

## 4 Prototipos

El desarrollo de TREG siguió el proceso de prototipación descrito anteriormente y utilizó la estrategia de ensayo y error para refinar los prototipos [32]. 5 iteraciones fueron necesarias para redefinir los requisitos y prototipar los cuatro ingredientes de la técnica de Workshops. Second Life fue utilizada como la principal herramienta de prototipación. Esta sección relata nuestra experiencia con el proceso de prototipación del juego, y muestra las características de cada iteración, los criterios de diseño utilizados y la evolución del producto usando métodos de prototipación a lo largo de las iteraciones.

### 4.1. Proceso de prototipación

El proceso de prototipación *Effective Prototyping Process* [9] fue adaptado para el desarrollo del juego en Second Life. Los pasos de cada una de las fases fueron realizados dependiendo del método de prototipación usado en cada iteración.

#### 4.1.1. Fase 1. Planificar

En esta fase, se deciden qué aspectos del software se prototipará y que no, validando suposiciones y requisitos.

##### **Paso 1. Verificar Requisitos**

Los requisitos del juego fueron descubiertos a partir de suposiciones, tomando en cuenta la audiencia del juego y sus objetivos. A lo largo del desarrollo de TREG, estos requisitos fueron reunidos y priorizados usando el grafo de Branching Stories, teniendo de esta forma una visión general de los requisitos a verificar.

##### **Paso 2. Desarrollar el flujo de tareas**

El flujo de tareas describe los pasos que el jugador tiene que ser para completar todas las actividades mientras navega por el juego. En este paso, fueron definidas las tareas y la narración de los escenarios.

El género de Branching Stories fue usado para mapear los escenarios del juego. El grafo de Branching Stories da una visión general del juego y guía la interacción de los diferentes escenarios. La plantilla de Escenarios fue usada para obtener un mayor detalle de la especificación del juego.

### **Paso 3. Definir el contenido y fidelidad**

En cada iteración fue definido el contenido a ser prototipado, dependiendo de la iteración desde un contenido a alto nivel que facilitó la definición del alcance del proyecto hasta el contenido detallado con la información educacional y con información que define el comportamiento del juego.

Se decidió que Second Life sería la principal herramienta de prototipación. Por esta razón, la fidelidad necesaria para la mayoría de prototipos fue alta.

## **4.1.2. Fase 2: Especificar**

Esta segunda fase cubre los resultados de las decisiones tomadas en los tres primeros pasos, donde las decisiones favorecen la determinación de las características, métodos y herramientas a utilizar en el desarrollo del juego.

### **Paso 4. Determinar las características**

Arnowitz [9] propone 8 características a ser determinadas antes de seleccionar el método de prototipación a utilizar. Estas características son: audiencia, etapa, velocidad, duración, expresión, estilo y medio. En la sección 4.2, se detallan las características que fueron determinadas para cada iteración.

### **Paso 5. Seleccionar un método**

Arnowitz [9] propone los siguientes métodos para prototipar: card sorting, wireframe, storyboard, paper, digital, blank model, video, Wizard-of-Oz y code prototype. Analizando las características, fueron definidos los prototipos para las 5 iteraciones: Quick Wireframe, Wireframe, Low-coded, High-coded.

### **Paso 6. Elija una herramienta de prototipación**

A pesar de que fue decidido, desde el inicio del proyecto, que Second Life sería la herramienta de prototipación, fueron necesarias otras herramientas para aclarar conceptos o especificar funciones como herramientas de ofimática y herramientas CASE.

## **4.1.3. Fase 3. Diseñar**

Esta fase se centra en la ejecución del prototipo a partir de la planificación y especificación realizadas en las fases anteriores.

