



Acta de Correcciones al Proyecto de Grado Ingeniería de Sistemas y computación

Fecha: 24 de febrero de 2022

Autores: Iván David Valderrama Corredor

Nombre del Proyecto de Grado: Técnicas de Ludificación Aplicadas a los Conceptos de Ingeniería de Líneas de Producto de Software

Director: Luisa Fernanda Rincón Pérez

Como indica el artículo 2.27 de las Directrices de Trabajo de Grado, he verificado que los estudiantes indicados arriba han implementado todas las correcciones que los Jurados del Proyecto de Grado definieron que se efectuaran, como consta en el Acta de Calificación correspondiente.

Luisa Rincón
Directora

Nota de Aceptación

Aprobado por el Comité de Trabajo de Grado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana para optar el título de Ingeniero de Sistemas y computación.



Dr. Hernán Camilo Rocha Niño
Decano de la Facultad de Ingeniería



ING. Gerardo Mauricio Sarria Montemiranda
Director Carrera Ingeniería de Sistemas y Computación.



ING. Luisa Fernanda Rincón Pérez
Director(a) Trabajo



ING. Andrés Navarro Newball
Jurado 1



ING. Juan Carlos Martínez Arias
Jurado 2

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería.
Ingeniería de Sistemas y Computación.
Proyecto de Grado.

Técnicas de Ludificación Aplicadas a los Conceptos de Ingeniería de Líneas de Producto de Software

Iván David Valderrama Corredor

Directora: Luisa Fernanda Rincón Pérez

24 de febrero de 2022



Santiago de Cali, 24 de febrero de 2022.

Señores

Pontificia Universidad Javeriana Cali.

Dr. Gerardo Mauricio Sarria

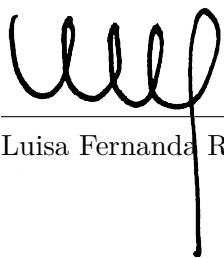
Director Carrera de Ingeniería de Sistemas y Computación.

Cali.

Cordial Saludo.

Por medio de la presente hago constar que en mi calidad de director de trabajo de grado he revisado el proyecto titulado “Técnicas de Ludificación Aplicadas a los Conceptos de Ingeniería de Líneas de Producto de Software” realizado por el estudiante de Ingeniería de Sistemas y Computación Iván David Valderrama Corredor (cod: 1887703), el cual se encuentra terminado y considero que cumple con los requisitos para ser sustentado.

Atentamente,

A handwritten signature in black ink, consisting of several loops and a long vertical stroke extending downwards from the end of the signature.

Luisa Fernanda Rincón Pérez

Santiago de Cali, 24 de febrero de 2022.

Señores

Pontificia Universidad Javeriana Cali.

Dr. Gerardo Mauricio Sarria

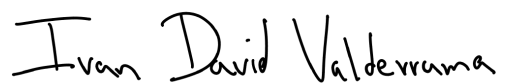
Director Carrera de Ingeniería de Sistemas y Computación.

Cali.

Cordial Saludo.

Me permito presentar a su consideración el proyecto de grado titulado “Técnicas de Ludificación Aplicadas a los Conceptos de Ingeniería de Líneas de Producto de Software” con el fin de cumplir con los requisitos exigidos por la Universidad y para que sea sometido a revisión del jurado y cumpla su aprobación, para conseguir posteriormente el título de Ingeniero de Sistemas y Computación.

Atentamente,



Iván David Valderrama Corredor

Código: 1887703

0.1. Resumen

La práctica exitosa de la ingeniería de líneas de producto de software brinda beneficios a las empresas contemporáneas, tales como la disminución en los tiempos y costos de desarrollo en sus productos de software. En la actualidad, pocas personas conocen el enfoque de líneas de producto de software y menos confianza tienen en él, por lo tanto, este trabajo propone implementar una herramienta que permita dar a conocer los conceptos introductorios de la ingeniería de líneas de producto de software, para que las personas sin experiencia previa puedan aprender estos conceptos de una manera más sencilla. Para ello, se ha implementado la metodología de ludificación en la herramienta desarrollada, permitiendo la creación de un prototipo web funcional con distintas técnicas mecánicas y dinámicas. Como resultado de este trabajo, se pudo observar resultados positivos obtenidos por medio de los usuarios inexpertos y expertos que probaron el prototipo.

Palabras Clave: Ingeniería de líneas de productos de software; ludificación; videojuego.

0.2. Abstract

The successful practice of software product line engineering provides benefits to contemporary companies, such as the reduction in development times and costs in their software products. Today, fewer people are aware of the software product line approach and have less confidence in it, therefore, this work proposes to implement a tool that allows to present the introductory concepts of software product line engineering, so that people without previous experience can learn these concepts in a simpler way. The gamification methodology has been implemented in the developed tool, allowing the creation of a functional web prototype with different mechanical and dynamic techniques. As a result of this work, it was possible to observe positive results obtained by inexperienced and expert users who tested the prototype.

Keywords: Software product line engineering; gamification; video game.

Índice general

0.1. Resumen	5
0.2. Abstract	5
1. Descripción del Problema	3
1.1. Formulación	4
1.2. Sistematización	4
1.3. Objetivos	4
1.3.1. Objetivo General	4
1.3.2. Objetivos Específicos	4
1.4. Justificación	5
1.5. Delimitaciones y Alcances	6
1.6. Entregables	6
1.7. Resultados Obtenidos y Contribuciones	6
2. Desarrollo del Proyecto	7
2.1. Áreas de Referencia	7
2.2. Marco de Referencia	7
2.2.1. Conceptos Relacionados con la Ingeniería de líneas de Producto de Software	7
2.2.2. Conceptos Relacionados con la Ludificación	11
2.2.3. Game Design Documentation Builder	14
2.2.4. Metodología de Personalización Masiva de Software - Charles W. Krueger, BigLever Software	14
2.3. Trabajos Relacionados	16
3. Diseño - Proceso de Ludificación	19
3.1. Público Objetivo	19
3.2. Grupos de Conceptos	20
3.3. Problemas que Enfrentan los Usuarios	21
3.4. Jugabilidad	21
3.4.1. Género	22
3.4.2. Sinopsis del juego	23
3.4.3. Resumen del Juego	23
3.4.4. Mecánicas del Videojuego	24
3.4.5. Progresión del Videojuego	25
3.4.6. Satisfacción de los Jugadores (Sentimientos Objetivo)	27
3.5. Selección de Técnicas Mecánicas y Dinámicas	28

4. Implementación - Prototipo Web	29
4.1. Tecnologías Planificadas	29
4.1.1. Motor Gráfico Godot Engine	29
4.1.2. Editor de Imágenes Paint.NET:	30
4.1.3. Editor de Voxel Artwork MagicaVoxel:	30
4.1.4. Editor de Sonido Audacity:	31
4.2. Servidores	31
4.3. Características Principales del Prototipo	34
4.3.1. Sistema de Diálogos Multilenguaje	34
4.3.2. Controles	35
4.3.3. Implementación de Mecánicas	36
4.3.4. Proceso de Creación de Personajes	38
4.3.5. Proceso de Animación	39
4.3.6. Proceso de Creación de Escenarios	39
4.3.7. Comparativa con la Ingeniería de Líneas de Productos de Software	40
4.3.8. Quices	43
4.3.9. Conexión de Escenas	45
4.3.10. Prototipo	46
5. Pruebas de Usuario	55
5.1. Pruebas Funcionales	55
5.1.1. Ejecución y Resultados	56
5.2. Pruebas de Usabilidad	61
5.2.1. Ejecución y Resultados	61
5.3. Pruebas de Percepción de Utilidad	62
5.3.1. Resultados de los Quices	63
6. Conclusiones	65
6.1. Objetivos Alcanzados	65
6.2. Experiencia Personal	66
6.3. Impacto de la Ludificación	66
6.4. Contribución	66
6.5. Futuras Etapas del Proyecto	66
6.6. Recursos	67
Bibliografía	69

Índice de figuras

1.1. Desarrollo Convencional vs. Desarrollo con ingeniería de líneas de Producto de Software. Tomado de [Óscar Díaz García and Gonzalez, 2007].	5
2.1. Variaciones en el tiempo y el espacio. Adaptado de [Krueger, 2006a] y [Krueger and Clements, 2013].	9
2.2. PLE Visto como una Fábrica. Tomado de [Krueger and Clements, 2013].	15
3.1. Mecánicas de atención al cliente. Imagen obtenida mediante el videojuego Restaurant Story 2.	24
3.2. Mecánicas de armar alimentos. Imagen obtenida mediante el videojuego Cooking Fever.	25
3.3. Mecánicas de Drag and Drop. Imagen obtenida mediante el videojuego Mi Increíble Cuerpo Humano.	25
4.1. Servidores Elaborados en Godot Engine.Elaboración Propia.	31
4.2. Diagrama de Despliegue UML - Servidores.Elaboración Propia.	32
4.3. Diagrama de Secuencia UML - Servidores.Elaboración Propia.	33
4.4. Tablas de Diálogos. Elaboración Propia.	34
4.5. Juegos e Internacionalización. Elaboración Propia.	35
4.6. Interfaz LoginScreen Multilenguaje. Elaboración Propia.	35
4.7. Mecánicas de Atención al Cliente. Elaboración Propia.	36
4.8. Mecánicas de Drag and Drop. Elaboración Propia.	36
4.9. Mecánicas de Armar Alimentos. Elaboración Propia.	37
4.10. Mecánicas de Automatización de Procesos. Elaboración Propia.	37
4.11. Proceso de Creación de Personajes. Elaboración Propia.	38
4.12. Proceso de Animación. Elaboración Propia.	39
4.13. Proceso de Creación de Escenarios. Elaboración Propia.	40
4.14. Infografía Grupo de Conceptos 1. Elaboración Propia.	42
4.15. Infografía Grupo de Conceptos 2. Elaboración Propia.	42
4.16. Infografía Grupo de Conceptos 3. Elaboración Propia.	43
4.17. Quiz - Grupo de Conceptos 1. Elaboración Propia.	44
4.18. Quiz - Grupo de Conceptos 2. Elaboración Propia.	44
4.19. Quiz - Grupo de Conceptos 3. Elaboración Propia.	45
4.20. Conexión de Escenas 3. Elaboración Propia.	45
4.21. Selección de Servidores. Elaboración Propia.	46
4.22. Inicio de Sesión. Elaboración Propia.	46
4.23. Crear Usuario. Elaboración Propia.	47
4.24. Lobby. Elaboración Propia.	47

4.25. Sección de Coleccionables. Elaboración Propia.	48
4.26. Selección de Niveles. Elaboración Propia.	48
4.27. Bienvenida - Grupo de Conceptos 1. Elaboración Propia.	49
4.28. Nivel Básico 1. Elaboración Propia.	49
4.29. Reseñas Clientes Nivel Básico 1 . Elaboración Propia.	50
4.30. Introducción Grupo de Conceptos 2. Elaboración Propia.	50
4.31. Nivel Básico 2 . Elaboración Propia.	51
4.32. Reseñas Clientes Nivel Básico 2. Elaboración Propia.	51
4.33. Introducción Grupo de Conceptos 3. Elaboración Propia.	52
4.34. Nivel Básico 3 . Elaboración Propia.	52
4.35. Reseñas Clientes Nivel Básico 3. Elaboración Propia.	53
5.1. Discord como Medio de Comunicacion. Elaboración Propia.	55
5.2. Diagrama de Comparativa de Quices.Elaboración Propia.	63

Índice de tablas

2.1. Técnicas Mecánicas. Adaptado de [Villay, 2016].	11
2.2. Técnicas Dinámicas. Adaptado de [Villay, 2016].	12
3.1. Tipos de géneros. Adaptado de [Corporation, 2021a].	22
3.2. Progresión del nivel 1, conceptos básicos. Adaptado de [Inc, 2020a].	26
3.3. Progresión del nivel 2, conceptos básicos. Adaptado de [Inc, 2020a].	26
3.4. Progresión del nivel 3, conceptos básicos. Adaptado de [Inc, 2020a].	27
4.1. Comparativa entre Motores de Videojuegos. Información obtenida de las paginas web de cada herramienta. Elaboración Propia.	30
5.1. Caso de Prueba 1 - Selección de Servidores. Elaboración Propia.	56
5.2. Caso de Prueba 2 - Crear Usuario. Elaboración Propia.	56
5.3. Caso de Prueba 3 - Inicio de Sesión. Elaboración Propia.	57
5.4. Caso de Prueba 4 - Lobby. Elaboración Propia.	57
5.5. Caso de Prueba 5 - Sección de Coleccionables. Elaboración Propia.	57
5.6. Caso de Prueba 6 - Selección de Niveles. Elaboración Propia.	58
5.7. Caso de Prueba 7 - Introducción Grupo de Conceptos (N). Elaboración Propia.	58
5.8. Caso de Prueba 8 - Nivel Básico 1. Elaboración Propia.	59
5.9. Caso de Prueba 9 - Nivel Básico 2. Elaboración Propia.	59
5.10. Caso de Prueba 10 - Nivel Básico 3. Elaboración Propia.	60
5.11. Caso de Prueba 11 - Reseñas Clientes Nivel Básico (N). Elaboración Propia.	60
5.12. Caso de Prueba 12 - Infografía (N). Elaboración Propia.	60
5.13. Caso de Prueba 13 - Quiz (N). Elaboración Propia.	61
5.14. Usabilidad. Elaboración Propia.	62
5.15. Percepción de Utilidad. Elaboración Propia.	62

Introducción

Disminuir los tiempos y costos de desarrollo de los productos de software es un reto de las empresas contemporáneas. En la actualidad, pocas personas conocen el enfoque de líneas de producto de software (SPL) y menos confianza tienen en él [Azanza et al., 2021]. Sin embargo, existen varios casos de éxito, por ejemplo, el caso de la empresa sd&m que desarrolla y fabrica componentes de interfaz de usuario (UI) para sistemas embebidos. Gracias a las líneas de producto de software, esta empresa pudo mantenerse dentro del cronograma, no sobrepasar el presupuesto y garantizar la alta calidad de los componentes [Birk, 2002].

Una de las estrategias pedagógicas contemporáneas que ha adquirido importancia en el campo de la enseñanza es la ludificación (conocida en inglés como *Gamification*). La ludificación permite transformar los enfoques educativos tradicionales para incorporar diversos procesos básicos como la emoción, la motivación y el aprendizaje asociativo [Vargas and Silva, 2019].

Este trabajo busca explorar el uso de técnicas de ludificación, para enseñar conceptos fundamentales de la ingeniería de líneas de producto de software. Para ello se desarrolló un prototipo, como herramienta de capacitación para entrenar a personas sin experiencia en líneas de producto de software.

Descripción del Problema

Una línea de producto de software es la agrupación de sistemas intensivos de software que pertenecen al mismo dominio, comparten características similares y tienen variabilidad. “*Estos sistemas implican una reutilización estratégica y planificada que produce resultados predecibles, los cuales son desarrollados a partir de partes prescritas reutilizables conocidas como núcleos (Core Assets)*” [Clements and Northrop, 2001].

La práctica exitosa de la ingeniería de líneas de producto de software puede involucrar beneficios para las empresas tales como la disminución de los tiempos, costo de desarrollo y el incremento de la calidad de los productos de software [Pohl et al., 2005]. Sin embargo, como sucede en todo cambio organizacional importante, la ingeniería de líneas de producto de software, implica cambios organizacionales en los procesos comerciales e incluso a nivel tecnológico. Por lo tanto, llevar un cambio integral a la organización de desarrollo de software no es una tarea fácil y los falsos comienzos pudiesen ser costosos e inevitables para las empresas [Frank J and Rommes, 2007, Northrop, 2008].

Un aspecto fundamental para lograr que las organizaciones puedan introducir cambios exitosamente, es la comunicación del cambio entre todos los agentes involucrados; con el fin de evitar inconvenientes que pudieran poner en peligro el éxito del proceso de cambio. Por esa razón, es importante contar con una metodología que le permita a las empresas, enseñar y comunicar los conceptos fundamentales del cambio de una manera rápida y atractiva para sus empleados [Kotter and Cohen, 2002].

El presente trabajo de investigación, propone usar técnicas de ludificación en una herramienta de software que facilite la enseñanza de conceptos introductorios de ingeniería de líneas de producto de software. Con esta herramienta se espera que las organizaciones interesadas en adoptar la ingeniería de líneas de productos de software para sus empresas, puedan facilitar el entrenamiento de los actores involucrados en este proceso por medio del uso de la herramienta.

1.1. Formulación

¿Cómo implementar una herramienta basada en técnicas de ludificación para introducir conceptos asociados con la ingeniería de líneas de producto de software?

1.2. Sistematización

- ¿Qué técnicas de ludificación pueden ser implementadas para entrenar personas sin experiencia previa en líneas de productos de software?
- ¿Cuáles son los conceptos de la ingeniería de líneas de producto de software para los que se aplicarán las técnicas de ludificación?
- ¿Cómo unificar las técnicas de ludificación y conceptos seleccionados en una solución que ayude a la enseñanza de los conceptos fundamentales de ingeniería de líneas de producto de software?
- ¿Cómo evaluar la solución implementada?

1.3. Objetivos

1.3.1. Objetivo General

Explorar el uso de técnicas de ludificación para enseñar los conceptos asociados con la ingeniería de líneas de producto de software mediante el desarrollo de una herramienta de software que facilite el proceso de enseñanza.

1.3.2. Objetivos Específicos

- Identificar las técnicas de ludificación a utilizar.
- Seleccionar los conceptos de la ingeniería de líneas de producto de software para los que se aplicarán las técnicas de ludificación.
- Implementar una herramienta basada en técnicas de ludificación que enseñe los conceptos de ingeniería de líneas de producto de software seleccionados.
- Evaluar la usabilidad y percepción de utilidad de la herramienta implementada.

1.4. Justificación

La ingeniería de líneas de producto de software crea una oportunidad favorable para las empresas debido a la reducción de los tiempos de producción, al incremento de la productividad por parte de los ingenieros de software y el ahorro generado a partir de iniciar y mantener un conjunto de productos de software con características similares [Óscar Díaz García and Gonzalez, 2007].

