

**Universidad de las Ciencias Informáticas**  
**Facultad 6**



**Título: VirtualRelax, plataforma para integrar módulos de entornos virtuales,  
para apoyar a los especialistas en el tratamiento del estrés.**

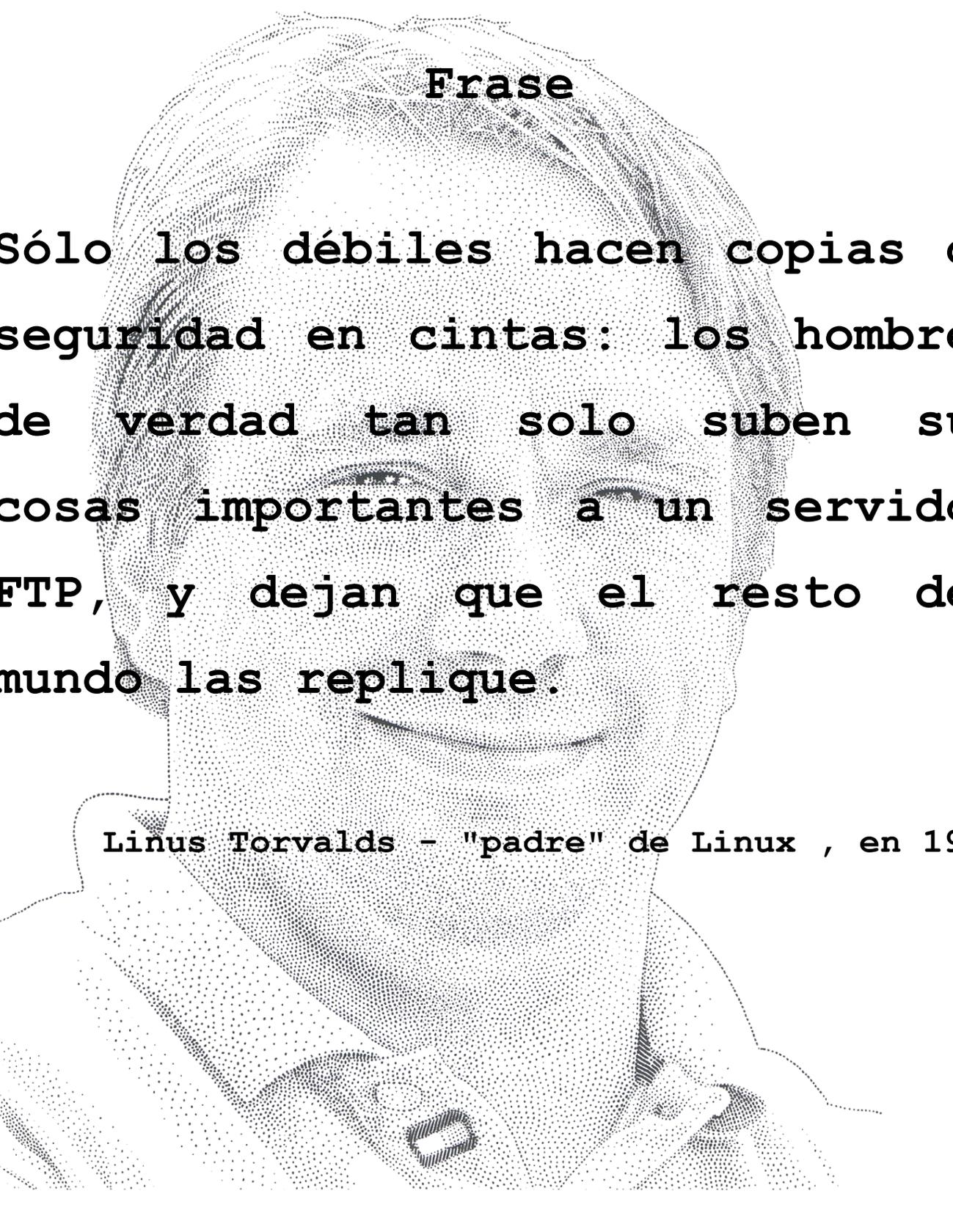
Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas

**Autor(es):** Daniel Valdés Fernández

**Tutor(es):** Ing. Roberto Elías Pérez Ozete

**Consultor:** Msc. Omar Correa Madrigal

La Habana, febrero de 2015



## Frase

Sólo los débiles hacen copias de seguridad en cintas: los hombres de verdad tan solo suben sus cosas importantes a un servidor FTP, y dejan que el resto del mundo las replique.

Linus Torvalds - "padre" de Linux , en 1996

## DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis que tiene por título: VirtualRelax, plataforma para integrar módulos de entornos virtuales, para apoyar a los especialistas en el tratamiento del estrés, y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

**Daniel Valdés Fernández**

**Ing. Roberto Elías Pérez Ozete**

---

Firma del Autor

---

Firma del Tutor

## DATOS DE CONTACTO

### **Generales del Tutor.**

**Nombre y Apellidos:** Roberto Elías Pérez Ozete.

**Especialidad:** Ingeniero en Ciencias Informáticas

#### **Síntesis del tutor:**

- Trabajador del centro Vertex de la Universidad de las Ciencias Informáticas.
- Categoría docente:
- 1 año de experiencia.

**Correo Electrónico:** reperez@uci.cu.

### **Generales del Consultor.**

**Nombre y Apellidos:** Omar Correa Madrigal.

**Especialidad:** Máster en Informática Aplicada.

#### **Síntesis del consultor:**

- Actual director del centro Vertex de la Universidad de las Ciencias Informáticas.
- Categoría docente: Asistente
- 9 años de experiencia.

**Correo Electrónico:** ocorrea@uci.cu.

## AGRADECIMIENTOS

Me gustaría agradecer de forma general a todo aquel que aportó su granito de arena en el desarrollo de este trabajo. A los mis amigos los elefantes azules, ojo yo no me drogo, en la uci hay elefantes azules, Ale, Edo, El Hacker y otros. A mis amigos Raudel y Alfredo, que intentan batir mi record de permanencia, chicos es todo suyos, aquí lo dejo yo. A Krlos y Deby, que están grabados el uno en la otra y viceversa, lo sé porque fui yo quien lo hizo. A los de una época pasada, compañeros de mis inicios en la universidad, los grupos 2 y 4, que casi hay que hacer el grupo 24, algunos que ya no se encuentran aquí y otros que no pudieron llegar a esta meta. A mi tutor, que antes de serlo ya era mi amigo y que así seguirá por tiempo indefinido. A los del año de mi hermano, que me contaron como a uno más en todo lo que hicieron. Adrián "Sevilla" ya no cuentas entre mis amigos, ya yo llamo a tu madre, mamá, tu eres un hermano y como tú, tu hermana, Aldito y los demás. A mi novia, que aunque hace poco que me acompaña, cito las palabras de Fidel Castro cuando dijo, "Nadie ha hecho tanto en tan poco tiempo". También gracias a la guitarra y el kfe, que tanto nos unió, la música no tiene tantos acordes como buenos ratos he pasado yo con ella y con quienes la tocan y disfrutan. Pueden quedárseme muchos y más con los años de " experiencia " que poseo, en tanto tiempo se puede llegar a conocer a mucha gente, para más referencia en directorio.uci.cu filtren la búsqueda por amigos de Daniel, aunque ya muchos no estén en esa base de datos.

No podría faltarme mi familia, que estuvo pendiente a cada paso del proceso de mi graduación. Posiblemente esto cuente como misión internacionalista, ya que el apoyo y la preocupación de mis familiares, tienen más horas de vuelo que cubana de aviación. Mis tíos y primos, mis abuelos, estos últimos capaces de enseñar que aun con 87 años vale la pena seguir estudiando, Estela y Marcelo descansan en paz, ya lo logré, Carmen, tu verás graduarse a mis hijos. A mi madre que aun sin tener creencia alguna en religiones, puso vasos espirituales y hasta prendió velas, velando por mi bienestar. A mi padre que con cero conocimientos de tecnología, se sentaba a mi lado y me decía, ¿en qué te puedo ayudar? Mi padrastro, nunca he sabido decir esa palabra sin sentir que de cierta manera suena ofensiva, sobre todo porque sé que me has querido como a un hijo y no como a un hijastro. Por último y no por eso menos

importante (nada más alejado de la realidad), mi hermano David, mi sangre, mis huesos, aun cuando no te sabias los números ya yo contaba contigo, no se me ocurren muchas situaciones de mi vida en las que tu no formes parte, gracias por estar ahí.

GRACIAS A TODOS

## DEDICATORIA

A mis padres, biológicos y no biológicos. Tengo más padres de los que me tocaban, pero los suficientes para ser quien soy. Gracias a todos, juntos son el equipo de desarrollo del software "ingeniero" que soy ahora.

## **RESUMEN**

Este trabajo de diploma se orientó a la creación de una plataforma, capaz de ejecutar y configurar los módulos de entornos virtuales creados por el centro VERTEX. Estos módulos son una herramienta para apoyar a los especialistas que aplican terapias de relajación a pacientes que presentan síntomas de estrés y patologías similares. Debido a las características que presenta el desarrollo de esta plataforma el centro decidió que se van a utilizar la metodología de desarrollo de software eXtreme Programming(XP) y el motor de videojuegos Unity 3D como principal herramienta, las cuales satisfacen las necesidades presentadas para el cumplimiento de los objetivos. Se tuvo en cuenta la existencia a nivel mundial de otros programas utilizados para la relajación con características semejantes, la documentación referente a los tratamientos de relajación y las técnicas más utilizadas para atender dichas patologías. El impacto social de los resultados de esta investigación radica en que esta nueva plataforma podrá incluir los módulos existentes y aquellos que puedan ser creados en un futuro, favoreciendo su aplicación en las terapias para tratar el estrés.

### **PALABRAS CLAVE**

Plataforma, aplicación, estrés, terapia, tratamiento, relajación.

## **ABSTRACT**

This dissertation is focused on creating a platform capable of running modules and configure virtual environments created by the center VERTEX, to be used to support specialists relaxation therapies applied to patients with symptoms of stress and related pathologies. Due to the characteristics inherent in the development of this platform, the center decided it will use the methodology of software development eXtreme Programming (XP) and engine Unity 3D gaming as the main tool, which meet the needs presented for compliance objectives. Worldwide the existence of other programs used for relaxation with similar characteristics, documentation relating to treatments and relaxation techniques used to address these diseases was taken into account. The social impact of the results of this research is that this new platform will include existing modules and those that may be created in the future, favoring its application in therapies to treat stress.

## **KEYWORDS**

Platform, application, stress, therapy, software, treatment, relaxation.

## ÍNDICE

|   |    |
|---|----|
| INTRODUCCIÓN.....   | 10 |
| CAPÍTULO 1. FUNDAMENTOS TEÓRICOS DE LA INVESTIGACIÓN .....                | 14 |
| 1.1. Plataformas informáticas .....                                       | 14 |
| 1.2. Conceptos asociados al dominio del problema. ....                    | 15 |
| 1.3. Software de Relajación .....   | 18 |
| 1.4. Simulación de entornos virtuales .....                               | 20 |
| 1.5. Entornos virtuales para terapia de relajación .....                  | 24 |
| 1.6. Metodología XP para el desarrollo de Software.....                   | 27 |
| 1.7. Herramientas .....   | 28 |
| 1.8. Motor gráfico .....  | 30 |
| Conclusiones del Capítulo .....   | 33 |
| CAPÍTULO 2. PROPUESTA DE SOLUCIÓN .....                                   | 35 |
| 2.1. Descripción de la plataforma propuesta.....                          | 35 |
| 2.2. Componentes a usar en la Implementación.....                         | 35 |
| 2.3. Requisitos no Funcionales .....                                      | 38 |
| 2.4. Fase de Exploración .....  | 39 |
| 2.5. Planificación y Entrega.....   | 41 |
| 2.6. Diseño del Sistema .....   | 42 |
| 2.7. Plan de Iteraciones .....  | 44 |
| 2.8. Bases Estructurales para los Futuros Módulos .....                   | 45 |
| CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA .....                     | 49 |
| 3.1. Pruebas de Aceptación.....   | 49 |
| 3.2. Validación de la plataforma .....                                    | 51 |
| 3.3. Modelo de Entrevista utilizada para la obtención de información..... | 53 |
| Conclusiones del Capítulo .....   | 54 |
| CONCLUSIONES .....  | 55 |
| RECOMENDACIONES.....  | 56 |

|                            |    |
|----------------------------|----|
| BIBLIOGRAFÍA.....          | 57 |
| GLOSARIO DE TÉRMINOS ..... | 60 |
| ANEXOS.....                | 62 |

## ÍNDICE DE TABLAS Y FIGURAS

|  |    |
|--|----|
| Tabla. 1: HU Configurar y modificar XML.....                                     | 40 |
| Tabla. 2: HU Cargar entorno.....   | 40 |
| Tabla. 3: Prueba de Aceptación 6.....  | 49 |
| Tabla. 4: Prueba de Aceptación 7.....  | 50 |
| Tabla. 5: HU Verificación del correcto funcionamiento del Entorno de Pecera..... | 64 |
| Tabla. 6: HU Verificación del correcto funcionamiento del Entorno de Playa.....  | 65 |
| Tabla. 7: HU Acceso y modificación de la variable Agregar Garza.....             | 66 |
| Tabla. 8: HU Acceso y modificación de la variable Agregar Gaviota.....           | 66 |
| Tabla. 9: HU Acceso y modificación de la variable Tipo de Cámara.....            | 67 |
| Tabla. 10: Plan de Entregas.....   | 67 |
| Tabla. 11: Plan de entregas por Iteraciones.....                                 | 68 |
| Tabla. 12: Patrones de Diseño.....   | 68 |
| Tabla. 13: Tarjeta CRC del Script "LoadLevel_OnClick.cs".....                    | 69 |
| Tabla. 14: Tarjeta CRC del Script "WriteXML.cs".....                             | 70 |
| Tabla. 15: Tarea de investigación 1.....   | 71 |
| Tabla. 16: Tarea de investigación 2.....   | 71 |
| Tabla. 17: Tarea de investigación 3.....   | 72 |
| Tabla. 18: Tarea de investigación 4.....   | 72 |
| Tabla. 19: Tarea de investigación 5.....   | 73 |
| Tabla. 20: Tarea de investigación 6.....   | 73 |
| Tabla. 21: Tarea de investigación 7.....   | 74 |
| Tabla. 22: Tarea de investigación 8.....   | 74 |
| Tabla. 23: Prueba de Aceptación 1.....   | 75 |
| Tabla. 24: Prueba de Aceptación 2.....   | 75 |
| Tabla. 25: Prueba de Aceptación 3.....   | 76 |
| Tabla. 26: Prueba de Aceptación 4.....   | 77 |
| Tabla. 27: Prueba de Aceptación 5.....   | 77 |



# INTRODUCCIÓN

En la actualidad, con el progreso de la humanidad, el desarrollo de las grandes ciudades y las tecnologías ha dado lugar a que las personas estén expuestas a constantes estados de tensión, vacío y depresión. La búsqueda de algunos entornos naturales para lograr el relajamiento debido al estrés ha sido de muy difícil acceso debido a las características del desarrollo industrial existente en el mundo actual. Para dar respuesta a esto, especialistas en la materia han buscado alternativas para lograr satisfacer las necesidades de los pacientes que vienen con problemas de esta índole.

Después de largas investigaciones ha surgido una gran diversidad de tratamientos para la relajación que consisten en un profundo descanso físico, psíquico y afectivo, inducido mediante métodos elaborados con los que cuentan los profesionales de esta rama para la asistencia a los pacientes. Como herramientas de apoyo a estas terapias existen varias aplicaciones informáticas para servir de ayuda a las terapias de relajación; estas en su mayoría son protectores de pantalla, videojuegos o videos 3D que ayudan al paciente a aliviar el estado físico-mental en el que se encuentran.

En Cuba se emplean diferentes métodos enfocados en este tema los cuales incluyen videos e imágenes de escenarios naturales, audios ambientales y canciones para las terapias de relajación de pacientes con problemas de estrés. Teniendo en cuenta las experiencias obtenidas en la participación del centro en terapias realizadas para el tratamiento del estrés, donde se utilizaron las alternativas informáticas antes mencionadas para la relajación visual como DreamAcuario (protector de pantalla), se demostró que estas son capaces de relajar. Estas presentan dificultades en su configuración, dado que no permiten modificaciones en tiempo de ejecución y esto provoca desconcentración en el paciente, interrumpiendo la terapia.

En el centro VERTEX de la Facultad 5, se han realizado dos entornos virtuales para las terapias de relajación, una playa y una pecera virtual. Estos módulos pertenecen a un proyecto aun en desarrollo conceptual, que contendrá todos los entornos virtuales necesarios, los cuales permiten un alto nivel de adaptación en las terapias de relajación. Para el proceso de desarrollo de estos módulos, se utilizaron como principal herramienta el motor gráfico Unity 3D y la metodología de desarrollo de software eXtreme

Programming, las cuales son propuesta para el desarrollo de esta plataforma manteniendo así la línea de trabajo del centro.