### **Paso 7. Seleccionar el criterio de diseño**

Si bien es cierto, Second Life promueve la libre creación de contenido, hay restricciones que se deben tomar en cuenta para modelar objetos y para utilizar el lenguaje de programación. El proceso de prototipación seguido ofrece guías para el diseño de páginas webs. Al haberse utilizado un ambiente virtual, se crearon criterios propios a seguir. La sección 4.3 muestra los criterios de diseño utilizados en TREG.

### **Paso 8. Crear el diseño**

En este paso fueron aplicados todos los criterios de diseño en el prototipo. En juegos es importante priorizar los elementos a desarrollar. TREG usó una estrategia “top-down” que primero lista los principales elementos del diseño y luego se va profundizando en los detalles de cada uno de los elementos.

#### **4.1.4.Fase 4. Resultados**

En esta fase el prototipo es evaluado por una audiencia interna, se realiza la validación del prototipo y se realiza la implantación o iteración.

### **Paso 9. Revisar el diseño**

El prototipo de TREG fue revisado por una audiencia interna formada por los investigadores involucrados en este proyecto. En este caso, uno de ellos es un experto en el contenido de TREG, en procesos de desarrollo de software.

### **Paso 10. Validar el diseño**

Después de la revisión del diseño es necesario validar el prototipo con stakeholders externos. Estas validaciones se realizaban principalmente con expertos en Second Life. En la última iteración, se realizaron pruebas de usabilidad donde todos los participantes estaban familiarizados con el desarrollo de software y además, correspondían a uno de estos 3 perfiles: expertos en Second Life, participantes que conocen otro mundo virtual pero no Second Life y participantes que nunca tuvieron contacto con mundos virtuales. En la sección 5 se muestra en detalle el proceso seguido en estas pruebas.

### **Paso 11. Implantar el diseño**

El ambiente de desarrollo de TREG fue un sandbox localizado en la isla de la PUC-Rio en Second Life. Sandbox responde a un término usado en Second Life donde es creada un área especial para la creación y de difícil acceso para un avatar (normalmente a más de 1000 metros sobre el terreno). Posteriormente, los prototipos validados son implantados en el área de producción.

## 4.2. Características de los prototipos de TREG

La selección del método de prototipación fue dada por las características del prototipo en cada iteración. Estas características fueron basadas en el proceso *Effective Prototyping Process* [9]. La Tabla 3 muestra las características y el prototipo seleccionado para cada iteración.

Tabla 3. Iteraciones y sus características

	Iteración 1	Iteración 2	Iteración 3	Iteración 4	Iteración 5
<b>Audiencia</b>	Interna	Interna	Interna	Interna	Externa
<b>Etapas</b>	Inicial	Inicial	Intermedia	Última	Última
<b>Velocidad</b>	Rápida	Rápida	Paulatina	Paulatina	Paulatina
<b>Duración</b>	Corta	Media	Media	Larga	Larga
<b>Expresión</b>	Conceptual	Experimental	Experimental	Experimental	Experimental
<b>Estilo</b>	Narrativo	Narrativo	Interactivo	Interactivo	Interactivo
<b>Medio</b>	Digital	Digital	Digital	Digital	Digital
<b>Fidelidad</b>	Baja	Media	Alta	Alta	Alta
<b>Prototipo</b>	<i>Quick Wireframe</i>	<i>Wireframe</i>	<i>Low-coded</i>	<i>High-coded</i>	<i>Versión del Prototipo</i>

- Audiencia: Interna/ Externa. Es importante determinar que la audiencia que va a ver o usar el prototipo, no necesariamente es el usuario final del software. En las iteraciones de TREG, la audiencia principalmente fue interna, formada por los investigadores que participaron de las revisiones de los prototipos. Excepto la última iteración, donde la audiencia externa fue formada por potenciales usuarios del juego.
- Etapas: Inicial/Intermediaria/Última. No es suficiente que se prototipe pensando en ahorrar los costos de crear un sistema. El acto de prototipar debe ser realizado de forma efectiva en relación al tiempo y costo y, principalmente, en tener un prototipo de utilidad para el proceso de desarrollo de software. Es por esto que este proceso puede ser realizado en cualquier etapa del ciclo de desarrollo. En TREG, cada una de las iteraciones del proceso de prototipación corresponden a cada iteración del ciclo de desarrollo del software.
- Velocidad: Rápida/Paulatina. Indica que tan rápido el prototipo será producido y el nivel de detalle que contendrá. En TREG, las primeras iteraciones se realizaron con más rapidez para obtener una retroalimentación mas continua. Las últimas iteraciones fueron más