En la **figura 1.1**, compara los costos entre el desarrollo convencional e ingeniería de líneas de producto de software. El entorno tradicional se enfoca principalmente en el producto, mientras que la ingeniería de líneas de producto de software está más orientada a la gestión de características similares, la reutilización planificada y la incorporación de nuevas variantes de manera sistemática [Óscar Díaz García and Gonzalez, 2007].

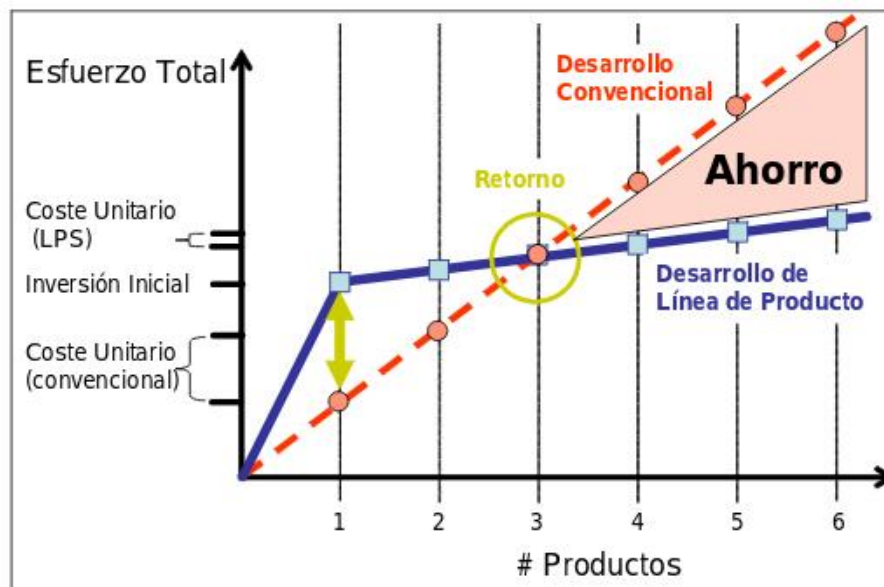


Figura 1.1: Desarrollo Convencional vs. Desarrollo con ingeniería de líneas de Producto de Software. Tomado de [Óscar Díaz García and Gonzalez, 2007].

El objetivo de este proyecto es la exploración de un conjunto de técnicas, como alternativas a los enfoques de enseñanza tradicionales, con el fin de alentar a que las personas conozcan qué es la ingeniería de líneas de producto de software y aceleren su proceso de capacitación. Para ello, en este trabajo se creó un prototipo funcional basado en técnicas de ludificación que tiene la finalidad de enseñar los conceptos básicos de la ingeniería de líneas de producto de software y con ello facilitar el proceso de capacitación a las personas sin conocimientos previos en el área, con el fin de familiarizarse con los conceptos básicos en esta área del conocimiento.

1.5. Delimitaciones y Alcances

1. El prototipo ha sido desarrollado mediante el motor gráfico Godot Engine.
2. El prototipo desarrollado funciona en una plataforma web y puede ser ejecutado en un solo PC.
3. El prototipo emplea cuatro técnicas mecánicas y tres técnicas dinámicas de ludificación.
4. Solo se abordaron conceptos introductorios de línea de producto de software: beneficios, costos, requisitos, características generales y terminología.

1.6. Entregables

- Trabajo de grado con toda la documentación respectiva a las técnicas de ludificación y los conceptos introductorios de línea de producto de software implementados. Además de los resultados obtenidos mediante las pruebas de funcionalidad, usabilidad y percepción de utilidad.
- Prototipo funcional en una plataforma web.
- Repositorio GitHub (Código fuente, requisitos de uso e instalación)

1.7. Resultados Obtenidos y Contribuciones

- Implementación de los conceptos introductorios de la ingeniería de líneas de producto de software.
- Diseño del prototipo basado en técnicas de ludificación.
- Desarrollo del prototipo funcional.

Desarrollo del Proyecto

2.1. Áreas de Referencia

Este proyecto se enfoca en las siguientes definiciones propias de las disciplinas como punto de partida en la contextualización.

- Ingeniería de Líneas de Productos de Software.
- Interacción Humano-Computador.
- Ludificación.

2.2. Marco de Referencia

2.2.1. Conceptos Relacionados con la Ingeniería de líneas de Producto de Software

En esta sección se definirá cada uno de los conceptos relacionados con la ingeniería de líneas de producto de software que serán empleados en los capítulos posteriores.

2.2.1.1. Ingeniería de Líneas de Producto de Software

Se puede definir la ingeniería de líneas de producto de software, como el desarrollo de productos que comparten características similares, respetando y gestionando sus diferencias, la cual permite obtener mayores beneficios en calidad, costos y tiempo [Clements and Northrop, 2001]. Cuando se habla de ingeniería, se hace referencia a todas las actividades involucradas en la planificación, producción, entrega, implementación, mantenimiento y retiro de productos. [Krueger and Clements, 2013].

2.2.1.2. Líneas de Producto de Software

Las líneas de producto de software, son un conjunto de sistemas de software que comparten características similares (**features**), satisfacen las necesidades específicas de un dominio o segmento particular de mercado y son desarrolladas a partir de un sistema común de activos base (**core assets**) de una manera preestablecida [Northrop, 2008].

2.2.1.3. Reúso | Reutilización de Software

El reúso (**reutilización de software**) es el proceso de creación de sistemas nuevos a partir de la utilización sistemática de artefactos de software existentes [Dusink and van Katwijk, 1995], los cuales no se limitan a fragmentos de código fuente, sino que pueden incluir estructuras de diseño, especificaciones, documentación, entre otros. En el artículo **Software Reuse** Charles W. Krueger menciona, cuatro tipos de dimensiones de la reutilización de software [Krueger, 1992]; que se describen a continuación:

- **Abstracción:** Todos los enfoques de reutilización de software, utilizan alguna forma de abstracción para los artefactos de software; la abstracción permite suprimir detalles innecesarios y enfatiza la información importante de los artefactos, con el fin de definir de manera más precisa la utilidad del artefacto, orientando cuando y cómo podría ser reutilizado, entre otros aspectos.
- **Selección:** En la mayoría de enfoques de reutilización de software, implementan estrategias para localizar, analizar, comparar, clasificar y seleccionar artefactos de software reutilizables, con el objetivo de brindar un mayor beneficio a los desarrolladores de software a la hora emplear un artefacto en específico.
- **Especialización:** Surge la necesidad, de especializar un artefacto generalizado en el momento en que varias tecnologías de reutilización cuentan con artefactos similares y posteriormente los combinan, con el fin de crear un solo artefacto generalizado. La manera de especializar el artefacto es a través de parámetros, transformaciones, limitaciones o distintos tipos de refinamientos.
- **Integración:** En esta dimensión, se introduce el concepto de **integration framework**, el cual es adoptado por varias tecnologías de reutilización; los desarrolladores de software emplean este framework con el objetivo de crear sistemas de software completos, por medio de la combinación de un conjunto de artefactos especializados.

2.2.1.4. Producto Intangible

En la metodología de personalización masiva de software, los productos son tangibles únicamente después de la etapa de producción automatizada (**configurador**), es decir, antes de esta etapa solo se tienen los **activos de software** y el **modelo de características**, no existe el producto (N) [Krueger, 2006a]. Lo expresado anteriormente, se puede visualizar en la figura **figura 2.2** donde los productos solo se encuentran en las **salidas de productos (Product Line Portfolio)**

2.2.1.5. Gestión de variaciones en el tiempo y el espacio

De la misma manera que en la ingeniería de software convencional, los artefactos en las líneas de producto de software están sujetos a mantenimiento y evolución [Krueger, 2006b]; en la **figura 2.1** se expone la diferencia entre variación en el tiempo y el espacio, la variación en el tiempo se puede apreciar en los activos principales y en los modelos de productos que evolucionan en el tiempo; estos se encuentran en la parte superior de la **figura 2.1**, la línea de base representa versiones estables y compatibles de los activos principales; por otro lado, en la parte inferior de la **figura 2.1**, se puede observar la variación en el espacio, debido a que el configurador toma los activos principales y los modelos de productos de una línea de base determinada y crea una instancia de los diferentes productos a lo largo del espacio de la línea de productos [Krueger, 2006a].

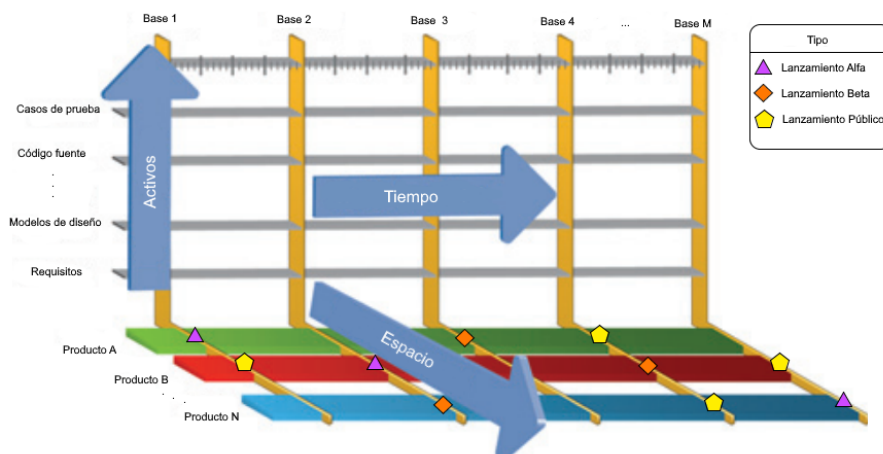


Figura 2.1: Variaciones en el tiempo y el espacio. Adaptado de [Krueger, 2006a] y [Krueger and Clements, 2013].

2.2.1.6. Variabilidad

La variabilidad se puede expresar como un conjunto de variaciones que se producen a partir de un producto en común, podemos tomar como ejemplo una fábrica que produce distintas opciones para un mismo producto, en este caso la fábrica produce camisetas, una camisa puede tener distinto tamaño, color, estampado, entre otras características. De manera similar la ingeniería de línea de producto de software aplica este concepto a los artefactos de software, se pueden ver variaciones en cualquier tipo de activo que deba gestionarse y sus variantes coinciden con el producto en común; pueden ser requisitos, implementaciones, casos de prueba, diseños, entre otros [Krueger and Clements, 2013].

2.2.1.7. Beneficios de la Ingeniería de Líneas de Producto de Software

El principal beneficio presentado en la **metodología de personalización masiva de software** es el intercambio efectivo de activos, el cual se traduce en el fácil acceso y continua evolución de los activos de software por parte de las empresas [Bosch, 2001], además de posibilitar la administración de un conjunto de productos que comparten características similares en vez de múltiples productos por separado, aportando enormes eficiencias en la producción y el mantenimiento [Krueger and Clements, 2013]; en el artículo escrito por Charles W. Krueger, *Introduction to the emerging practice of software productline development*, publicado en *Methods & Tools*, se listan dichos beneficios [Krueger, 2006b]:

- Aumento en el número total de productos que pueden implementarse y administrarse de manera efectiva.
- Aumento en la calidad del producto.
- Reducción en el tiempo promedio para crear e implementar un producto nuevo.
- Reducción en el tiempo promedio de comercialización por producto.
- Reducción en el número promedio de defectos por producto.
- Reducción en el esfuerzo promedio de implementación, mantenimiento y actualización de un producto.
- Reducción del riesgo del producto.
- Capacidad para afectar la personalización masiva.

2.2.1.8. Desventajas de la Ingeniería de Líneas de Producto de Software

Como se mencionó en la subsección anterior, esta metodología se centra principalmente en el correcto funcionamiento y administración de los activos de software; lo cual implica que si no se cuenta con una buena cultura organizacional, no existe una entidad o algún tipo de necesidad que se centre en los activos compartidos, causando a futuro problemas en la arquitectura y componentes de todo el sistema [Bosch, 2001].

2.2.2. Conceptos Relacionados con la Ludificación

En esta sección, se definirá cada uno de los conceptos relacionados con la metodología de ludificación, videojuegos e interacción, que posteriormente serán abordados para la elaboración del prototipo web.

2.2.2.1. Ludificación

Uno de los temas más recurrentes cuando se habla de innovación en el campo de la enseñanza es la ludificación, la cual es una de las estrategias pedagógicas contemporáneas que ha adquirido mayor relevancia en los últimos años [Vargas and Silva, 2019]. La ludificación se fundamenta principalmente en el uso de técnicas de juego orientadas a resolver problemas prácticos, con el fin de involucrar a grupos o audiencias específicas [Ysmar Vianna and Tanaka, 2014]. La ludificación emplea técnicas que se pueden considerar como reglas pedagógicas que permiten mantener la motivación y concentración del usuario a lo largo del juego, determinar la complejidad del juego y los diversos elementos dentro del mismo [Prakash and Rao, 2015]. Las técnicas anteriormente mencionadas se dividen en 2 grupos expuestos a continuación:

- En la **tabla 2.1** se puede observar las técnicas mecánicas, que se centran en recompensar al usuario en función de los objetivos alcanzados.
- En la **tabla 2.2** se pueden observar las técnicas dinámicas, que hacen referencia a la motivación del propio usuario para jugar y seguir adelante en la consecución de sus objetivos.

Técnicas Mecánicas	Descripción
Recolección de Puntos	Se establece un valor cuantitativo a un conjunto de acciones y se almacenan una vez que se completan.
Progreso de Niveles	Se definen un conjunto de retos por nivel que aumentan en dificultad a medida que el jugador completa un nivel.
Premios	El jugador recibe recompensas cada vez que completa una serie de objetivos los cuales puede coleccionar.
Clasificaciones	Exponer las habilidades de un jugador en una lista a partir del puntaje obtenido por nivel. Fomenta la competitividad entre jugadores.
Misiones	Objetivos, actividades o retos preestablecidos que ponen a prueba al jugador o jugadores en un ambiente específico.
Feedback	Le permite al jugador saber que decisiones son correctas o incorrectas.

Tabla 2.1: Técnicas Mecánicas. Adaptado de [Villay, 2016].

Técnicas Dinámicas	Descripción
Progreso	Esta técnica se centra en la necesidad humana de una constante evolución rumbo a una meta, permite que el jugador tenga la motivación para continuar con los objetivos que se le planteen a lo largo de las misiones.
Competitividad	Esta técnica es recurrente en el ámbito deportivo, permite que las personas den su máximo esfuerzo en la actividad que están realizando con el fin de sobresalir frente a los demás.
Reconocimiento o estatus	Esta técnica se enfoca principalmente en reconocer las habilidades y el desempeño que ha tenido el jugador a lo largo de las actividades efectuadas. Mediante el uso logros el jugador puede presumir a los demás jugadores las hazañas que ha completado.
Altruismo	Esta técnica tiene un fin más espiritual, pues a menudo los seres humanos tienden a hacer acciones solidarias que retribuyen al jugador de una manera cualitativa, por ejemplo misiones de rescate.
Diversión	Esta técnica está orientada a que el jugador realice actividades propuestas por voluntad propia y no porque esté forzado a completarlas, pues el factor de diversión incentiva que el jugador tenga el deseo de completar las actividades.

Tabla 2.2: Técnicas Dinámicas. Adaptado de [Villay, 2016].

2.2.2.2. Videojuego

Una de las áreas de entretenimiento electrónico que más ha tenido auge en la actualidad han sido los videojuegos, los cuales han logrado obtener picos de hasta 24,277,811 usuarios conectados de manera simultánea mediante la plataforma de distribución digital de videojuegos más grande del mundo conocida como Steam [Corporation, 2021b]. Los videojuegos permiten que uno o varios usuarios se conecten de manera remota en un mismo mundo virtual, este mundo se compone de una serie de reglas que determinan la interacción con el mismo y varían dependiendo de las dimensiones (**2D** o **3D**) en las que se desarrolle el videojuego. Cada usuario conectado en este mundo toma el control de un personaje ficticio, el cual debe seguir una historia lineal llena de objetivos y recompensas, en donde aumentará la dificultad del videojuego a medida que se avance en el progreso de la historia [Vallejo and Martín, 2017].

2.2.2.3. Interacción

El término Interacción Humano-Computador por sus siglas en inglés “HCI”, fue adoptado en el año 1980, para describir un nuevo campo de estudio orientado al análisis de todos los aspectos que relacionan la interacción entre el usuario y el computador. HCI “*es una disciplina que se ocupa del diseño, la evaluación y la implementación de sistemas informáticos interactivos para uso humano y con el estudio de los principales fenómenos que los rodean*”[Perlman, 1994], en términos más simples, HCI se puede ver como la actividad realizada por parte de los usuarios en función de los periféricos del computador, es decir, se está realizando una interacción cuando el usuario emplea acciones con el ratón y el teclado [Issa and Isaias, 2015]; los factores humanos como la atención, motivación, memoria, emociones, entre otros, están relacionados directamente con el HCI, debido a que sin alguno de ellos, los usuarios tienden a experimentar apatía, inactividad, pasividad y dificultad en el aprendizaje [González and Blanco, 2008].

2.2.2.4. Motor de Videojuegos

Los motores de videojuegos son herramientas conformadas por un conjunto de funcionalidades, orientadas al diseño, estructura y acoplamiento del desarrollo de videojuegos; le permite a los desarrolladores construir escenarios 2D Y 3D, facilitar el cálculo de colisiones entre distintos objetos, administrar y optimizar la memoria empleada por el videojuego, tener pleno control en las animaciones, entre otras funcionalidades [Villay, 2016]. En la actualidad existen varios motores gráficos que proveen distintos beneficios a la hora de desarrollar un videojuego, estos son algunos de los más utilizados:

- **Unity:** Es un motor de videojuegos multiplataforma, desarrollado por Unity Technologies, el cual provee funcionalidades para la creación de videojuegos AAA, está orientado al desarrollo 2D y 3D con un gran portafolio de ejemplos desarrollados por ellos mismos, además del alcance en distintas plataformas como realidad aumentada, realidad virtual, plataformas móviles, de escritorio, consolas, tv y la web [Technologies, 2021].
- **Unreal:** Es un motor de videojuegos multiplataforma, desarrollado por Epic Games bajo licencia privativa, emplea un sistema visual de scripting llamado “Blueprints”, el cual fue creado por Epic Games facilitando el desarrollo de videojuegos, está orientado al desarrollo 2D y 3D con un mejor manejo de las físicas de partículas e iluminación que sus competidores, cuenta con distintas plataformas de desarrollo como lo son consolas, plataformas de realidad aumentada y virtual, entre otros[Games, 2021].
- **Cryengine:** Es un motor de videojuegos multiplataforma desarrollado por Crytek, ha cambiado a distintos tipos de licencias en el transcurso del tiempo, desde software propietario, licencia libre, entre otros tipos de licencias, está orientado al desarrollo 3D y posee buenas funcionalidades con el manejo de luz, simulación de vegetación, estructuración de escenarios, entre otros[Crytek, 2021].

