

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 1



SAATI. Sistema informático de apoyo al aprendizaje de la asignatura
Teleinformática.



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

Autor: Rafael Angel Frances León

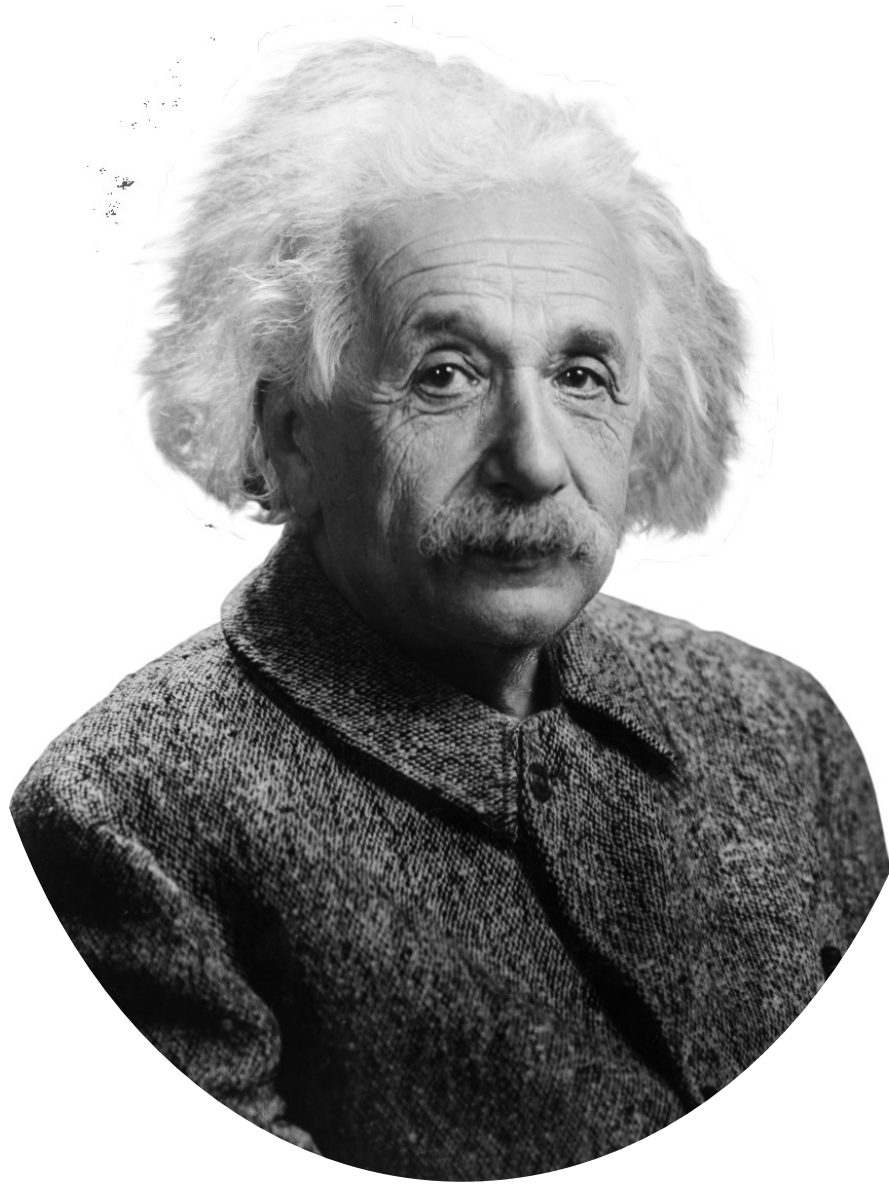
Tutor:

Ing. Dargel Veloz Morales

Cotutor:

Ing. Osmel Mojena Dubet

La Habana, junio de 2019



“Si buscas resultados distintos, no hagas siempre lo mismo”

Albert Einstein.

Dedicatoria

Especialmente a mis padres, quienes han velado por mi formación día a día y son mi motivación fundamental,

A mi hermano, quien me ha apoyado en todo momento a pesar de la distancia,

A mi familia en general, quienes esperan con ansias este triunfo,

A mis amigos de la universidad,

A los de fuera,

A todos los que han hecho posible que hoy me gradúe como ingeniero en ciencias informáticas,

A ustedes, con todo mi corazón.

Rafael Angel Frances León.

Declaración jurada de autoría

Declaro por este medio que yo **Rafael Angel Frances León**, con carné de identidad **95053040909** soy el autor principal del trabajo titulado “**Sistema informático de apoyo al aprendizaje de la asignatura Teleinformática**” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firmamos la presente a los _____ días del mes de _____ de _____

Rafael Angel Frances León

Autor

Ing. Dargel Veloz Morales

Tutor

Ing. Osmel Mojena Dubet

Cotutor

Resumen

En la actualidad la incorporación de las Tecnologías de la Información y las Comunicaciones (TIC) al proceso de enseñanza-aprendizaje ha servido de apoyo en la formación de estudiantes en todos los niveles. La Universidad de las Ciencias Informáticas es un centro de estudios superiores caracterizado por el uso de las tecnologías del aprendizaje en el proceso docente-educativo. Como parte del plan de estudio de la carrera Ingeniería en Ciencias Informáticas, se imparte la asignatura Teleinformática, la cual presenta resultados que pueden ser mejorados. Dentro de los recursos que contribuyen a ello se encuentra la motivación. La presente investigación presenta un sistema informático de apoyo al aprendizaje de la asignatura Teleinformática, el cual permite resolver ejercicios aplicando técnicas de gamificación. El funcionamiento de esta aplicación se basa en tecnología *e-learning*, al permitir la retroalimentación de los usuarios a través del sistema. Para lograrlo se analizaron las herramientas informáticas con características similares a la solución propuesta, asimismo se hizo un estudio de los métodos y tecnologías más factibles a utilizar para lograr un correcto diseño del sistema, a través del análisis de las metodologías existentes que rigen el desarrollo de software. Se presenta una estrategia de pruebas realizada al sistema, con el objetivo de entregar al cliente una solución confiable. Como resultado se obtuvo SAATI, una herramienta de apoyo a la asignatura Teleinformática.

Palabras clave: e-learning, gamificación, Proceso de enseñanza-aprendizaje, software, Teleinformática.

Índice de contenido

Introducción	1
CAPÍTULO 1. Fundamentos teóricos de la investigación	6
1.1. Introducción	6
1.2. Conceptos asociados al dominio del problema	6
1.3. Análisis de sistemas asociados a la enseñanza y el aprendizaje	9
1.4. Entorno de desarrollo.....	12
1.4.1. Metodologías de desarrollo de software	12
1.4.2. Modelado de software.....	12
1.4.3. Lenguajes y herramientas de programación	13
1.4.4. Servidor Web.....	16
1.5. Herramienta de Control de Versiones	17
1.6. Conclusiones del capítulo	17
CAPÍTULO 2. Análisis y diseño de la herramienta	18
2.1. Modelado del negocio.....	18
2.1.1. Modelo conceptual.....	18
2.1.2. Características de la propuesta de solución.....	19
2.2. Especificación de requisitos.....	20
2.3. Historias de usuario	22
2.4. Patrón arquitectónico	24
2.5. Patrones de diseño	25
2.6. Modelo de datos	28
2.7. Diagrama de Clases del diseño	29

2.8	Conclusiones del capítulo	31
CAPÍTULO 3. Implementación y pruebas de la propuesta de solución		32
3.1	Introducción	32
3.2	Estándares de codificación	32
3.3	Diagrama de componentes	36
3.4	Diagrama de despliegue	39
3.5	Interfaz gráfica de usuario	39
3.6	Pruebas de software	42
3.6.1	Pruebas de aceptación	42
3.6.2	Pruebas de Carga y Estrés	46
3.7	Conclusiones del capítulo	48
CONCLUSIONES GENERALES		50
RECOMENDACIONES		51
REFERENCIAS		52
ANEXOS.....		56

Índice de tablas

Tabla 1: Requisitos Funcionales (Elaboración propia)	20
Tabla 2: Plantilla para las historias de usuario (Elaboración propia)	22
Tabla 3: Historia de Usuario Registrar usuario (Elaboración propia)	23
Tabla 4: Historia de Usuario Evaluar respuesta (Elaboración propia).....	23
Tabla 5: Estereotipos web para UML (Elaboración propia)	29
Tabla 6: Estándares de codificación (Elaboración propia).....	32
Tabla 7: Descripción de los ficheros que componen los componentes (Elaboración propia).....	37
Tabla 8: Cuadro Lógico de Iadov (Elaboración Propia)	42
Tabla 9: Índices para el cálculo de satisfacción grupal (efdeportes, 2002).....	44
Tabla 10: Resultados de las pruebas de carga y estrés (Elaboración propia)	48
Tabla 11: Historia de usuario Registrar usuario (Elaboración propia).....	56
Tabla 12: Historia de usuario Autenticar usuario (Elaboración propia)	56
Tabla 13: Historia de usuario Mostrar mundos (Elaboración propia)	56
Tabla 14: Historia de usuario Mostrar niveles (Elaboración propia).....	57
Tabla 15: Historia de usuario Mostrar preguntas (Elaboración propia)	57
Tabla 16: Historia de usuario Mostrar tabla de posiciones global (Elaboración propia)	58
Tabla 17: Historia de usuario Mostrar tabla de posiciones del mundo (Elaboración propia)	58
Tabla 18: Historia de usuario Mostrar tabla de posiciones de nivel (Elaboración propia)	58
Tabla 19: Historia de usuario Evaluar respuesta (Elaboración propia)	59
Tabla 20: Historia de usuario Mostrar logros de nivel.....	59
Tabla 21: Historia de usuario Mostrar puntuación acumulada	59
Tabla 22: Historia de usuario Mostrar créditos acumulados (Elaboración propia)	60