paulatinas por el tiempo invertido en la codificación y en un prototipo con mayor fidelidad y calidad de los detalles visuales.

- Duración: Corta/Media/Larga. Esta característica está referida a dos aspectos:
  - ¿Cuál es el tiempo de vida del prototipo? ¿Será un prototipo que sólo se usará una vez para probar algún concepto? ¿o será iterado varias veces e involucra un diseño final del software?
  - ¿Cuál es el alcance que el prototipo puede influir en el producto? ¿Será un corto sketch para mostrar una idea o será un punto de referencia para la creación del producto?

En TREG, la duración fue corta en el primer prototipo dado que fue usado para clarificar el objetivo del proyecto y entender porque algunos entornos no serían utilizados. Las siguientes iteraciones se desarrollaron con una duración mayor y la mayoría de objetos y scripts continuaron siendo usados en el proyecto.

- Expresión: Conceptual/Experimental. La expresión es demostrada a través de qué tan tangible o abstracto puede ser el diseño visual del producto y la interacción con el prototipo. Los prototipos conceptuales se enfocan más en el entendimiento de un concepto o de una faceta particular del producto, como por ejemplo, la primera iteración de TREG fue prototipada para entender el objetivo del proyecto usando slides y una maqueta simple y no está destinada a la interacción con los objetos como fueron las otras interacciones que estaban más orientadas al “look and feel” del producto.
- Estilo: Narrativo/Interactivo. Esta característica define si un prototipo es activo y pasivo. En el estilo narrativo existe un camino predeterminado para mostrar el prototipo. A partir de los prototipos low-coded de TREG, el estilo interactivo estaba presente, dejando a los participantes interactuar libremente con el juego.
- Medio: Físico/Digital. Usando un medio físico (usualmente prototipos tangibles) se logra una mayor flexibilidad en la interacción pero la fidelidad que se tiene con el sistema es mucho menor a usar un medio digital. En TREG se usó sólo medios digitales, principalmente utilizando Second Life como herramienta.
- Fidelidad: Baja/Media/Alta. Esta característica está referida al contenido del prototipo que puede ser: diseño de la información,

diseño de la interacción y modelo de navegación, diseño visual, contenido editorial, expresión de la marca y desempeño y comportamiento del sistema. Cada una de estas características desempeña una función diferente y no todas aplican a todos los productos. En TREG, se enfatizó en el contenido de alta fidelidad usando Second Life (principalmente para los prototipos codificados).

### 4.3. Criterios de Diseño

Los criterios de diseño tienen como objetivo guiar el diseño, construcción y presentación de los prototipos. El diseño de TREG fue realizado usando los criterios descritos a continuación para resolver reglas contradictorias al modelar objetos 3D o programar en Second Life.

#### Memoria del usuario

No sobrecargar la memoria del usuario. En este sentido, los objetos con la misma funcionalidad deberán lucir iguales:

- Tanto Miss Workshop que se encuentra en el área de recepción como la que se encuentra en la cocina deberán tener el mismo aspecto para que el jugador pueda reconocerla.
- En el Panel de Simulación, las respuestas para los ingredientes de Personas Adecuadas y Propósito Compartido deberán aparecer en la misma posición y bajo el mismo formato de texturas.
- Asimismo, la información que aparece en “Learn More” deberá aparecer en el mismo formato de textura para todos los ingredientes.
- Los elementos de los ingredientes de Trabajo Previo y Espacio Compartido deben funcionar de la misma manera. Es decir, cuando se haga clic en uno de ellos cambie a un color diferente que sea identificable por el jugador.
- La retroalimentación de las respuestas correctas en el ingrediente de Trabajo Previo, deberá cambiar de al mismo color mostrando las respuestas correctas en los tres paneles.
- Second Life facilita la visualización de objetos que tienen funcionalidad, al hacer que el cursor cambie al pasar el mouse por dichos objetos.

