validado quiero poder ver las diferentes opciones de la aplicación en un menú principal	<p>Luego de validado el usuario, se debe mostrar la opción que lleva a los juegos y la opción para ver las estadísticas en el caso del tutor; o los logros y trofeos en el caso de que el usuario validado sea un jugador.</p> <p>Si el usuario da clic en “Jugar”, se debe desplegar la pantalla de</p>

		<p>selección de categoría de juego.</p> <p>Si el usuario tutor escoge la opción para ver las estadísticas, se debe desplegar la pantalla que contiene las mismas para todos los jugadores asociados a él.</p> <p>Si el usuario jugador escoge la opción de logros y trofeos, se debe desplegar la pantalla que contiene sus puntuaciones, logros y trofeos obtenidos en cada mini-juego.</p>
<p>Pantalla de Selección de categoría de juego</p>	<p>Como usuario quiero que los juegos de memoria y atención estén debidamente separados y sean accesibles desde un menú.</p>	<p>La pantalla debe contener dos iconos que diferencie de categoría a los juegos.</p> <p>Cuando se pulse el icono de memoria se debe desplegar la pantalla de selección de juegos de memoria.</p> <p>Cuando se pulse el icono de atención se debe desplegar la pantalla de selección de juegos de atención.</p>

Sistema de Tiempo	Como usuario quiero que para los juegos de “Secuencia de Frutas” y “¿Dónde están las Perlas?”, exista un cronometro que dicte el tiempo en que se debe completar cada ronda	<p>El cronometro debe ser visible y fácil de entender</p> <p>El sistema debe indicar cuando está próximo a acabarse el tiempo de la ronda.</p> <p>Al acabarse el tiempo, se debe terminar la ronda automáticamente.</p>
-------------------	---	---

## Diseño e implementación del Sprint

A continuación se describe el proceso por el cual se llegó a los productos que constituyen este proyecto, es decir cómo se realizó la implementación.

### 6.3.1.1 Página Principal

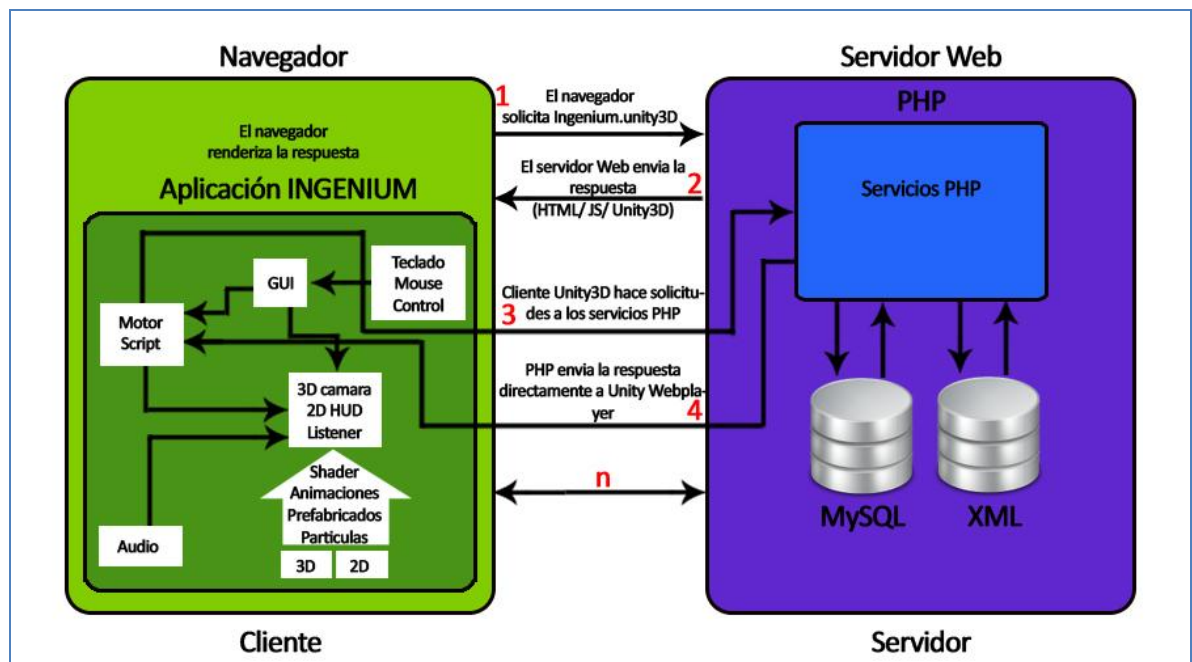
La pagina principal donde está contenida la aplicación, se encuentra publicada en el sitio <http://www.omninegocio.com/WebPlayer/WebPlayer.html>. Para que se permita ejecutar la aplicación, se requiere que el navegador web tenga instalado el complemento que exige Unity3d, herramienta con la cual se creó la aplicación completa.

La tecnología desarrollada por Unity technologies permite exportar cualquier aplicación a diferentes plataformas que incluyen, Play Station 3, Xbox 360, Nintendo Wii, dispositivos IOS, dispositivos Android y navegadores web para computadores de escritorio Mac y PC. Para este ultimo el navegador requiere tener instalado el plug-in “Unity web Player” el cual está disponible en la página oficial de Unity3d <http://unity3d.com>.

Al desarrollar una aplicación en unity3d para navegadores web, el motor se encarga de compilar la aplicación y genera un archivo de extensión .unity3d el cual se puede tratar de manera similar al archivo .swf que genera Macromedia flash.

La pagina principal de este proyecto está constituida por un archivo html en el que se encuentra embebida la aplicación, la cual automáticamente se carga en el equipo del usuario para ser visualizada en el navegador, siguiendo el esquema mostrado en la Figura 15, que además representa la arquitectura cliente servidor de la aplicación.

Figura 15. Arquitectura Cliente Servidor de INGENIUM



Para esta aplicación se seleccionó un tamaño de visualización de 960x600 pixeles, el cual es un tamaño común para aplicaciones que se ejecutan en navegadores web y permite una visualización adecuada en la mayoría de monitores y pantallas actuales.

La página principal de la aplicación detecta automáticamente mediante un código Javascript si el navegador tiene o no instalado el plug-in “Unity Web Player”, y si no lo está, indica que éste debe ser descargado de la página oficial de Unity3d mostrando el vínculo para ello. El código Javascript es generado por el compilador de Unity3d cuando se le da la opción de “Build and Run” en la ventana principal del Engine de Unity3d.

### **6.3.1.2 Sistema de Registro y Login**

El sistema de registro permite a los usuarios tener una cuenta propia y única. Permite registrarse, es decir, crear una cuenta nueva; naturalmente permite iniciar sesión con el nombre de usuario y contraseña que el usuario indica cuando crea la cuenta. Adicionalmente, si el usuario en algún momento olvida su nombre de usuario o contraseña, éste puede recuperarla fácilmente a través de su correo electrónico.

Internamente, la cuenta del usuario, es necesaria para el manejo de datos del usuario dentro de la aplicación y poder realizar las funciones de seguimiento a éste (almacenar y mostrar los puntajes, los trofeos, el progreso estadístico, entre otros).

La figura 16 muestra la ventana principal de registro y login. En ella el usuario puede iniciar sesión de inmediato, o registrarse si no tiene una cuenta aún. Si es un usuario ya registrado y no recuerda su usuario y/o contraseña, el botón respectivo le da la opción de recordarla.

Figura 16. Pantalla Principal Sistema de Registro y Login

SISTEMA DE REGISTRO Y LOGIN

Usuario:

Contraseña:

RECORDARME

INICIAR SESIÓN

REGISTRARSE

OLVIDÉ MI USUARIO/CONTRASEÑA

La figura 17 muestra el formulario de registro que el usuario debe llenar. En él, el usuario debe especificar si es un jugador o un tutor. Si es jugador, se pide el registro de la edad, el grado escolar y la persona que será su tutor, para esto, el niño debe escoger entre los tutores que ya están registrados en el sistema, es decir, que el sistema consulta la base de datos de tutores para desplegarlos ahí. Si es tutor, se pide el campo de acción de éste.

Los datos comunes para los dos usuarios es el nombre con el que iniciará sesión, la contraseña y el correo electrónico.

Es importante mencionar que los campos a llenar son validados, como se define en los criterios de aceptación.

Figura 17. Formulario de Registro

The image shows a registration form titled "FORMULARIO DE REGISTRO". The form contains the following fields and values:

- Usuario:** jugador1
- Contraseña:** [masked]
- Confirmar:** [masked]
- E-mail:** jugador1@ejemplo.com
- Confirmar:** jugador1@ejemplo.com
- Tipo Usuario:** JUGADOR
- Edad:** 9 Años
- Grado Escolar:** CUARTO
- Tutor:** JOSE

At the bottom of the form, there are three buttons: "ATRÁS", "ENVIAR", and "LIMPIAR".

Luego de registrado el usuario, el sistema envía instrucciones para activar la cuenta al correo electrónico del usuario. El usuario solo podrá usar la aplicación si activa su cuenta a través de su correo, esto para promover el acompañamiento de un adulto al momento del registro (para el caso de los niños). Una vez el usuario active su cuenta, puede iniciar sesión, ver figura 18.

Figura 18. Iniciar Sesión

The image shows a login and registration interface titled "SISTEMA DE REGISTRO Y LOGIN". It contains the following elements:

- Usuario:** A text input field containing "jugador1".
- Contraseña:** A password input field with masked characters "\*\*\*\*\*".
- RECORDARME:** A checkbox that is currently checked.
- Buttons:** Four buttons are located at the bottom: "INICIAR SESIÓN", "REGISTRARSE", "OLVIDÉ MI USUARIO/CONTRASEÑA", and a button with an "X" icon.

Todo el Sistema de Registro y Login utiliza la clase GUI de Unity3d desde código C#. En términos de Unity3d, el sistema de Registro y Login, se creó como una escena la cual se denomina RegistroLogin.unity.

Las llamadas a la base de datos se realizan mediante scripts PHP.

Cuando el usuario pulsa "iniciar sesión" se despliega el menú principal de la aplicación el cual está contenido en otra escena, es decir, este botón tiene programado saltar de la escena RegistroLogin.unity a la escena que contiene todos los menús de la aplicación, llamada menus.unity.

### 6.3.1.3 Menú Principal de la Aplicación

El menú principal de la aplicación contiene tres botones (ver figura 19). El de la parte superior lleva al menú de categoría de juegos después de la animación que mueve la cámara a través del escenario. El botón de la izquierda muestra los puntajes y trofeos, si el que está en sesión es un jugador; o muestra las estadísticas de los jugadores, si el que está en sesión es un tutor. El botón de la derecha lleva a cualquiera de los dos tipos de usuarios a la pantalla de correos, donde puede enviar mensajes al correo electrónico de sus compañeros jugadores y su tutor, en el caso del usuario jugador; o a sus jugadores a cargo en el caso del tutor.

Cada botón es un GUI Texture que tiene asignado un script diferente en lenguaje C# que controla estas acciones y son hijos de un objeto vacío el cual tiene un script que administra la animación de cada uno de estos. Son además elementos 2D que tienen una posición dentro de la escena.

El menú principal está contenido en la escena menus.unity junto con los demás menús.

Figura 19. Menú Principal



#### 6.3.1.4 Pantalla de Selección de Categoría de Juego

Esta pantalla contiene los botones que permiten seleccionar la categoría de juegos, el botón de la izquierda, lleva al submenú que contiene los juegos de memoria y el botón de la derecha, lleva al submenú que contiene los juegos de atención (ver figura 20). A partir de esta pantalla en la navegación de ventanas, el usuario siempre verá un botón que lo devuelve atrás; en esta pantalla, si el botón es presionado, llevará al menú principal.

Cada botón es un elemento 2D, GUI texture. Los botones de “memoria” y “atención” tienen asignado el mismo script que mediante un condicional controla la cámara y la mueve acercándose a dos partes del planeta dependiendo de cuál de los dos botones es presionado. De acuerdo con lo anterior, el botón de memoria lleva a los dos juegos que entrenan esta habilidad (ver figura 21); y el botón de atención lleva a los dos juegos de atención (ver figura 22).

Este menú está contenido dentro de la misma escena `menus.unity`, y se mantendrá en la misma escena si cualquiera de los dos botones de selección de categoría es presionado.

Figura 20. Pantalla de Categorías de Juego



Figura 21. Pantalla Juegos de Memoria



Figura 22. Pantalla Juegos de Atención



### 6.3.1.5 Sistema de Tiempo

Tres de los juegos que se seleccionaron para desarrollar tienen un sistema de tiempo en su versión original. Como se pensó que era conveniente no cambiar la base esencial de cada juego seleccionado, se realizó el sistema de tiempo que a través de un script controla el tiempo máximo para que el jugador realice cada ronda.

Para ello se creó un prefabricado que se denominó Reloj (ver figura 23) y está constituido por una imagen y un script que controla el material del prefabricado manipulando la propiedad alpha de la textura. Este prefabricado fue posteriormente utilizado en los juegos de “Secuencia de Frutas”, “¿Dónde están las Perlas?” y “Alimenta los Peces”.

