



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
Mención en Tecnologías de la Información

**Software didáctico de apoyo al
aprendizaje de técnicas básicas de
Virtualización**

Autor:

Silvia Arias Herguedas



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
Mención en Tecnologías de la Información

**Software didáctico de apoyo al
aprendizaje de técnicas básicas de
Virtualización**

Autor:

Silvia Arias Herguedas

Tutores:

Dr. Benjamín Sahelices Fernández

Alma María Pisabarro Marrón

*A mi familia, por enseñarme el valor del esfuerzo.
A David, por apoyarme cada día.*

Agradecimientos

Me gustaría agradecer, en primer lugar, a mis tutores, Benjamín Sahelices y Alma Pisabarro, por la oportunidad de realizar este proyecto y por ayudarme en la realización de este mediante sus explicaciones y aclaraciones que han sido muy útiles.

En segundo lugar, a mis amigos y compañeros de clase por la ayuda proporcionada y sus puntos de vista, así como, la compañía y ánimos que en muchos casos han sido muy necesarios. Sin ellos, este camino no sería lo mismo.

Por último, y más importante, a mi familia porque sin saber de qué iba el tema me han apoyado siempre, sobre todo en los momentos malos, animándome a seguir.

Resumen

Las máquinas virtuales, desde que se comenzaron a utilizar hasta la actualidad, han supuesto un gran avance al permitir crear nuevos sistemas teniendo otros como anfitriones, así como ejecutar programas de forma que parezca que se ejecutan sobre una máquina real. Esto supone que para el mismo hardware se puedan tener varios sistemas ejecutándose al mismo tiempo, cosa que en un sistema físico real estaría limitado. A su vez, las máquinas virtuales han hecho posible desarrollar la computación en la nube, algo de gran importancia hoy en día.

Para poder traducir una dirección en un entorno virtual, es necesaria la compartición de la memoria física del sistema y asignarla a la memoria física de la máquina virtual. Para ello se realiza un mapeado a dos niveles: de memoria virtual a física y de física a memoria de la máquina.

El estudio de este proceso a nivel detallado puede ser complejo para los estudiantes. A fin de ayudarles, se considera la realización de una herramienta *online* que les sirva de apoyo en sus estudios y en la que se introducen técnicas de gamificación para alcanzar distintos objetivos como la motivación del alumnado y la facilitación del aprendizaje. Como resultado, se ha desarrollado una aplicación web para el aprendizaje *online* que, a su vez, integra un simulador.

Este trabajo está incluido dentro del Proyecto de Innovación Docente, “Gamificación en la Enseñanza Universitaria“, que tiene como objetivo facilitar el proceso de adquisición de conocimiento del alumno.

Abstract

Virtual machines, since they started to be used until nowadays, have implied a great progress by allowing to create new systems using others as hosts, as well as running programs so it seems they are running on a real machine. This implies that for the same hardware it is possible to have various systems running at the same time, which would be limited on a real physical system. At the same time, virtual machines have made possible the development of cloud computing, of great importance today.

In order to translate an address on a virtual environment, it is necessary to share the system's physical memory and allocate it to the virtual machine's physical memory. To achieve that, a two level mapping is done: virtual memory to physical and physical to machine memory.

The study of this process on a detailed level might be complex for students. In order to help them, the development of an online tool that supports them in their studies and in which gamification techniques are introduced is considered, with the objective of achieving some goals such as students' motivation and learning facilitation. As a result, a web application has been developed that includes online learning and a simulator as well.

This project is included on a "Teacher Innovation Project" called *Gamificación en la Enseñanza Universitaria*, whose purpose is to make easier the student's process of acquiring knowledge.

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	1
1.2.1. Objetivos personales	2
1.3. Estructura de la memoria	3
2. Contexto del proyecto	5
2.1. Virtualización de la memoria	5
2.2. Sistemas semejantes	8
2.3. Gamificación	11
2.3.1. Juego serio	12
2.3.2. Elementos de la gamificación	12
2.4. Aplicación en este trabajo	14
3. Planificación	17
3.1. Planificación del trabajo	17
3.2. Planificación inicial	17
3.2.1. Distribución temporal	18
3.2.2. Análisis de riesgos	18
3.3. Estimación de costes	20
3.3.1. Recursos	20
3.3.2. Presupuesto	22
3.4. Historias de usuario	23
3.5. Seguimiento del proyecto	31
3.5.1. Sprint 1	32
3.5.2. Sprint 2	34
3.5.3. Sprint 3	35
3.5.4. Sprint 4	36
3.5.5. Sprint 5	37
4. Análisis y diseño del proyecto	39
4.1. Análisis	39
4.1.1. Modelo de dominio	39
4.1.2. Diagramas de secuencia	40

4.1.3. Diagrama de clases	44
4.2. Diseño	45
4.2.1. Arquitectura	45
4.2.2. Diagrama de despliegue	45
4.2.3. Patrones aplicados	46
4.2.4. Diagramas de secuencia	48
4.2.5. Diagrama de clases	50
4.2.6. Modelo de base de datos	51
5. Implementación	53
5.1. Codificación	53
5.1.1. Simulador	53
5.1.2. Web	55
5.2. Herramientas utilizadas	55
5.2.1. Unity	55
5.2.2. Netbeans	56
5.2.3. Apache Tomcat	57
5.3. Pruebas realizadas	58
6. Conclusiones y trabajo futuro	65
Apéndices	69
A. Acrónimos	69
B. Manual de instalación	71
B.1. Instalación de Java	71
B.2. Instalación de Netbeans	71
B.3. Base de datos con Apache Derby	72
C. Manual de uso	73
D. Preguntas de los tests	83
D.1. Nivel 1	83
D.2. Nivel 2	86
D.3. Nivel 3	88
E. Contenido del CD	93
Bibliografía	97

Índice de figuras

2.1. Mapeado a dos niveles en un entorno virtual.	5
2.2. Virtualización de la memoria usando EPT.	6
2.3. Simulador de traducción.	7
2.4. Simulador de memoria virtual con Paginación.	10
2.5. Simulador de jerarquías de memoria.	10
2.6. Simulador de traducción.	11
2.7. Elementos de la gamificación.	12
3.1. Diagrama de Gantt del trabajo.	17
3.2. Burndown chart global.	32
3.3. Prototipo inicial del simulador.	33
3.4. Simulador tras sprint 1.	33
3.5. Burndown chart sprint 1.	34
3.6. Burndown chart sprint 2.	35
3.7. Burndown chart sprint 3.	36
3.8. Burndown chart sprint 4.	37
3.9. Burndown chart sprint 5.	38
4.1. Modelo de dominio web.	39
4.2. Modelo de dominio simulador.	40
4.3. Ver Tests - Análisis.	41
4.4. Ver Ranking - Análisis.	42
4.5. Elegir dirección - Análisis.	43
4.6. Reiniciar simulación - Análisis.	43
4.7. Diagrama de clases Web - Análisis.	44
4.8. Diagrama de clases Simulador - Análisis.	45
4.9. Diagrama de despliegue	46
4.10. Modelo-Vista-Controlador	47
4.11. Patrón DAO	48
4.12. Ver Test - Diseño	48
4.13. Ver Ranking - Diseño	49
4.14. Elegir dirección - Diseño.	49
4.15. Reiniciar simulación - Diseño.	50
4.16. Diagrama de clases Web - Diseño	50

4.17. Diagrama de clases Simulador - Diseño	51
4.18. Esquema entidad-relación.	51
B.1. Java DB.	72
B.2. Opciones al ejecutar un comando.	72
C.1. Página de inicio de sesión.	73
C.2. Página de registro.	74
C.3. Menú superior.	74
C.4. Página principal	75
C.5. Página de teoría: conceptos básicos.	75
C.6. Página de teoría: virtualización de memoria.	76
C.7. Página de tests.	76
C.8. Resultados de un test.	77
C.9. Detalle de pregunta correcta.	77
C.10. Detalle de pregunta incorrecta.	77
C.11. Detalle de pregunta no contestada.	77
C.12. Página del simulador.	78
C.13. Perfil de usuario.	78
C.14. Inicio del simulador.	79
C.15. Estado del simulador tras elegir la dirección.	80
C.16. Detalle de flechas para avanzar y retroceder.	80
C.17. Detalle de botón reiniciar	80
C.18. Estado final del simulador.	81

Índice de tablas

2.1. Elementos gamificados del proyecto.	14
3.1. Riesgos del proyecto.	19
3.2. Plan de contingencia para cada riesgo.	20
3.3. Ordenador portátil: características.	22
3.4. Monitor: características.	22
3.5. Presupuesto	23
3.6. HU01 - Crear tablas de las estructuras.	23
3.7. HU02 - Elegir dirección a traducir.	24
3.8. HU03 - Flechas para ir hacia delante y retroceder.	24
3.9. HU04 - Pasos.	24
3.10. HU05 - Texto con explicación.	24
3.11. HU06 - Mostrar flechas en el proceso.	25
3.12. HU07 - Mostrar etiquetas en las tablas.	25
3.13. HU08 - Numerar celdas.	25
3.14. HU09 - Mostrar puntero a CR3.	25
3.15. HU10 - Mostrar puntero a EPT.	26
3.16. HU11 - Mostrar animación dirección elegida.	26
3.17. HU12 - Borrado.	26
3.18. HU13 - GPA vacío.	26
3.19. HU14 - Reiniciar simulación.	27
3.20. HU15 - Mostrar animación dirección GPA a traducir.	27
3.21. HU16 - Mostrar etiquetas en las flechas.	27
3.22. HU17 - Mostrar etiquetas en las direcciones.	27
3.23. HU18 - Número de proceso.	28
3.24. HU19 - Terminar simulación al no haber espacio.	28
3.25. HU20 - Login.	28
3.26. HU21 - Logout.	28
3.27. HU22 - Registro.	29
3.28. HU23 - Ver Ranking.	29
3.29. HU24 - Ver Teoría.	29
3.30. HU25 - Tests.	29
3.31. HU26 - Ver soluciones del test.	30
3.32. HU27 - Ver simulador	30

3.33.HU28 - Ver perfil.	30
3.34.HU29 - Número de preguntas acertadas.	30
3.35.HU30 - Nivel	31
3.36.HU31 - Subir nivel.	31
5.1. TEST 01.	58
5.2. TEST 02.	58
5.3. TEST 03.	58
5.4. TEST 04.	58
5.5. TEST 05.	59
5.6. TEST 06.	59
5.7. TEST 07.	59
5.8. TEST 08.	59
5.9. TEST 09.	59
5.10.TEST 10.	60
5.11.TEST 11.	60
5.12.TEST 12.	60
5.13.TEST 13.	60
5.14.TEST 14.	61
5.15.TEST 15.	61
5.16.TEST 16.	61
5.17.TEST 17.	61
5.18.TEST 18.	61
5.19.TEST 19.	62
5.20.TEST 20.	62
5.21.TEST 21.	62
5.22.TEST 22.	62
5.23.TEST 23.	62
5.24.TEST 24.	63
5.25.TEST 25.	63

Capítulo 1

Introducción

1.1. Motivación

Dado el tiempo limitado con el que cuentan los docentes y la cantidad de materia a la hora de impartir clase, se presenta un problema con respecto a qué parte del temario hay que darle más importancia. Este es el caso de la asignatura de "Estructura de sistemas operativos", donde no se imparte todo lo que se desearía sobre virtualización. El proceso de virtualización de la memoria virtual es un problema complejo que no da tiempo a abordar durante el transcurso de la asignatura, es por esto que se decide crear este trabajo enmarcado dentro del Proyecto de Innovación Docente "Gamificación en la Enseñanza Universitaria".

A fin de hacer que los estudiantes conozcan más sobre este tema, se piensan distintas formas de conseguir que les resulte llamativo, dejando de lado las formas clásicas de aprendizaje. El principal reto que nos encontramos es el de crear una herramienta que sea fácil de usar y útil para que aprendan sobre el tema y profundicen en él, al ser de tanta importancia actualmente.

Sumado a esto, cada alumno tiene su ritmo de aprendizaje, es interesante permitir a cada uno que dedique el tiempo que necesite, y en el lugar que pueda, sin estar sujeto a horarios de clases ni a una asistencia obligatoria. Esta característica, convierte este trabajo en una herramienta muy útil en la educación *online*, puntera hoy en día.

1.2. Objetivos

El objetivo principal que se pretende alcanzar mediante la realización de este proyecto es ofrecer a los estudiantes una herramienta que les permita conocer mejor los conceptos relacionados con la virtualización de la memoria virtual y el proceso que se realiza al traducir una dirección en un entorno virtual a una dirección de la máquina real. Se trata de un problema complejo, que, mediante la herramienta a desarrollar, se pretende mostrar de la forma más sencilla posible. Para alcanzar este fin, se creará un software didáctico accesible para los alumnos que les dé la oportunidad de comprender los

conceptos, realizar tests e interactuar con un simulador.

Se abordan tres partes que se pueden diferenciar en la aplicación de forma clara:

- **Teoría:** se expondrá a los alumnos los fundamentos teóricos básicos para conocer los conceptos y familiarizarse con ellos. Estos se dividirán en dos apartados: virtualización de memoria y conceptos básicos.
- **Tests:** el alumno podrá realizar preguntas de tipo test para comprobar que ha comprendido los conceptos estudiados previamente. Estos tests se dividen por niveles, aumentando la dificultad de las preguntas y permitiendo al alumno subir de nivel a medida que va resolviendo cuestiones correctamente.
- **Simulación:** se permitirá al usuario interactuar con un simulador para comprender el proceso de traducción de una dirección de un proceso en un entorno virtual a una dirección física de la memoria de la máquina real. Entre sus opciones, permitirá al alumno avanzar o retroceder y reiniciar a valores iniciales. A su vez, contará con una breve explicación en cada paso para facilitar su comprensión.

Además, es importante que la aplicación sea usable y motive a los alumnos a utilizarla. Se intentan conseguir dos objetivos principales aplicando técnicas de gamificación:

- **Facilitar el aprendizaje:** hacer que el alumno aprenda de forma fácil mediante las siguientes técnicas:
 - **Distintos ritmos de aprendizaje:** cada estudiante puede seguir el ritmo indicado por él mismo según sus capacidades.
 - **Realimentación:** los resultados los pueden ver de forma inmediata para comprobar sus aciertos y errores.
 - **No hay penalización por fallos:** el restar puntos por fallos puede desmotivar al alumno y ocultar el objetivo principal que es del aprender.
- **Motivación:** conseguir llamar la atención e interés del alumno mediante:
 - **Competitividad:** la aplicación permitirá al alumno subir de nivel según su número de respuestas acertadas haciendo que aumente su competitividad consigo mismo. A su vez, contará con un gráfico social de alumnos con las mejores puntuaciones, lo que permitirá que concursen entre ellos para intentar superarse.
 - **Recompensas:** a final de curso o cuando el docente lo decida se repartirán recompensas para los alumnos que más alto queden en el ranking.

1.2.1. Objetivos personales

A la hora de realizar este proyecto el principal objetivo que intento abarcar es el de aprender nuevas tecnologías y crear un proyecto que sea interesante para los estudiantes y pueda ayudarles en su

formación académica. Uno de los principales retos es el de crear un proyecto desde cero partiendo de una idea que hay que desarrollar. Para este fin se necesitan englobar los siguientes aspectos:

- Estudiar la documentación y bibliografía relacionada con el tema en cuestión. En este caso, el estudio sobre la virtualización de la memoria virtual y técnicas de gamificación.
- Aprender a utilizar el motor Unity de desarrollo de videojuegos a nivel básico para poder implementar la simulación.
- Recordar cómo se implementa un entorno web con herramientas ya conocidas como Java y Apache Tomcat.
- Aprender a documentar un proyecto desde su fase inicial.

1.3. Estructura de la memoria

El documento está estructurado en distintas partes que contienen desde la explicación de conceptos teóricos necesarios para la comprensión de este proyecto hasta pruebas realizadas sobre el trabajo final. También se incluyen los respectivos manuales y anexos necesarios.

En primer lugar, se comienza por explicar en qué contexto se desarrolla el proyecto, exponiendo herramientas que puedan ser similares y qué novedades se aportan con respecto a estas. También se explican los conceptos de gamificación y técnicas que se pueden aplicar a este trabajo así como sus ventajas.

Se continúa desarrollando el análisis de la aplicación. Partiendo de las historias de usuario, se expone una solución arquitectónica y los distintos diagramas de secuencia y clases para documentarla.

Más adelante, se comentan las herramientas y tecnologías que se han utilizado y el motivo por el que se han elegido estas y no otras. También las pruebas que se han ido realizando incrementalmente mientras se implementaba el software.

Por último, se añaden los apéndices necesarios que comprenden desde los manuales de instalación y uso hasta las preguntas tests que se recopilaron para la parte web del proyecto.

Capítulo 2

Contexto del proyecto

2.1. Virtualización de la memoria

El problema de la virtualización de la memoria, en un sistema tradicional, es relativamente sencillo. El sistema operativo mantiene mapeos de memoria virtual a memoria de la máquina usando tablas de páginas. Este proceso constituye mapeado de un solo nivel, es decir, de dirección virtual a máquina real.

Sin embargo, en un entorno virtual es necesario un mapeado a dos niveles. En primer lugar, de memoria virtual a física y, en segundo, de memoria física a memoria de la máquina. Esto requiere la compartición de la memoria física del sistema y asignarla dinámicamente a la memoria física de la máquina virtual. El sistema operativo invitado controla el mapeado de direcciones virtuales a físicas de la máquina virtual pero no puede acceder de forma directa a la memoria real. Es el VMM o el *Virtual Machine Manager* el encargado de mapear de la memoria física del invitado a la actual de la máquina.

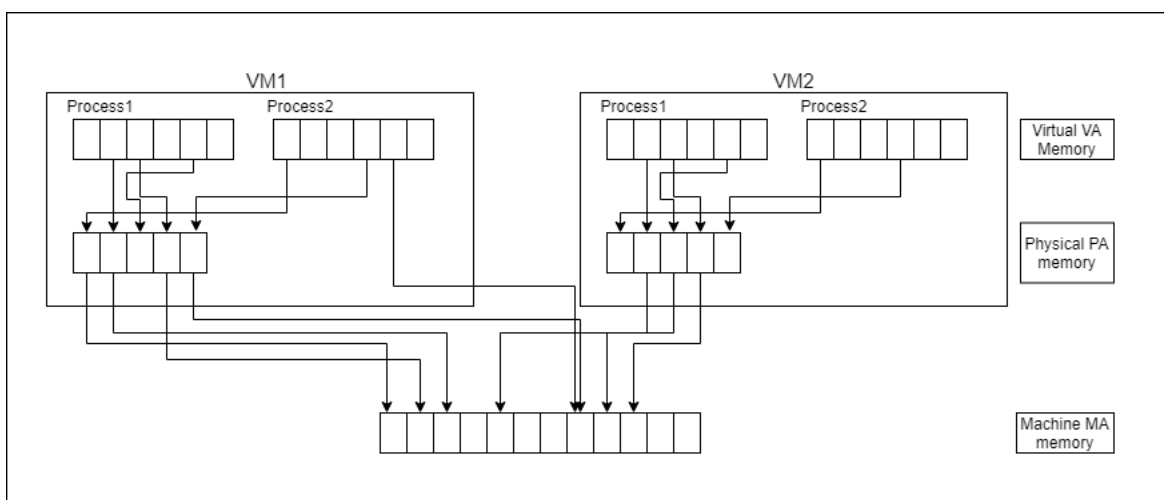


Figura 2.1: Mapeado a dos niveles en un entorno virtual.

En la figura 2.1, cada proceso tiene una dirección en la memoria virtual de la máquina que se traduce a una dirección de la memoria física. El siguiente paso de traducción a memoria de la máquina lo realiza el VMM.

Para evitar en algunos casos tener que hacer dos niveles de traducción, los procesadores utilizan el TLB o *Translation Lookaside Buffer* que funciona como una caché, mapeando directamente de la memoria virtual a memoria máquina. Por otro lado, el VMM utiliza tablas de páginas extendidas o EPT para traducir direcciones físicas de la máquina virtual a direcciones físicas del *host*.

El problema a simular es el de traducción de una dirección virtual en un entorno virtual a una dirección física en la máquina real. Para ello se utiliza el proceso descrito en la figura 2.2, donde se realiza una traducción con paginación a cuatro niveles.

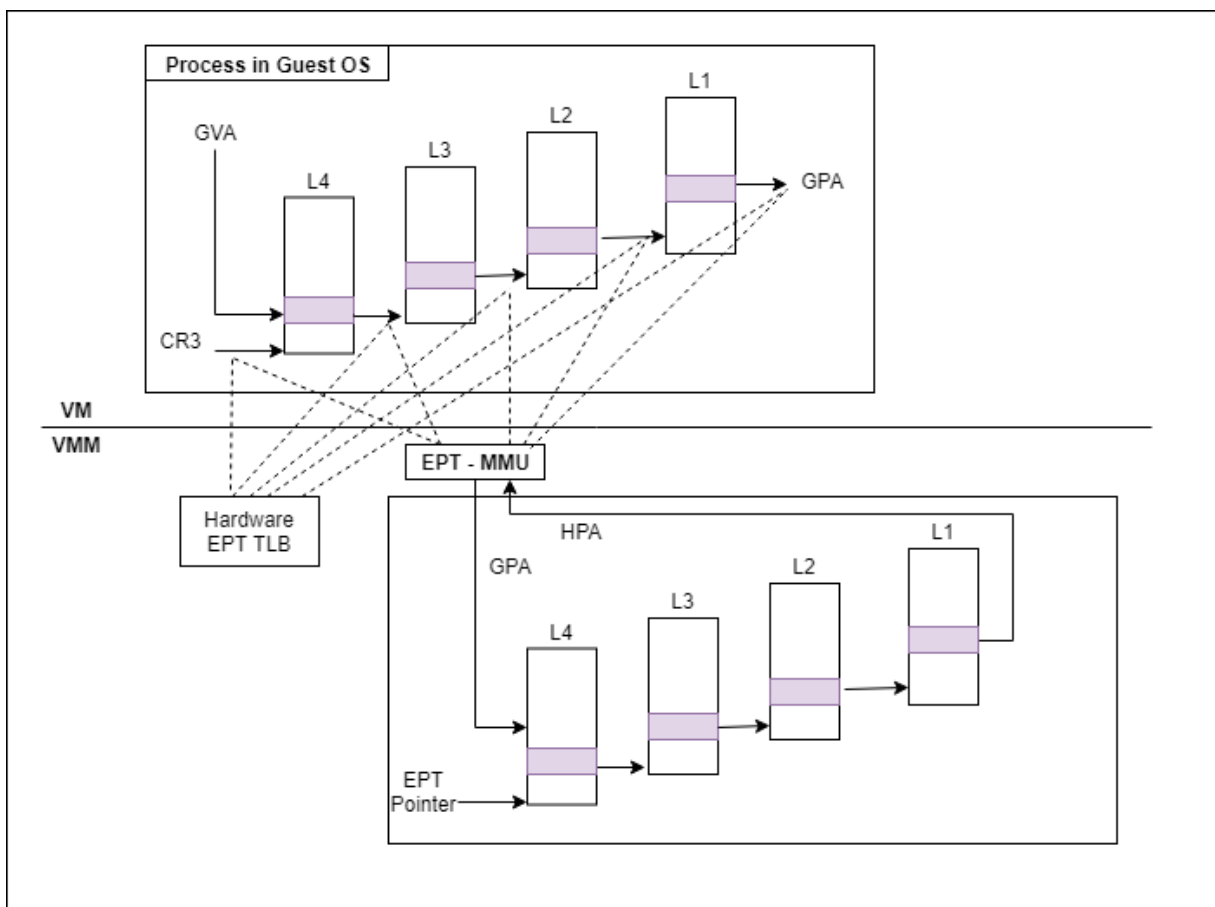


Figura 2.2: Virtualización de la memoria usando EPT.

En la máquina virtual, la CPU genera una dirección virtual, GVA, que será la que se va a traducir. Mediante el registro CR3 se localiza la tabla de páginas de cuarto nivel. Sin embargo, el número de marco de la memoria física de la máquina invitado, que indica dónde está esta tabla de páginas, es una dirección de tipo GPA, es decir, física del invitado. Por tanto, es necesario traducirla a una dirección física del *host*, HPA. En este caso, al utilizar el TLB, primero se va a buscar una entrada en esta tabla. Si ya se encuentra una entrada con esta dirección, la devuelve directamente. Si no, es necesario utilizar

Contexto del proyecto

las EPT. Mediante un puntero, se obtiene la EPT de nivel cuatro, que, en la entrada correspondiente, tendrá escrito el número de marco que apunta a la EPT de nivel tres. Así sucesivamente, hasta que se obtiene la dirección traducida en el último nivel. Este proceso, hay que repetirlo en los cuatro niveles de tablas de páginas, traduciendo cada dirección física del invitado para obtener la del *host*. Como resultado, se obtiene un marco que corresponde con la dirección física concordante con la virtual.

Con el objetivo de simplificar este problema complejo, se realiza el simulador utilizando paginación a dos niveles y sin utilizar un TLB, aunque se plantea su uso para una mejora futura.