- La funcionalidad deberá ser la misma para abrir y cerrar las puertas del túnel conector de la cocina al cuarto de simulación.

### **Patrones de diseño visual**

Los patrones de diseño dan uniformidad a los objetos del prototipo al aplicar un diseño consistente en los objetos.

- Se debe utilizar un mismo color para los nombres de las etiquetas de los NPCs.
- Dado que las texturas a ser utilizadas en los cuadros de los ingredientes de Trabajo Previo y Espacio Compartido son transparentes, todo el ambiente deberá cambiar para blanco de forma tal que haga contraste con los elementos en el ambiente y puedan ser visualizados por el jugador sin problemas.
- Las texturas que contienen letras deberán utilizar los mismos tonos (tener en cuenta el fondo negro, letras blancas y líneas celestes).
- Deberá utilizarse la misma textura panorámica para la visualización externa de la cocina y del cuarto de simulación.

### **Uso de Second Life**

Se utilizarán las funcionalidades ya incluidas en el cliente de Second Life. Asimismo, deberá tenerse en cuenta las siguientes consideraciones de implementación en Second Life.

- La comunicación entre los objetos de TREG y el jugador se dará por medio del Local Chat de Second Life. Se utilizará el canal 0 para enviar mensajes públicos al jugador.
- Se utilizará la funcionalidad de enviar objetos en Second Life. Se toma como premisa que en estos casos, el jugador guarda los objetos en su inventario y sabe como manipularlos.
- Dado que una isla en Second Life tiene como parámetro máximo el uso de 15 000 prims, se utilizará la función `llRezObject` para hacer aparecer los objetos en el ambiente. Cuando el jugador termina de utilizar un ingrediente, estos objetos son eliminados y aparecen los objetos del próximo ingrediente que desea simular. Con esta característica se reduce el número de prims en la isla, usando sólo los prims que se necesitan por cada ingrediente.

- Los notecards de Second Life serán usados para enviar información al jugador con contenido educacional de los ingredientes de Workshops.
- Cuando sea posible, se utilizará notecards con el contenido de la comunicación entre objetos y avatares de tal forma que permita algún futuro cambio de idioma.
- La plataforma de desarrollo usando el sandbox se mantendrá aislada del ambiente de producción.

### Desempeño del juego

El bajo desempeño computacional que puede existir en el juego es perjudicial para la inmersión del jugador ya que pocos segundos pueden hacer mucha diferencia. Al tener un juego en Second Life, el desempeño depende también de la congestión de los servidores de Linden Labs, la velocidad de conexión del jugador y la configuración técnica de la CPU y tarjeta de video que está usando. Para esto, se debe tomar en cuenta las siguientes recomendaciones:

- La comunicación entre objetos vinculados utilizará la función `llMessageLinked` para enviar algún mensaje a los otros objetos vinculados. Además usará el evento `link_message` para recibir el mensaje. Estas funciones trabajan más rápido que las funciones de eventos `listen` que deberán ser usadas para la comunicación sólo de objetos no vinculados.
- Se deberá minimizar los eventos `listen` en el canal 0. La comunicación entre objetos no vinculados utilizará la función `llSay` para enviar algún mensaje a los otros objetos del ambiente. Además, usará el evento `listen` para recibir el mensaje.
- Para dar mayor seguridad a la comunicación entre objetos, el canal de comunicación entre objetos no vinculados será un número entero negativo de 5 dígitos generado aleatoriamente. Se deberá tener en cuenta la performance e el retraso de los eventos en LSL. La Tabla 4 muestra las funciones que ocasionan mayor retraso.