Figura 23. Prefabricado Reloj



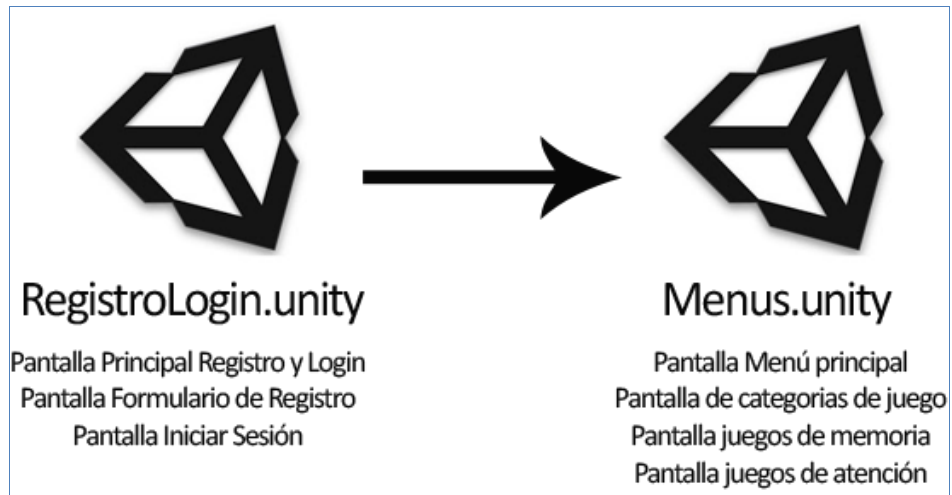
#### **6.3.1.6 Resultado final del Sprint Uno**

Al finalizar esta primer etapa de implementación, se realizó una revisión de la misma, y se estableció como un primer producto final, los productos operativos que resultaron de cada historia de usuario.

A este punto ya se tenía elaborado todo el sistema para registrar usuarios, y las pantallas de los menús Principal, de categoría de juegos y de los juegos de memoria y atención.

La figura 24, representa la navegación entre escenas, y enuncia las pantallas contenidas en cada una.

Figura 24. Navegación de escenas hasta el Sprint 1



### 6.3.2 SPRINT DOS

#### Sprint Backlog

Tabla 4. Sprint Backlog, Sprint Dos.

ID	NOMBRE	PRIORIDAD	ESFUERZO (Días)	ESTADO
<b>Sprint 2</b>				
6	Juego SIMON	79	10	Hecho
7	Juego SECUENCIA DE FRUTAS	78	10	Hecho

#### Historias de Usuario: Análisis del Sprint

Tabla 5. Historias de Usuario, Sprint Dos.

TITULO	DESCRIPCIÓN	CRITERIOS DE ACEPTACIÓN
Juego SIMON	Como usuario, quiero que al seleccionar el juego Simon, se cargue dicho juego en el que	El juego debe generar una secuencia aleatoria de botones que comienza con un único botón

	<p>deba memorizar una secuencia que el juego genera.</p>	<p>para que el jugador la repita.</p> <p>Cada uno de los cuatro botones del juego Simon, debe emitir un sonido propio al reproducirse la secuencia automática y al ser presionado.</p> <p>A medida que el jugador acierte las rondas, la secuencia debe aumentar en un nuevo botón aleatorio.</p> <p>Si se consigue acertar la secuencia, el juego debe continuar a una siguiente ronda, de lo contrario, el juego debe terminar.</p> <p>Debe mostrarse al jugador, que acertó o falló.</p>
<p>Juego SECUENCIA DE FRUTAS</p>	<p>Como usuario quiero que el juego exponga un ejercicio que involucre memorizar y luego recordar una secuencia de imágenes.</p>	<p>La secuencia de imágenes debe mostrarse en la pantalla y permanecer unos segundos con el fin de que el usuario la memorice.</p> <p>Los botones que el jugador presiona para repetir la secuencia mostrada, deben ser intuitivos y representar apropiadamente las imágenes de la secuencia</p> <p>La ronda debe terminar si es presionado un botón que no siga</p>

		<p>secuencia correcta y se debe mostrar al jugador que falló</p> <p>Si el jugador acierta el orden de la secuencia, la ronda debe terminar, indicarse que la ganó y además se debe acumular los puntos obtenidos.</p> <p>Las imágenes deben ser de fácil identificación.</p>
--	--	--

## Diseño e implementación del Sprint

### 6.3.2.1 Juego SIMON

Las funcionalidades del juego ya han sido descritas en el capítulo 5 de este libro. Aquí se menciona cómo se elaboró el juego.

Con el fin de facilitar la implementación del juego, se diseñó una arquitectura para el juego basada en el modelo cliente-servidor (ver figura 25). Esta arquitectura es similar en todos los juegos.

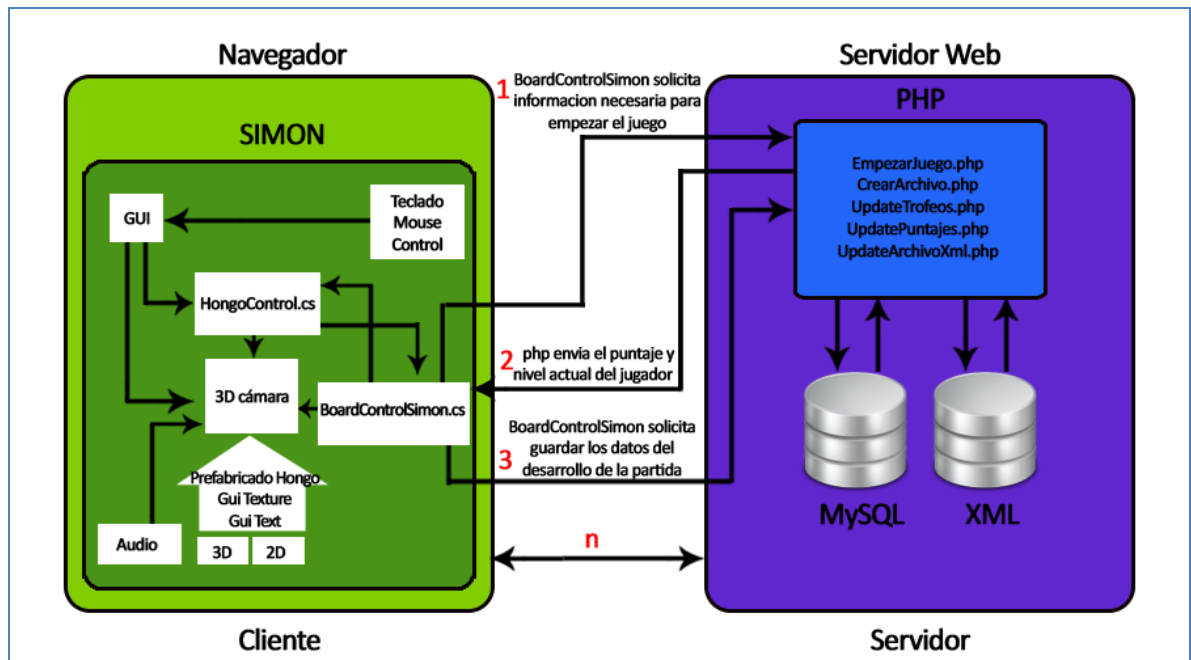
El juego y todos sus elementos están contenidos en una escena que se llamó *Simon.unity* y está constituido principalmente por un *objeto de juego* y un *prefabricado*. La inteligencia del juego está distribuida en dos clases. El *objeto de juego* responde a una de estas clases, a la que se llamó *BoardControlSimon.cs*, y el *prefabricado* responde a la otra clase denominada *HongoControl.cs* y contiene un modelo 3d de un hongo que responde a clics del usuario y es instanciado cuatro veces en la escena.

La clase *BoardControlSimon.cs* es la encargada del control de las rondas, de los niveles, de crear y reproducir la secuencia aleatoria, de aumentar la secuencia en

un botón cuando el jugador acierta, y de finalizar la partida en cuanto el jugador se equivoca. El manejo de datos como los puntajes, los trofeos y las estadísticas del jugador, es controlado por diferentes métodos de esta clase que se encargan de solicitar datos de la base de datos para iniciar el juego, y de enviar datos para actualizar la misma cuando el jugador finaliza una partida.

La clase *HongoControl.cs*, a través del método *OnMouseDown*, llama el método que controla la animación del hongo y la reproducción del sonido propio, además se comunica con la clase *BoardControlSimon.cs* cuando el hongo es presionado para que esta analice si la secuencia repetida por el jugador era la correcta.

Figura 25. Arquitectura Cliente-Servidor del juego SIMON



Para el diseño visual se aplicó la investigación recopilada acerca de software para niños. Las interfaces se hicieron siguiendo los criterios expuestos en el capítulo 3.4, que resultaron en interfaces amigables, con botones y modelos 3d llamativos y de colores contrastantes, acompañados de textos grandes y legibles. Para el objeto con el que el usuario interactúa durante el desarrollo del juego, se usó un modelo 3d de un hongo, adoptando así para este juego una ambientación de un

jardín en el que se encuentran los 4 hongos. Todos los modelos 3d de este juego fueron diseñados por los autores de este proyecto. Ver pantalla de juego SIMON figura 10.

### 6.3.2.2 Juego Secuencia de Frutas

En este apartado se describe cómo se diseñó y elaboró este juego. En la figura 26 se muestra la arquitectura que se diseñó para luego implementar el juego.

En este juego también se definió una clase control denominada *BoardControlFrutas.cs* que fue asignada a un *objeto de juego*. Para administrar los cuatro niveles en este juego, se utilizó un *objeto de juego* por cada nivel, que a su vez contiene otros en cantidades que dependen del nivel que representa (tres para el nivel uno, cuatro para el nivel dos, cinco para el nivel tres y seis para el nivel cuatro), estos objetos son utilizados para que *BoardControlFrutas.cs* sepa cuantas frutas debe instanciar y en que posiciones, un ejemplo de esto, es que en el primer nivel se muestran tres frutas. Cada prefabricado contiene un modelo 3d de una fruta diferente y el script *ControlFruta.cs* que se encarga de administrar la interacción del usuario con el objeto 3D.

Todos estos elementos de juego más los modelos 3D usados como escenografía están contenidos en una escena llamada *Frutas.unity*.

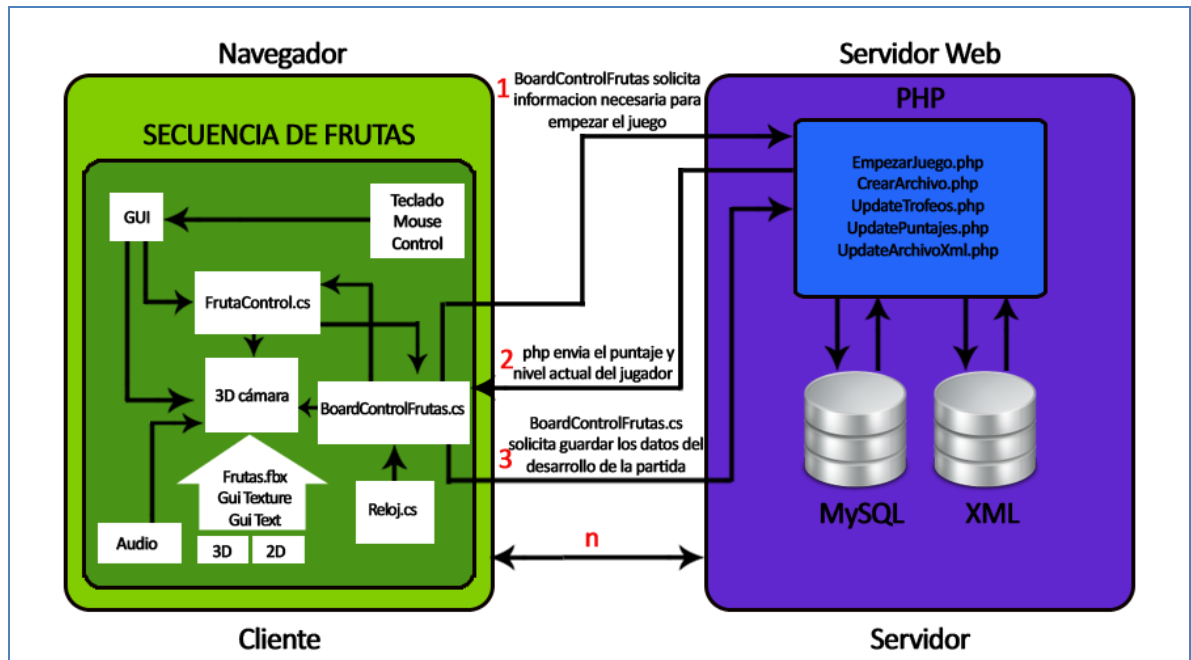
La clase *BoardControlFrutas.cs* es la encargada de revisar el número de rondas jugadas, aumentar o disminuir los niveles dependiendo del desempeño del jugador, instanciar el número de frutas por nivel en un orden aleatorio, comunicar a la clase *FrutaControl.cs* que realice las animaciones de las frutas según sea necesario (aparecerlas y desaparecerlas), mostrar en pantalla los botones que representan las frutas y que el jugador presiona para repetir la secuencia, y por último es la encargada de verificar si el jugador resolvió la ronda bien o mal desplegando un mensaje según sea el caso. El manejo de datos como los puntajes, los trofeos y las estadísticas del jugador realizados en cada partida, es

controlado por diferentes métodos de esta clase que se encargan de solicitar datos de la base de datos para iniciar el juego, y de enviar datos para actualizar la base de datos, y los archivos XML necesarios para realizar las estadísticas.

La clase *FrutaControl.cs*, llama al método que reproduce la animación de aparecer y desaparecer las futas y controla el sonido que se reproduce para cada una de estas, además se comunica con la clase *BoardControlFrutas.cs* para que esta reproduzca los botones que el usuario presiona para repetir la secuencia y analice si fueron presionados en el mismo orden de la secuencia generada por el juego.

Este juego utiliza el prefabricado Reloj para limitar el tiempo que tiene el jugador para resolver cada ronda; el script del prefabricado se comunica constantemente con la clase *BoardControlFrutas.cs* enviando el valor del tiempo en el que el jugador resolvió la ronda, si el jugador no pudo resolver la ronda en el tiempo dado que en el caso de este juego es de 8 segundos, el script de Reloj enviará un mensaje *BoardControlFrutas.cs* para que finalice la ronda. Ver pantalla de Juego Secuencia de Frutas figura 11.