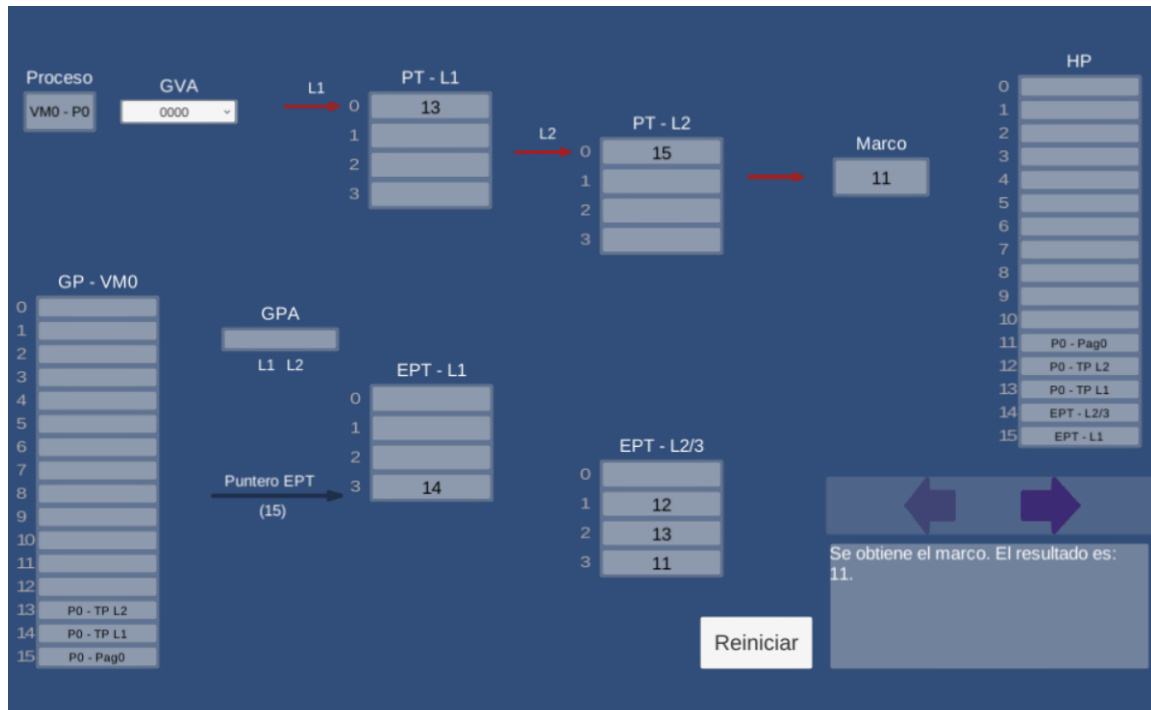


Figura 2.3: Simulador de traducción.

En la figura 2.3 se muestra cómo quedaría el simulador tras traducir una dirección. El problema que se intenta resolver mediante este programa es el que se ha explicado anteriormente pero simplificado en algunos aspectos:

- Se trata de una máquina virtual con un proceso que genera una dirección GVA de cuatro bits.
- Se utilizan dos niveles para las tablas de páginas y las EPT, en lugar de cuatro. Esto permite que sea más sencillo de explicar y entender.
- No se utiliza TLB, aunque se podría añadir como mejora. Sin embargo, es más fácil ver el problema sin usarlo.
- Las memorias físicas, tanto de la máquina real como virtual, direccionan hasta dieciseis direcciones, debido a la restricción de cuatro bits antes expuesta.
- Tanto la memoria física del host como la virtual se van ocupando de abajo a arriba, para simplificar la solución.

2.2. Sistemas semejantes

Las herramientas *online* para el aprendizaje suponen para los estudiantes uno de los principales medios en su formación al ser accesibles para todo el mundo. En este contexto, los simuladores son un mecanismo muy útil que está presente en muchos campos de estudio. En este caso, nos centramos en un simulador aplicado al estudio de los sistemas operativos, en concreto, la técnica de virtualización de memoria en entornos virtuales mediante la traducción de direcciones.

El concepto de simulador nace a principios de los años 60, definido como una herramienta basada en representaciones gráficas y cálculos numéricos que permite reproducir y manipular virtualmente situaciones reales o basadas en la realidad. Con el uso de un simulador, el usuario puede adquirir habilidades que mediante la teoría le costaría más conseguir.

En la educación, los estudiantes pueden hacer uso de los simuladores para recrear escenarios donde aplicar los conocimientos teóricos y entender los conceptos.

Como puede verse, presentan ventajas e inconvenientes para los usuarios que comparten con la gamificación, ya que podemos considerar un simulador como una herramienta gamificada. Entre los beneficios que aportan se encuentran algunos de los siguientes:

- Apoyan al aprendizaje convencional como otra forma de estudio.
- Proporcionan un alto nivel de interactividad para el usuario.
- Su objetivo es enseñar un escenario y contenido concreto.
- El usuario se convierte en un ser activo al interactuar con él.

En el campo de las ciencias, hay gran cantidad de materias que utilizan simuladores al permitir realizar experimentos a un bajo coste. En el ámbito de la informática también se encuentran simuladores. Como el caso que se plantea es el de traducción de direcciones virtuales en un sistema invitado a físicas en un sistema real, se han buscado soluciones similares y las diferencias que se aprecian con respecto a la solución propuesta en este trabajo. No obstante, no se han encontrado resoluciones que apliquen el problema que se ha expuesto, por lo que se realiza una comparación con respecto a aspectos generales.

Las características que presentan los simuladores actuales de forma genérica son las siguientes:

- Tienen una clara dependencia con el sistema operativo. La mayoría de las soluciones encontradas están preparadas para ser instaladas en un sistema operativo concreto, ya sea Linux o Windows.
- Cuentan con varias ventanas para configurar la simulación y para mostrar mensajes al usuario.
- El tema principal es la paginación en memoria, centrándose en algunos casos en técnicas o sistemas concretos sin entrar en entornos de máquinas virtuales.

Contexto del proyecto

- No hay diferenciación clara de los pasos que se llevan a cabo, mostrando demasiada información a la vez, de forma que se puede llegar a confundir al usuario.
- Las soluciones en algunos casos parecen más asociadas a resolver un problema de programación que a aportar un apoyo para el aprendizaje.
- Los diseños son muy básicos y, en algunos casos, los colores poco cuidados.

Tras analizar estos aspectos, se presentan las características que describen a la solución propuesta en el proyecto, a pesar de no ser el mismo problema el que se resuelve. Las ventajas que aporta son las siguientes:

- No depende de la plataforma al estar integrado en una página web accesible desde cualquier navegador.
- Solo hay una ventana con la información presentada lo más clara posible.
- Los pasos se describen a medida que el usuario va avanzando, permitiéndole volver hacia atrás si fuera necesario.
- La solución está planteada para complementar el aprendizaje, por lo que se ha desarrollado de la forma más sencilla posible.
- Se utilizan flechas para indicar los pasos que se van realizando y para mostrar la obtención de tablas y direcciones.
- El uso de etiquetas permite saber de dónde se obtienen los valores.
- Se utilizan colores neutros para no distraer al usuario.

A modo comparativo con respecto a los aspectos citados anteriormente, se presentan dos ejemplos de simuladores.

En primer lugar, uno creado con el lenguaje Java que simula cómo se administra la memoria virtual en un sistema operativo concreto. Se trata de un programa con varias variables configurables, lo que le añade complejidad, y una ventana en la que va apareciendo texto. Se trata de un simulador poco intuitivo para el usuario y, aunque parece muy completo en cuanto al proceso y variables que se pueden cambiar, no parece la mejor forma de aprender el proceso que describe.

En la figura 2.4 se puede ver una imagen previa del programa.

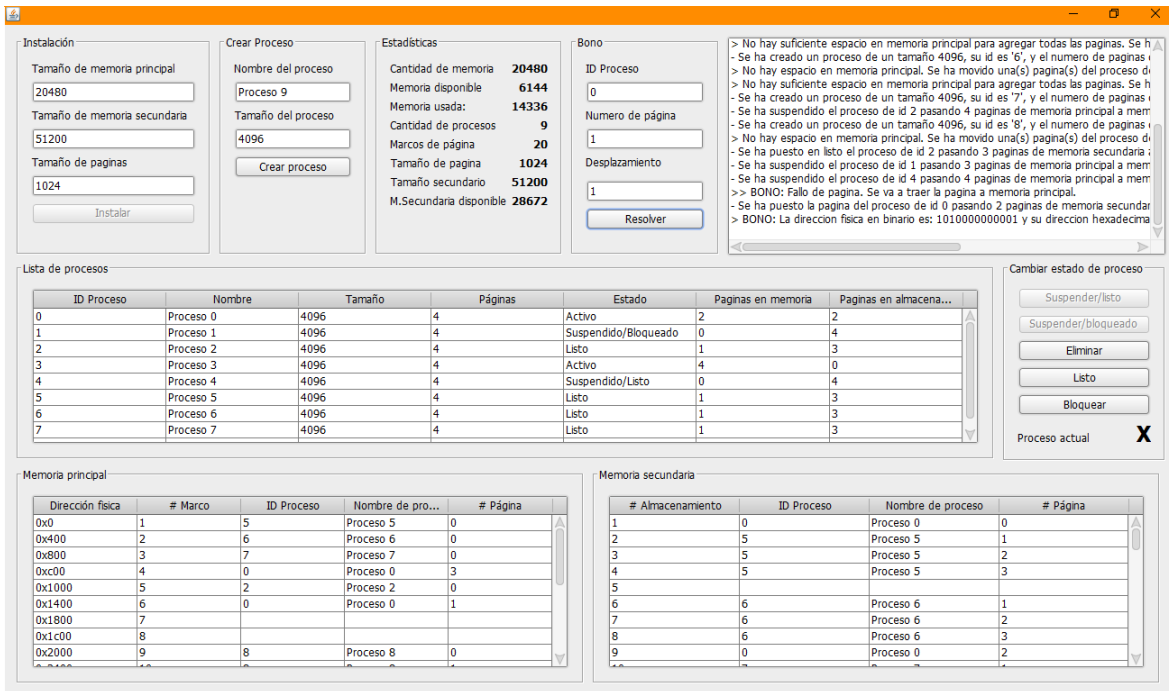


Figura 2.4: Simulador de memoria virtual con Paginación.

En segundo lugar, se trata de un simulador sobre jerarquías de memoria que comienza por la configuración de este, añadiendo las estructuras que se quieran usar y configurar una caché. Tras estas configuraciones, aparecen las ventanas de la simulación mostrando valores y estructuras, pero no textos con explicaciones. Sin embargo, los resultados los muestra en otra ventana a parte.

La figura 2.5. es una imagen de la ventana principal de este simulador.



Figura 2.5: Simulador de jerarquías de memoria.

Contexto del proyecto

Por último, se muestra el simulador que incluye este proyecto (figura 2.6) cuya solución es más sencilla con el fin de que sea lo más clara posible para los estudiantes incluyendo explicaciones en cada paso y las estructuras lo más simple posibles:

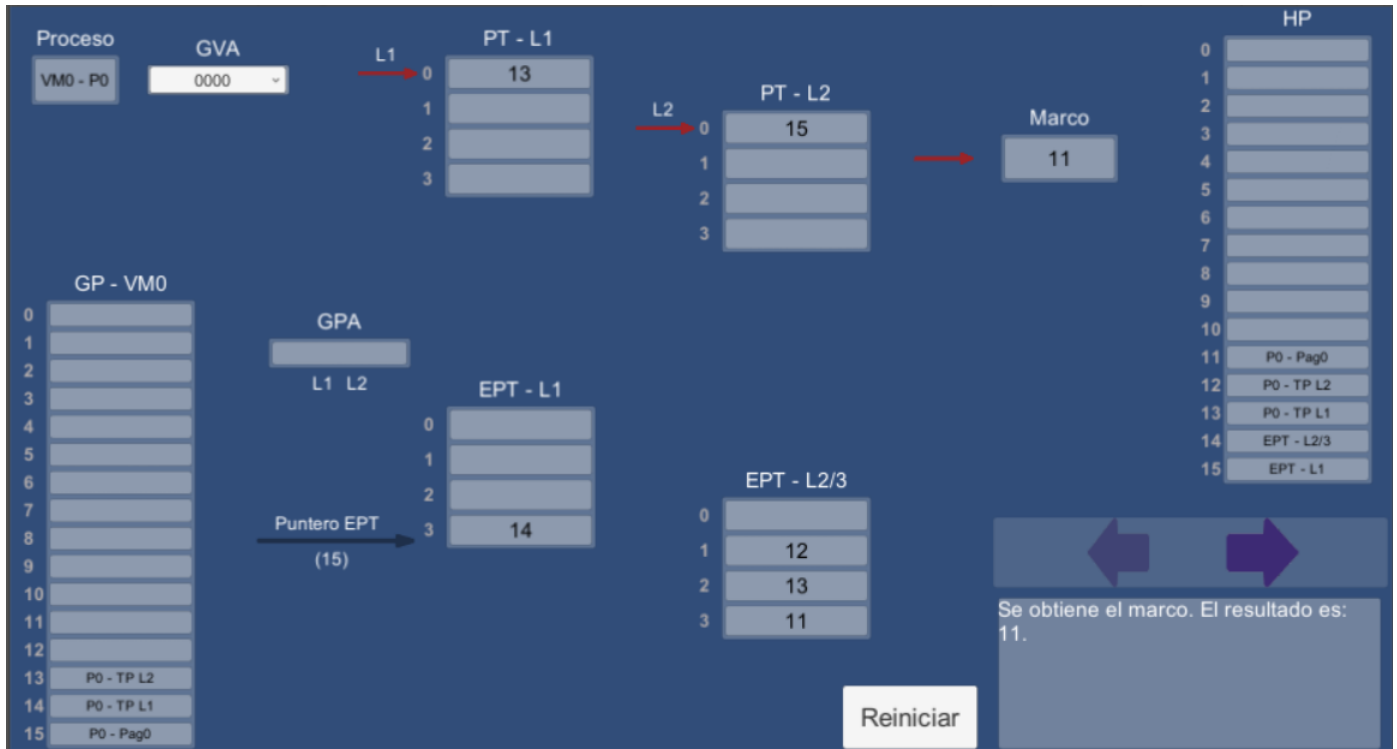


Figura 2.6: Simulador de traducción.

2.3. Gamificación

En el presente, el término gamificación ha ganado una gran importancia sobre todo en el ámbito de la educación. La gamificación representa el uso de técnicas, elementos y dinámicas propias de los juegos en actividades no lúdicas con el fin de conseguir distintos objetivos como el de adquirir conocimiento.

En este área, siempre ha habido elementos relacionados con los juegos, por ejemplo, la obtención de puntos cuando se realiza un trabajo de forma correcta, lo que se podría ver como una recompensa. Desde esta perspectiva clásica, los estudiantes obtienen buenas notas si lo hacen bien y malas si lo hacen mal. Si a final de curso obtienen buenos resultados, "suben de nivel", es decir, de curso. Sin embargo, estas técnicas convencionales no llaman la atención de los alumnos. Es por ello que se buscan nuevas formas de motivación a través de la gamificación.

No obstante, la gamificación cuenta con una gran cantidad de ventajas y desventajas que no tienen por qué darse en todas las actividades gamificadas.

2.3.1. Juego serio

Un juego serio hace referencia a aquellos que son desarrollados con un propósito distinto al de la diversión, como puede ser el de aprender. Este tipo de juegos se pueden encontrar en muchos ámbitos: publicidad, educación, salud, política, negocios...

En este caso, los juegos relacionados con la educación son en los que nos vamos a centrar. Por ejemplo, juegos educativos divididos por edades o niveles y juegos de entrenamiento que ayudan a aprender más rápido y desarrollar competencias.

Las ventajas y desventajas de los juegos serios son similares a los de la gamificación que se explican brevemente en los siguientes apartados, como el *feedback*, aumento de la motivación o el trabajo colaborativo.

2.3.2. Elementos de la gamificación

La gamificación cuenta con una serie de elementos que se pueden integrar en un juego y que se dividen en tres categorías:

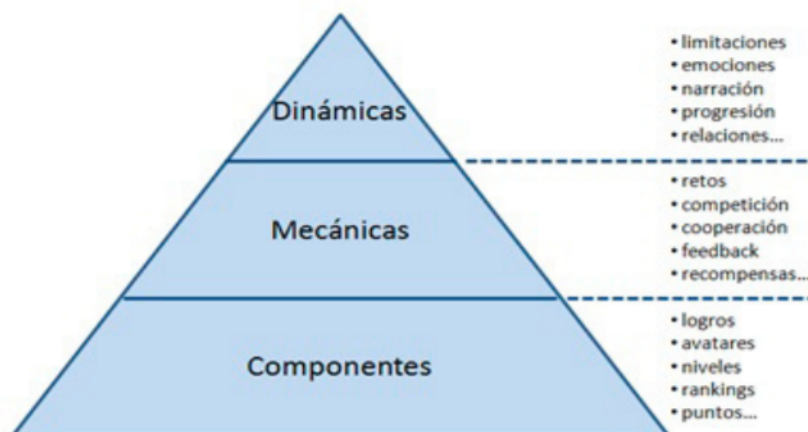


Figura 2.7: Elementos de la gamificación.

En primer lugar, los componentes son los recursos de los que se dispone y las herramientas que se van a utilizar a la hora de diseñar la actividad. En segundo lugar, las mecánicas del juego son los componentes básicos del juego, las reglas y su funcionamiento. Por último, las dinámicas son la forma en que se aplican las mecánicas y determinan el comportamiento de los estudiantes.

En el siguiente listado se describe brevemente los componentes de cada uno de estos elementos.

1. Dinámicas

- **Emociones:** se puede sentir curiosidad, competitividad, felicidad, frustración...

Contexto del proyecto

- **Narración:** hilo conductor como base para el proceso de aprendizaje.
- **Progresión:** evolución y desarrollo del jugador.
- **Relaciones:** interacción social, compañerismo...
- **Restricciones:** limitaciones que se pueden dar.

2. Mecánicas

- **Colaboración:** trabajo cooperativo para conseguir un objetivo.
- **Competición:** con los demás o con uno mismo. Hay ganadores y perdedores.
- **Desafíos:** tareas que supongan un reto.
- **Recompensas:** por cada logro el alumno recibe un premio.
- **Retroalimentación:** cómo lo está haciendo el alumno en cada momento.
- **Suerte:** influencia del azar.
- **Transacciones:** tratos entre los jugadores.
- **Turnos:** participación secuencial o alternativa.

3. Componentes

- **Avatar:** cada jugador tiene una imagen para representarle visualmente.
- **Colecciones:** elementos que se pueden acumular.
- **Combate:** batalla definida entre los participantes.
- **Desbloqueo de contenidos:** nuevos elementos que se van desbloqueando según se consi-
guen objetivos.
- **Equipos:** trabajo en equipo con un objetivo común.
- **Gráficos sociales:** los participantes saben cómo va el resto.
- **Huevos de pascua:** elementos escondidos que hay que buscar.
- **Insignias:** representación visual de los logros que se obtienen.
- **Límites de tiempo:** competición con el tiempo y con uno mismo.
- **Misiones:** desafíos concretos con recompensas.
- **Niveles:** de dificultad y progresión.
- **Puntos:** recompensas para representar la progresión.
- **Clasificación y barras de progreso:** representación gráfica de la progresión.
- **Regalos:** compartición de recursos con otros.
- **Tutoriales:** familiarizarse con el juego aprendiendo las normas.

2.4. Aplicación en este trabajo

En el desarrollo de este proyecto se han aplicado algunas de las técnicas explicadas anteriormente.

Mediante este trabajo se busca ofrecer al estudiante una herramienta de educación *online* que permita a cada usuario seguir su propio ritmo de aprendizaje además de contar con las ventajas que tiene la enseñanza general. A su vez, permite que el alumno pueda aprender o estudiar el tema sin tener que mantener el contacto directo con el profesor lo que le proporciona una mayor libertad.

Por otra parte, cuenta con tendencias de gamificación, en particular, los juegos serios. Aunque la implementación de uno se sale del ámbito de este proyecto, es modificable para adaptarlo. El paso más intuitivo para ello sería el de convertir los tests en un juego serio añadiendo un componente gráfico.

	Descripción	Cómo
Mecánicas		
Competición	El alumno compite consigo mismo y con los demás por obtener mejores puntuaciones para subir de nivel.	Perfil para ver su puntuación. Ranking de usuarios.
Recompensas	El alumno que mejor puntuación consiga recibe una recompensa.	El profesor decide cuál es la recompensa y las condiciones.
Retroalimentación o Feedback	El alumno conoce sus fallos y aciertos al momento.	Representación visual de los aciertos y fallos de las preguntas.
Componentes		
Combate	Los alumnos compiten entre ellos por alcanzar un puesto alto en la clasificación.	Ranking de usuarios.
Desbloqueo de contenidos	Los alumnos pueden subir de nivel lo que hace que se desbloqueen nuevas preguntas.	Nuevas preguntas en el test acorde al nivel.
Gráfico social	Se puede consultar cómo van los mejores alumnos.	Ranking de los usuarios.
Niveles	Los tests van por niveles progresivos.	Cada nivel tiene unos test, el usuario va subiendo cuando alcanza un número de preguntas acertadas. Hasta un máximo de tres niveles.
Clasificación	Los usuarios pueden ver la clasificación final de los demás.	Clasificación de usuarios.

Tabla 2.1: Elementos gamificados del proyecto.

Contexto del proyecto

A pesar de no tratarse de un juego, se pueden apreciar varios elementos que se han aplicado para considerar el factor gamificado.

Mediante la tabla 2.1 se explica qué se ha utilizado, cómo y dónde.

Entre las ventajas que además se aprecian también en este trabajo, se pueden encontrar las siguientes:

- **Aumentan la motivación de los alumnos:** mostrarles el problema en cuestión como algo útil y divertido de forma que el aprenderlo les motive a hacerlo.
- **Ritmo de aprendizaje:** está sujeto a las características personales del alumno. Es el que decide en todo momento a qué ritmo realiza la actividad.
- **Aprender de los fallos:** el alumno puede tener fallos lo que hará que pueda aprender de ellos revisándolos.
- **Feedback:** el alumno recibe los resultados de forma inmediata. También el docente los recibe de forma que puede seguir la trayectoria de sus estudiantes.
- **Aumenta la dificultad de manera progresiva:** los niveles de dificultad de un juego hacen que los estudiantes desarrollen habilidades para progresar gradualmente.
- **Incrementa la adquisición de conocimiento:** los alumnos desarrollan y practican la resolución de problemas a través de distintos niveles de dificultad.
- **Práctica sobre un contexto real:** se aplican prácticas relacionadas con el tema real y se estudia sobre ellas.
- **Promueve el trabajo en equipo y la cooperación:** los estudiantes pueden participar por equipos lo que hace que aprendan a trabajar como un equipo y cooperar entre ellos.

Por otro lado, los juegos también tienen inconvenientes. No se dan en el caso del simulador al no ser un juego serio.

- **Tiende a cuantificar, reducir y clasificar:** si se utilizan los juegos como herramienta única de evaluación los resultados pueden estar sesgados.
- **El alumno se focaliza en ganar:** se le olvida que el objetivo es aprender.
- **Hacer trampas:** el alumno se puede ver tentado a hacer trampas para ganar.
- **Intensificar la desmotivación:** los alumnos que no resulten ganadores pueden desmotivarse.
- **Inconvenientes específicos del juego.**
- **Puede requerir tener acceso a Internet o a un ordenador:** esto puede limitar la actividad al ser posible que alguien no tenga acceso a ello.

- **Distrae a los estudiantes del objetivo de aprender:** si el juego está mal diseñado puede llevar a confusión.
- **No tiene por qué cumplir con las necesidades de aprendizaje de todos los estudiantes.**

Capítulo 3

Planificación

3.1. Planificación del trabajo

En la figura 3.1, se muestra el diagrama de Gantt del trabajo, que comienza con el aprendizaje de las herramientas a utilizar y de los fundamentos teóricos necesarios para poder llevar a cabo el proyecto. Esta fase comprende desde febrero hasta abril. El desarrollo del proyecto se realiza desde abril hasta junio. Por último, hasta julio se revisa el trabajo y se realizan los cambios oportunos.

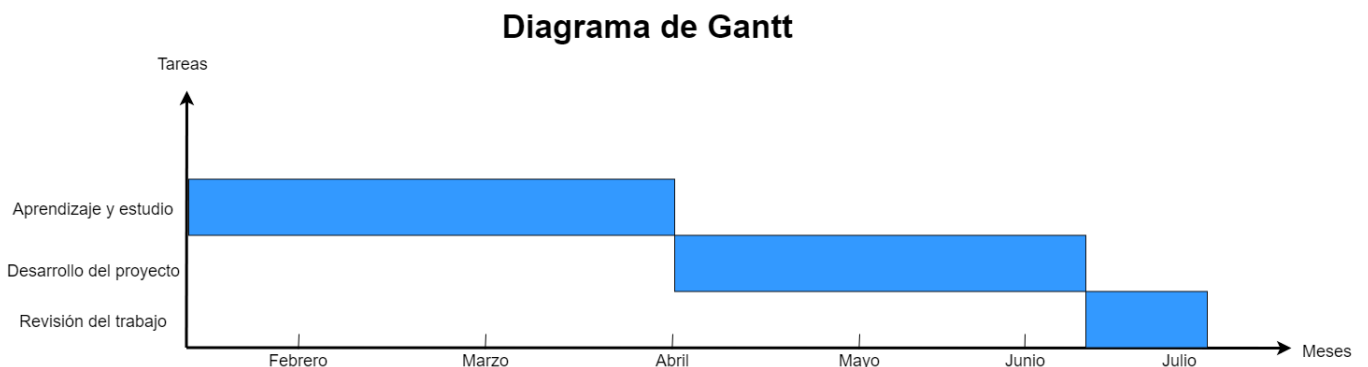


Figura 3.1: Diagrama de Gantt del trabajo.