- **Godot:** Es un motor de videojuegos multiplataforma de código abierto bajo la licencia permisiva del MIT, manejar este tipo de licenciamiento le garantiza al desarrollador libre uso comercial de los productos desarrollados, este motor gráfico está orientado a la creación de juegos 2D y 3D y tiene una interfaz integrada con múltiples herramientas para facilitar el desarrollo del proyecto. Godot Engine soporta múltiples plataformas de escritorio (**Linux, macOS y Windows**), plataformas móviles (**Android y iOS**) y basadas en la web (**HTML 5**) [Studios, 2021].

2.2.3. Game Design Documentation Builder

Es un documento de trabajo creado por la empresa Softweir Inc, que permite crear un esquema del proyecto del videojuego a desarrollar. Al diseñar un videojuego es importante llevar un control más riguroso de todos los diversos elementos que componen un videojuego, entre estos elementos se encuentran el público objetivo al cual está orientado el videojuego, los distintos niveles, el tipo de historia, género, entre otros [Inc, 2020b].

2.2.4. Metodología de Personalización Masiva de Software - Charles W. Krueger, BigLever Software

Para este proyecto se ha seleccionado la metodología de personalización masiva de software, como referencia para el proceso de ludificación de los conceptos básicos de la ingeniería de líneas de producto de software, debido a que en la actualidad es una de las metodologías más usadas en la industria y con mayor número de casos reales reportados por la empresa BigLever Software Inc [Lanman et al., 2011] [Krueger and Clements, 2018] [Lanman et al., 2013].

- **Core Assets | Entradas de Activos:** Son los requerimientos, componentes del código fuente, casos de prueba, arquitectura y documentación; todos estos activos pueden ser modificados o compuestos de distinta manera, pueden provenir de diversas fuentes como artefactos reutilizables de una biblioteca, artefactos rediseñados de productos ya existentes o artefactos diseñados desde cero. Una característica importante de las entradas de activos es que poseen un rol específico dentro de una arquitectura similar [Krueger, 2006b].
- **Product Specifications | Modelo de Características:** Describe las opciones para cada una de las características, ya sean comunes o variables de los productos en las líneas de producto de software; cada producto es definido únicamente por su modelo de características [Krueger, 2006b].

- **Product Configurator | Configurator:** También conocido como mecanismo de producción y proceso. El configurador es el medio para componer y estructurar productos a partir de las entradas de activos, durante la producción se emplean las decisiones de producto para determinar cómo ajustar los puntos de variación de las entradas de activos; existen 3 tipos de estado a los que puede llegar el configurador por lo que en este proyecto, se tuvo en cuenta el estado completamente automatizado, el cual obtiene suficiente información del modelo de características para poder generar automáticamente las salidas de productos de software [Krueger, 2006b].
- **Product Line Portfolio | Salidas de Productos de Software:** Es la colección de todos los productos que se pueden generar a partir de los activos de software y el modelo de características [Krueger, 2006b].

En la **figura 2.2** se puede visualizar de una manera más detallada la integración de cada uno de los conceptos previamente expuestos en subsección anterior; en una primera instancia se tienen las entradas de activos (**Core Assets**) que posteriormente serán necesarios para que el configurador (Product Configurator) pueda integrar, estructurar y generar sus respectivas salidas de productos de software (**Product Line Portfolio**), teniendo como referencia el modelo de características (**Product Specifications**).

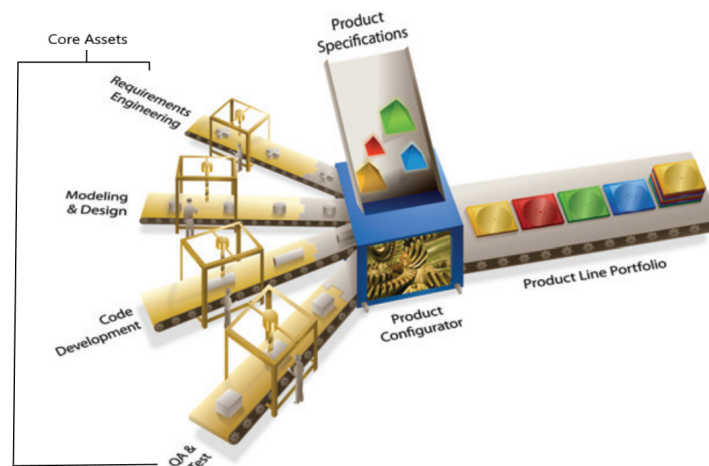


Figura 2.2: PLE Visto como una Fábrica. Tomado de [Krueger and Clements, 2013].

2.3. Trabajos Relacionados

- Un estudio académico realizado en el año 2018 por el autor Juan Carlos Hernández Palencia, denominado **Prácticas de Ludificación en el Desarrollo de Software Representadas en el Núcleo de SEMAT**, propuso emplear la metodología de ludificación en el núcleo de SEMAT para el desarrollo de las prácticas, mediante las mecánicas de valoración cuantitativa de tareas y entrega de recompensas. En el desarrollo del trabajo se realizó un análisis de estudio de caso en el ámbito educativo, en el que las principales conclusiones del trabajo señalan que este tipo de metodologías facilitan la creación de nuevas prácticas, al identificar explícitamente los elementos que las componen y principalmente, sirven como referencia para adaptar, comparar y mejorar las actividades realizadas a lo largo de las prácticas. El estudio académico toma relevancia en la medida que aporta a la detección de nuevos patrones de prácticas de ludificación, donde se crean diseños de estrategias pedagógicas favorables para el desarrollo del proyecto actual [Palencia, 2018].
- En alianza con COLCIENCIAS, Pontificia Universidad Javeriana Cali y el Instituto para niños ciegos y sordos del valle del Cauca en el año 2017, presentaron un trabajo titulado como **Videojuego para Apoyar el Proceso de Descripción Estática en la Terapia del Lenguaje Aplicada a Niños**, que propuso desarrollar un prototipo funcional por medio de un videojuego como herramienta lúdica, para apoyar el proceso de descripción estática como terapia del lenguaje realizada a niños usando la metodología de mecánicas de juegos o también conocidas como metodología de ludificación, con una jerarquía de mini-juegos, desarrollada en el motor gráfico Construct2, empleando la herramienta de diseño gráfica Pixton.com y utilizando los recursos de Kenney.nl. Los principales resultados de este trabajo fueron el prototipo funcional para computador y móvil; en cuanto a las conclusiones, la herramienta permitió mantener la atención y niveles de motivación de los menores. En relación con la presente investigación, este trabajo aporta en el impacto pedagógico que han generado los resultados expuestos anteriormente, ya que por medio de la personalización de actividades se ha logrado motivar y capturar la atención de los niños evaluados mejorando los resultados que se obtenían en las actividades anteriores al uso de la herramienta [Villay, 2016].

- La Universidad de Valladolid en el año 2016, realizó un trabajo denominado **Estudio de la Gamificación de una Empresa para Incentivar la Motivación**. Este trabajo propuso como objetivo principal, conocer los distintos conceptos relacionados con el mundo de la gamificación para así poder realizar una simulación de implantación de un sistema de ludificación. Esta simulación se realizó con la versión de prueba de Hoopla con la empresa El Corte Inglés, bajo la premisa que en los últimos meses el departamento de electrónica ha generado un margen inferior de ganancia con respecto a los últimos años a causa de la desmotivación que sufren sus empleados por las reformas laborales que ha hecho la compañía. Los resultados al implantar el nuevo sistema basado en técnicas de ludificación en uno de los centros reportó beneficios en el margen de ganancias tras su implementación, por lo tanto, el sistema resultó ser exitoso [Caballero, 2016]. Esta referencia académica se relaciona con el presente trabajo al valorar los conceptos y técnicas relacionadas en la metodología de ludificación y que buscan solucionar un problema o llegar a un objetivo en particular.
- Cristina Martínez Martínez de la Universidad de Valladolid, realizó un trabajo en el año 2017, denominado **Estudio de Gamificación en una Empresa para Mejorar la Fidelización de los Clientes**, se planteó como objetivo principal, conocer los distintos conceptos relacionados con el mundo de la ludificación para así, poder estudiar una aplicación ludificada para incentivar la fidelidad de los clientes de una empresa. La autora implementó su estudio en una cadena de restaurantes de comida americana, demostrando con este caso una excelente solución para el restaurante frente a necesidad de la fidelidad de los clientes. Los resultados de la aplicación del prototipo de ludificación permitieron generar una imagen positiva y de fidelización hacia la marca del restaurante de comida americana, aumentando el interés de los usuarios que desde un principio hayan tenido predisposición de consumo. El trabajo anterior se puede relacionar con el presente trabajo en que mediante el uso de la metodología de ludificación, se obtiene una mayor atención por parte del público objetivo, reforzando su fidelidad hacia una marca o idea en cuestión; otro aspecto que se resalta es la disposición, tiempo y presupuesto que deben afrontar las empresas al implementar métodos lúdicos para sus estrategias de mercadeo [Martínez, 2017].

Diseño - Proceso de Ludificación

En el presente capítulo se ha optado por adaptar secciones específicas del documento **Game Design Documentation Builder** [Inc, 2020b] previamente mencionado en la subsección **2.2.3**, con el fin de detallar de una manera más profesional las características principales para el desarrollo posterior del prototipo. De igual manera en el desarrollo de este capítulo, se implementan secciones que agrupan y exhiben los conceptos básicos presentados en el **capítulo 2**. Además de exponer las técnicas de ludificación (**técnicas mecánicas y dinámicas**) presentadas en la subsección **2.2.2.1. Ludificación**.

3.1. Público Objetivo

Se determinó que el público objetivo, estará enfocado principalmente en empresas que deseen adoptar la ingeniería de línea de producto de software, más específicamente la metodología de personalización masiva de software; para ello, se ha separado el público objetivo en 2 grupos que se detallarán a continuación:

- **Quienes Toman las Decisiones:**
 - Empresarios
 - Líderes técnicos

- **Quienes se ven Afectados por la Opción de la Ingeniería de Líneas de Producto de Software:**
 - Desarrolladores
 - Equipo de mercadeo y ventas
 - Equipo de calidad

3.2. Grupos de Conceptos

En esta sección, se expondrán los grupos de conceptos que fueron empleados para desarrollar la historia del prototipo. Con el fin de aplicar las técnicas de ludificación, se ha optado por ordenar y separar los distintos conceptos trabajados en las subsecciones **2.2.1. Conceptos Relacionados con la Ingeniería de líneas de Producto de Software** y **2.2.4. Metodología de Personalización Masiva de Software** en 3 grupos de conceptos; los cuales comparten características similares permitiendo llevar una continuidad a lo largo de los niveles, de esta manera el usuario puede asociar los conceptos del mismo grupo y entender los conceptos más fácilmente. El primer grupo consta de conceptos directamente ligados con la metodología de personalización masiva de software, el segundo grupo está conformado por los conceptos de la ingeniería de líneas de producto de software y el tercer grupo lo compone conceptos de ambos temas; este orden fue determinado por el documento **METHODS & TOOLS** [Krueger, 2006a]. A continuación, se presentará cada grupo de conceptos:

Grupo de Conceptos 1

- Core Assets | Entradas de Activos.
- Product Specifications | Modelo de Características.
- Product Line Portafolio |Salidas de Productos de Software.

Grupo de Conceptos 2

- Variabilidad.
- Reuso | Reutilización de Software.
- Gestión de variaciones en el tiempo y el espacio.

Grupo de Conceptos 3

- Product Configurator | Configurator.
- Producto Intangible.
- Beneficios de la Ingeniería de Líneas de Producto de Software.

3.3. Problemas que Enfrentan los Usuarios

En esta sección, se han determinado los problemas que están directamente relacionados con el aprendizaje de cada uno de los conceptos que pertenecen a su respectivo grupo, por lo que cada problema expuesto a continuación determina lo que se pretende enseñar del concepto.

Historia Parte 1

- Insatisfacción por parte de los clientes, debido a que los tiempos de producción son más largos y los clientes se sienten insatisfechos.
- Diferenciar activos comunes y variables.
- Entender el modelo de características.
- Diferenciar entradas y salidas de productos de software.

Historia Parte 2

- Entender el concepto de variabilidad.
- Seleccionar activos posibles para ser reutilizables.
- Identificar las características específicas de la gestión de la variación en el tiempo y el espacio.

Historia Parte 3

- Entender la funcionalidad del configurador.
- Entender el concepto de producto intangible.
- Identificar las ventajas de implementar las líneas de productos de software.

3.4. Jugabilidad

Esta sección tiene como principal objetivo determinar y limitar el género, la experiencia del jugador, las reglas de juego, la progresión del mismo, la satisfacción del jugador y los elementos (factores y mecánicas) que interactúan en el mundo.

3.4.1. Género

A continuación, en la **tabla 3.1** se presentan los distintos géneros (**categorías**) que existen actualmente en el área de videojuegos, la información ha sido extraída de la plataforma Steam previamente mencionada en la subsección **2.2.2.2. Videojuego**. Mediante la siguiente tabla, se ha definido que el género correspondiente al desarrollo del videojuego es **simulación**, y el subgénero que más se adapta a la propuesta de desarrollo es **construcción y automatización**; se ha seleccionado este género y subgénero partiendo del enfoque que tiene la metodología de personalización masiva de software, la cual se encarga de automatizar los procesos de creación y mantenimiento de los activos de software mediante el uso del configurador.

Género	Subgéneros	
Acción	Acción Roguelike Disparos en primera persona Lucha y artes marciales Plataforma y corredores	Arcade y ritmo Disparos en tercera persona Pelear callejeras
Rol	En grupo Por turnos Roguelike Rol japonés (JRPG)	Estrategia de rol RPG de acción Rol de aventuras
Estrategia	Cartas y mesa Defensa de torres Estrategia por turnos Militares	Ciudades y colonización Estrategia en tiempo real Gran estrategia y 4X
Aventura y Casual	Aventura Casuales Novela visual Rompecabezas	Buena trama Metroidvania Rol de aventuras
Simulación	Agricultura y artesanía Construcción y automatización Espacio y vuelo Sandbox y de física	Citas De vida e inmersivos Negocios y magnates
Deportes y Carreras	Carreras Deportes de equipo Simulador de carreras Todos los deportes	Caza y pesca Deportes individuales Simulador deportivo

Tabla 3.1: Tipos de géneros. Adaptado de [Corporation, 2021a].

3.4.2. Sinopsis del juego

En este punto del diseño, se decidió que la temática a trabajar a lo largo del prototipo sería por la línea de alimentos, ya que estos ejemplifican cada uno de los conceptos, por lo tanto, se determinó que todo transcurriría en un restaurante de pizzas, puesto que dicho contexto, representa de forma adecuada el proceso de creación de productos con características similares (**Pizzería**). Para lograr mayor afinidad con el público objetivo se decidió agregar un personaje principal, llamado David, el cual cumple la función de guía a lo largo del desarrollo de la historia.

David es el dueño de un restaurante que desea mejorar los tiempos de producción de las pizzas que vende, con el fin de obtener mayores ganancias y aumentar la calidad de sus productos. Mediante la implementación de las líneas de productos en el restaurante, David pretende lograr estos objetivos y además aprender cada uno de los conceptos a lo largo de los niveles.

3.4.3. Resumen del Juego

- **Mundo Jugador (Lobby):** Es un espacio destinado para que el jugador pueda observar los coleccionables, puntuaciones y medallas obtenidas a lo largo de los niveles que ha completado.
- **Historia Parte 1 (Grupo Conceptos 1):** En esta primera parte de la historia el jugador tiene como objetivo principal, realizar cada una las pizzas de manera manual, debido a que se pretende que el jugador entienda de primera instancia la importancia y el impacto que puede generar las líneas de producto en las empresas. Al finalizar todas las actividades propuestas, se muestra una comparativa entre los recursos del restaurante y el **grupo de conceptos 1**; por último, se realiza una prueba corta con el fin de evaluar los conocimientos aprendidos en cada uno de los niveles.
- **Historia Parte 2 (Grupo Conceptos 2):** En esta segunda parte el jugador debe realizar las pizzas que se especifican en el menú, con la diferencia de que ahora posee un equipo de trabajo encargado de sectores específicos en el restaurante. Esto le permite al jugador evidenciar cómo se optimiza el tiempo empleado y mejora la calidad de los ingredientes (activos) por medio de la aplicación de las líneas de producto de software. Igual que el final del anterior nivel, se realiza una comparativa y una prueba corta.
- **Historia Parte 3 (Grupo Conceptos 3):** La tercera parte de la historia está principalmente orientada a la explicación del configurador, por lo que el escenario es simplificado a un único dispositivo encargado de generar las pizzas por medio de los anteriores conceptos previamente mencionados en las partes 1 y 2. En dicha instancia, se finaliza nuevamente con la comparativa y prueba corta

3.4.4. Mecánicas del Videojuego

Se definen las mecánicas empleadas desde el tiempo que le toma al jugador completar cada actividad, las recompensas que obtiene el jugador al finalizar una actividad en específico (Pueden ser retribuciones monetarias, audibles o visuales) y hasta el estilo de dificultad (si es estática o incremental). A continuación, se expondrán las mecánicas seleccionadas para el desarrollo del prototipo.

- **Mecánicas de Atención al Cliente**

Son todas las acciones por parte del jugador que afectan los indicativos de los NPC (Non-Player Character), revelando información y permitiendo obtener mejores resultados, por ejemplo, el nivel de satisfacción en el videojuego **RestaurantStory 2**, que se puede observar en la **figura 3.1**, es directamente proporcional al tiempo en el que se le entrega la comida al NPC cliente.