Tabla 23: Historia de usuario Mostrar comodines disponibles (Elaboración propia).....	60
Tabla 24: Historia de usuario Utilizar comodín (Elaboración propia)	61
Tabla 25: Historia de usuario Activar audio de fondo (Elaboración propia)	61
Tabla 26: Historia de usuario Desactivar audio de fondo (Elaboración propia).....	61
Tabla 27: Reporte de evaluación final Facultad 1. 2016-2017.....	64
Tabla 28: Reporte de evaluación final 2017-2018 (Elaboración propia)	64

Índice de figuras

Ilustración 1: Modelo del Dominio (Elaboración propia).....	19
Ilustración 2: Flujo de trabajo del patrón arquitectónico Modelo-Vista-Plantilla (Python-Django. Framework de desarrollo web para perfeccionistas basado en el modelo MTV., 2012).....	24
Ilustración 3: Estructura de directorios del sistema (Elaboración propia).....	25
Ilustración 4: Código fuente para mostrar los mundos (Elaboración propia).....	26
Ilustración 5: Decoradores para el control de sesiones y permisos (Elaboración propia)	28
Ilustración 6: Modelo de Datos (Elaboración propia).	28
Ilustración 7:Diagrama de clases del diseño de la HU Mostrar Pregunta (Elaboración propia).....	31
Ilustración 8: Diagrama de Componentes (Elaboración propia).....	37
Ilustración 9: Representación del Diagrama de Despliegue (Elaboración propia).....	39
Ilustración 10: Interfaz principal del sistema (Elaboración propia)	40
Ilustración 11: Interfaz de selección mundos (Elaboración propia).....	40
Ilustración 12: Interfaz de selección del nivel (Elaboración propia)	41
Ilustración 13: Interfaz de realización de ejercicios (Elaboración propia).....	41

Introducción

En la denominada enseñanza tradicional los conocimientos son transmitidos de manera conceptualizada bajo un sistema de comunicación unidireccional, en donde un cúmulo de contenidos de información es suministrado a los alumnos para lograr su aprendizaje. En dicho proceso de enseñanza tradicional juegan un papel preponderante las habilidades verbales del maestro y los recursos bibliográficos que emplea para la comprensión por parte de los educandos (Navarro, 2009).

La enseñanza innovadora plantea un cambio significativo en la perspectiva mundial, respecto a la cultura, las formas de acceso al conocimiento, a la interpretación de la realidad, y, por ende, en la forma de aprender y educar. En la búsqueda de la enseñanza innovadora surge el paradigma de la educación multimedia, el cual constituye un área de conocimiento y una ciencia aplicada que influye directamente en la evolución y desarrollo del proceso de enseñanza-aprendizaje (Navarro, 2009), en lo adelante (PEA). Esto se ha venido aplicando apoyado con el avance de las tecnologías, de ahí que la creación de modelos de enseñanza basados en la utilización de las Tecnologías de la Información y las Comunicaciones (TIC), ocupa un lugar importante en los procesos de enseñanza actuales.

Muchos docentes ven en las TIC un medio que les puede facilitar su labor, otros consideran que por ser algo con lo que los jóvenes conviven diariamente, el PEA mejorará; sin embargo, conforme van implementando cursos en los que las incorporan, se van dando cuenta que las TIC no son solo un medio de transmisión de conocimientos, sino que pueden convertirse en herramientas valiosas, que promueven ambientes de aprendizaje colaborativos, donde el docente deja de ser el centro del proceso para convertirse en un mediador de los temas que se traten en un curso (Delgado Fernández, y otros, 2009).

Hoy en día es muy fácil que un estudiante tenga acceso a espacios electrónicos de información ordenada, distribuida, común y compartida, para que el aprendizaje se lleve a un plano en donde el estudiante mediante el acompañamiento de una herramienta de apoyo pueda adquirir y adueñarse de este, logrando de esta manera alcanzar un aprendizaje significativo y que a su vez puedan aprovechar las ventajas que ofrecen los computadores como recursos didácticos.

Uno de estos espacios son los Entornos Virtuales de Enseñanza y Aprendizaje (EVEA), conocido en inglés como *Virtual Learning Environment (VLE)*, que puede ser definido como un sistema de software diseñado para facilitar la gestión de cursos, sean completamente a distancia o como complemento de cursos

INTRODUCCIÓN

presenciales (Moreno Cadavid, y otros, 2015). Con respecto a las características particulares que debe poseer un EVEA, las mismas varían según el autor que se consulte.

Según Stojanovic (2008), un EVEA debe permitir el acceso a materiales de estudio y a fondos de recursos, así como el enlace entre dichos materiales, entre sí y con información ubicada en la Web; ofrecer, mediante diversas actividades y tareas, posibilidades de autorregulación del aprendizaje; ser un espacio social y un marco para el comportamiento interactivo; permitir que los alumnos sean productores de información, proporcionando una experiencia más rica en el aprendizaje individual e integrar múltiples herramientas. Entre ellas se puede mencionar foros de discusión, mensajería, calendario, entre otras.

Estos entornos han sufrido una rápida evolución en los últimos años, fundamentalmente aquellos especializados en el sector universitario. En la educación superior, estos sistemas presentan grandes oportunidades tanto para los docentes como, y, sobre todo, para los estudiantes en términos de accesibilidad y de flexibilidad.

Por tanto, ser un mediador en entornos virtuales, no significa cambiar el espacio de un aula tradicional a un aula virtual, cambiar los libros por documentos electrónicos, las discusiones en clase por foros virtuales o las horas de atención a estudiantes por encuentros en chat o foros de conversación. Significa encontrar nuevas estrategias que nos permitan mantener activos a nuestros estudiantes aún cuando estos se encuentren en distintas partes del mundo, promoviendo la construcción de conocimientos y la colaboración (Delgado Fernández, y otros, 2009).

Esta tendencia obliga a buscar nuevas alternativas que contribuyan a mejorar el proceso de enseñanza haciéndolo más atractivo para los estudiantes. Por lo que estos avances en las ciencias de la educación de conjunto con las transformaciones tecnológicas y el acceso a las redes, han provocado que la educación a distancia vaya en incremento.

En Cuba la educación a distancia tuvo sus comienzos a partir de 1971 bajo la modalidad de enseñanza dirigida. Actualmente existen varios EVEA o plataformas *e-learning*, principalmente en varios centros de Educación Superior. Una plataforma e-learning es una plataforma tecnológica *LMS (Learning Management System)* que facilita la formación de personas a través de internet. Gracias a ella podemos gestionar cursos por Internet y ayudar a los educadores a crear comunidades de aprendizaje en línea. El Centro Nacional de Educación a Distancia, con sede en la Universidad de las Ciencias Informáticas (UCI), contribuye al

INTRODUCCIÓN

desarrollo y la excelencia de la educación a distancia y semipresencial en Cuba, incrementando su competitividad a través de la difusión, la mejora continua y la aplicación creadora de las TIC.

A pesar de estas investigaciones realizadas, aún no se obtienen los resultados deseados sobre la formación en los EVEA, puesto que los estudiantes universitarios aún no tienen una participación real y significativa en el proceso de formación profesional. Se debe tomar en cuenta que tanto docentes como estudiantes deben estar precisamente capacitados para utilizar este tipo de educación, porque no basta con tener un material que supere las expectativas de todos si no se sabe cómo utilizarlos (Guaña-Moya, y otros, 2015). De lo contrario, esto constituye un desaprovechamiento de las ventajas que brindan las TIC en el PEA, en cuanto a la utilización, por ejemplo, de técnicas de gamificación como parte de los elementos que pudieran ayudar en la asimilación de los contenidos que presentan un nivel mayor de dificultad.

La UCI cuenta con un EVEA, la plataforma para la gestión del aprendizaje Xauce ZERA v2.1, desarrollada por el Centro de Tecnologías para la Formación (FORTES). Esta plataforma integra las principales especificaciones y estándares educativos, desarrollados y utilizados a nivel mundial en plataformas de aprendizaje colaborativo e incluye cursos de todas las asignaturas de la carrera Ingeniería en Ciencias Informáticas, pero también presenta las deficiencias antes mencionadas.