Tabla 4. Funciones con mayor retraso (en segundos)

Retraso	Función
20.0	llRefreshPrimURL
20.0	llSetPrimURL
20.0	llEmail
10.0	llLoadURL
10.0	llGetSimulatorHostname
5.0	llTeleportAgentHome
3.0	llSendRemoteData
3.0	llRemoteLoadScriptPin
3.0	llRemoteLoadScript
3.0	llRemoteDataReply
3.0	llGiveInventoryList
2.0	llSetParcelMusicURL.
2.0	llParcelMediaQuery
2.0	llParcelMediaCommandList
2.0	llInstantMessage
2.0	llGetParcelPrimOwners

- El desempeño del juego se puede ver muy afectado por el uso de muchas texturas diferentes en el ambiente puesto que cada una de ellas deberá ser descargada al cliente del jugador. Se deberá minimizar el uso texturas. Por ejemplo, en el panel de control, se deberá usar sólo una textura para todas las respuestas, objetos transparentes se posicionaron sobre las respuestas con la funcionalidad deseada.

#### 4.4. Iteraciones

La secuencia de pasos relatada en la Sección 4.1 se siguió en cada una de las iteraciones del desarrollo. De esta manera, se fue refinando los requisitos, definiendo los objetos necesarios e integrando la solución. La audiencia que participó de las sesiones de revisión y validación del diseño de los prototipos fueron el autor, el orientador y co-orientador de este trabajo. Adicionalmente, en la última iteración, las pruebas de usabilidad fueron realizadas con 2 evaluadores y 9 participantes.