### **Situación problemática**

El centro VERTEX, basándose en las experiencias adquiridas en los diferentes encuentros con especialistas en terapias de relajación, estableció el uso de alternativas informáticas para apoyar los tratamientos para aliviar el estrés. Este propuso crear distintos módulos de simulación virtual de entornos naturales.

La creación de los primeros entornos se realizó de manera independiente y sin un control sobre la uniformidad de los parámetros configurables del mismo. Estos parámetros son los responsables de los disímiles cambios del ambiente en cuestión, para darle forma a la simulación. Atendiendo a la difícil organización de la información presente en estos precursores, se crearon dos nuevos entornos con características de configuración similares y de fácil acceso, siendo probados en una de las terapias de relajación aplicadas en el centro Martin Luther King Jr.

A pesar de los beneficios alcanzados con la ejecución de estos módulos en las terapias, los mismos no permiten unificar las informaciones obtenidas de ellos y no proveen al usuario facilidades de uso y configuración.

Debido a la situación anteriormente planteada se define el siguiente **problema a resolver**: ¿Cómo unificar las informaciones obtenidas de los módulos producidos por el centro VERTEX, proporcionando facilidades de uso y configuración de los mismos a los especialistas en terapias de relajación?

Para dar solución al mismo se plantea como **objetivo general**: Desarrollar una plataforma para unificar las informaciones obtenidas en los módulos de simulación virtual y configurar las características para su uso en las terapias de relajación de pacientes con estrés.

Por lo que se puede plantear como **objeto de estudio**: Plataformas para terapias de relajación.

Delimitando como **campo de acción**: Los módulos producidos por Vertex para las terapias de relajación.

A partir de lo anteriormente planteado se propone como **idea a defender**: Con el desarrollo de la plataforma VirtualRelax, siendo esta capaz de unificar la información contenida en los módulos desarrollados por Vertex y permitiendo configurarlos, entonces facilitará la obtención de datos por especialistas para mejorar las terapias de relajación aplicadas a pacientes con estrés y patologías similares.

**Posibles resultados:** Una plataforma para unificar la información que muestran los módulos de simulación virtual, acceder, y configurar las características de estos módulos para su uso en las terapias de relajación de pacientes con estrés. Las bases estructurales con la que deben ser creados los futuros módulos, para que puedan ser incluidos en la aplicación y le permitan a esta configurarlos de manera sencilla.

Para darle cumplimiento al objetivo general se definen las siguientes **tareas de investigación**:

- Realizar el marco teórico de la investigación, citando los principales conceptos y tecnologías relacionadas con la temática de la investigación.
- Caracterizar los elementos que conforman un entorno virtual para las terapias de relajación.
- Seleccionar y caracterizar las herramientas a emplear en el desarrollo de la solución.
- Implementación de una plataforma que responda a las necesidades de los especialistas de las terapias de relajación.
- Evaluación del grado de aceptación de los especialistas como validación de la aplicación.

Para obtener los conocimientos necesarios para desarrollar la investigación se utilizaron los siguientes métodos de investigación científica:

## **Métodos teóricos**

**Histórico - Lógico:** El uso de este método permitió realizar un estudio del estado del arte de los principales conceptos relacionados con las terapias de relajación, así como el desarrollo de las tecnologías que fueron utilizadas para el desarrollo de la solución propuesta.

**Analítico - Sintético:** La práctica de este, permitió analizar de forma individual cada una de las ramas que abarca esta investigación como son los términos terapia, estrés, relajación, software, simulación y entornos virtuales, para luego sintetizarlos en la solución.

## **Métodos empíricos**

**Observación Científica:** Se empleó como método referencial que ayudó a diagnosticar el problema de la investigación, al realizar una observación simple de los entornos virtuales creados por el centro VERTEX y otras aplicaciones utilizadas en terapias de relajación. Este procedimiento aportó conocimientos para establecer las características y elementos fundamentales para la solución y la comprobación de la idea a defender de este trabajo.

El desarrollo de la investigación está conformado por tres capítulos:

- **Capítulo 1: Fundamentos Teóricos de la Investigación,** se realizará un estudio de los principales conceptos y herramientas relacionados con las terapias de relajación y un análisis de soluciones propuestas por el centro Vertex. De igual manera se definirán las herramientas y tecnologías a utilizar para.
- **Capítulo II Propuesta de solución,** se describe la propuesta de solución a partir de la elaboración del campo de acción y detallando los artefactos generados como parte de la metodología seleccionada para el desarrollo de la Plataforma VirtualRelax.
- **Capítulo III Validación de la solución propuesta,** está dedicado a la validación de la investigación desarrollada por parte de los especialistas del centro Vertex y especialistas en terapias de estrés y otras patologías.

# CAPÍTULO 1. FUNDAMENTOS TEÓRICOS DE LA INVESTIGACIÓN

En el presente capítulo se precisan elementos teóricos que sustentan la investigación y desarrollo del software, temas teóricos importantes relacionados con el producto y varios conceptos asociados al mismo, lenguajes y herramientas para la creación del software. Se relacionan los temas referentes a la patología del estrés, técnicas y terapias asistidas por profesionales para tratamiento, así como una investigación que recoge en si la existencia de herramientas o aplicaciones dedicadas a dicho tratamiento y los aportes que aún se le pueden hacer al tema. Se presentarán los conceptos y tendencias actuales obtenidos durante la investigación que referencian la solución a la problemática que propone el tema de este trabajo.

## 1.1. Plataformas informáticas

Se le conoce como plataforma en informática a un sistema que sirve como base para hacer funcionar determinados módulos de hardware o de software con los que es compatible. Dicho sistema está definido por un estándar alrededor del cual se determina una arquitectura de hardware y una plataforma de software (incluyendo entornos de aplicaciones). Al definir plataformas se establecen los tipos de arquitectura, sistema operativo, lenguaje de programación o interfaz de usuario compatibles. (aprendizaje, 2015)

Existen básicamente dos **tipos de plataformas**:

### ***Arquitectura de hardware***

- x86 (las típicas PC).
- Consola de videojuego (Xbox, Playstation).

### ***Framework de software***

- Sistemas operativos (Windows, Linux, Mac OS).
- Plataformas de software (Java - JDK y JRE, .NET Framework, Mozilla, Adobe).

En este trabajo se centra en las plataformas de software, teniendo en cuenta que no se construye ningún dispositivo o consola para dar solución al problema planteado al inicio.

## **1.2. Conceptos asociados al dominio del problema**

La vorágine de la cotidianidad, el exceso de trabajo y los distintos problemas que afronta la sociedad hoy en día, provocan que las personas sufran de un constante estado de estrés y este se puede definir como: “diversas reacciones físicas y emocionales recibidas por un sujeto, que le resultan excesivas y dará lugar a que comience a sentir ansiedad y a experimentar diversos síntomas. Este concepto proviene del término inglés stress y está asociado al estado que se produce a partir de hechos que generan agobio o angustia.” (Grupo de Definición, 2013)

Otra forma de definir el estrés es como el estado de gran tensión nerviosa, generalmente causado por un exceso de trabajo, que suele provocar diversos trastornos físicos y mentales. Situación en la que un organismo o alguno de sus órganos sufren presiones del medio o exigencias superiores a lo habitual, por lo que puede llegar a enfermar (Hierrezuelo, 2004).

El estrés representa una forma de agotamiento físico y/o mental, que producto al nivel que posea traerá como consecuencia una serie de afecciones al organismo. Limitando al individuo que lo padece, de realizar sus quehaceres en plena capacidad o de realizarlos de manera alguna. La exposición constante a situaciones atípicas a las de la zona de confort de una persona, puede llevar a su padecimiento.

Una de las vías más utilizadas para frenar o amortiguar los efectos del estrés es la relajación. El término relajación se entiende como el estado breve o largo de actividad metabólica, nerviosa y consciente reducida, que se puede medir y definir en el plano subjetivo, fisiológico y motor (Complementaria, 2000).

La relajación también pudiera definirse de acuerdo con otros autores como: “la acción y efecto de relajar o relajarse (aflojar, ablandar, distraer el ánimo con algún descanso). La relajación, por lo tanto, está asociada a reducir la tensión física y/o mental.” (Grupo de Definición, 2013)

El término relajación adquiere sus mayores funcionalidades en lo referente al tratamiento del estrés. También es relacionado con la entrada en un estado de comodidad el cual es medible en diversos planos

tales como el subjetivo, que su caracterización es una experiencia consciente de los estados afectivos, por ejemplo: la sensación de placer versus displacer, bienestar versus malestar, alegría versus tristeza. (Complementaria, 2000)

Entre los planos donde es medible la relajación se encuentran además el fisiológico, que se caracteriza por cambios viscerales, somáticos y corticales como cambios en el ritmo cardiaco, disminución de la tensión muscular, cambios de los ritmos electroencefalográficos, temperatura interna, etc. (Complementaria, 2000)

Este último plano brinda más recursos o variables para la medición de la relajación, entre ellos se encuentra el ritmo cardiaco o los latidos (pulsaciones) del corazón. Los centros Dr. Martin Luther King Jr. y el de rehabilitación nacional Julio Díaz, aplican diversas terapias de relajación basadas en los planos antes mencionados. De acuerdo con las experiencias adquiridas en estos lugares existen dos formas de medir los niveles de relajación, relacionadas con este elemento de tipo fisiológico. Si la medición de las pulsaciones al terminar la sesión de terapia dirigida a un solo paciente (Caso 1), muestra un resultado menor o igual a las pulsaciones iniciales, se puede considerar como aceptable, siendo no aceptable en el caso contrario. Si fuera dirigida a varios pacientes o un grupo de ellos (Caso 2) y el porcentaje de las personas en las cuales disminuye o se mantiene el número de pulsaciones finales con respecto a las iniciales es mayor que el porcentaje de las personas en las que el número de pulsaciones finales son altas con respecto a las iniciales, entonces el nivel de relajación es aceptable, en caso contrario es considerado no aceptable. Lo anteriormente expuesto se puede mostrar de la siguiente manera:

Caso 1: un paciente:

$$NR = \begin{cases} \text{Si } NPF \leq NPI & \longrightarrow \text{aceptable} \\ \text{Caso contrario} & \longrightarrow \text{no aceptable} \end{cases}$$

Donde NPI es el Número de Pulsaciones Iniciales, NPF es el número de Pulsaciones Finales y NR se define como Nivel de Relajación.

Caso 2: varios pacientes:

$$\begin{cases} \text{Si } (PPD + PPM) > PPA & \longrightarrow \text{aceptable} \end{cases}$$

$NR = \text{Si } (PPD + PPM) \leq PPA \rightarrow \text{no aceptable}$

Donde PPA es el Promedio de Pulsaciones que Aumentan, PPD es el Promedio de Pulsaciones que Disminuyen, PPM es el Promedio de Pulsaciones que se Mantienen y NR se define como Nivel de Relajación.

### **Formas de terapia de relajación**

Hoy en día existen varias clases de terapias de relajación, que utilizan una gran variedad de técnicas; sin embargo, la mayoría de ellas comparten ciertas características. En muchos de los métodos de este tipo, la persona opta por recostarse o adoptar una postura que permita la rápida relajación de los músculos, ubicándose en un lugar tranquilo y con los ojos cerrados. Los siguientes pasos difieren un tanto, dependiendo de la técnica a utilizar.

En el entrenamiento autogénico, respuesta a la relajación y en diferentes formas de meditación, la persona enfoca su mente hacia las sensaciones internas, como la respiración. Las variantes de imágenes dirigidas, emplean una visualización deliberada de escenarios o acciones, como caminar por una playa tranquila. Las técnicas de relajación progresiva en cambio, implican una disminución gradual de la tensión en los músculos. A continuación se exponen dos de los diversos tipos de terapias de relajación:

**Relajación Autoalusiva:** se centra en unificar todos los elementos que conforman la experiencia presente de forma simultánea, todos los niveles de la realidad presente se unifican en el ejercicio de relajación. (Barrios, 2013)

**Relajación Paradójica:** pretende la comprensión de uno mismo utilizando como principal herramienta la observación. La observación es un proceso, un cambio en la parte que genera una reestructuración de la totalidad y ese cambio total permite una nueva exploración de una nueva realidad potencial. (Barrios, 2013)

Donde la relajación autoalusiva es una relajación en busca de la unidad, es una técnica inclusiva que pretende tomar conciencia de todo aquello que sucede a la vez integrándolo en una realidad mayor y generando así una nueva realidad, que podrá a su vez ser integrada de nuevo, y así sucesivamente. Y la paradójica se basa en el principio fundamental de que los pacientes intenten llevar a cabo la conducta que están evitando. De esta manera, el proceso circular, que se mantiene a sí mismo, se rompe, puesto que el

intentar realizar la conducta no deseada es incompatible con la ansiedad anticipatoria y, por tanto, la neutraliza.

Todas estas técnicas se aprenden mejor con la ayuda de un profesional entrenado y el procedimiento estándar es integrarse a una clase de grupo complementada con una práctica regular en casa; por lo que, si el usuario es lo suficientemente diligente, la experiencia sugiere que podrá desarrollar la habilidad de entrar en un estado relajado a voluntad, incluso en medio de una situación muy estresante. (Tejeda, 2014)

### **1.3. Software de Relajación**

“Software” es una palabra que proviene del idioma inglés, pero que gracias a la masificación de uso, ha sido aceptada por la Real Academia Española (RAE). Según la RAE, un software es un conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar distintas tareas en una computadora. Se considera que el software es el equipamiento lógico e intangible de un ordenador. En otras palabras, el concepto de software abarca a todas las aplicaciones informáticas, como los procesadores de textos, las planillas de cálculo y los editores de imágenes. (Definición, 2013)

Después del análisis realizado, se define el concepto de software de relajación como las aplicaciones informáticas que constituyen herramientas para facilitar el trabajo en los procesos psicoterapéuticos dedicados a lograr altos niveles de relajación. (Hernández, 2014)

Un software de relajación no es más que una aplicación informática o conjunto de estas, que dadas sus características visuales y sonoras produce sensaciones de calma y bienestar a la persona que lo utilice.

#### **Tipos de software de relajación**

Los software de relajación existentes, debido a sus características o formas de ejecución, se pueden separar en tres grupos. Esto permite diferenciarlos entre sí y poder optar por el más beneficioso en un momento dado. Dependiendo de sus funcionalidades y tipos, estos grupos serán definidos en protectores de pantalla, software de audio y software de videojuegos. A continuación se muestran unos ejemplos de los mismos:

#### ***Refrescadores de Pantalla***

Existe una inmensa cantidad de estos hasta el momento, simulando diversas locaciones que van desde entornos sencillos como los de pecera, hasta algunos mucho más complejos como los que simulan locaciones enteras; playas, entornos selváticos, saltos de agua e incluyendo ambientes fantásticos que simulen lugares donde coexistan la paz y la tranquilidad. Algunos de ellos son:

- **Dream Aquarium (Anexos Figura 1):** es un acuario virtual y protector de pantalla que simula un acuario de agua dulce con un alto nivel de realismo. Muestra el comportamiento natural y el movimiento de sus habitantes. Estas son algunas de sus características: Aletas articuladas, movimiento de los ojos, branquias y boca, rayos de luz, ondulaciones de la tierra, movimiento suave de las plantas por los cuales los peces pueden nadar, sombras suaves, secuencias de burbujas configurables, alimentación automática. Añade más de un centenar de peces y cambia la configuración sin tener que salir del acuario. (Kapler, 2013)

### ***Software de Sonido***

Algunos tipos de aplicaciones solo son compendios o bancos de sonido que combinan estos para lograr un efecto tranquilizador y pausado, logrando así el estado de relajación. Como ejemplo existe:

- **Halotea (Anexos Figura 2):** La generación de los temas de sonido es única en la aplicación, sobre la base de repeticiones al azar con parámetros aleatorios de la reproducción de los sonidos individuales de alta calidad. Esto permite la creación de un sonido realista y único. Todos los presets están divididos en grupos, de modo que es fácil de encontrar presets requeridos, así como añadir fácilmente con el tema de sonido elegido. (Ibaiondo, 2014)

### ***Videojuegos para la relajación***

El mundo de los videojuegos no ha estado ajeno a los procesos terapéuticos, algunos son solo sencillos juegos flash en los cuales se interactúa solo para acceder a alguna biblioteca de efectos sonoros o visuales que ayudan en el proceso de la relajación, entre ellos:

- **Proteus (Anexos Figura 3):** no es un juego, más bien es una aplicación audiovisual. Un programa de relajación interactiva, que sumerge al usuario en una isla de colores y píxeles gordos sin objetivo concreto. En Proteus la persona se encuentra ubicada en una isla desierta. Esta isla se genera automáticamente en cada nueva partida y no existen dos islas iguales. Tampoco se puede hacer pausa ni grabar la partida. La variedad de elementos y sonidos extraños hacen que se despierte la curiosidad de moverse por el entorno. Necesariamente no tiene que terminarse el

juego para pasar un momento agradable solo con caminar por la isla un tiempo determinado cada día permite que se llegue a eliminar el estrés. (Kanaga, 2012)

## **1.4. Simulación de entornos virtuales**

La simulación a modo general es la capacidad de actuar o aparentar por un individuo o escenario una situación hasta cierto punto irreal en un momento dado. En el caso de los entornos virtuales será la cualidad que les permita brindar realismo o credibilidad a un escenario digitalmente creado.