Figura 26. Arquitectura del juego SECUENCIA DE FRUTAS



El diseño visual del juego adopta los criterios de diseño visual para niños. Los botones que el jugador debe presionar son intuitivos con la imagen entendible. Las frutas que el jugador ve en la pantalla son modelos 3d, pero se ven como imágenes 2D porque la cámara los ve desde una vista ortográfica, pues la vista que da el modo en perspectiva podría causar confusión ya que las frutas tienen diferentes posiciones dentro de la pantalla.

Para la realización de los modelos 3D y de todos los elementos del escenario, se eligió un tema de bosque que consta de la vista de arboles, flores, piedras y por supuesto las frutas.

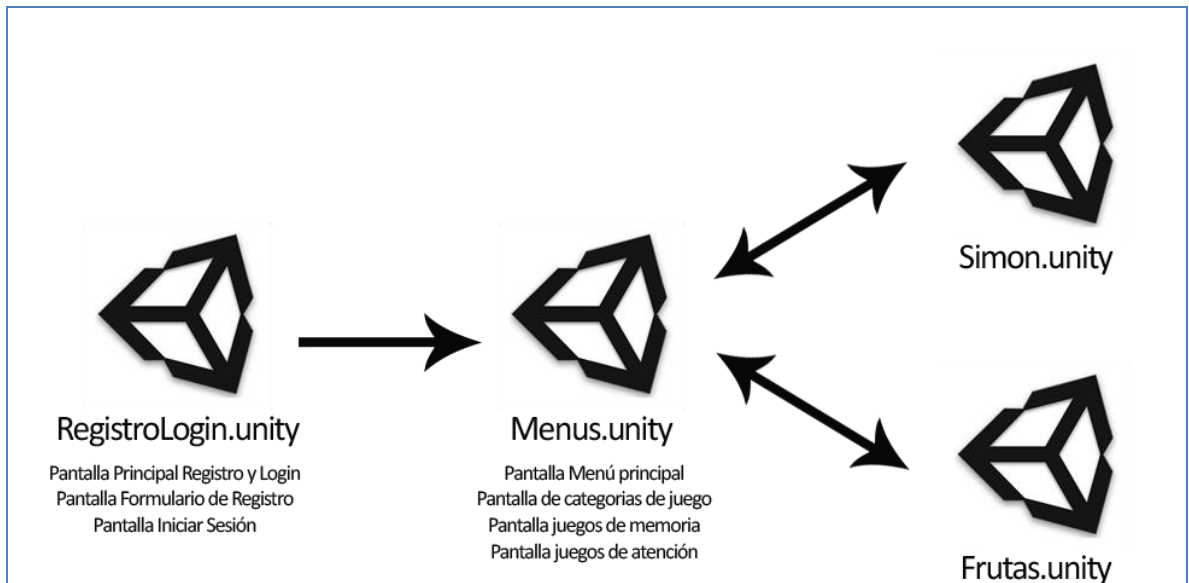
### 6.3.2.3 Resultado final del Sprint Dos

Al finalizar esta segunda etapa de implementación, se realizó una revisión de la misma, y se estableció como producto final de este sprint, los productos operativos que resultaron de cada historia de usuario, es decir, el juego Simon y el juego Secuencia de Frutas.

El resultado de este sprint se incorpora al software anclando estas dos escenas a la escena de Menu.unity realizada en el sprint anterior.

La figura 27, representa la navegación entre escenas.

Figura 27. Navegación entre escenas hasta el Sprint 2



### 6.3.3 SPRINT TRES

#### Sprint Backlog

Tabla 6. Sprint Backlog, Sprint Tres.

ID	NOMBRE	PRIORIDAD	ESFUERZO (Días)	ESTADO
<b>Sprint 3</b>				
8	Juego ¿DÓNDE ESTÁN LAS PERLAS?	77	20	Hecho

## Historias de Usuario: Análisis del Sprint

Tabla 7. Historias de Usuario, Sprint Tres.

TITULO	DESCRIPCIÓN	CRITERIOS DE ACEPTACIÓN
Juego ¿DÓNDE ESTÁN LAS PERLAS?	Como usuario, quiero que al seleccionar este juego, se muestre unas imágenes a las cuales deba prestar atención para luego de que se oculten y cambien de posición, saber su nueva ubicación.	<p>El juego debe ubicar en la pantalla una cantidad de almejas según el nivel, donde algunas de ellas lleven una perla en su interior. La perla se ubica en almejas aleatoriamente</p> <p>Visualmente debe haber clara distinción entre las almejas que contienen perla y las que no.</p> <p>Cuando el jugador acierte la ronda, se le debe mostrar tal cosa.</p> <p>Cuando el jugador se equivoque en la ubicación de una perla, debe mostrarse cuál era la correcta.</p> <p>A medida que el jugador acierte las rondas, el nivel de dificultad debe aumentar.</p>

### Diseño e implementación del Sprint Tres

#### 6.3.3.1 Juego ¿Dónde están las Perlas?

¿Dónde están las perlas? Está basado en el ¿donde están los huevos? del doctor Kawashima en su videojuego “Train your brain with Dr. Kawashima”. Las funcionalidades del juego ya han sido descritas en el capítulo 5 de este libro. Aquí se menciona cómo se elaboró el juego.

En la figura 28 se puede apreciar la arquitectura cliente/servidor diseñada para este juego.

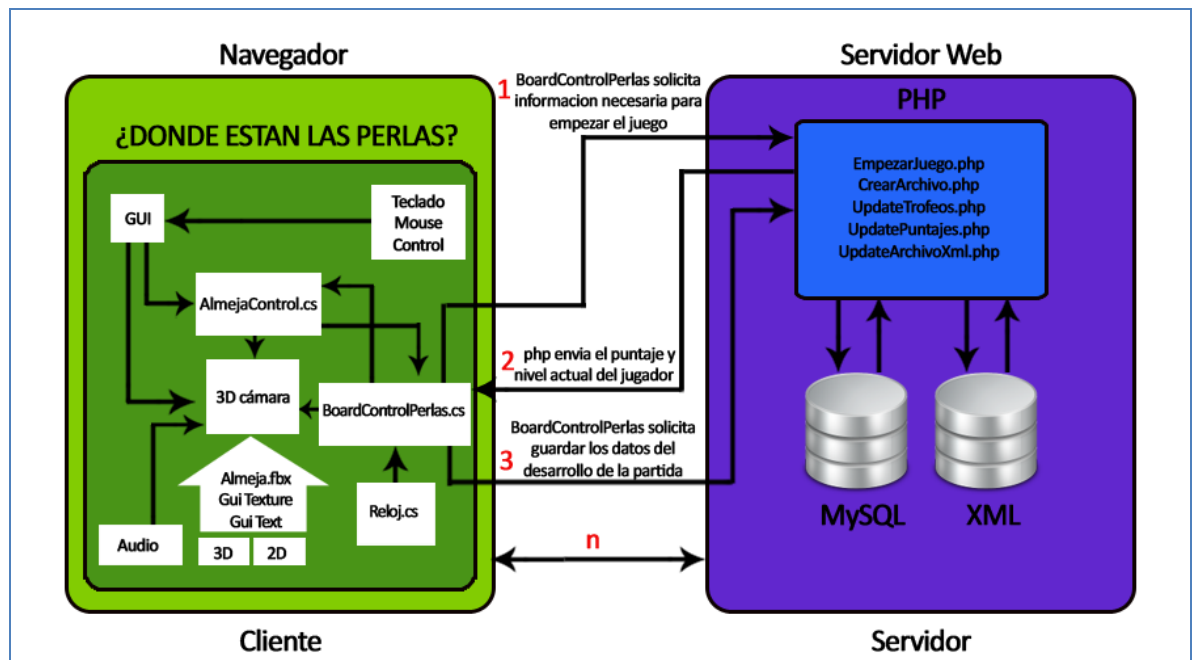
El juego básicamente está constituido por un objeto de juego el cual tiene atado un script que se encarga de la administración del juego y se llama *BoardControlPerla.cs*; para hacer fácil la identificación del nivel, cuantas almejas deben instanciarse por nivel y en que posiciones deben aparecer, se utilizaron otros cuatro objetos de juego que representan los cuatro niveles disponibles, cada uno de estos objetos contiene a su vez un número de objetos de juego, dependiendo del nivel que representan, de esta manera *BoardControlPerla.cs* accede a cada “nivel” instanciando y ubicando cada almeja en su posición según el número y posiciones de objetos de juego que encuentre en cada nivel (tres objetos de juego para el primer nivel, cuatro para el segundo, cinco para el tercer y seis para el cuarto nivel). Por último está el prefabricado “Almeja” el cual contiene el modelo 3D de la almeja, con su respectiva perla, la animación de abrir la almeja y cerrarla, y el script *ControlAlmeja.cs* encargado de administrar principal la interacción con el objeto 3D; todos estos elementos de juego más los modelos 3D usados como escenografía están contenidos en una escena llamada *Perlas.unity*.

La clase *BoardControlPerlas.cs* es la encargada de revisar el número de rondas jugadas, aumentar o disminuir los niveles dependiendo del desempeño del jugador, instanciar el número necesario de Almejas por nivel, comunicar a la clase *AlmejaControl.cs* que realice las animaciones de la almeja según sea necesario y de verificar si el jugador resolvió la ronda bien o mal desplegando un mensaje según sea el caso. El manejo de datos como los puntajes, los trofeos y las estadísticas del jugador realizados en cada partida, es controlado por diferentes métodos de esta clase que se encargan de solicitar datos de la base de datos para iniciar el juego, y de enviar datos para actualizar la misma así como también a los archivos XML necesarios para realizar las estadísticas.

En este juego se usa el prefabricado, Reloj el cual tiene como función limitar el tiempo que tiene el jugador para resolver cada ronda, el script del prefabricado se comunica constantemente con la clase *BoardControlPerlas.cs* enviando el valor del tiempo en el que el jugador resolvió la ronda, si el jugador no pudo resolver la ronda en el tiempo dado que en el caso de este juego es de 8 segundos, el script de Reloj enviara un mensaje *BoardControlPerlas.cs* para que declaro que el jugador perdió la ronda.

La clase *AlmejaControl.cs*, a través del método *OnMouseDown*, llama al método que controla las animaciones de abrir y cerrar la almeja así como el sonido que se reproduce para cada una de estas, además se comunica con la clase *BoardControlPerlas.cs* cuando la almeja es presionada para que esta analice si la almeja contenía una perla o no.

Figura 28. Arquitectura Juego ¿DÓNDE ESTÁN LAS PERLAS?



El diseño de los botones y las interfaces de este juego es consistente con el de los juegos anteriores por lo tanto conserva los criterios descritos en el apartado diseño visual para niños que se encuentra en el marco de referencia.

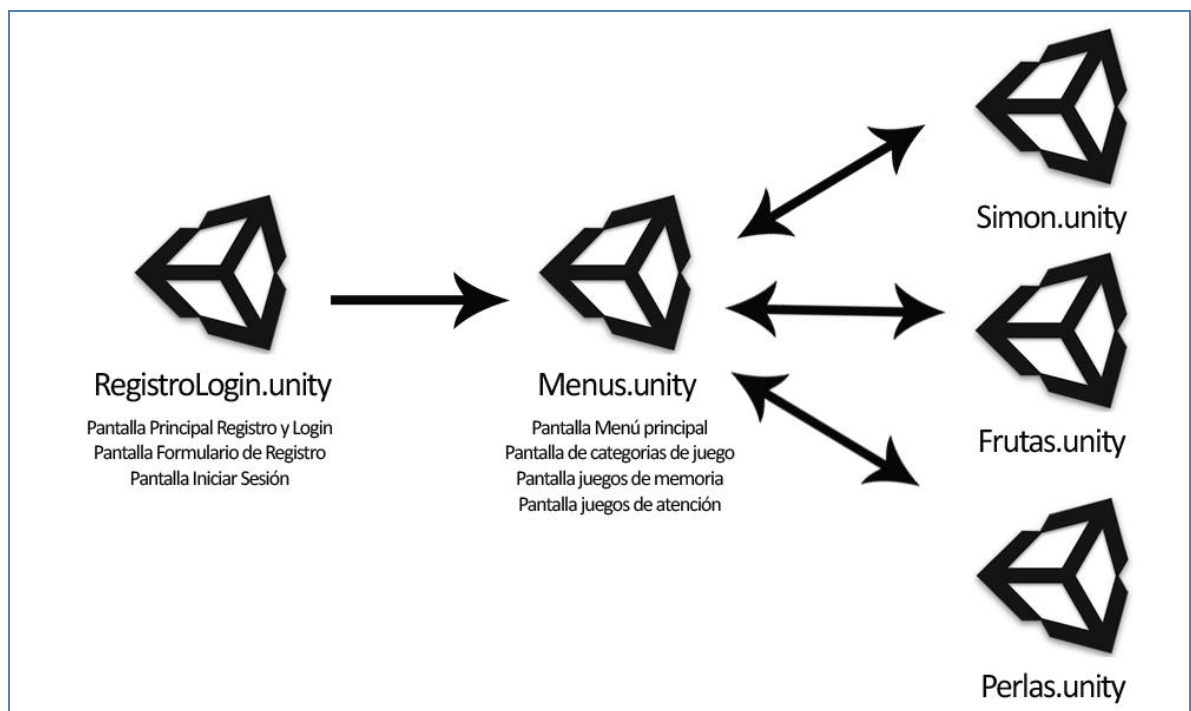
Para la realización de los modelos 3D, tanto como para las almejas como para todos los elementos del escenario, se eligió un tema marino que consta de la vista de un mar, unas piedras, unas cuantas islas y unas gaviotas que lo sobre vuelan, todo esto se realizo bajo el criterio descrito en diseño visual para niños el cual menciona que a los niños les gusta las metáforas geográficas.