Figura 3.1: Mecánicas de atención al cliente. Imagen obtenida mediante el videojuego Restaurant Story 2.

- **Mecánicas de Armar Alimentos**

Son todas las acciones por parte del jugador que permiten preparar un platillo por medio de un conjunto de ingredientes específicos, por ejemplo, en el videojuego **Cooking Fever**, que se encuentra en la **figura 3.2**, el jugador debe preparar los platos solicitados por parte de los NPC clientes, dependiendo de los ingredientes de cada pedido y en el tiempo asignado.



Figura 3.2: Mecánicas de armar alimentos. Imagen obtenida mediante el videojuego Cooking Fever.

■ Mecánicas de Drag and Drop

Son todas las acciones por parte del jugador que permiten mover objetos específicos alrededor del mundo, por ejemplo, en el videojuego **Mi Increíble Cuerpo Humano**, que se puede observar en la **figura 3.3**, el jugador tiene que reconstruir el cuerpo humano mediante la selección de distintas partes del cuerpo que posteriormente serán colocadas en sus respectivas áreas.

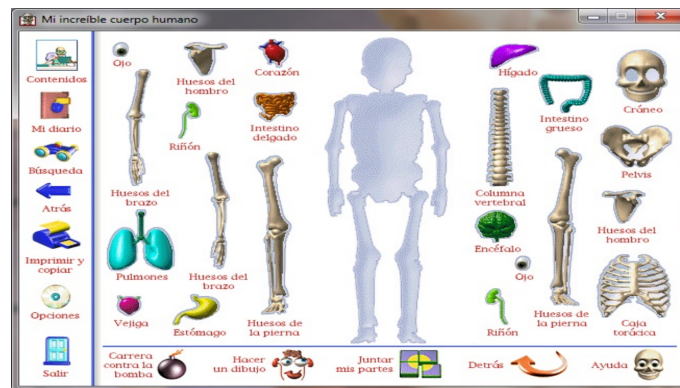


Figura 3.3: Mecánicas de Drag and Drop. Imagen obtenida mediante el videojuego Mi Increíble Cuerpo Humano.

3.4.5. Progresión del Videojuego

A continuación se detallarán los objetivos, recompensas y tiempo destinado para cada una de las actividades planteadas.

- **Historia Parte 1**

Objetivo	Recompensa	Tiempo
Preparar los ingredientes necesarios de la pizza solicitada	Ninguna	4 Segundos por cada ingrediente
Preparar la pizza con los ingredientes correctos	+1 Estrella	50 Segundos por cada pizza
Hornear la pizza	Ninguna	4 Segundos por cada pizza
Entregar la pizza	Ninguna	2 Segundos por cada pizza
Completar las 3 entregas del nivel sin errores	Trofeo del nivel	7 - 9 minutos en completar el nivel
Nota: Las siguientes acciones quitan 1 estrella	Entregar la pizza con los ingredientes incorrectos Quemar pizza en el horno	

Tabla 3.2: Progresión del nivel 1, conceptos básicos. Adaptado de [Inc, 2020a].

- **Historia Parte 2**

Objetivo	Recompensa	Tiempo
Preparar la pizza con los ingredientes correctos	+1 Estrella	50 Segundos por cada pizza
Hornear pizza	Ninguna	4 Segundos por cada pizza
Entregar pizza	Ninguna	2 Segundos por cada pizza
Completar las 3 entregas del nivel sin errores	Trofeo del nivel	5 - 7 minutos en completar el nivel
Nota: Las siguientes acciones quitan 1 estrella	Entregar la pizza con los ingredientes incorrectos Quemar pizza en el horno	

Tabla 3.3: Progresión del nivel 2, conceptos básicos. Adaptado de [Inc, 2020a].

- **Historia Parte 3**

Objetivo	Recompensa	Tiempo
Solicitar pizza con los ingredientes correctos	+1 Estrella	50 Segundos por cada pizza
Entregar pizza	Ninguna	2 Segundos por cada pizza
Completar las 3 entregas del nivel sin errores	Trofeo del nivel	3 - 5 minutos en completar el nivel
Nota: Las siguientes acciones quitan 1 estrella	Entregar la pizza con los ingredientes incorrectos	

Tabla 3.4: Progresión del nivel 3, conceptos básicos. Adaptado de [Inc, 2020a].

3.4.6. Satisfacción de los Jugadores (Sentimientos Objetivo)

Las emociones son sentimientos intensos provocados por personas o los acontecimientos experimentados, experimentar emociones suele ser la principal causa por la que las personas se motivan a jugar videojuegos [Ravaja et al., 2004]. Por este motivo, se han seleccionado 3 emociones objetivo, que causan un efecto positivo en el jugador y que se complementen con las **técnicas dinámicas** que se seleccionarán en una sección posterior.

- **Motivación:** Es el resultado de experiencias previas relacionadas con los diferentes vínculos establecidos entre estímulos y respuestas, es decir, que los jugadores se encontraran más motivados si el juego proporciona una retroalimentación a la hora de ejecutar acciones positivas y negativas [Vargas and Silva, 2019]; además, vemos reflejado este sentimiento en la técnica **progreso**, de la **tabla 2.2**, ya que permite incentivar al jugador a completar los objetivos que se le planteen a lo largo de las misiones.
- **Diversión:** Como se puede observar, en la **tabla 2.2**, la diversión permite focalizar la atención del jugador en una acción de forma agradable, logrando realizar actividades propuestas por voluntad propia y no porque esté forzado a completarlas.
- **Interés:** Este sentimiento tiene en cuenta aspectos del contenido y preferencias de cada individuo, lo que implica que, existirá una mayor motivación por parte de los jugadores si la ludificación satisface los intereses de los participantes en relación con el contexto [Vargas and Silva, 2019].

3.5. Selección de Técnicas Mecánicas y Dinámicas

En la siguiente sección, se especificarán las técnicas mecánicas y dinámicas de la subsección **2.2.2.1. Ludificación**, que serán empleadas en el prototipo, estas técnicas permiten resaltar los sentimientos objetivo de la subsección **3.4.6. Satisfacción de los Jugadores** y son compatibles con las mecánicas expuestas en la subsección **3.4.4. Mecánicas del Videojuego**.

Técnicas Mecánicas

- **Progreso de Niveles:** Esta mecánica está relacionada con la mecánica dinámica de **progreso**, ya que permite crear distintos niveles con dificultades variadas, que motivan al jugador a completar los objetivos que se le presenten.
- **Premios:** Esta mecánica está relacionada con la mecánica dinámica de **reconocimiento o estatus**, debido a que le brinda al jugador objetos con los que puede presumir a los demás jugadores las hazañas que ha completado.
- **Misiones:** Esta mecánica permite complementar la mecánica de **progreso de niveles**, ya que permite añadir distintos tipos de eventos que desafíen al jugador y aumente el interés en la misión actual.
- **Feedback:** Esta mecánica está relacionada con el sentimiento de **motivación**, puesto que permite tener una retroalimentación directa con las acciones que emite el jugador.

Técnicas Dinámicas

- **Progreso:** Esta mecánica está relacionada con el sentimiento objetivo **motivación**, debido a que se genera este sentimiento por parte del jugador, frente a los objetivos a la hora de que exista una constante evolución rumbo a una meta.
- **Reconocimiento o estatus:** Esta mecánica está relacionada con el sentimiento objetivo **interés**, ya que por medio de logros o artículos, el jugador los puede exhibir frente a los demás jugadores, aumentando su interés en la actividad que está realizando con el fin de conseguir los logros o artículos.
- **Diversión:** Esta mecánica está relacionada con el sentimiento objetivo **diversión**, puesto que se pretende que el jugador realice cada una de las actividades propuestas por voluntad propia y no que este obligado a ejecutarlas.

Implementación - Prototipo Web

El capítulo en curso presenta de manera detallada el proceso de desarrollo del prototipo, desde la implementación con servidores (**Gateway, Authentication, World y Python-Client**), hasta el prototipo cliente desde el cual los usuarios pueden interactuar con el mundo virtual. En cada uno de los niveles existe una parte de la historia secuencial y un grupo de conceptos expuestos en la sección **3.2. Grupos de Conceptos**. Por otro lado, se exponen las mecánicas seleccionadas en la subsección **3.4.4. Mecánicas del Videojuego** y los recursos visuales empleados para realizar las técnicas de ludificación en el prototipo.

4.1. Tecnologías Planificadas

Las herramientas que se presentan a continuación, han sido seleccionadas porque cumplen con las siguientes características:

- Son de libre uso.
- Cuentan con la mayoría de herramientas necesarias para la creación de mecánicas y recursos necesarios para el proyecto.
- Existe una experiencia previa.

4.1.1. Motor Gráfico Godot Engine

Como anteriormente se mencionó en la sección **2.2.2.4. Motor de Videojuegos** y basándose en las características presentadas en la siguiente tabla comparativa, se determinó que el motor gráfico más recomendable para el desarrollo del prototipo es Godot Engine. Este motor fue seleccionado porque provee herramientas para realizar proyectos 2D y cuenta con una API Web integrada que permite un control más óptimo de las conexiones cliente-servidor las cuales son fundamentales para almacenar los datos de los usuarios que serán posteriormente analizados en el **capítulo 5**.

En la **tabla 2.3** se presenta una comparativa entre cada uno de los motores de videojuegos previamente expuestos, en la cual se definen las características esenciales como precio, plataformas soportadas, lenguaje de programación, curva de aprendizaje y recursos adicionales.

Motores de Videojuegos	Unity	Cryengine	Unreal	Godot
Precio	Plus \$40*mes / Pro \$150*mes	5 % ingresos (por proyecto)	5 % ingresos (a partir de \$3000*trimestre)	Gratuito
Plataformas Soportadas	Windows, Linux, Mac OS (OSX), HTML5, Andrioid, IOS, Realidad Aumentada, Realidad Virtual y Realidad Virtual			
	PlayStation 4			Necesita empresas de publicación (Terceros)
	Xbox One			
	Nintendo Switch			
	PSP Vita	No aplica		
	Nintendo 3DS			No aplica
	Google Stadia			
11+	4+	2+	2+	
Lenguaje de Programacion	C#		No Aplica	C#
	No Aplica		C++	
			Blueprints	No Aplica
	No Aplica			Visual Scripting Gd Script
Curva de Aprendizaje	Baja/Media		Media / Alta	Baja / Media
Recursos Adicionales	Asset Store	Marketplace		Godot Asset Library(open access)

Tabla 4.1: Comparativa entre Motores de Videojuegos. Información obtenida de las paginas web de cada herramienta. Elaboración Propia.

4.1.2. Editor de Imágenes Paint.NET:

Es un software orientado a la edición de imágenes y fotografía, actualmente solo se ejecuta en los sistemas operativos de windows y cuenta con una interfaz de usuario amigable e intuitiva que está basada en la funcionalidad de capas; muy similar al sistema de capas del software **Photoshop**. Además, este editor tiene una versión gratuita [dotPDN LLC, 2021]. Mediante la herramienta Paint.NET, fueron desarrollados los sprites empleados a lo largo del prototipo, con el fin de obtener recursos personalizados que fueron adecuados a los niveles que se detallarán más adelante en el documento.

4.1.3. Editor de Voxel Artwork MagicaVoxel:

Es un software ligero, intuitivo y de rápida construcción; que le brinda a los artistas herramientas para la creación de objetos 3D, sprites 2D, escenas (2D Y 3D), que pueden ser exportados en formatos Obj, Png, Vox, entre otros; permite la renderización de la escena en tiempo real, cuenta con una licencia gratuita de libre uso en proyectos institucionales o comerciales, puede ser ejecutado en sistemas Windows y MacOS [Ephtracy, 2021]. Por medio del software MagicaVoxel, se desarrollaron las escenografías de los tres niveles propuestos más adelante del documento, esta herramienta se caracteriza principalmente por el uso de bloques a los cuales se les puede asignar distintos tipos

de materiales con características particulares para posteriormente renderizarlos a preferencia del desarrollador y poder obtener efectos de iluminación, reflexión en objetos, entre otros.

4.1.4. Editor de Sonido Audacity:

Es un software de código abierto, enfocado a la edición y grabación de audio multipista, puede ser ejecutado en varios sistemas operativos como Windows, macOS, GNU / Linux, entre otros [Team, 2021]. Audacity permitió extraer, editar y exportar las pistas que fueron empleadas en los niveles del videojuego, con el fin de mejorar la experiencia de los usuarios finales y ambientar los niveles.

4.2. Servidores

Esta sección está orientada a explicar, el funcionamiento de cada uno de los servidores con los que se comunica y trabaja el prototipo. Como se puede observar en la **figura 4.1**, existe un total de cuatro servidores que interactúan entre sí, con el fin de brindar la mayor estabilidad posible a cada una de las sesiones a las que se conectan los usuarios.

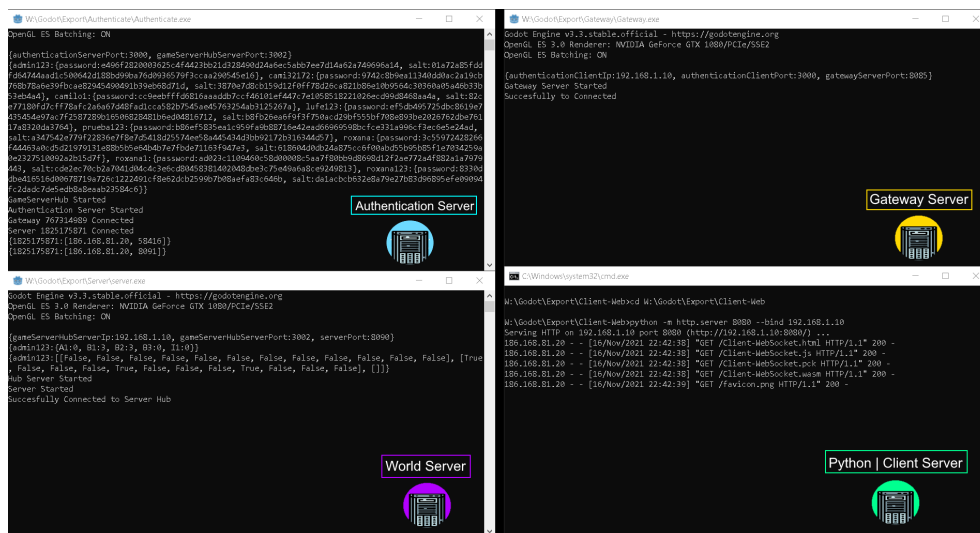


Figura 4.1: Servidores Elaborados en Godot Engine.Elaboración Propia.

A continuación se expondrá de manera detallada las funciones, datos almacenados e interacción que tienen cada uno de los servidores, para ello, se emplean los diagramas visualizados en las **figuras 4.2 y 4.3**.

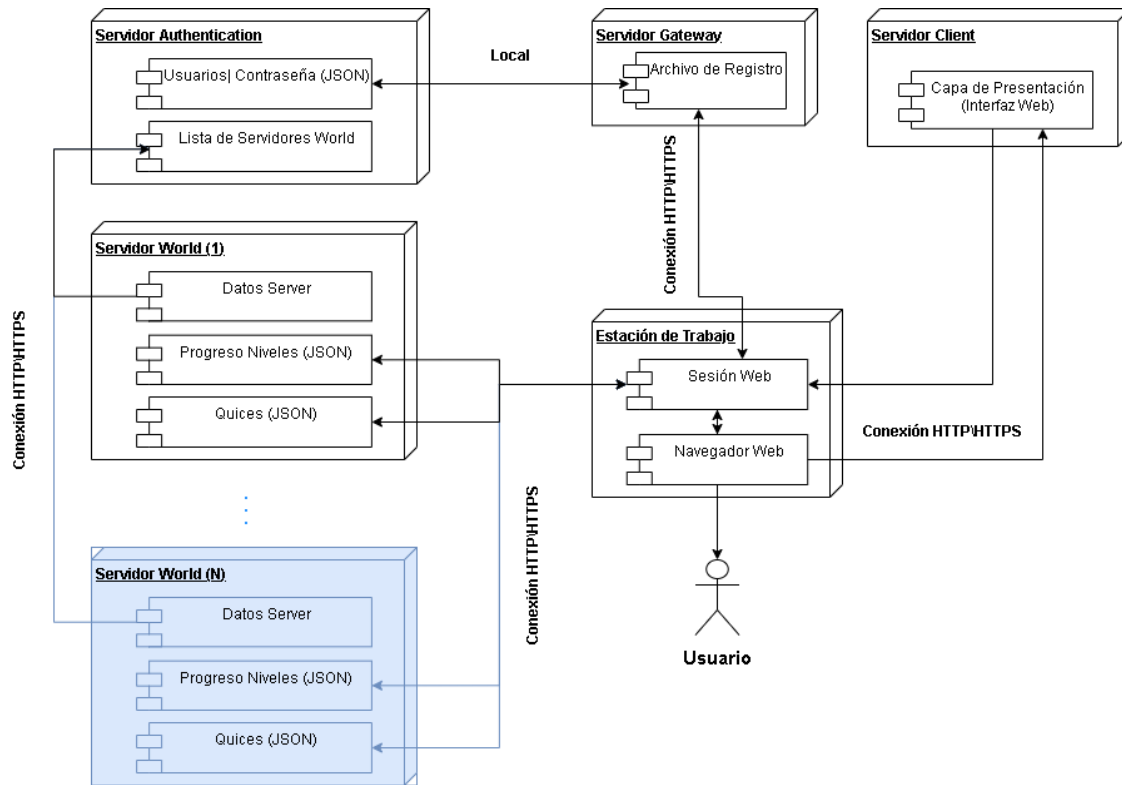


Figura 4.2: Diagrama de Despliegue UML - Servidores. Elaboración Propia.

- Client-Server (Python):** Como se puede observar en la **figura 4.2** el **Client-Server** es el servidor encargado de brindar la capa de presentación a cada usuario que realice una petición por medio de un navegador web a la dirección **ip pública:puerto abierto**, por medio de esta capa (**sesión de usuario**) el usuario puede interactuar con los elementos visuales del videojuego, más adelante del documento se detallan los elementos visuales (son todos los elementos que conforman el mundo del videojuego).
- Gateway Server:** Como se puede observar en las **figuras 4.2 y 4.3**, es el servidor encargado de recibir las primeras peticiones por parte del usuario dirigidas al **Authentication Server**, ya que en primera instancia el usuario debe crear o iniciar sesión con una cuenta almacenada en el servidor de autenticación. El **Gateway Server** proporciona seguridad, funcionando de manera similar a un cortafuegos, por lo que los datos sensibles de los usuarios estarán más protegidos por medio de este servidor a ataques cibernéticos.