La Teleinformática en específico, al ser una asignatura técnica, hace que sea más difícil para los estudiantes poder acceder e incorporar el conocimiento. Carece de mecanismos propios que logren motivar a los estudiantes a aprender de manera autónoma. Realizando un análisis de los resultados finales obtenidos en la asignatura, en los cursos 2016-2017 y 2017-2018 (Ver Anexo 3), se obtuvo un 67,92% y un 78,07% de aprobados respectivamente en la Facultad 1, con una calidad de nota de 33,96% y 25,44% respectivamente, los cuales no son muy favorables, pero pueden ser mejorables.

Por otra parte, los estudiantes hoy en día cuentan en su mayoría con medios tecnológicos, dígase ordenadores personales o teléfonos inteligentes, los cuales no se utilizan como es debido en las aulas, para mejorar la calidad de las clases e incluso facilitar el acceso a información en tiempo real, lo que aleja al estudiante de Ingeniería en Ciencias Informáticas del objeto de su profesión. La interacción estudiante-profesor, en el PEA, se centra en actividades presenciales y no se aprovecha de esta manera un aprendizaje ubicuo. Los medios de enseñanza existentes en Teleinformática no incorporan recursos digitales, o simplemente carecen del atractivo adecuado, para lograr motivación al estudio, entre los estudiantes.

INTRODUCCIÓN

Además, se evidencia un bajo aprovechamiento de las variantes competitivas existentes que propician un mayor rendimiento evaluativo en el PEA.

En correspondencia con lo antes expuesto se plantea como **problema de investigación**: ¿Cómo contribuir a través de un sistema informático al proceso de enseñanza-aprendizaje de la asignatura Teleinformática?

Se identifica como **objeto de estudio** de la investigación: procesos de desarrollo de software.

Se define como **campo de acción**: el proceso de desarrollo de software de apoyo a la enseñanza-aprendizaje.

Para darle solución al problema anteriormente planteado, se define como **objetivo general**: Desarrollar un sistema informático que contribuya al proceso de enseñanza-aprendizaje de la asignatura Teleinformática.

Para darle cumplimiento al objetivo planteado se responderán las siguientes **preguntas científicas**:

1. ¿Cuáles son los supuestos teóricos que sustentan el desarrollo y la utilización de SAATI?
2. ¿Qué aspectos deben tenerse en cuenta para realizar el análisis y diseño de SAATI?
3. ¿Cómo implementar SAATI, a partir del análisis y diseño realizado?
4. ¿Qué resultados se obtendrán al validar el desempeño de SAATI?

Acorde a las preguntas científicas ya expuestas, se han trazado los siguientes **objetivos específicos**:

1. Identificar el marco teórico conceptual con respecto a los componentes motivacionales, las herramientas y tecnologías asociadas.
2. Diseñar SAATI.
3. Implementar SAATI.
4. Validar el correcto funcionamiento de la aplicación implementada.

Para el desarrollo de la investigación se utilizaron los siguientes métodos científicos:

Métodos Teóricos:

- ❖ **Observación:** se utilizó para constatar las deficiencias existentes en herramientas para el estudio de Teleinformática utilizadas en la UCI, así como investigar sobre el funcionamiento, ventajas y desventajas de otras aplicaciones educativas similares.
- ❖ **Analítico-Sintético:** se empleó para examinar la bibliografía correspondiente a los contenidos de la asignatura Teleinformática y para analizar los resultados de la asignatura en cursos anteriores, con el objetivo de utilizar la información para el desarrollo de SAATI.
- ❖ **Modelación:** se utilizó para la modelación de las funcionalidades de la aplicación. Permitió la abstracción de la realidad al modelo informático y de ahí buscar la manera de implementarlo, concretando las características de cada uno de los contenidos seleccionados.

Métodos Empíricos:

- ❖ **Encuesta:** se utilizó para conocer el nivel de satisfacción de una muestra de estudiantes, sobre la propuesta de solución.

El presente documento está estructurado en tres capítulos:

Capítulo 1: Fundamentos teóricos de la investigación: este capítulo contiene la base teórica para entender el problema planteado, se abordan conceptos fundamentales para entender la investigación. Se analiza el estado del arte de los sistemas asociados al PEA existentes. Se realiza una valoración de la importancia de la Teleinformática para la formación de futuros profesionales de las ciencias informáticas. Se argumenta el uso de las TIC en la educación utilizando técnicas de gamificación como recurso de apoyo al PEA. Finalmente se caracterizan las herramientas, lenguajes y metodologías estudiadas para la implementación.

Capítulo 2: Análisis y diseño de la herramienta: en este capítulo se presentan las fases de planificación y diseño definidas por la metodología Proceso Unificado Ágil (*AUP*, por sus siglas en inglés) variación UCI. Se realiza la propuesta de solución y se planifica el proceso de desarrollo de software. Se definen, entre otros, las historias de usuarios, los requisitos funcionales y no funcionales y la arquitectura del sistema.

Capítulo 3: Implementación y pruebas de la propuesta de solución: se definen los estándares de codificación a utilizar y se realiza el diagrama de componentes y de despliegue de la solución. Además, se realizan pruebas para verificar la calidad de SAATI, se muestran los resultados y las interfaces principales.

CAPÍTULO 1. Fundamentos teóricos de la investigación

1.1. Introducción

En el presente capítulo se establecen conceptos fundamentales que sustentan la investigación. Se realiza un análisis del estado del arte de las herramientas educativas que presentan características similares a la aplicación que será desarrollada. Se realiza una valoración de la importancia del estudio de Teleinformática dentro de la Informática, en la formación de futuros profesionales de las TIC, se argumenta sobre el uso de las TIC en el PEA, y se enfatiza en el uso de elementos de gamificación en el PEA. Finalmente se caracterizan las herramientas, lenguajes y metodologías consideradas para dar solución al problema de investigación.

1.2. Conceptos asociados al dominio del problema

Para ayudar a entender el desarrollo de la investigación, se relacionarán a continuación los conceptos principales que sirven de soporte.

Un **sistema informático** según Alegsa “es un conjunto de partes o recursos formados por el hardware parte tangible, software parte lógica de un computador y las personas que lo emplean, que se relacionan entre sí para almacenar y procesar información con un objetivo en común” (Alegsa, 2019).

Uno de los sistemas informáticos más comúnmente utilizados hoy en día son las **aplicaciones web**, las cuales se definen como un sistema de hipermedia¹ donde los recursos se encuentran vinculados unos a otros, por lo que debe verse como un sistema de nodos interconectados a través de vínculos. Estos vínculos proporcionan la forma para navegar entre los recursos de la aplicación (Martinez, y otros, 2013).

En el contexto de la investigación se hará referencia a las aplicaciones web con fines educativos o de enseñanza en línea (*e-learning*), relacionadas con el PEA.

El término **e-learning** consiste en la educación y capacitación a través de Internet. Este tipo de enseñanza permite la interacción del usuario con el material mediante la utilización de diversas herramientas informáticas (e-ABC, 2017).

¹ Se incluye no sólo texto, sino también otros medios: imágenes, audio, vídeo

Este nuevo concepto educativo, posibilitado por Internet, se posiciona como la forma de capacitación predominante. Esta tecnología ha transformado la educación, abriendo puertas al aprendizaje individual y organizacional. Es por ello que ha ocupado un lugar destacado dentro de las organizaciones empresariales y educativas (e-ABC, 2017).

El término "*e-learning*" es la simplificación de *Electronic Learning*. El mismo reúne a las diferentes tecnologías, y a los aspectos pedagógicos de la enseñanza y el aprendizaje. Entre los beneficios del *e-learning* se encuentran (e-ABC, 2017):

- ❖ **Reducción de costos:** permite reducir y hasta eliminar gastos de traslado, alojamiento, material didáctico.
- ❖ **Rapidez y agilidad:** Las comunicaciones a través de sistemas en la red confiere rapidez y agilidad a las comunicaciones.
- ❖ **Acceso *just-in-time*:** los usuarios pueden acceder al contenido desde cualquier conexión a Internet, cuando les surge la necesidad.
- ❖ **Flexibilidad de la agenda:** no se requiere que un grupo de personas coincidan en tiempo y espacio.

Como **PEA** se define “el movimiento de la actividad cognoscitiva de los alumnos bajo la dirección del maestro, hacia el dominio de los conocimientos, las habilidades, los hábitos y la formación de una concepción científica del mundo” (Autores, 2004). Se considera que en este proceso existe una relación dialéctica entre profesor y estudiante, los cuales se diferencian por sus funciones; el profesor debe estimular, dirigir y controlar el aprendizaje de manera tal que el alumno sea participante activo, consciente en dicho proceso, o sea, “enseñar” y la actividad del alumno es “aprender”. Algunos autores consideran como componentes del proceso de enseñanza a los objetivos, el contenido, los métodos, los medios y su organización los que conforman una relación lógica interna (Ortiz, 2008).