### 6.3.3.2 Resultado final del Sprint Tres

Al finalizar esta tercera etapa de implementación, se realizó una revisión de la misma, verificando que todos los criterios de aceptación descritos en las historias de usuario se cumplieran y dando como resultado la escena que contiene el juego ¿Dónde están las perlas? plenamente operativo.

La navegación entre escenas incluyendo el producto del sprint 3 se puede apreciar en la figura 29.

Figura 29. Navegación entre escenas hasta el Sprint 3



### 6.3.4 SPRINT CUATRO

#### Sprint Backlog

Tabla 8. Sprint Backlog, Sprint Cuatro

ID	NOMBRE	PRIORIDAD	ESFUERZO (Días)	ESTADO
<b>Sprint 4</b>				
9	Juego ALIMENTA LOS PECES	76	20	Hecho

#### Historias de Usuario: Análisis del Sprint

Tabla 9. Historias de usuario, Sprint Cuatro

TITULO	DESCRIPCIÓN	CRITERIOS DE ACEPTACIÓN
Juego ALIMENTA LOS PECES	Como usuario, quiero que al iniciar el juego, haya una cantidad de peces nadando, cuya finalidad de juego sea alimentarlos solo una vez, de modo que deba estar atento a los peces para no repetir la dosis a uno o varios.	Quiero que la aplicación provea tanta cantidad de comida como peces haya en cada ronda.  Quiero que el juego no permita alimentar un pez inmediatamente después de haber alimentado otro, debe indicarse un tiempo de espera para alimentar a un siguiente pez  Si la ronda se logra perfecta, el juego debe aumentar en uno el número de peces para la siguiente ronda.  La ronda debe finalizar cuando se haya gastado toda la comida en

		<p>alimentar los peces.</p> <p>Cada pez debe recorrer el escenario siguiendo trayectorias aleatorias.</p> <p>Cada pez se ilumina cuando el mouse se pose sobre él. Si se da clic sobre él, se debe identificar si ya había sido alimentado anteriormente o si es la primera vez.</p> <p>Cuando se termina una ronda, se debe distinguir cuales peces fueron alimentados una vez, cuales más de una vez y cuales faltaron por alimentar.</p>
--	--	---

## Diseño e implementación del Sprint Cuatro

### 6.3.4.1 Juego Alimenta los Peces

En la figura 30 se puede apreciar la arquitectura cliente/servidor diseñada para este juego.

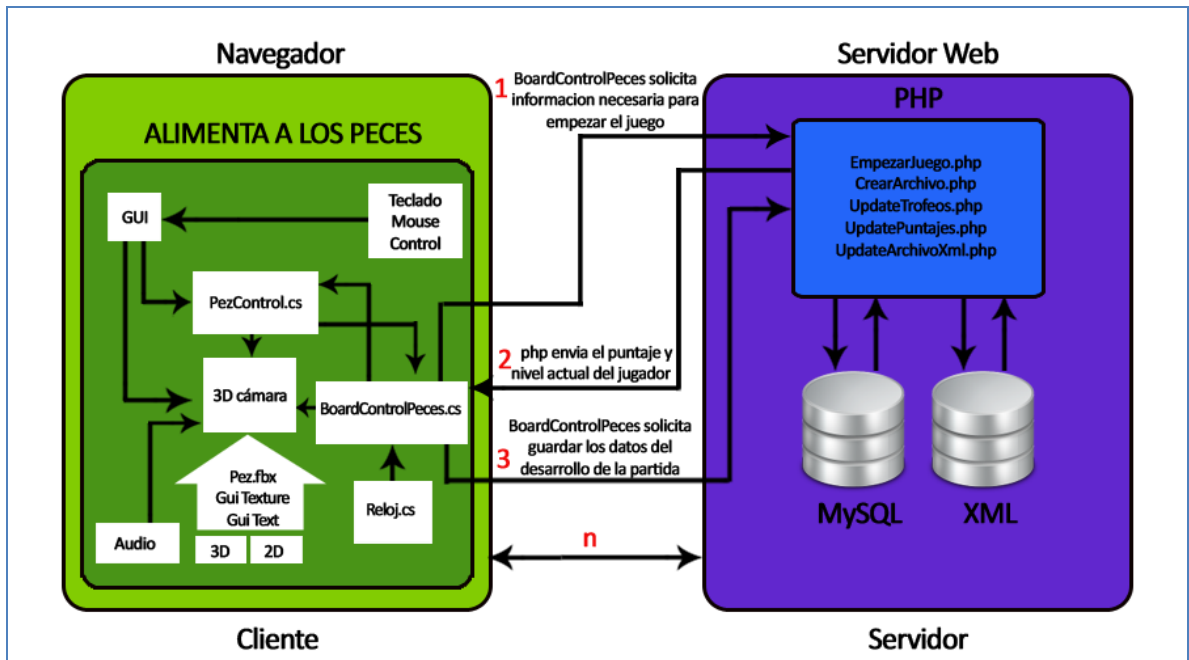
Alimenta a los peces está constituido como en los juegos anteriores, por un objeto de juego, que tiene atado un script encardo de la lógica principal del juego y de la comunicación de datos, llamado *BoardControlPeces.cs*, además de un prefabricado llamado PEZ el cual contiene el modelo 3D del pez, la animación del pez nadando y el script encargado de la controlar interacción entre el jugador y el pez, este juego está contenido en una escena llamada *Peces.unity*.

La clase *BoardControlPeces.cs* es la encargada de revisar el número de rondas jugadas, aumentar o disminuir los niveles dependiendo del desempeño del jugador, instanciar los peces que sean necesarios según el nivel en el que se encuentra el jugador y verificar si todos los peces fueron alimentados solo una vez, esta clase se comunica con la clase *PezControl.cs* al final de cada ronda para mostrar al jugador cuales peces fueron alimentado una vez, cuales fueron alimentados dos o más veces y cuáles no fueron alimentados, y muestra en pantalla el rendimiento al jugador, además al iniciar el juego se encarga de solicitar los datos de la base de datos necesarios para empezar y cuando se finaliza la partida del juego envía los resultados para que se actualice las base de datos y los archivos XML.

En este juego se utiliza el prefabricado Reloj para que el jugador tenga que esperar un tiempo cada vez que alimentó a un pez para poder proseguir a alimentar al siguiente, este tiempo es de 3 seg, una vez transcurrido este la clase *Reloj.cs* se comunica con la clase *BoardControlPeces.cs* para habilitar lo opción que permite presionar el siguiente pez.

La clase *PezControl.cs* es la encargada de verificar cuando un pez fue presionado, si un pez fue presionado por primera vez, dibuja un contorno verde alrededor del pez y emite un sonido en señal de que la acción fue correcta, si ya ha sido presionado aunque sea una vez, dibuja un contorno rojo alrededor del pez y emite un sonido de error, esta clase también dibuja un contorno amarillo alrededor del pez cuando se desata el evento *Onmouseover()*, una de la labores más importantes es generar el camino aleatorio de los peces una vez el pez fue instanciado, cada vez que se alimenta un pez la clase *PezControl.cs* envía un mensaje a la clase *BoardControlPeces.cs* informándole el estado del pez para que esta verifique si la ronda quedo bien o mal.

Figura 30. Arquitectura del Juego ALIMENTA LOS PECES



El diseño de los botones y las interfaces de este juego es consistente con el de los juegos anteriores y los criterios expuestos en marco de referencia sobre el diseño visual para niños.

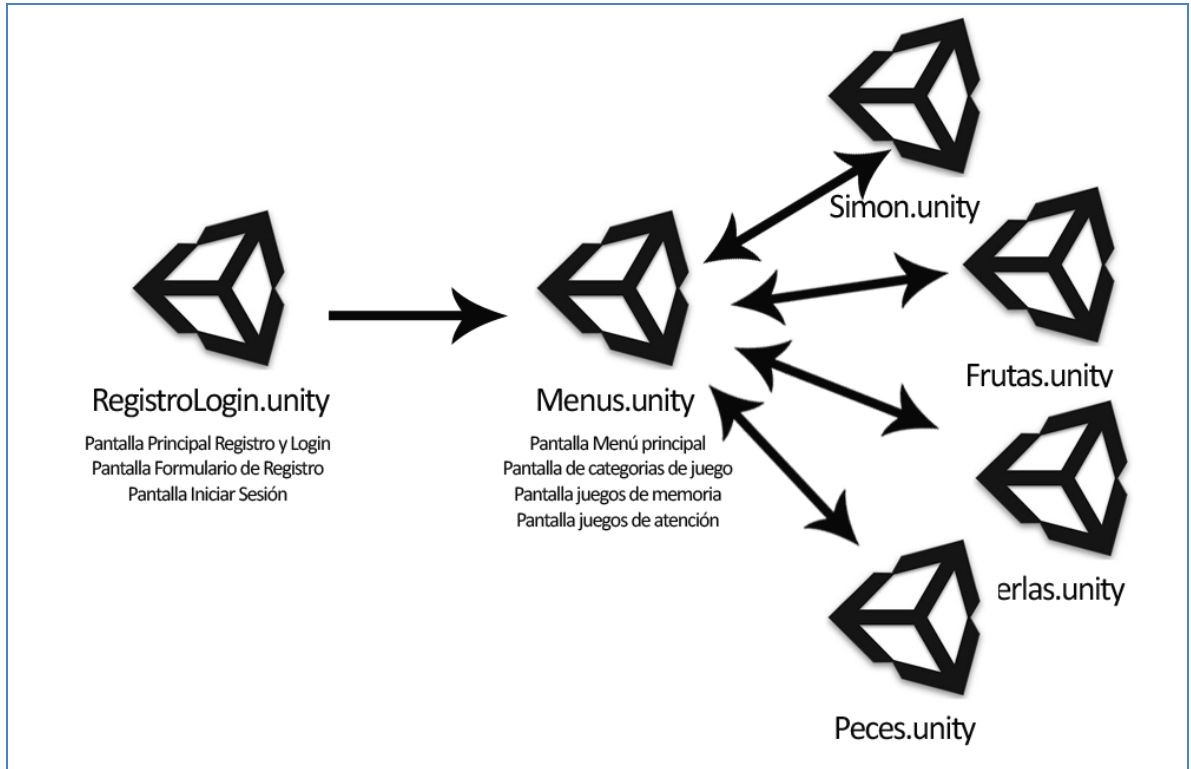
Para la realización de los modelos 3D, tanto los peces, como los fondos y demás elementos del escenario, se utilizó como tema, un ambiente submarino haciendo de nuevo uso del criterio de que a los niños les agradan las metáforas geográficas.

### 6.3.4.2 Resultado final del Sprint Cuatro

Al finalizar la cuarta etapa de implementación, se verificó que todos los criterios de aceptación descritos en las historias de usuario se cumplían y se concretó la escena que contiene el juego Alimenta a los peces plenamente operativo.

La navegación entre escenas incluyendo el producto del sprint 4 se puede apreciar en la figura 31.

Figura 31. Navegación entre escenas hasta el Sprint 4



### 6.3.5 SPRINT CINCO

#### Sprint Backlog

Tabla 10. Sprint Backlog, Sprint Cinco

ID	NOMBRE	PRIORIDAD	ESFUERZO (Días)	ESTADO
<b>Sprint 5</b>				
10	Pantalla información fin de partida	70	6	Hecho
11	Pantalla de Logros y Trofeos	60	7	Hecho
12	Pantalla de Estadísticas	50	7	Hecho

## Historias de Usuario: Análisis del Sprint

Tabla 11. Historias de Usuario, Sprint Cinco

TITULO	DESCRIPCIÓN	CRITERIOS DE ACEPTACIÓN
Pantalla de información fin de partida	Como usuario quiero que al terminar cada partida de juego pueda visualizar un resumen del desempeño en ésta.	<p>Esta pantalla debe mostrarse siempre que el jugador termina una partida de juego</p> <p>Debe contener los puntos obtenidos por las rondas perfectas, por tiempo y los puntos restados por rondas fallidas.</p> <p>Debe mostrarse al jugador si obtuvo un nuevo trofeo por una buena puntuación</p> <p>Al cerrar esta pantalla puede ubicarse de nuevo en el juego para volver a jugarlo o ir al menú anterior y volver al menú principal.</p>
Pantalla de Logros y Trofeos	como usuario quiero que el sistema almacene y me permita visualizar el puntaje máximo y los trofeos obtenidos en cada juego	<p>La pantalla debe contener información actualizada en todo momento: el puntaje máximo obtenido por el jugador y los trofeos que este lleve en cada juego.</p> <p>El contenido grafico de esta pantalla debe ser agradable, atrayente y estimulante pues es</p>

		donde el jugador ve plasmado su desempeño en los juegos
Pantalla de Estadísticas	Como usuario tutor quiero poder ver el progreso de los jugadores relacionados a mí, mediante diferentes gráficos estadísticos.	El sistema permite supervisar el rendimiento individual de todos los jugadores relacionados al tutor, en cada uno de los juegos disponibles.  El sistema permite comparar el rendimiento de cada jugador con los demás jugadores de un mismo tutor.

## Diseño e implementación del Sprint Cinco

### 6.3.5.1 Pantalla información fin de partida

La pantalla fin de partida es la encargada de mostrarle al jugador su desempeño durante una partida, por lo tanto se despliega cuando el partida se termina, en esta el usuario puede encontrar la información correspondiente a los puntos obtenidos en la partida jugada los cuales se dividen en:

- Puntos por tiempo: entre menos se demore el jugador en resolver cada ronda más alto será este puntaje, aunque se muestra el tiempo total de todas las rondas solo se da puntos por las rondas completada correctamente, este puntaje solo aplica para los juegos, ¿Dónde están las perlas? y Secuencia de frutas.
- Puntos por ronda bien: estos puntos se da para cada ronda que ha sido realizada satisfactoriamente, el cálculo de los puntos depende de cada uno

de los juegos.