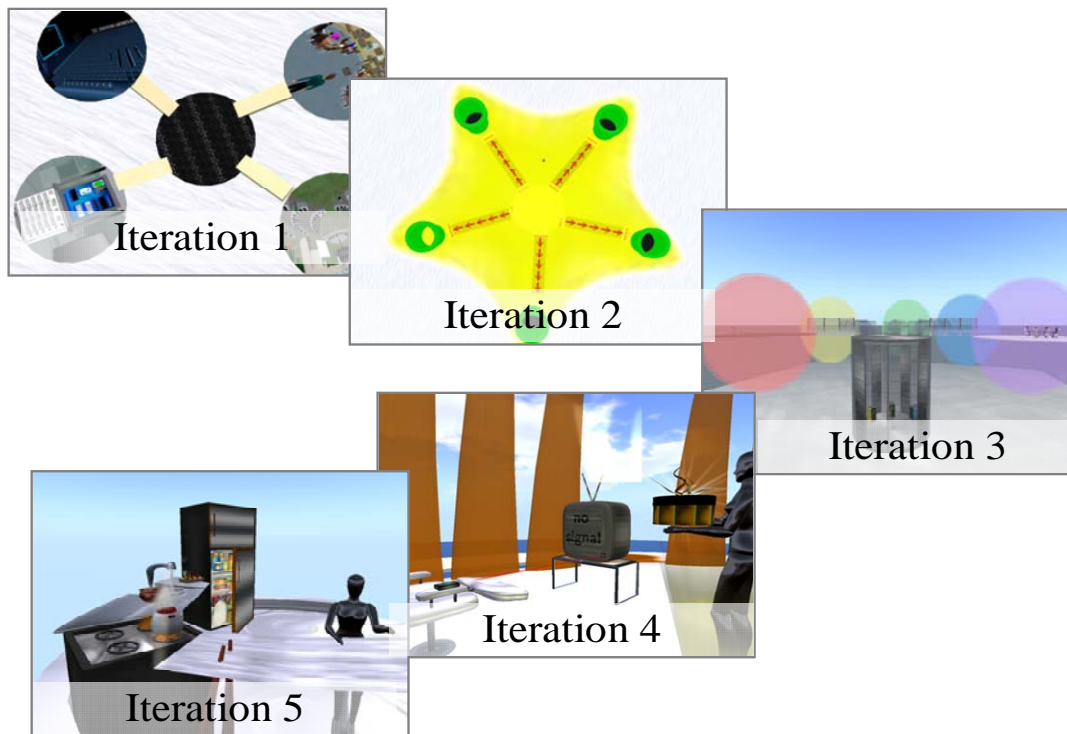


Figura 20. Prototipos desarrollados en TREG

La Figura 20 muestra la transición de la apariencia de TREG por estas iteraciones. Se puede observar como el proceso de prototipación favoreció el refinamiento del prototipo desde una maqueta de baja fidelidad hasta la versión final del juego.

#### 4.4.1. Iteración 1.

*De una Idea hacia un prototipo Quick Wireframe.*

Al inicio del proceso de prototipación, la idea principal fue la creación de un ambiente para el entrenamiento en Ingeniería de Requisitos. A partir de esta idea se creó un prototipo Quick Wireframe. Este método de prototipación da una aproximación anticipada de la idea del software pudiendo tener una perspectiva a alto nivel enfocada en los aspectos estructurales de la aplicación [9]. En esta primera iteración, se utilizó este método para presentar una maqueta de baja fidelidad creada en Second Life mostrando la idea de estructuras con los diferentes objetivos para el entrenamiento. Se propuso tener un área de recepción para conocer el proyecto como punto de partida para otras las áreas del proyecto, juegos mono usuario donde exista un entrenamiento en las principales técnicas de Ingeniería de Requisitos, un juego tipo laberinto donde el

jugador abriría puertas hasta llegar al punto final de un determinado entrenamiento, salones para clases o grupos de discusión y un auditorio.

Se realizó una maqueta en Second Life donde cada una de estas áreas representadas con un óvalo con una textura tal como se muestra en Figura 21. En el medio de la maqueta representaba al área de recepción que unía a las demás áreas.

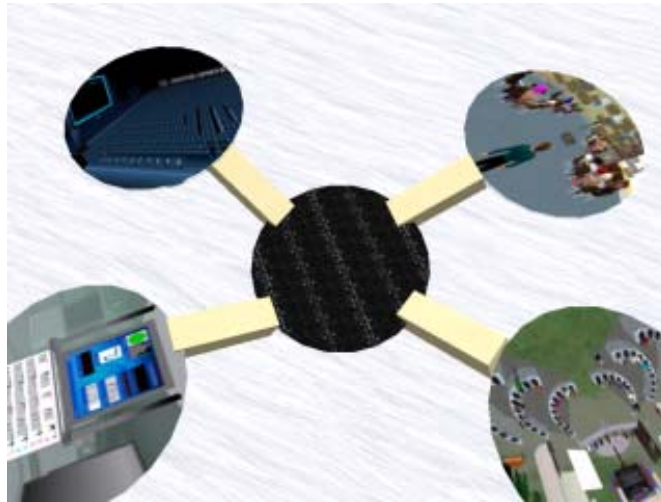


Figura 21. Prototipo de la Iteración 1.

Al revisar la maqueta, ésta mostraba sólo texturas para dar una idea general del área y no mostraba los objetos en 3D que se iban a crear pero sirvió para formular el propósito del proyecto: un juego no tradicional para el entrenamiento en Ingeniería de Requisitos. Este prototipo fue un instrumento para clarificar el objetivo real del proyecto.

Se acordó que los otros prototipos que utilicen como herramienta Second Life, se construyan los objetos 3D para poder tener una visión más próxima al producto.