3.2. Planificación inicial

Para la planificación de este proyecto se utiliza una metodología ágil basada en algunos aspectos de Scrum. No se puede utilizar esta última debido a que no se cumplen algunos aspectos, por ejemplo, el equipo de desarrollo no tiene entre dos y siete miembros o no existe el rol de *Scrum Manager*.

Partiendo de esto, se definen los roles que se dan en el desarrollo de este proyecto:

- **Equipo de desarrollo:** en este caso, está formado por una sola persona, Silvia Arias. Es la

encargada de crear el producto.

- **Product owner:** encargado de comunicarse entre el equipo de desarrollo y el usuario final. En este proyecto serían Benjamín Sahelices y Alma Pisabarro.
- **Cliente:** es el que recibe el producto finalizado. A partir de sus necesidades se obtienen las historias de usuario. Serían Benjamín Sahelices y Alma Pisabarro.

Una vez definidos los roles, se definen los artefactos que se utilizan para medir el progreso del proyecto:

- **Product backlog:** lista total de las historias de usuario ordenadas por prioridad que están en el alcance del proyecto.
- **Sprint backlog:** identifica los objetivos, historias de usuario y tareas asociadas al sprint actual.

3.2.1. Distribución temporal

En la distribución temporal del proyecto se tiene en cuenta el tiempo disponible por parte de la desarrolladora para la realización de éste, así como, la normativa indicada para los Trabajos de Fin de Grado. Para ello se establecen cinco *sprints* de tres semanas cada uno, divididos de la siguiente forma:

- **Sprint 1:** Se establecen las funcionalidades que tendrá la aplicación y se crea un prototipo de diseño para el simulador. Se comienzan a realizar algunas historias de usuario del simulador.
- **Sprint 2:** Se continúa con la implementación del simulador, realizando más historias de usuario. A su vez, se va profundizando en el aprendizaje de Unity.
- **Sprint 3:** Se continúa con la implementación del simulador añadiendo otras funcionalidades que incluyen la lógica de escritura de tablas y dibujado de flechas.
- **Sprint 4:** Se realizan las plantillas para la aplicación web y se implementan otras funcionalidades del simulador.
- **Sprint 5:** Implementación de la lógica de la parte web y finalización del desarrollo del simulador. También se integran ambas partes.

Los cinco sprints hacen un total de quince semanas, repartidas entre abril y junio. Se ha decidido que fueran de esta longitud de tiempo debido a la falta de experiencia en la tecnología utilizada, el tiempo disponible y por causas externas al proyecto que se explicarán en la descripción de cada sprint.

3.2.2. Análisis de riesgos

En la tabla 3.1, se describen los riesgos de forma descendente, es decir, de más crítico a menos crítico.

Planificación

Se realiza un análisis para detectar los posibles riesgos que podrían darse durante el desarrollo del proyecto. Una vez identificados estos, también se exponen las posibles respuestas que se puedan tener, así como, un plan de contingencias en caso de que finalmente sucedieran estos.

Riesgo	Prob.	Descripción	Impacto	Exposición	Protección
1.Pérdida de datos.	Baja.	Se pierden los datos del código fuente.	Catastrófico.	Moderado.	Utilizar un sistema de control de versiones para almacenar el código y actualizarlo asiduamente.
2.Falta de experiencia en Unity.	Alta	Se desconoce cómo implementar cierta funcionalidad con el motor.	Crítico.	Alta.	Estudio previo de la herramienta.
3.Problemas al integrar Unity en la aplicación web.	Alta	A la hora de integrar el simulador con la aplicación web se dan problemas de compatibilidad o de visualización gráfica.	Crítico.	Alta.	Realizar compilaciones del proyecto según se va realizando para ver cómo queda.
4.Planificación muy optimista.	Alta	Los tiempos son muy cortos o muy largos.	Crítico.	Alta.	Hacer una planificación realista según el tiempo disponible del desarrollador.
5.Suspense en la asignatura "DIAS".	Media	Suspense en la asignatura Diseño, Integración y Adaptación del software que requeriría tener que prepararla para la convocatoria extraordinaria.	Crítico.	Moderado.	Estudiar la asignatura día a día.
6.Enfermedad.	Baja	El desarrollador cae enfermo de forma imprevista.	Crítico.	Baja.	Ninguno.

Tabla 3.1: Riesgos del proyecto.

En la tabla 3.2, se describe el plan de contingencia para cada riesgo.

Riesgo	Plan de Contingencia
1.Pérdida de datos.	Descargar el código fuente del repositorio.
2.Falta de experiencia en Unity.	Dedicar más horas para resolver el problema o consultar la documentación.
3.Problemas al integrar Unity en la aplicación web.	Revisar la documentación para ver cómo integrarlo.
4.Planificación muy optimista.	Modificar el plan de trabajo.
5.Suspenseo en la asignatura "DIAS".	Estudiar para el examen y dedicarle más horas al proyecto.
6.Enfermedad.	Modificar las horas y el plan de trabajo.

Tabla 3.2: Plan de contingencia para cada riesgo.

3.3. Estimación de costes

La estimación de costes se realiza a partir de los recursos utilizados en el proyecto que se describen en los apartados siguientes.

3.3.1. Recursos

Se ha decidido utilizar una agrupación para los recursos en tres categorías: personal, software y entorno. Para cada uno de ellos se indica una descripción, la disponibilidad y la fecha de necesidad.

Personal

Se enmarcan en este grupo los miembros del equipo de desarrollo, en este caso, solo hay una. Estará encargado del desarrollo completo del proyecto por lo que será necesario durante todo el proceso desde el momento del inicio hasta el final.

Para establecer el coste de este recurso, es necesario saber cuál es el salario medio de un desarrollador en España. En la actualidad, un programador *junior*, es decir, con menos de cinco años de experiencia, tiene un salario en torno a los 18.000€/año, lo que hacen 9.375€/hora.

Otro recurso en esta categoría es el de la ubicación, es decir, el lugar donde se encuentra el personal trabajando. El proyecto se ha desarrollado en dos lugares: la Escuela de Ingeniería Informática y espacio de trabajo personal. El coste que supone el uso de estos espacios no es posible calcularlo.

Software

En cuanto a los recursos software habría que tener en cuenta componentes de experiencia completa o parcial, es decir, diseños, datos o códigos de pruebas diseñados para otros proyectos similares al software que se va a construir y que se pueden aplicar al proyecto. También se incluye el software de terceros que pudiese incorporarse en el proyecto de manera similar a estos componentes citados. No es este el caso, ya que se ha creado el software desde cero, por lo que se trata de componentes nuevos.

Estos componentes se utilizan durante todo el desarrollo del proyecto.

Entorno

En este marco se tienen en cuenta los componentes hardware y software utilizados para el desarrollo del proyecto. Los componentes hardware son necesarios para poder utilizar los de tipo software. Estos son necesarios durante todo el transcurso del trabajo.

En cuanto a elementos software se listan los siguientes:

- **Unity:** para el desarrollo del simulador.
- **Visual Studio 2017:** para programar la parte del simulador en C#.
- **Netbeans IDE:** para el desarrollo de la web.
- **Bootstrap:** permite añadir estilos a la parte web.
- **Apache Tomcat:** como servidor web para la aplicación.
- **Java 8:** como lenguaje de desarrollo en la parte web.
- **C#:** como lenguaje de desarrollo para Unity.
- **Bitbucket:** como repositorio de control de versiones.
- **Git Bash 2.16.2:** para obtener y subir los cambios al repositorio desde local.
- **Astah Professional:** para el diseño del software.
- **Latex:** para la memoria del proyecto.
- **Draw.io:** para la realización de esquemas dibujados.
- **Correo UVa:** comunicación con los tutores.
- **WhatsApp:** comunicación con los tutores.
- **Aplicación Recordatorios:** para realizar una lista de tareas.

De este listado, el único recurso que no está disponible sin coste es el programa Astah Professional aunque se cuenta con licencia proporcionada por la Escuela para su uso. Por otra parte, en el caso de Visual Studio se utiliza la licencia para estudiantes que permite utilizarlo de forma gratuita.

El precio para la licencia de Astah es de 22.50€ cada tres meses.

La aplicación "Recordatorios", solo está disponible para sistemas iOS.

Los recursos hardware para el proyecto comprenderían:

- **Ordenador portátil:** para el desarrollo de la aplicación.
- **Monitor:** como herramienta de trabajo complementaria al ordenador.

En las tablas 3.3 y 3.4, se especifican las características de estos recursos.

1- Ordenador portátil		
Descripción:	Ordenador portátil HP del año 2018.	
Componentes	Procesador:	Intel Core i7 8550U CPU @ 1.80GHz 1.99GHz.
	Memoria RAM:	16 GB DDR4.
	Tarjeta gráfica:	NVIDIA GeForce 940MX.
	Disco SSD:	PCIe NVMe 256 GB.
	Disco Duro:	SATA 1 TB.
	Resolución de la pantalla:	1920x1080 ppp.
Precio:	999.90€	

Tabla 3.3: Ordenador portátil: características.

2-Monitor		
Descripción:	Monitor HP LP2065	
Componentes	Pulgadas:	20"
	Resolución de la pantalla:	1600x1200 ppp.
Precio:	140€	

Tabla 3.4: Monitor: características.

3.3.2. Presupuesto

Tras especificar los recursos y el coste estimado que estos podrían tener, se establece el presupuesto mostrado en la tabla 3.5.

Recurso	Precio por unidad	Cantidad	Subtotal
1. Ordenador portátil	999.9€	1	999.9€
2. Monitor	140€	1	140€
3. Licencia Astah	22.50€	1	22.50€
4. Coste trabajador	3000€/mes	2 meses	6000€
Total:			7162.4€

Tabla 3.5: Presupuesto

3.4. Historias de usuario

Mediante las historias de usuario se describe la funcionalidad que será creada y que será útil para el usuario. Estas historias se incluyen dentro del *product backlog* y en cada sprint se decide cuáles se van a implementar.

Entre las características que tienen se encuentran las siguientes:

- **Independencia entre ellas:** cada historia representa un requisito y no debe depender de otra. Si se diera el caso, habría que combinar las historias o escribirlas de forma que no suceda.
- **Negociables:** se deben de poder clarificar las historias.
- **Valoradas por los clientes o los usuarios:** deben de tener un valor para ellos.
- **Estimables:** tiempo que lleva realizarla.
- **Pequeñas:** son más fáciles de estimar.
- **Verificables:** cubren requisitos funcionales y se pueden verificar.

En las siguientes tablas se detallan todas las historias de usuario ordenadas por prioridad. Primero se muestran todas las del simulador y después las de la web.

HU01 - Crear tablas de las estructuras.	
Número: 1	
Prioridad: Alta	
Estimación: 3	
Descripción:	Como usuario quiero que se muestren las tablas de las estructuras cuando se necesiten.
Condición de satisfacción:	Mostrar las tablas colocadas en el lugar correspondiente en el paso que corresponda.

Tabla 3.6: HU01 - Crear tablas de las estructuras.

HU02 - Elegir dirección a traducir.	
Número: 2	
Prioridad: Alta	
Estimación: 2	
Descripción:	Como usuario quiero elegir la dirección GVA que se va a traducir.
Condición de satisfacción:	Añadir un dropdown con las opciones de direcciones.

Tabla 3.7: HU02 - Elegir dirección a traducir.

HU03 - Flechas para ir hacia delante y retroceder.	
Número: 3	
Prioridad: Alta	
Estimación: 2	
Descripción:	Como usuario quiero poder avanzar y retroceder en la simulación.
Condición de satisfacción:	Mostrar dos botones de flechas para avanzar y retroceder. Volver al paso anterior.

Tabla 3.8: HU03 - Flechas para ir hacia delante y retroceder.

HU04 - Pasos.	
Número: 4	
Prioridad: Alta	
Estimación: 15	
Descripción:	Como usuario quiero que la simulación cuente con pasos para ir avanzando o retrocediendo a ellos.
Condición de satisfacción:	Añadir opción flechas para que pueda cambiar de paso.

Tabla 3.9: HU04 - Pasos.

HU05 - Texto con explicación	
Número: 5	
Prioridad: Alta	
Estimación: 2	
Descripción:	Como usuario quiero ver una explicación de lo que sucede en cada paso.
Condición de satisfacción:	Mostrar un texto explicatorio en cada paso.

Tabla 3.10: HU05 - Texto con explicación.

HU06 - Mostrar flechas en el proceso.	
Número: 6	
Prioridad: Alta	
Estimación: 20	
Descripción:	Como usuario quiero que los pasos de obtención de direcciones o tablas se muestren con flechas.
Condición de satisfacción:	Mostrar flechas de color rojo en casa paso.

Tabla 3.11: HU06 - Mostrar flechas en el proceso.

HU07 - Mostrar etiquetas en las tablas.	
Número: 7	
Prioridad: Alta	
Estimación: 1	
Descripción:	Como usuario quiero que las tablas se distingan con etiquetas.
Condición de satisfacción:	Añadir etiquetas encima de cada tabla.

Tabla 3.12: HU07 - Mostrar etiquetas en las tablas.

HU08 - Numerar celdas.	
Número: 8	
Prioridad: Alta	
Estimación: 1	
Descripción:	Como usuario quiero que las celdas de cada tabla estén numeradas para saber qué posición representan.
Condición de satisfacción:	Se añaden etiquetas a la izquierda de cada celda en cada tabla.

Tabla 3.13: HU08 - Numerar celdas.

HU09 - Mostrar puntero a CR3.	
Número: 9	
Prioridad: Alta	
Estimación: 1	
Descripción:	Como usuario quiero que aparezca un puntero de CR3 para saber dónde apunta.
Condición de satisfacción:	Se añade una flecha en CR3 que apunta a la tabla de páginas.

Tabla 3.14: HU09 - Mostrar puntero a CR3.

HU10 - Mostrar puntero a EPT.	
Número: 10	
Prioridad: Alta	
Estimación: 1	
Descripción:	Como usuario quiero que aparezca un puntero a EPT para saber cuándo se obtiene la tabla.
Condición de satisfacción:	Se añade una flecha que apunta a la EPT de primer nivel.

Tabla 3.15: HU10 - Mostrar puntero a EPT.

HU11 - Mostrar animación dirección elegida.	
Número: 11	
Prioridad: Alta	
Estimación: 2	
Descripción:	Como usuario quiero que cuando elija la dirección se muestre en la memoria física de la máquina virtual en amarillo para saber dónde se localiza.
Condición de satisfacción:	Mostrar una animación que resalte el fondo en amarillo durante unos segundos.

Tabla 3.16: HU11 - Mostrar animación dirección elegida.

HU12 - Borrado.	
Número: 12	
Prioridad: Alta	
Estimación: 20	
Descripción:	Como usuario quiero que al volver al paso anterior se borren los datos del paso siguiente para poder repetirlo.
Condición de satisfacción:	Se borran los datos necesarios para volver al estado anterior.

Tabla 3.17: HU12 - Borrado.

HU13 - GPA vacío.	
Número: 13	
Prioridad: Alta	
Estimación: 1	
Descripción:	Como usuario quiero que cuando no se traduzca una dirección GPA aparezca vacía la tabla.
Condición de satisfacción:	Se escribe un espacio en blanco.

Tabla 3.18: HU13 - GPA vacío.

HU14 - Reiniciar simulación.	
Número: 14	
Prioridad: Media	
Estimación: 4	
Descripción:	Como usuario quiero poder reiniciar la simulación para empezar de cero.
Condición de satisfacción:	Añadir un botón para restablecer todos los valores.

Tabla 3.19: HU14 - Reiniciar simulación.

HU15 - Mostrar animación dirección GPA a traducir.	
Número: 15	
Prioridad: Media	
Estimación: 2	
Descripción:	Como usuario quiero que cuando se traduzca una dirección GPA se muestre esta en rojo para saber qué dirección traduce.
Condición de satisfacción:	Mostrar una animación que resalte el fondo en rojo durante unos segundos.

Tabla 3.20: HU15 - Mostrar animación dirección GPA a traducir.

HU16 - Mostrar etiquetas en las flechas.	
Número: 16	
Prioridad: Media	
Estimación: 2	
Descripción:	Como usuario quiero que las flechas tengan etiquetas para saber de dónde vienen.
Condición de satisfacción:	Añadir etiquetas encima de cada flecha donde sea necesario.

Tabla 3.21: HU16 - Mostrar etiquetas en las flechas.

HU17 - Mostrar etiquetas en las direcciones.	
Número: 17	
Prioridad: Media	
Estimación: 1	
Descripción:	Como usuario quiero que las direcciones GVA y GPA tengan etiquetas que muestren L1 y L2 para poder diferenciarlo.
Condición de satisfacción:	Añadir etiquetas debajo del campo de la dirección con estas etiquetas.

Tabla 3.22: HU17 - Mostrar etiquetas en las direcciones.

HU18 - Número de proceso.	
Número: 18	
Prioridad: Media	
Estimación: 1	
Descripción:	Como usuario quiero saber a qué proceso pertenece la dirección virtual en cada traducción.
Condición de satisfacción:	Añadir un campo con el número del proceso.

Tabla 3.23: HU18 - Número de proceso.

HU19 - Terminar simulación al no haber espacio.	
Número: 19	
Prioridad: Media	
Estimación: 2	
Descripción:	Como usuario quiero que, al quedarse sin espacio la memoria física, acabe la simulación y muestre un mensaje.
Condición de satisfacción:	Detectar cuando se queda sin espacio. Avisar mediante un mensaje en rojo. Desactivar las flechas de la simulación.

Tabla 3.24: HU19 - Terminar simulación al no haber espacio.

HU20 - Login.	
Número: 20	
Prioridad: Alta	
Estimación: 2	
Descripción:	Como usuario quiero poder iniciar sesión en la aplicación web para acceder a la funcionalidad.
Condición de satisfacción:	Mostrar formulario de inicio de sesión con nombre de usuario y contraseña.

Tabla 3.25: HU20 - Login.

HU21 - Logout.	
Número: 21	
Prioridad: Alta	
Estimación: 1	
Descripción:	Como usuario quiero poder cerrar sesión en la aplicación web.
Condición de satisfacción:	Mostrar botón en el menú superior para cerrar sesión. Vuelve a la página de inicio de sesión.

Tabla 3.26: HU21 - Logout.

HU22 - Registro.	
Número: 22	
Prioridad: Alta	
Estimación: 6	
Descripción:	Como usuario quiero poder registrarme en la aplicación para acceder a la funcionalidad.
Condición de satisfacción:	Mostrar formulario de registro accesible desde la página de login.

Tabla 3.27: HU22 - Registro.

HU23 - Ver Ranking.	
Número: 23	
Prioridad: Alta	
Estimación: 2	
Descripción:	Como usuario quiero ver un ranking de usuarios por nivel para ver los mejores puestos.
Condición de satisfacción:	Mostrar tabla con lista ordenada por nivel de los diez mejores usuarios en la página principal.

Tabla 3.28: HU23 - Ver Ranking.

HU24 - Ver Teoría.	
Número: 24	
Prioridad: Alta	
Estimación: 2	
Descripción:	Como usuario quiero ver contenido teórico relacionado con el tema del simulador.
Condición de satisfacción:	Mostrar pestaña de teoría en el menú superior. Mostrar contenido teórico dentro de esta página. Dividirlo por bloques.

Tabla 3.29: HU24 - Ver Teoría.

HU25 - Tests.	
Número: 25	
Prioridad: Alta	
Estimación: 10	
Descripción:	Como usuario quiero poder hacer tests relacionados con la teoría para saber si entiendo los conceptos.
Condición de satisfacción:	Mostrar pestaña de tests en el menú superior. Mostrar cinco preguntas acordes al nivel.

Tabla 3.30: HU25 - Tests.

HU26 - Ver soluciones del test.	
Número: 26	
Prioridad: Alta	
Estimación: 10	
Descripción:	Como usuario quiero saber si mis respuestas a las preguntas son correctas para poder revisarlas.
Condición de satisfacción:	Mostrar visualmente las respuestas. Mostrar las respuestas incorrectas en rojo. Mostrar un mensaje en las no contestadas. Mostrar las respuestas correctas en verde

Tabla 3.31: HU26 - Ver soluciones del test.

HU27 - Ver simulador.	
Número: 27	
Prioridad: Alta	Valor:
Estimación: 2	
Descripción:	Como usuario quiero poder acceder al simulador para interactuar con él.
Condición de satisfacción:	Añadir opción simulador en el menú superior. Mostrar el simulador en esta página.

Tabla 3.32: HU27 - Ver simulador

HU28 - Ver perfil.	
Número: 28	
Prioridad: Alta	
Estimación: 2	
Descripción:	Como usuario quiero poder ver la información de mi perfil.
Condición de satisfacción:	Añadir opción de ver perfil en el menú superior. Mostrar datos del usuario en esta página.

Tabla 3.33: HU28 - Ver perfil.

HU29 - Número de preguntas acertadas.	
Número: 29	
Prioridad: Alta	
Estimación: 1	
Descripción:	Como usuario quiero poder ver el número de preguntas acertadas.
Condición de satisfacción:	Añadir opción de ver perfil en el menú superior. Mostrar este dato en esta página.

Tabla 3.34: HU29 - Número de preguntas acertadas.

HU30 - Nivel.	
Número: 30	
Prioridad: Alta	
Estimación: 1	
Descripción:	Como usuario quiero poder ver el nivel en el que estoy.
Condición de satisfacción:	Añadir opción de ver perfil en el menú superior. Mostrar este dato en esta página.

Tabla 3.35: HU30 - Nivel

HU31 - Subir nivel.	
Número: 31	
Prioridad: Alta	
Estimación: 1	
Descripción:	Como usuario quiero subir de nivel para mejorar mi progreso.
Condición de satisfacción:	Con 35 preguntas acertadas sube a nivel 2. Con 70 sube a nivel 3.

Tabla 3.36: HU31 - Subir nivel.

3.5. Seguimiento del proyecto

El trabajo pendiente del proyecto está repartido en cinco sprints en el que se desarrollan treinta y una historias de usuario. En el gráfico *burndown* global se puede ver el número de tareas pendientes, representadas por las historias de usuario, y los sprints del proyecto en los que se realizarán estas.

En los siguientes apartados se describe cómo ha sido el desarrollo de los *sprints*, qué tareas se han conseguido realizar y cómo ha sido el progreso en cada uno de ellos. Para verlo, se crean los *burndown charts* de cada uno de ellos, donde se ve el trabajo realizado y el pendiente. También se describe un pequeño desarrollo de cada uno de ellos.

En la figura 3.2, se puede ver este gráfico a nivel de todo el proyecto.

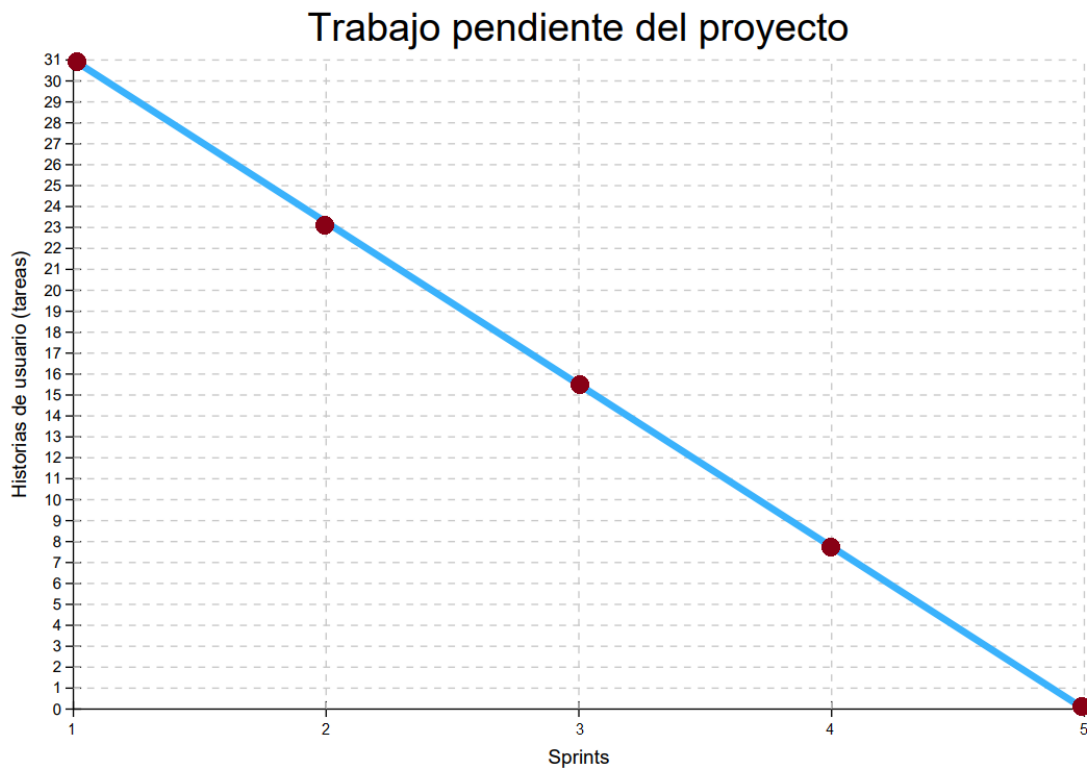


Figura 3.2: Burndown chart global.

3.5.1. Sprint 1

Spring backlog

En este sprint se realizarán las siguientes historias de usuario: HU01, HU02, HU03, HU05, HU07, HU08, HU11.

Descripción

Durante el tiempo de este sprint se comienza realizando un prototipo de diseño de cómo será el simulador para saber cómo colocar los elementos. Después se comienzan a realizar las historias de usuario elegidas en el *sprint backlog*.