La investigación se centra específicamente en el PEA de la asignatura **Teleinformática**, la cual no es más que la ciencia que estudia la posibilidad de establecer un diálogo a distancia entre equipos informáticos, utilizando como medio de transmisión las redes de Telecomunicaciones. Otra definición puede ser “conjunto de máquinas, técnicas y métodos relacionados entre sí que permiten el proceso de datos a distancia y que participan de la convergencia entre las Telecomunicaciones y la Informática” (FOROUZAN, 2007).

Dicha asignatura es fundamental en el plan de formación de un ingeniero en ciencias informáticas. Quien debe ser capaz de aplicar valiosas competencias como diseñar, desarrollar y mantener aplicaciones

informáticas a gran escala, teniendo en cuenta los roles asociados al ciclo de vida del software. Dirigir y gestionar la planificación, implementación, configuración y mantenimiento de la infraestructura tecnológica de las organizaciones. Todo esto haciendo uso de sus conocimientos de Teleinformática (UCI, 2018).

Para lograr incidir novedosamente en el proceso antes mencionado, se introduce el término **gamificación**, que según Espinosa (2016), es el empleo de mecánicas de juego en entornos y aplicaciones no lúdicas con el fin de potenciar la motivación, la concentración, el esfuerzo, la fidelización y otros valores positivos comunes a todos los juegos. Se trata de una nueva y poderosa estrategia para influir y motivar a grupos de personas.

También puede ser definida como una técnica de aprendizaje que traslada la mecánica de los juegos al ámbito educativo-profesional con el fin de conseguir mejores resultados, ya sea para absorber mejor algunos conocimientos, mejorar alguna habilidad, o bien recompensar acciones concretas, entre otros muchos objetivos (Arteta, 2016).

Este tipo de aprendizaje gana terreno en las metodologías de formación debido a su carácter lúdico, que facilita la interiorización de conocimientos de una forma más divertida, generando una experiencia positiva en el usuario. El modelo de juego realmente funciona porque consigue motivar a los alumnos, desarrollando un mayor compromiso de las personas, e incentivando el ánimo de superación. Se utilizan una serie de técnicas mecánicas y dinámicas extrapoladas de los juegos (Díaz, 2015).

La técnica mecánica definida por Carreño (2015), es la forma de recompensar al usuario en función de los objetivos alcanzados. Algunas de las técnicas mecánicas más utilizadas son las siguientes:

- ❖ **Acumulación de puntos:** se asigna un valor cuantitativo a determinadas acciones y se va acumulando a medida que se realizan.
- ❖ **Escalado de niveles:** se define una serie de niveles que el usuario debe ir superando para llegar al siguiente.
- ❖ **Obtención de premios:** a medida que se consigan diferentes objetivos se van entregando premios a modo de colección.
- ❖ **Regalos:** bienes que se dan al jugador o jugadores de forma gratuita al conseguir un objetivo.
- ❖ **Clasificaciones:** clasificar a los usuarios en función de puntos u objetos logrados, destacando los mejores en una lista o ranking.
- ❖ **Desafíos:** competiciones entre los usuarios, el mejor obtiene los puntos o el premio.

Las técnicas dinámicas presentadas por Rivero (2015), hacen referencia a la motivación del propio usuario para jugar y seguir adelante en la consecución de sus objetivos. Algunas de las técnicas dinámicas más utilizadas son las siguientes:

- ❖ **Recompensas:** obtener un beneficio merecido.
- ❖ **Estatus:** establecerse en un nivel jerárquico social valorado.
- ❖ **Logro:** como superación o satisfacción personal.
- ❖ **Competición:** por el simple afán de competir e intentar ser mejor que los demás.

Teniendo en cuenta las técnicas anteriores presentadas por ambos autores, se adoptará un híbrido de las mismas en la investigación.

1.3. Análisis de sistemas asociados a la enseñanza y el aprendizaje

Caribbean Mind Forge (CMF)

Plataforma desarrollada en la Universidad de las Ciencias Informáticas (UCI) para proporcionar un espacio donde personas del mundo puedan intercambiar experiencias y conocimientos, probar, mejorar y compartir habilidades en la resolución de problemas matemáticos y de agilidad mental. El sistema está concebido para que se desarrollen competencia de habilidades, como Cálculos matemáticos, Memoria Fotográfica, Tecleo rápido, Cálculo de fechas del calendario, Sudoku, Agilidad mental. El usuario debe inscribirse en una de las competencias disponibles y tratar de, en el menor tiempo posible, cumplir con el reto que se le plantee.

Además, cuenta con un archivo de problemas que están disponible las 24 horas para que los navegantes resuelvan e intenten colocarse en los puestos superiores del ranking de CMF. La idea es resolver ejercicios sin equivocarse. El sistema da un veredicto de su solución en Aceptada o Denegada. Para ello se rige por un sistema de puntuación. De igual forma, los usuarios pueden resolver sudokus y ubicarse en un ranking según su esfuerzo.

Duolingo

Es una novedosa plataforma de aprendizaje virtual que, a diferencia de otros sitios, brinda la oportunidad de aprender idioma de manera gratuita. Esta plataforma apuesta por el *crowdsourcing*, es decir, por el trabajo y el bien comunitario. Además de las unidades de aprendizaje, la página web ofrece la posibilidad de tomar un examen de nivel, así como también la oportunidad de traducir textos de Internet a todos aquellos

usuarios que deseen hacerlo, a través de un trabajo colaborativo, comparando más tarde cada una de las respuestas y escogiendo la que mayor puntuación haya obtenido (GAVARRI, 2016).

Otras características:

- ❖ Presenta una interfaz visual agradable
- ❖ Ayuda con la motivación del usuario
- ❖ La mayoría del aprendizaje es visual
- ❖ Penaliza los errores cometidos
- ❖ Es la plataforma gratuita de enseñanza de idiomas más utilizada actualmente

Netacad

Plataforma de aprendizaje en línea de la empresa Cisco. El programa *Networking Academy* de Cisco lleva ofreciendo desde hace 20 años paquetes de formación básicos, medios y avanzados, diseñados conjuntamente con empresas e instituciones educativas. Los currículos, según explica Cisco, responden a las demandas actuales y futuras, incluyendo Internet de las Cosas (IoT), ciberseguridad, *smart grids* o redes de nueva generación. En cuanto a los resultados del programa hay que destacar que el 70% de los alumnos (71% en España) que completan un curso avanzado consiguen un nuevo trabajo, ascienden de categoría profesional o mejoran sus condiciones salariales. Además, NetAcad utiliza una plataforma *cloud* para extender la formación a múltiples colectivos a escala global, incluyendo aquellos en riesgo de exclusión social: víctimas de violencia de género, refugiados, discapacitados y mayores de 65 años, entre otros.

Actualmente, NetAcad se apoya en una red de 10.400 academias (escuelas, colegios y universidades) en 180 países. Con 22.000 formadores y cursos disponibles en 20 idiomas, suma 1,3 millones de estudiantes concurrentes (el 24% mujeres), es la mayor aula tecnológica del mundo.

Moodle

Su nombre proviene del acrónimo de *Modular Object Oriented Dynamic Learning Enviromennt* (Entorno Modular de Aprendizaje Dinámico Orientado a Objetos). Es una herramienta de software libre y gratis. Además, se retroalimenta del trabajo realizado por múltiples instituciones y participantes que colaboran en red, lo cual nos permite acceder libremente e incorporar a nuestra asignatura múltiples módulos y recursos creados por otros usuarios. Actualmente existen en el mundo cerca de 330.000 cursos registrados de 196

países y en 70 lenguas diferentes. La *Open University* del Reino Unido cuenta con 180.000 alumnos registrados (Martinez, 2008).

Se trata de una herramienta de *e-learning*, ya que posibilita el aprendizaje no presencial de los alumnos, aspecto este a considerar con muchos de los alumnos que no pueden acudir a clases por su situación laboral o personal. Lo que hace preciso contar con una herramienta que facilite la virtualidad, aspecto fundamental con el nuevo formato de tutorías que obligará a un mayor trabajo organizativo. Al igual que la gestión de las prácticas y los trabajos, derivados de la implantación de pedagogías más activas en consonancia con la filosofía de la escuela nueva (Martinez, 2008).

Otras características:

- ❖ Funciona sobre Linux, Mac y Windows
- ❖ No es necesario saber programar para poder utilizarlo
- ❖ Los profesores pueden añadir una clave de acceso a los cursos lo que nos permite diferentes opciones como abrir el curso sólo a nuestros estudiantes, o convidar a invitados e incluso a otros profesores a trabajar y cooperar en nuestra asignatura
- ❖ Dispone de una excelente documentación de apoyo en línea y comunidades de usuarios que pueden solucionar cualquier duda, por medio de los diferentes foros destinados a ello
- ❖ Cada participante del curso puede convertirse en profesor además de alumno, pudiendo proporcionar conocimientos exhaustivos sobre un tema en concreto o ayudar a otros compañeros con sus dudas y su proceso de aprendizaje (Martinez, 2008).