#### 4.4.2. Iteración 2.

##### *De un prototipo Quick Wireframe hacia un prototipo Wireframe*

En esta iteración, se utilizó un prototipo Wireframe con mayor fidelidad que el anterior construyéndose objetos 3D en Second Life. El prototipo Wireframe mostraba el área de recepción con 5 teleports para las áreas tentativas. En la Figura 22, se muestra el área de recepción de la iteración 2. En esta se construyeron teleports (objetos en verde) para las áreas del juego en cada una de las puntas de la estrella. Estas áreas tentativas para el desarrollo del juego

basadas en conceptos de Ingeniería de Requisitos fueron: proceso, técnicas, gestión, análisis y validación.

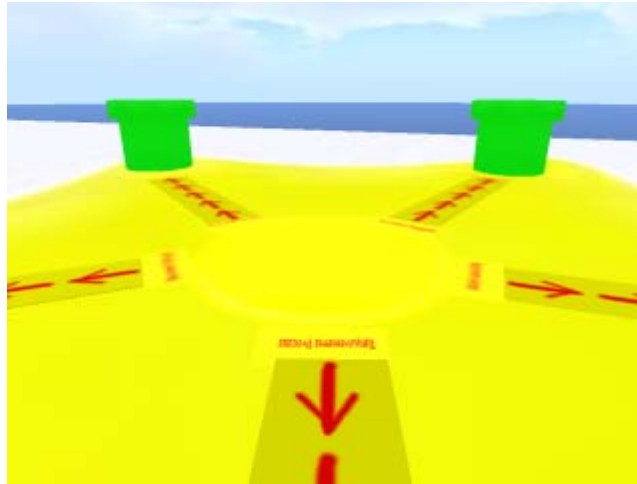


Figura 22. Prototipo de la Iteración 2.

Este prototipo contribuyó a explorar la idea planteada en la iteración anterior. De este modo, se decidió comenzar el estudio con la técnica de Workshops. Además, se mantuvo la idea de utilizar teleports para las áreas del juego.

#### 4.4.3. Iteración 3.

*De un prototipo Wireframe hacia un prototipo Low-coded*

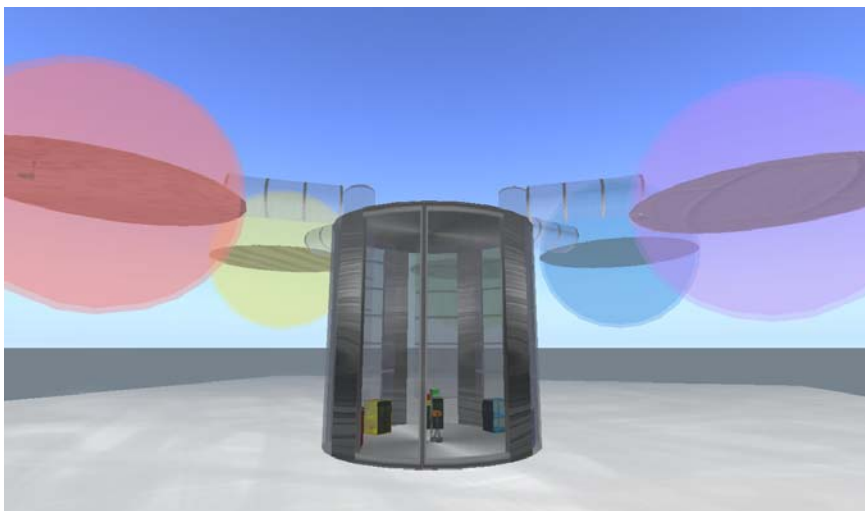


Figura 23. Prototipo de la Iteración 3.

Los prototipos codificados muestran una alta fidelidad del producto y son interactivos lo cual posibilita refinamientos incrementales en el diseño pero a su

vez están vinculados al riesgo del rechazo total o de grandes modificaciones [9]. En esta iteración se decidió comenzar por desarrollar un prototipo Low-coded, donde existe una gran fidelidad del producto pero no involucra un alto riesgo de codificación.