En el transcurso de este *sprint* se consiguieron implementar las historias de usuario pactadas en el *sprint backlog*. Aunque los textos de explicación (HU-05) se revisarán más adelante, cuando esté acabado el simulador, por si fuera necesario algún cambio.

El prototipo creado se muestra en la figura 3.3.

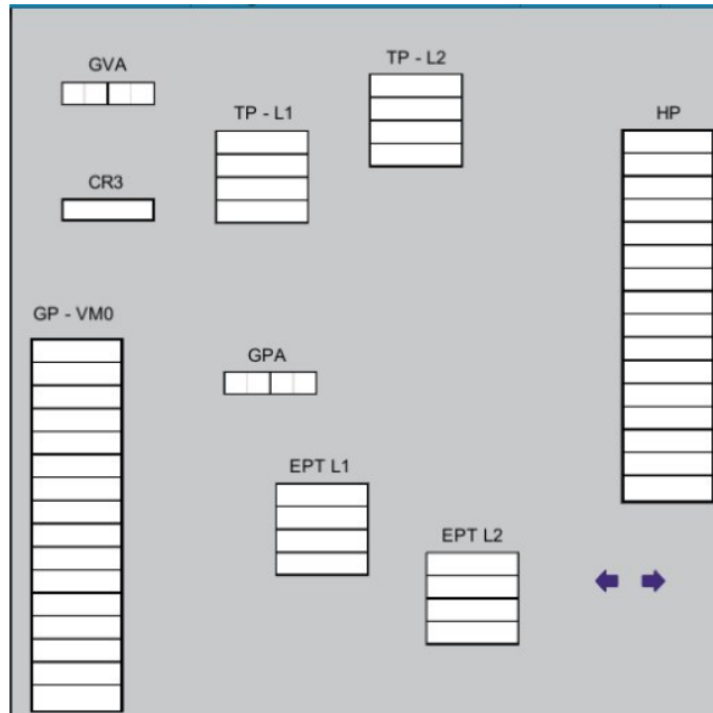


Figura 3.3: Prototipo inicial del simulador.

Tras la implementación de las historias marcadas, el estado del simulador es el de la figura 3.4.

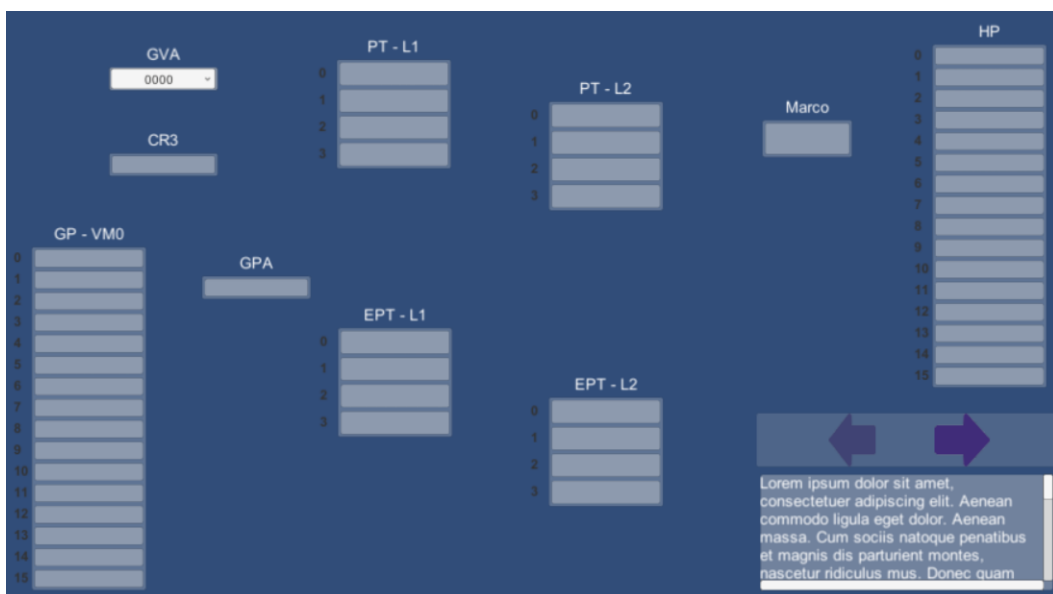


Figura 3.4: Simulador tras sprint 1.

Burndown chart del sprint

Como se puede ver en la figura 3.5, durante el sprint se desarrollaron las tareas propuestas acabando según la planificación, aunque se dieron casos en el que el trabajo estuvo atrasado y adelantado pero no de forma muy notable.

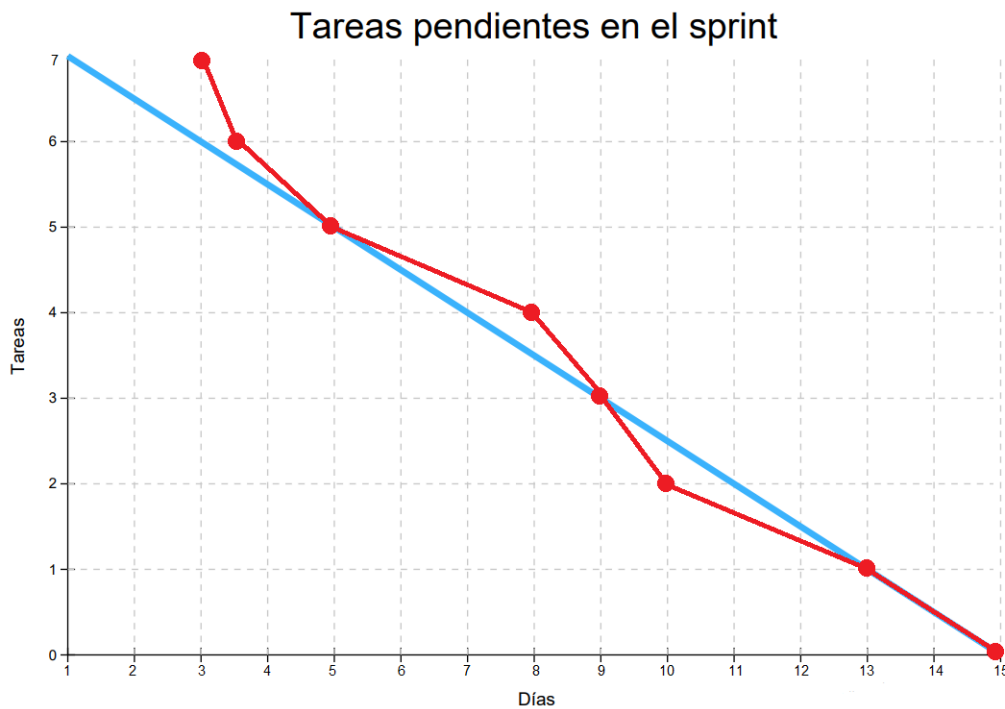


Figura 3.5: Burndown chart sprint 1.

3.5.2. Sprint 2

Spring backlog

En este *sprint* se realizarán las siguientes historias de usuario: HU04, HU06, HU09, HU10, HU16, HU17.

Descripción

En el intervalo de este *sprint* se comienzan a realizar historias de usuario de mayor complejidad, como es la implementación de pasos en la simulación. Sin embargo, no se consigue terminar, por lo que se continuará con ella en el siguiente *sprint*.

Burndown chart del sprint

Durante el desarrollo de este *sprint*, como se muestra en el gráfico, el trabajo se adelantó en dos tareas a la planificación, pero debido a que no se pudo conseguir la implementación de una de las historias de usuario, se atrasa la ejecución de tareas.

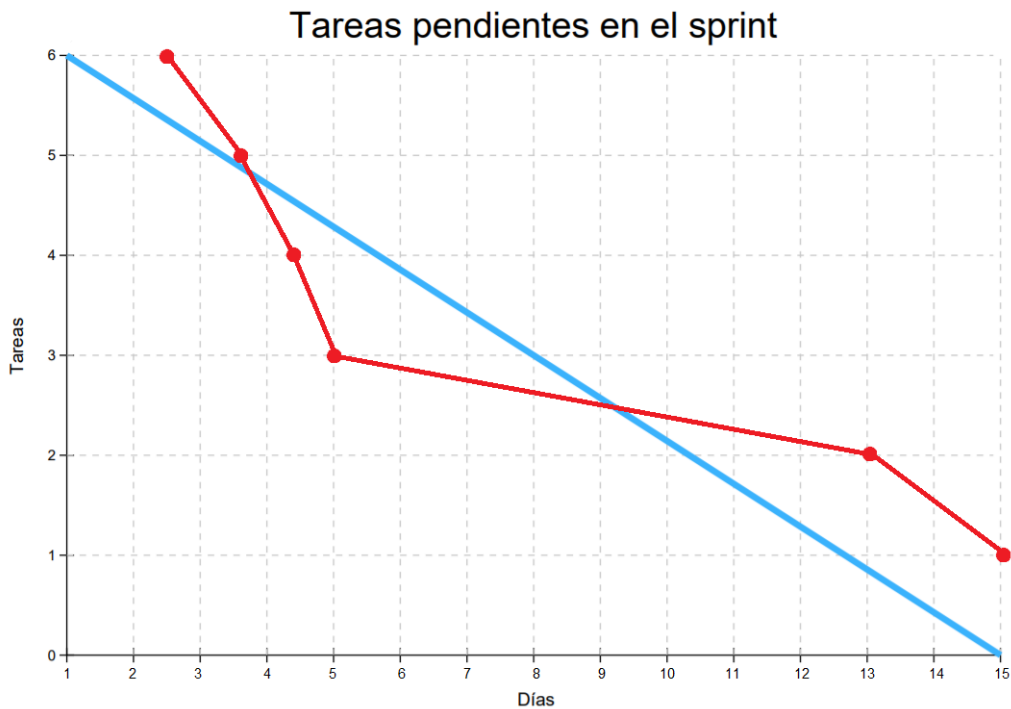


Figura 3.6: Burndown chart sprint 2.

3.5.3. Sprint 3

Sprint backlog

En este *sprint* se realizarán las siguientes historias de usuario: HU06 , HU04, HU13, HU15.

Descripción

Para este periodo, se establecen menos historias de usuario a implementar debido a que el tiempo disponible es menor por la disponibilidad de la desarrolladora. Aunque se continúa con la historia que no se pudo acabar en el *sprint* anterior y se comienza con el dibujado de flechas, que no funciona correctamente, por lo que se pasa al siguiente *sprint* para revisarlo de nuevo.

Burndown chart del sprint

De nuevo sucede lo mismo que en el sprint anterior debido a la historia de usuario no completada.

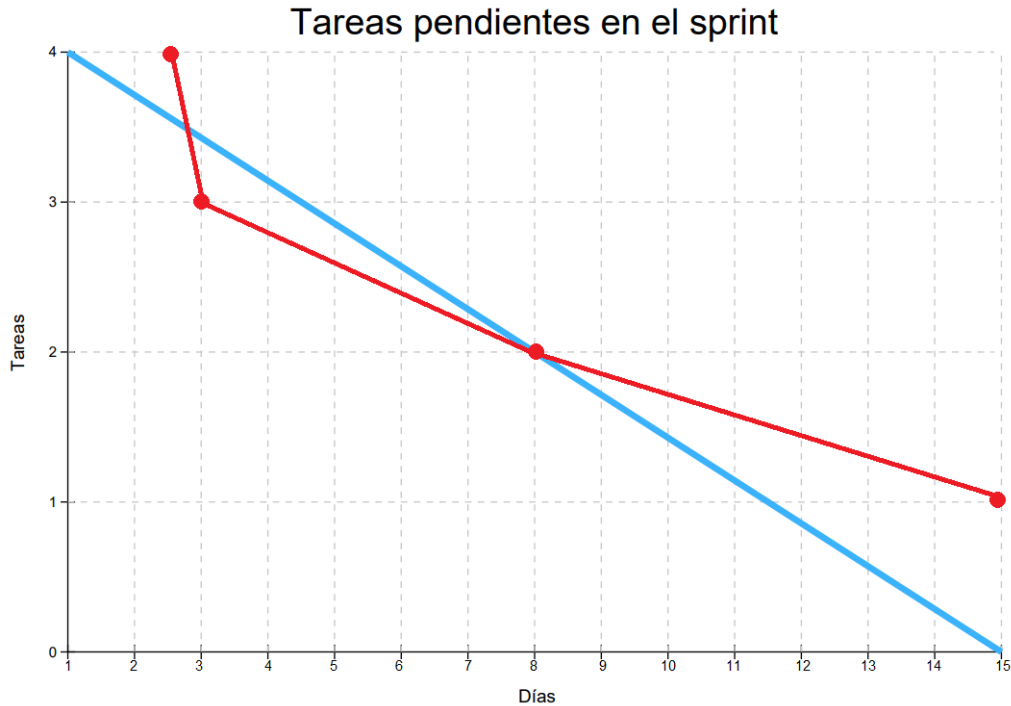


Figura 3.7: Burndown chart sprint 3.

3.5.4. Sprint 4***Spring backlog***

En este *sprint* se realizarán las siguientes historias de usuario: HU06, HU18, HU19, HU20, HU21, HU28, HU29.

Descripción

Se comienzan a implementar funcionalidades de la aplicación web además de terminar algunas del simulador. El dibujado de flechas (HU-06) continúa sin funcionar correctamente por lo que se pasa al siguiente *sprint*.

Burndown chart del sprint

En este *sprint*, se adelantó el trabajo con varias tareas, sin embargo, de nuevo sucede lo mismo que en los dos casos anteriores, debido a la historia de usuario no completada.

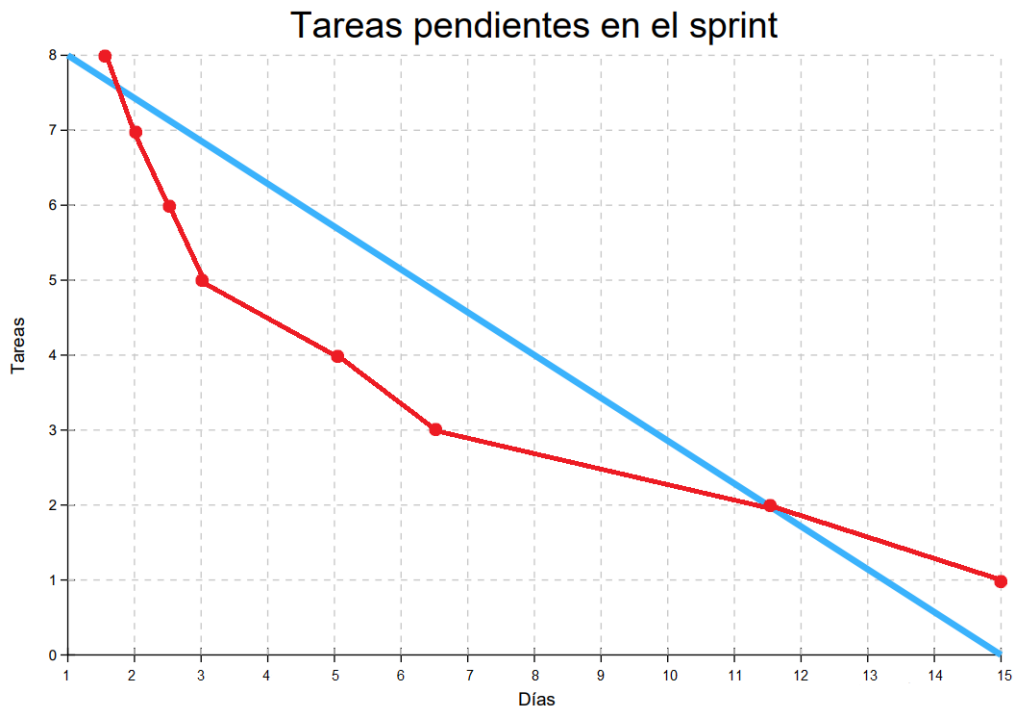


Figura 3.8: Burndown chart sprint 4.

3.5.5. Sprint 5

Spring backlog

En este *sprint* se realizarán las siguientes historias de usuario: HU06, HU22, HU23, HU24, HU25, HU26, HU27, HU30, HU31.

Descripción

En el último *sprint*, se termina la funcionalidad de la aplicación web y del simulador. De nuevo, el dibujo de flechas se realiza en este *sprint* y se termina.

Burndown chart del sprint

Finalmente, aunque se empieza con trabajo atrasado, se consigue llegar a la planificación deseada.

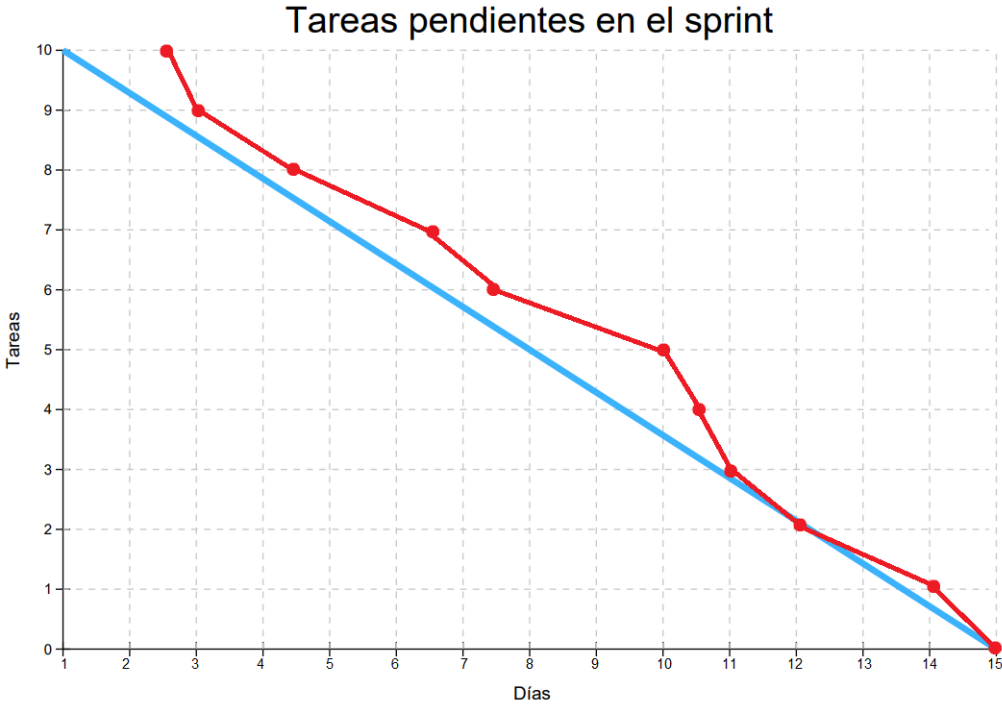


Figura 3.9: Burndown chart sprint 5.

Capítulo 4

Análisis y diseño del proyecto

4.1. Análisis

En los siguientes apartados se muestran los diagramas de análisis de la aplicación.

4.1.1. Modelo de dominio

En el caso del modelo de dominio, se cuenta con dos al haber desarrollado las dos partes de forma separada: uno para la web y otro para el simulador.

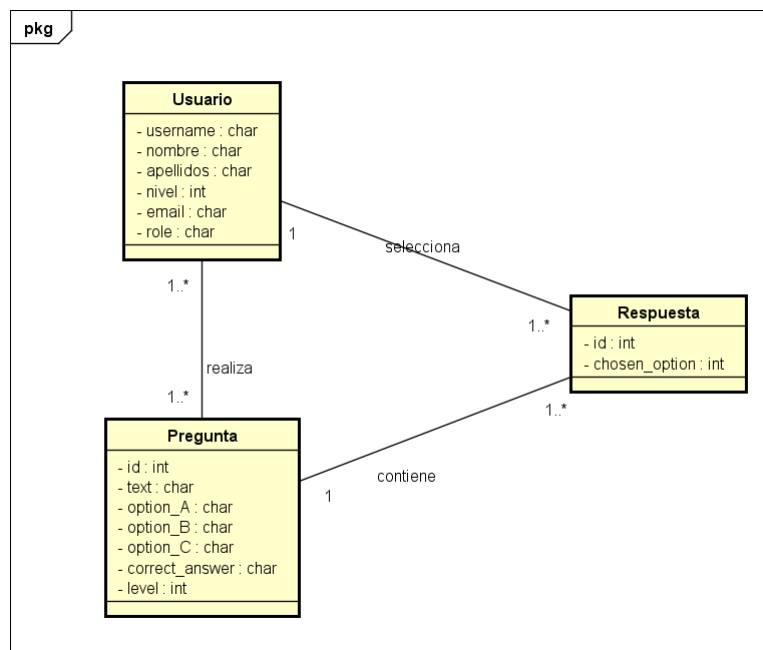


Figura 4.1: Modelo de dominio web.

En el caso de la web, son necesarias tres clases: Usuario, Pregunta y Respuesta. Cada usuario tiene varias respuestas, una por pregunta que realiza, sin embargo, puede realizar la cantidad de preguntas que quiera. Una pregunta, puede tener a su vez, asociadas varias respuestas (figura 4.1). Se entiende por respuesta la opción elegida por el usuario.

En el simulador, se utiliza una especialización ya que varias clases heredan de la clase Table al incorporar otras funcionalidades a mayores. Cada tabla, además, cuenta con varias pilas (figura 4.2).

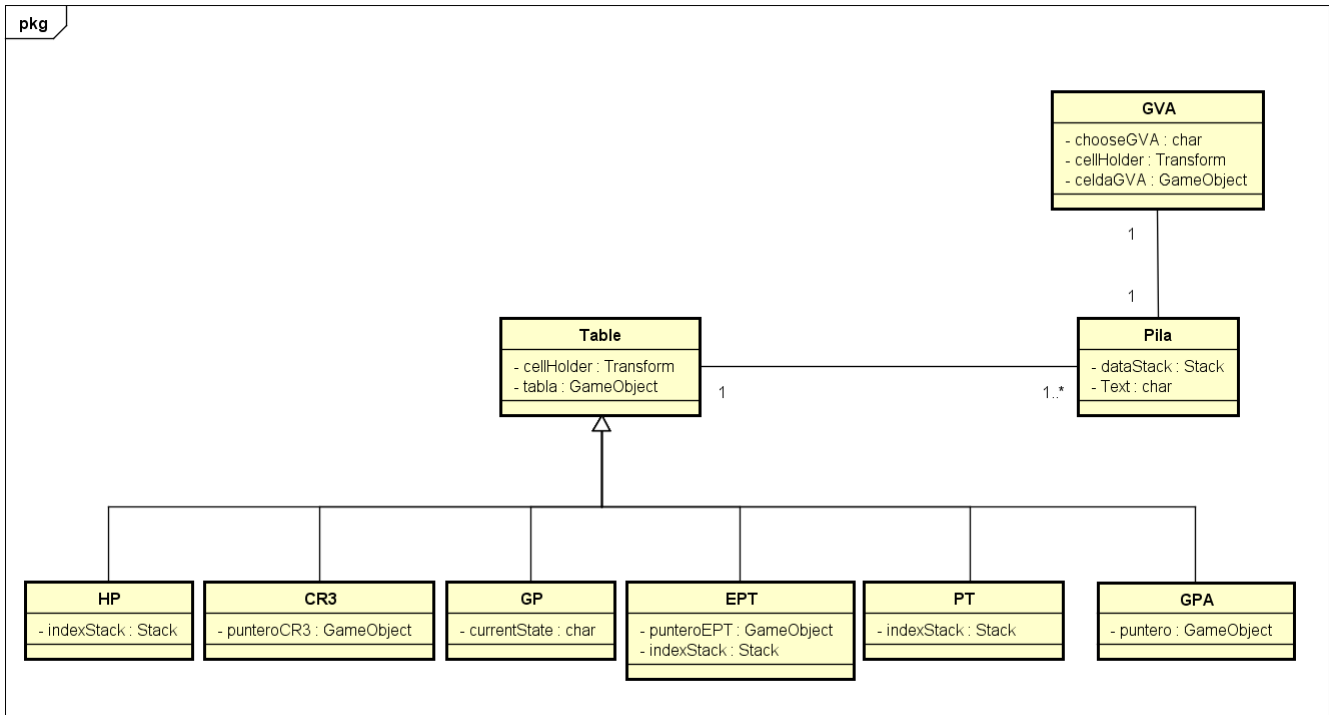


Figura 4.2: Modelo de dominio simulador.

4.1.2. Diagramas de secuencia

Se muestran los diagramas de secuencia a nivel análisis más importantes.

Para su realización se ha utilizado BCE (Boundary-Control-Entity), donde cada una representa lo siguiente:

- **Boundary:** presentan la información al usuario.
- **Control:** modelan el comportamiento.
- **Entity:** almacenan la información.

Ver Tests

El usuario solicita ver tests y el sistema le pide al control que los muestre, que accede a las preguntas para poder devolverlos y mostrarlos.