Conclusiones del análisis de las soluciones existentes

Luego de realizado un análisis de algunos de los principales sistemas informáticos de tipo web que se utilizan en la enseñanza y el aprendizaje, se concluye que las aplicaciones estudiadas presentan elementos positivos a tener en cuenta, de ellos se puede obtener experiencia acerca de las principales características que se pueden incluir en la propuesta de solución y de la forma en la que los usuarios intercambian con cada una de ellas. A pesar de que todos los sistemas estudiados están diseñados con fines educativos y permiten la auto preparación de los usuarios, son insuficientes en cuanto a la utilización de estrategias de gamificación. Por lo cual no se pueden adaptar a la UCI para trabajar los contenidos de la asignatura

Teleinformática haciendo uso de estrategias de gamificación y según el plan de estudios de la carrera Ingeniería en Ciencias Informáticas de dicha institución. Cabe mencionar además que al ser la propuesta de solución un producto propio de la UCI, se hace más viable para futuros cambios.

1.4. Entorno de desarrollo

Para obtener un software con la calidad requerida se hace necesaria la utilización de metodologías y herramientas que permitan conformar el área de trabajo. Para el desarrollo de SAATI, es necesario realizar un estudio de las herramientas, lenguajes y tecnologías utilizadas en el desarrollo de la propuesta.

1.4.1. Metodologías de desarrollo de software

En la Ingeniería de Software, una metodología de desarrollo de software es la encargada de la separación de este proceso en distintas fases o etapas, que contienen actividades enfocadas a una mejor planificación y administración del mismo. Puede incluir la definición previa de una serie de artefactos que son creados y completados con el equipo que desarrolla o mantiene la aplicación (CMS, 2008).

Existen dos tipos de metodologías de desarrollo de software que engloban el gran número de estas: las ágiles y las pesadas o tradicionales. Las metodologías ágiles son aquellas donde los individuos y las interacciones entre ellos son más importantes que las herramientas y los procesos empleados, la funcionalidad de un producto de software es más importante que generar documentación exhaustiva y para estas metodologías dar respuesta a un cambio es más importante que el seguimiento estricto de un plan (Canós, et al., 2003).

Para el desarrollo de la solución propuesta se seleccionó Proceso Unificado Ágil (*AUP*) por sus siglas en inglés, variación UCI, la cual presenta 3 fases: Inicio, Ejecución y Cierre. Esta variación se escoge debido a que logra estandarizar el proceso de desarrollo de software, dando cumplimiento a las buenas prácticas que define CMMI-DEV² nivel 2. Se logra hablar un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajos.

1.4.2. Modelado de software

El modelado de software es una técnica para tratar con la complejidad inherente a estos sistemas. El uso

² Modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software.

de modelos ayuda al ingeniero de software a "visualizar" el sistema a construir. Además, los modelos de un nivel de abstracción mayor pueden utilizarse para la comunicación con el cliente. Por último, las herramientas de modelado y las de Ingeniería de Software Automatizada pueden ayudar a verificar la corrección del modelo (Academic, 2012).

Lenguaje Unificado de Modelación (UML)

UML en su versión 2.5, por sus siglas en inglés, *Unified Modeling Language*: es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el *OMG (Object Management Group)*. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un plano o modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables (Cortéz M. I., 2017).

Herramienta de modelado UML

Para modelar el sistema se hará uso de una herramienta de Ingeniería de Software Asistida por Computadoras (*CASE*) por sus siglas en inglés. Las herramientas *CASE* son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el diseño de proyectos, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras (Sommerville, 2005).

Visual Paradigm para *UML* en su versión 8.0, se selecciona como herramienta *CASE* de modelado profesional, que utiliza *UML* para la completa representación de las etapas por las que transita un producto de software. Este permite la realización de una amplia gama de diagramas como: casos de uso, de actividades, de despliegue, entre otros, así como la generación de código fuente desde los mismos y la documentación asociada al proceso que esté siendo modelado (Paradigm, 2013).

1.4.3. Lenguajes y herramientas de programación

Python

Python en su versión 3.5, es un lenguaje de programación poderoso y fácil de aprender. Cuenta con

estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. Su elegante sintaxis y tipos de datos dinámicos junto con su naturaleza interpretada, hacen de este un lenguaje ideal para *scripting* y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas (Sevilla V. d, 2016).

JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar *ECMAScript*³. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (*client-side*), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas (Park, et al., 2015).

Librería jQuery

jQuery, es una librería multiplataforma de *JavaScript*, que permite simplificar la manera de interactuar con los documentos *HTML*⁴, manipular el árbol *DOM*⁵, manejar eventos, desarrollar animaciones y agregar interacción con la técnica *AJAX*⁶ a páginas web. Es además un software libre y de código abierto, permitiendo su uso en proyectos tanto libres como privados. *jQuery*, al igual que otras librerías, ofrece una serie de funcionalidades basadas en *JavaScript* que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta, se logran grandes resultados en menos tiempo y espacio (Gutiérrez, 2009).

Lenguaje marcado de hipertexto (HTML 5)

Es un nuevo lenguaje de marcado de hipertexto para presentar y estructurar el contenido en internet. Es la quinta revisión y nueva versión del estándar *HTML*. *HTML5* ofrece nuevas características que proporcionan no sólo un rico soporte multimedia (video y audio), sino que también mejoran el apoyo para la creación de aplicaciones web. Esta nueva tecnología puede proporcionar una solución rentable para implementar

³ Especificación de lenguaje de programación publicada por *ECMA International*, organización internacional basada en membresías de estándares para la comunicación y la información.

⁴ Del inglés *HyperText Markup Language* (Lenguaje de marcas de hipertexto)

⁵ Del inglés *Document Object Model* (Modelo de Objetos del Documento o Modelo en Objetos para la Representación de Documentos).

⁶ Del inglés, acrónimo de JavaScript asíncrono y XML.

aplicaciones para apoyar diferentes dispositivos (por ejemplo, ordenador, tableta, teléfono inteligente personal) sin tener que construir aplicaciones diferentes para cada tipo de dispositivos (Gauchat, 2014).

Hojas de Estilo en Cascada (CSS 3)

Hojas de estilo en cascada (CSS), siglas en inglés de *Cascading Stylesheets*, es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de las páginas web, e interfaces de usuario escritas en *HTML* o *XHTML*; el lenguaje puede ser aplicado a cualquier documento *XML*, incluyendo *XHTML*, *SVG*, *XUL*, *RSS*. También permite aplicar estilos no visuales, como las hojas de estilo auditivas (Sebastián, 2015).

Django

Se seleccionó *Django* en su versión 2.0.2 como *framework* de desarrollo web del lado del servidor, pues, es construido sobre *Python*, fomenta el desarrollo ágil permitiendo la construcción de aplicaciones de alta calidad en un breve período de tiempo. Utiliza una modificación de la arquitectura Modelo – Vista – Controlador (MVC) llamada Modelo – Plantilla – Vista (*MTV*) por sus siglas en inglés. *Django* impulsa el desarrollo de código limpio y promueve la utilización de buenas prácticas de desarrollo web como el principio *RY*⁷. Entre algunos casos de éxito se encuentran: *Instagram*⁸, *Pinterest*⁹ y *The New York Times* (Condori, 2012).

PostgreSQL

Se seleccionó *PostgreSQL* en su versión 11.0.2, como Sistema Gestor de Bases de Datos (SGBD) objeto-relacional que se encuentra distribuido bajo la licencia de software libre BSD. Es mantenido por la organización *PostgreSQL Global Development Team* y cuenta con una amplia comunidad de usuarios y programadores que colaboran activamente. Se centra en que el software sea robusto, de calidad, fácil de mantener y se destaca por su estabilidad, potencia, robustez y facilidad de administración. Funciona muy

⁷ Principio orientado a reducir la repetición de información de todo tipo, especialmente útil en arquitecturas de múltiples niveles.

⁸ Red social y aplicación para subir fotos y videos. Disponible en: <https://www.instagram.com/>

⁹ Plataforma para compartir imágenes. Disponible en <https://es.pinterest.com/>

bien con grandes cantidades de datos y una alta concurrencia de usuarios, accediendo paralelamente al sistema (Martínez, 2010).

PyCharm

PyCharm en su versión 2018.2.4, se seleccionó como Entorno de Desarrollo Integrado (IDE) por sus siglas en inglés, multiplataforma utilizado para desarrollar en el lenguaje de programación *Python*. Provee funcionalidades que permiten una experiencia única y aumentan en gran medida la productividad:

- ❖ Completamiento de código de manera inteligente y señalamiento de errores con reparación de los mismos de forma automatizada
- ❖ Integración con marcos de trabajo de desarrollo web modernos como *Django*
- ❖ Depurador integrado y herramientas de pruebas
- ❖ Soporte para bases de datos e integración con sistemas de control de versiones

En adición a *Python*, *PyCharm* soporta *JavaScript*, *CoffeeScript*, *TypeScript*, *HTML/CSS*, *Cython*, lenguajes de plantilla, *AngularJS*, *Node.js* y más (JetBrains, 2016).