- Puntos por ronda mal: representa el puntaje otorgado cuando la ronda no es completa correctamente y está expresado en números negativos, entre mayor sea el desempeño por ronda menor será la penalización.
- Puntaje Total: es la suma de los puntajes anteriores.
- Trofeo alcanzado: Los trofeos son incentivos diseñados para el jugador, existen tres trofeos, trofeo de bronce, trofeo de plata y trofeo de oro, para obtener cada uno de estos trofeos se debe obtener cierto puntaje que depende de cada juego.

La pantalla de fin de partida se construyo como un prefabricado sin ningún modelo 3d que tiene atado un script llamado *FinPartida.cs* el cual usa las propiedades de unityGUI para desplegar la información que recibe de los *BoardControl* de cada juego los cuales realizan el cálculo del puntaje durante el desarrollo de cada juego, de esta manera el prefabricado pudo ser instanciado en cada escena del juego y es activado al final de cada partida.

La figura 32 muestra la pantalla de fin partida de una partida del juego de Secuencia de Frutas.

Figura 32. Pantalla fin de partida



	Rondas	Puntos
	00:16	+1009
	6	+1500
	3	-1050
<b>TOTAL</b>		<b>1459</b>

Has Conseguido el Trofeo:

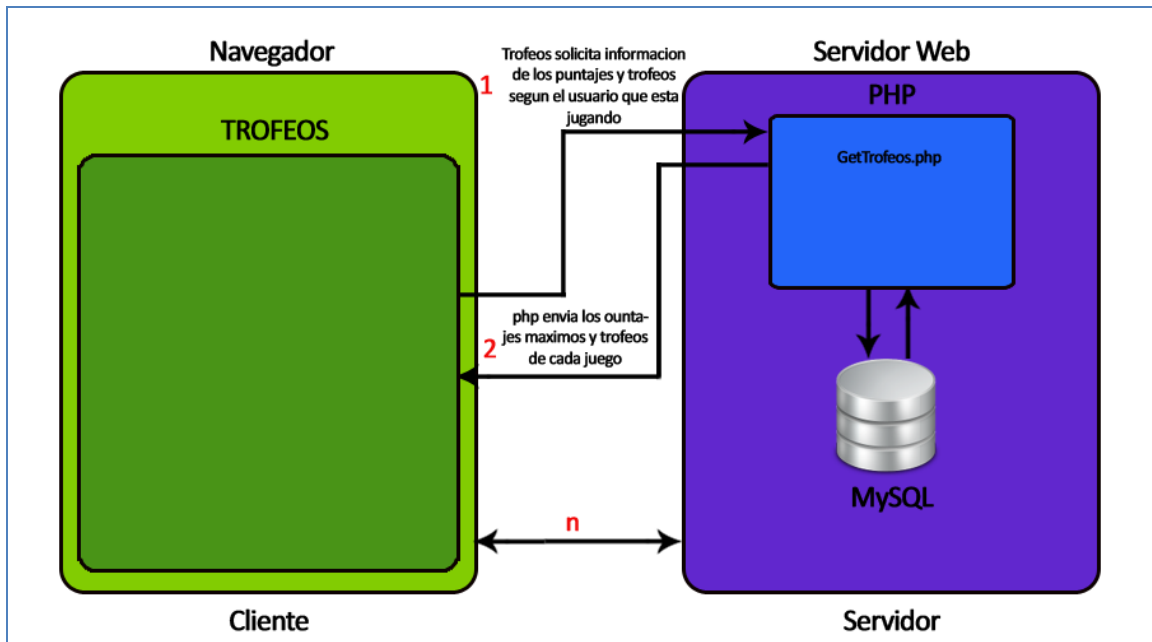
### 6.3.5.2 Pantalla de Logros y Trofeos

La pantalla de Logros y Trofeos se encuentra en una escena llamada Trofeos.unity. Para acceder a esta escena se debe dar clic al botón de resultados el cual se encuentra el menú principal de la aplicación, solo los usuarios denominados como jugadores, es decir los niños, pueden acceder ella. En esta los niños podrán encontrar una descripción de su desempeño en cada uno de los juegos, el cual se ve reflejado simplemente el puntaje máximo y todos los trofeos alcanzados durante las partidas. Esta escena fue diseñada principalmente como incentivo para animar al jugador a jugar, y superar su desempeño tratando de obtener todos los trofeos de los juegos.

Las funciones de la pantalla están realizadas por un script llamado *Trofeos.cs* que solicita a la base de datos los puntajes y trofeos de cada juego al iniciar la escena, el script escribe en la cabecera de la pantalla el nombre del jugador y en una casilla

designada para juego los puntajes máximos del mismo, por ultimo *Trofeos.cs* dibuja en la pantalla los trofeos obtenidos en cada partida según la información que reciba de la base datos siguiendo el esquema cliente servidor de la Figura 33.

Figura 33. Arquitectura Cliente/servidor Escena Trofeos



La pantalla de Trofeos, gráficamente está constituida por cinco paneles en el primero se puede ver el nombre del jugador, y en los otros cuatro la información de cada juego, incluyendo el puntaje máximo y los trofeos, para identificar cada juego se uso un icono que lo representa a cada uno, todos los textos en esta pantalla son grandes y legibles, los símbolos e iconos son fácilmente relacionables y siguen la tendencia de diseño utilizada en las demás escenas (figura 34).

Figura 34. Pantalla Trofeos



### 6.3.5.3 Pantalla de Estadísticas

Esta pantalla se encuentra contenida en la escena Estadisticas.unity, ha esta escena solo pueden acceder los usuarios denominados como tutores desde el menú principal del juego en el botón de resultado.

Para empezar la descripción de esta pantalla se debe aclarar en primera instancia, que unity3D no contiene por defecto una forma de tabular datos así es que los autores de este proyecto tuvieron que realizar toda la lógica necesaria para poder realizar las graficas necesarias.

En esta escena lo primero que se encontrará serán los cuatro iconos de los juegos en la parte superior de la pantalla, una vez el tutor haga clic sobre alguno de los iconos el manejador principal llamado *Estadisticas.cs* solicitara la información de los nombres de los jugadores asociados al tutor que está conectado, los niveles

máximos y los puntajes máximos de cada uno de los jugadores para el juego cuyo icono fue presionado en primer lugar, esta información se obtiene de la base de datos y se verá listada en la pantalla en orden descendente para que el tutor pueda apreciar que jugador tiene el puntaje más alto en cada aplicación, enfrente de la información el tutor podrá ver dos botones, el primero esta nombrado como “individual” y el segundo como “comparativo” dependiendo de qué botón presione el tutor el manejador principal desplegara información estadística del jugador todo esto siguiendo un esquema de cliente servidor como se muestra en la figura 35.

Cuando el tutor presiona el botón “individual” el manejador solicita el registro histórico del desempeño del jugador el cual se encuentra contenido en un archivo XML, cada jugador tiene asociados cuatro archivos XML , uno por cada juego en el que se encuentra contenido el desempeño del jugador por cada día, guardando solo los puntajes máximos de cada día jugado, el script *Estadísticas.cs* recibirá la información referente solo a los últimos 20 días de juego y los graficará en una tabla de puntos versus tiempo.

El script *Estadísticas.cs* realizará una grafica de líneas y puntos utilizando la información recibida del archivo XML (ver figura 36), para graficar se utilizo un prefabricado con un componente LineRender, el cual permite dibujar en unity una línea manipulando una serie de características propias del como, número de vértices, ancho de la línea, posición de cada vértice en el espacio 3d y el color, este prefabricado es instanciado el número de veces que se necesite, según el número de datos que se haya recibido, y será manipulado para que sus coordenadas X e Y de cada vértice coincidan proporcionalmente con el valor de los puntos y las fechas obtenidos, de esta manera se determino el puntaje máximo que puede ser alcanzado en cada juego, por ejemplo, 10000 en secuencia de frutas, y a este se le asigno una posición de 6 en Y del espacio 3d de unity, todos los demás puntos por tanto, se encontraran en posiciones entre 0 y 6, para las coordenadas en X que hacen referencia al tiempo se utilizo el mismo método.

Entonces cuando el tutor presiona el botón *Estadísticas.cs*, hace la solicitud de los datos al archivo XML y los grafica usando las instancias del prefabricado con *linerRender*, y aunque las líneas en realidad también tienen una componente espacial Z la cámara tiene activada su propiedad ortográfica que solo le permite percibir dos coordenadas, en este caso, X y Y, dando la impresión de que realmente es una grafica bidimensional.

Finalmente cuando el tutor hace clic en el botón “comparativo”, el script *Estadísticas.cs* realiza una comparación del jugador para el que se presionó el botón, basado en la mediana, de esta manera se realizan dos graficas, la primera hace referencia a los puntajes máximos de todos los jugadores, el script calcula la mediana de todos los puntajes máximos y la dibuja en la pantalla, después ubica al jugador en la misma grafica, de esta manera el tutor podrá ver el desempeño del jugador comparado con el de los otros, apreciando que tan parecido esta su puntaje al de la mediana, el segundo grafico sigue el mismo proceso descrito pero esta vez es para el nivel máximo alcanzado (ver figura 37).