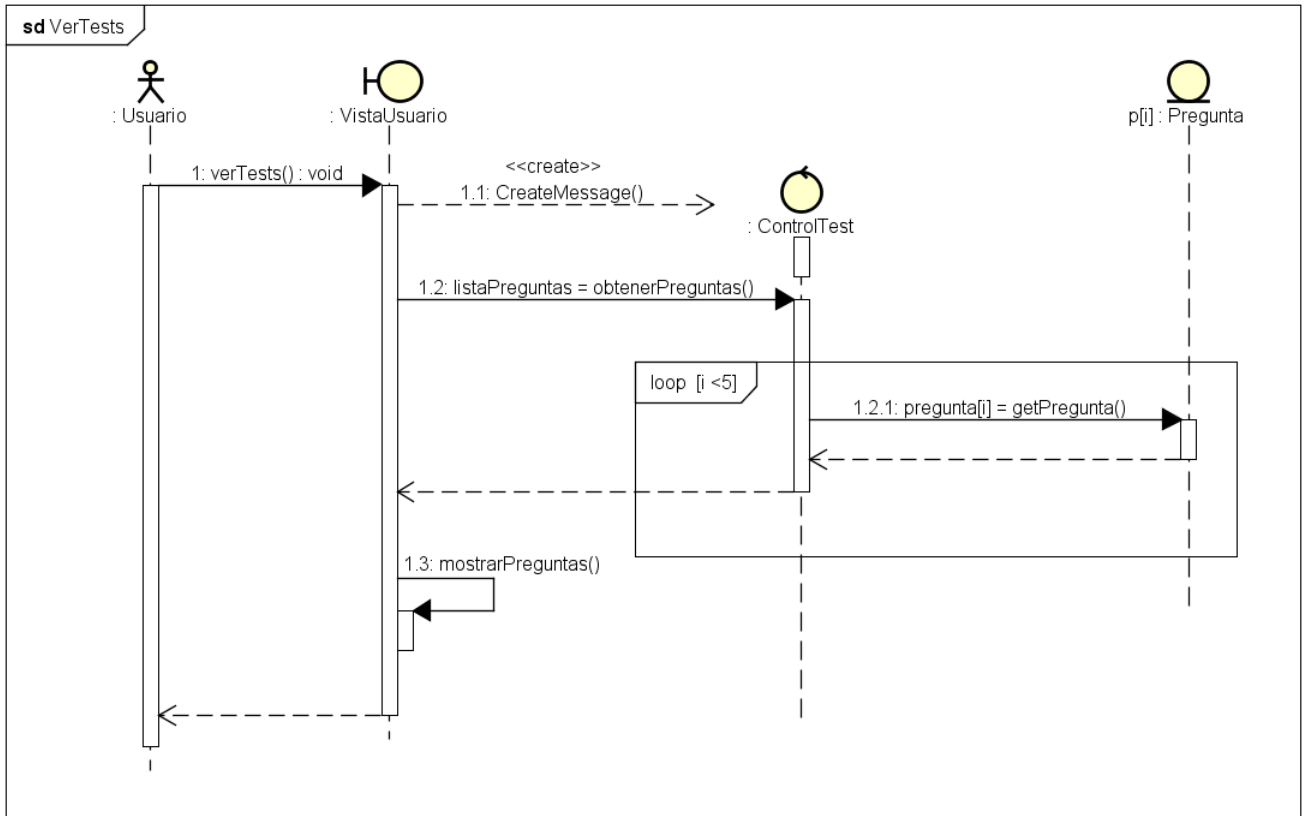


Figura 4.3: Ver Tests - Análisis.

Ver Ranking

El usuario solicita ver el ranking de usuarios y el sistema le pide al control que los muestre, que accede a los usuarios para poder devolverlos y mostrarlos.

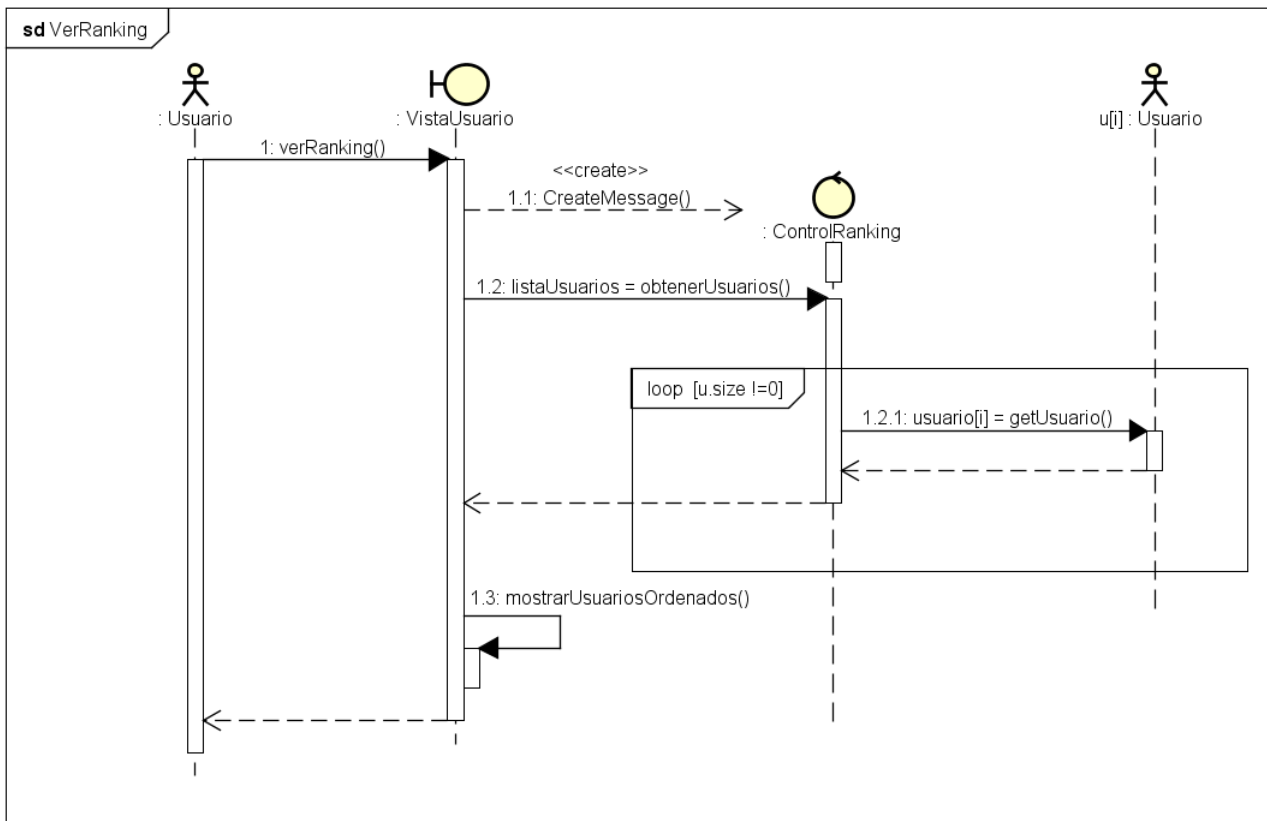


Figura 4.4: Ver Ranking - Análisis.

Elegir dirección

El usuario elige la dirección en la vista y el controlador la procesa para obtener el valor elegido y mostrarlo.

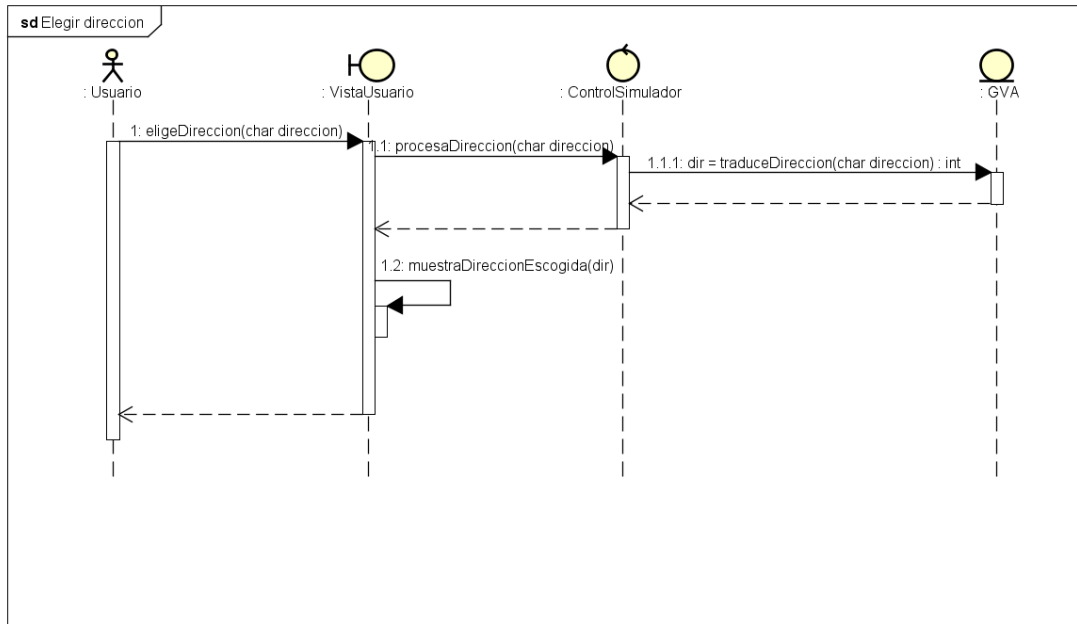


Figura 4.5: Elegir dirección - Análisis.

Reiniciar simulación

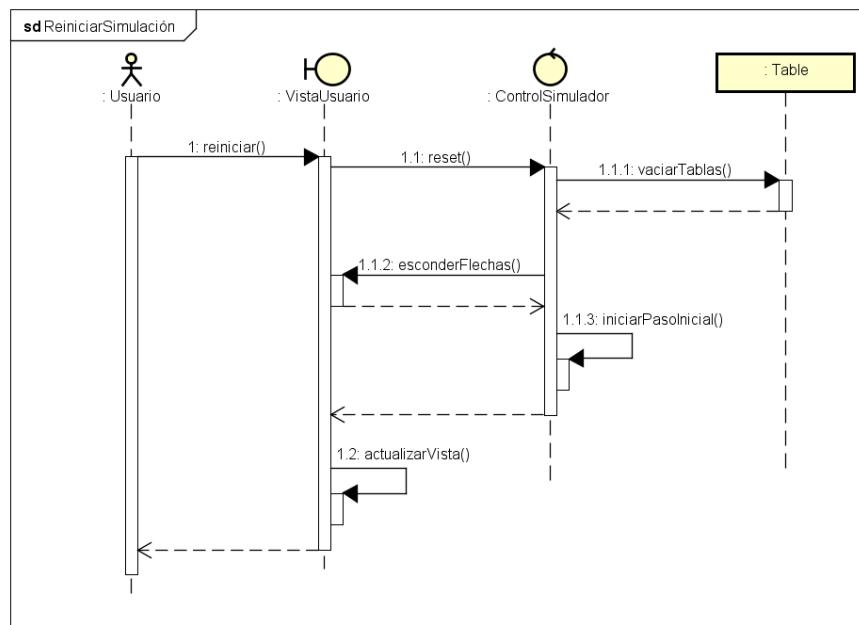


Figura 4.6: Reiniciar simulación - Análisis.

El usuario solicita reiniciar la simulación. El controlador, por su parte, vacía las tablas y esconde las flechas (figura 4.6).

4.1.3. Diagrama de clases

Web

El diagrama de clases para web incorpora distintas operaciones que son necesarias.

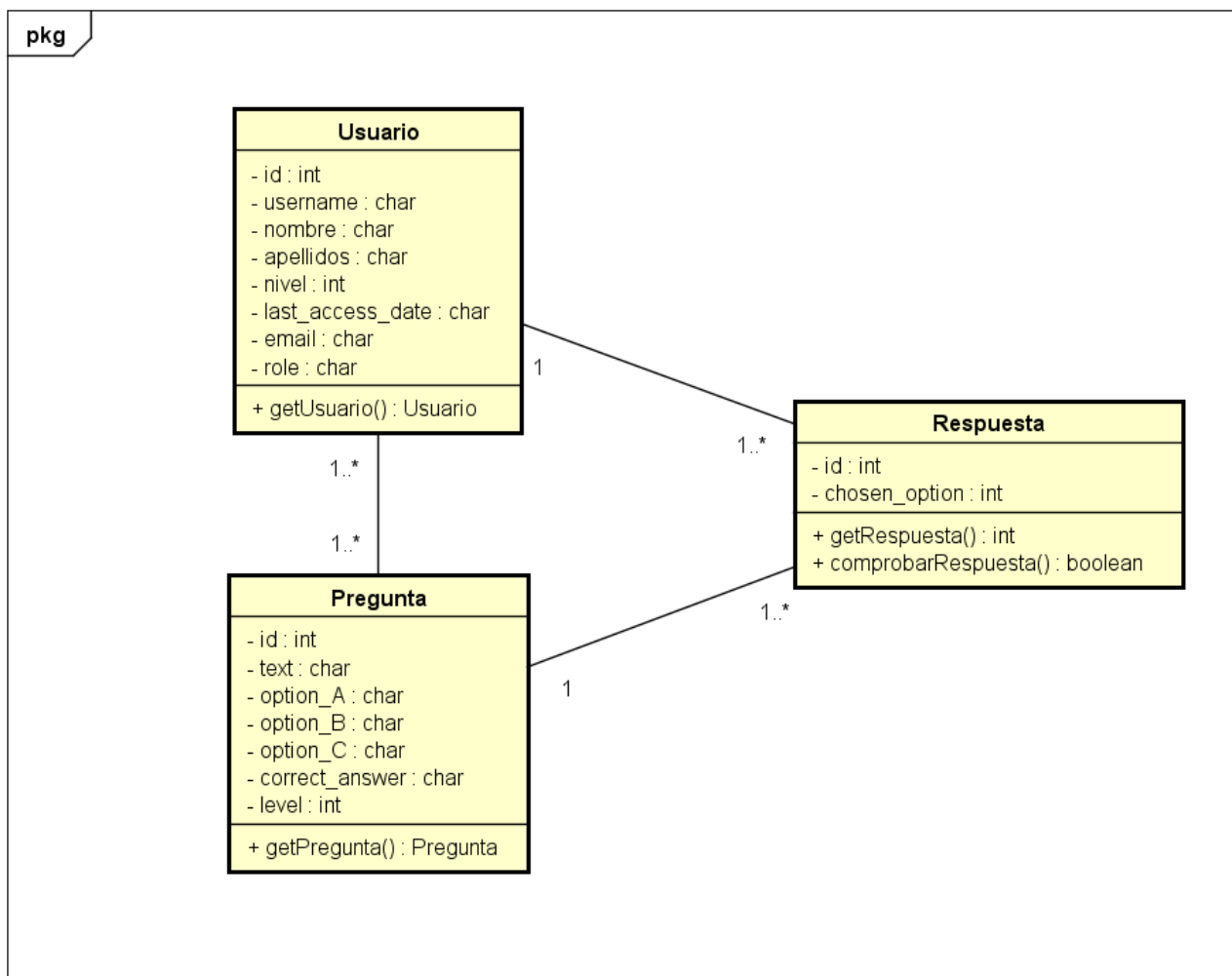


Figura 4.7: Diagrama de clases Web - Análisis.

Simulador

El diagrama de clases para el simulador incorpora distintas operaciones que son necesarias.

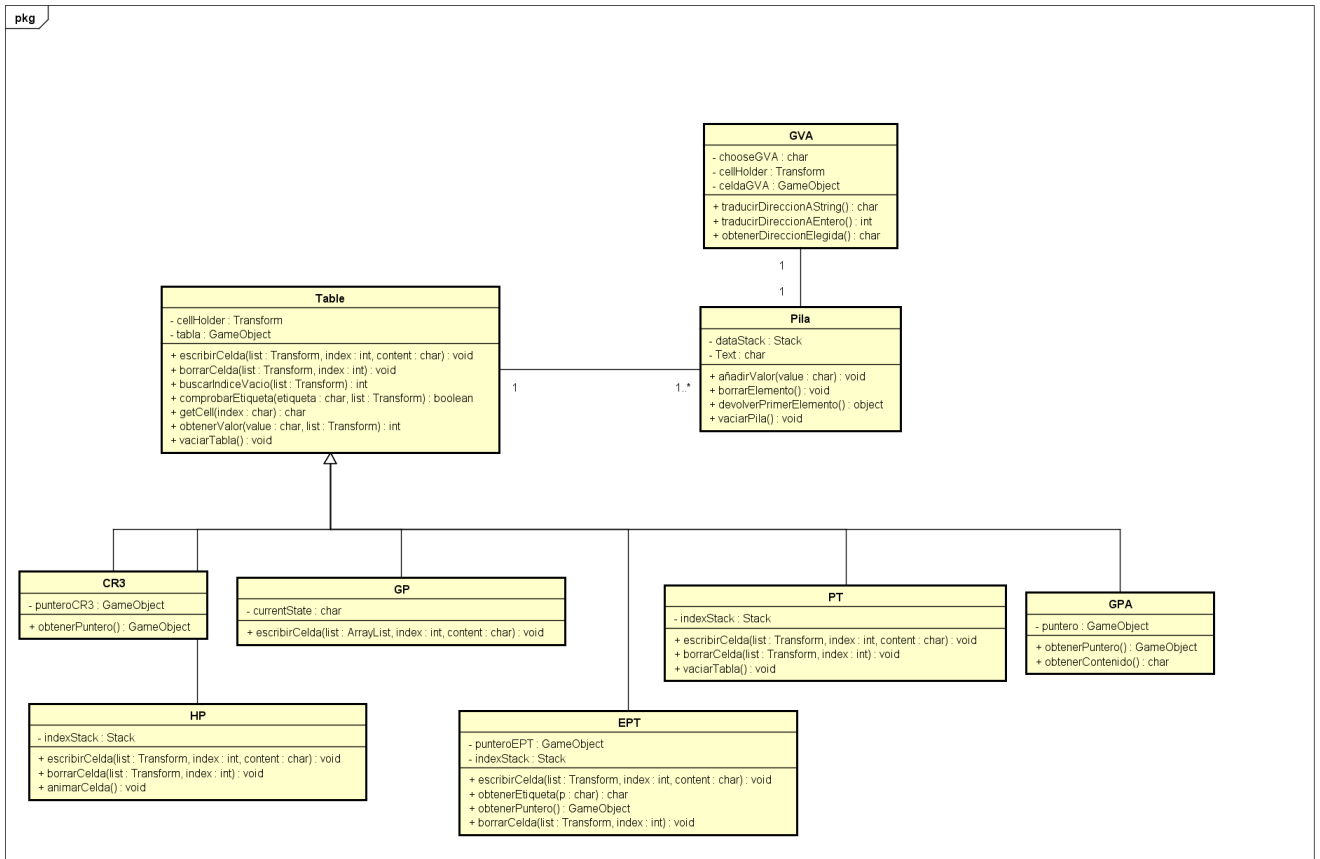


Figura 4.8: Diagrama de clases Simulador - Análisis.

4.2. Diseño

4.2.1. Arquitectura

A partir de las funcionalidades descritas en las historias de usuario y las herramientas utilizadas, en los siguientes apartados se describe la arquitectura usada.

4.2.2. Diagrama de despliegue

El diagrama de despliegue describe los distintos componentes hardware necesarios para la aplicación.

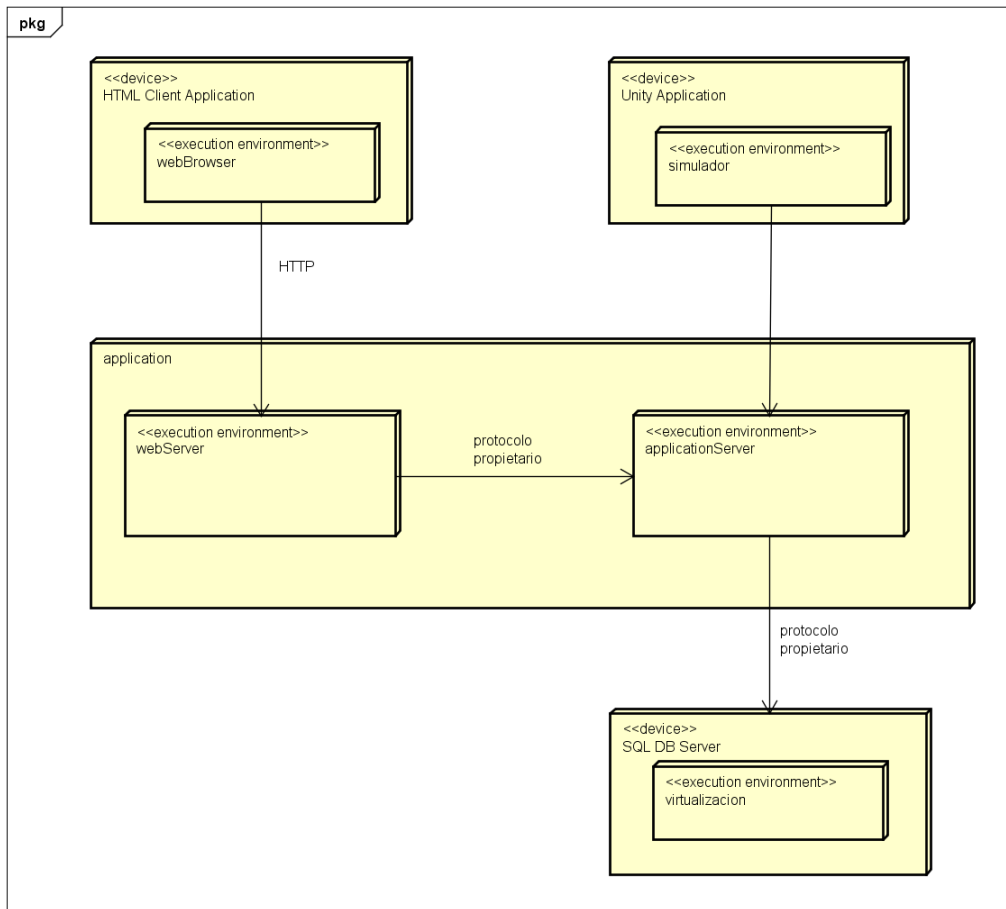


Figura 4.9: Diagrama de despliegue

Estos comprenden:

- **Clientes:** navegador web para acceder a la aplicación mediante la conexión al servidor web usando el protocolo HTTP.
- **Aplicación:** cuenta con dos entornos de ejecución, el servidor web, que recibe peticiones, y el servidor de aplicaciones, que proporciona la funcionalidad.
- **Aplicación Unity:** el simulador unido al servidor de aplicaciones al estar integrado en él.
- **Servidor de Bases de datos:** un servidor SQL con una base de datos llamada “virtualizacion”.

4.2.3. Patrones aplicados

MVC - Modelo Vista Controlador

El patrón Modelo-Vista-Controlador se utiliza cuando hay varias vistas que cambian con más frecuencia que los datos. Permite separar los objetos del dominio de las interfaces de usuario. En la figura

4.10, se muestra el funcionamiento de este patrón. El controlador puede modificar la vista y el modelo, y estos dos se pueden comunicar entre sí.

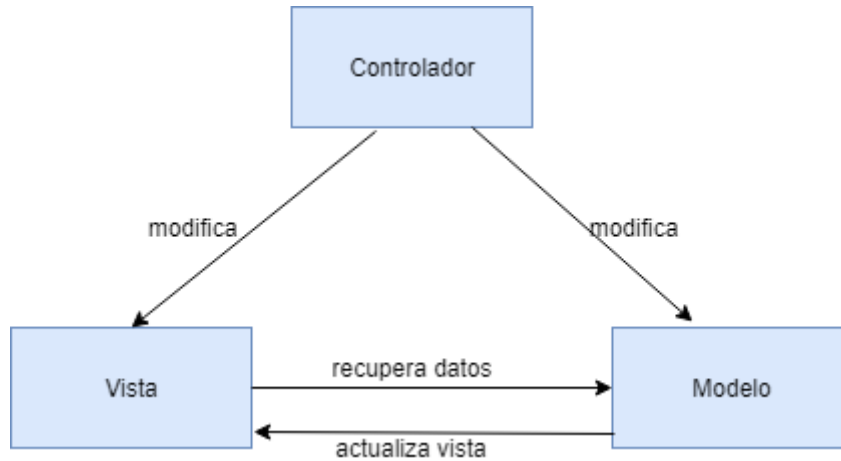


Figura 4.10: Modelo-Vista-Controlador

Domain Model

Mediante el uso de este patrón, se organiza el dominio alrededor de sus conceptos, el objeto de dominio incorpora tanto comportamiento como datos.

DAO - Data Access Object

El patrón DAO, permite encapsular el acceso a datos suministrando una interfaz común entre la aplicación y la base de datos. Gestiona la conexión para obtener y almacenar datos.

En este caso, se cuenta con tres tablas en la base de datos, Usuario, Pregunta y Respuesta y es necesaria una clase DAO para cada una de ellas.