1.4.4. Servidor Web

Un servidor web es un programa que utiliza el protocolo de transferencia de hipertexto, *HTTP* (*Hypertext Transfer Protocol*), para servir los archivos que forman páginas web a los usuarios, en respuesta a sus solicitudes, que son reenviados por los clientes *HTTP* de sus computadoras (Soni, 2016).

Apache HTTP Server

Apache es una aplicación gratuita que convierte un ordenador en un servidor web. Es de código abierto, flexible, rápido y eficiente. *Apache* permite negociar protocolos *HTTP* entre una máquina que haría de servidor web y los otros ordenadores que deseen ver un determinado sitio web. Es multiplataforma y gracias a su popularidad es fácil encontrar documentación para profundizar en las funcionalidades que ofrece. Se caracteriza por su gran escalabilidad, seguridad y rendimiento. Tiene soporte para varios lenguajes como *Perl*, *Python* y *PHP*, lo que permite desarrollar aplicaciones web de gran calidad (Foundation, 2011).

1.5 Herramienta de Control de Versiones

Se utiliza *GitLab Enterprise Edition*, servicio que se brinda por parte del Departamento Técnico de la Dirección General de Producción de la UCI anclado en la URL¹⁰ <https://codecomunidades.prod.uci.cu/>. *GitLab* es un software de código abierto para colaborar en el código. Administrar los repositorios de git con controles de acceso detallados que mantienen su código seguro. Realizar revisiones de código y mejorar la colaboración con solicitudes de fusión. Cada proyecto también puede tener un rastreador de problemas y un wiki. Es utilizado por más de 100,000 organizaciones.

1.6 Conclusiones del capítulo

El estudio de los conceptos relacionados con el dominio de la investigación permitió un mejor entendimiento del negocio. Mediante el análisis de los sistemas homólogos además de obtener experiencia acerca de los principales elementos que se pueden incluir en la aplicación, se observaron algunas carencias que poseen los mismos en cuanto al uso de la gamificación, además se constató que dichos sistemas no podían ser utilizados en la UCI para el aprendizaje de Teleinformática haciendo uso de técnicas de gamificación, por tal motivo se ratifica la necesidad de una nueva herramienta y la selección de las tecnologías que permitan dar comienzo al desarrollo de software. Las herramientas seleccionadas facilitan el desarrollo en cuanto al modelado y la implementación de la aplicación, de forma que se logre una correcta puesta en marcha para el PEA de la asignatura Teleinformática, guiada por la metodología de desarrollo de software *AUP* variación UCI.

¹⁰ Localizador Uniforme de Recursos; secuencia de caracteres que sigue un estándar y que permite denominar recursos dentro del entorno de Internet para que puedan ser localizados.

CAPÍTULO 2. Análisis y diseño de la herramienta

En el presente capítulo se presentan las fases de planificación y diseño definidas por la metodología *AUP* variación UCI, en su escenario número cuatro. Se realiza la propuesta de solución y se planifica el proceso de desarrollo de software, identificando los requisitos funcionales y no funcionales de la propuesta de solución, así como las historias de usuario de cada requisito funcional, patrones de diseño y patrón arquitectónico.

2.1 Modelado del negocio

2.1.1 Modelo conceptual

Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. También se les denomina modelos conceptuales, modelo de objetos del dominio y modelos de objetos de análisis (Larman., 2003).

Componen el modelo conceptual definido para la presente investigación:

UCI: Universidad de las Ciencias Informáticas, centro de altos estudios donde se pondrá en práctica la propuesta de solución.

Teleinformática: asignatura para la cual se desarrolla SAATI.

Estudiante: usuario que utilizará el sistema, para su estudio individual y realizar tareas y evaluaciones orientadas a responder preguntas para su preparación individual.

Profesor: usuario que se encarga de brindar la información necesaria para el estudio individual de los estudiantes y orientar los diferentes tipos de evaluación.

Tema: contiene todos los temas de la asignatura Teleinformática.

Evaluación: unidad de evaluación y retroalimentación que contiene todos los ejercicios definidos por los profesores para los estudiantes.

Gamificación: estrategia que se usa en la evaluación y retroalimentación para apoyar la motivación de los estudiantes al estudio de la Teleinformática.

Competencia: elemento que se activa dentro de la evaluación al hacer uso de la gamificación, y promueve la competitividad entre los estudiantes.

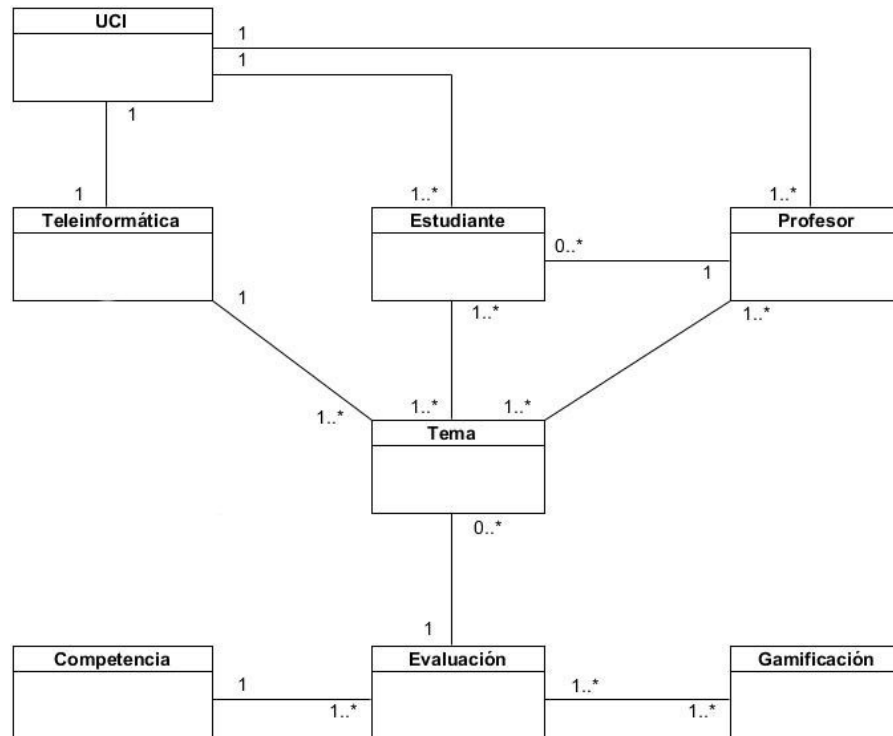


Ilustración 1: Modelo del Dominio (Elaboración propia).

2.1.2 Características de la propuesta de solución

Una vez analizada la situación problemática de la investigación, la solución propuesta consiste en una aplicación para el entrenamiento de habilidades en la asignatura Teleinformática en la carrera Ingeniería en Ciencias Informáticas.

Estructura y funcionamiento

SAATI cuenta con dos fundamentos en los que se basa para lograr una correcta organización del PEA, los cuales se describen a continuación:

Fundamento de capacitación: el estudiante tendrá a su disposición un curso interactivo bastante completo sobre Teleinformática, desglosado en temas de la asignatura que abarcan objetivos principales mediante la realización de ejercicios para el aprendizaje. Además, podrá ver su avance en la asignatura, cuáles objetivos va venciendo y en cuáles tiene dificultades. También podrá revisar en cualquier momento su trayectoria en el sistema para la auto preparación.

Fundamento motivacional: la aplicación le facilita al estudiante un ambiente llamativo que brinda un sistema de puntuación para posicionarlo en un lugar con respecto a los demás, promoviendo así la competitividad y llamando a la sistematicidad en la interacción con la aplicación. Un sistema de bonificación que le permiten hacer uso de mecanismos de desbloqueo cognitivo para que el estudiante no se sienta atrapado y tenga un medio para seguir avanzando, esto se realiza a través de la utilización de comodines que se obtienen en la tienda virtual de la aplicación.

El sistema cuenta además con las siguientes características:

- ❖ Presenta una tabla de posiciones global.
- ❖ Tiene rol de estudiante y profesor. Cada uno con sus respectivas funcionalidades.
- ❖ Tienda. Contiene comodines que se pueden comprar usando los créditos obtenidos.
- ❖ Presenta varios mundos refiriéndose a los temas de la asignatura. Los mismos incluyen niveles y estos a su vez una serie de preguntas.
- ❖ Se le otorgan al estudiante puntos y créditos a medida que va resolviendo las preguntas.

2.2 Especificación de requisitos

Según Martínez e Imelda (2011), un requisito es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste. En el otro extremo, es una definición detallada y formal de una función del sistema.

Requisitos Funcionales.

Tabla 1: Requisitos Funcionales (Elaboración propia).

RF	Funcionalidad	Prioridad
RF1	Registrar usuario (nombre, usuario, contraseña, correo, área, sexo)	Alta
RF2	Autenticar usuario (usuario, contraseña)	Alta
RF3	Mostrar mundos	Alta
RF4	Mostrar niveles	Alta

CAPÍTULO 2: Análisis y diseño de la herramienta.