Figura 35. Arquitectura Cliente/Servidor Escena Estadísticas

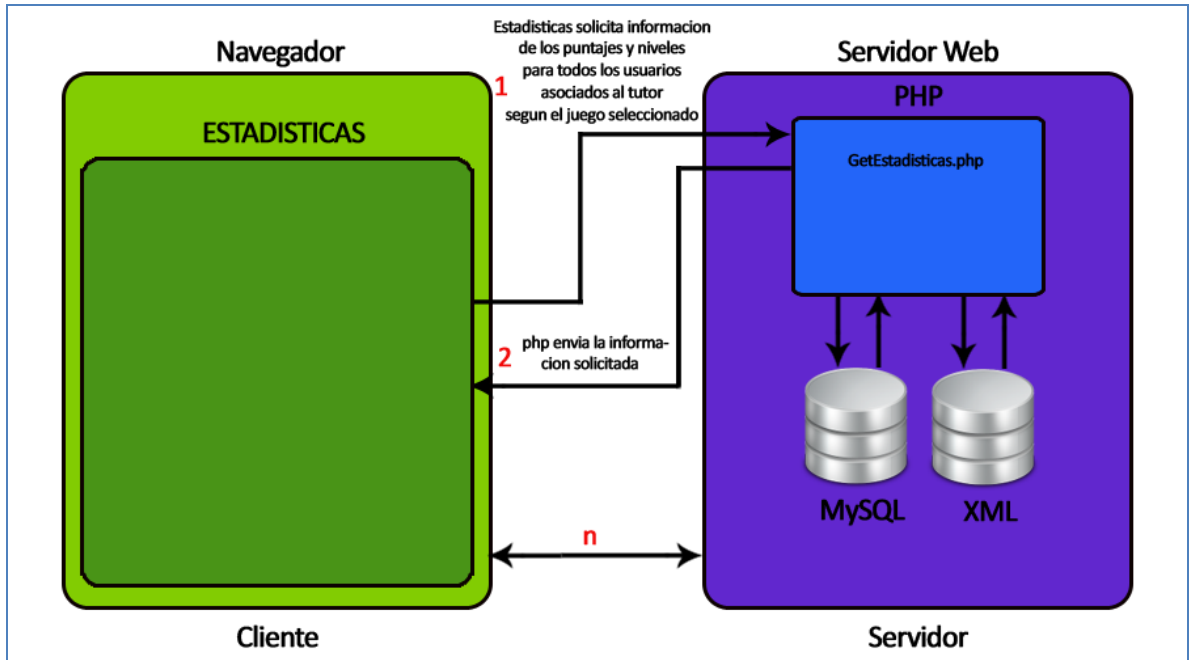


Figura 36. Pantalla de Estadísticas: Resultados individual

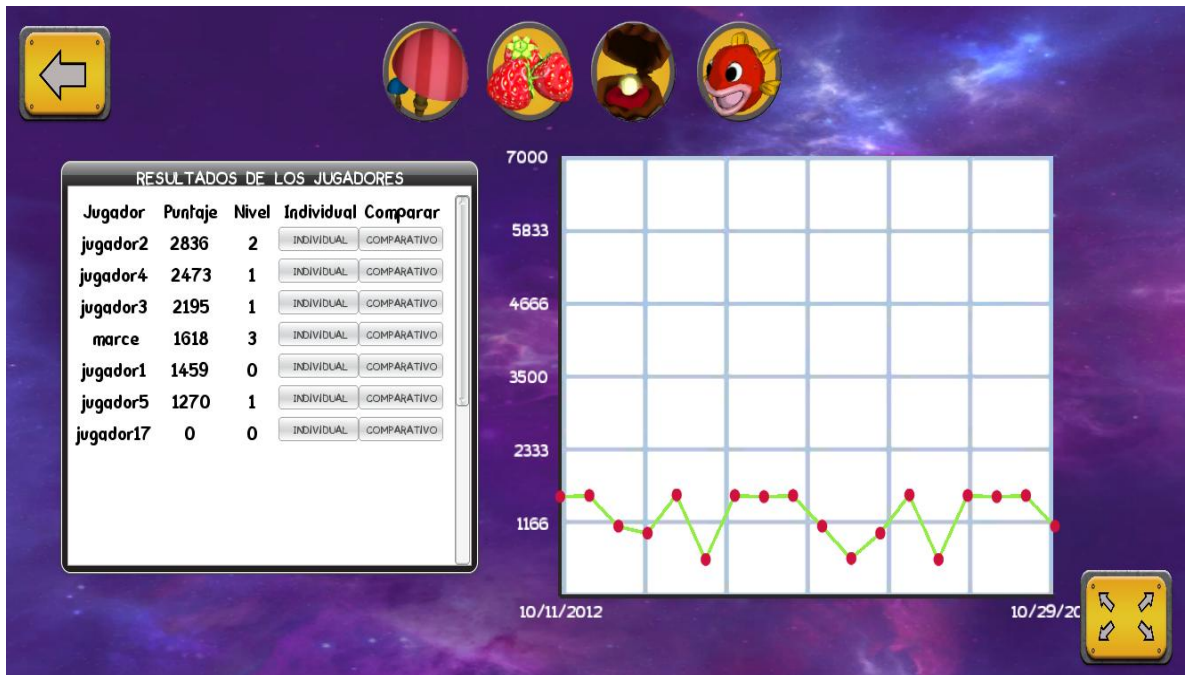
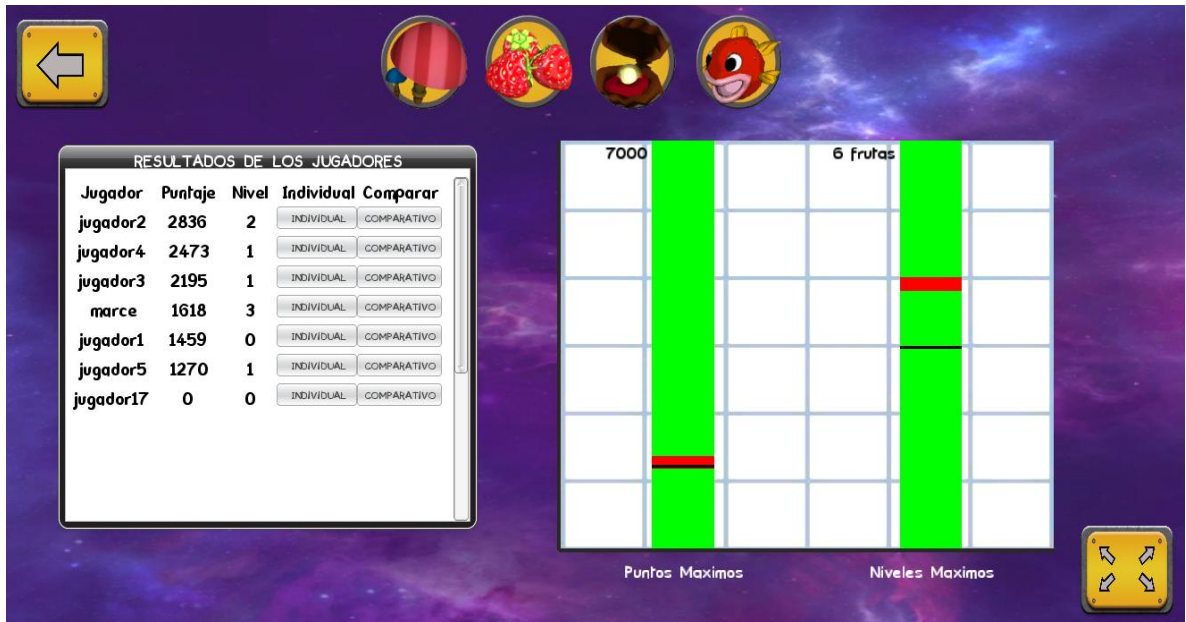


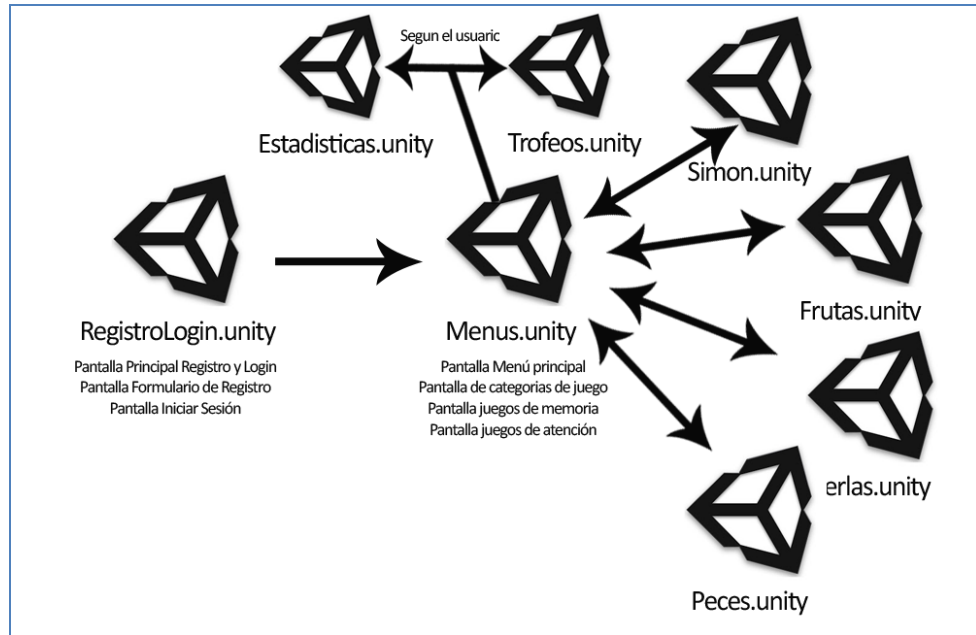
Figura 37. Pantalla de Estadística: Resultado Comparativo



#### 6.3.5.4 Resultado final del Sprint Cinco

Una vez finalizado el sprint cinco se obtuvo dos nuevas escenas que cumplían con los criterios de aceptación descritos en las historias de usuario, estas escenas fueron agregadas al proyecto general y enlazadas con las demás escenas según es requerido, de esta manera obtenemos el mapa entre escenas descrito en la figura 38.

Figura 38. Navegación entre escenas hasta el Sprint 5



### 6.3.6 SPRINT SEIS

#### Sprint Backlog

Tabla 12. Sprint Backlog, Sprint Seis

ID	NOMBRE	PRIORIDAD	ESFUERZO (Días)	ESTADO
<b>Sprint 6</b>				
13	Modo Vista Completa	40	4	Hecho
14	Pantallas de Tutorial	39	8	Hecho
15	Sistema de Mensajes	35	8	Hecho

#### Historias de Usuario: Análisis del Sprint

Tabla 13. Historias de Usuario, Sprint Seis.

TITULO	DESCRIPCIÓN	CRITERIOS DE ACEPTACIÓN
Modo Vista	Como usuario quiero que el	Debe mostrarse en todo momento

Completa	juego pueda ocupar toda la pantalla independientemente de la resolución del computador.	<p>en la pantalla, el botón que tenga la funcionalidad de entrar en pantalla completa.</p> <p>Al presionar el botón la aplicación debe ocupar completamente la pantalla del computador.</p> <p>Al estar en pantalla completa, y presionar de nuevo el botón, o presionar la tecla "escape" del teclado, la aplicación debe salir de modo pantalla completa.</p>
Pantalla de Tutorial	Como usuario quiero poder desplegar una pantalla que contenga información que sirva como guía para desarrollar los juegos.	<p>Cada juego debe tener el botón que muestre su tutorial en pantalla.</p> <p>Los tutoriales deben estar basados en imágenes y texto y explicar de manera muy sencilla cómo jugar.</p> <p>El tutorial debe poderse consultar durante el desarrollo del juego en cualquier momento.</p> <p>El tutorial debe poder cerrarse en cualquier punto de su desarrollo.</p> <p>Debe poderse adelantar y retroceder según se requiera.</p>

Sistema de Mensajes	Como usuario quiero poder comunicarme con otros usuarios del sistema a través de mensajes que pueda dejar en sus correos electrónicos.	<p>El sistema debe listar todos los usuarios a los que el usuario en sesión pueda enviar correos</p> <p>El sistema debe permitir que de manera muy fácil se pueda seleccionar a quién se quiere enviar un mensaje</p> <p>El sistema debe enviar el mensaje al correo electrónico de quien el usuario seleccionó.</p> <p>Debe notificarse al usuario si el mensaje fue enviado satisfactoriamente o no.</p>
---------------------	--	--

## Diseño e implementación del Sprint Seis

### 6.3.6.1 Modo Vista Completa

El modo de vista completa fue diseñado para que el usuario pueda disfrutar de la aplicación de una manera más cómoda, la idea de esta funcionalidad es simplemente que al presionar un botón la aplicación ocupe la totalidad de la pantalla del computador, independientemente de la maquina que el usuario este usando.

Según la historias de usuario, el botón que hace posible esta funcionalidad debe estar presente todo momento en la aplicación, debe hacer que al ser presionado, la aplicación ocupe toda la pantalla y si el juego ya está en pantalla completa al presionarlo de nuevo

La aplicación debe volver a su tamaño inicial, para que la funcionalidad se encontrara en todas las escenas del juego se realizo un prefabricado el cual contiene un script llamado *btnFullScreen.cs* este usa las propiedades de unityGUI, para dibujar un botón que al ser presionado altera la resolución de aplicación modificándola de 960x600 px, la cual es la resolución por defecto de la aplicación, a la resolución total de la pantalla que esté usando el usuario, cuando se está en modo pantalla completa si se vuelve a presionar el botón el script vuelve a alterar la resolución de la aplicación a su estado original, de esta manera el prefabricado simplemente fue instanciado en cada una de las escenas que constituyen el juego, y siempre podrá ser encontrado en la parte inferior izquierda de la pantalla.

#### **6.3.6.2 Pantalla de Tutorial**

Esta pantalla fue desarrolla con el fin de que el usuario que esté usando cualquiera de los juegos, pueda recibir un instructivo en caso de que no entienda como se debe jugar. El tutorial se puede desplegar en cualquier momento, incluso si el juego ya ha sido iniciado, y muestra una serie de imágenes que ilustran la forma correcto de realizar los juegos, ayudadas por breves textos que se encuentran debajo de cada una, todos los juego tiene su propia pantalla de tutorial y se pueden desplegar dando clic al botón que tiene el símbolo de interrogación y se encuentra ubicado sobre el botón de pantalla completa.

La pantalla de tutorial es también un prefabricado que esta instanciado en cada juego, el script *Tutorial.cs* que pertenece a este tiene como variable publica una texture 2D, es decir una imagen que puede ser cambiada según se necesite, para cada juego se creó una serie de imágenes que muestran el desarrollo correcto de cada juego de principio a fin y se ubicaron en el prefabricado.

El script despliega una pantalla usando UnityGUI, en el que se muestra las imágenes del tutorial y tres botones una para avanzar en el tutorial, otro para retroceder y uno para cerrar los tutoriales, en el momento en que el tutorial es

desplegado, el juego se detiene usando las propiedades Time de Unity y una vez se cierra se puede seguir jugando normalmente

Para ver las pantallas de los tutoriales de cada juego, ir al anexo B.

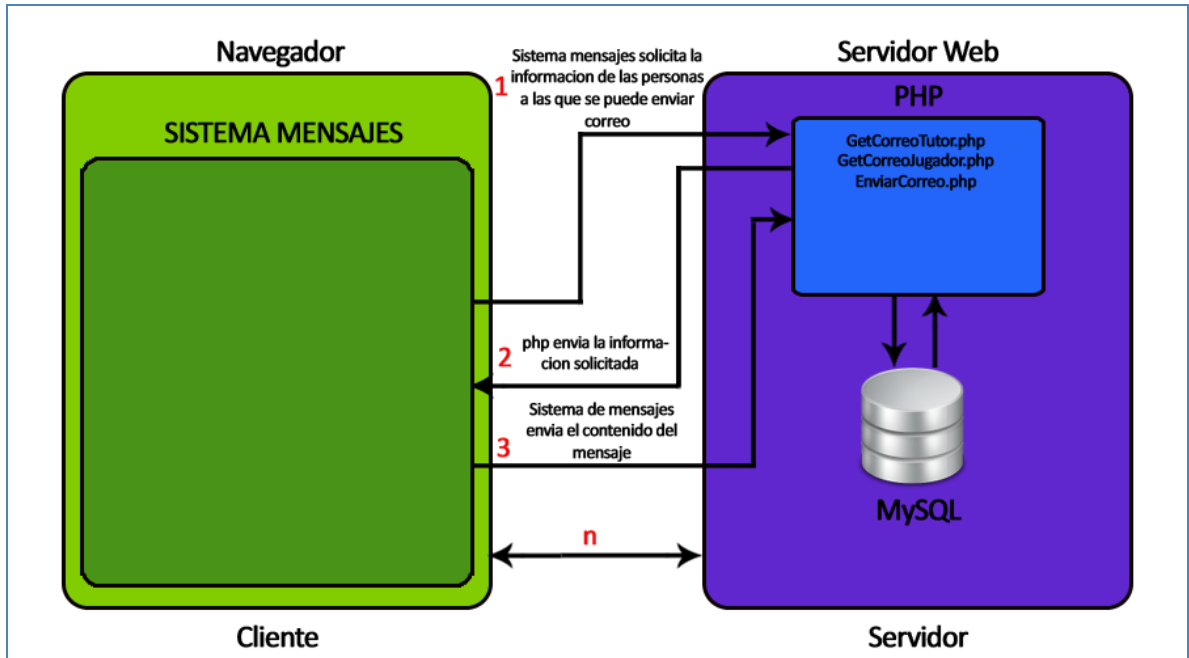
### **6.3.6.3 Sistema de mensajes**

Este sistema permite al usuario enviar mensajes a otros usuarios cuyo tutor es común, en el caso de los usuarios jugadores, en el caso de los tutores el sistema permite enviarle mensaje a los jugadores que están asociados a él, cuando se inicia la escena se despliega una lista de los usuarios a los cuales es posible enviar correos junto al nombre de cada uno de los usuarios se encuentra un botón que al ser presionado despliega una pantalla que contiene un campo para colocar el asunto del mensaje y otro para escribir el mensaje, los mensajes son enviados a la cuenta de correo que el usuario ingreso en su registro.