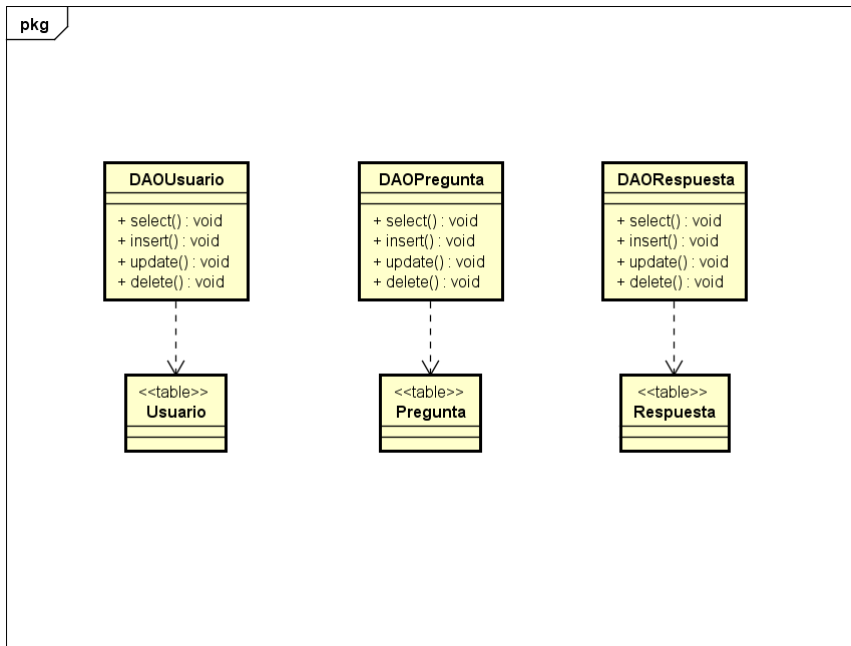


Figura 4.11: Patrón DAO

4.2.4. Diagramas de secuencia

Ver Tests

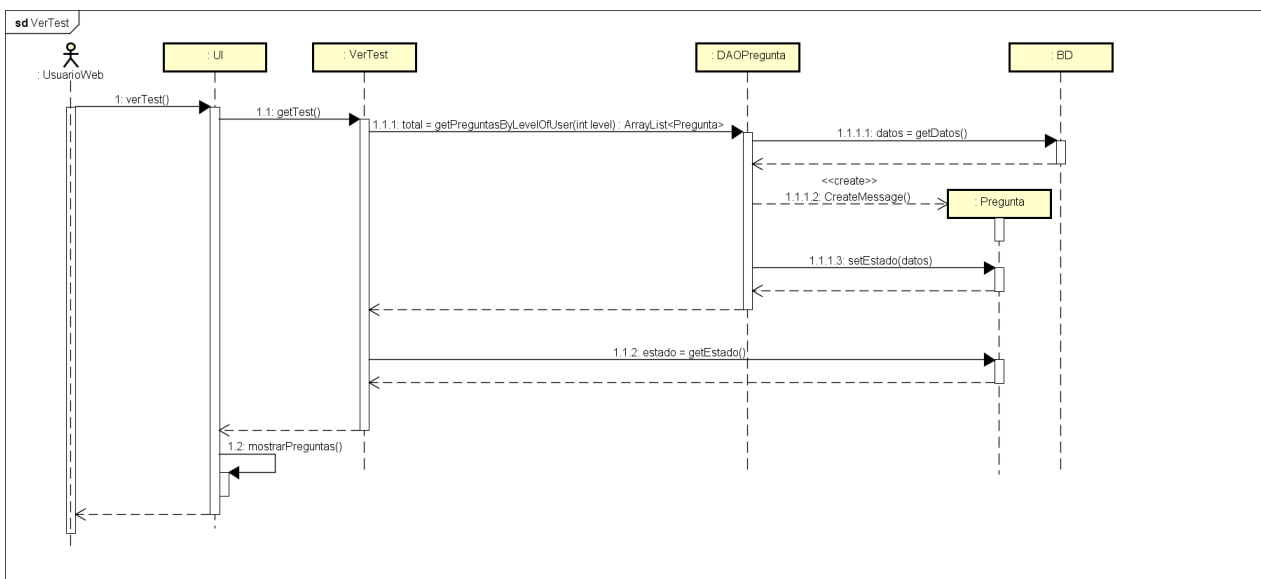


Figura 4.12: Ver Test - Diseño

Ver Ranking

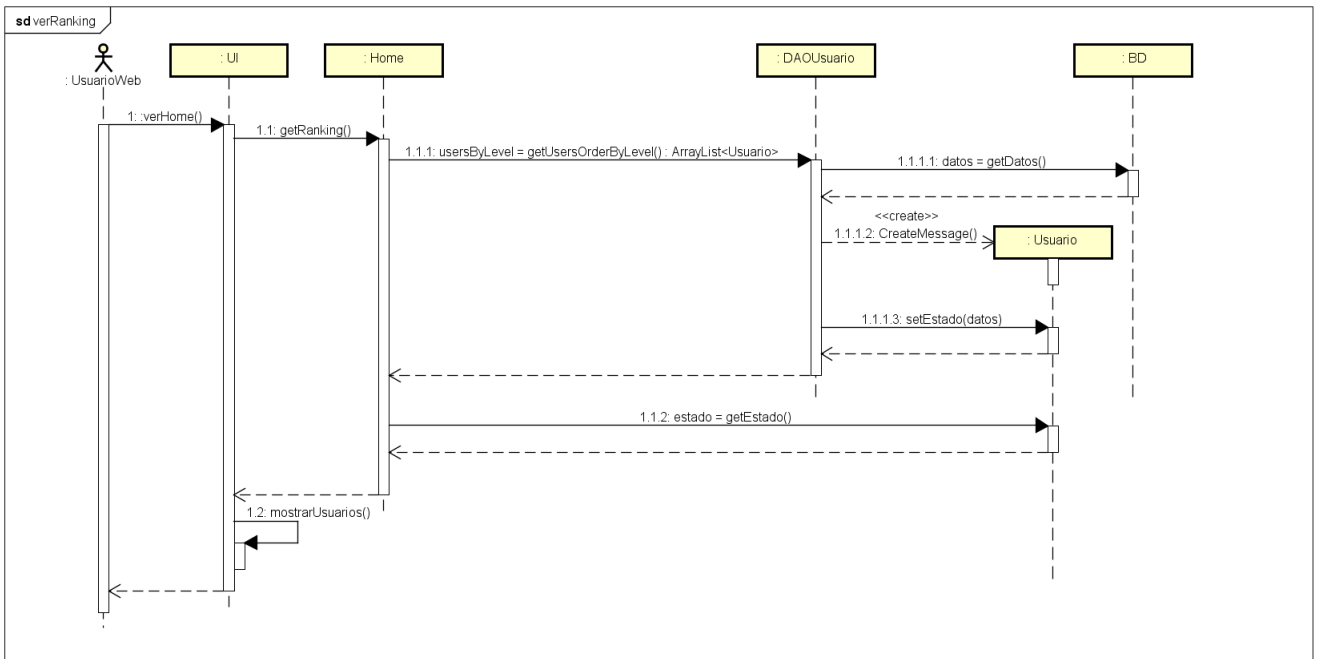


Figura 4.13: Ver Ranking - Diseño

Elegir dirección

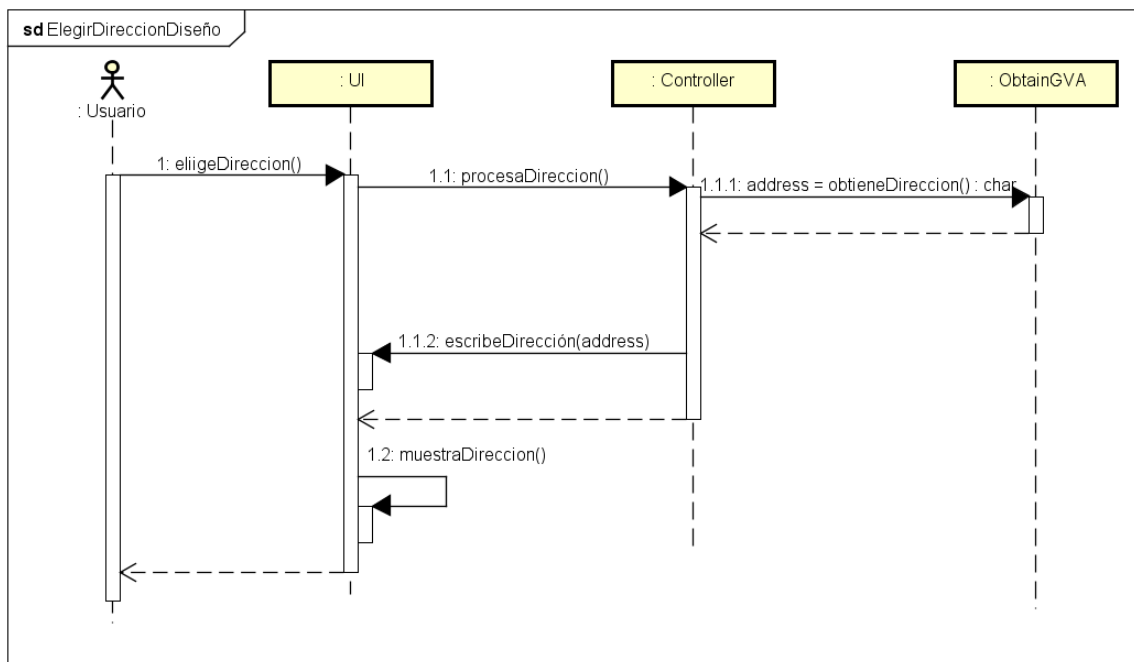


Figura 4.14: Elegir dirección - Diseño.

Reiniciar simulación

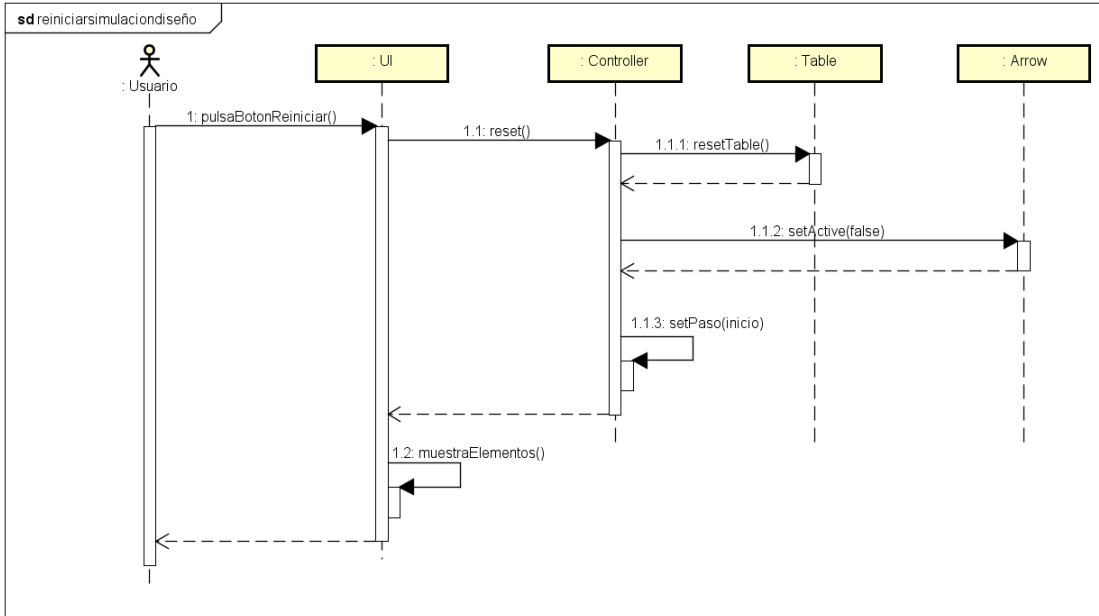


Figura 4.15: Reiniciar simulación - Diseño.

4.2.5. Diagrama de clases

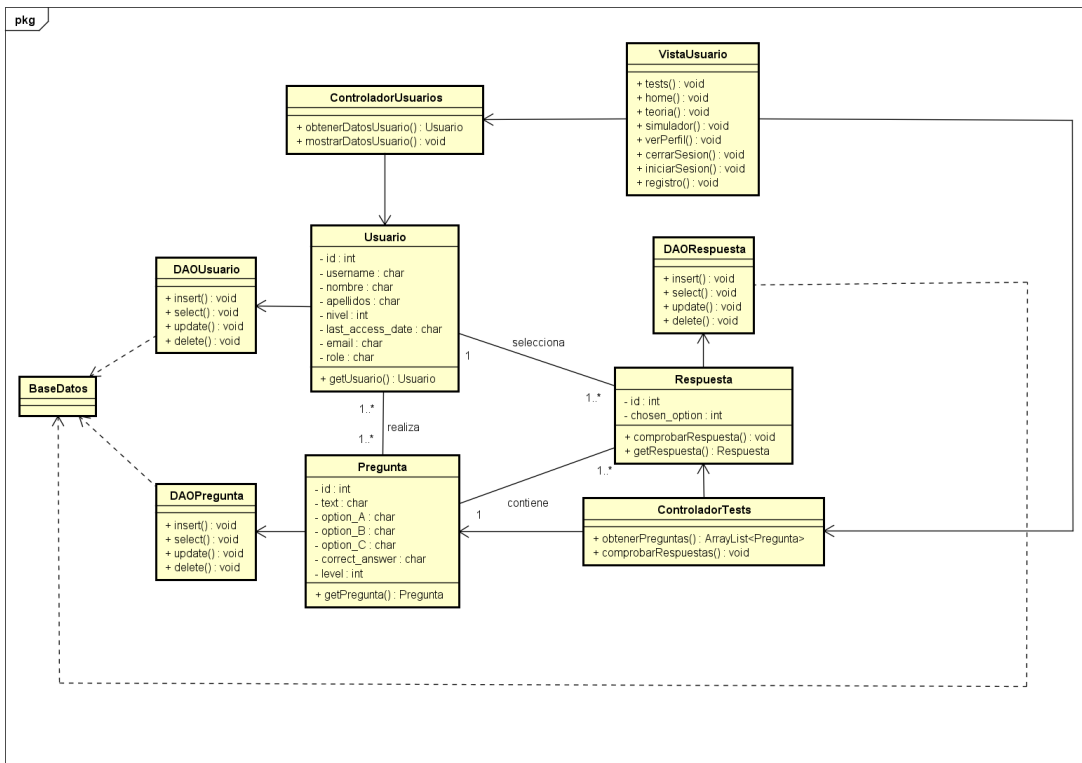


Figura 4.16: Diagrama de clases Web - Diseño

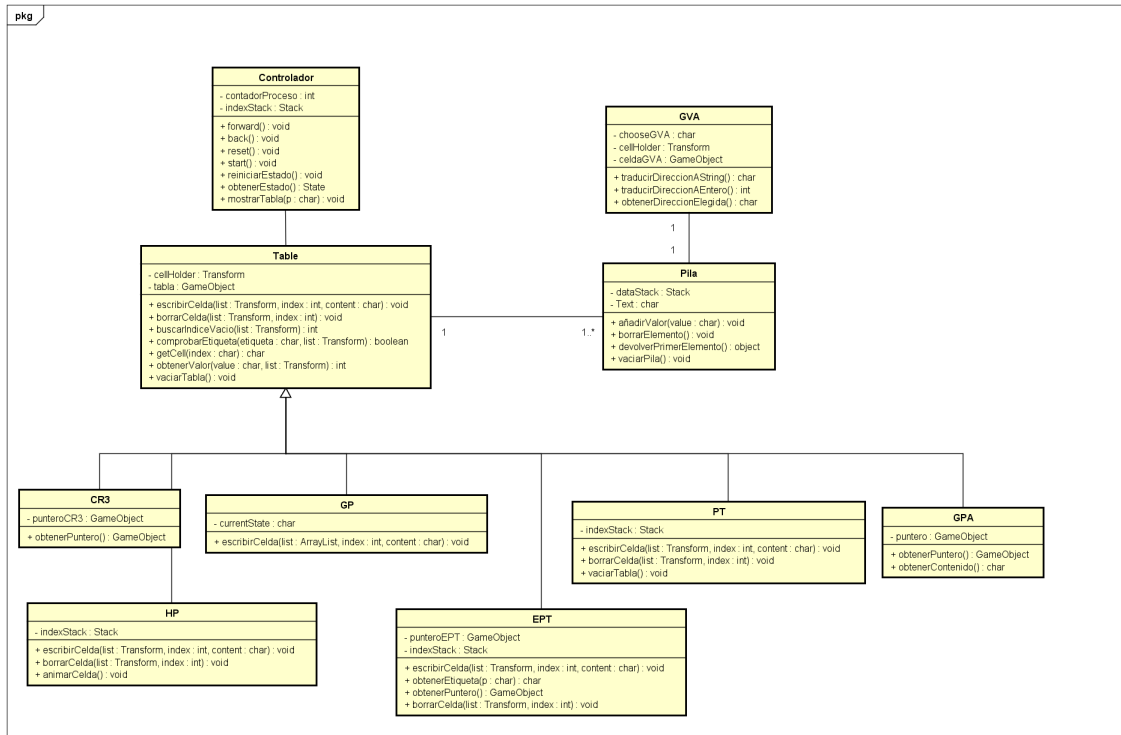


Figura 4.17: Diagrama de clases Simulador - Diseño

4.2.6. Modelo de base de datos

En la base de datos se tienen tres tablas: answers, users y questions. En la figura 4.18 se muestra la estructura de la base de datos.

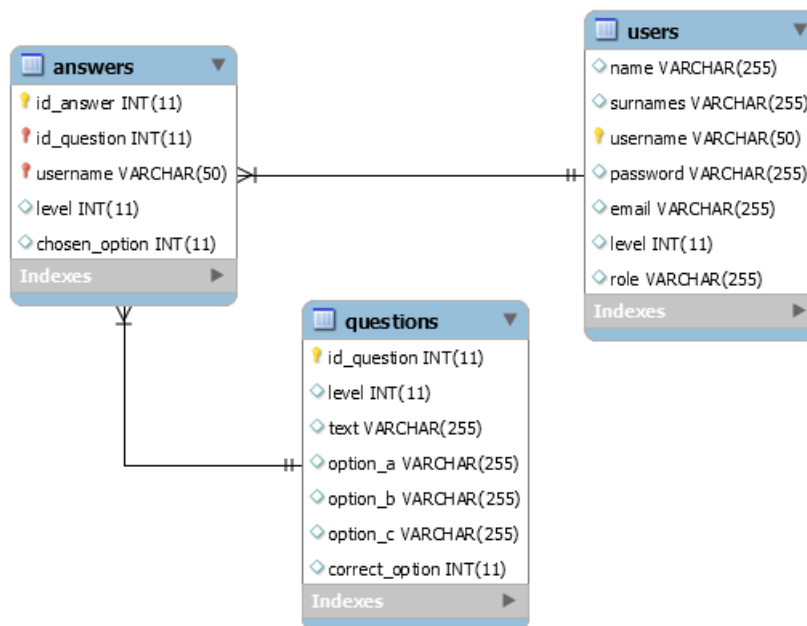


Figura 4.18: Esquema entidad-relación.

Capítulo 5

Implementación

5.1. Codificación

Cabe destacar en el desarrollo del código de este proyecto sobre todo la parte del simulador, al tener mayor complejidad, para la que se ha utilizado Unity y C# como lenguaje de programación. En los siguientes apartados se intenta explicar las partes más importantes de la implementación de algunos aspectos que pueden resultar interesantes.

También se explicarán algunos puntos destacables de la aplicación web, aunque en menor medida ya que son soluciones simples que no requieren aclaración.

5.1.1. Simulador

Estructuras de tablas

Las estructuras de las tablas necesarias, gráficamente, se han hecho mediante la combinación de varios elementos de la interfaz gráfica de usuario con los que cuenta Unity. En primer lugar, es necesario el uso de un *Canvas*. Este elemento es un área en el que se colocan todos los elementos que constituyen la interfaz.

Para la representación de las tablas, se han utilizado *panels* situados dentro de este elemento explicado. De modo que se tiene un padre compuesto por otros *panels*. A su vez, se han añadido otros elementos complementarios que serán necesarios: un texto para escribir el contenido de la celda y un *label*, necesario en algunos casos.

El resultado, es la obtención de una estructura similar a una lista compuesta por celdas, que comprenderían el elemento almacenado en cada posición.

Además, se utiliza herencia ya que se tiene una clase *Table* de la que heredan otras que sobreescriben métodos de esta o añaden nuevas funcionalidades a mayores.

Pasos de la simulación

Los pasos de la simulación se han obtenido mediante el uso de una máquina de estados. Para su implementación se ha utilizado una estructura de tipo *enum*, con todos los pasos como elementos. Para pasar de uno a otro, basta con incrementar o decrementar el valor de esta estructura.

Cada uno de los pasos se trata como un caso en el que se desarrollan ciertos eventos que se van mostrando gráficamente.

Borrado hacia atrás

Se trata de un problema complejo, pues hay que tener en cuenta que la simulación tiene que volver al estado del paso anterior. Para ello se pueden dar ciertas circunstancias: borrado de varias tablas, borrado de flechas, escribir el texto explicativo precedente...

El mayor reto es el de borrar el valor escrito en las tablas y mostrar el anterior si fuera necesario. Para ello, se han utilizado dos técnicas dependiendo de la situación. Las tablas representativas de las memorias físicas se rellenan de abajo arriba, sin embargo, las tablas de páginas y las EPT no, ya que el valor donde se escribe viene marcado por la dirección.

En el primer caso, se ha utilizado una pila por cada celda como una estructura de datos en la que se almacenan los valores escritos en dicha posición, de forma que al volver al paso anterior se borra el primero y se obtiene justo el que está inmediatamente debajo para mostrarlo.

En el segundo caso, también se utiliza una pila por celda para los valores escritos en cada posición. No obstante, para saber cuál es la posición donde se escribió en el último paso, se utiliza otra pila de valores de índices para conocer dónde hay que borrar.

Dibujado de flechas

Las flechas que van apareciendo para clarificar lo que sucede en cada paso se dibujan de forma dinámica mediante un elemento de Unity llamado *Line Renderer*. Este componente, permite dibujar líneas indicando cuantas posiciones tienen y cuáles son estas. En el caso de esta problemática, tienen dos posiciones, inicio y fin.

El final de la flecha está marcado por el inicio de la celda donde se escribe y el principio de esta se calcula a partir de una función que calcula el punto medio entre dos puntos y un *offset* establecido.

Detección de espacio no disponible

La detección de espacio no disponible en la memoria física se realiza mediante un evento que se activa cuando se da este caso. Al activarse, se ejecuta un método que permite desactivar las flechas y

Implementación

mostrar un mensaje en rojo de finalización. Así, se paraliza la simulación.

Este evento se establece para la tabla de la memoria física donde es necesario hacer esta comprobación de espacio.

5.1.2. Web

Obtención de preguntas

Las preguntas que se muestran en el apartado de tests se consiguen mediante una consulta a base de datos. En concreto, se buscan las que tengan el mismo nivel que el usuario y se obtienen cinco de estas de forma aleatoria.

Para almacenarlas se utiliza una estructura de tipo *ArrayList* de preguntas que se recorrerá después para mostrar los datos que interesan de cada una.

Corrección de respuestas

Tras realizar los tests y dar a enviar, el alumno puede ver inmediatamente la solución a las preguntas y saber cuáles están bien o mal. Se compara la respuesta elegida con la correcta de la pregunta, si coinciden, se muestra coloreada en verde la respuesta, y, si no lo es, en rojo. También se puede dar el caso de que no responda alguna pregunta. Esto se tiene en cuenta, mostrando un mensaje al usuario y no contabilizando la respuesta como verdadera o falsa.

5.2. Herramientas utilizadas

En los siguientes apartados se describen las herramientas y tecnologías utilizadas en este proyecto y la razón de su elección. Algunas de ellas desconocidas al comienzo y se han ido aprendiendo durante el desarrollo.

5.2.1. Unity

Unity es un motor de videojuegos multiplataforma creado por Unity Technologies en el año 2005. Permite crear aplicaciones para Web, Windows, dispositivos móviles y distintas consolas como *Xbox* o *Play Station*. Entre sus características principales destacan: compatibilidad de uso con otros motores como Blender o Adobe Photoshop, desarrollo para varias plataformas y múltiples lenguajes para el *scripting* como C#, Boo o UnityScript.

En cuanto a las licencias, cuenta con tres: personal, plus y pro. Siendo la primera gratis y el resto de pago. En este proyecto se ha utilizado la versión personal al ser la más asequible, además de estar

orientada a estudiantes y principiantes en la plataforma.

La opción WebGL para construir el proyecto permite publicar contenido web desarrollado con Unity y usar su API para ejecutarlo en un navegador. Es necesario usar esta opción, al integrar el simulador dentro de la web. Cuando se compila con esta opción, se generan los siguientes archivos: un fichero *index.html*, una carpeta *Template Data* y una carpeta *Build*. El primer fichero es el que utiliza el navegador para acceder a ello. Las carpetas citadas contienen componentes para personalizar la página y los ficheros generados en la compilación, respectivamente.

Se decidió utilizar esta herramienta por ser la más completa para implementar el simulador, sobre todo, por su interfaz gráfica que permite diseñarlo de forma sencilla. Además, supone un reto aprender una tecnología completamente nueva.

C# y Visual Studio

El lenguaje C# está incluido en el paradigma de programación orientado a objetos desarrollado por *Microsoft* como parte de .NET. Es similar al lenguaje Java, es por esto que se eligió como opción para el desarrollo y no las otras que permite Unity.

Visual Studio, por otro lado, es un IDE para sistemas operativos Windows que permite programar en varios lenguajes, entre los que destacan C#. Las licencias con las que cuenta son tres: community, professional y enterprise. En este caso, se utilizó la licencia proporcionada por Windows para estudiantes.

Se decidió utilizar esta herramienta y no la nativa de Unity porque ofrece una mejor interfaz que facilita la programación.

5.2.2. Netbeans

Se trata de un IDE libre creado en el año 2000 por la *Apache Software Foundation* que soporta principalmente el lenguaje Java y permite integrar varios Frameworks como Java Spring MVC o Hibernate. Aunque también permite integrar otros lenguajes como C# y C++. A su vez, se puede crear un proyecto para Java Web y utilizar el servidor Apache Tomcat para alojarlo.

Se escogió esta opción al ser conocida y permitir integrar todo lo necesario para desarrollar la parte web del proyecto de forma sencilla. A pesar de integrar otro servidor, se decide usar Tomcat, de nuevo por ser una herramienta ya conocida previamente.

HTML y CSS

HyperText Markup Language o HTML es un lenguaje de marcado utilizada para desarrollar páginas web permitiendo definir la estructura básica de estas. A su vez, es un estándar que han adoptado todos

Implementación

los navegadores web.

Por otro lado, CSS o *Cascading Stylesheets* nos permite darle estilo a lo que vamos creando. En este caso, se ha usado el *framework* Bootstrap que contiene plantillas de diseño. La ventaja de esta herramienta es que es sencilla de usar y el resultado es mejor que el de crear las clases de estilo por uno mismo.