- Authentication Server:** Como se puede observar en las **figuras 4.2 y 4.3**, es el servidor encargado de almacenar y validar los datos privados (**usuarios id y contraseñas**) de los usuarios que realizan peticiones por medio del **Gateway Server**, después de la validación de datos, el servidor le brinda al usuario una lista de direcciones ip y puertos de los **World Servers** que estén activos al momento de que un efectuar la petición, con el fin de continuar con el progreso almacenado en el servidor seleccionado por el usuario.
- World Server:** Como se puede observar en las **figuras 4.2 y 4.3**, es el servidor encargado de interactuar con el usuario por medio de peticiones de actualización y almacenamiento de los datos de la sesión actual, los datos que son almacenados en este servidor no son sensibles como los almacenados por el **Authentication Server**, ya que solo se almacena la cantidad de estrellas obtenidas por nivel y los resultados obtenidos por medio de los quices.

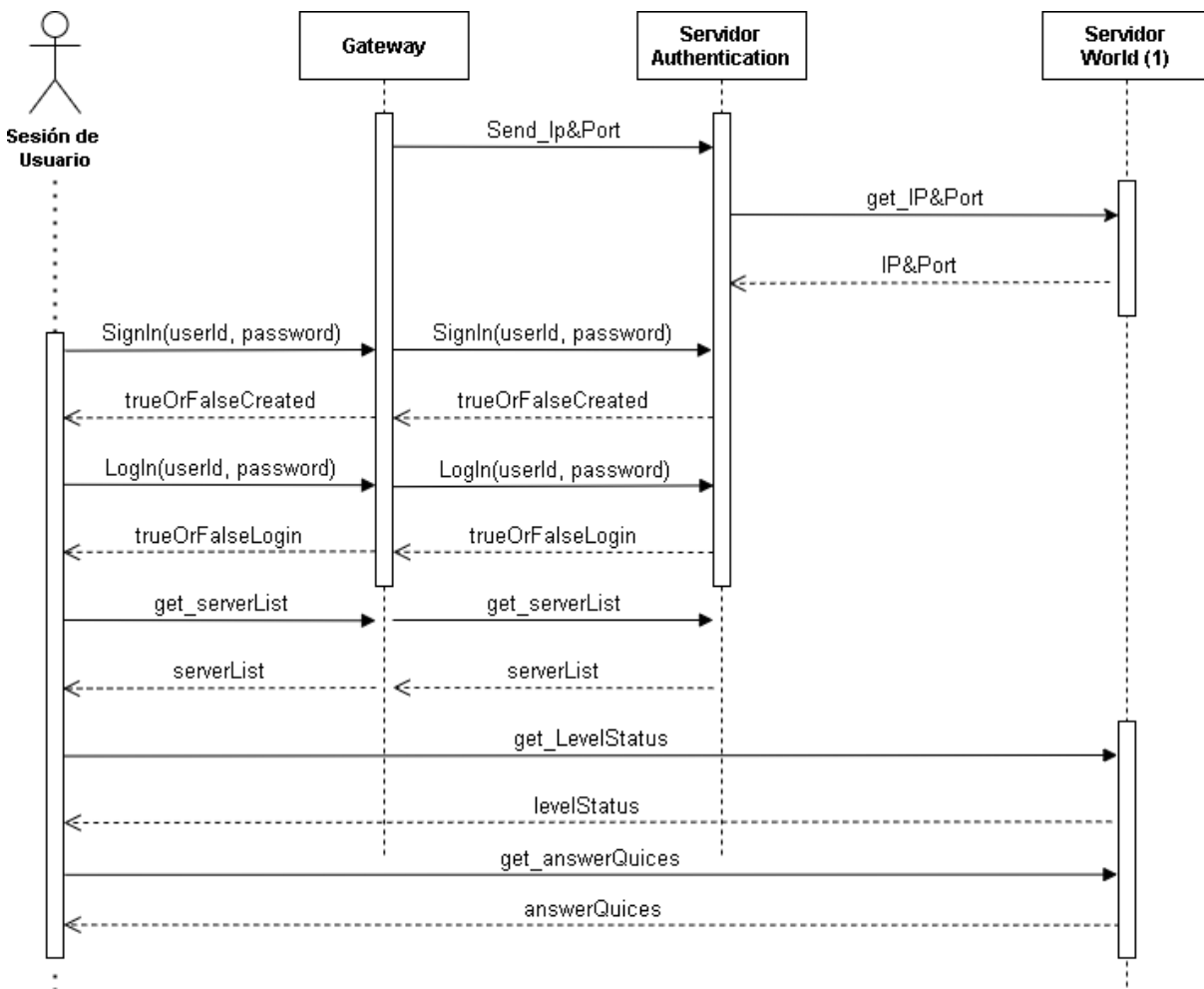


Figura 4.3: Diagrama de Secuencia UML - Servidores.Elaboración Propia.

4.3. Características Principales del Prototipo

Esta sección tiene como principal objetivo exponer el uso de las herramientas planteadas en la subsección **3.1.1. Tecnologías Planificadas**, ya que por medio de ellas fue posible la creación de los recursos empleados en los elementos visuales y auditivos del videojuego, permitiendo crear un prototipo más atractivo y con recursos personalizados que brindan la temática de restaurante en los niveles básicos.

4.3.1. Sistema de Diálogos Multilinguaje

Se determinó que para la siguiente etapa de pruebas, el prototipo debía ser multilingüe con el fin de que los usuarios no tuvieran problemas a la hora de leer y comprender los diálogos, textos e imágenes del videojuego; Godot Engine facilita la creación de videojuegos y aplicaciones multilingües mediante el uso de un sistema key:value, para emplear este sistema, se organizan los datos en una tabla de excel como se observa en la **figura 4.4 Tablas de Diálogos** y posteriormente se exporta en un archivo CSV; por último, se importa el archivo CSV directamente al motor gráfico y en configuración de proyecto se añade la ruta como se muestra en la **figura 4.5 Juegos e Internacionalización**, los archivos CSV deben guardarse en codificación UTF-8 sin una **marca de orden de bytes**.

	A	B	
1	id	en	es
2	SS1SEARCH	Search	Buscar
3	SS1CONNECT	Connect	Conectar
4	SS1EXIT	Exit	Salir
5	L1TITLE	PLE	PLE
6	L1USERNAMETITLE	Username	Nombre de Usuario
7	L1PASSWORDTITLE	Password	Clave de Usuario
8	L1LOGINBUTTON	Login	Iniciar sesión
9	L1CREATEBUTTON	Create Account	Crear una cuenta
10	L2PASSWORDREPEATTITLE	Repeat Password	Repite la Contraseña
11	L2CANCELBUTTON	Cancel	Cancelar
12	SS2PLAYERSTATISTICS	Player Statistics	Estadísticas del jugador
13	SS2PROGRESS	Progress:	Progreso:
14	SS2BASICCONCEPTS	Basic Concepts:	Conceptos Básicos:
15	SS2INTERMEDIATECONCEPT	Intermediate Concepts:	Conceptos Intermedios:
16	SS2ADVANCEDCONCEPT	Advanced Concepts:	Conceptos Avanzados:
17	STATS1	Player Statistics	Estadísticas del Jugador
18	TROPHYB	Basic	Básico

Figura 4.4: Tablas de Diálogos. Elaboración Propia.

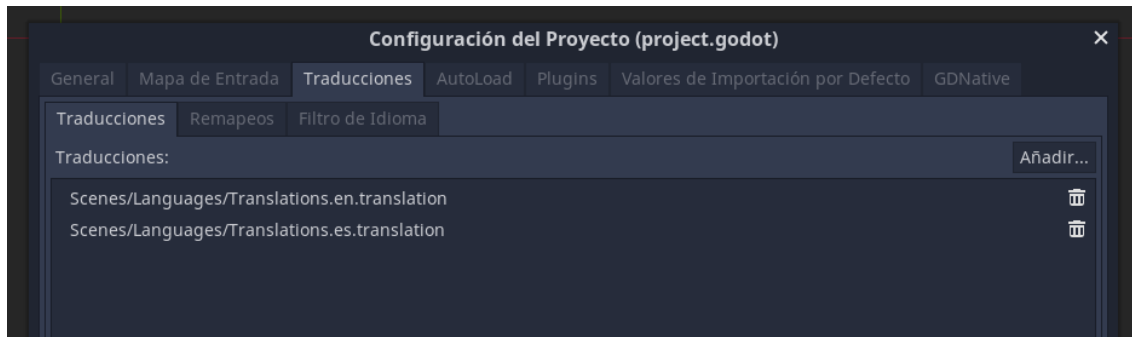


Figura 4.5: Juegos e Internacionalización. Elaboración Propia.

A continuación en la figura **figura 4.6 Interfaz LoginScreen Multilenguaje**, se muestra una comparativa entre los distintos estados en que puede ser visualizada la interfaz loginScreen; el estado “id” solamente puede ser visualizado en el ambiente de desarrollo, en cambio, los estados “es” y “en” solo pueden ser visualizados en la ejecución de la escena y además, son determinados por el objeto **TranslationServer**, el cual puede ser modificado a conveniencia del desarrollador.

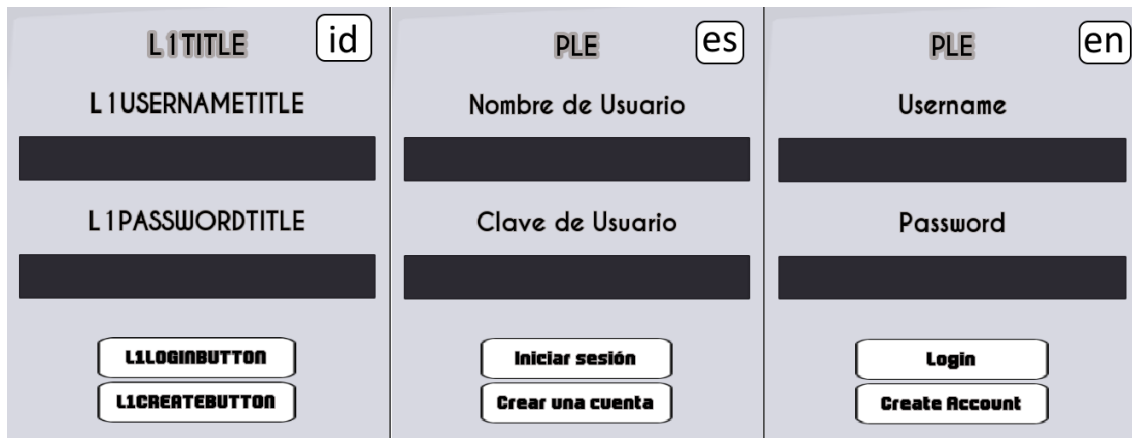


Figura 4.6: Interfaz LoginScreen Multilenguaje. Elaboración Propia.

4.3.2. Controles

Pensando en la manera más simple e intuitiva posible para que el usuario pudiera interactuar con el mundo virtual, se determinó que todas las interacciones por parte del usuario fueran por medio del cursor, de tal manera que solo tuviera que observar elementos específicos que dieran respuestas visuales y permitieran su respectiva interacción, dentro de los niveles el usuario puede acercar y alejar la pantalla con la rueda del ratón, además, el usuario puede cerrar el nivel actual con la tecla ESC.

4.3.3. Implementación de Mecánicas

Las mecánicas seleccionadas en la subsección **3.4.3.1. Mecánicas del Videojuego** fueron implementadas en función a las limitaciones obtenidas por parte de los controles mencionados en la subsección anterior, a continuación, se expondrán las mecánicas implementadas en cada uno de los niveles.

- **Mecánicas de Atención al Cliente**

Esta mecánica permite informar al usuario mediante el uso de un personaje con distintas expresiones faciales, textos, símbolos y partículas, si su pizza es correcta o incorrecta con respecto a la orden solicitada por el cliente; en la **figura 4.7**, se puede observar lo anteriormente mencionado.



Figura 4.7: Mecánicas de Atención al Cliente. Elaboración Propia.

- **Mecánicas de Drag and Drop**

Por medio de esta mecánica el usuario puede seleccionar cualquiera de los ingredientes que se encuentren dentro de la nevera y proceder a realizar alguna de las 3 acciones disponibles en el nivel (**cortar**, **amasar** y **cocinar**), además, facilita la creación de mecánicas futuras. En la **figura 4.8**, se realiza la acción de cortar con el ingrediente tomate, cada acción que se realice tiene un tiempo de espera de 5 segundos.



Figura 4.8: Mecánicas de Drag and Drop. Elaboración Propia.

- **Mecánicas de Armar Alimentos**

Mediante la implementación y adaptación de la **mecánica de drag and drop** basada en la temática del restaurante, el usuario tiene la capacidad de armar distintos tipos de pizzas con los ingredientes previamente preparados y almacenadas en la mesa de preparación como se puede observar en la **figura 4.9**.



Figura 4.9: Mecánicas de Armar Alimentos. Elaboración Propia.

- **Mecánicas de Automatización de Procesos**

Esta técnica se crea a partir de la necesidad de mostrarle al usuario de una manera práctica como la **ingeniería de líneas de productos de software** puede optimizar los procesos del restaurante, como se expone en la **figura 4.10**, mediante el uso de los “botones chefs” el usuario puede asignar un chef a cualquiera de las 3 acciones disponibles en el nivel (**cortar, amasar y cocinar**); cada 5 segundos, el chef preparará el ingrediente específico de su área.

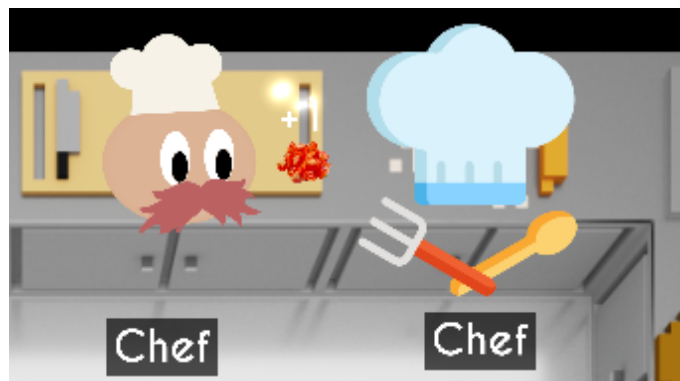


Figura 4.10: Mecánicas de Automatización de Procesos. Elaboración Propia.

4.3.4. Proceso de Creación de Personajes

Para el acercamiento y facilidad de comprensión de los diferentes textos descriptivos, informativos y guías presentados a lo largo del prototipo, se optó por la creación de personajes guías amigables y atractivos para el acompañamiento del usuario; por medio de los personajes, se pretende reforzar la aceptación e interiorización de la información que cada uno ofrece.

Como se puede observar en la **figura 4.11**, en el proceso de creación de personajes se empleó el concepto de variabilidad mencionado en la subsección **2.2.1.6**, tomando como base el mismo personaje, agregando diferentes tonos de piel, aplicaciones de cabello, color de ojos y vestuario, compartiendo las mismas características visuales de forma y color; uno de los personajes guía principal es el chef, el cual lleva las características anteriormente mencionadas, más la aplicación del vestuario típico de esta profesión (flipina y gorro).

Finalmente, la paleta de color que se empleó en el proceso de creación consta de los llamados colores pastel los cuales se caracterizan por su calidez y neutralidad permitiendo llevar la atención del usuario a la información que estos brindan sin distraerse o ser interrumpidos por colores llamativos.



Figura 4.11: Proceso de Creación de Personajes. Elaboración Propia.

4.3.5. Proceso de Animación

Godot Engine ofrece un apartado para realizar animaciones 2D, con un recurso conocido como **SpriteFrames**, como se expone en la **figura 4.12**, mediante el uso de este recurso se puede agrupar y enlistar de manera secuencial cada una de las animaciones que ejecutará el nodo **AnimatedSprite**, para posteriormente ser utilizadas con las funciones de la clase (`play()`, `stop()`, entre otras funciones); en este apartado, se pueden definir características específicas de la animación, como la velocidad de imágenes por segundo, definir un loop de ejecución, entre otras características.

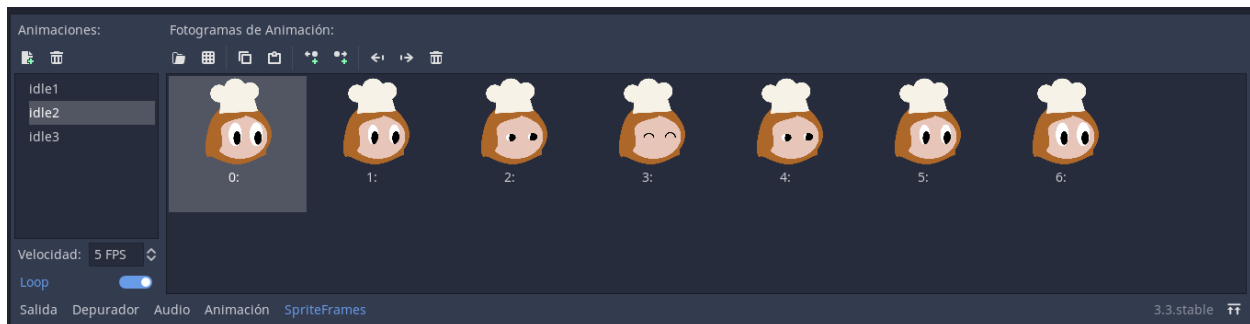


Figura 4.12: Proceso de Animación. Elaboración Propia.

4.3.6. Proceso de Creación de Escenarios

Mediante la herramienta **MagicaVoxel** fueron creados cada uno de los escenarios en los que transcurren las historias de los niveles básicos, debido a que este es un proyecto de ludificación en el que se buscan nuevas formas creativas, atractivas e interactivas de enseñar los conceptos de la ingeniería de líneas de productos de software a las personas interesadas en adquirir estos nuevos conocimientos; se realizaron varios prototipos de escenarios, que permitieran en primera instancia conservar la temática de un restaurante y en segunda, además, de tener la mejor distribución posible de objetos interactivables por parte de los usuarios.