Una nueva área de recepción mostraba un edificio con elevadores para las diversas áreas tal como se muestra en la Figura 24. Los scripts de los teleports a las áreas fueron reutilizados en los elevadores. Se comenzó a prototipar el área de Workshops. En el primer incremento de esta iteración, texturas que representaban a participantes del workshop fueron mostradas y se decidió cambiar estas texturas por objetos 3D que simulasen estos participantes (NPCs). En el siguiente incremento, se mostraron los NPCs que formarían parte del ingrediente de Personas Adecuadas. Inclusive se mostró un NPC que caminaba en el ambiente pero se descartó la utilización de esta funcionalidad en los NPCs debido a los retrasos que podría causar en computadoras con bajo rendimiento.

Se vio la posibilidad de incluir en el área de recepción información del juego. En esta iteración se descubrió la necesidad de usar un género de simulaciones para juegos (Branching Stories). Asimismo, se decidió usar la técnica de Escenarios para especificar el juego tal como se mostró en el documento [33].

#### 4.4.4. Iteración 4.

*De un prototipo Low-coded hacia un prototipo High-coded*

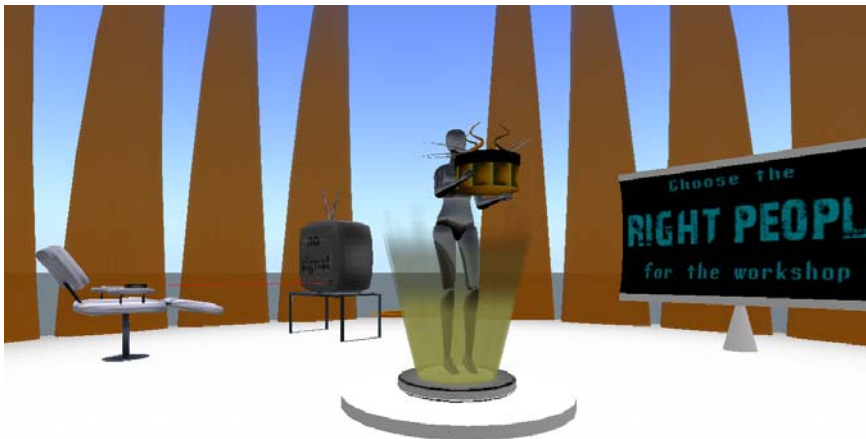


Figura 24. Prototipo de la Iteración 4.

Esta iteración fue la de mayor duración para la programación de los elementos. En esta iteración se incluyeron los acuerdos de implementación de la iteración anterior y se terminó de programar los ingredientes de TREG.

Una nueva área de recepción con mayor fidelidad fue redefinida tal como se muestra en la Figura 24. Esta área incluyó al NPC para guiar al jugador y para ofrecer los recursos necesarios para el juego. Asimismo, el ambiente de la metáfora de la cocina fue modelado e programado. EL HUD que incluye la puntuación y el número de ingredientes capturados fue integrado al ambiente de simulación y a la cocina.

Las simulaciones fueron definidas usando Branching Stories y escenarios. Una nueva opción atractiva surgió con la inclusión de maquinas de estados para la implementación en LSL. Utilizando el diseño del software especificado en la Sección 3.3, se terminó la implementación del ingrediente de Personas Adecuadas y se prototiparon los demás ingredientes del juego.

#### 4.4.5. Iteración 5.

*De un prototipo High-coded prototype hacia la implantación de TREG.*



Figura 25. Prototipo de la Iteración 5.

En esta iteración, el equipo validó los conceptos utilizados en la prototipación del producto y se realizaron refinamientos en el diseño del juego. Posteriormente se realizaron las pruebas de usabilidad tal como se detalla en la Sección 5. Finalmente, la versión de TREG de los ingredientes de planificación de la técnica de Workshops fue implantada en el área de producción tal como se muestra en la Figura 25.