El sistema de mensajes consta de una clase llamada *Mensajes.cs* encargada de toda la lógica, esta cuando se inicia la escena solicita a la base de datos la información de los usuarios a los que se puede enviar correos y la despliega en la pantalla, después cuando el usuario presione el botón para enviar correo a cualquiera de los otros usuarios muestra la pantalla donde se puede ingresar el asunto del correo y el mensaje, cuando se envía el mensaje, este script envía la información del mensaje a un archivo PHP alojado en el servidor el cual tiene la función de enviar el mensaje al correo particular del usuario, siguiendo el modelo cliente/servidor de la figura 39.

Todo el sistema de mensajes está contenido en una escena llamada Correo.unity a la cual se puede acceder a través del botón llamado Correo, que se encuentra en el menú principal de la aplicación.

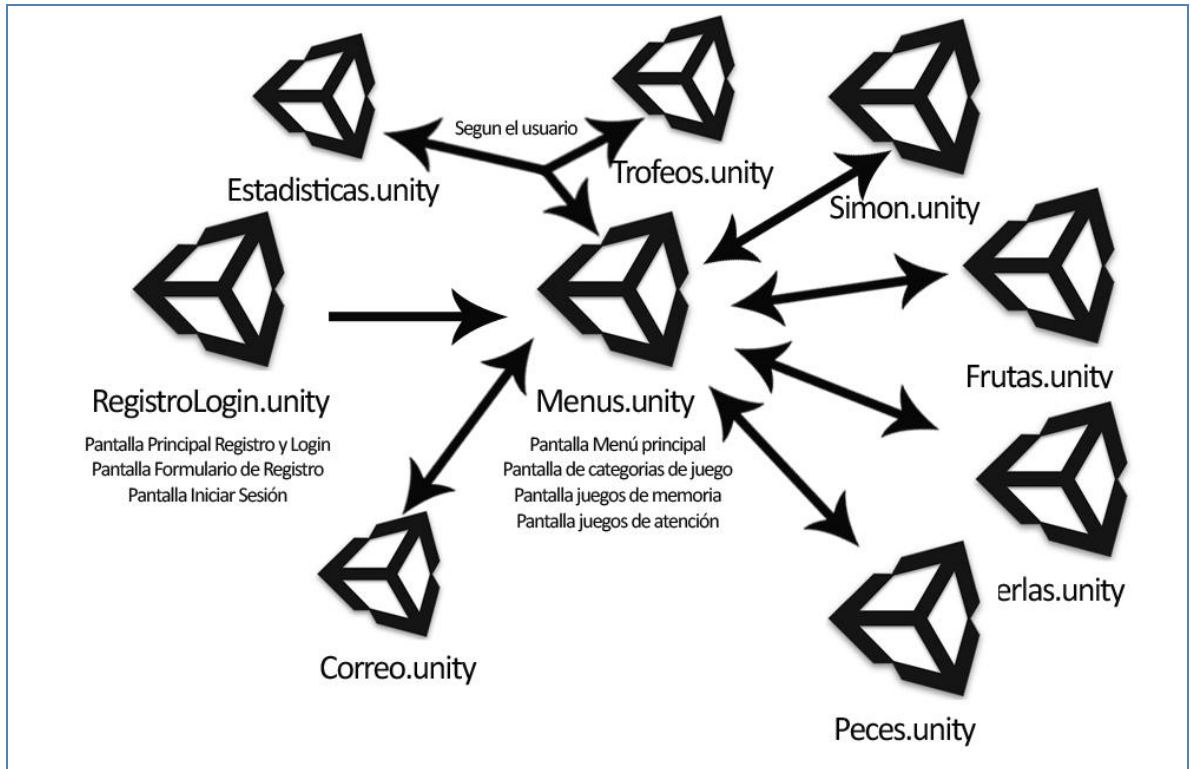
Figura 39. Arquitectura cliente-servidor de la escena de mensajes



#### 6.3.6.4 Resultado final del Sprint Seis

Con la finalización de esta última etapa se obtuvo la última escena contenida en este proyecto y los últimos detalles de las demás escenas concluyendo de esta manera todo lo planteado en el product Backlog, y cumpliendo todos los criterios de aceptación planteados en las historias de usuario, el producto de este último sprint es acoplado al proyecto, la navegación final entre escenas se puede apreciar en la figura 40.

Figura 40. Navegación entre escenas: Aplicación completa.



Con la finalización de esta etapa se da por concluido el desarrollo de la aplicación la cual conserva una línea de diseño visual a lo largo de todas las escenas, el cual es apropiado y optimo para niños de grados de básica primaria permitiéndole interactuar con ambientes que contienen textos grandes y legibles, botones grandes con graficas y mensajes fáciles de entender, los cuales son fáciles de ubicar, colores saturados, fuertes, bien definidos y contrastantes que hacen que sean fáciles de ubicar en todo momento, animaciones y modelos 3d fáciles de entender y asociar, metáforas geográficas en todos sus escenarios que atraen la atención del usuario promoviendo la creatividad e interés del jugador.

Juegos fáciles de entender y realizar acompañados de un sistema de puntos y trofeos el cual mantiene a los jugadores interesados en la realización de las actividades propuestas promoviendo que el jugador quiera entrenar sus habilidades cognitivas por sí mismo, y que lo vea simplemente como una actividad lúdica, aun cuando sepa que está relacionado con el ámbito académico.

El tutor se va encontrar con lo mencionado anteriormente, más un sistemas de resultados que le va a permitir monitorear a los jugadores a su cargo de una manera fácil sin la necesidad de tener un conocimiento profundo de estadística teniendo la capacidad de verificar el desempeño de cada uno en el tiempo además de poder comparar el desempeño de cada jugador en comparación con el del resto del grupo.

Todo esto acompañado con un sistema de mensaje que aunque sencillo, permite a los usuarios poder comunicarse sin ningún problema, para poder compartir sus puntajes, sus experiencias en el juego, animarse a superar sus puntajes o cualquier otro cosa que los jugadores consideren pertinente.

## 7. PRUEBAS DE FUNCIONALIDAD Y USABILIDAD

La metodología acogida para la realización de este proyecto permitió ir evaluando cada producto funcional resultado de cada sprint cuando este finalizaba. Luego, terminados todos los sprints, se tuvo un producto final que cumple con todos los requisitos funcionales y no funcionales establecidos en el product backlog.

Sin embargo, se hizo necesario y se estableció como objetivo de este proyecto, hacer pruebas del software. Para las pruebas se tuvieron en cuenta dos aspectos: lo funcional del software y la facilidad de usarlo, entenderlo y aprenderlo, es decir la usabilidad.

### PRUEBA DE LA APLICACIÓN.

Cuando se dio por finalizada la aplicación, se ejecutó la prueba del software en el Colegio Fundación UIS de Floridablanca, Santander, donde se contó con la participación de 15 niños de básica primaria: cinco niños de tercero, cinco de cuarto y cinco de quinto en quienes se puso a prueba el software por espacio de una hora únicamente anunciándoles previo a esto, que lo que iban a usar era un juego.

Para la prueba con los niños, se determinó la siguiente lista de funcionalidades a evaluar siguiendo los criterios de aceptación de cada historia de usuario.

- Jugar el mini-juego Simon
- Jugar el mini-juego Secuencia de Frutas
- Jugar el mini-juego ¿Dónde están las perlas?
- Jugar el mini-juego Alimenta los Peces
- Revisar trofeos

- Enviar correo
- Acceder menú juego
- Entrar y salir de pantalla completa
- Volver a menú anterior
- Abrir, revisar y cerrar tutorial
- Acceder Trofeos
- Volver a jugar
- Revisar puntaje fin partida

Se elaboró también un test de la aplicación el cual se les pidió que llenaran al final de la prueba.

**TEST DE LA APLICACIÓN INGENIUM: Programa de entrenamiento de las habilidades cognitivas: memoria y atención.**

1. ¿Te agradó el tema de la aplicación?
  - Si
  - No
2. ¿Es para ti fácil de entender cómo usar la aplicación para realizar las diferentes actividades como jugar, enviar correos, acceder a los tutoriales?
  - Si
  - No
3. ¿La información dada y las ayudas, son suficientes para entender el funcionamiento de la aplicación?
  - Si
  - No
4. ¿Es fácil de identificar y ubicar todos los elementos de las pantallas?
  - Si
  - No

5. ¿Te parece fácil la navegación entre las pantallas, por ejemplo de menús a juegos?
- Si
  - No
6. ¿Consideras que es fácil navegar en cada pantalla, encontrar los elementos de esta, como botones, texto, campos para llenar y elementos clickeables en los juegos?
- Si
  - No
7. ¿Es fácil de aprender cómo jugar cada uno de los juegos y en general a usar la aplicación?
- Si
  - No
8. ¿Te parece agradable el diseño visual de cada juego y en general de toda la aplicación (imágenes, animaciones, elementos, botones, fondos y colores)?
- Si
  - No
9. ¿Durante la actividad, la aplicación funcionó siempre bien?, si tu respuesta es no y observaste algún error por favor menciónalo en el espacio asignado.
- Si
  - No
- 
10. ¿La actividad te pareció de carácter netamente académico?
- Si
  - No
11. ¿Te sientes motivado a usar de nuevo la aplicación?
- Si
  - No

12. ¿Qué le agregarías a la aplicación, qué piensas que le hace falta?

---

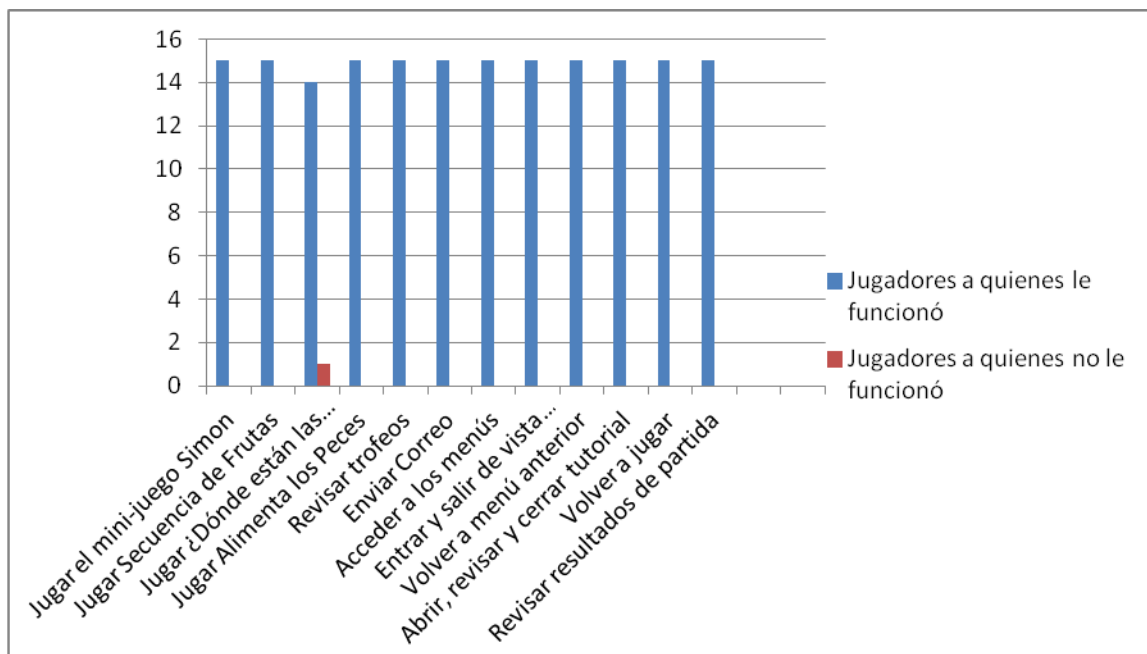
---

## ANÁLISIS DE LOS RESULTADOS

La funcionalidad del software se probó por observación; se le pidió a los niños que realizaran las tareas que describe la lista de funcionalidades a probar, y se iba anotando a cuántos de ellos les funcionaba bien.

Durante la prueba se observó solo un error en el juego ¿Dónde están las perlas?, el cual fue corregido posteriormente. El análisis general puede verse en la figura 41.

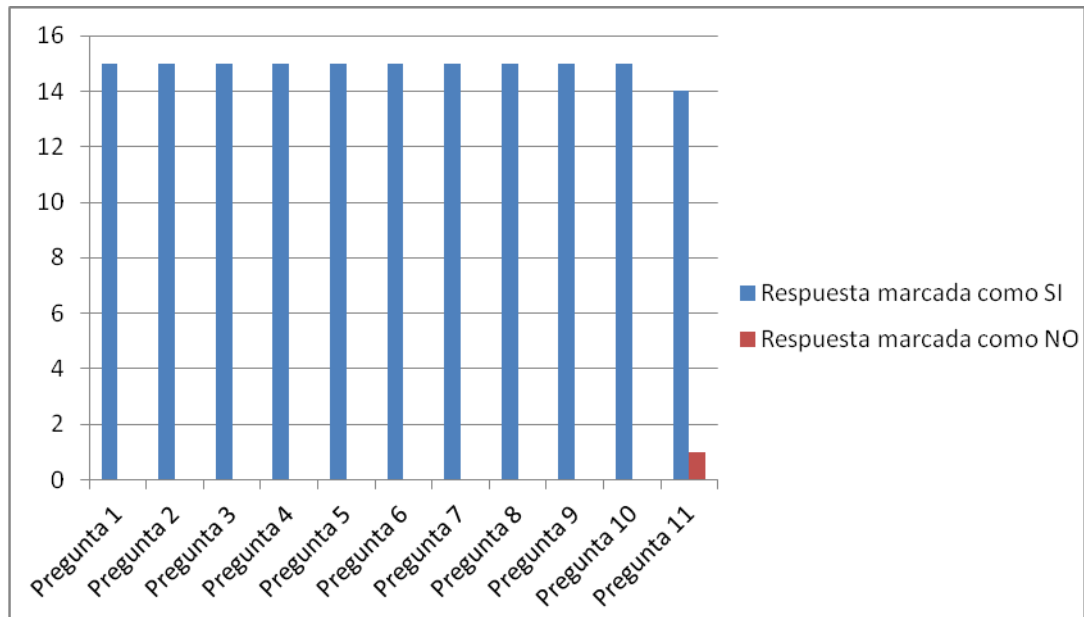
Figura 41. Análisis de funcionalidades



Los resultados de probar la funcionalidad fueron satisfactorios, en general se logró una aplicación buena funcionalmente. El error que en su momento hubo se debió a la ausencia de una línea de código que luego fue añadida.