En la **figura 4.13**, se puede apreciar un escenario tridimensional construido por medio de bloques; como anteriormente se mencionó, **MagicaVoxel** provee este tipo de bloques con el fin de poder modelar cualquier tipo de objeto que el diseñador desee realizar, además, brinda la posibilidad de añadir materiales a los bloques que contengan un color similar en el apartado de **Render**, para lograr un acabado profesional y realista.



Figura 4.13: Proceso de Creación de Escenarios. Elaboración Propia.

4.3.7. Comparativa con la Ingeniería de Líneas de Productos de Software

Uno de los recursos visuales que se empleó con el fin de realizar un proceso de ludificación efectivo fue el uso de infografías para explicar en cada nivel el respectivo grupo de conceptos asignado, a continuación, se expondrán cada una de las infografías y su respectivo grupo de conceptos; en los niveles 2 y 3, se emplearon diálogos al inicio de los niveles para completar la información de cada infografía.

Nivel 1 - Grupo de Conceptos 1(Detallar la Figura 4.14):

- **Concepto 1 - Activos:** Se representan los activos mediante los ingredientes utilizados en el nivel, además, se resalta la distinción entre activos comunes y variables.
- **Concepto 2 - Modelo de Características:** Se expone un modelo de características del restaurante, para que el usuario pueda comprenderlo de una manera más dinámica.
- **Concepto 3 - Salidas de Productos de Software:** Se presentan los cuatro productos que se realizaron en el nivel para dar a entender de una manera clara que son las salidas de productos de software.

Nivel 2 - Grupo de Conceptos 2(Detallar la Figura 4.15):

- **Concepto 1 - Variabilidad:** Se exponen cuatro posibles variaciones del “producto pizza” con sus respectivos ingredientes, dando a entender que es posible tener un producto A que no comparte los mismos ingredientes que un producto B.
- **Concepto 2 - Reúso:** En las mecánicas se ha reforzado la idea de que existen ingredientes indispensables para la creación de cualquier tipo de pizza, estos ingredientes son la masa, la salsa de tomate y el queso. En la infografía (**figura 4.15**), se muestra de manera explícita por medio de los tres ejemplos, cuáles son los ingredientes que están siendo reutilizados en cada uno de los productos.
- **Concepto 3 - Gestión de Variaciones en el Tiempo y el Espacio:** Para este último concepto se emplean los diálogos iniciales del nivel en el que se le explica al usuario que los artefactos (**ingredientes**) están sujetos a mantenimiento y evolución (**cambian en el tiempo**), además, la cantidad de productos puede aumentar(**cambian en el espacio**).

Nivel 3 - Grupo de Conceptos 3(Detallar la Figura 4.16):

- **Concepto 1 - Configurador:** Se presenta un modelo muy similar al modelo “**PLE FACTORY CONFIGURATOR**” del estándar ISO 26580 [**INCOSE, 2018**] utilizando la temática del restaurante, con el fin de que los usuarios puedan familiarizarse con futuros artículos de la **ingeniería de líneas de producto de software**. Como se puede apreciar en la infografía (**figura 4.16**), se han reutilizado los recursos visuales del nivel 1 para reforzar y unificar los conceptos vistos en el nivel.
- **Concepto 2 - Producto Intangible:** Se representa el hecho de que los productos son tangibles únicamente después de la etapa de producción y proceso (**Configurador**); mediante el uso del configurador y el producto D (**pizza pepperoni**).
- **Concepto 3 - Beneficios de la Ingeniería de Líneas de Producto de Software:** Se exponen textualmente cuatro beneficios que obtienen las empresas a la hora de implementar las líneas de productos de software.

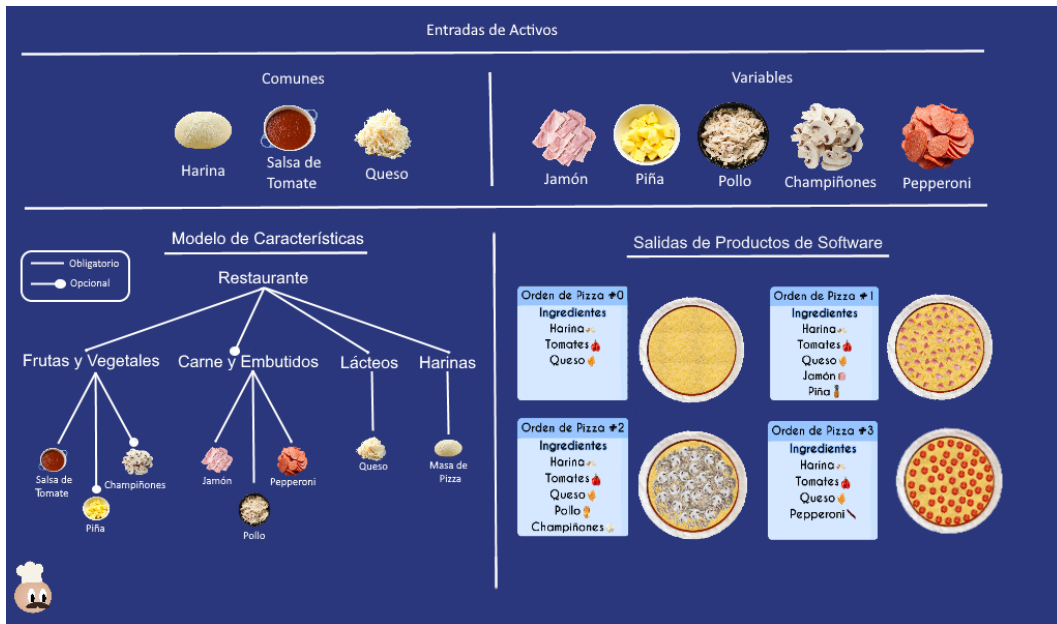


Figura 4.14: Infografía Grupo de Conceptos 1. Elaboración Propia.

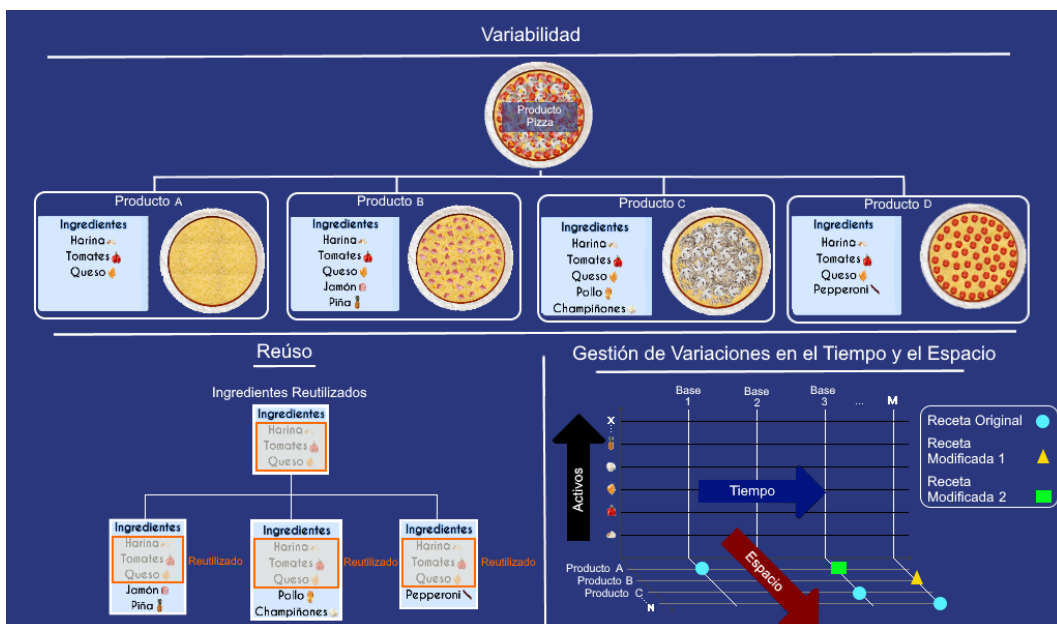


Figura 4.15: Infografía Grupo de Conceptos 2. Elaboración Propia.

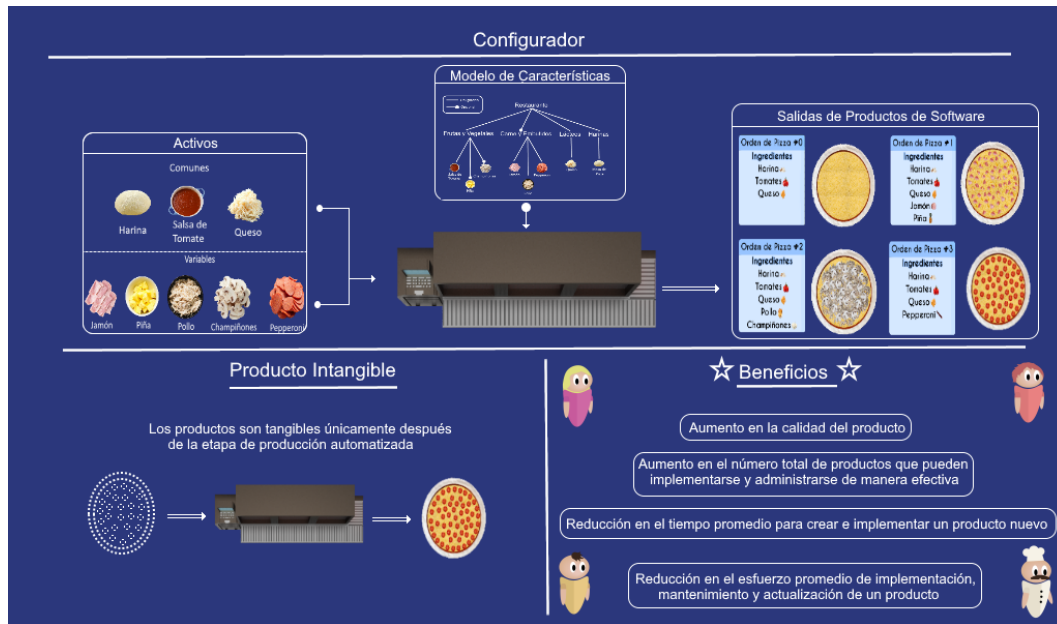


Figura 4.16: Infografía Grupo de Conceptos 3. Elaboración Propia.

4.3.8. Quices

Con el propósito de recibir retroalimentación de los conceptos aprendidos en cada nivel por parte de los usuarios y evaluar la efectividad de las técnicas de ludificación aplicadas a lo largo del prototipo, se decidió aplicar un quiz al finalizar el análisis de cada infografía. A continuación, en las **figuras 4.17, 4.18 y 4.19** se pueden visualizar cada una de las preguntas realizadas a los usuarios, las respuestas de los usuarios quedan almacenadas en un archivo JSON dentro del **Authentication Server**.

Cada una de las preguntas y respuestas del juego pueden ser modificadas directamente desde el excel **Translations**, esto es posible gracias a que como se explicó en la subsección **4.3.1. Sistema de Diálogos Multilenguaje**, cada una de las preguntas y respuestas están almacenadas en la escena como un id único, el cual puede variar dependiendo del archivo CSV que sea importado en el proyecto.

Quiz #1

1. Seleccione los Ingredientes Comunes:

A) Harina. B) Salsa de Tomate.
 C) Piña. D) Pollo.

2. Seleccione los Ingredientes Opcionales:

A) Jamón. B) Harina.
 C) Salsa de Tomate. D) Pepperoni.

3. Seleccione las salidas de productos del restaurante:

A) Champiñones. B) Orden de Pizza #1.
 C) Harina. D) Orden de Pizza #3.

Aceptar

Figura 4.17: Quiz - Grupo de Conceptos 1. Elaboración Propia.

Quiz #2

1. Seleccione las Variaciones del Producto Pizza:

A) Producto A (Queso,...). B) No existen variaciones.
 C) Producto C (Pollo,...). D) Todas las anteriores.

2. Seleccione los Ingredientes Reutilizables:

A) Champiñones. B) Pollo, Queso y Jamón.
 C) Pepperoni y Piña. D) Harina, Salsa de Tomate y Queso.

3. Seleccione las Características de la Gestión de Variaciones en el Tiempo y el Espacio:

A) Los activos principales evolucionan en el tiempo ('M'). B) Una colección de productos con características similares ('N').
 C) Los artefactos están sujetos a mantenimiento y evolución. D) Ninguna de las anteriores.

Aceptar

Figura 4.18: Quiz - Grupo de Conceptos 2. Elaboración Propia.



Figura 4.19: Quiz - Grupo de Conceptos 3. Elaboración Propia.

4.3.9. Conexión de Escenas

Godot Engine emplea un sistema de escenas que permite a los desarrolladores agrupar todos los nodos que se encuentren en ella, permitiendo llevar un control en cada uno de los módulos del proyecto. En la **figura 4.20**, se puede observar cada escena marcada por un color único y el flujo de eventos que se ejecutan de manera secuencial.

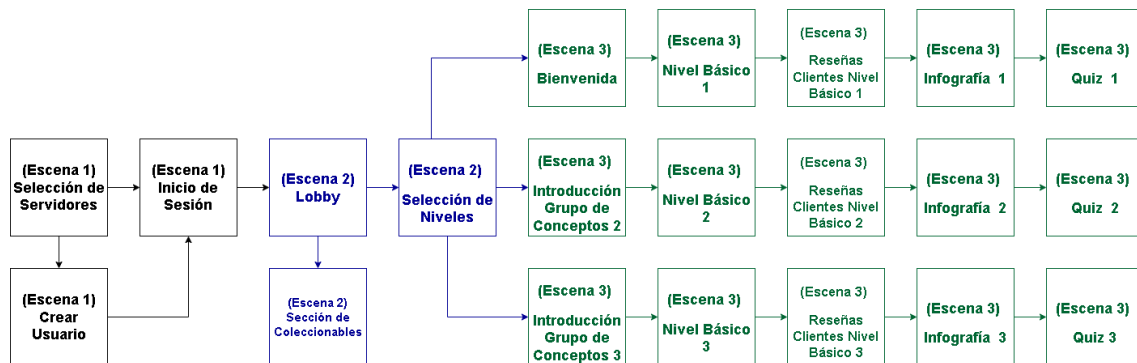


Figura 4.20: Conexión de Escenas 3. Elaboración Propia.

4.3.10. Prototipo

A continuación, se expondrán las imágenes del prototipo teniendo en cuenta la subsección anterior **conexión de escenas**, con el fin de llevar un orden en cada una de las imágenes, las escenas de infografía y quices ya han sido presentadas en subsecciones anteriores, por lo que no serán presentadas a continuación.

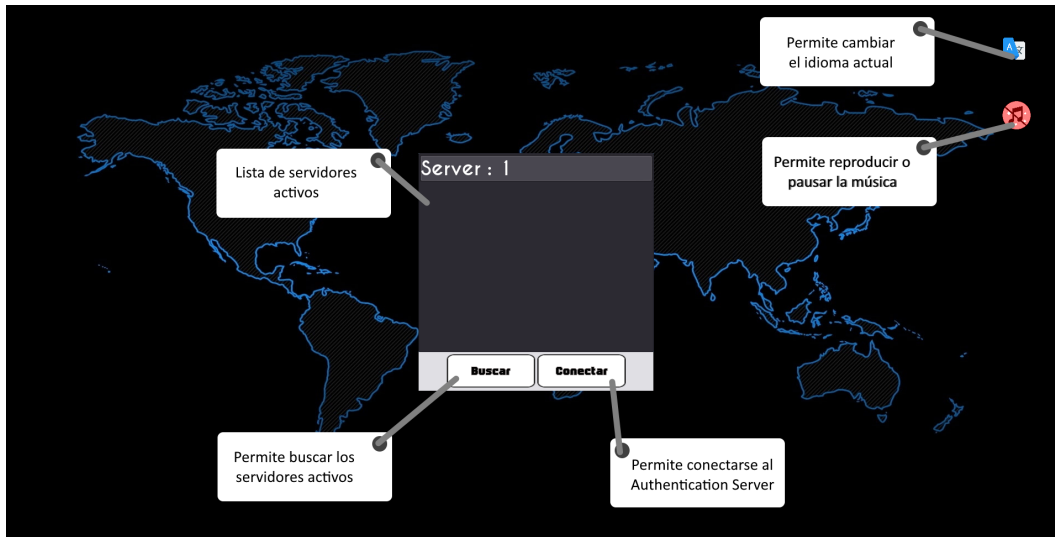


Figura 4.21: Selección de Servidores. Elaboración Propia.



Figura 4.22: Inicio de Sesión. Elaboración Propia.



Figura 4.23: Crear Usuario. Elaboración Propia.



Figura 4.24: Lobby. Elaboración Propia.

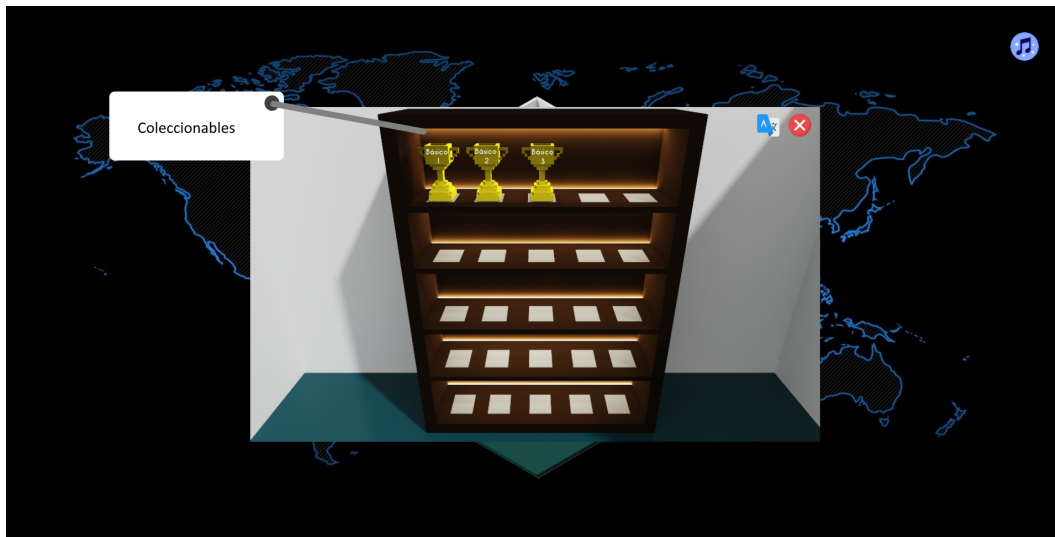


Figura 4.25: Sección de Coleccionables. Elaboración Propia.