Puede definirse a la simulación como la experimentación con un modelo que imita ciertos aspectos de la realidad. Esto permite trabajar en condiciones similares a las reales, pero con variables controladas y en un entorno que se asemeja al real, pero que está creado o acondicionado artificialmente. (Cabrera, 2012)

El objetivo es que la simulación permita comprobar el comportamiento de una persona, de un objeto o de un sistema en ciertos contextos que, si bien no son idénticos a los reales, ofrecen el mayor parecido posible. Así, es posible corregir fallos antes de que la experiencia, efectivamente, se concrete en el plano de lo real. (Hernández, 2014)

La simulación se puede considerar también, como una herramienta de análisis, que permite sacar conclusiones sin la necesidad de trabajar directamente con el sistema real que se está simulando. Esta es especialmente útil cuando no se dispone de dicho sistema real o resulta demasiado arriesgado realizar experimentos con él. (Hernández, 2014)

En muchas áreas de la ingeniería se utilizan los simuladores como una herramienta de trabajo más. Por ejemplo, en el diseño de nuevos fármacos se suelen utilizar modelos moleculares que sirven para simular por medio de computadoras la interacción de compuestos químicos. Los ingenieros de automóviles también utilizan modelos computarizados para analizar el impacto de los choques en la seguridad de los viajeros. (Hernández, 2014)

### **Tipos de simulación existentes**

Simulación, según la Real Academia española, se conceptualiza como la representación de algo, fingiendo o imitando lo que no es. Según el Handbook of Simulation (1998), es una imitación de las

operaciones de un sistema o proceso real a lo largo del tiempo, esto último en sistemas más complejos. Involucra la generación de una historia artificial del comportamiento del sistema y a partir de la historia se efectúan inferencias relativas a las características operacionales del sistema real que representa. También permite describir y analizar el sistema real que representa. (Cabrera, 2012)

La simulación puede llegar a convertirse en la metodología ideal para resolver muchos problemas reales, ayudando en la toma de decisiones, apoyando en el diseño de sistemas mucho antes de que estos sean construidos e incluso probando políticas de funcionamiento antes de que estas sean implantadas. Aunque esta no resuelve los problemas por sí misma, sí ayuda a identificar los problemas relevantes y a evaluar cuantitativamente las soluciones alternativas. Dentro de los tipos de simulaciones se pueden encontrar:

- **Simulación Predictiva:** en ésta, se interesan por los resultados absolutos finales, no por las comparaciones. Determinan promedios e intervalos de confianza de una corrida de simulación con valores específicos en las variables de decisión (varias corridas, mejores resultados).

Este tipo de simulación se puede utilizar para realizar pronósticos, por lo que es necesario contar con datos históricos de entrada confiables, se utiliza en procesos de decisiones que se repiten. Ejemplo: predecir el número de pacientes que necesitan trasplante de riñón. (Cabrera, 2012)

- **Simulación Comparativa:** En la simulación comparativa se determina cuando una opción es mejor que otra. Se debe especificar detalladamente qué significado tiene la palabra "mejor", para definir cuáles serán los datos de salida a comparar. Mejor significa disminuir el tiempo de espera o es un compromiso entre tiempo de servicio, largo de cola y costo por servidor. Se puede usar para tomar decisiones casuales o repetitivas, utilizar datos de entrada y salidas confiables.

Si los objetivos no son claros, se proveerá de un rango variado de resultados, que le permitan al usuario definir a posteriori, la importancia relativa de cada uno de ellos. Si los resultados o los datos de salida son claros, se puede usar técnicas de hipótesis estadística de los resultados. (Cabrera, 2012)

- **Simulación Investigativa:** La simulación investigativa indica factores que afectan el flujo de entidades en el sistema, pero no requiere de respuestas precisas, por lo que la calidad de los datos de entrada no es estricta.

- **Simulación visual interactiva:** La técnica de simulación visual interactiva es adecuada para apoyar la toma de decisiones. (Cabrera, 2012)
- **Simulación de caja negra:** Pensando al modelo como parte de un proceso de toma de decisiones, es conveniente, a veces, considerar el modelo como una caja negra, de donde salen flechas con datos, derivados directamente de los objetivos (y que difieren de un problema a otro) y a donde ingresan flechas con datos relacionados estrechamente con las hipótesis de trabajo del modelo. (Cabrera, 2012)

La solución planteada se encuentra enmarcada dentro de las simulaciones del tipo visual interactiva. Las mismas toman como elemento primordial la observación y la interacción con el entorno, rasgos que son fundamentales en dicha solución.

### **Características fundamentales de la simulación de entornos 3D**

Los entornos 3D pueden ser total o parcialmente ficticios, estos poseen características que pueden asemejarse a entornos reales, en los cuales intervienen diferentes factores ambientales. En los últimos años, los avances en hardware y software de ordenador han sentado las bases para el diseño de entornos virtuales (EV) con más calidad y fidelidad. Estas mejoras en la tecnología de los mismos han revivido el interés en el uso de los entornos virtuales para brindar conocimiento. Hay muchas ventajas en la utilización de los mundos virtuales, por ejemplo; ofrece un ambiente de formación flexible y rentable que se puede volver a configurar rápida y fácilmente para proporcionar la formación específica para la misión. También ofrece a los instructores la oportunidad de exponer a los estudiantes a situaciones que de otro modo serían imposibles (es decir, potencialmente mortal) para recrear en escenarios de formación de la vida real. Además, ofrece una oportunidad única para que los entrenadores puedan evaluar a sus alumnos, ya sea en tiempo real, mediante la congelación de la formación en los puntos críticos, o mediante la reproducción de todo el escenario de la formación sobre la terminación. (C. E. Lathan, 2002)

Los mundos virtuales tienen su origen en la simulación militar y en concreto en los simuladores de vuelo, donde el principal problema consiste en extraer de la base de datos visual (presumiblemente grande) el mundo visible en cada instante en función de la posición del observador o cámara virtual, en el escenario simulado. Tiempo después, con la comercialización de esta tecnología para uso civil surgió el concepto de lo que se conoce como Realidad Virtual que todavía hoy perdura y gana terreno en muchos ámbitos de la

sociedad, por ejemplo los gráficos 3D en entornos inmersivos que usan artefactos como guantes, cascos, etc., en busca de mayores grados de interacción con el ambiente virtual. (Aula, 2014)

La utilización de entornos virtuales comienza a desarrollarse en la enseñanza a distancia. Los entornos virtuales han sido utilizados como espacios de intercambio de información, asociados a una serie de recursos telemáticos ideados para el estudio a distancia y para facilitar la comunicación entre el profesor y su grupo de estudiantes.

El salto hacia entornos tridimensionales ha supuesto un gran avance y ha permitido el conocimiento de mundos pasados, presentes o incluso futuros; permite caminar por el interior de edificios que dejaron de existir hace siglos, interactuar con objetos de maneras imposibles en el mundo real, o estudiar objetos demasiado frágiles o de difícil acceso; ayudando a su vez a la conservación del mismo. Permite ensayar técnicas de restauración sobre modelos sintéticos, así como explorar diferentes teorías sobre su construcción, todo ello teniendo en cuenta no dañar el original.

La infografía 3D ha supuesto un importante perfeccionamiento en la reproducción de la realidad. La simulación es un proceso que permite estudiar un sistema físico sustituyéndolo por otro más fácilmente observable o medible. Mediante la simulación pueden abarcarse diversos intereses: un astronauta puede aprender a pilotar una lanzadera espacial, un gobierno puede prever la evolución de la población en su país, un meteorólogo puede predecir tormentas y un arqueólogo puede reconstruir una ciudad perdida a partir de sus restos. Estos enfoques requieren diferentes niveles de complejidad en la simulación: así, el astronauta necesitará sentir la dinámica del sistema para poder percibir aceleraciones, mientras que el arqueólogo puede conformarse con unos gráficos 3D que muestren su ciudad con un elevado nivel de realismo. (Calderon, 2004)

Los entornos virtuales tridimensionales entre sus características, poseen un espacio compartido donde participan muchos usuarios simultáneamente; cuentan con una interfaz gráfica donde se encuentran ambientes tridimensionales inmersivos; los caracterizan la inmediatez, la interactividad y la persistencia, permitiendo que la interacción tenga lugar en tiempo real y que los usuarios puedan crear, modificar y poseer contenidos virtuales, además de que el entorno virtual persiste aún en ausencia de sus creadores. También contribuyen a la formación de comunidades sociales donde el conocimiento se comparte. (Hernández, 2014)

**Entorno adaptable:** Se considera que es un entorno adaptable aquel que posee características configurables las cuales permiten el manejo del mismo en diferentes situaciones a través de la práctica. (Hernández, 2014)

## **1.5. Entornos virtuales para terapia de relajación**

Actualmente en la Universidad de las Ciencias Informáticas, específicamente en la Facultad 5 el Centro Vertex, tiene como prioridad la creación de aplicaciones de la naturaleza de los videojuegos, paseos virtuales, entornos virtuales, refrescadores de pantallas y más. Recientemente se dio a la tarea de crear dos entornos virtuales que apoyaran las terapias de relajación realizadas en centros nacionales, pero con perspectiva también a clientes extranjeros interesados en el tema.

Los entornos ya existentes son una playa y una pecera, ambos bien ambientados en la realidad de los lugares que simulan. Estos entornos fueron constituidos independientemente como módulos de una aplicación y por este motivo más allá de la simulación virtual solo permiten una configuración básica y trabajosa de las características de estos medios.

El centro además de estos módulos ya creados, tiene planteado el estudio para la futura creación de otros escenarios ambientados para la simulación de varios paisajes, que pudieran ser incluidos en la solución que permita la configuración y manejo de todos ellos.

### **Entorno de Playa. Características**

Esta simulación de un entorno de playa es altamente configurable a través de un archivo XML, que le permite al especialista encargado de las terapias de relajación, el control del entorno en tiempo real para que la aplicación se actualice en cuanto ocurra cualquier cambio en el mismo. Este archivo tiene variables definidas que permiten su modificación y realizan los cambios en la simulación. Inicialmente el sistema realiza los cambios de forma manual.

El entorno consta con dos garzas, las cuales son controladas por un script que les permite el movimiento por puntos definidos en el entorno. Al comenzar la simulación o activar el movimiento de una de ellas, esta calculará la distancia al punto más cercano y se moverá hasta él, al llegar decidirá que animación realizar, una vez concluido esto, recalculará la distancia hacia otro punto, repitiendo el ciclo hasta que se desactive la misma. Los puntos estarán dados por un grafo conexo donde por cada punto habrá una serie de caminos, posibilitando que el movimiento de la misma no sea de forma continua e igual en todo momento. Se usa este comportamiento después de un estudio realizado de cómo se mueven las garzas basado en documentales del reino animal.

Las gaviotas son controladas por un script que les permite el movimiento sobre un punto determinado. Inicialmente están dispersas por el terreno, donde realizan un estudio topográfico del mismo para no colisionar con ninguna montaña, a continuación calcularán la distancia hacia un punto dado y emprenderán el movimiento en línea recta hasta pasarse del mismo una distancia determinada, en ese momento lo rodearán y comenzarán el movimiento de nuevo pero esta vez a una altura y velocidad diferente. Después de revisar materiales, como documentales y de pasar horas mirando cómo se mueven las gaviotas reales, se decidió utilizar este tipo de comportamiento ya que se asemeja al movimiento de las mismas.

El paso del día es otra de las opciones que se puede configurar en el entorno. Basado en criterios de especialistas en terapias de relajación se definen tres configuraciones principales para lograr que el sistema funcione según los gustos de las personas a las que se les realizará la terapia, estas configuraciones vienen dadas por una opción de elegir un tiempo de por la mañana, otro de por la tarde y la combinación de ambos que forman el paso de un día, desde el amanecer hasta el ocaso.

Además el sistema consta con cinco cámaras fundamentales sugeridas por los especialistas en terapias, basados en que parte del entorno resulta más relajante para las personas y de qué forma utilizar el juego de cámaras para lograr el buen funcionamiento de la terapia.

Una primera cámara situada frente a la isla, debido a que varios especialistas comentan que en las terapias de relajación, en su fase inicial es importante que los pacientes se vayan relacionando lentamente con el entorno. Una segunda con una vista aérea de parte de la isla y el mar, ya que para muchas personas esta vista del océano resulta muy relajante. La tercera cámara seguirá a una de las garzas en su

andar por el entorno, ya que para muchas personas la tranquilidad y soledad con que se mueve la garza por el entorno resulta en extremo relajante. Una cuarta cámara dirigida a observar el movimiento de las gaviotas, ya que el vuelo de las mismas resulta de agrado y tranquilidad para muchas personas. Una quinta cámara permite el movimiento por toda la escena en caso de que los pacientes quieran interactuar directamente con el entorno.

### **Entorno de Pecera. Características (Anexos Figuras 9 y 10)**

Teniendo como objetivo la simulación de un entorno de pecera que sea adaptable a las terapias de relajación, este escenario consta de al menos dos peces y a lo sumo de dieciséis de estos logrando así la interactividad de más de uno de estos con sus similares. Es un entorno estático, lo cual quiere decir que todos sus elementos siempre se mantienen en la misma posición. Se opta por la mejor configuración de los elementos físicos tales como los efectos visuales, modelos tridimensionales y física del ambiente para permitir la interacción de los peces con el entorno. Esta simulación tiene funcionalidades para la configuración, tales como agregar o eliminar peces, realizar acercamientos de cámara a estos e incluso alimentarlos.

Los **peces** poseen direcciones, las cuales están asociadas a las componentes X, Y y Z de las coordenadas en el espacio tridimensional. La modificación de dichas direcciones es un elemento importante en la validación de la evasión de obstáculos, así como también en la implementación de otras funcionalidades, ya que sirven para desviar al pez en caso de encontrar objetos u otros peces en su camino. El movimiento de los peces se realiza teniendo en cuenta el espacio en el cual se van a desplazar. Los objetivos se definen de forma aleatoria en dicho espacio. Una vez alcanzado un objetivo, este se vuelve a recalcular de forma aleatoria y dentro de un volumen definido, generando otro punto, el cual será el nuevo objetivo. (Hernández, 2014)

El **alimento** es uno de los efectos que están presentes en la simulación, dado que aporta realismo a la misma y logra un flujo natural que no debe estar ausente en este tipo de ambiente. Unos segundos después que la comida es arrojada en la escena, uno de los actores tomado de forma aleatoria, el cual se dirige en ese momento a un punto, desvía su curso para buscarla, cuando la alcanza, ésta desaparece, simulando el efecto de que el actor comió. Luego de esto continua la generación de puntos aleatorios. (Hernández, 2014)

El **enfoco y seguimiento** de los actores de forma individual, es logrado a partir de lo que se conoce como enfoque y paneo de cámara, el cual se encarga de seleccionar de forma aleatoria, en una primera etapa, el actor el cual será seguido por la cámara para luego realizar el enfoque y el **paneo** en éste.

El **paneo** es toda toma de video hecha a partir del desplazamiento de la cámara desde un punto fijo llamado eje en dirección rotativa sobre este, de manera circular lenta o parcialmente circular de la escena.

## **1.6. Metodología XP para el desarrollo de Software**

Como parte de la solución a la problemática existente y luego de investigar acerca de las soluciones similares es necesaria la selección de un conjunto de herramientas y tecnologías que se utilizarán para confeccionar dicha solución.

Sin lugar a dudas, las metodologías ágiles, son un tema de especial interés a la hora de hablar del desarrollo de un software y la ingeniería del software. Estas aportan al equipo de trabajo una mayor flexibilidad a la hora de enfrentarse a los retos que puedan presentarse durante el proceso de desarrollo, permitiendo mantener la calidad como uno de los aspectos del producto.

Basado en sus criterios y experiencias en el desarrollo de software, el centro VERTEX, cliente que solicita este trabajo como solución a su problemática, orientó que se utilizara la metodología XP en el proceso para arribar a dicha solución. Exponiendo su satisfacción con los resultados en procesos anteriores y la facilidad que les brindó el uso de sus herramientas.

La metodología XP (Extreme Programming) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (Tellez, 2012)

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las prácticas. La mayoría de las prácticas propuestas por XP no son novedosas sino que en alguna forma ya habían sido propuestas en ingeniería del software e incluso demostrado su valor en la práctica. El mérito de XP es integrarlas de una forma efectiva y complementarlas con otras ideas desde la perspectiva del negocio, los valores humanos y el trabajo en equipo. (Penadés, 2009)

## 1.7. Herramientas

El uso de herramientas para el tratamiento de imágenes y ajustes de sonido, son realmente útiles a la hora de confeccionar una aplicación que abarque varios tipos de componentes, como la referida en este trabajo. Estas son capaces de proporcionar funcionalidades de edición, con las que pudiera no contar la plataforma en la que se desarrolla un programa dado, favoreciendo que el resultado final cuente con la calidad requerida.

### *Tratamiento de imágenes*

Existen numerosas aplicaciones dedicadas al trabajo con imágenes. Algunas de estas son más abarcadoras que otras en lo que se refiera a la cantidad de acciones sobre el archivo a modificar. Esta diferencia entre las aplicaciones también está marcado por la finalidad del uso de la misma, es decir, si es solo para la corrección de la imagen, el montaje y fusión con otras imágenes, o ambos. También pueden ser diferenciadas por la plataforma sobre la cual trabajan o si es software privativo o gratuito.

- **Pixlr**

Pixlr es un editor de imágenes en línea muy básico y simple que le permite editar, ajustar, filtrar y manipular imágenes en línea. Eso significa que puedes cargar sus imágenes y editarlas en su navegador. Pixlr se construye para los no profesionales, los usuarios que tienen necesidades básicas de edición. (creativo, 2014)

- **Gimp**

GIMP es un editor de imágenes de código abierto gratuito para Mac OS X y a menudo es considerado el "Photoshop gratuito" por su interfaz y funciones muy similares a Photoshop. Es un software en fase beta desarrollado por voluntarios, así que sus actualizaciones y estabilidad pueden ser un inconveniente. Pero muchos usuarios reflejan que este software no viene con ningún gran problema y que disfrutan el utilizarlo. Por favor nota que GIMP es compatible con Mac OS X o posteriores. (Cervera, 2013)

- **Adobe Photoshop CS6**

El programa Adobe Photoshop CS6 proporciona aún más potencia de edición de imágenes, nuevas opciones creativas y el motor Adobe Mercury Graphics Engine para lograr un rendimiento increíblemente rápido. Con nuevas funciones basadas en el contenido, crea diseños y películas fascinantes mediante los nuevos flujos de trabajo y herramientas rediseñadas.

Siguiendo la línea de desarrollo del centro Vertex, se optó por utilizar Photoshop como herramienta para el tratamiento de las imágenes necesarias en la creación de la plataforma VirtualRelax. Además de las ventajas ofrecidas por las innumerables funcionalidades de dicha herramienta.

### ***Edición de Sonido***

El número de aplicaciones dedicadas a la edición de sonido, al igual que con las herramientas para tratar imágenes, es bastante amplio. Entre estas aplicaciones existen algunas diferencias atendiendo a formatos de sonido permitidos, la plataforma para la que fueron diseñados y las posibilidades de edición de los archivos.

- **Sound Editor**

Sound Editor te permite grabar, reproducir y editar archivos Mp3, WMA o WAV de forma sencilla gracias a las acciones Copiar, cortar y pegar. Podrás grabar audio desde tu micrófono o entrada de línea. (Windows, 2014)

- **Adobe Audition CS6**

La aplicación Adobe Audition CS6 proporciona las herramientas profesionales que necesita para dotar del mejor sonido a sus producciones de video y audio. Gestiona de forma competente una amplia gama de

tareas de producción de audio, incluidas la grabación, mezcla y restauración de sonido. Puede gestionar la mayoría de los formatos de sonido, así como trabajar directo sobre el audio en archivos de video.

La superioridad de Audition sobre otras herramientas de edición de sonido, es notable. Manteniendo la línea de desarrollo de Vertex y haciendo uso de esta superioridad, se propone su uso para editar los archivos de audio que puedan estar presentes en la plataforma.

## **1.8. Motor gráfico**

Un motor gráfico es un conjunto de programas que enlazados forman una rutina, por ello a veces solo dicen que es una rutina de programas, que permiten la creación de un videojuego. En un principio un programador no requería de grandes herramientas para realizar un videojuego, como ocurre con el famoso juego de tenis que se jugaba en una pantalla monocromática o en un televisor. Sin embargo al aumentar el poder de los procesadores gráficos se fueron haciendo necesarias nuevas herramientas, no solo para crear los videojuegos si no para poder correrlos. (Tejeda, 2014)

Es aquí donde empiezan los motores gráficos. El término parece que nace de la comparación con un motor de automóvil, se dice que la carrocería es todo lo que se ve en el videojuego mientras que el motor es lo que hace funcionar todo. Hace casi unos veinte años (en la década de los 90) el mercado de los videojuegos entra en el mundo de tres dimensiones (el famoso 3D) y con ellos nacen los motores de videojuegos o motores gráficos. Al parecer fue la compañía Origin System la que generó el primero de ellos para el juego Ultima Underworld (al motor se le conoce como el motor de Ultima Underworld y se hizo costumbre darles nombre en base al primer juego que produjeron), luego está el mítico Doom. (Tejeda, 2014)

Ambos lograban los efectos en 3D utilizando figuras “planas” que se colocaban virtualmente un poco lejos unas de otras, dando la sensación de profundidad. En el caso del primero, se utilizaban texturas que daban la sensación de profundidad, mientras que en el segundo se usaban planos superpuestos para lograr tal efecto. Voxel fue un emulador de vuelos logrado con un motor más complejo que los anteriores. Se han citado los primeros tres pero para que puedan darse una idea hay un desarrollo de más de tres años entre el primero y el segundo. Hoy día la mayoría de los motores utilizan una plataforma o interface que se basa en rutinas preestablecidas, las más populares son Direct3D para Windows, Glide API y OpenGL tanto para Linux, Windows y Mac. Tanto Glide como OpenGL son de uso libre y Direct3D es

propiedad de Microsoft. Estas plataformas permiten el renderizado de las figuras. El término viene de "render" en inglés y se puede decir que es "interpretar" o "representar", aunque no tiene en español un equivalente representativo. (Hernández, 2014)

Lo que se hace con este renderizado, es hacer que el computador forme una figura plana (2D) a partir de una "real" (3D), con esto se ahorra información y la calidad de la imagen es muy alta. Por cierto también se les denomina motores de renderizado y utilizan vectores de posición, formando pequeños triángulos que dan la forma. Estos triángulos son calculados por medio de complejas funciones matemáticas que la computadora realiza. Al renderizar se "suaviza" la forma y la textura parece plana y no formada por los triángulos. (Hernández, 2014)

Esto añadido a los motores gráficos genera efectos realmente espectaculares como los que utilizaron para crear la película "Toy Story", la primera generada totalmente con motores gráficos de última generación, como pueden ver no solo en juegos se aplican los motores gráficos. Esto lleva a un problema que enfrentan todos los desarrolladores de motores gráficos. Lleva mucho tiempo crearlos para luego generar el juego, entonces ocurre que cuando el producto (el juego) está terminado pues ya está obsoleto, de allí que exista una competencia tan feroz en este mercado. Para solventar esto muchos desarrolladores "liberan" los códigos de programación (los famosos códigos fuentes) que permiten generar los motores gráficos para que otros los mejoren y así puedan competir más lealmente. (Anónimo, 2014)

Como se plantea anteriormente los motores gráficos no solo se usan en la creación de videojuegos, tienen muchas más aplicaciones en el campo de la realidad virtual, una de ellas es la de las simulaciones, por lo que para escoger un motor gráfico para este trabajo se realizó un estudio de los motores más usados y potentes en la actualidad, después de esto en conjunto con el cliente se seleccionó Unity 3D por presentar un grupo de características necesarias para el desarrollo de la aplicación. A continuación se relacionan algunas de las principales características que se tomaron en cuenta para la selección de este motor.

- Plataformas para las que desarrolla: Android, iOS, Mac, Windows, Web, XBOX LA, Wii, PS Network
- Licencias (Dólares): Indie- Gratuita, Unity Pro- \$1500, Android Básica- \$400, iOS Básica- \$400, Android Pro- \$1500 (requiere Unity Pro), iOS Pro- \$1500 (requiere Unity Pro)
- Lenguaje de programación que soporta (Scripting): C#, Javascript, Boo

- Extensiones o plugins: C#, C++
- Importación de Assets: Modelos 3D animados (formatos nativos de Blender, 3DStudio, MaxMaya, Cinema 4D), Texturas (PNG, JPG, TGA, BMP), Sonidos (OGG, MP3, WAV), Videos (AVI, MPG)

Unity 3D es un motor gráfico que hace el proceso de producción simple, presenta una interfaz de producción amigable y sencilla, brindando un set1 de pasos lógicos para construir casi cualquier panorama concebible de juego o simulación. Establece el uso del concepto Game Object (GO), donde se puede estudiar los componentes del juego en objetos dóciles, que está compuesto por muchos componentes y scripts para su funcionamiento en la escena. (Ecured, 2014)

#### **Dentro de sus herramientas se encuentran:**

- **Assets:** Son los bloques constructivos de todo lo que el Unity posee en sus proyectos. Se guardan en forma de archivos de imagen, modelos del 3D y archivos de sonido, el Unity se refiere a los archivos que se usarán para crear su juego como activos. (Ecured, 2014)
- **Game Objects:** Cuando un activo es usado en una escena de juego, se convierte en un "Game Object". Todo Game Objects contiene al menos un componente con el que comenzar, es decir, el componente Transform. Transformación simple la cual le dice al motor de Unity la posición, rotación, y la escala de un objeto. (Ecured, 2014)
- **Components:** Los componentes vienen en formas diversas. Pueden ser para crear comportamiento, definiendo apariencia, e influenciando otros aspectos de la función de un objeto en el juego. Los componentes comunes de producción de juego, vienen contruidos dentro del Unity, desde el Rigidbody, hasta elementos más simples, como luces, las cámaras, los emisores de partículas, y más. (Ecured, 2014)
- **Scripts:** El Scripting es una parte esencial de Unity, ya que define el comportamiento del juego. El Scripting es la forma en la que el usuario define el comportamiento del juego (o las normas) en Unity. El lenguaje de programación recomendado para Unity es JavaScript, aunque C# o Boo pueden ser igualmente usados. En Mac, es llamado como Unitron, y en PC, Uniscite. (Ecured, 2014)

- **Prefabs:** Almacena los objetos como activos para ser reusado en partes diferentes del juego, y luego creados o copiados en cualquier momento. (Ecured, 2014)

## Conclusiones del Capítulo

En este capítulo se realizó un estudio de los medios existentes que hacen referencia a la situación problemática planteada al comienzo, así como herramientas que pudieran facilitar el desarrollo de una solución para dicha problemática:

- El estrés se puede definir como una enfermedad, que es la respuesta de un organismo sometido a constantes estados de tensión física o mental. La forma más común de tratar a pacientes con estrés es a través de las terapias de relajación asistidas por profesionales de la rama.
- Existe una gran variedad de aplicaciones con características que las diferencian entre sí, pero que mantiene en común el objetivo de apoyar a los profesionales que aplican terapias de relajación para tratar a pacientes con estrés.
- Los software de simulación de entornos naturales, son aplicaciones prácticas para apoyar a los profesionales en el tratamiento de pacientes con estrés.
- El uso de plataformas de software, permite la integración de aplicaciones con similitudes entre sí.

Producto de la reciente investigación, se detectó que:

- En el caso del Entorno de Pecera, solamente se tiene la versión ejecutable y no el código del mismo, trayendo esto consigo la imposibilidad de acceder a las variables utilizadas en su implementación, la cual además presenta errores en condiciones de parada y demás métodos que ejecuta la aplicación, tales como la funcionalidad agregar pez que se produce a muy alta velocidad y sin detenerse, provocando la caída del sistema en ejecución.
- Se mantiene presente el uso del Entorno de Pecera en el documento, promoviendo la futura reimplementación o modificación del mismo en futuras iteraciones de la propuesta de solución de este trabajo.

- En el caso del Entorno de Playa, presenta situaciones que imposibilitan un resultado óptimo de la aplicación, como pudiera ser una variable de control para lograr la ejecución en tres tiempos prevista para mejorar su uso en las terapias de relajación.
- La posibilidad de agregar la variable de control que permita lograr la ejecución en tres tiempos resulta muy difícil debido a las limitaciones que suponen modificar los archivos originales del entorno.

## **CAPÍTULO 2. PROPUESTA DE SOLUCIÓN**

En este capítulo se expone la solución al problema de investigación presentado anteriormente. La solución se plantea detalladamente haciendo una descripción sobre la estructura, detalles de implementación y funcionamiento de la aplicación, basada en la solución propuesta a continuación.

### **2.1. Descripción de la plataforma propuesta**

Como objetivo principal, este trabajo tiene la tarea de crear una plataforma con una interfaz gráfica amigable y de fácil uso, que permita el control y modificación en tiempo real de los módulos de simulación virtual creados por el centro Vertex, que son el Entorno de Playa y el Entorno de Pecera. El desarrollo de esta plataforma fue definido, producto a la investigación previa realizada por el centro Vertex, en coordinación con especialistas en las terapias de relajación y futuros usuarios u operadores de la misma, ya que los entornos previamente desarrollados solo permiten configuraciones básicas y para realizarlas se debe detener la simulación alterando los resultados y la validez de la terapia.

En una primera etapa de desarrollo, la plataforma debe ser capaz de permitir la interacción en tiempo real de la información ofrecida por los entornos virtuales, como los actores que se relacionan y el entorno en cuestión, además de las funcionalidades que asignan comportamiento a los actores, para estos dos módulos creados inicialmente. La figura 11 de los Anexos, es uno de los objetos creados con la herramienta Photoshop, una imagen que muestra las ventanas y paneles de configuración de la solución, organizada de manera teórica.

Para una segunda etapa de desarrollo debería poder asimilar este nivel de manejo o configuración no solo para estos dos módulos, sino también para la futura creación de otros módulos con escenarios similares y con el mismo objetivo.

### **2.2. Componentes a usar en la Implementación**

El motor gráfico Unity 3D posee una serie de componentes que permiten un uso hábil y sencillo de la herramienta. Los cuales serán utilizados en su mayoría en el desarrollo de la aplicación, integrando además los objetos o elementos creados con otros programas, como en este caso Adobe Audition y Adobe Photoshop.

### ***Audio Listener***

El Audio Listener (Audio Oyente) actúa como un dispositivo de micrófono. Este recibe la entrada desde cualquier Audio Source (Fuente de Audio) en la escena y reproduce los sonidos a través de los altavoces del ordenador. Para la mayoría de las aplicaciones adquiere mayor sentido adjuntar este oyente a la cámara principal. Si el oyente de audio está dentro de los confines de la Reverb Zone (Zona de Reverberación), la reverberación es aplicada a todos los sonidos audibles en la escena. Además, los Audio Effects (Efectos de Audio) pueden ser aplicados al oyente y esto lo aplicará a todos los sonidos audibles en la escena.

### ***Audio Source***

El Audio Source (Fuente de Audio) reproduce un clip de audio en la escena. Si este audio es un clip en 3D, la fuente será reproducida detrás de la posición dada y se irá atenuando con el aumento de la distancia. El audio puede ser esparcido afuera por medio de los altavoces (Stereo 7.1) y modificado entre 2D y 3D. Esto puede ser controlado a distancia a través de las curvas de disminución. También, si el oyente está dentro de una o múltiples zonas de reverberación, la reverberación es aplicada a la fuente. Los filtros individuales pueden ser aplicados a cada uno de las fuentes de audio para una experiencia auditiva más enriquecedora.

### ***Cámara***

Exactamente como las cámaras son usadas en las películas para mostrar la historia a la audiencia, las cámaras en Unity son usadas para mostrarle al usuario el entorno. Siempre se tendrá al menos una cámara en la escena, pero se pueden tener más de una. El uso de múltiples cámaras, brinda la posibilidad de una pantalla dividida para varios usuarios o la creación de avanzados efectos personalizados. Las cámaras pueden ser animadas o ser controladas a través de la física. Prácticamente todo lo que se pueda imaginar es posible con el uso de las cámaras, y se puede usar una típica o única cámara para acomodar el estilo del escenario.

### **Componentes de Interfaz de Usuario**

Estos son los componentes que realmente permitirán al usuario interactuar con la aplicación o visualizar lo que sucede en ella. Cada uno de los componentes será configurado a medida que avance el desarrollo de

la aplicación, para que una vez que se muestre al usuario, este pueda visualizar y trabajar con ella de manera sencilla.

### ***Canvas***

Los componentes de Canvas (Lienzo) representan el espacio abstracto en el cual la UI (user interface o interfaz de usuario) es colocada y renderizada. Todos los elementos de la UI deben ser hijos o heredar del Objeto de Juego (GameObject) que tiene adjuntado un componente de Canvas.

### ***Button***

El control de tipo Button (Botón) responde al clic producido por el usuario para iniciar o confirmar una acción. Ejemplos similares incluyen a los botones Aceptar y Cancelar usados en los formularios web.

### ***Image***

El control de Image (Imagen) muestra una imagen no interactiva para el usuario. Este puede ser usado para la decoración, los iconos, etc... y además la imagen puede ser cambiada desde un Script (Código) para reflejar los cambios en otros controles. Este control es similar al control Raw Image (Imagen Cruda), pero ofrece más opciones para la animación de imágenes y archivando exactamente el control de rectángulo.

### ***Text***

El control Text (Texto) muestra una pieza no interactiva de texto al usuario. Esto puede ser utilizado para proveer encabezados o etiquetas para otros controles de tipo UI o mostrar instrucciones u otros textos.

### ***Script***

La escritura de código es un ingrediente esencial en todas las aplicaciones. Incluso las más sencillas, necesitan del código para responder a la información que el usuario introduce y arreglar que los eventos durante la ejecución sucedan cuando sea el momento. Más allá de esto, el código puede ser utilizado para crear efectos gráficos, control físico del comportamiento de los objetos o cualquier implemento del sistema personalizado de inteligencia artificial para los personajes en el escenario.

## 2.3. Requisitos no Funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Los requisitos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto. (Pressman, 2008)

Este trabajo pretende construir una aplicación con fuertes características informáticas, así como terapéuticas, que posea una interfaz de configuración de la misma, que resulte sencilla para cualquiera de los usuarios finales y con una calidad gráfica aceptable logrando un alto grado de realismo en la terapia.

Estos son algunos de los principales requisitos no funcionales a tener en cuenta a la hora de confeccionar la aplicación:

- **Usabilidad:** El producto final será usado por los especialistas que aplican terapias de relajación. Estos podrán realizar configuraciones en el programa, para mejorar su uso en el tratamiento de personas que padecen altos niveles de estrés u otras dolencias de esa índole. Por lo tanto, su interfaz debe ser sencilla y entendible para usuarios que no posean muchas habilidades en el uso de ordenadores.
- **Rendimiento:** La aplicación debe lograr valores en la frecuencia de visualización superiores o iguales a 30 cuadros por segundo (FPS por sus siglas en inglés). Esta es la frecuencia mínima en la que se deben visualizar este tipo de aplicaciones, para obtener una calidad aceptable.
- **Hardware del cliente:** Es necesario para el funcionamiento del sistema mínimo un microprocesador Intel Pentium IV a 2,6 GHz, 512 MB RAM, 32 bits de profundidad de color y 128 MB de memoria de video integrado o 64 MB de video dedicado, compatible con Open GL o DirectX 3D. Estos requisitos de hardware fueron obtenidos, luego de la ejecución en ordenadores con estas características, de los entornos virtuales existentes. Las pruebas se realizaron con estos entornos porque son los de mayor consumo de recursos de esta clase.
- **Software:** El sistema es compatible con Unity3D sobre Windows. Debe tener instalado las bibliotecas del lenguaje C++ (VisualC++ Redistributable 2005 o superior) y DirectX 9.0 o superior.

La instalación de las bibliotecas mencionadas, es necesaria ya que estas poseen componentes de software que posibilitan la ejecución de la herramienta (Unity 3D) en el sistema operativo (Windows).

## **2.4. Fase de Exploración**

La metodología XP se enfoca principalmente en la satisfacción del cliente, ya que trata de ofrecer al usuario final el programa que él necesita y en el momento que lo requiere. También permite aprovechar al máximo las ventajas del trabajo en grupo, porque crea un elevado nivel de colaboración y comunicación, lo que hace que tanto los jefes de proyecto, los clientes y desarrolladores, sean parte del equipo y protagonistas en el desarrollo de la aplicación.

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. (Penadés, 2009)

### ***Historias de Usuario***

Las Historias de Usuarios (HU) solamente proporcionarán detalles sobre la estimación del riesgo y cuánto tiempo será empleado en su implementación. Es el cliente el encargado de asignarle una prioridad a cada HU y es el equipo de desarrollo el encargado de asignarle un costo, este se traduce en las semanas que llevará el desarrollo de las mismas. Si las historias según lo planificado demoran en desarrollarse se sugiere dividirla en historias más pequeñas. También, es importante destacar, que las HU nuevas pueden describirse en cualquier momento, con esto se comprueba la flexibilidad de la metodología.

Las HU serán representadas mediante tablas divididas por las siguientes secciones:

- Código: Código que identifica la historia Sería HU- Número Consecutivo.
- Nombre: Nombre que identifica la historia de usuario.
- Referencia: Se hace referencia a los requisitos que tiene en cuenta la HU.
- Prioridad: Prioridad puede ser Alta, Media o Baja (Según Cliente).
- Iteración Asignada: Que iteración se desarrollará (Según su importancia).
- Puntos Estimados: Tiempo en semanas que se le asignará. (Estimado).

- Descripción: Breve descripción del proceso que define la historia.
- Observaciones: Alguna acotación importante de señalar acerca de la historia.

**Tabla. 1: HU Configurar y modificar XML.**

| Historia de Usuario  |   |
|--|---|
| <b>Número:</b> 6   | <b>Nombre:</b> Configurar y modificar XML |
| <b>Usuario:</b> Autor  |   |
| <b>Modificación de Historia Número:</b>  | <b>Iteración Asignada:</b> 1              |
| <b>Prioridad en Negocio:</b> Alta<br>(Alta / Media / Baja)   | <b>Puntos Estimados:</b> 2                |
| <b>Riesgo en Desarrollo:</b> Alta<br>(Alta / Media / Baja)   | <b>Puntos Reales:</b> 2                   |
| <b>Descripción:</b> Se mide la capacidad de acceso, trabajo y configuración del archivo xml y todas sus variables, del Entorno de Playa, desde la plataforma garantizando la modificación de todos los actores que intervienen en la escena en cada ejecución de la misma. |   |
| <b>Observaciones:</b> Las modificaciones realizadas desde la plataforma al archivo de configuración son guardadas y aceptadas por el entorno cuando se ejecuta manualmente.  |   |

**Tabla. 2: HU Cargar entorno.**

| Historia de Usuario                     |                               |
|---|-------------------------------|
| <b>Número:</b> 7                        | <b>Nombre:</b> Cargar entorno |
| <b>Usuario:</b> Autor                   |                               |
| <b>Modificación de Historia Número:</b> | <b>Iteración Asignada:</b> 1  |

|  |                            |
|--|----------------------------|
| <b>Prioridad en Negocio:</b> Alta<br>(Alta / Media / Baja)   | <b>Puntos Estimados:</b> 2 |
| <b>Riesgo en Desarrollo:</b> Alta<br>(Alta / Media / Baja)   | <b>Puntos Reales:</b> 1    |
| <b>Descripción:</b> Probar que la plataforma sea capaz de cargar y ejecutar el Entorno de Playa aplicando las modificaciones al archivo xml de configuración del mismo.  |                            |
| <b>Observaciones:</b> La plataforma es capaz de cargar el entorno y modificar sus parámetros, solo con el inconveniente de haber tenido que realizar una interfaz de configuración específica para sus variables y no de manera general o común para más de un módulo. |                            |

Las restantes HU se muestran en las Tablas 5, 6, 7, 8 y 9 de los Anexos.

## 2.5. Planificación y Entrega

En esta fase el programador establece la prioridad de cada historia de usuario y correspondientemente realiza una estimación del esfuerzo necesario para cada una de ellas. Se determina un cronograma en conjunto con el cliente a partir de los acuerdos tomados en las reuniones. El resultado de esta fase es el plan de Entregas.

### ***Plan de Entregas***

El plan de entregas se elabora con el objetivo de que los programadores obtengan una estimación de las historias de usuario en cuanto al nivel de detalle, o sea, para fijar el período de tiempo que se puede tardar en la implementación de cada una. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura.

Los elementos que se toman en cuenta para la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable. (Penadés, 2009)

Antes de la culminación del producto final, se realizan varias versiones en un rango de fechas calculadas o planificadas como resultado del Plan de Entregas (Anexos Tablas 8 y 9).

## **2.6. Diseño del Sistema**

El diseño crea una estructura para la organización de la lógica del sistema y permite que sea escalable con cambios en un solo lugar. Los diseños se crean con el mayor grado de sencillez posible, dividiéndolos en varias partes que pudieran tornarse complejas.

En la metodología XP no es necesaria la descripción del sistema a través de diagramas de clase utilizando notación UML, sino que se guía por técnicas como las tarjetas CRC (Contenido, Responsabilidad y Colaboración).

También pueden ser utilizados otros recursos como los diagramas para obtener una mejor visión y comunicación entre el equipo de trabajo, siempre y cuando genere información importante.

### ***Arquitectura del Sistema***

Los estilos arquitectónicos favorecen un tratamiento estructural que concierne más bien a la teoría, la investigación académica y la arquitectura en el nivel de abstracción más elevado. Mientras que los patrones se ocupan de cuestiones más cercanas al diseño, la práctica, la implementación, el proceso, el refinamiento y el código. La unión de estos estilos con los patrones, ya sean arquitectónicos o de diseño, complementan la arquitectura de un sistema.

La plataforma se desarrolla sobre el motor gráfico de Unity 3D y el cual posee un estilo de arquitectura en 3 capas, definiendo cómo organizar el modelo de diseño en capas simplificando la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. La ventaja ofrecida por este estilo arquitectónico es que en el caso de que exista algún error o la necesidad de algún cambio, solo se hacen los cambios en el nivel afectado, sin perturbar el correcto funcionamiento del resto del sistema. Las capas definidas se describen a continuación y se muestran en la figura 9 de los Anexos:

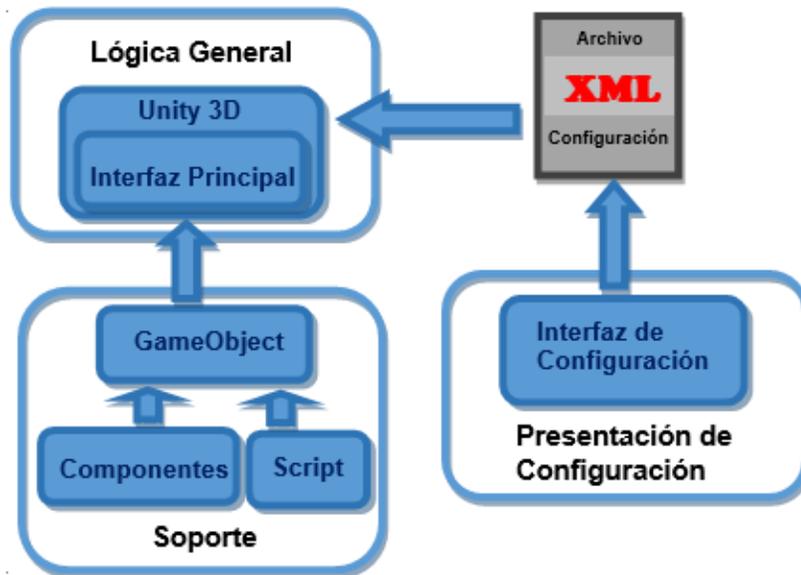


Fig. 1: Arquitectura de tres capas.

- Capa de Presentación de configuración: Esta capa está formada por la interfaz de configuración del sistema.
- Capa de Lógica general: Esta capa la forma la interfaz principal de Unity 3D, que es donde están todos los componentes con los que interactúa el usuario para conformar el escenario. Dentro de estos componentes están la vista de juego, el editor de escenario, el inspector, entre otros.
- Capa de Soporte: Está formada por las entidades, que representan objetos que van a ser manejados o utilizados por toda la aplicación y por las librerías para la física, los gráficos, el sonido, y el almacenamiento de los datos brindados por el motor gráfico de Unity 3D.

### **Patrones de Diseño**

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de aplicaciones y otros ámbitos referentes al diseño de interacción o interfaces. Entre ellos se encuentran los Patrones Generales de Software para Asignar Responsabilidades llamados comúnmente GRASP, el acrónimo de General Responsibility Assignment Software Patterns. Los patrones GRASP son

aquellos que describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones.

Los Patrones que describe GRASP son Experto, Creador, Alta Cohesión, Bajo Acoplamiento y Controlador. Debido a la arquitectura en capas y las herramientas utilizadas en este trabajo se recurre a todos estos patrones, haciendo mayor énfasis en el uso del patrón Experto, puesto asigna la responsabilidad a la clase WriteXML de acceder al archivo xml, que contiene las variables de configuración de los entornos virtuales y el patrón Creador, en la clase LoadLevel\_OnClick permitiéndole crear instancias de otras clases para obtener información de estas. Pero también se utilizan los restantes creando una arquitectura robusta. Estos patrones se muestran en la Tabla 10 de los Anexos.

### ***Tarjetas CRC del Sistema***

La utilización de tarjetas CRC (Clase-Responsabilidad-Colaboración), como las utilizadas para este trabajo que se muestran en los Anexos en las tablas 13 y 14, permite realizar un inventario de las clases que se necesitan para implementar el sistema y la forma en que van a interactuar, de esta forma se pretende facilitar el análisis y discusión de las mismas por parte de varios actores del equipo de proyecto, con el objetivo de que el diseño sea lo más simple posible verificando las especificaciones del sistema.

## **2.7. Plan de Iteraciones**

En la fase actual se realizan varias iteraciones sobre el sistema. En cada una de estas iteraciones se van implementando las historias de usuario seleccionadas. Al comenzar cada iteración tiene lugar una revisión del plan de iteraciones y de ser necesario se le realizan modificaciones. Posteriormente se procede a la descomposición de las historias de usuario en tareas de desarrollo, que son asignadas a una o varias personas responsables de su implementación. Estas tareas son definidas estrictamente, por lo que pueden ser descritas en lenguaje técnico.

Seguidamente se describen las tareas de desarrollo realizadas en cada una de las iteraciones.

### ***Iteración 1***

Esta primera iteración tiene como objetivo darles cumplimiento a las historias de usuario 1 y 2. Con la finalidad de dar cumplimiento a las mismas se asignaron dos tareas de investigación. Como se muestra en las Tablas 15 y 16 de los Anexos.

### ***Iteración 2***

Esta segunda iteración tiene como objetivo el cumplimiento de las historias de usuario 3,4 y 5. Para esto se le asignaron al programador tres tareas de investigación. Como se muestra en las Tablas 17, 18 y 19 de los Anexos.

### ***Iteración 3***

El objetivo de esta tercera iteración es darles cumplimiento a las historias de usuario 6 y 7. Para su realización se le asignaron tres tareas de investigación al programador. Como se muestra en las Tablas 20, 21 y 22 de los Anexos.

## **2.8. Bases Estructurales para los Futuros Módulos**

Producto a que los módulos de entornos Playa y Pecera fueron creados independientemente, existen limitaciones que reducen las funcionalidades de la aplicación para la que se realizó este trabajo. Aunque estos fueron creados con la finalidad de ser incluidos a un programa, no estaban sentadas las bases o características que estos debían de tener para satisfacer y asimilar las configuraciones que les serian realizadas a través de la aplicación base.

A raíz de esta situación, se decidió proponer una estructura sobre la cual estos módulos deberían ser construidos. Obteniendo de esta manera una mayor similitud en su composición y garantizando que puedan incluirse sin contratiempo en el programa controlador, asumiendo las configuraciones aplicadas por este.

### ***Variables del Entorno para Controlar***

Todos los entornos de simulación dedicados a las terapias de relajación, están constituidos por diferentes actores, ambientes y cámaras que juntos componen la escena. Cada uno de ellos representa un atributo configurable del medio y permiten lograr los efectos deseados. En estos entornos la ausencia u omisión

de uno o más de estos elementos, no determina el nivel de visualización para la relajación, como por ejemplo en el refrescador de pantalla Dream Aquarium, ha tenido casos de relajación aun sin la interacción de los peces, solo con el entorno.

Para la creación de los futuros módulos se trabajará de la siguiente forma, atendiendo a las variables públicas de configuración y métodos en las que intervienen las mismas.

En el caso de las variables públicas referentes a los actores, serían:

- **Actor:** Se debe controlar la exposición de estos en la escena a través de un enumerador (enum) que contendrá la cantidad de dicho elemento en el entorno, cada actor presente debe tener su propio enumerador, distinguiéndose entre ellos por la numeración desde 1 hasta N, es decir *Actor1, Actor2, ..., ActorN*. Para el caso del mayor número de actores de este tipo en la escena estaría controlada por el parámetro *Muchos*, para una aparición parcial se utilizará *Pocos* y para la ausencia total de este elemento se establecerá *Ninguno*. Quedando de la siguiente manera *public enum Actor1 (Muchos, Pocos, Ninguno)*.
- **Camara:** Las cámaras que participan en la simulación pueden ser de varios tipos dependiendo de su enfoque, dígame perspectiva, paneo o seguimiento. Esta variable será controlada a través de un enumerador que contenga estos tipos de enfoque. Se utilizará un máximo de dos perspectivas en la escena controladas por los atributos *Perspectiva1* y *Perspectiva2*. Para la variable de control de seguimiento de algún actor en el sistema se utilizará *Seguimiento* y para el paneo de alguna cámara seleccionada previamente por el desarrollador del módulo será *Paneo*. Quedando estructurado el enumerador de la manera siguiente *public enum Camara (Perspectiva1, Perspectiva2, Seguimiento, Paneo)*.
- **Atmosfera:** En todos los entornos en los cuales pudiese estar presente el control atmosférico, este evento se controlará por tres enumeradores que contengan las posibles modificaciones de ambiente. Uno hará referencia a la iluminación, *Atmosfera1*, controlando los valores *Oscuro, Tenue* o *Intenso*. Los fenómenos meteorológicos serán controlados por el segundo enumerador, *Atmosfera2*, a través de los valores *Nublado, Neblina* o *Lluvia*. El tercer enumerador, *Atmosfera3*, controlará el momento del día, dígame *Mañana, Tarde* o *Noche*. Quedando reflejado de la siguiente

manera *public enum Atmosfera1 (Oscuro, Tenue, Intenso), public enum Atmosfera2 (Nublado, Neblina, Lluvia)* y *public enum Atmosfera3 (Manana, Tarde, Noche)*.

- **TiempoE:** Será un tipo de variable que permita cambiar la configuración de la ejecución en tres etapas cronometradas. Se establecerán tres variables de tipo entero (int), que contendrán valores numéricos referentes al tiempo en minutos que se ejecutara la configuración establecida para ese tiempo. El desarrollador del módulo debe tener en cuenta de que cada tiempo de ejecución lleva implícito la configuración de cada una de las variables anteriormente expuestas, haciendo que el entorno cambie en cada uno de los tiempos. Constituyéndose *TiempoE1, TiempoE2 y TiempoE3* para cada uno de ellos, quedando *public int TiempoE1, public int TiempoE2 y public int TiempoE3*.

Para cada uno de los distintos módulos, sus desarrolladores entregarán un documento de texto en el que se referencien a que actores, cámaras o ambientes representan cada una de las variables.

### ***Dirección y Configuración del Archivo de Guardado***

Cada módulo debe poseer un archivo de tipo XML que contenga la configuración que se va a ejecutar. Este archivo de tener el nombre de Perfil.xml y debe ser archivado en una carpeta con nombre Datos. Para guardar este archivo se hará utilizando el código *Application.dataPath* dentro del script de Unity para guardar. Quedando conformado de la siguiente manera con valores hipotéticos:

```
<UserData xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <_iUser>
    <_FileName>Perfil<_FileName/>
    <_FileExt>.xml</_FileExt>
    <_Archive/>
    <Actor1>Pocas</ Actor1>
    <Actor2>Ninguno</ Actor2>
    <Camara>Seguimiento </Camara>
    <Atmosferal>Oscuro</ Atmosferal>
    <Atmosfera2>Lluvia</ Atmosfera2>
    <Atmosfera3>Tarde</ Atmosfera3>
    <TiempoE1>5</ TiempoE1>
    < TiempoE2>15</ TiempoE2>
    < TiempoE3>10</ TiempoE3>
  </_iUser>
</UserData>
```

### ***Extensión de los Módulos Creados***

Después de la realización del módulo este se empaquetara quedando el nombre del mismo con la extensión .unitypackage, ejemplo Playa.unitypackage.

El empaquetamiento de los módulos permite guardar la configuración predeterminada que haga el desarrollador de estos y a la vez ser incluidos exactamente igual en la aplicación VirtualRelax para realizar una actualización con los módulos nuevos.

### **Conclusiones del Capítulo**

En este capítulo se mostró el proceso del desarrollo de la plataforma, basado en las decisiones tomadas en el primer capítulo en lo que se refiere a metodología de desarrollo de aplicaciones y herramientas a utilizar. Con la culminación de este capítulo quedan sentadas las bases para a etapa de realización de pruebas al programa y la validación del mismo mediante su despliegue.

- Las herramientas utilizadas para arribar a la solución, poseen gran cantidad de componentes que aportan al trabajo fiabilidad y sencillez.
- El uso de metodologías y patrones de diseño y arquitectura, permite organizar todo el proceso de desarrollo. Además de garantizar la integridad tanto del proceso como del resultado.
- Se logró crear una plataforma capaz de ejecutar y configurar los módulos de entornos virtuales producidos por el centro VERTEX atendiendo a las deficiencias presentadas por estos.
- Se estableció una estructura estándar para los entornos virtuales que se desarrollen en el centro VERTEX, habilitando su inclusión en el producto final de este trabajo.

## CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

En este capítulo se llevarán a cabo las pruebas de aceptación previamente diseñadas y se documentarán los resultados. El objetivo de su realización es que los programadores puedan probar las funcionalidades del sistema con pruebas descritas mediante las historias de usuario y certificar que la calidad que posee es la esperada por el cliente, ya sean sus características y/o su rendimiento.

### 3.1. Pruebas de Aceptación

Con el objetivo de identificar y corregir fallos u omisiones cometidas, además de comprobar la correcta implementación de las funcionalidades, fueron realizadas las pruebas de caja negra y caja blanca correspondientes a la aplicación. Como mecanismo de detección de errores, la metodología XP, utilizada en el desarrollo de este trabajo, propone las Pruebas de Aceptación, las cuales son realizadas por el cliente en compañía de al menos uno de los integrantes del equipo de desarrollo y son orientadas a las funcionalidades del sistema. Estas son diseñadas a partir de las Historias de Usuario y tiene como objetivo verificar que el programa está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales fue construido.

Las pruebas de aceptación son consideradas como las pruebas de caja negra, estas son utilizadas para comprobar el cumplimiento de los requisitos funcionales de la plataforma, ya que tienen una gran importancia para el éxito de una iteración y aseguran que las funcionalidades cumplen con lo que se espera de ellas. Marcan el camino a seguir indicándole al equipo de desarrollo las funcionalidades más relevantes. Permiten a los programadores conocer que es lo que resta por hacer. Además de brindar la posibilidad de que las pruebas se asemejen al ambiente de producción. Las pruebas principales referentes a este trabajo, se muestran a continuación, las restantes se encuentran en las Tablas 23, 24, 25, 26, 27, 28 y 29 de los Anexos.

**Tabla. 3: Prueba de Aceptación 6.**

|                             |
|-----------------------------|
| <b>Prueba de aceptación</b> |
|-----------------------------|

|   |                              |
|---|------------------------------|
| <b>Caso de prueba:</b> 6  | <b>Número de historia:</b> 6 |
| <b>Nombre de prueba:</b> Escribir y guardar el archivo XML de configuración   |                              |
| <b>Descripción:</b> Permite la escritura sobre el archivo de configuración del entorno, desde la aplicación. Aplicando cambios en tiempo real.  |                              |
| <b>Condiciones de ejecución:</b> Se introducen los valores a las variables que se desean modificar.   |                              |
| <b>Entradas/Pasos de ejecución:</b> Se introducen valores para modificar las variables correspondientes a los actores de la escena (Garzas, Gaviotas y Cámaras). Pueden modificarse todos o solo algunos de los actores. Se verifica que el archivo de configuración asimile los valores añadidos y pueda guardarlos. |                              |
| <b>Resultado esperado:</b> La escena se ejecuta con los valores introducidos y es modificada durante su ejecución. Si se modifica uno solo o varios de los actores, el resto continúa sin cambios.  |                              |
| <b>Evaluación de la prueba:</b> Satisfactoria   |                              |

**Tabla. 4: Prueba de Aceptación 7.**

| <b>Prueba de aceptación</b>   |                              |
|---|------------------------------|
| <b>Caso de prueba:</b> 7  | <b>Número de historia:</b> 7 |
| <b>Nombre de prueba:</b> Cargar entorno   |                              |
| <b>Descripción:</b> Ya la aplicación es una sola, contiene el módulo entre sus atributos y puede modificarlo y ejecutarlo.          |                              |
| <b>Condiciones de ejecución:</b> Iniciar simulación desde aplicación.   |                              |
| <b>Entradas/Pasos de ejecución:</b> Iniciar simulación desde aplicación.  |                              |
| <b>Resultado esperado:</b> La aplicación se ejecuta sin contratiempos y es capaz de cargar, modificar y cerrar el Entorno de Playa. |                              |
| <b>Evaluación de la prueba:</b> Satisfactoria   |                              |

## 3.2. Validación de la plataforma

Para validar este sistema se realizaron principalmente entrevistas como la mostrada en el anexo Entrevista 1: Modelo de Entrevista para Validar, a especialistas del centro VERTEX interesados en el tema, donde expresan su opinión y conformidad con la plataforma luego de probar sus funcionalidades individualmente. Además de las demostraciones funcionales de las Pruebas de Aceptación en concordancia con las Historias de Usuario. Las opiniones recogidas por los especialistas son las siguientes:

- Ing. Victor Manuel Armas Pis: La plataforma VirtualRelax es un sistema informático que permite con una interfaz sencilla integrar los módulos de entornos naturales para terapias de relajación. Se demuestra una buena utilización de Unity 3D como framework y lenguaje de programación C# en su desarrollo, logrando agregar y configurar los actores que intervienen en la escena fácilmente.
- Ing. Gelson Rafael Saurín Ojeda: VirtualRelax presenta una interfaz amigable al usuario, completamente intuitiva y limpia. Permite efectivamente integrar módulos de entornos naturales para terapias de relajación, cumpliendo así con creces su objetivo principal. Se evidencia según su comportamiento un correcto proceso de desarrollo utilizando las tecnologías apropiadas para la tarea. Es una plataforma estable, robusta y de sencilla configuración y utilización.
- Ing. Alexis Echemendía González: La aplicación VirtualRelax posee una interfaz gráfica acorde para un usuario básico, la muestra presenta un diseño minimalista e intuitivo. El tiempo de carga de los módulos es bastante rápido y estos son accesibles fácilmente desde la plataforma. Se evidencia un buen uso de las tecnologías, así como técnicas de programación relacionadas a los lenguajes utilizados. La plataforma es muy estable y no presenta caídas durante su ejecución.
- Ing Luis Guillermo Silva Rojas: La plataforma cuenta con una interfaz gráfica confiable, intuitiva y organizada, que permite una interrelación cómoda entre el software y el usuario. La integración de los módulos es bastante fuerte, lo que permite con su uso, que las terapias de relajación tengan mejores resultados, cumpliendo de esta forma el objetivo de la misma. Se usó la tecnología requerida correctamente, cumpliendo con las metodologías para el desarrollo de este tipo de

aplicaciones, dándole uso a los conocimientos de los lenguajes de programación. La plataforma es estable y soporta todas las configuraciones.

Los resultados arrojados por las entrevistas fueron satisfactorios, evidenciando que la aplicación, a pesar de los inconvenientes presentados por los módulos, cumple con la necesidad de unificar sus funcionalidades e información visual, en favor de un mejor uso en las terapias de relajación.

También cuenta con las validaciones realizadas a los módulos con los que trabaja la aplicación, como es el caso del Entorno de Playa, el cual fue presentado en varios eventos científicos, dentro de los que se encuentran:

- NeuroRehabana 2014 - Evento realizado en el Palacio de las Convenciones de la Habana del 10 al 14 de Marzo del año 2014, obteniendo buenos resultados y capturando información para el mejoramiento del sistema.
- Inauguración de la exposición de salud en Expocuba - Evento realizado el 30 de mayo del mismo año, al cual asistió el ministro de salud, en el mismo se obtuvieron buenos resultados y aceptación por parte de los visitantes de la instalación, en su mayor parte compuesto por personal del ministerio de salud y médicos en general.

Como parte de las pruebas de aceptación de los módulos, se realizaron pruebas betas de estos. Estas pruebas betas, también denominadas pruebas de campo, se realizaron con grupos de ancianos en el Centro Memorial Martin Luther King, Jr., para observar el impacto de su uso. Para la realización de las mismas, se tomó una muestra de 16 ancianos, a los que se les realizó una terapia grupal asistida por un especialista en el tema; inicialmente se les tomaron las pulsaciones a todos, lo cual sirvió como variable para medir el nivel de relajación que logra la aplicación, se realizó la terapia y posteriormente se volvió a tomar las pulsaciones, esto arrojó que del total de la muestra, a 2 ancianos le aumentaron las pulsaciones, 3 se mantuvieron igual y 11 redujeron las mismas; la relajación, en general se muestra con la disminución de las pulsaciones, aunque dependiendo de las características de la terapia o de la persona pueden mantenerse o aumentar.

### 3.3. Modelo de Entrevista utilizada para la obtención de información

- ¿Cómo se realizan las terapias de relajación utilizando aplicaciones informáticas?
- ¿Qué características debe presentar el VirtualRelax para satisfacer una terapia de relajación?

#### **Resultados:**

Las terapias de relajación en las que se utiliza algún tipo de aplicación, son mayormente terapias grupales. Los pacientes que las reciben, son agrupados por la afinidad de sus patologías, es decir, el estrés que padecen no siempre es provocado por las mismas situaciones o conjunto de estas y por lo tanto la manera de tratarlo efectivamente no es la misma. Antes de comenzar la utilización del programa en la terapia, el terapeuta comienza un proceso similar a la hipnosis, en el cual sumerge a los pacientes en un estado de trance a un mundo acorde a lo que se les mostrará en pantalla. Este proceso se realiza atendiendo a las reacciones de los pacientes a los distintos componentes mostrados en la pantalla, puesto que sus posibles fobias o miedos a distintos elementos como animales, agua, sol u otros, pueden interferir con el resultado esperado.

VirtualRelax debe ser una aplicación altamente configurable, capaz de administrar los módulos anteriormente creados para apoyar las terapias de relajación y debe dar la posibilidad de incluir a los futuros módulos que puedan ser creados con el mismo fin. Además deberá dar apoyo a estudios recientes en el área terapéutica sobre los efectos de las variaciones en el entorno de simulación, teniendo la funcionalidad de variar la intervención de los componentes del medio en distintas etapas de la ejecución del programa. También debe tener una interfaz sencilla y amigable que preserve sus funcionalidades, puesto que los usuarios finales que harán uso de la aplicación, no son especialistas en el uso de ordenadores, ni están comprometidos con las responsabilidades que esto pueda traer consigo.

## **Conclusiones del Capítulo**

A través de la validación y las pruebas realizadas se puede concluir que se cumple con el objetivo principal de este trabajo, la creación de una plataforma capaz de administrar de una mejor y más cómoda manera los módulos de simulación de entornos existentes, sentando bases para futuras versiones con los módulos que sean creados y así poder liberar el producto.

## CONCLUSIONES

Con la culminación de este trabajo investigativo que abarcó los temas referentes a las plataformas integradoras, simulaciones de entornos virtuales, herramientas de diseño y terapias de relajación, es posible llegar a las siguientes conclusiones:

- Se comprobó que con la plataforma desarrollada, se pueden manejar de manera sencilla y sin interrupción, los módulos sobre los cuales opera.
- Se establece que las herramientas utilizadas (Unity 3D), los lenguajes de programación (Java Script, C Sharp) y metodologías (XP), fueron factibles para el desarrollo de esta plataforma.
- Se demostró a partir de las pruebas de aceptación y la validación de la plataforma que las funcionalidades desarrolladas satisfacen las necesidades planteadas por el cliente.

La liberación por el centro VERTEX de la plataforma desarrollada, permitirá su uso en las terapias de relajación y el tratamiento de pacientes con estrés, apoyando y facilitando el trabajo de los especialistas en esta área.

## RECOMENDACIONES

Motivando la continuidad en el desarrollo de la aplicación aquí presentada, el autor recomienda:

- La creación de una mayor cantidad de módulos con requisitos no funcionales como los de esta plataforma, que sean capaces de asistir o apoyar terapias como la referida en el trabajo.
- La corrección de los entornos Playa y Pecera, atendiendo a los errores de configuración y ejecución que presentan actualmente, enfocándolos a la estructura planteada para los módulos en este trabajo.

## BIBLIOGRAFÍA

- 3D, Unity. 2011.** Unity Technologies. *Unity 3D. Physics.* [En línea] 2011. <http://unity3d.com/unity/features/physics>.
- Alfonso, Arlan Hernandez. 2012.** *Análisis y diseño del módulo de gestión de los recursos humanos para el sistema GESPRO.* La Habana : s.n., 2012.
- Alonso, Jesus Alonso. 2009.** *Video Juegos 3D.* Cataluña : Universidad Abierta de Cataluña, 2009.
- Anonimo. 2014.** Mis Respuestas.com. [En línea] 10 de Marzo de 2014. <http://www.misrespuestas.com>.
- aprendizaje, Red de. 2015.** Red de aprendizaje. [En línea] 2015. <http://www.reddeaprendizaje.com>.
- Astudillo, Marcello Visconti y Hernan. 2007.** *Fundamentos de la Ingenieria de Software.* 2007.
- Aula, Comunidad. 2014.** Comunidad Aula. *Mi experiencia Aula.* [En línea] 2014. <https://comunidadaula.wordpress.com/tag/interactividad/>.
- Autores, Varios. 2013.** NYU Lagone Medical Center. [En línea] 2013. <http://irb.med.nyu.edu/>.
- Barrios, Francisco. 2013.** La Relajacion. [En línea] 2013. <http://www.larelajacion.com/larelajacion/43autoalusiva.php>.
- Cervera, Alonso. 2013.** Wondershare. [En línea] 2013. <http://www.wondershare.es/photo-editing-tips/free-mac-photo-editor.html>.
- Complementaria, Programa Nacional de Medicina. 2000.** *Manual de Relajacion.* Lima : Centro de Documentación Carlos Enrique Paz Roldán, 2000.
- creativo, Mi yo. 2014.** Mi yo creativo. [En línea] 2014. <http://www.miyocreativo.com>.
- Day, Robert A. 2005.** *Como escribir y publicar trabajos científicos.* Washington DC : The Oryx Press, 2005.
- Definicion, Grupo de. 2013.** Grupo de Definicion. *Entornos virtuales 3D clasicos e inteligentes.* [En línea] 2013. [www.definicion.de](http://www.definicion.de).
- Ecured. 2014.** Ecured. [En línea] 2014. <https://www.ecured.cu/Unity3D>.
- Hernández, Alexander Navarro. 2014.** *Simulación de pecera virtual como herramienta para las terapias de relajación.* La Habana : Universidad de las Ciencias Informaticas, 2014.
- Hierrezuelo, Haydee Asela. 2004.** *Relajacion 21.* La Habana : s.n., 2004.

- Ibaiondo. 2014.** iPad Foros. *IPad Foros*. [En línea] 2014. <http://www.ipadforos.com/software/halotea-102-software-creador-sonidos-naturaleza-para-relajarse-t32761.html>.
- Kanaga, Ed Key y David. 2012.** Proteus. [En línea] 2012. <http://www.visitproteus.com/>.
- Kapler, Alan. 2013.** Dream Aquarium. [En línea] 2013. <http://www.dreamaquarium.com/>.
- Llavori, Rafael Berlanga. 2010.** Repositori Universitat Jaume I. [En línea] 2010. <http://repositori.uji.es/xmlui/>.
- Meloleo. 2014.** Meloleo. *Meloleo*. [En línea] 5 de Abril de 2014. <http://www.meloleo.com/10-juegos-flash-para-relajarse>.
- Metodologia de la investigacion. Hernández, S.,R. 2008.* Habana, 475 pp. : Felix Varela, 2008.
- Ordoñez, Maite Rodriguez y Meylin. 2007.** *La Metodología XP aplicable al desarrollo del Software Educativo en Cuba*. La Habana : Universidad de las Ciencias Informáticas, 2007.
- otros, Gerardo Abraham Morales y. 2010.** *Proceso de Desarrollo para Videojuegos*. Juarez : Universidad Autonoma de Juarez, 2010.
- otros, Matthew Ashenden y. 2013.** Pzizz. [En línea] 2013. [http://pzizz.com/software\\_intro.asp](http://pzizz.com/software_intro.asp).
- Penadés, Patricio Letelier y M<sup>a</sup> Carmen. 2009.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Valencia : Universidad Politécnica de Valencia, 2009.
- Pressman, Roger S. 2008.** *Ingeniería de Software, un enfoque práctico*. McGraw-Hill : s.n., 2008.
- Riva, Giuseppe. 2004.** *Immersive Virtual Telepresence*. Amsterdam : IOS Press, 2004.
- Rondon, Kenia Labori Izary. 2007.** *Metodologías para el desarrollo de Software multimedia , analisis comparativo y propuesta*. La Habana : Universidad de las Ciencias Informáticas, 2007.
- Sanchez, Orlando. 2013.** *Módulo para la creación de videojuegos para jugadores virtuales de tipo batalla utilizando Unity 3D*. La Habana : Universidad de las Ciencias Informáticas, 2013.
- Software, Vector Media. 2011.** Vector Media Software. [En línea] 2011. <http://www.vectormediasoftware.com/atmdeluxehome.htm>.
- Team, Rixane. 2013.** Rixane Interactive. [En línea] 2013. <http://www.rixane.com/fantastic-ocean-3d-screensaver.html>.
- Tejeda, Ronniel Arencibia. 2014.** *Simulación de un entorno de playa para*. La Habana : Universidad de las Ciencias Informaticas, 2014.

**Tellez, Gabriel. 2012.** *XP Practices: A Successful Tool for Increasing and Transferring Practical Knowledge in Short-Life Software Development Projects.* 2012.

**Tellez, Linda Luna. 2012.** Scribd. [En línea] 28 de Abril de 2012. <http://es.scribd.com>.

**UnityTechnologies. 2012.** Scripting. *Unity 3D.* [En línea] 2012. <http://unity3d.com/unity/features/scripting>.

*Using virtual environments as training simulators.* **C. E. Lathan, M. R. Tracey, M. M. Sebrechts, D. M. Clawson & G. A. Higgins. 2002.** 2002, Handbook of virtual environments: Design, implementation, and applications, págs. 403-414.

*Vol VIII.* **Calderon, Miguel Lozano y Carlos. 2004.** 2004, Revista Iberoamericana de Inteligencia Artificial, págs. 85-94.

**Windows, Xataka. 2014.** Xataka Windows. [En línea] 2014. <http://www.xatakawindows.com/bienvenidoawindows8/las-11-mejores-aplicaciones-de-edicion-de-sonido-para-windows>.

## GLOSARIO DE TÉRMINOS

**2D:** En geometría y análisis matemático, un objeto o ente es tridimensional si tiene dos dimensiones. Cada uno de sus puntos puede ser localizado especificando dos números dentro de un cierto rango.

**3D:** En geometría y análisis matemático, un objeto o ente es tridimensional si tiene tres dimensiones. Es decir cada uno de sus puntos puede ser localizado especificando tres números dentro de un cierto rango.

**Assets:** (Bloques de construcción), son utilizados por el editor de entornos de Unity 3D para estructurar y organizar los elementos que se utilizan en los videojuegos.

**Bit:** Es la unidad básica y mínima que puede transmitirse en un ordenador.

**Boo:** Es un lenguaje de programación orientado a objetos. Se considera como un dialecto de Python por su similitud con el mismo, aunque en su creación se tuvieron en cuenta también elementos de Perl y Ruby.

**C++:** Es un lenguaje de programación orientado a objetos derivado del C. Creado en 1983 por Bjarne Stroustrup.

**C#:** Es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft.

**DirectX 3D:** Parte de DirectX (conjunto de bibliotecas para multimedia), propiedad de Microsoft. Consiste en una API para la programación de gráficos 3D.

**GB:** Se trata de una unidad de medida y su equivalencia es de 1024 MB.

**GHz:** Unidad de medida para las frecuencias del reloj interno de un procesador electrónico.

**Intel Pentium IV:** Es un microprocesador de séptima generación basado en la arquitectura x86 y fabricado por Intel.

**JavaScript:** es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

**MB:** También conocido como megaocteto se refiere a una medida de cantidad en la informática.

**Open GL:** Es una librería escrita en C que permite la manipulación de gráficos 3D a todos los niveles.

**Plugin:** Es un complemento que se relaciona con otra aplicación para aportarle una nueva función.

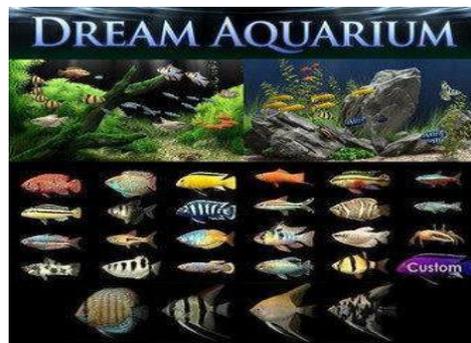
**RAM:** (Random Access Memory) es donde el computador guarda los datos que está utilizando en el momento presente.

**Realidad Virtual:** Es una ciencia basada en el empleo de ordenadores y otros dispositivos, cuyo fin es producir una apariencia de realidad que permita al usuario tener la sensación de estar presente en ella.

**Render:** Es el proceso de generar una imagen desde un modelo. Este término técnico es utilizado por los animadores o productores audiovisuales y en programas de diseño en 3D.

**Script:** Los script son casi siempre interpretados. El uso habitual de los scripts es realizar diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario.

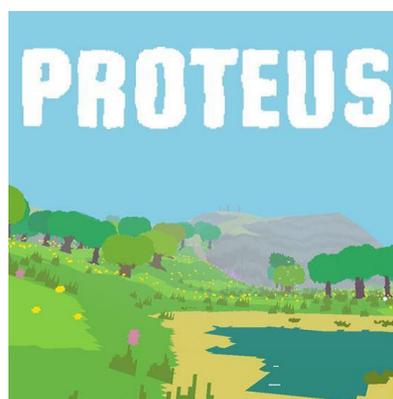
# ANEXOS



**Fig. 2: Dream Aquarium.**



**Fig. 3: Halotea.**



**Fig. 4: Proteus.**



**Fig. 5: Vista general del entorno de pecera.**



**Fig. 6: Vista de acercamiento (Paneo de cámara) del entorno de pecera.**

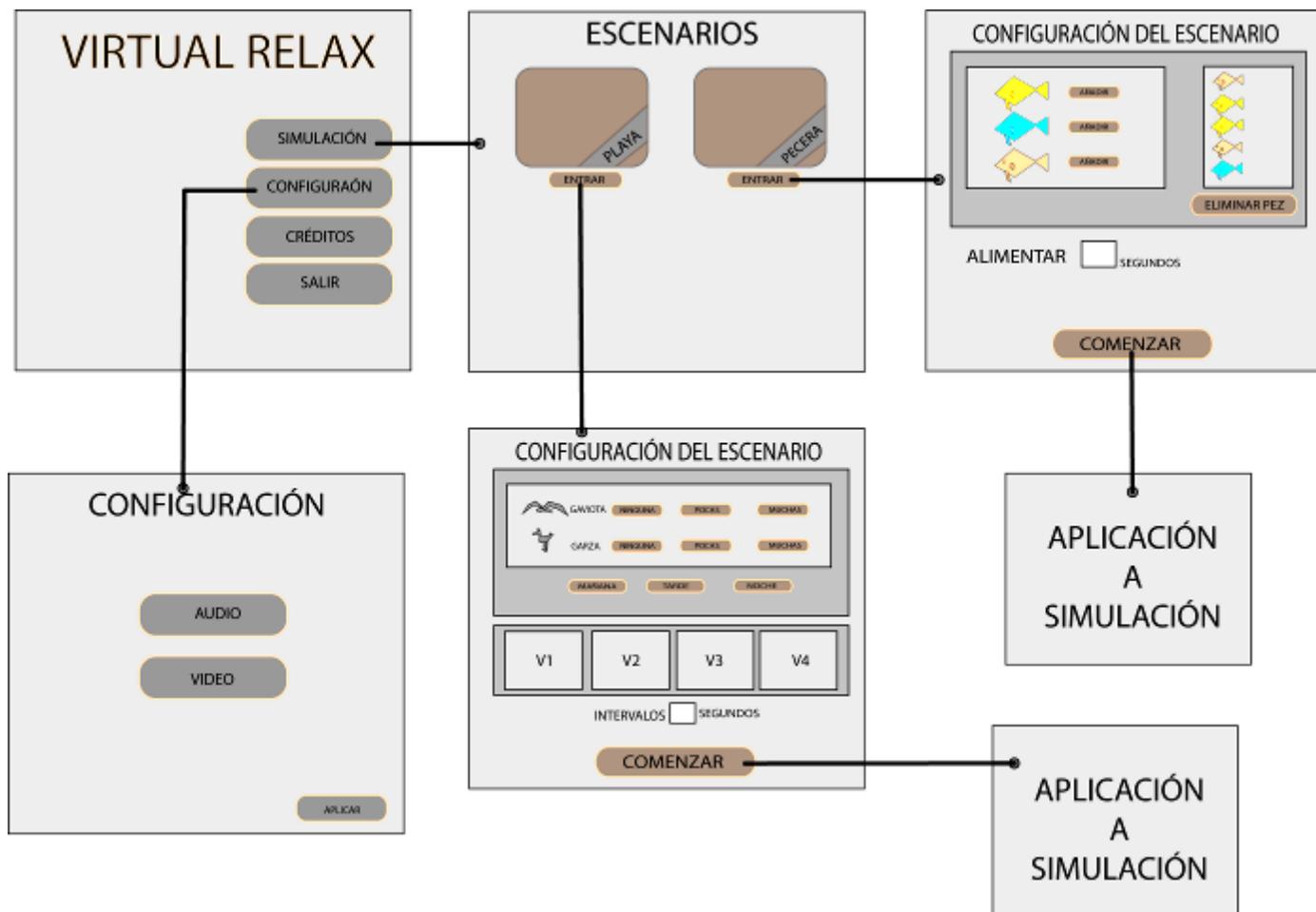


Fig. 7: Solución Propuesta.

Tabla. 5: HU Verificación del correcto funcionamiento del Entorno de Pecera.

| Historia de Usuario |  |
|---------------------|--|
| Número: 1           | Nombre: Verificación del correcto funcionamiento del Entorno de Pecera |
| Usuario: Autor      |  |

|  |                              |
|--|------------------------------|
| <b>Modificación de Historia Número:</b>  | <b>Iteración Asignada: 1</b> |
| <b>Prioridad en Negocio:</b> Alta<br>(Alta / Media / Baja)   | <b>Puntos Estimados: 1</b>   |
| <b>Riesgo en Desarrollo:</b> Alta<br>(Alta / Media / Baja)   | <b>Puntos Reales: 1</b>      |
| <b>Descripción:</b> Para ejecutar y probar funcionalmente el Entorno de Pecera, se le introducen datos para una configuración básica al archivo xml, este archivo es el responsable de controlar el comportamiento de algunos de los componentes del entorno cuando se inicia la simulación. |                              |
| <b>Observaciones:</b> No se puede acceder al archivo xml, el Centro VERTEX solo cuenta con el ejecutable y no con las carpetas del proyecto donde se encuentra el archivo de configuración.  |                              |

**Tabla. 6: HU Verificación del correcto funcionamiento del Entorno de Playa.**

|   |  |
|---|--|
| <b>Historia de Usuario</b>  |  |
| <b>Número:</b> 2  | <b>Nombre:</b> Verificación del correcto funcionamiento del Entorno de Playa |
| <b>Usuario:</b> Autor   |  |
| <b>Modificación de Historia Número:</b>   | <b>Iteración Asignada: 1</b>   |
| <b>Prioridad en Negocio:</b> Alta<br>(Alta / Media / Baja)  | <b>Puntos Estimados: 1</b>   |
| <b>Riesgo en Desarrollo:</b> Alta<br>(Alta / Media / Baja)  | <b>Puntos Reales: 1</b>  |
| <b>Descripción:</b> Para ejecutar y probar funcionalmente el Entorno de Playa, se le introducen datos para una configuración básica al archivo xml, este archivo es el responsable de controlar el comportamiento de algunos de los componentes del entorno cuando se inicia la simulación. |  |
| <b>Observaciones:</b> Se pudo ejecutar sin contratiempos el entorno mencionado configurando el archivo xml, no obstante se detectaron deficiencias en el módulo, no corregibles puesto que es una versión ejecutable.   |  |

**Tabla. 7: HU Acceso y modificación de la variable Agregar Garza.**

| <b>Historia de Usuario</b>  |   |
|---|---|
| <b>Número:</b> 3  | <b>Nombre:</b> Acceso y modificación de la variable Agregar Garza |
| <b>Usuario:</b> Autor   |   |
| <b>Modificación de Historia Número:</b>   | <b>Iteración Asignada:</b> 1                                      |
| <b>Prioridad en Negocio:</b> Alta<br>(Alta / Media / Baja)  | <b>Puntos Estimados:</b> 1  |
| <b>Riesgo en Desarrollo:</b> Alta<br>(Alta / Media / Baja)  | <b>Puntos Reales:</b> 1   |
| <b>Descripción:</b> Acceder a través de la plataforma a la variable controladora del número de garzas en el Entorno de Playa dentro del archivo xml.  |   |
| <b>Observaciones:</b> Permite comprobar el acceso de la variable controladora del número de garzas en la escena desde la plataforma y probar su modificación desde la misma, sin tener que hacerlo directamente en el archivo de configuración xml. |   |

**Tabla. 8: HU Acceso y modificación de la variable Agregar Gaviota.**

| <b>Historia de Usuario</b>                                 |   |
|--|---|
| <b>Número:</b> 4   | <b>Nombre:</b> Acceso y modificación de la variable Agregar Gaviota |
| <b>Usuario:</b> Autor                                      |   |
| <b>Modificación de Historia Número:</b>                    | <b>Iteración Asignada:</b> 1  |
| <b>Prioridad en Negocio:</b> Alta<br>(Alta / Media / Baja) | <b>Puntos Estimados:</b> 1  |
| <b>Riesgo en Desarrollo:</b> Alta<br>(Alta / Media / Baja) | <b>Puntos Reales:</b> 1   |

|   |
|---|
| <b>Descripción:</b> Acceder a través de la plataforma a la variable controladora del número de gaviotas en el Entorno de Playa dentro del archivo xml.  |
| <b>Observaciones:</b> Permite comprobar el acceso de la variable controladora del número de gaviotas en la escena desde la plataforma y probar su modificación desde la misma, sin tener que hacerlo directamente en el archivo de configuración xml. |