RF5	Mostrar preguntas	Alta
RF6	Mostrar tabla de posiciones global	Media
RF7	Mostrar tabla de posiciones del mundo	Media
RF8	Mostrar tabla de posiciones de nivel	Media
RF9	Evaluar respuesta (respuesta: <i>input</i> , <i>checkbox</i> , <i>select</i>)	Alta
RF10	Mostrar logros del nivel	Media
RF11	Mostrar puntuación acumulada	Media
RF12	Mostrar créditos acumulados	Media
RF13	Mostrar comodines disponibles en el nivel	Media
RF14	Utilizar comodín	Alta
RF15	Activar audio de fondo	Baja
RF16	Desactivar audio de fondo	Baja
RF17	Revisar recorrido	Baja
RF18	Mostrar comodín en la tienda	Media
RF19	Comprar comodín en la tienda	Media
RF20	Mostrar costo en créditos de comodín en la tienda	Media
RF21	Otorgar puntuación	Alta
RF22	Otorgar créditos	Media
RF23	Cerrar sesión	Baja
RF24	Terminar nivel	Baja
RF25	Mostrar titulo Rey del mundo	Baja

Requisitos no funcionales

RNF1: SAATI debe ser una aplicación web.

RNF2: El sistema debe ser tolerante a fallos, y mostrar solo la información necesaria para orientar al usuario.

RNF3: La propuesta de solución debe ser capaz de ejecutarse en los navegadores empleados en la UCI, específicamente Mozilla Firefox y Google Chrome.

RNF4: El sistema debe permitir que 700 usuarios interactúen con él de manera concurrente.

RNF5: La información manejada por el sistema estará protegida de accesos no autorizados.

RNF6: Ante los errores que puedan ocasionarse en el sistema no se deben mostrar detalles de información que puedan comprometer su seguridad e integridad.

RNF7: El sistema estará disponible las 24 horas del día, todos los días de la semana, garantizando el acceso a la información en cualquier momento.

RNF8: Se permitirá realizar modificaciones posteriores para adaptar mejoras al sistema o en caso que cambien las necesidades de los clientes.

2.3 Historias de usuario

En la metodología de desarrollo de software *AUP* variación UCI los requisitos que debe cumplir el software son especificados por los clientes en las denominadas historias de usuario. Estas historias son descompuestas en tareas de programación y asignadas a los programadores. A continuación, se describen las historias de usuario del sistema de acuerdo a la siguiente plantilla:

Tabla 2: Plantilla para las historias de usuario (Elaboración propia).

Historia de Usuario	
Número: <i>Número de la Historia de Usuario (HU), incremental en el tiempo.</i>	Usuario: <i>El usuario del sistema que utiliza o protagoniza la historia.</i>
Nombre de historia: <i>El nombre de la HU, sirve para identificarla fácilmente entre los desarrolladores y los clientes.</i>	
Prioridad en negocio: <i>Qué tan importante es para el cliente (Alta, Media o Baja).</i>	Riesgo en desarrollo: <i>Qué tan difícil es para el desarrollador (Alto, Medio o Bajo)</i>

CAPÍTULO 2: Análisis y diseño de la herramienta.

Programador Responsable: *Persona que se encarga de la implementación de la funcionalidad.*

Descripción: *La descripción de la historia, detallando las operaciones del usuario y opcionalmente las respuestas del sistema.*

Para el diseño de la propuesta de solución fueron generadas un total de 25 Historias de Usuarios (HU) (Ver Anexo 1), a continuación, solo se muestran 2 de ellas correspondientes a los requisitos de mayor prioridad.

Tabla 3: Historia de Usuario Registrar usuario (Elaboración propia).

Historia de Usuario	
Número: HU_1	Usuario: Usuario
Nombre de historia: Registrar usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo
Programador Responsable: Rafael Frances León	
Descripción: El usuario debe registrarse en el sistema mediante un formulario correspondiente, para poder acceder al sistema	

Tabla 4: Historia de Usuario Evaluar respuesta (Elaboración propia).

Historia de Usuario	
Número: HU_9	Usuario: Estudiante
Nombre de historia: Evaluar respuesta	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Programador Responsable: Rafael Frances León	
Descripción: luego de terminar la pregunta el estudiante presiona el botón evaluar y el sistema califica su respuesta y le otorga la puntuación correspondiente	

2.4 Patrón arquitectónico

El *framework* de desarrollo web Django emplea una modificación del patrón arquitectónico Modelo-Vista-Controlador (MVC), llamada *Model-Template-View (MTV)*, que sería Modelo-Plantilla-Vista, esta forma de trabajar permite que sea pragmático.

A partir de lo planteado se asumirá el patrón arquitectónico *MTV* de Django del cual se explica su funcionamiento a continuación y se podrá observar en la ilustración 2:

- 1- El navegador web envía una solicitud
- 2- La vista interactúa con el modelo para obtener los datos
- 3- La vista llama a la plantilla
- 4- La plantilla muestra la respuesta a la solicitud del navegador

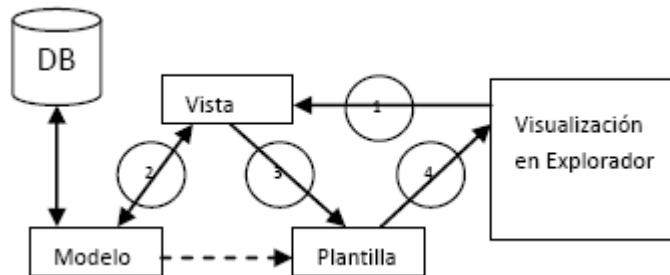


Ilustración 2: Flujo de trabajo del patrón arquitectónico Modelo-Vista-Plantilla (Python-Django. Framework de desarrollo web para perfeccionistas basado en el modelo MTV., 2012)

Modelo: Contiene toda la información sobre los datos. Cada una de las entidades de la base de datos se encuentra en el modelo en forma de clases de Python, y sus atributos se almacenan en variables con ciertos parámetros. También estos archivos poseen métodos, lo que permite indicar y controlar el comportamiento de los datos.

Vista: Es la capa de la lógica de negocios, contiene la lógica que accede al modelo y la delega a la plantilla apropiada. Esta capa sirve de “puente” entre el modelo y la plantilla, se presenta en forma de funciones de *Python* y su función principal es determinar qué datos serán visualizados en las plantillas.

Plantilla: Recibe los datos de la vista y luego los organiza para la presentación al navegador web. Básicamente es una página *HTML (HyperText Markup Language)* con algunas etiquetas extras que son propias del *Django*.

Para cumplir con la arquitectura seleccionada, la implementación de SAATI cuenta con la siguiente estructura:

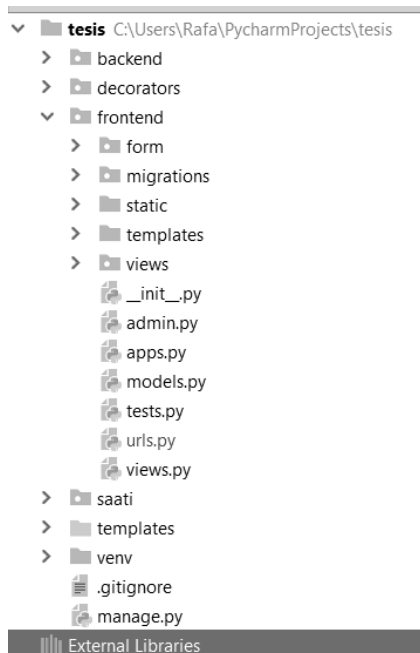


Ilustración 3: Estructura de directorios del sistema (Elaboración propia).

En la estructura de directorios anterior se pudo observar cómo se encuentran distribuidos en el sistema los elementos que conforman la arquitectura *MTV*.

2.5 Patrones de diseño

Un patrón es una descripción del problema y la esencia de su solución, de modo que la solución puede reutilizarse en diferentes configuraciones. El patrón no es una especificación detallada. Más bien, puede considerarla como una descripción de sabiduría y experiencia acumuladas, una solución bien probada a un problema común (Sommerville, 2011).

Patrones *GRASP*

Los Patrones de Principios Generales para Asignar Responsabilidades (*GRASP*) por sus siglas en inglés, describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones.

Según Visconti (2012), los patrones *GRASP* son:

Experto: garantiza que la responsabilidad de la creación de un objeto o la implementación de un método, recaiga sobre la clase que conoce toda la información necesaria para crearlo, lo que contribuye a un adecuado encapsulamiento, favoreciendo la robustez y fácil mantenimiento del sistema.

Creador: se asigna la responsabilidad a una clase de crear cuando contiene, agrega, compone, almacena o usa otra clase, lo que brinda una alta posibilidad de reutilizar la clase creadora.

Este patrón se pone de manifiesto en las vistas, donde se implementan las clases controladoras para crear objetos del modelo de datos, permitiendo acceder a estos de forma directa en cada una de las vistas para enviar la información necesaria hacia las plantillas, respondiendo a las peticiones del usuario a través de un navegador web.

```
from django.views.generic.list import ListView
from backend.models.Mundo import Mundo

class ListMundos(ListView):
    model = Mundo
    template_name = 'Mundos/mundos.html'
```

Ilustración 4: Código fuente para mostrar los mundos (Elaboración propia)

Bajo acoplamiento: es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras clases.