Figura 4.26: Selección de Niveles. Elaboración Propia.



Figura 4.27: Bienvenida - Grupo de Conceptos 1. Elaboración Propia.



Figura 4.28: Nivel Básico 1. Elaboración Propia.



Figura 4.29: Reseñas Clientes Nivel Básico 1 . Elaboración Propia.



Figura 4.30: Introducción Grupo de Conceptos 2. Elaboración Propia.



Figura 4.31: Nivel Básico 2 . Elaboración Propia.

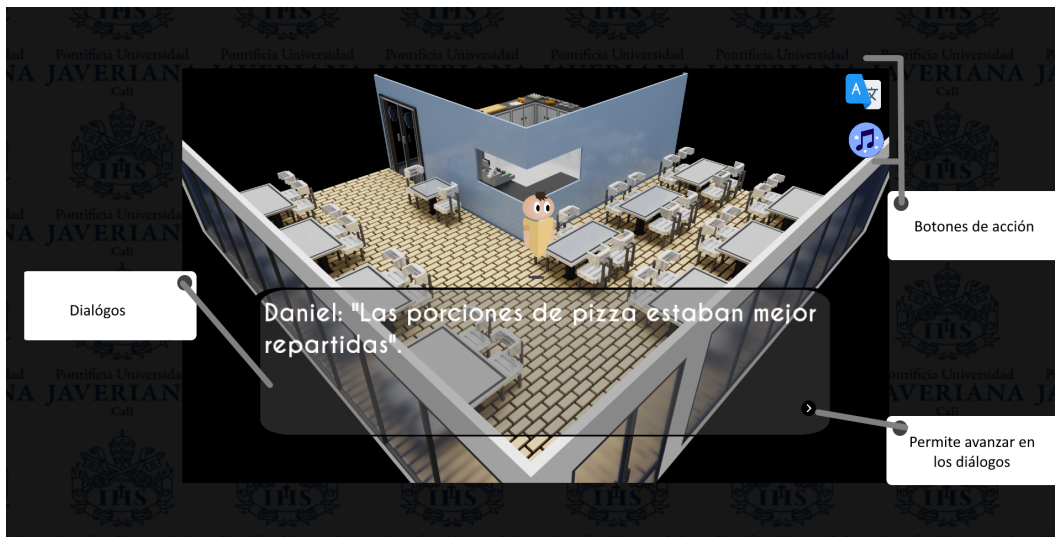


Figura 4.32: Reseñas Clientes Nivel Básico 2. Elaboración Propia.



Figura 4.33: Introducción Grupo de Conceptos 3. Elaboración Propia.



Figura 4.34: Nivel Básico 3 . Elaboración Propia.



Figura 4.35: Reseñas Clientes Nivel Básico 3. Elaboración Propia.

Pruebas de Usuario

En este capítulo se expondrán las pruebas de funcionalidad, usabilidad y percepción de utilidad realizadas con el fin mostrar la efectividad del prototipo desarrollado. Para poder realizar las pruebas anteriormente mencionadas, se abrieron los puertos del router necesarios, para que los usuarios evaluados pudieran acceder al prototipo de manera remota. La comunicación con los usuarios para las pruebas tuvo lugar en la herramienta Discord, como se muestra en la **figura 5.1**.

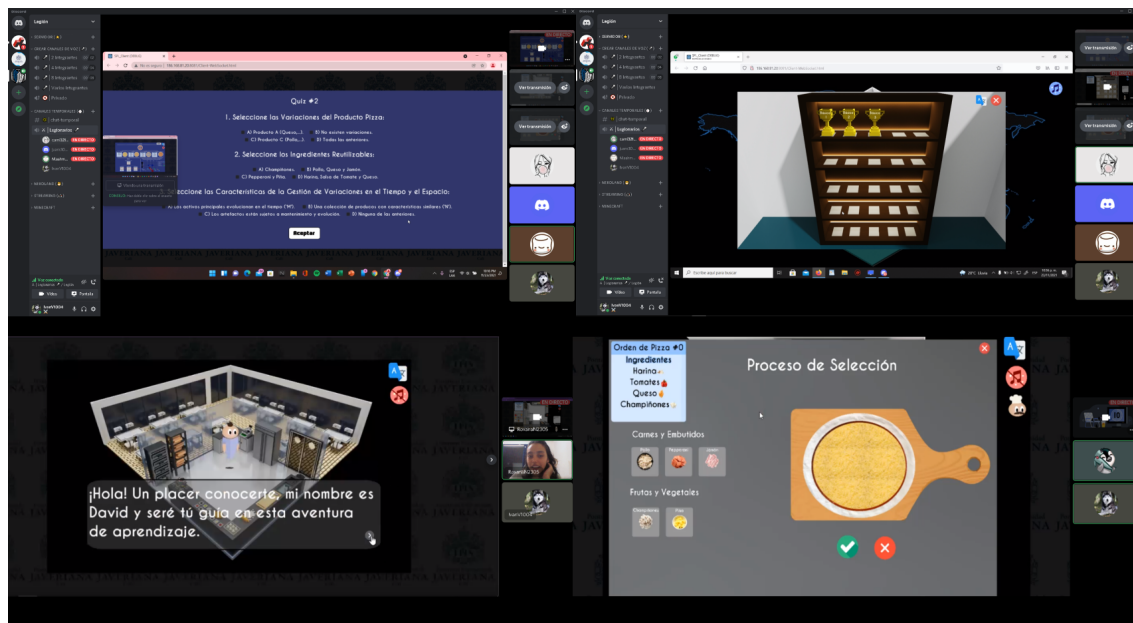


Figura 5.1: Discord como Medio de Comunicación. Elaboración Propia.

5.1. Pruebas Funcionales

Debido a que lo más importante en este proyecto es ofrecerle al usuario una experiencia gratificante a la hora de empezar a aprender los conceptos de la ingeniería de líneas de producto de software, se hace imprescindible, corregir los fallos de funcionamiento que puedan alterar la experiencia de juego; mediante la implementación de casos de prueba en los distintos eventos que componen cada escena, se puede llevar un control más riguroso de estos fallos y ser corregidos a tiempo.

5.1.1. Ejecución y Resultados

A continuación se detallarán los casos de prueba realizados en los distintos eventos; después de explicar cada evento, se encuentra una tabla que detalla su respectivo caso de prueba.

Escena 1

- **Selección de Servidores:** Este evento es el encargado de realizar la petición al **Gateway Server**, con el fin de obtener la lista de **World Servers** disponibles para su conexión.

Selección de Servidores		
Prerrequisitos	Resultado Esperado	Resultado Obtenido
Se hace necesario tener activo el Authentication Server , ya que como se explicó en la sección 4.1 es el encargado de guardar la lista de World Servers activos.	En el momento en que el usuario realice el llamado, el Gateway Server le provee una lista con los World Servers disponibles.	El evento funcionó como se esperaba. A pesar de recibir múltiples peticiones, el Gateway Server es capaz de resolverlas.

Tabla 5.1: Caso de Prueba 1 - Selección de Servidores. Elaboración Propia.

- **Crear Usuario:** Este evento es el encargado de entregar la información administrada por el usuario a el **Authentication Server** para ser almacenados y posteriormente servir como credenciales de autenticación de la cuenta.

Crear Usuario		
Prerrequisitos	Resultado Esperado	Resultado Obtenido
Se hace necesario haber seleccionado un servidor disponible de la lista de World Servers activos.	En el momento en que el usuario digite todas las credenciales solicitadas por la interfaz (usuario id, contraseña y verificar contraseña), el Gateway Server responde que el usuario ha sido creado de manera exitosa en el Authentication Server .	El evento funcionó como se esperaba. Cada uno de los usuarios que realizaron las pruebas de usabilidad, pudieron crear su cuenta de manera exitosa.

Tabla 5.2: Caso de Prueba 2 - Crear Usuario. Elaboración Propia.

- **Inicio de Sesión:** Este evento es el encargado de validar las credenciales del usuario y permitir la redirección del **Gateway Server** al **World Server** previamente seleccionado.

Inicio de Sesión		
Prerrequisitos	Resultado Esperado	Resultado Obtenido
Se hace necesario haber creado una cuenta en el Authentication Server .	En el momento en que el usuario digite las credenciales correctas, el Gateway Server le permite realizar la conexión directamente con el World Server previamente seleccionado.	El evento funcionó como se esperaba. Cada uno de los usuarios que realizaron las pruebas de usabilidad, pudieron acceder a sus respectivas cuentas.

Tabla 5.3: Caso de Prueba 3 - Inicio de Sesión. Elaboración Propia.

Escena 2

- **Lobby:** Este evento es el encargado de crear un espacio en el que el usuario puede acceder a la **Sección de Coleccionables** y a la **Selección de Niveles**.

Lobby		
Prerrequisitos	Resultado Esperado	Resultado Obtenido
Se hace necesario tener conexión con el World Server seleccionado.	En el momento en que el usuario realiza la conexión con el servidor, se carga automáticamente el espacio lobby.	El evento funcionó como se esperaba. Cada uno de los usuarios que realizaron las pruebas de usabilidad, pudieron acceder a su respectivo lobby.

Tabla 5.4: Caso de Prueba 4 - Lobby. Elaboración Propia.

- **Sección de Coleccionables:** Este evento es el encargado de cargar la información de cada uno de los trofeos obtenidos por el usuario.

Sección de Coleccionables		
Prerrequisitos	Resultado Esperado	Resultado Obtenido
Se hace necesario tener conexión con el World Server seleccionado.	En el momento en que el usuario presiona la estantería de trofeos, se visualizan los trofeos obtenidos a lo largo del juego.	El evento funcionó como se esperaba. Se realizaron varias pruebas con diferentes cuentas para verificar la carga de información de los trofeos.

Tabla 5.5: Caso de Prueba 5 - Sección de Coleccionables. Elaboración Propia.

- **Selección de Niveles:** Este evento es el encargado de cargar el progreso del usuario en el **World Server** seleccionado, cada nivel puede tener hasta un total de 3 estrellas, las cuales retroalimentan al jugador sobre su desempeño en el nivel realizado; para desbloquear los niveles inhabilitados se deben haber desbloqueado los niveles anteriores al inhabilitado.

Selección de Niveles		
Prerrequisitos	Resultado Esperado	Resultado Obtenido
Se hace necesario tener conexión con el World Server seleccionado.	En el momento en que el usuario presiona el mapa de niveles, se visualizan los niveles con sus respectivas estrellas.	El evento funcionó como se esperaba. Se realizaron varias pruebas con diferentes cuentas para verificar la carga de progreso de cada una de las cuentas.

Tabla 5.6: Caso de Prueba 6 - Selección de Niveles. Elaboración Propia.

Escena 3

Nota: Con la finalidad de optimizar el desarrollo y gestión de niveles, se crearon eventos conjuntos que son ejecutados con un id específico en cada nivel para visualizar la información necesaria.

- **Introducción | Grupo de Conceptos (N):** Este evento es el encargado de cargar todos los elementos visuales y diálogos (con el idioma actual).

Introducción Grupo de Conceptos (N)		
Prerrequisitos	Resultado Esperado	Resultado Obtenido
Se hace necesario que el usuario previamente seleccione el nivel desde el mapa de niveles.	En el momento en que el usuario presiona el botón de continuar, se carga el diálogo siguiente.	El evento funcionó como se esperaba. Se realizaron varias pruebas de carga de diálogos, cambiando el idioma actual.

Tabla 5.7: Caso de Prueba 7 - Introducción | Grupo de Conceptos (N). Elaboración Propia.

- **Nivel Básico 1:** Este evento es el encargado de ejecutar cada una de las mecánicas planeadas para el nivel (**Atención al Clien, Drag and Drop y Armar Alimentos**).

Nivel Básico 1		
Prerrequisitos	Resultado Esperado	Resultado Obtenido
Se hace necesario que el usuario previamente finalice todos los diálogos del evento anterior.	En el momento en que el usuario realiza cada una de las acciones permitidas en el nivel, obtiene un resultado esperad; cortar, amasar y cocinar genera un ingrediente específico almacenable en la tabla de preparación, la tabla de preparación permite crear pizzas o restablecer los ingredientes cuando el usuario se equivoque, el horno permite cambiar el estado de la pizza a cocida o quemada dependiendo del tiempo que la pizza lleve en el horn, la tabla de entrega genera uno de los 2 estados dependiendo de los ingredientes de la pizza.	El evento funcionó como se esperaba. Se realizaron cada una de las acciones permitidas en el nivel, con distintos ingredientes y forzando cada uno de los posibles estados que permitieran las acciones.

Tabla 5.8: Caso de Prueba 8 - Nivel Básico 1. Elaboración Propia.

- **Nivel Básico 2:** Este evento es el encargado de ejecutar cada una de las mecánicas planeadas para el nivel (**Atención al Cliente, Armar Alimentos y Automatización de Procesos**).

Nivel Básico 2		
Prerrequisitos	Resultado Esperado	Resultado Obtenido
Se hace necesario que el usuario previamente finalice todos los diálogos del evento anterior.	En el momento en que el usuario realiza cada una de las acciones permitidas en el nivel, obtiene un resultado esperado; las acciones de preparar pizza, hornear y entregar son las mismas del nivel 1(las acciones de cortar, amasar y cocinar no son requeridas en este nivel), al presionar el botón chef, se genera automáticamente un chef que se encargará de una tarea específica (cortar, amasar o cocinar).	El evento funcionó como se esperaba. Se realizaron cada una de las acciones permitidas en el nivel, con distintos ingredientes y forzando cada uno de los posibles estados que permitieran las acciones.

Tabla 5.9: Caso de Prueba 9 - Nivel Básico 2. Elaboración Propia.

- **Nivel Básico 3:** Este evento es el encargado de ejecutar cada una de las mecánicas planeadas para el nivel (**Atención al Clien, Armar Alimentos y Automatización de Procesos - Mejorado**).

Nivel Básico 3		
Prerrequisitos	Resultado Esperado	Resultado Obtenido
Se hace necesario que el usuario previamente finalice todos los diálogos del evento anterior.	En el momento en que el usuario realiza cada una de las acciones permitidas en el nivel, obtiene un resultado esperado; la acción de entregar es la misma que en el nivel 1(las acciones de preparar pizza, hornear, cortar, amasar y cocinar no son requeridas en este nivel), al presionar el configurador, se despliega una interfaz muy similar a la mesa de preparación, con la diferencia de que siempre genera pizzas cocidas.	El evento funcionó como se esperaba. Se realizaron cada una de las acciones permitidas en el nivel, con distintos ingredientes y forzando cada uno de los posibles estados que permitieran las acciones.

Tabla 5.10: Caso de Prueba 10 - Nivel Básico 3. Elaboración Propia.

- **Reseñas Clientes Nivel Básico (N):** Este evento es el encargado de cargar todos los elementos visuales y diálogos de los clientes (con el idioma actual).

Reseñas Clientes Nivel Básico (N)		
Prerrequisitos	Resultado Esperado	Resultado Obtenido
Se hace necesario que el usuario haya completado cada una de las órdenes del nivel.	En el momento en que el usuario presiona el botón de continuar, se carga el diálogo siguiente.	El evento funcionó como se esperaba. Se realizaron varias pruebas cargando una y otra vez los diálogos. Además, a medida que se cargaban los diálogos, se cambiaba el idioma actual.

Tabla 5.11: Caso de Prueba 11 - Reseñas Clientes Nivel Básico (N). Elaboración Propia.

- **Infografía (N):** Este evento es el encargado de presentar la infografía del nivel.

Infografía (N)		
Prerrequisitos	Resultado Esperado	Resultado Obtenido
Se hace necesario que el usuario previamente finalice todos los diálogos del evento anterior.	En el momento en que el usuario finaliza el último diálogo, se visualiza la infografía del nivel en progreso.	El evento funcionó como se esperaba. En cada nivel se visualiza su respectiva infografía.

Tabla 5.12: Caso de Prueba 12 - Infografía (N). Elaboración Propia.

- **Quiz (N):** Este evento es el encargado de cargar el quiz del nivel (con el idioma actual) y posteriormente enviar las respuestas al **World Server** seleccionado.

Quiz (N)		
Prerrequisitos	Resultado Esperado	Resultado Obtenido
Se hace necesario que el usuario previamente finalice la visualización de la infografía.	En el momento en que el usuario presiona el botón de siguiente, se visualiza el quiz de nivel; este quiz como se pudo observar en la subsección 4.3.5 , cuenta de respuestas múltiples y toda su información es almacenada en el World Server .	El evento funcionó como se esperaba. Se realizaron varias pruebas almacenando múltiples respuestas y posteriormente visualizaron desde el archivo JSON.

Tabla 5.13: Caso de Prueba 13 - Quiz (N). Elaboración Propia.

5.2. Pruebas de Usabilidad

Las pruebas de usabilidad se pueden definir como un método utilizado para evaluar los primeros diseños durante el proceso de desarrollo de un producto [Tarkkanen et al., 2015], en el área de videojuegos, estaría orientada a la facilidad de lectura de diálogos, interfaces gráficas atractivas, mecánicas intuitivas con un nivel de complejidad no tan alto, entre otras características.

5.2.1. Ejecución y Resultados

Para este proceso, se decidió realizar una primera prueba con 5 estudiantes de edades entre 21-24 años, de los cuales 3 de ellos tienen conocimientos en el área de programación y 2 en el área de diseño, con el fin de evaluar el impacto del prototipo en los jóvenes que no tienen experiencia previa con los conceptos de la ingeniería de líneas de producto de software. Como se indicó previamente al inicio del capítulo, mediante el uso de la herramienta discord se realizó todo el proceso de acompañamiento y control de la experiencia de cada usuario, con el fin de recibir comentarios constructivos que permitan mejorar la calidad del prototipo y verificar si es posible cumplir con los objetivos propuestos en cada nivel; además, comprobar si la experiencia en los niveles, los diálogos e infografías son realmente útiles para comprender el grupo de conceptos a los que se enfrentan en cada nivel los usuarios.