Por otra parte, para evaluar la usabilidad del software se analizó las respuestas del test que debían responder. El análisis de esto se aprecia en la figura 42.

Figura 42. Análisis de Usabilidad



Del análisis de usabilidad del software, puede observarse que en este aspecto el software también muestra acierto. El software es fácil de entender, usar y aprender. Todos los elementos en él resultaron ser intuitivos, y en la mayoría de los jugadores, no hubo la necesidad siquiera de usar los tutoriales para jugar. Además de esto se pudo observar un gran interés por la aplicación, incluso el deseo de volverlo a jugar en sus casas. Tal cosa es posible porque el juego está disponible en la web y cada niño creó su usuario para la prueba.

En la pregunta numero 12 donde se preguntaba qué le agregarían a la aplicación, la mitad de los jugadores contestaron que nada, y la otra mitad sugirieron más niveles, más juegos, e incluso surgió de uno de ellos la idea de jugar dentro del castillo que tiene el planeta.

Para concluir, se observó que el diseño visual era el correcto, los botones, colores, e imágenes utilizadas fueron las apropiadas y, por lo tanto, no hubo necesidad de

explicarles mucho. Sin embargo, si surgía alguna duda por parte de los niños, se les aclaraba.

## CONCLUSIONES

- La metodología de Desarrollo ágil, Scrum es una elección correcta cuando se desarrolla software con muchas funcionalidades, permitiendo a los desarrolladores concentrarse más en el desarrollo de la aplicación y documentando solo lo estrictamente necesario.
- La herramienta de desarrollo de juegos Unity3D es una herramienta versátil y fácil de usar que agiliza el trabajo del desarrollador, haciendo sencilla la integración de la parte grafica del juego y la lógica del mismo, el uso conjunto de esta herramienta con la metodología ágil Scrum hace que el proyecto se desenvuelva de una manera rápida y fluida.
- El buen diseño visual constituye uno de los puntos fuertes de una herramienta juego, ya que conlleva que la aplicación se torne consistente, intuitiva, predecible y sobre todo, muy fácil de usar, estos criterios son claves a la hora de lidiar con una población como la de niños de grados de básica primaria; este resultado de diseño visual puede hacer que funcionalidades como tutoriales sean poco necesarios.
- Los niños de básica primaria disfrutaban de las aplicaciones con alto contenido grafico, movimiento, sonido, animaciones, colores vivos, imágenes fáciles de entender, y metáforas geográficas que estimulen la imaginación y la aventura.
- La usabilidad debe primar sobre la funcionalidad como eje en el desarrollo de una aplicación cuya población objetivo es infantil, sin embargo, el buen funcionamiento de la aplicación no debe ser descuidada, como se pudo

observar en la prueba piloto los niños se pierden interés cuando encuentran errores o algún retardo.

## RECOMENDACIONES

- Sobre la base actual del Proyecto, se puede aumentar la cantidad de juegos que tiene, tanto en las categorías de entrenamiento que ya posee, como en otras nuevas, que se puedan agregar, entre las que se encuentran, el control cognitivo, la velocidad de procesamiento, el razonamiento y otras, esto resultaría en una aplicación aun más completa para el entrenamiento cognitivo.
- Se recomienda para futuros proyectos con alto contenido de gráficos 3d como el presente, que se usen modelos optimizados con relativamente pocos polígonos y vértices ya que modelos de otra naturaleza aumenta los requerimientos de la máquina para dibujarlos, así mismo se recomienda la utilización del motor para juegos Unity 3d para el desarrollo de este tipo de aplicaciones dado que es una herramienta muy potente y fácil de usar que cuenta además, con una licencia libre.
- Se recomienda a la comunidad de la EISI incentivar el desarrollo de proyectos de esta naturaleza, ya que estos son realizados con poca frecuencia y la experiencia de su realización pueden enriquecer al ingeniero de sistemas UIS en varios campos del conocimiento relativos a su disciplina, que seguramente le serán de mucha utilidad en su vida profesional.
- Dado el trasfondo científico en el campo de la neuropsicología de este proyecto, puede ser utilizado para futuras investigaciones referentes a cuestiones de la memoria trabajo y la atención selectiva en niños de básica primaria.
- Aunque esta herramienta está diseñada para el uso de niños de básica primaria sin realizar excepciones de ningún tipo, se recomienda realizar

pruebas en poblaciones de niños de estos grados con algún tipo de déficit de memoria o atención medicamente diagnosticado como motivo de futura investigación.

## REFERENCIAS BIBLIOGRÁFICAS

1. ESCOBAR, G. E. (1999). LUDOMÁTICA: DISEÑO GRÁFICO PARA AMBIENTES EDUCATIVOS LÚDICOS, CREATIVOS, COLABORATIVOS E INTERACTIVOS. Programa para la Investigación Científica y Tecnológica III Etapa Ludomática, 13-30.
2. Michael Scanlon, K. S. (2007). Using the Web for Behavioral Research and Intervention: Evidence from Cognitive Training. San Francisco, CA: Lumos Labs.
3. ESCOBAR, G. E. (1999). DISEÑO GRÁFICO PARA AMBIENTES EDUCATIVOS E INTERACTIVOS PARA LOS NIÑOS Y NIÑAS DEL PROYECTO LUDOMÁTICA. Informática Educativa UNIANDES - LIDIE , 5-14.
4. Sarkar, K., Drescher, D., & Scalon, M. (19 de Enero de 2007). Working memory improvement following web-based cognitive training. Recuperado el 20 de Noviembre de 2011, de lumosity: [http://i.lumosity.com/pdf/working\\_memory.pdf](http://i.lumosity.com/pdf/working_memory.pdf)
5. Scanlon, M., Drescher, D., & Sarkar, K. (19 de Enero de 2007). Improvement of Visual Attention and Working Memory Through a Web-based Training Program. Recuperado el 19 de Enero de 2012, de Lumosity: [http://i.lumosity.com/pdf/visual\\_attention.pdf](http://i.lumosity.com/pdf/visual_attention.pdf)
6. Herrera, M., & Latapie, I. (9 de Marzo de 2010). Diseñando para la educación. Recuperado el 16 de Enero de 2012, de nosolousabilidad.com: [http://nosolousabilidad.com/articulos/disenio\\_educacion.htm](http://nosolousabilidad.com/articulos/disenio_educacion.htm)

7. Hassan, Y. (21 de Septiembre de 2004). Diseño web orientado a niños. Recuperado el 16 de Enero de 2012, de nosolousabilidad.com: [http://www.nosolousabilidad.com/articulos/disenio\\_orientado\\_ninos.htm](http://www.nosolousabilidad.com/articulos/disenio_orientado_ninos.htm)
8. Glende, M. H. (2006). Use ofr the electronic Game SIMON as a Measure of Working Memory Span in College Age Adult. Journal of Behavioral and Neuroscience Research , 1-7.
9. McLeod, S. (2008). simply psychology. Recuperado el 14 de Junio de 2012, de simply psychology: <http://www.simplypsychology.org/working%20memory.html>.
10. Quijano, J. (11 de 03 de 2012). <http://www.genbetadev.com>. Recuperado el 12 de 04 de 2012, de <http://www.genbetadev.com>: <http://www.genbetadev.com/metodologias-de-programacion/pilas-de-producto-hablando-de-scrum>
11. Quijano, J. (01 de 02 de 2012). <http://www.genbetadev.com>. Recuperado el 15 de 03 de 2012, de <http://www.genbetadev.com>: <http://www.genbetadev.com/metodologias-de-programacion/historias-de-usuario-una-forma-natural-de-analisis-funcional>

## ANEXOS

### ANEXO A. Glosario de Términos de Unity3D

**Prefabricado:** Es un GameObject reutilizable almacenado en la vista Proyecto. Los prefabricados se pueden insertar en cualquier número de escenas, varias veces por escena. Cuando se agrega un Prefab a una escena, se crea una instancia del mismo. Todas las instancias prefabricadas están vinculadas al Prefab original y son esencialmente clones del mismo. No importa cuántas veces exista en el proyecto, al realizar cualquier cambio en el Prefab original el cambio se aplicara a todos los clones.

**Gui:** Unity 3D ofrece múltiples formas y objetos que permiten al usuario crear las interfaces de usuario, entre estos se encuentran:

**GUITexture:** se muestran como imágenes planas en 2D. Están hechas especialmente para los elementos de la interfaz de usuario, botones o adornos. Su posicionamiento y escalado se realiza solo a lo largo de los ejes x e y, y se mide en coordenadas de la pantalla, en lugar de las coordenadas del mundo.

**GUI Text:** muestra texto de cualquier fuente en la pantalla.

**UnityGUI:** permite crear una amplia variedad de interfaces gráficas de usuario. Todo el control y creación de este tipo de elementos se crea mediante código.

**Sistemas de Partículas:** Las partículas son esencialmente imágenes 2D en el espacio 3D. Se utilizan principalmente para efectos como el humo, el fuego, gotas de agua, o las hojas. Un sistema de partículas está compuesto de tres

componentes separados: emisor de partículas, Animador de partículas, y un representador de partículas.

**Material:** Se aplican sobre los modelos utilizados en el proyectos y básicamente son un conjunto de instrucciones que le permite a unity saber cómo interpretar la luz y la sombra sobre el modelo al que está relacionado.

**Escena:** Las escenas contienen los objetos del juego. Se pueden utilizar para crear un menú principal, los niveles individuales, y cualquier otra cosa. Cada archivo de escena único es un nivel único. En cada escena, se puede crear de cero el entorno deseado para el juego.

**Texturas:** Las texturas traen los modelos, las partículas y las interfaces a la vida, son archivos de imagen o película que se encargan de envolver los objetos.

**Cámara:** Las cámaras son los dispositivos que capturan y muestran el mundo al jugador. Mediante la personalización y la manipulación de cámaras, se pueden hacer múltiples efectos como el falso 2D o la impresión de primera persona y se pueden tener tantas cámaras en una escena como se quiera. Se pueden en configurar en cualquier orden, y, en cualquier lugar de la pantalla.

**Itween:** ITween es un sistema de animación simple, potente y fácil de usar para Unity 3d, que permite crear múltiples efectos de movimiento e incluso puede ser usado para animar elementos GUI, es un solo archivo de C # que se puede utilizar con cualquiera de los lenguajes de programación que soporta la Unity, así como con todas sus versiones.

## ANEXO B. Pantalla de Tutoriales

Pantalla de tutorial del juego SIMON:



Secuencia de imágenes que muestra el tutorial:



En este juego, debes memorizar el orden en que se iluminan y suenan los hongos.

En la primer ronda, uno de los cuatro hongos se ilumina y suena, debes memorizar cuál fue.

Ahora haz clic sobre el hongo que se iluminó previamente, y en la siguiente ronda, se iluminará el mismo y uno más.

En cada ronda debes recordar un hongo más y repetir la serie. Trata de recordar los que más puedas, pues pierdes cuando te equivocas.

Pantalla de tutorial del juego SECUENCIA DE FRUTAS:



Secuencia de imágenes que muestra el tutorial:



En este juego, tienes que memorizar el orden en que aparecen las frutas.

Diferentes frutas aparecen una después de otra, de izquierda a derecha, en el área superior. Trata de recordar el orden exacto.

Ahora haz clic sobre las frutas en el mismo orden en que aparecieron.

Resuelve todas las rondas mostradas (1), hasta completar el juego

Pantalla de tutorial del juego ¿DÓNDE ESTÁN LAS PERLAS?



Secuencia de imágenes que muestra el tutorial:



En este juego debes permanecer atento Las almejas se cierran y cambian de po-Ahora haz clic sobre la almeja que creas Resuelve todas las rondas mostradas(1)  
a las almejas que en su interior contienen sición tres veces. Debes fijarte en donde contiene la perla.  
una perla. quedará la que contiene la perla. hasta completar el juego.

Pantalla de tutorial del juego ALIMENTA LOS PECES:



Secuencia de imágenes que muestra el tutorial:



En este juego debes alimentar a los peces que nadan en el fondo del mar. Tienes las galletas exactas para alimentar a cada pez solo una vez.

Haz clic sobre el pez para alimentarlo. Tienes que esperar unos segundos que el reloj te indica, para alimentar al próximo pez.

Recuerda que tienes la comida exacta para alimentarlos solo una vez, así que no pierdas de vista los peces que ya has alimentado.

Si alimentas los peces correctamente, en la siguiente ronda tendrás un pez más. Resuelve todas las rondas mostradas (1) hasta completar el juego.

¿POR QUÉ BRILLAN LOS PECES?  
 Resplandor verde: Pez alimentado correctamente.  
 Resplandor rojo: Pez alimentado incorrectamente.  
 Resplandor azul: Pez no alimentado.

## ANEXO C. Imágenes de las Pruebas de Funcionalidad y Usabilidad

Usuarios ingresando a la aplicación:



Usuarios jugando SIMON:



Usuarios jugando SECUENCIA DE FRUTAS:



Usuarios jugando ¿DÓNDE ESTÁN LAS PERLAS?:



Usuarios jugando ALIMENTA LOS PECES:

