

2 Marco teórico

En esta sección se definen los conceptos principales a ser utilizados en este documento: mundos virtuales, la técnica de Workshops para la Elicitación de Requisitos y diagramas para el modelamiento del juego. Se hace una descripción del mundo virtual Second Life que será usado como plataforma de desarrollo de TREG. El contenido del juego utiliza la técnica de Workshops como base para la simulación. Además, se da una visión general de las técnicas a ser utilizadas en el desarrollo del software.

2.1. Mundos Virtuales

Un mundo virtual es un ambiente interactivo multiusuario simulado por una computadora y presenta las siguientes características [10]:

1. Ambiente compartido: Los usuarios comparten simultáneamente el mismo mundo;
2. Interfaz de usuario gráfica: Ambiente gráfico que puede variar desde un ambiente en 2D o 3D;
3. Inmediato: La interacción con el mundo ocurre en tiempo real;
4. Interactividad: El mundo permite a los usuarios a realizar cambios, construir o personalizar su contenido;
5. Persistencia: El ambiente deberá existir continuamente; y
6. Socialización/Comunidad: Estos mundos promueven la formación de grupos como equipos, clubes, vecinos, gremios, etc.

Los mundos virtuales son creados con diversos propósitos [10] como juegos comerciales, donde la gran mayoría cae en esta categoría, también conocidos como de MMPORG (Massively Multiplayer Online Role Playing Game). También hay aquellos orientados a la educación como Active Worlds o Mokitown. Los mundos virtuales también pueden tener propósitos especializados como American's Army para el entrenamiento militar y AgoraXhange para la expresión política. Finalmente, están los mundos virtuales que no tienen ningún propósito específico como por ejemplo Second Life, The Sims, Kaneva y There.

En ellos, los usuarios sociabilizan, y construyen o decoran un espacio. En este proyecto, se explora Second Life dada la libertad que ofrece en la creación y programación del mundo virtual.

2.1.1. Second Life

El mundo virtual Second Life fue creado por Linden Labs en el 2003 [2]. Los usuarios, también llamados residentes, representados como avatares interactúan, crean y realizan negocios. Los avatares, que pueden ser configurables a cualquier forma física, interactúan con otros avatares de muchas formas; pudiendo explorar, sociabilizar, participar de conferencias, entre otras cosas. Además, pueden realizar negocios intercambiando productos o servicios, comprando y vendiendo con la moneda virtual Linden Dollars. Los residentes son quienes crean el contenido en Second Life, construyendo su propio mundo o elementos del mundo.

El contenido de Second Life puede ser creado a través de sus editores de terreno, de objetos y de scripts. La construcción de objetos es posible por medio de un editor propio de Second Life que permite crear, modelar y vincular los objetos creados a partir objetos 3D básicos llamados primitivos (*prims*) como cajas, esferas o cilindros. Los objetos pueden contener hasta 255 prims y pueden ser editados de diversas formas al aplicarle texturas, rotarlos, redimensionarlos, cambiar la física de los objetos, entre otras propiedades.

El comportamiento de estos objetos y su comunicación con avatares y otros objetos se realiza por medio de scripts. Los scripts son programados usando un editor propio del lenguaje orientado a eventos LSL (Linden Scripting Language) y son interpretados y ejecutados por el servidor de Second Life. Los scripts en LSL deben tener por lo menos el estado por defecto con algún evento. El cambio de eventos desencadena a que los objetos se muevan, escuchen, hablen, funcionen como un vehículo o cambien sus parámetros como color, tamaño o posición.

2.1.2. Machinima

Machinima es una técnica que utiliza mundos virtuales para crear piezas audiovisuales en tiempo real y con técnicas de grabación efectuadas en el mismo mundo donde los actores o eventos son controlados por humanos, scripts

o elementos con inteligencia artificial. [11]. Esta técnica captura cierto punto de vista de un mundo virtual, la graba, edita y exhibe.

2.2. Workshops: Técnicas en Ingeniería de Requisitos

Este trabajo utiliza los conceptos de la técnica de Workshops ofrecidos por el libro *Requirements by Collaboration* [8] que define un workshop como “una reunión estructurada donde un grupo de involucrados con el proyecto y expertos en determinado dominio fue cuidadosamente seleccionado para definir, crear, redefinir y concordar en entregables que representen los requisitos de los usuarios”.

Workshops para la captura de requisitos, es un variante de la técnica de IBM denominada Joint Application Design (JAD), también conocida como Joint Application Development [12]. La propuesta del libro [8] es ofrecer una guía para planificar, diseñar y ejecutar workshops colaborativamente.

En este texto aparecerá la palabra Workshops con la primera letra en mayúscula cada vez que se haga referencia a esta técnica y en minúscula cuando puede ser aplicado a cualquier tipo de workshop.

Los ingredientes de un workshop

Para realizar un workshop, Gottesdiener [8] ofrece los siguientes “ingredientes” (propuesta de mejores prácticas en Workshops) para manejar los aspectos impredecibles de este tipo de reuniones para los participantes.

1. Propósito compartido. –El propósito del workshop debe ser definido y comunicado a los participantes desde el inicio de tal forma que todos tengan conocimiento del mismo.

2. Personas adecuadas. –Las personas adecuadas y en el momento correcto deben ser reunidas para realizar el workshop.

3. Ambiente común. – Los participantes podrían estar en el mismo ambiente o en ambientes distribuidos y las actividades podrían realizarse de manera síncrona o asíncrona. Debe crearse un ambiente apropiado con todas las herramientas necesarias para realizar las actividades del workshop.

4. Grupos variados. - Se debe considerar la diferencia de conocimiento, experiencia y personalidad que existen en los grupos de clientes, usuarios y desarrolladores de software. Los facilitadores deberán reconocer las fortalezas y debilidades de cada uno de ellos.

5. Trabajo Previo. – Un workshop es un proceso que requiere de entradas. Este material debe ser preparado con anticipación para que el workshop no tenga retrasos y las actividades programadas ocurran adecuadamente.

6. Preguntas dirigidas. – El flujo de trabajo del grupo es dirigido por las preguntas. Deberán definirse las preguntas adecuadas para el momento adecuado.

7. Juego serio. – Se podrán programar actividades recreacionales para que los participantes contribuyan a encontrar requisitos incompletos o incorrectos.

8. Confianza. – Se deberá reforzar la seguridad entre los participantes y la confianza mutua.

9. Proceso variado. – Dependiendo de las actividades a realizar, éstas podrán ser distribuidas individualmente, en subgrupos o para todo el grupo.

10. Prueba de aceptación. – Se deberán establecer un grupo de criterios establecidos para aceptar el entregable.

11. Cierre colaborativo. – Se deberá seleccionar el tipo de participación para tomar decisiones que sean congruentes con el objetivo del workshop y que satisfagan las ideas de los participantes.

12. Estructura flexible. – Se debe tomar en cuenta que las actividades planeadas pueden cambiar o pueden ser ajustadas.

13. Dos lados del cerebro. – Se debe tomar en cuenta que los participantes son todos diferentes y las capacidades deben ser explotadas. El lado derecho de los cerebros es mas visual lo que contribuye con la intuición mientras que el lado izquierdo es más lineal y analítico lo que contribuye con la exactitud.

14. Retroalimentación frecuente. – Si algún punto en el workshop no fue claro o si ocurre algún problema, la estructura del workshop deberá ser lo más flexible posible para retroceder y revisar este punto.

2.3.Técnicas para el desarrollo del juego

El desarrollo del juego combina técnicas de desarrollo de software y juegos. Estas técnicas son: Proceso de prototipación para definir la metodología a seguir, técnicas para el diseño del juego (Branching Stories, escenarios y diagramas UML).

2.3.1. Proceso de prototipación

La creación de este juego utilizará un proceso iterativo e incremental. La Figura 2 muestra el proceso utilizado basado en el libro *Effective Prototyping for Software Makers* [9].

Fase 1	Planificar	Paso 1. Verificar los requisitos Paso 2. Desarrollar el flujo de tareas Paso 3. Definir el contenido y la fidelidad
Fase 2	Especificación	Paso 4. Determinar las características Paso 5. Escoger un método Paso 6. Escoger las herramientas
Fase 3	Diseño	Paso 7. Seleccionar el criterio de diseño Paso 8. Crear el diseño
Fase 4	Resultados	Paso 9. Revisar el diseño Paso 10. Validar el diseño Paso 11. Implantar el prototipo

Figura 2 The Effective Prototyping Process de [9].

Los 11 pasos del proceso de prototipación están agrupados en 4 fases: Planificar, especificar, diseñar y resultados. En la fase de planificar se determinan las necesidades de los prototipos y a organizar el proceso de prototipación para luego determinar qué aspectos del software serán o no prototipados. En la segunda fase, se especifican los elementos necesarios (características, método y herramientas) para cubrir las decisiones tomadas en la fase de planificación. La fase de Diseñar es la ejecución del prototipo utilizando criterios de diseño que se deben definir. En la última fase, se obtienen los resultados desde la revisión, validación e implantación del prototipo.

2.3.2. Branching Stories

Aldrich [5] identificó cuatro géneros tradicionales para la simulación de juegos: Branching Stories, modelos basados en juegos, hoja de cálculo interactiva y laboratorios/productos virtuales. Este trabajo utilizará el género de Branching Stories para generar diversas situaciones donde el estudiante deberá

seleccionar el camino a seguir. La Figura 3 muestra la topología de una estructura de Branching Stories.

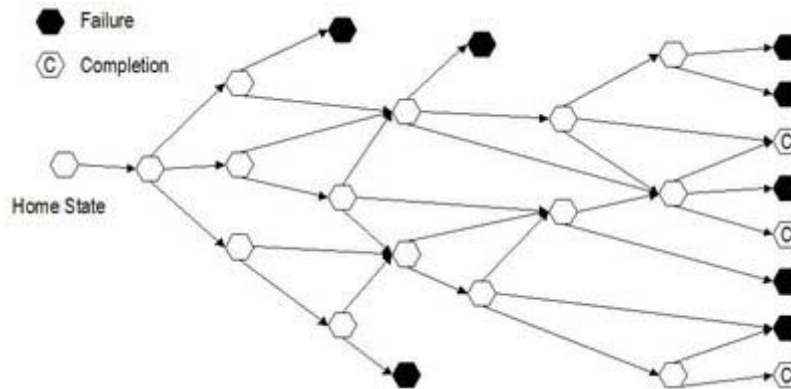


Figura 3 Estructura de Branching Stories de [5].

Los caminos de un juego en un grafo de Branching Stories son presentados en forma de preguntas con varias alternativas de respuesta que guiarán al estudiante a nuevos caminos o rutas cerradas, dependiendo de las decisiones tomadas. Este género tiene un impacto en la decisión tomada lo que permitirá al estudiante encontrar el camino correcto. Si el camino seleccionado no fue el adecuado, no podrá completar el camino u obtendrá los resultados de sus acciones.

2.3.3. Escenarios

Escenarios es una técnica de especificación de requisitos usada para describir situaciones con énfasis en el comportamiento del ambiente. La Figura 4 muestra la instanciación de uno de los escenarios de TREG utilizando la plantilla basada en [13].

La técnica de Escenarios [13] es usada para especificar el juego. Las excepciones de la plantilla guían los caminos que puede seguir el jugador y mantienen los caminos definidos en el grafo de Branching Stories.

Scenario	
ID:	WS001
Title:	Right People Context
Goal:	Define the initial state of the Right People Ingredient in the workshop technique. The trainee must select the Right People for the workshop.
Context:	A new session will start. The trainee role plays as the Workshop Facilitator, 3 participants are seated around the table and 5 are stand in the environment.
Resources:	Boards with descriptions, table and 7 chairs.
Actors:	Trainee, Non Player Characters (NPCs)
Episodes:	<ol style="list-style-type: none"> 1. There is described the project case: <ul style="list-style-type: none"> o Company Pookysoft sells computers and accessories in the country. o Project purpose: The TREG system will administrate: the stock, the requests (on line and at the store), the product tracking and the customer registration. o Workshop purpose: Identify the primary use cases focus on the sales module. 2. The roles and NPCs characteristics are exposed: <ul style="list-style-type: none"> o 3 NPCs are seated around the table. These NPCs will participate in the meeting. The Finance Manager as an Indirect User, the Requirement Analyst as a content participant and a Administrative Assistant as the Recorder. There is a chair reserved for the Facilitator (trainee). o 5 NPCs are located in the “Making Workshop” room. The characteristics of these NPCs are presented in Table 1. o The trainee must select 1 of the 4 arrangements of participants for sitting in the 3 free chairs for the workshop: <ol style="list-style-type: none"> a) Project Sponsor, Direct User 1, Direct User 2 b) Project Sponsor, Direct User 1, Workshop Sponsor c) Direct User 1, Direct User 2, Direct User 3 d) Direct User 1, Direct User 2, Workshop Sponsor 3. The trainee selects the option c. 4. When 2 go to WS002
Exceptions	<p>Go to WS003 when student selects option a.</p> <p>Go to WS004 when student selects option b.</p> <p>Go to WS005 when student selects option d.</p> <p>Otherwise go to WS001</p>

Figura 4 Escenario del Ingrediente Personas adecuadas.

2.3.4. Diagramas de clases

Un diagrama de Clases representa las clases que serán utilizadas dentro del software, las relaciones que existen entre ellas, y sus operaciones y atributos. Los diagramas de Clases por definición son estáticos, esto es,

representan que partes interactúan entre sí, no lo que ocurre en un determinado momento. Estos diagramas son comúnmente usados para [14]:

- Explorar conceptos en forma de un modelo de dominio.
- Analizar los requisitos en forma de un modelo conceptual o de análisis.
- Describir detalladamente un diseño orientado a objetos.

2.3.5. Diagramas de comunicación

Un diagrama de comunicación, inicialmente llamado un diagrama de colaboración, es un diagrama de interacción que muestra información similar a los diagramas de secuencia pero su foco principal es en la relación de objetos. [14]. Los diagramas de comunicación representan una combinación de información tomada desde el diagrama de clases, secuencia, y diagrama de casos de uso describiendo tanto la estructura estática como el comportamiento dinámico de un sistema. Este diagrama modela el flujo de mensajes entre los objetos o partes de ellos. Los objetos se relacionan con conectores de asociación entre ellos. Los mensajes se agregan a las asociaciones y se muestran como flechas cortas apuntando en la dirección del flujo del mensaje. La secuencia de los mensajes se muestra a través de un esquema enumerado [14].

2.3.6. Maquinas de estados

Una máquina de estados define el comportamiento dinámico de objetos, casos de uso y sistemas completos. Un diagrama de máquinas de estados es una representación gráfica de los posibles estados de un objeto, los eventos que causan la transición entre uno y otro estado y las acciones desencadenadas por el cambio de estados [14].

Un estado representa una etapa en el comportamiento patrón de un objeto. El estado inicial del objeto, también conocido como estado de creación, ocurre cuando es creado, por otro lado el estado final determina el fin de las transiciones en el objeto. Una transición es un cambio de un estado a otro y es desencadenada por un evento.

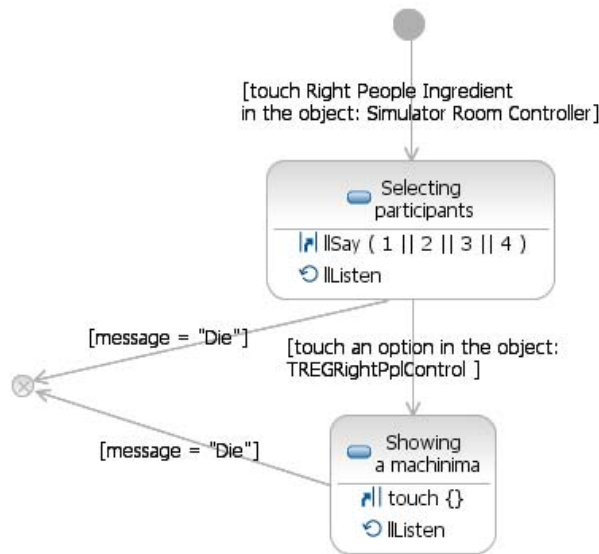


Figura 5 Diagrama de estados de uno de los elementos de TREG

La Figura 5 muestra el diagrama de estados de uno de los elementos de TREG usando la notación UML. Las fechas que llevan de un estado a otro toman el nombre de transiciones y en la figura se muestran sus etiquetas (entre corchetes) que son las condiciones que llevan al cambio de evento. Asimismo, en esta figura cada estado está representado un rectángulo con los bordes redondeados y posee el nombre del estado y las acciones que realiza el estado cuando es invocado.

2.4. Trabajos relacionados

Los juegos son una propuesta no tradicional para el entrenamiento en ingeniería de requisitos [15]. Hay algunos ejemplos en este campo pero ninguno desarrollado en un mundo virtual. RE-O-Poly explora las mejores prácticas en Ingeniería de Requisitos usando un tablero basado en el clásico juego de Monopolio [16]. El proyecto FAB ATM desarrollado por la Universidad de Deakin, combina simulaciones con actividades realizadas en clases para la enseñanza de principios de IT/IS e Ingeniería de Requisitos [17].

Los mundos virtuales son un ambiente efectivo para juegos educativos [18]. Second Life posibilita la creación de contenidos educativos como clases, paneles de discusión y juegos [4]. Ohio University desarrollo dos juegos multiusuarios en Second Life para la educación en Ingeniería de Software [19]. El primero se basó en un ejercicio denominado *Groupthink Software Specification Exercise* [20] que tiene como objetivo enseñar cómo escribir

especificaciones correctamente. La Figura 6-A muestra una experiencia de una sesión de este ejercicio en Second Life. Este juego está disponible para usar en la isla de esta universidad [21]. El otro juego fue desarrollado tomando como base el juego SimSE donde los estudiantes simulan la gestión de un proyecto de software [22]. Este juego se encuentra disponible en [23].



Figura 6 A Groupthink game (A). Hospital Scenario en Play2Train (B)

El entrenamiento a través de juegos se muestra como un camino para mejorar las habilidades de las personas de una forma entretenida. Actualmente, empresas y universidades están en nuevas formas de entrenamiento en Second Life. De este modo, Idaho State University desarrolló juegos basados en simulaciones para entrenamiento. Un juego para entrenamiento en larga escala fue creado para simular los principales ejercicios del cuidado de la salud y preparación para emergencias. [24]. La Figura 6 -B muestra el escenario de un hospital donde el ejercicio de la pandemia de fiebre fue simulado [25]. IBM desarrolló una serie de juegos llamada *Open Encounters of z Virtual Kind for challenging skills in IBM technologies and Open Source*. Estos juegos intentan mejorar las habilidades de los usuarios de herramientas de IBM como IBM System z mainframe, Service Oriented Architecture (SOA), the Cell/B.E. processor, Grid computing, Linux, Java, entre otras tecnologías [26].