**Tabla. 9: HU Acceso y modificación de la variable Tipo de Cámara.**

| Historia de Usuario   |  |
|---|--|
| <b>Número:</b> 5  | <b>Nombre:</b> Acceso y modificación de la variable Tipo de Cámara |
| <b>Usuario:</b> Autor   |  |
| <b>Modificación de Historia Número:</b>   | <b>Iteración Asignada:</b> 1                                       |
| <b>Prioridad en Negocio:</b> Alta<br>(Alta / Media / Baja)  | <b>Puntos Estimados:</b> 1   |
| <b>Riesgo en Desarrollo:</b> Alta<br>(Alta / Media / Baja)  | <b>Puntos Reales:</b> 1  |
| <b>Descripción:</b> Acceder a través de la plataforma a la variable controladora del tipo de cámara a usar en la escena, es decir la vista que será mostrada del Entorno de Playa dentro del archivo xml.   |  |
| <b>Observaciones:</b> Permite comprobar el acceso de la variable controladora del tipo de cámara en la escena desde la plataforma y probar su modificación desde la misma, sin tener que hacerlo directamente en el archivo de configuración xml. |  |

**Tabla. 10: Plan de Entregas.**

| Entregas | Historias de Usuario | Número |
|----------|----------------------|--------|
|----------|----------------------|--------|

|           |  |   |
|-----------|--|---|
| Entrega 1 | Verificación del correcto funcionamiento del Entorno de Pecera | 1 |
|           | Verificación del correcto funcionamiento del Entorno de Playa  | 2 |
| Entrega 2 | Acceso y modificación de la variable Agregar Garza             | 3 |
|           | Acceso y modificación de la variable Agregar Gaviota           | 4 |
|           | Acceso y modificación de la variable Tipo de Cámara            | 5 |
| Entrega 3 | Configurar y modificar XML                                     | 6 |
|           | Cargar entorno   | 7 |

**Tabla. 11: Plan de entregas por Iteraciones.**

| Entregas  | Iteración 1<br>4ta Semana Octubre | Iteración 2<br>1ra Semana Diciembre | Iteración 3<br>2da Semana Enero |
|-----------|-----------------------------------|-------------------------------------|---------------------------------|
| Entrega 1 | Versión 1                         | Versión 2                           | Versión 3 (Final)               |
| Entrega 2 |                                   | Versión 1                           | Versión 2 (Final)               |
| Entrega 3 |                                   |                                     | Versión 1 (Final)               |

**Tabla. 12: Patrones de Diseño.**

| Nombre del Patrón | Descripción   | Utilización |
|-------------------|---|-------------|
| Patrón Experto    | Asignan una responsabilidad al experto en información: la clase que | WriteXML.cs |

|                          |  |   |
|--------------------------|--|---|
|                          | cuenta con la información necesaria para cumplir la responsabilidad.   |   |
| Patrón Bajo Acoplamiento | Asignan las responsabilidades de forma tal que las clases se comuniquen con el menor número de clases que sea posible.   | Todas las clases                        |
| Patrón Alta Cohesión     | Asigna a una o varias clases responsabilidades moderadas dentro de un área funcional y colaboran con otras para llevar a cabo las tareas. Caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. | Todas las clases                        |
| Patrón Creador           | Este patrón guía la asignación de responsabilidades en la creación de objetos.   | LoadLevel_OnClick.cs                    |
| Patrón Controlador       | Asigna la responsabilidad del manejo de eventos en el sistema, dependiendo de la acción de un actor externo.   | Componentes interactivos de la Interfaz |

**Tabla. 13: Tarjeta CRC del Script “LoadLevel\_OnClick.cs”.**

|                             |
|-----------------------------|
| <b>LoadLevel_OnClick.cs</b> |
|-----------------------------|

|   |  |
|---|--|
| <b>Descripción:</b> Se encarga de cargar las diferentes escenas en el proyecto dependiendo de los valores seleccionados a través de la UI |  |
| <b>Atributos:</b>   |  |
| <b>Nombre:</b>  | <b>Descripción:</b>                        |
| <b>Responsabilidades:</b>   |  |
| <b>LoadLevel(String)</b>  | Carga la escena que se pasa por parámetros |
| <b>Quit()</b>   | Salida de la aplicación                    |

**Tabla. 14: Tarjeta CRC del Script "WriteXML.cs".**

|  |   |
|--|---|
| <b>WriteXML.cs</b>   |   |
| <b>Descripción:</b> Se encarga de la comunicación visual, verifica la existencia del archivo XML y escribe los datos seleccionados a través de la UI |   |
| <b>Atributos:</b>  |   |
| <b>Nombre:</b>   | <b>Descripción:</b>                         |
| <b>CantGaviotas</b>  | La cantidad de gaviotas en la escena        |
| <b>CantGarzas</b>  | La cantidad de garzas en la escena          |
| <b>Camaras</b>   | Selección del tipo de cámara                |
| <b>Responsabilidades:</b>  |   |
| <b>WriteFile()</b>   | Escribe en el archivo XML                   |
| <b>SaveData()</b>  | Guarda las modificaciones en el archivo XML |
| <b>Gaviotas(String)</b>  | Escribe la cantidad de gaviotas             |
| <b>Garzas(int)</b>   | Escribe la cantidad de garzas               |
| <b>Camaras(String)</b>   | Escribe que cámara selecciona el usuario    |

**Tabla. 15: Tarea de investigación 1.**

|   |                                    |
|---|------------------------------------|
| <b>Tareas de investigación</b>  |                                    |
| <b>Número:</b> 1  | <b>Número de Historia:</b> 1       |
| <b>Nombre de la tarea:</b> Comprobación del funcionamiento del entorno Pecera |                                    |
| <b>Tipo de tarea:</b> Investigación   | <b>Puntos estimados:</b> 0.5       |
| <b>Fecha de inicio:</b> 15 de octubre   | <b>Fecha de fin:</b> 21 de octubre |
| <b>Programador responsable:</b> Daniel Valdés Fernández                       |                                    |
| <b>Descripción:</b> Se comprueba la ejecución y funcionalidad del módulo      |                                    |

**Tabla. 16: Tarea de investigación 2.**

|  |                                    |
|--|------------------------------------|
| <b>Tareas de investigación</b>   |                                    |
| <b>Número:</b> 2   | <b>Número de Historia:</b> 2       |
| <b>Nombre de la tarea:</b> Comprobación del funcionamiento del entorno Playa |                                    |
| <b>Tipo de tarea:</b> Investigación  | <b>Puntos estimados:</b> 0.5       |
| <b>Fecha de inicio:</b> 22 de octubre  | <b>Fecha de fin:</b> 26 de octubre |
| <b>Programador responsable:</b> Daniel Valdés Fernández                      |                                    |
| <b>Descripción:</b> Se comprueba la ejecución y funcionalidad del módulo     |                                    |

**Tabla. 17: Tarea de investigación 3.**

|   |                                     |
|---|-------------------------------------|
| <b>Tareas de investigación</b>  |                                     |
| <b>Número:</b> 3  | <b>Número de Historia:</b> 3        |
| <b>Nombre de la tarea:</b> Control de la variable Agregar Garza                                 |                                     |
| <b>Tipo de tarea:</b> Desarrollo  | <b>Puntos estimados:</b> 1          |
| <b>Fecha de inicio:</b> 28 de noviembre   | <b>Fecha de fin:</b> 1 de diciembre |
| <b>Programador responsable:</b> Daniel Valdés Fernández   |                                     |
| <b>Descripción:</b> Se accede desde la aplicación a las variables de control del entorno(Garza) |                                     |

**Tabla. 18: Tarea de investigación 4.**

|   |                                     |
|---|-------------------------------------|
| <b>Tareas de investigación</b>                                    |                                     |
| <b>Número:</b> 4  | <b>Número de Historia:</b> 4        |
| <b>Nombre de la tarea:</b> Control de la variable Agregar Gaviota |                                     |
| <b>Tipo de tarea:</b> Desarrollo                                  | <b>Puntos estimados:</b> 1          |
| <b>Fecha de inicio:</b> 2 de diciembre                            | <b>Fecha de fin:</b> 3 de diciembre |
| <b>Programador responsable:</b> Daniel Valdés Fernández           |                                     |

**Descripción:** Se accede desde la aplicación a las variables de control del entorno(Gaviota)

**Tabla. 19: Tarea de investigación 5.**

|  |                                     |
|--|-------------------------------------|
| <b>Tareas de investigación</b>   |                                     |
| <b>Número:</b> 5   | <b>Número de Historia:</b> 5        |
| <b>Nombre de la tarea:</b> Control de la variable Agregar Cámara                                 |                                     |
| <b>Tipo de tarea:</b> Desarrollo   | <b>Puntos estimados:</b> 1          |
| <b>Fecha de inicio:</b> 3 de diciembre   | <b>Fecha de fin:</b> 5 de diciembre |
| <b>Programador responsable:</b> Daniel Valdés Fernández  |                                     |
| <b>Descripción:</b> Se accede desde la aplicación a las variables de control del entorno(Cámara) |                                     |

**Tabla. 20: Tarea de investigación 6.**

|  |                                 |
|--|---------------------------------|
| <b>Tareas de investigación</b>   |                                 |
| <b>Número:</b> 6   | <b>Número de Historia:</b> 6    |
| <b>Nombre de la tarea:</b> Modificación del archivo de configuración XML |                                 |
| <b>Tipo de tarea:</b> Desarrollo   | <b>Puntos estimados:</b> 1.5    |
| <b>Fecha de inicio:</b> 6 de enero                                       | <b>Fecha de fin:</b> 8 de enero |

|   |
|---|
| <b>Programador responsable:</b> Daniel Valdés Fernández                                     |
| <b>Descripción:</b> Se prepara la aplicación, para controlar los módulos, desde su interfaz |

**Tabla. 21: Tarea de investigación 7.**

|  |                                  |
|--|----------------------------------|
| <b>Tareas de investigación</b>   |                                  |
| <b>Número:</b> 7   | <b>Número de Historia:</b> 6     |
| <b>Nombre de la tarea:</b> Configuración de autoguardado en ejecución  |                                  |
| <b>Tipo de tarea:</b> Desarrollo   | <b>Puntos estimados:</b> 1.5     |
| <b>Fecha de inicio:</b> 9 de enero   | <b>Fecha de fin:</b> 13 de enero |
| <b>Programador responsable:</b> Daniel Valdés Fernández  |                                  |
| <b>Descripción:</b> Se prepara al Sistema para guardar las modificaciones realizadas en tiempo real y pueda partir de ellas en su reinicio |                                  |

**Tabla. 22: Tarea de investigación 8.**

|   |                              |
|---|------------------------------|
| <b>Tareas de investigación</b>                          |                              |
| <b>Número:</b> 8  | <b>Número de Historia:</b> 7 |
| <b>Nombre de la tarea:</b> Composición de la aplicación |                              |
| <b>Tipo de tarea:</b> Desarrollo                        | <b>Puntos estimados:</b> 1.5 |

|   |                                  |
|---|----------------------------------|
| <b>Fecha de inicio:</b> 14 de enero   | <b>Fecha de fin:</b> 15 de enero |
| <b>Programador responsable:</b> Daniel Valdés Fernández   |                                  |
| <b>Descripción:</b> Permite agregar el entorno y ejecutarlo con las modificaciones previas o modificarlo en tiempo real |                                  |

**Tabla. 23: Prueba de Aceptación 1.**

|   |                              |
|---|------------------------------|
| <b>Prueba de aceptación</b>   |                              |
| <b>Caso de prueba:</b> 1  | <b>Número de historia:</b> 1 |
| <b>Nombre de prueba:</b> Ejecutar entorno Pecera  |                              |
| <b>Descripción:</b> Comprobar que el entorno contiene todos los elementos necesarios para simular un entorno de pecera. |                              |
| <b>Condiciones de ejecución:</b> Iniciar simulación   |                              |
| <b>Entradas/Pasos de ejecución:</b> Iniciar simulación  |                              |
| <b>Resultado esperado:</b> El entorno se ejecute sin fallos y el correcto funcionamiento de sus modificadores           |                              |
| <b>Evaluación de la prueba:</b> Fallida   |                              |

**Tabla. 24: Prueba de Aceptación 2.**

|                             |                              |
|-----------------------------|------------------------------|
| <b>Prueba de aceptación</b> |                              |
| <b>Caso de prueba:</b> 2    | <b>Número de historia:</b> 2 |

|   |
|---|
| <b>Nombre de prueba:</b> Ejecutar entorno Playa   |
| <b>Descripción:</b> Comprobar que el entorno contiene todos los elementos necesarios para simular un entorno de pecera. |
| <b>Condiciones de ejecución:</b> Iniciar simulación   |
| <b>Entradas/Pasos de ejecución:</b> Iniciar simulación  |
| <b>Resultado esperado:</b> El entorno se ejecute sin fallos y el correcto funcionamiento de sus modificadores           |
| <b>Evaluación de la prueba:</b> Satisfactoria   |

**Tabla. 25: Prueba de Aceptación 3.**

| <b>Prueba de aceptación</b>   |                              |
|---|------------------------------|
| <b>Caso de prueba:</b> 3  | <b>Número de historia:</b> 3 |
| <b>Nombre de prueba:</b> Acceso y modificación de la variable Agregar Garza   |                              |
| <b>Descripción:</b> El caso de prueba permite comprobar que se pueda acceder al archivo de configuración XML y modificar la cantidad de garzas en el entorno. |                              |
| <b>Condiciones de ejecución:</b> Se debe de introducir correctamente los parámetros a través de la interfaz creada.   |                              |
| <b>Entradas/Pasos de ejecución:</b> En la interfaz creada se selecciona la cantidad de garzas que se desean en la escena (valores numéricos).                 |                              |
| <b>Resultado esperado:</b> En la escena se muestran las garzas en correspondencia a los valores introducidos a través de la interfaz creada.                  |                              |
| <b>Evaluación de la prueba:</b> Satisfactoria   |                              |

**Tabla. 26: Prueba de Aceptación 4.**

| <b>Prueba de aceptación</b>   |                              |
|---|------------------------------|
| <b>Caso de prueba:</b> 4  | <b>Número de historia:</b> 4 |
| <b>Nombre de prueba:</b> Acceso y modificación de la variable Agregar Gaviota   |                              |
| <b>Descripción:</b> El caso de prueba permite comprobar que se pueda acceder al archivo de configuración XML y modificar la cantidad de gaviotas en el entorno. |                              |
| <b>Condiciones de ejecución:</b> Se debe de introducir correctamente los parámetros a través de la interfaz creada.   |                              |
| <b>Entradas/Pasos de ejecución:</b> En la interfaz creada se selecciona la cantidad de gaviotas que se desean en la escena (valores numéricos).                 |                              |
| <b>Resultado esperado:</b> En la escena se muestran las gaviotas en correspondencia a los valores introducidos a través de la interfaz creada.                  |                              |
| <b>Evaluación de la prueba:</b> Satisfactoria   |                              |

**Tabla. 27: Prueba de Aceptación 5.**

| <b>Prueba de aceptación</b>  |                              |
|--|------------------------------|
| <b>Caso de prueba:</b> 5   | <b>Número de historia:</b> 5 |
| <b>Nombre de prueba:</b> Acceso y modificación de la variable Tipo de Cámara   |                              |
| <b>Descripción:</b> El caso de prueba permite comprobar que se pueda acceder al archivo de configuración XML y modificar los tipos de cámaras en el entorno. |                              |
| <b>Condiciones de ejecución:</b> Se debe de introducir correctamente los parámetros a través de la interfaz creada.  |                              |
| <b>Entradas/Pasos de ejecución:</b> En la interfaz creada se selecciona el tipo de cámara que se desea en la escena (uso de herramienta de selección).       |                              |

**Resultado esperado:** La visión de la escena durante la ejecución se corresponde con el tipo de cámara seleccionado.

**Evaluación de la prueba:** Satisfactoria

## **MODELO DE ENTREVISTA A LOS ESPECIALISTAS PARA LA VALIDACION DE LA PLATAFORMA**

¿Cuál es su opinión acerca de la interfaz gráfica de la plataforma?

¿Considera que la plataforma integra los módulos de entornos virtuales?

¿Cree Ud. correcto el uso de las tecnologías y técnicas utilizadas en el desarrollo de la plataforma?

¿Qué características durante la ejecución de la plataforma considera relevantes?

**Entrevista 2: Modelo de Entrevista para Validar.**