Cada usuario interactuó con los distintos eventos que componen cada escena, brindando aportes significativos que fueron corregidos posteriormente a su experiencia, varios aportes giraban en torno a etiquetar los ingredientes y objetos que se encontraban en los distintos eventos, para facilitar la búsqueda de los mismos. En la **tabla 5.14** se exponen los aportes realizados por los usuarios.

Usuario(Área de Conocimiento)	Aportes
Usuario 1 (Programación)	“Las mecánicas son sencillas e intuitivas”.
Usuario 2 (Programación)	“Me parece muy bien que pueda cambiar el lenguaje cuando lo necesite”.
Usuario 3 (Programación)	“Los diálogos deberían tener más ejemplos ”.
Usuario 4 (Diseño)	“Los escenarios y personajes combinan muy bien con la temática de la pizzería”.
Usuario 5 (Diseño)	“No poner varias mayúsculas en los títulos, porque visualmente se ve desordenado”.

Tabla 5.14: Usabilidad. Elaboración Propia.

Para la segunda etapa se realizó la prueba a él experto en la ingeniería de líneas de productos de software Danilo Beuche , el cual expuso los siguientes aportes:

- Sería bueno recibir retroalimentación al finalizar un nivel (sonoro y auditivo) de manera positiva o negativa dependiendo el caso.
- Mejorar el tipo de pregunta dentro de los quices.
- La interfaz y el diseño en general es muy llamativo.

5.3. Pruebas de Percepción de Utilidad

Se puede entender la percepción de utilidad, como el impacto que genera el prototipo o sistema en el área de interés. A continuación, en la **tabla 5.15** se exponen los comentarios hechos por los usuarios con respecto a la experiencia obtenida después de realizar cada uno de los 3 niveles de conceptos básicos.

Usuario(Área de Conocimiento)	Opiniones
Usuario 1 (Programación)	“A pesar de que los conceptos eran complejos. Los videojuegos ayudan a entenderlos de una manera más fácil”.
Usuario 2 (Programación)	“Es increíble cómo cada nivel era más automatizado”.
Usuario 3 (Programación)	“El ambiente del videojuego es muy tranquilo y llama mucho la atención”.
Usuario 4 (Diseño)	“El uso de infografías me pareció correcto, ayudan mucho”.
Usuario 5 (Diseño)	“Muy buena idea que un juego divierta y enseñe al mismo tiempo”.

Tabla 5.15: Percepción de Utilidad. Elaboración Propia.

Para esta segunda etapa, de manera similar a la anterior, se realizó la prueba a él experto en la ingeniería de líneas de productos de software Danilo Beuche el cual expuso los siguientes aportes:

- Debería de haber una mayor variedad de alimentos y combinaciones, ya que permitiría enfrentar a los jugadores a problemas relacionados con las líneas de productos de software.
- Implementar casos en los que un cliente realiza una orden que no está en el menú.
- Implementaría el prototipo nivel por nivel, con el fin de observar los comentarios, aspectos y problemas que los usuarios puedan percibir.

5.3.1. Resultados de los Quices

En la **figura 5.2**, se exponen las respuestas obtenidas de los usuarios por medio del **World Server**. Como anteriormente se mencionó, la cantidad total de participantes en estas pruebas fue de 5 personas que cumplían con el perfil (**no eran expertos en los conceptos de la ingeniería de líneas de producto software y estaban dispuestos a aprenderlos**).

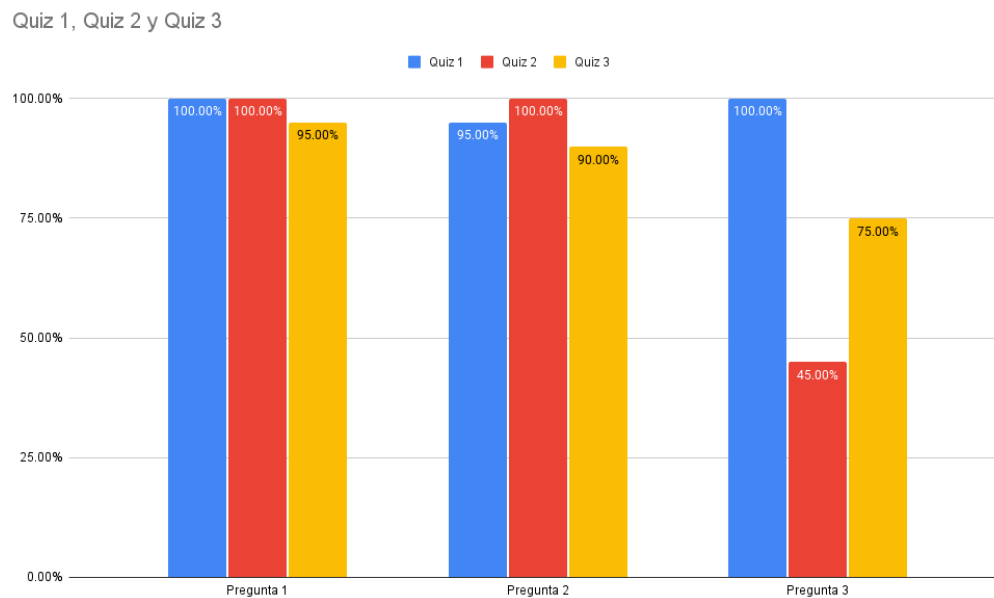


Figura 5.2: Diagrama de Comparativa de Quices.Elaboración Propia.

Conclusiones

Este capítulo está enfocado en analizar y verificar, que los objetivos propuestos se hayan cumplido de manera exitosa, contar la experiencia vivida a lo largo del proceso de desarrollo del prototipo, resaltar la importancia y el impacto que pueden tener las nuevas técnicas de aprendizaje, como la ludificación; exponer la contribución que genera el prototipo a la ingeniería de líneas de producto de software, indicar las futuras etapas del proyecto y finalmente detallar todos los recursos realizados en el proyecto.

6.1. Objetivos Alcanzados

- **Objetivo General:** El objetivo general fue alcanzado. Mediante la aplicación de técnicas de la metodología de ludificación, fue posible crear una herramienta que permita aprender los conceptos introductorios de las líneas de producto de software para las personas que no poseen experiencia previa en esta área del conocimiento.
- **Objetivo 1 (Identificar):** Este objetivo específico fue alcanzado. Se identificaron y seleccionaron las técnicas de ludificación que fueron empleadas en el diseño del prototipo, cuatro técnicas mecánicas (**progreso de niveles, premios, misiones y feedback**) y tres técnicas dinámicas (**progreso, reconocimiento o estatus y diversión**), en la sección 3.5 se detalla cada técnica.
- **Objetivo 2 (Seleccionar):** Este objetivo específico fue alcanzado. Se seleccionaron y agruparon los conceptos introductorios que abarco el prototipo en tres grupos, primer grupo (**entradas de activos, modelo de características y salidas de productos de software**), segundo grupo (**variabilidad, reuso y gestión de variaciones en el tiempo y espacio**) y tercer grupo (**configurador, producto intangible y beneficios de la ingeniería de líneas de producto de software**), en la sección 2.2.1 y 2.2.4 se detallan cada uno de los conceptos.
- **Objetivo 3 (Implementar):** Este objetivo específico fue alcanzado. Se implementó una herramienta web basada en técnicas de ludificación y que permite asociar conceptos introductorios de las líneas de producto de software. Esta herramienta fue inspirada en la historia de un chef que desea mejorar los ingresos del restaurante adaptando los conceptos de la ingeniería de líneas de producto a los procesos que se llevan a cabo dentro del restaurante, en los capítulos 3 y 4 se detalla el proceso de diseño y desarrollo de todo el prototipo, desde el desarrollo de la historia, hasta la conexión de escenas.

- **Objetivo 4 (Evaluar):** Este objetivo específico fue alcanzado. Se evaluó la herramienta mediante pruebas de funcionalidad, usabilidad y percepción de utilidad con la ayuda de usuarios expertos e inexpertos. En el capítulo 5 se detalla los resultados obtenidos en cada una de las pruebas, cabe resaltar que fueron positivos.

6.2. Experiencia Personal

A lo largo del desarrollo del documento, la implementación de cada uno de los servidores, el desarrollo de interfaces, prototipado de escenarios, creación de personajes, búsqueda y selección de pistas de audio, implementación de mecánicas y finalmente la creación del prototipo, se puede reconocer que ha sido un proceso de gran formación personal, donde se obtuvieron conocimientos de la ingeniería de líneas de producto de software, técnicas de ludificación aplicadas, programación, entre otros. Se hace una invitación a continuar con estas nuevas técnicas lúdicas que permitan estimular el proceso de aprendizaje de una manera novedosa.

6.3. Impacto de la Ludificación

Uno de los puntos que más se destaca en el proceso de desarrollo de todo el documento, es la búsqueda e implementación de técnicas de ludificación, que permitan facilitar el proceso de enseñanza de los conceptos básicos de la ingeniería de líneas de producto de software. Se hace necesario crear nuevos modelos de aprendizaje que permitan cautivar a las personas y sean estimuladas de manera positiva en todo el proceso de aprendizaje, como se explica en la **subsección 3.4.6 Satisfacción de los Jugadores**, la motivación juega un papel fundamental en las emociones y las emociones afectan de manera directa al individuo.

6.4. Contribución

El prototipo diseñado permite que pueda ser divulgado y conceda un acceso más inmediato a los procesos de capacitación de los conceptos de la ingeniería de líneas de producto de software, en el ejercicio de aprendizaje de una forma más dinámica e innovadora, ayudando al profesor o monitor a realizar un proceso de enseñanza más interactivo mediante el uso del prototipo.

6.5. Futuras Etapas del Proyecto

El proyecto no termina aquí, pues todavía faltan conceptos intermedios y avanzados que pueden ser implementados en versiones futuras del prototipo; además de mecánicas innovadoras y nuevas técnicas de ludificación. Uno de los comentarios recibidos por parte de los jurados fue cambiar el ejemplo utilizado en los quices para evaluar que efectivamente se están aprendiendo los conceptos asociados a la ingeniería de líneas de producto de software y no el ejemplo aplicado al restaurante, por lo que se tendrá en cuenta para una futura actualización del prototipo; adicionalmente se debe

de tener en cuenta todos los aportes realizados de los usuarios y el experto en futuras versiones del prototipo.

6.6. Recursos

A continuación, se detalla cada uno de los recursos creados para la sustentación del documento, cada recurso puede ser descargado desde el enlace de Google Drive. Los proyectos realizados en Godot Engine también se encuentran desde el enlace de GitHub:

- 4 proyectos realizados en el motor gráfico Godot Engine (Authenticate, Gateway, Server y Client).
- 5 Ejecutables para los sistemas operativos Windows (Authenticate, Gateway, Server, Client-Web, Server2 para pruebas).
- Hoja de excel con todos los ids del prototipo(multilingüe).

Enlace Google Drive

https://drive.google.com/drive/folders/1_hL00h8ZdokvDDdpBjM0yQC1gUAkbghA?usp=sharing

Enlace GitHub

<https://github.com/IvanJC1004/SPLE/tree/master>

Bibliografía

- [Azanza et al., 2021] Azanza, M., Montalvillo, L., and Díaz, O. (2021). 20 years of industrial experience at SPLC.
- [Birk, 2002] Birk, A. (2002). Three case studies on initiating product lines: Enablers and obstacles.
- [Bosch, 2001] Bosch, J. (2001). Software product lines: Organizational alternatives. In *Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001*, pages 91–100.
- [Caballero, 2016] Caballero, A. R. (2016). Estudio de la gamificación de una empresa para incentivar la motivación.
- [Clements and Northrop, 2001] Clements, P. and Northrop, L. M. (2001). *Software Product Lines: Practices and Patterns*. Addison-Wesley Professional, 1st edition.
- [Corporation, 2021a] Corporation, V. (2021a). Steam plataforma de distribución digital de videojuegos.
- [Corporation, 2021b] Corporation, V. (2021b). Steam plataforma de distribución digital de videojuegos.
- [Crytek, 2021] Crytek (2021). Cryengine motor de videojuegos.
- [dotPDN LLC, 2021] dotPDN LLC (2021). Paint.NET editor de imágenes y fotografía.
- [Dusink and van Katwijk, 1995] Dusink, L. and van Katwijk, J. (1995). Reuse dimensions. *SIG-SOFT Softw. Eng. Notes*, 20(SI):137–149.
- [Ephtracy, 2021] Ephtracy (2021). MagicaVoxel editor de voxel artwork.
- [Frank J and Rommes, 2007] Frank J, van der Linden, K. S. and Rommes, E. (2007). Software product lines in action: The best industrial practice in product line engineering. *Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering*.
- [Games, 2021] Games, E. (2021). Unreal Engine motor de videojuegos.
- [González and Blanco, 2008] González, C. and Blanco, F. (2008). Interacción, motivación y emociones con videojuegos. *Revista Electrónica Teoría de la Educación. Educación y Cultura en la Sociedad de la Información.*, 9.
- [Inc, 2020a] Inc, S. (2020a). Gamedesign documentation builder.
- [Inc, 2020b] Inc, S. (2020b). Gamedesign documentation builder.

- [INCOSE, 2018] INCOSE (2018). Primer ISO 26580 INCOSE - Feature-based systems and software product line engineering. Technical report, Product Line Engineering International Working Group.
- [Issa and Isaias, 2015] Issa, T. and Isaias, P. (2015). *Usability and Human Computer Interaction (HCI)*, pages 19–36. Springer London, London.
- [Kotter and Cohen, 2002] Kotter, J. P. and Cohen, D. S. (2002). *The Heart of Change: real-life stories of how people change their organizations*. Harvard Business School Press, Boston, Massachusetts.
- [Krueger, 2006a] Krueger, C. (2006a). Chapter 3. In *Tutorial: New Methods Behind the New Generation of Software Product Line Success Stories*, pages 39–59.
- [Krueger and Clements, 2013] Krueger, C. and Clements, P. (2013). Systems and software product line engineering with biglever software gears. In *Proceedings of the 17th International Software Product Line Conference Co-located Workshops, SPLC '13 Workshops*, page 136–140, New York, NY, USA. Association for Computing Machinery.
- [Krueger, 1992] Krueger, C. W. (1992). Software reuse. *ACM Computing Surveys*, 24:131–183.
- [Krueger, 2006b] Krueger, C. W. (2006b). Introduction to the emerging practice of software product line development. *Methods & Tools - Fall 2006*.
- [Krueger and Clements, 2018] Krueger, C. W. and Clements, P. (2018). Feature-based systems and software product line engineering with gears from BigLever. pages 1–4.
- [Lanman et al., 2013] Lanman, J., Darbin, R., Rivera, J., Clements, P., and Krueger, C. W. (2013). The challenges of applying service orientation to the U.S. Army’s live training software product line. In *Proceedings of the 17th International Software Product Line Conference on - SPLC '13*, page 244, New York, New York, USA. ACM Press.
- [Lanman et al., 2011] Lanman, J., Kemper, B., Rivera, J., and Krueger, C. W. (2011). Employing the Second Generation Software Product-line for Live Training Transformation. In *Proceedings of The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*, number 11083, pages 1–12.
- [Martínez, 2017] Martínez, C. M. (2017). Estudio de gamificación en una empresa para mejorar la fidelización de los clientes.
- [Northrop, 2008] Northrop, L. (2008). Software product lines essentials.
- [Palencia, 2018] Palencia, J. C. H. (2018). Prácticas de ludificación en el desarrollo de software representadas en el núcleo de semat.

- [Perlman, 1994] Perlman, G. (1994). Book review: Human-computer interaction, by jenny preece, yvonne rogers, helen sharp, david benyon, simon holland, and tom carey (addison-wesley, 1994). *SIGCHI Bull.*, 26(4):82–85.
- [Pohl et al., 2005] Pohl, K., Böckle, G., and van der Linden, F. (2005). *Software Product Line Engineering*. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Prakash and Rao, 2015] Prakash, E. C. and Rao, M. (2015). *International Series on Computer Entertainment and Media Technology Transforming Learning and IT Management through Gamification*. Springer International Publishing.
- [Ravaja et al., 2004] Ravaja, N., Salminen, M., Holopainen, J., Saari, T., Laarni, J., and Järvinen, A. (2004). Emotional response patterns and sense of presence during video games: Potential criterion variables for game design. *ACM International Conference Proceeding Series*, 82:339–347.
- [Studios, 2021] Studios, O. (2021). Godot motor de videojuegos.
- [Tarkkanen et al., 2015] Tarkkanen, K., Harkke, V., and Reijonen, P. (2015). Are we testing utility? analysis of usability problem types. In Marcus, A., editor, *Design, User Experience, and Usability: Design Discourse*, pages 269–280, Cham. Springer International Publishing.
- [Team, 2021] Team, A. (2021). Audacity editor y grabador de audio multipista.
- [Technologies, 2021] Technologies, U. (2021). Unity motor de videojuegos.
- [Vallejo and Martín, 2017] Vallejo, D. and Martín, C. (2017). *Arquitectura del Motor*. Bubok.
- [Vargas and Silva, 2019] Vargas, I. R. P. and Silva, J. A. R. (2019). La ludificación como herramienta pedagógica: algunas reflexiones desde la psicología. *Revista de Estudios y Experiencias en Educación*, 18:161–175.
- [Villay, 2016] Villay, J. I. (2016). Videojuego para apoyar el proceso de descripción estática en la terapia del lenguaje aplicada a niños.
- [Ysmar Vianna and Tanaka, 2014] Ysmar Vianna, Maurício Vianna, B. M. and Tanaka, S. (2014). Como reinventar empresas a partir de jogos.
- [Óscar Díaz García and Gonzalez, 2007] Óscar Díaz García and Gonzalez, S. T. (2007). Líneas de producto software. In *Fábricas de Software: Experiencias, tecnologías y organización*, pages 61–79. Ra-Ma.