El uso de Bootstrap facilita el uso de estilos además de dar un aspecto más elegante. Su amplia documentación y tutoriales hacen que utilizarlo sea muy sencillo. Por ello, se decide utilizarlo en este proyecto.

Java Web

La aplicación web se desarrolla con Java utilizando la tecnología *servlet*. Se trata de clases de este lenguaje que se utilizan para procesar solicitudes de un servidor web. Pueden procesar peticiones de todo tipo, pero comúnmente se utilizan del lado del servidor.

Por otro lado, se utilizan páginas JSP o *JavaServer Pages* que permiten crear páginas dinámicas utilizando HTML y el lenguaje Java.

Esta tecnología se escoge en este caso debido al conocimiento previo de ella y por su facilidad de uso.

5.2.3. Apache Tomcat

Tomcat es un servidor creado por la *Apache Software Foundation* en el año 1999 que permite contener aplicaciones desarrolladas con tecnologías Java Web como los *servlets*. El IDE utilizado, Netbeans, permite instalar este servidor de forma integrada a él de forma que es más accesible y hace que la configuración sea más sencilla.

Apache Derby y JDBC

Apache Derby es un sistema gestor de bases de datos relacional que utiliza el lenguaje SQL para sus consultas. Se utiliza de una forma sencilla al estar integrado en Netbeans. Para conectarse a esta base de datos mediante Java se utiliza JDBC o *Java Database Connectivity* independiente del sistema operativo que se utilice.

Se ha decidido utilizar Derby ya que su integración con Tomcat lo hace sencillo de usar y el lenguaje para las consultas no difiere de los ya conocidos.

5.3. Pruebas realizadas

Las pruebas realizadas se llevan a cabo a la vez que se implementan las historias de usuario, teniendo en cuenta el resultado que se desea obtener y el que se obtiene realmente. Algunas historias de usuario más complejas han sido más complicadas de conseguir, de ahí que hayan fallado más veces.

	Historia de usuario	Resultado esperado
TEST 01	HU01 - Crear tablas de las estructuras.	Se crean distintas tablas que se muestran ordenadas en la pantalla.
Número	1	
Validación:	OK	
Resultado obtenido:	El esperado.	

Tabla 5.1: TEST 01.

	Historia de usuario	Resultado esperado
TEST 02	HU02 - Elegir dirección a traducir.	Se puede elegir una dirección de un dropdown.
Número	1	2
Validación:	FALLO	OK
Resultado obtenido:	Escribe una opción vacía.	El esperado.

Tabla 5.2: TEST 02.

	Historia de usuario	Resultado esperado
TEST 03	HU03 - Flechas para ir hacia delante y retroceder	Aparecen dos botones de flecha derecha e izquierda en los que se puede hacer click.
Número	1	
Validación:	OK	
Resultado obtenido:	El esperado.	

Tabla 5.3: TEST 03.

	Historia de usuario	Resultado esperado		
TEST 04	HU04 - Pasos.	Se puede avanzar y retroceder los pasos del simulador volviendo al estado anterior.		
Número	1	2	3	4
Validación:	FALLO	FALLO	FALLO	OK
Resultado obtenido:	El estado inicio y final no coinciden con lo que debería pasar.	Se muestran elementos que no deberían en algunos pasos.	No se escribe en las posiciones correctas.	El esperado.

Tabla 5.4: TEST 04.

Implementación

	Historia de usuario	Resultado esperado
TEST 05	HU05 - Texto con explicación	Aparece un texto explicativo debajo de lo botones de flecha.
Número	1	
Validación:	OK	
Resultado obtenido:	El esperado.	

Tabla 5.5: TEST 05.

	Historia de usuario	Resultado esperado			
TEST 06	HU06 - Mostrar flechas en el proceso.	En cada paso aparecen las flechas necesarias.			
Número	1	2	3	4	5
Validación:	FALLO	FALLO	FALLO	FALLO	OK
Resultado obtenido:	Las flechas no aparecen en el lugar correcto.	Las puntas de flecha no se muestran.	Algunas flechas no se muestran.	Las flechas no se dibujan en la posición correcta.	El esperado.

Tabla 5.6: TEST 06.

	Historia de usuario	Resultado esperado
TEST 07	HU07 - Mostrar etiquetas en las tablas.	Se muestran etiquetas que indican qué es cada elemento.
Número	1	
Validación:	OK	
Resultado obtenido:	El esperado.	

Tabla 5.7: TEST 07.

	Historia de usuario	Resultado esperado
TEST 08	HU08 - Numerar celdas.	Las tablas tienen etiquetas con cada posición.
Número	1	
Validación:	OK	
Resultado obtenido:	El esperado.	

Tabla 5.8: TEST 08.

	Historia de usuario	Resultado esperado
TEST 09	HU09 - Puntero CR3. HU10 - Puntero EPT.	Se muestra una flecha representando el puntero.
Número	1	
Validación:	OK	
Resultado obtenido:	El esperado.	

Tabla 5.9: TEST 09.

TEST 10	Historia de usuario	Resultado esperado
	HU11 - Mostrar animación dirección elegida.	Aparece una animación que destaca en color amarillo la dirección elegida en la memoria física de la máquina virtual.
Número	1	2
Validación:	FALLO	OK
Resultado obtenido:	Solo se muestra la primera vez.	El esperado.

Tabla 5.10: TEST 10.

TEST 11	Historia de usuario	Resultado esperado	
	HU12 - Borrado	Al retroceder se borran los datos que se escribieron en el paso siguiente.	
Número	1	2	3
Validación:	FALLO	FALLO	OK
Resultado obtenido:	No se borran ni las flechas ni de las tablas EPT ni PT.	No se borra de las tablas EPT ni PT.	El esperado.

Tabla 5.11: TEST 11.

TEST 12	Historia de usuario	Resultado esperado
	HU13 - GPA vacío.	Cuando no se está traduciendo una dirección GPA esta se muestra vacía.
Número	1	
Validación:	OK	
Resultado obtenido:	El esperado.	

Tabla 5.12: TEST 12.

TEST 13	Historia de usuario	Resultado esperado	
	HU14 - Reiniciar simulación.	Todos los valores de las tablas vuelven a su estado inicial y se ocultan las que no aparecen al principio.	
Número	1	2	3
Validación:	FALLO	FALLO	OK
Resultado obtenido:	No se borran todos los valores correctamente.	No se borran todas las flechas.	El esperado.

Tabla 5.13: TEST 13.

	Historia de usuario	Resultado esperado
TEST 14	HU10 - Mostrar animación dirección GPA a traducir.	Aparece una animación que destaca en color rojo la dirección GPA que se va a traducir.
Número	1	2
Validación:	FALLO	OK
Resultado obtenido:	Solo se muestra la primera vez.	El esperado.

Tabla 5.14: TEST 14.

	Historia de usuario	Resultado esperado
TEST 15	HU16 - Mostrar etiquetas en las flechas. HU17 - Mostrar etiquetas en las direcciones.	Se muestran etiquetas que indican qué es cada elemento.
Número	1	2
Validación:	FALLO	OK
Resultado obtenido:	No se muestran bien las etiquetas de las flechas.	El esperado.

Tabla 5.15: TEST 15.

	Historia de usuario	Resultado esperado
TEST 16	HU18 - Número de proceso.	Se muestra el proceso actual.
Número	1	
Validación:	OK	
Resultado obtenido:	El esperado.	

Tabla 5.16: TEST 16.

	Historia de usuario	Resultado esperado
TEST 07	HU19 - Terminar simulación al no haber espacio.	No se pueden utilizar las flechas y aparece un mensaje en rojo.
Número	1	2
Validación:	FALLO	OK
Resultado obtenido:	Las flechas no se desactivan.	El esperado.

Tabla 5.17: TEST 17.

	Historia de usuario	Resultado esperado
TEST 18	HU20 - Login HU21 - Logout	Se puede iniciar y cerrar sesión.
Número	1	2
Validación:	FALLO	OK
Resultado obtenido:	No muestra la pantalla principal al iniciar sesión.	El esperado.

Tabla 5.18: TEST 18.

	Historia de usuario	Resultado esperado
TEST 19	HU22 - Registro.	Se puede registrar un nuevo usuario.
Número	1	2
Validación:	FALLO	OK
Resultado obtenido:	No guarda el usuario en la base de datos.	El esperado.

Tabla 5.19: TEST 19.

	Historia de usuario	Resultado esperado
TEST 20	HU23 - Ver ranking	Se muestra un ranking de los diez mejores usuario colocados por nivel.
Número	1	
Validación:	OK	
Resultado obtenido:	El esperado.	

Tabla 5.20: TEST 20.

	Historia de usuario	Resultado esperado
TEST 21	HU24 - Ver teoría.	Se muestra la teoría colocada por bloques.
Número	1	
Validación:	OK	
Resultado obtenido:	El esperado.	

Tabla 5.21: TEST 21.

	Historia de usuario	Resultado esperado
TEST 22	HU25 - Tests. HU26 - Ver soluciones del test.	Se muestran las preguntas y tras contestar se ven las soluciones.
Número	1	2
Validación:	FALLO	OK
Resultado obtenido:	No se muestra la pregunta correcta cuando la respuesta es incorrecta.	El esperado.

Tabla 5.22: TEST 22.

	Historia de usuario	Resultado esperado
TEST 23	HU27 - Ver simulador.	Se muestra el simulador en la pestaña correspondiente.
Número	1	
Validación:	OK	
Resultado obtenido:	El esperado.	

Tabla 5.23: TEST 23.

Implementación

	Historia de usuario	Resultado esperado
TEST 24	HU28 - Ver perfil. HU29 - Número de preguntas acertadas. HU30 - Nivel.	Se muestran los datos del usuario.
Número	1	2
Validación:	FALLO	OK
Resultado obtenido:	No se muestran correctamente las preguntas acertadas.	El esperado.

Tabla 5.24: TEST 24.

	Historia de usuario	Resultado esperado
TEST 25	HU31 - Subir nivel.	El usuario sube de nivel cuando llega a un número de preguntas.
Número	1	
Validación:	OK	
Resultado obtenido:	El esperado.	

Tabla 5.25: TEST 25.

Capítulo 6

Conclusiones y trabajo futuro

Como resultado de este proyecto se ha obtenido una herramienta *online* cuyo fin es permitir a los alumnos aprender de manera más sencilla un problema que puede parecer muy complejo. A pesar de ser un tema de gran importancia en la actualidad, no se han encontrado muchos recursos que faciliten su aprendizaje. Sin embargo, gracias a la herramienta Unity se ha podido realizar el simulador y aprender una tecnología completamente nueva.

Durante estos años en el grado se han adquirido muchos conocimientos que se han podido aplicar a este proyecto, como el conocimiento en desarrollo de páginas web, la planificación de proyectos y las bases de datos. Todo esto ha permitido, junto a lo estudiado nuevo, construir la solución propuesta. A su vez, se han aprendido aspectos de otros temas como la gamificación y los juegos serios, añadiendo puntos característicos de estos.

Aunque en este caso no se trata de un juego serio, en un futuro se pueden añadir nuevas funcionalidades que lo convertirían en uno. Se podría conseguir mediante la implementación de los tests gráficos. También sería interesante añadir un rol profesor como usuario en la aplicación, que le permitiera acceder a datos de alumnos para ver su progreso, y añadir nuevas preguntas a los tests.

Por otra parte, hay otros aspectos que se podrían incorporar y que añadirían complejidad al simulador, como pueden ser: utilizar la estructura TLB, añadir varios procesos en distintas máquinas virtuales o permitir al usuario elegir el tamaño de las direcciones a traducir y las estructuras.

La sencillez de la solución desarrollada permite que sea fácil de utilizar para el alumno y que no le produzca rechazo a la hora de utilizarlo, dos puntos claves para que sea una buena herramienta que facilita el aprendizaje.

Apéndices

Apéndice A

Acrónimos

- **GVA:** Guest Virtual Address.
- **GPA:** Guest Physical Address.
- **HPA:** Host Physical Address.
- **TLB:** Translation Lookaside Buffer.
- **VMM:** Virtual Machine Manager.
- **API:** Application Program Interface.
- **IDE:** Integrated Development Environment.
- **JRE:** Java Running Environment.
- **JDK:** Java Development Kit.

Apéndice B

Manual de instalación

En los siguientes puntos se describen los pasos a seguir para instalar cada herramienta necesaria para ejecutar la aplicación web.

B.1. Instalación de Java

Para la instalación de Java desde un sistema operativo Windows basta con descargarse el instalador de la última versión desde la página oficial de Java, es decir, la 8. También es necesario instalarse el JRE y el JDK.

En un sistema Linux, se descarga un fichero .tar.gz y se siguen los siguientes pasos desde línea de comandos:

1. Cambiar al directorio donde se quiere instalar:

```
$ cd nombre_ruta_acceso_directorio
```

2. Mover el fichero .tar.gz al directorio actual.
3. Desempaquetarlo e instalar Java:

```
$ tar zxvf jre-8u73-linux-x64.tar.gz
```

4. Borrar el fichero .tar.gz para liberar espacio en disco.

B.2. Instalación de Netbeans

Desde la página oficial de Netbeans se descarga el instalador para Windows. En este caso, se ha elegido la versión completa que tiene incluido el servidor Apache Tomcat que va a usarse.

En el caso de Linux, se descarga un script .sh que se ejecuta desde terminal con el siguiente comando:

```
$ sh netbeans-8.2-linux.sh
```

Tras su ejecución, se siguen los pasos indicados.

B.3. Base de datos con Apache Derby

Se crea la base de datos desde la ventana de *Services* en Netbeans. En la opción "Java DB" (figura B.1.) se crea una nueva base de datos y se le da un nombre.

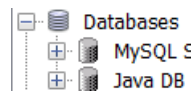


Figura B.1: Java DB.

Para crear la estructura de la base de datos se ejecuta el script creación.sql y para poblar la base de datos de test el de test.sql. Para ello, se conecta a la base de datos y sobre esta se elige la opción *execute command* con el botón derecho del ratón. Tras esto, aparece un editor donde se escriben las sentencias a ejecutar y se pulsa el botón que permite llevarlo a cabo.

En la figura B.2, se ven las opciones y la línea que indica la conexión.



Figura B.2: Opciones al ejecutar un comando.

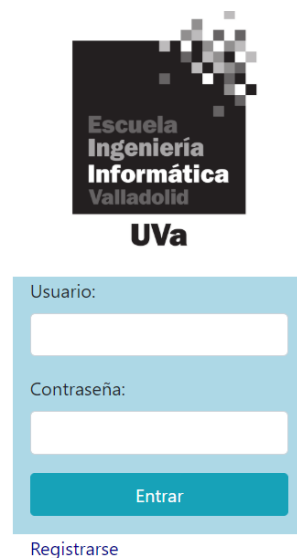
Tras crearla, es necesario cambiar el fichero context.xml de la aplicación con los datos de conexión a la base de datos:

```
<Resource auth="Container" driverClassName="org.apache.derby.jdbc.ClientDriver" logAbandoned="true" maxActive="100" maxIdle="30" maxWait="10000" name="jdbc/virtualizacion" password="" removeAbandoned="true" removeAbandonedTimeout="60" type="javax.sql.DataSource" url="jdbc:derby://localhost:1527/virtualizacion" username="root"/>
```

Apéndice C

Manual de uso

Al acceder a la aplicación web desde el navegador aparece la página de inicio de sesión (figura C.1.) desde la que se puede entrar con un usuario y contraseña previamente registrado.



Escuela
Ingeniería
Informática
Valladolid

UVa

Usuario:

Contraseña:


Entrar

[Registrarse](#)

Figura C.1: Página de inicio de sesión.

Desde la página de inicio de sesión se puede ir a la de registro (figura C.2.) para crear una cuenta. A su vez, desde esta última, se puede volver a la anterior. En el formulario de registro hay que tener en cuenta tres aspectos:

- El nombre de usuario es único.
- La contraseña tiene que tener más de cinco caracteres.
- Todos los campos son obligatorios.
- Si se intenta crear un usuario que ya existe, se muestra un mensaje al usuario indicándolo.



Escuela
Ingeniería
Informática
Valladolid

UVA

Usuario:

Email:

Nombre:

Apellidos:

Contraseña:

[¿Tienes ya una cuenta?](#)

Figura C.2: Página de registro.

Tras iniciar sesión en la aplicación, el usuario accede a la página principal, donde aparece un menú en la parte superior (figura C.3.) desde el que puede acceder a todas las opciones posibles:

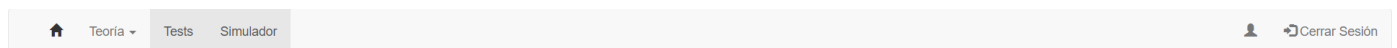


Figura C.3: Menú superior.

- **Icono de Home:** accede a la pantalla principal (figura C.4).
- **Teoría:** aparece un menú con las distintas páginas de teoría que puede ver el usuario (figura C.5 y figura C.6).
- **Tests:** desde esta pestaña el usuario puede realizar tests y ver su resultado tras darle al botón de enviar (figura C.7 y figura C.8). Para cada nivel, se muestran distintas preguntas aumentando la dificultad de éstas. Si la respuesta es correcta, se muestra en verde (figura C.9), si es incorrecta se muestra en rojo (figura C.10) y la correcta en verde y sino se contesta, se muestra un mensaje y la opción correcta en verde (figura C.11).
- **Simulador:** accede al simulador con el que puede interactuar (figura C.12).
- **Icono de perfil:** el usuario puede ver los datos de su cuenta, su nivel y el número de preguntas acertadas (figura C.13). Según este número, el usuario puede subir de nivel, al llegar a 35, al nivel 2 y al llegar a 70, al nivel 3.

- **Botón de cerrar sesión:** el usuario cierra su sesión y vuelve a la página de inicio.

Bienvenido, admin.

Ranking

Puesto	Nombre de usuario	Email	Nivel
1	dlecontem	dlecontem@sitemeter.com	3
2	jbuncel	jbuncel@wikipedia.org	3
3	hannandk	hannandk@netvibes.com	3
4	pbuckbeeg	pbuckbeeg@shutterfly.com	3
5	kcutilif	kcutilif@tumblr.com	3
6	trobice	trobice@google.ru	3
7	Imoomand	Imoomand@bigcartel.com	3
8	rlindborgc	rlindborgc@infoseek.co.jp	3
9	jpistolb	jpistolb@nyu.edu	3
10	awaywell9	awaywell9@psu.edu	3

Figura C.4: Página principal

Conceptos básicos

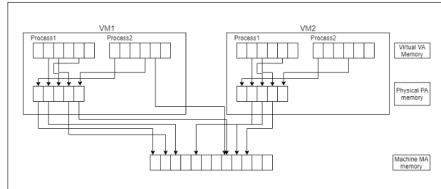
- GVA - Guest Virtual Address**
Dirección virtual del invitado generada por la CPU de la máquina virtual que necesita ser traducida.
- Guest CR3**
El registro CR3 controla el comportamiento cuando está el direccionamiento virtual activado y permite al procesador traducir a direcciones físicas mediante la localización del directorio de páginas y la tabla de páginas del proceso actual. En el caso del invitado tiene el mismo fin.
- GPA - Guest Physical Address**
Dirección física de la memoria física del sistema invitado.
- HPA - Host Physical Address**
Dirección física de la máquina host.
- EPT - Extended Page Table**
La tabla de páginas extendida es una tecnología desarrollada por Intel para la MMU.
- PT - Page Table**

Figura C.5: Página de teoría: conceptos básicos.

Virtualización de la memoria virtual

Mapeado a dos niveles en un entorno virtual

En un entorno virtual se realiza un mapeado a dos niveles: de memoria virtual a física y de memoria física a memoria de la máquina. Este proceso requiere compartir la memoria física del sistema y asignarla de forma dinámica a la memoria física de la máquina virtual. El sistema operativo invitado controla el mapeado de direcciones virtuales a físicas del entorno virtual, pero no puede acceder directamente a la memoria real. Para ello, se utiliza el VMM, que se encarga de mapear de memoria física del invitado a la real.



En la imagen, se muestra este proceso de mapeado. Hay dos máquinas virtuales, cada una con dos procesos en memoria virtual y estos se mapean a física de la máquina virtual. Después de esta memoria física, se mapea a la actual.

Proceso de traducción

En la máquina virtual, la CPU genera una dirección virtual, GVA, que será la que se vaya a traducir. Primero se localiza la tabla de páginas de último nivel, mediante el registro CR3. Este número de marco que lo indica es una dirección de tipo GPA que es necesario traducir a una HPA. Así, se obtiene una física de la máquina real. Este proceso de traducción lo realiza el VMM, primero utiliza el TLB para comprobar si ya está allí la traducción. Si no, tiene que acceder a las tablas EPT. La primera tabla, que se obtiene mediante el puntero EPT, tendrá escrito en la posición adecuada el número de marco que apunta a la tabla de siguiente nivel. Así, hasta que se completa la traducción de la

Figura C.6: Página de teoría: virtualización de memoria.

Aquel que se ejecuta sobre el sistema operativo invitado.

Aquel que se ejecuta sobre el hardware.

Pregunta 3

La función principal del registro CR3 es...

- Apuntar a la TP de Pi.
- Localizar la TP de primer nivel del proceso actual.
- Permitir la traducción de direcciones virtuales a físicas.

Pregunta 4

¿Cuándo se produce un fallo de página?

- Siempre que se produce una interrupción en el sistema al traducir direcciones.
- Nunca, puesto que siempre encuentra las direcciones que busca.
- Siempre que se produce una interrupción debida a que no se encuentra una dirección en la memoria.

Pregunta 5

¿Cuál es la función principal del TLB?

- Permitir cachear parte de la TP y acceder a la traducción de forma más rápida.
- Guardar puntero a la TP de un proceso.
- La misma que la de TP.

Enviar

Figura C.7: Página de tests.

The screenshot shows a web interface for a test. At the top, there are navigation tabs: 'Teoría', 'Tests', and 'Simulador'. A user icon and 'Cerrar Sesión' are in the top right. The main content area contains three question cards:

- Pregunta 1:** ¿Cuándo se produce un fallo de página?
Siempre que se produce una interrupción en el sistema al traducir direcciones.
Nunca, puesto que siempre encuentra las direcciones que busca.
Siempre que se produce una interrupción debida a que no se encuentra una dirección en la memoria.
- Pregunta 2:** ¿Qué es un hypervisor de tipo 1?
Aquel que se ejecuta sobre un sistema operativo.
Aquel que se ejecuta sobre el sistema operativo invitado.
Aquel que se ejecuta sobre el hardware.
- Pregunta 3:** La función principal del registro CR3 es...
Apuntar a la TP de Pi.
Localizar la TP de primer nivel del proceso actual.
Permitir la traducción de direcciones virtuales a físicas.

Figura C.8: Resultados de un test.

The screenshot shows a single question card:

Pregunta 1

¿Quién genera la dirección GVA?

La CPU en la máquina virtual.
No hay ninguna dirección GVA.
La CPU en la máquina host.

Figura C.9: Detalle de pregunta correcta.

The screenshot shows a single question card:

Pregunta 2

¿Cómo se obtiene la primera EPT a la que se accede?

Mediante la dirección de la memoria física.
Mediante la dirección GPA.
Mediante el puntero EPT.

Figura C.10: Detalle de pregunta incorrecta.

The screenshot shows a single question card:

Pregunta 3

No contestada.

¿Cuál de las siguientes afirmaciones sobre el TLB es falsa?

El TLB funciona como una caché.
Puede evitar tener que hacer dos niveles de traducción.
Es necesario usar el EPT aunque el TLB dé un acierto.

Figura C.11: Detalle de pregunta no contestada.



Figura C.12: Página del simulador.

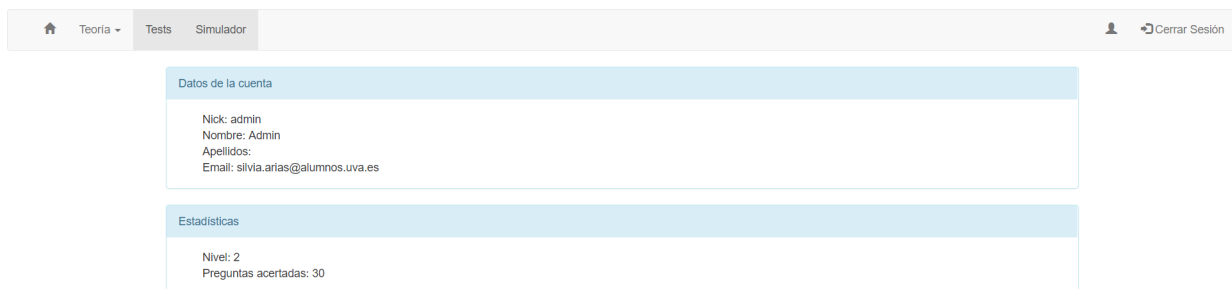


Figura C.13: Perfil de usuario.

Las funciones que puede realizar el usuario en el simulador son las siguientes:

- **Elegir la dirección que se va a traducir:** mediante un *dropdown*. Aparece al inicio de la simulación (figura C.14 y figura C.15).
- **Avanzar al siguiente paso:** mediante unas flechas que se pueden pulsar durante toda la simulación (figura C.16).
- **Retroceder al paso anterior:** mediante unas flechas que se pueden pulsar durante toda la simulación (figura C.16).
- **Reiniciar a valores iniciales:** mediante un botón (figura C.17).

Otros aspectos gráficos que hay que tener en cuenta:

- Nada más elegir la dirección y avanzar al siguiente paso, se muestra el proceso actual, la dirección a traducir y dónde está localizada en la memoria física del invitado, mediante una animación en color amarillo que dura unos segundos.
- Las celdas donde se escribe están señaladas por una flecha roja según se van sucediendo los pasos.
- Al quedarse sin espacio en la memoria física aparece un mensaje en rojo indicándolo y se desactivan las flechas para avanzar o retroceder.
- La celda de GPA cuando se va a traducir una dirección de este tipo muestra una animación en color rojo resaltándola. Cuando no se está traduciendo ninguna, aparece vacía.
- Cada tabla cuenta con etiquetas para identificarlas.



Figura C.14: Inicio del simulador.

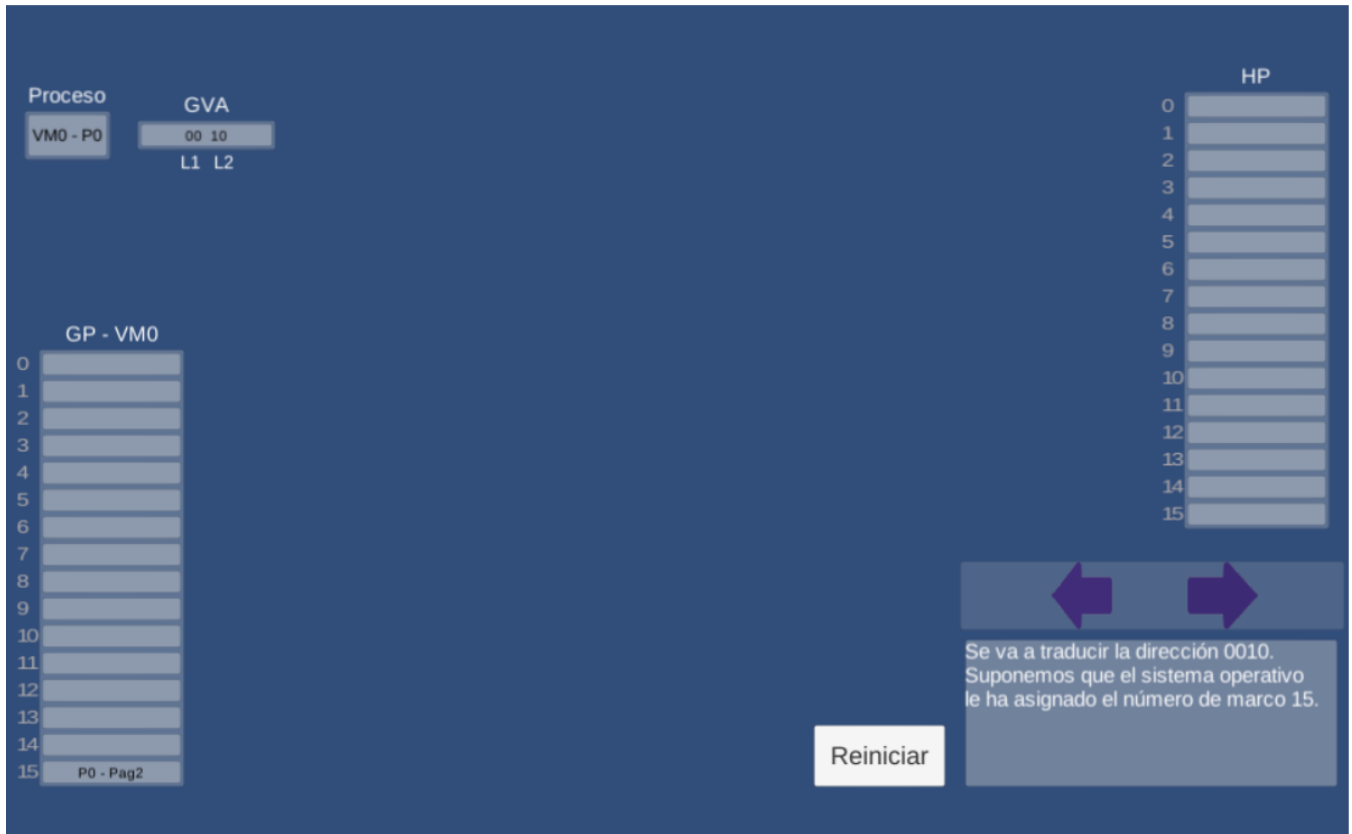


Figura C.15: Estado del simulador tras elegir la dirección.

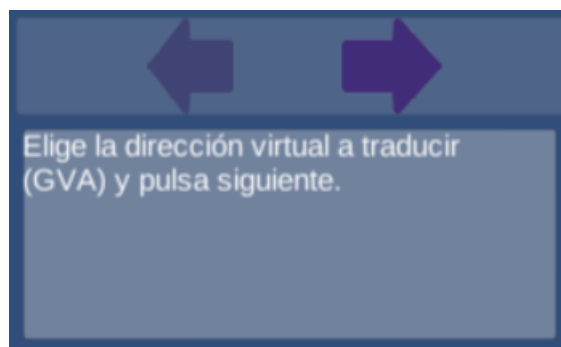


Figura C.16: Detalle de flechas para avanzar y retroceder.

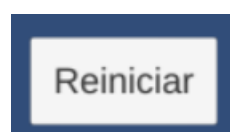


Figura C.17: Detalle de botón reiniciar

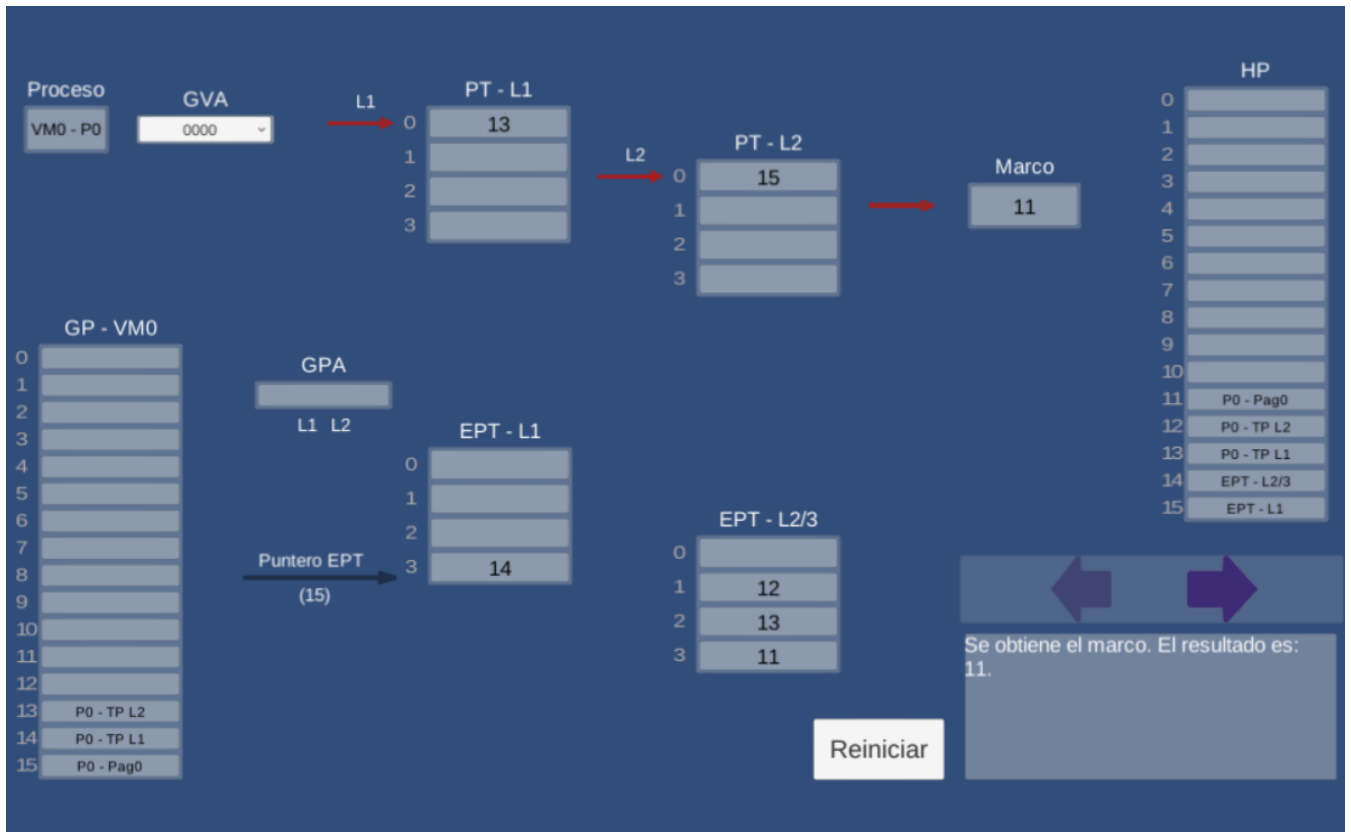


Figura C.18: Estado final del simulador.

Apéndice D

Preguntas de los tests

D.1. Nivel 1

1. ¿Qué es la MMU?
 - a) La memoria física del host.
 - b) Una unidad que tiene todas las referencias a memoria.
 - c) Una unidad de memoria.
2. ¿Qué permite la MMU?
 - a) Guardar referencias a direcciones físicas.
 - b) Llevar a cabo la traducción de direcciones virtuales a físicas a nivel hardware.
 - c) Obtener las direcciones virtuales.
3. ¿Es necesario una Tabla de Páginas para memoria virtual?
 - a) No, se tratan de estructuras físicas del host.
 - b) Sí.
 - c) Sí, pero no se usa igual que para la memoria física.
4. ¿Qué sucede si no está la página en memoria principal?
 - a) Se carga la página en memoria sin provocar ningún fallo.
 - b) Se produce un fallo de página.
 - c) Nada, el proceso continúa igual.
5. ¿Qué se entiende por sistema operativo del host?
 - a) El sistema operativo de la máquina virtual.
 - b) El sistema operativo sobre el que se ejecuta el hypervisor de la máquina virtual.

c) El sistema operativo virtual y host es el mismo.

6. ¿Qué se entiende por sistema operativo invitado?

a) El sistema operativo que se ejecuta en la máquina virtual.

b) No hay ninguna diferencia entre el sistema operativo virtual y sobre el que se está ejecutando.

c) El sistema operativo sobre el que se ejecuta el hypervisor de la máquina virtual.

7. ¿Pueden coexistir varios sistemas operativos en la misma máquina?

a) No, solo se puede tener un sistema operativo instalado a la vez.

b) Sí, es lo que permite la virtualización de máquinas.

c) No, hay que instalarlos cada uno en una partición del disco.

8. ¿Cuántos tipos de hypervisores hay?

a) El tipo 1.

b) El tipo 2.

c) El tipo 1 y 2.

9. ¿Qué componentes se pueden virtualizar?

a) CPU.

b) Memoria.

c) CPU, Memoria y dispositivos de E/S, entre otros.

10. ¿Cuál es la idea principal del concepto de máquina virtual?

a) Abstractar el hardware.

b) Abstractar el hardware de la computadora formando entornos de ejecución diferentes.

c) Crear una ilusión de entornos de ejecución.

11. ¿Cuáles de las siguientes opciones son beneficios de una máquina virtual?

a) Cada máquina virtual está aislada de las demás.

b) Es posible la compartición directa de recursos.

c) No es necesario probar los cambios en el sistema operativo.

12. ¿Cuántas tablas de páginas debe mantener el sistema operativo?

a) Una para cada proceso.

b) Una para todos los procesos.

c) Ninguna.

Preguntas de los tests

13. ¿Cuántas EPT debe mantener el sistema operativo?
- a) Una para cada proceso.
 - b) Una para todos los procesos.
 - c) Ninguna.
14. ¿Cuándo se produce un fallo de TLB?
- a) Cuando no se encuentra la página en la tabla de páginas.
 - b) Cuando no hay una entrada en el TLB para la página.
 - c) Cuando se encuentra la página en la tabla de páginas pero no en el TLB.
15. ¿Qué memoria se asigna a la máquina virtual?
- a) La misma que la máquina real.
 - b) Se asigna siempre un tamaño fijo.
 - c) La memoria física de la máquina se comparte con la de la máquina virtual.
16. ¿Cuántos niveles de mapeado son necesarios en un entorno virtual?
- a) Uno.
 - b) Tres.
 - c) Dos.
17. ¿Cuántos niveles de mapeado son necesarios en un entorno tradicional?
- a) Uno.
 - b) Tres.
 - c) Dos.
18. ¿Por qué tiene mapeado de dos niveles un entorno virtual?
- a) Porque es necesario mapear de virtual a física y de física a memoria máquina.
 - b) Solo necesita uno.
 - c) Porque es necesario mapear de virtual a memoria máquina.
19. ¿Quién es el encargado de mapear de memoria física del invitado a memoria máquina?
- a) TLB.
 - b) MMU.
 - c) VMM.
20. ¿Se puede evitar tener que hacer dos niveles de traducción?
- a) Sí, siempre con el TLB.
 - b) No, siempre hay que hacer dos.
 - c) Sí, algunas veces con el uso del TLB.

D.2. Nivel 2

1. La función principal del registro CR3 es. . .
 - a) Apuntar a la TP de Pi.
 - b) Localizar la TP de primer nivel del proceso actual.
 - c) Permitir la traducción de direcciones virtuales a físicas.
2. ¿Qué diferencia una GVA y una GPA?
 - a) Nada, son el mismo concepto.
 - b) GPA es la dirección física que se obtiene como traducción de GVA.
 - c) GVA es la dirección virtual del invitado y GPA la física.
3. ¿Cuál es la función principal del TLB?
 - a) Permitir cachear parte de la TP y acceder a la traducción de forma más rápida.
 - b) Guardar puntero a la TP de un proceso.
 - c) La misma que la de TP.
4. ¿Cuándo se produce un fallo de página?
 - a) Siempre que se produce una interrupción en el sistema al traducir direcciones.
 - b) Nunca, puesto que siempre encuentra las direcciones que busca.
 - c) Siempre que se produce una interrupción debida a que no se encuentra una dirección en la memoria.
5. ¿Qué es un hypervisor de tipo 1?
 - a) Aquel que se ejecuta sobre un sistema operativo.
 - b) Aquel que se ejecuta sobre el sistema operativo invitado.
 - c) Aquel que se ejecuta sobre el hardware.
6. ¿Qué es un hypervisor de tipo 2?
 - a) Aquel que se ejecuta sobre un sistema operativo.
 - b) Aquel que se ejecuta sobre el sistema operativo invitado.
 - c) Aquel que se ejecuta sobre el hardware.
7. ¿Es posible tener un sistema invitado distinto del sistema host utilizando virtualización?
 - a) Sí.
 - b) No.

Preguntas de los tests

- c) Depende de los recursos hardware.
8. ¿Dónde se genera la dirección GVA?
- a) En la máquina host.
 - b) En la máquina virtual.
 - c) No hay ninguna dirección GVA.
9. ¿Quién genera la dirección GVA?
- a) La CPU en la máquina virtual.
 - b) No hay ninguna dirección GVA.
 - c) La CPU en la máquina host.
10. ¿Cómo se obtiene la primera EPT a la que se accede?
- a) Mediante la dirección de la memoria física.
 - b) Mediante la dirección GPA.
 - c) Mediante el puntero EPT.
11. ¿Cuál de las siguientes afirmaciones sobre el TLB es falsa?
- a) El TLB funciona como una caché.
 - b) Puede evitar tener que hacer dos niveles de traducción.
 - c) Es necesario usar el EPT aunque el TLB dé un acierto.
12. ¿Qué sucede cuando se produce un fallo de TLB?
- a) Que se usa el EPT para obtener el marco.
 - b) Que no se puede traducir la dirección.
 - c) Se vuelve a consultar al TLB.
13. ¿Qué tipo de direcciones se pueden dar en el entorno virtual?
- a) GVA, GPA.
 - b) GVA, HPA.
 - c) HPA, GPA.
14. ¿Qué se necesita para poder localizar a la tabla de páginas de primer nivel del proceso?
- a) TLB.
 - b) El puntero EPT.
 - c) El registro CR3.
15. ¿Al traducir una dirección de tipo GPA qué se obtiene?
- a) Otra dirección GPA.

- b) Una dirección HPA.
 - c) Una dirección GVA.
16. ¿Qué se obtiene como resultado de traducir una GVA?
- a) Un marco de la memoria física de la máquina real.
 - b) Una dirección GPA.
 - c) Un marco de la memoria física de la máquina virtual.
17. ¿Qué sucede si se produce un acierto de TLB?
- a) Que se obtiene directamente la traducción.
 - b) Que se utiliza el EPT pero es más rápido.
 - c) No hay diferencia.
18. ¿Qué permite un hypervisor?
- a) Traducir direcciones GVA.
 - b) Utilizar diferentes técnicas para varios sistemas operativos.
 - c) Depende si es de tipo 1 o de tipo 2.
19. ¿Cuál no es una funcionalidad del VMM?
- a) Crear, parar e iniciar máquinas virtuales.
 - b) Controlar cada máquina virtual.
 - c) Traducir direcciones GVA.
20. ¿Cuál de las siguientes afirmaciones es cierta?
- a) El VMM permite crear y ejecutar máquinas virtuales.
 - b) Una dirección GVA es lo mismo que una GPA.
 - c) Solo hay un tipo de VMM.

D.3. Nivel 3

1. ¿Qué permite el registro CR3?
- a) Apuntar a la TP de Pi.
 - b) Obtener la tabla EPT de nivel 1.
 - c) Localizar la TP de primer nivel del proceso actual.
2. ¿Qué diferencia una HPA y una GPA?
- a) Nada, son el mismo concepto.

Preguntas de los tests

- b) GPA es la dirección física que se obtiene como traducción de HPA.
c) GPA es la dirección física del invitado y HPA la física del host.
3. ¿Cuál no es una función del TLB?
- a) Permitir cachear parte de la TP y acceder a la traducción de forma más rápida.
b) Hacer el proceso de traducción más rápido.
c) Mapear direcciones GVA a HPA.
4. ¿Cuándo no se produce un fallo de página?
- a) Siempre que se produce una interrupción en el sistema al traducir direcciones.
b) Cuando se encuentra la dirección en la memoria.
c) Siempre que se produce una interrupción debida a que no se encuentra una dirección en la memoria.
5. ¿Qué es el espacio de intercambio o swap?
- a) Es una zona del disco que se usa para intercambiar procesos que no han de mantenerse en la memoria física.
b) Es una zona de la memoria donde se intercambian procesos.
c) Es una zona del disco que se usa para intercambiar procesos que no han de mantenerse en la memoria virtual.
6. ¿Cuándo es necesario utilizar la zona de swap?
- a) Siempre que haya procesos en memoria física.
b) Siempre que no haya espacio para más procesos en la memoria física.
c) No se puede utilizar con la memoria física de la máquina virtual.
7. ¿Cuál es el tamaño mínimo para la memoria física si la GVA es de cuatro bits?
- a) 4.
b) 16.
c) 32.
8. ¿Cuántas EPT de nivel dos habrá si la dirección GPA a traducir tiene cuatro bits?
- a) 2.
b) 4.
c) 8.
9. Si la dirección GPA es 0110, ¿en qué posición del EPT de nivel dos se escribirá?
- a) En la 1.

b) En la 0.

c) En la 2.

10. Si la dirección GVA es 1011, ¿en qué posición de la tabla de páginas de nivel dos escribirá?

a) En la 1.

b) En la 2.

c) En la 3.

11. Si la dirección GVA es 1100, ¿en qué posición del EPT de nivel uno escribirá?

a) En la 0.

b) En la 3.

c) En ninguna.

12. Si la dirección HPA es 0111, ¿en qué posición de la tabla de páginas de nivel dos escribirá?

a) En ninguna.

b) En la 1.

c) En la 3.

13. ¿De cuántos bits es la dirección GVA si la memoria física tiene 64 direcciones?

a) 4 bits.

b) 8 bits.

c) 6 bits.

14. ¿De cuántos bits es la dirección GVA si la memoria física tiene 256 direcciones?

a) 4 bits.

b) 8 bits.

c) 12 bits.

15. ¿Cuántas EPT de nivel 2 habrá si la dirección GPA es de 6 bits?

a) 4 tablas.

b) 16 tablas.

c) 8 tablas.

16. Si una dirección GPA es 1101, ¿Cuántos bits representan a L2?

a) 6.

b) 2.

c) 3.

Preguntas de los tests

17. Si una dirección GVA es 0110, ¿qué valor tiene L1?
- a) 01.
 - b) 11.
 - c) 10.
18. ¿Qué se obtiene como resultado de traducir una dirección virtual GVA?
- a) Una página.
 - b) Un marco.
 - c) Una dirección.
19. ¿Qué supondría añadir un TLB al simulador?
- a) El proceso sería el mismo.
 - b) La parte del EPT no haría falta.
 - c) Se añadiría un paso para comprobar el TLB y, en caso de acierto, no sería necesario usar la EPT.
20. ¿Qué se escribe en el puntero CR3?
- a) Una dirección que apunta a la tabla de páginas de primer nivel.
 - b) Un marco donde se localiza el puntero a la tabla de páginas de primer nivel.
 - c) Una página donde se localiza el puntero a la tabla de páginas de primer nivel.

Apéndice E

Contenido del CD

El contenido del CD está distribuido de la siguiente forma:

- **memoria.pdf:** archivo con la memoria del trabajo.
- **fuentes.zip:** código fuente de la aplicación.
 - **Sim:** contiene el código desarrollado con Unity.
 - **TFG:** contiene el código de la aplicación web.
 - **Base de datos:** contiene los scripts creacion.sql y test.sql.

Bibliografía

- [1] K. Hwang, J. Dongarra, and G. Fox, *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*. Morgan Kaufmann, 2011.
- [2] W. Stallings, *Sistemas operativos: aspectos internos y principios de diseño*. Pearson Prentice-Hall, 2005.
- [3] A. Silberschatz, *Fundamentos de sistemas operativos*. McGraw Hill, 2006.
- [4] Software de simulación en la enseñanza. [Online]. Available: <https://macyte.wordpress.com/2011/05/15/software-de-simulacion-en-la-ensenanza/>
- [5] Simulador de memoria virtual. [Online]. Available: <https://github.com/Ricardo96r/Simulador-de-memoria-virtual/blob/master/README.md>
- [6] Simulador de jerarquías de memoria. [Online]. Available: https://www.researchgate.net/publication/228522509_SJM_Un_simulador_de_jerarquias_de_memoria_orientado_a_la_docencia_de_arquitectura_de_computadores
- [7] Beneficios de la gamificación en la formación. [Online]. Available: <https://www.nubemia.com/beneficios-de-la-gamificacion-en-la-formacion/>
- [8] Gamification in education: Advantages and disadvantages. [Online]. Available: <https://gamification101blog.wordpress.com/2016/04/28/advantages-and-disadvantages/>
- [9] Gamification in education: What, how, why bother? [Online]. Available: http://www.academia.edu/570970/Gamification_in_Education_What_How_Why_Bother
- [10] Juego serio. [Online]. Available: https://es.wikipedia.org/wiki/Juego_serio
- [11] ¿qué son los “serious games“ (juegos serios)? [Online]. Available: <http://www.exelweiss.com/blog/356/serious-games-juegos-serios/>
- [12] Gamificar: El uso de los elementos del juego en la enseñanza de español. [Online]. Available: https://cvc.cervantes.es/ensenanza/biblioteca_ele/aepe/pdf/congreso_50/congreso_50_09.pdf
- [13] P. de la Fuente Redondo, *Apuntes de la asignatura “Planificación y Gestión de Plataformas Informáticas“*, 2017.
- [14] Burndown generator. [Online]. Available: <http://www.burndowngenerator.com/>

-
- [15] R. S. Pressman, *Ingeniería del Software: un enfoque práctico*. Mc Graw Hill, 2010.
- [16] Astah professional. [Online]. Available: <http://astah.net/>
- [17] J. M. M. Corral, *Apuntes de la asignatura "Diseño, Integración y Adaptación del Software"*, 2018.
- [18] Unity. [Online]. Available: <https://unity3d.com/es>
- [19] Manual de unity. [Online]. Available: <https://docs.unity3d.com/es/current/Manual/index.html>
- [20] C#. [Online]. Available: https://es.wikipedia.org/wiki/C_Sharp
- [21] Visual studio. [Online]. Available: <https://www.visualstudio.com/downloads/>
- [22] Microsoft visual studio. [Online]. Available: https://es.wikipedia.org/wiki/Microsoft_Visual_Studio
- [23] Netbeans. [Online]. Available: <https://netbeans.org/>
- [24] Netbeans. [Online]. Available: <https://es.wikipedia.org/wiki/NetBeans>
- [25] Html. [Online]. Available: <https://es.wikipedia.org/wiki/HTML>
- [26] Html tutorial. [Online]. Available: <https://www.w3schools.com/html/>
- [27] Bootstrap tutorial. [Online]. Available: <https://www.w3schools.com/bootstrap/>
- [28] Hoja de estilos en cascada. [Online]. Available: https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada
- [29] Bootstrap. [Online]. Available: https://es.wikipedia.org/wiki/Bootstrap_%28framework%29
- [30] Java servlet. [Online]. Available: https://es.wikipedia.org/wiki/Java_Servlet
- [31] Java. [Online]. Available: https://www.java.com/es/download/help/linux_x64_install.xml
- [32] Apache tomcat. [Online]. Available: <http://tomcat.apache.org/>
- [33] Apache derby. [Online]. Available: <http://db.apache.org/derby/>
- [34] Java database connectivity. [Online]. Available: https://es.wikipedia.org/wiki/Java_Database_Connectivity