Este patrón ya viene incluido con *Django*, el cual permite un bajo acoplamiento entre las piezas, lo que evita las dependencias, por ejemplo, a la hora de realizar cambios en las configuraciones de las *URL*, en la *BD* y las plantillas *HTML*, basta solo con realizarlo una sola vez.

Alta Cohesión: asigna responsabilidades de manera que una clase no tenga muchas funcionalidades no relacionadas o no realice un trabajo excesivo. Este patrón incrementa la claridad y facilita la comprensión

del diseño. Al usar este patrón se garantiza una mejor eficiencia en cuanto al tiempo de respuesta de las peticiones del usuario.

Una de las características de *Django* es la organización del trabajo en cuanto a la estructura del proyecto, lo cual permite crear y trabajar con clases con una alta cohesión. Por ejemplo, se puede observar en el sistema que cada clase controladora se ajusta a manejar solo las responsabilidades correspondientes a las entidades con las que se relaciona. Esto hace posible que el sistema sea flexible a cambios sustanciales con efecto mínimo.

Controlador: Asignar la responsabilidad de administrar un mensaje de evento del sistema a una clase que representa el sistema global, dispositivo, subsistema o representa un escenario de caso de uso en el que tiene lugar el evento del sistema.

Este patrón es empleado en todo el sistema debido a que cada uno de los eventos generados por el usuario es redirigido a una clase o función controladora que realiza las operaciones solicitadas, manteniendo siempre la alta cohesión (Larman., 2003).

Patrones GoF

Representan patrones que dan soluciones técnicas basadas en programación orientada a objeto (POO), que favorecen la reutilización del código (Guerrero, 2013).

El catálogo de patrones más famoso es el contenido en el libro “*Design Patterns: Elements of Reusable Object-Oriented Software*”, también conocido como: El libro *GOF (Gang-Of-Four Book)*. Según este documento, estos patrones se clasifican según su propósito en creacionales, estructurales y de composición, mientras que respecto a su ámbito se clasifican en clases y objetos:

Decorator (Decorador): Permite agregar responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades.

Como ejemplo de este patrón se evidencia la utilización de los decoradores de *Django* o definidos por el programador, cómo es el caso de *login_required*:

```
@login_required(login_url='login')
@permission_required(perm='nivel.list_nivel',reverse_lazy('403_frobiden'))
def dispatch(self, request, *args, **kwargs):
    super(ListNivel, self).dispatch(*args, **kwargs)
```

Ilustración 5: Decoradores para el control de sesiones y permisos (Elaboración propia)

2.6 Modelo de datos

Un modelo de datos puede definirse como abstracciones mediante las cuales puede realizarse una representación de los problemas que se desean resolver. Es la estructura de una base de datos a la cual se denomina Esquema (Cabello, 2010).

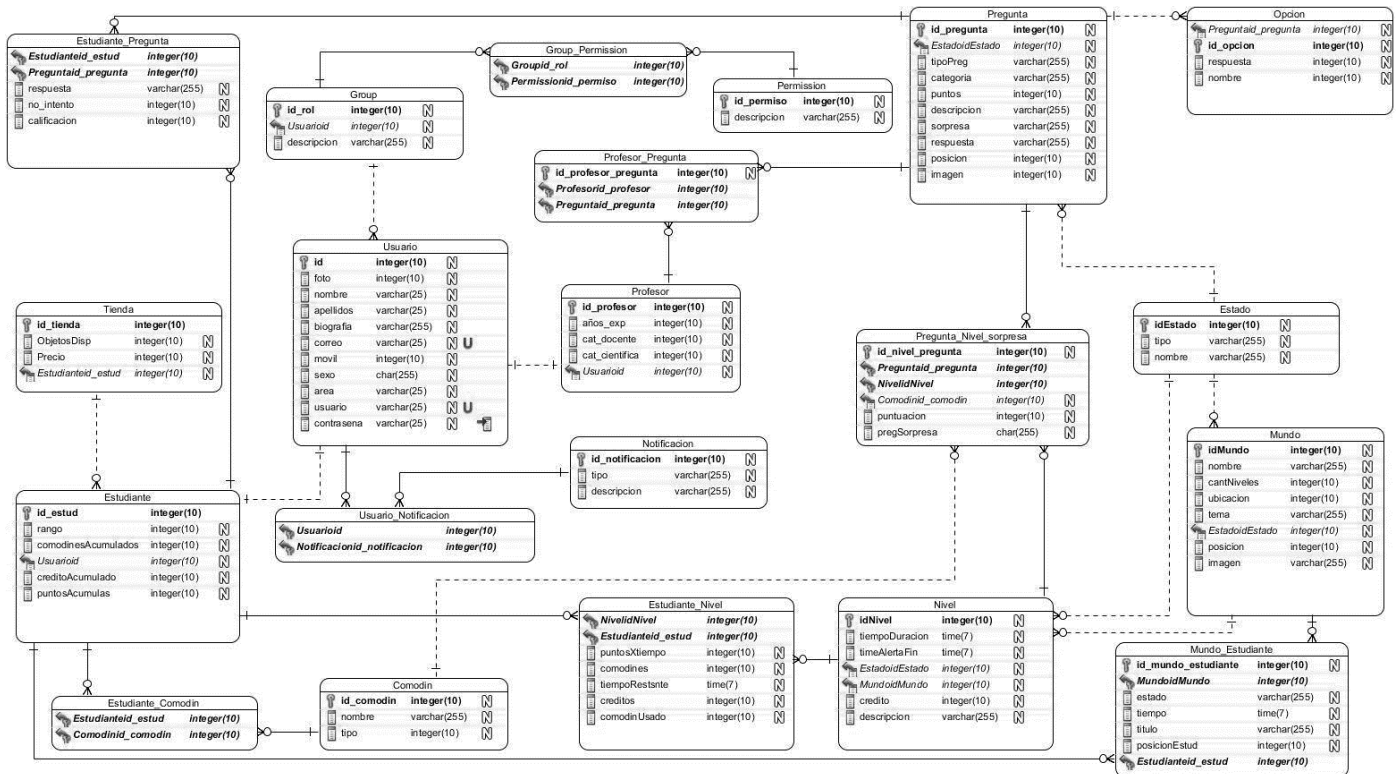


Ilustración 6: Modelo de Datos (Elaboración propia).




2.7 Diagrama de Clases del diseño

Un Diagrama de Clases de Diseño muestra la especificación para las clases de software de una aplicación. Incluye la siguiente información:

- ❖ Clases, asociaciones y atributos
- ❖ Interfaces, con sus operaciones y constantes
- ❖ Métodos
- ❖ Navegabilidad
- ❖ Dependencias

Se utilizan los siguientes estereotipos:

Tabla 5: Estereotipos web para UML (Elaboración propia).

Estereotipos para las clases	
Estereotipo	Descripción
 <i>Client Page</i>	Representan páginas que son dibujadas por el navegador web y pueden ser una combinación de algún o algunos lenguajes de marcado, <i>scripts</i> del lado del cliente, islas de datos, etc.
 <i>Server Page</i>	Representa una página web que tiene <i>scripts</i> ejecutados por el servidor. Estos <i>scripts</i> interactúan con los recursos que se encuentran al alcance del servidor. Sólo puede mantener relaciones con objetos que se encuentren en el servidor.
 <i>Form</i>	Representa una colección de campos de entrada que forman parte con una página del lado cliente (<i>Client Page</i>). Tiene una correspondencia directa con la etiqueta <code><FORM></code> de <i>HTML</i> .
Estereotipos para las Relaciones entre las Clases	
<i>Link</i>	Representa un apuntador desde una " <i>client page</i> " hacia una " <i>client</i> "

	<i>page</i> ” o “ <i>server page</i> ”. Corresponde directamente con una etiqueta <a> de <i>HTML</i>
<i>Submit</i>	Esta relación siempre se da entre una “ <i>form</i> ” y una “ <i>server page</i> ”, por supuesto, la “ <i>server page</i> ” procesa los datos que la “ <i>form</i> ” le envía (<i>submits</i>)
<i>Build</i>	Sirve para identificar cuáles “ <i>server page</i> ” son responsables de la creación de una “ <i>client page</i> ”. Una “ <i>server page</i> ” puede crear varias “ <i>client page</i> ”, pero una “ <i>client page</i> ” sólo puede ser creada por una sola “ <i>server page</i> ”. Esta relación siempre es unidireccional
<i>Redirect</i>	Esta es también una relación unidireccional que indica que una página web redirige hacia otra. En caso de que la página origen sea una “ <i>client page</i> ” esta asociación corresponderá con la etiqueta “ <i>META</i> ” y valor <i>HTTP-EQUIV</i> de “ <i>Refresh</i> ”.

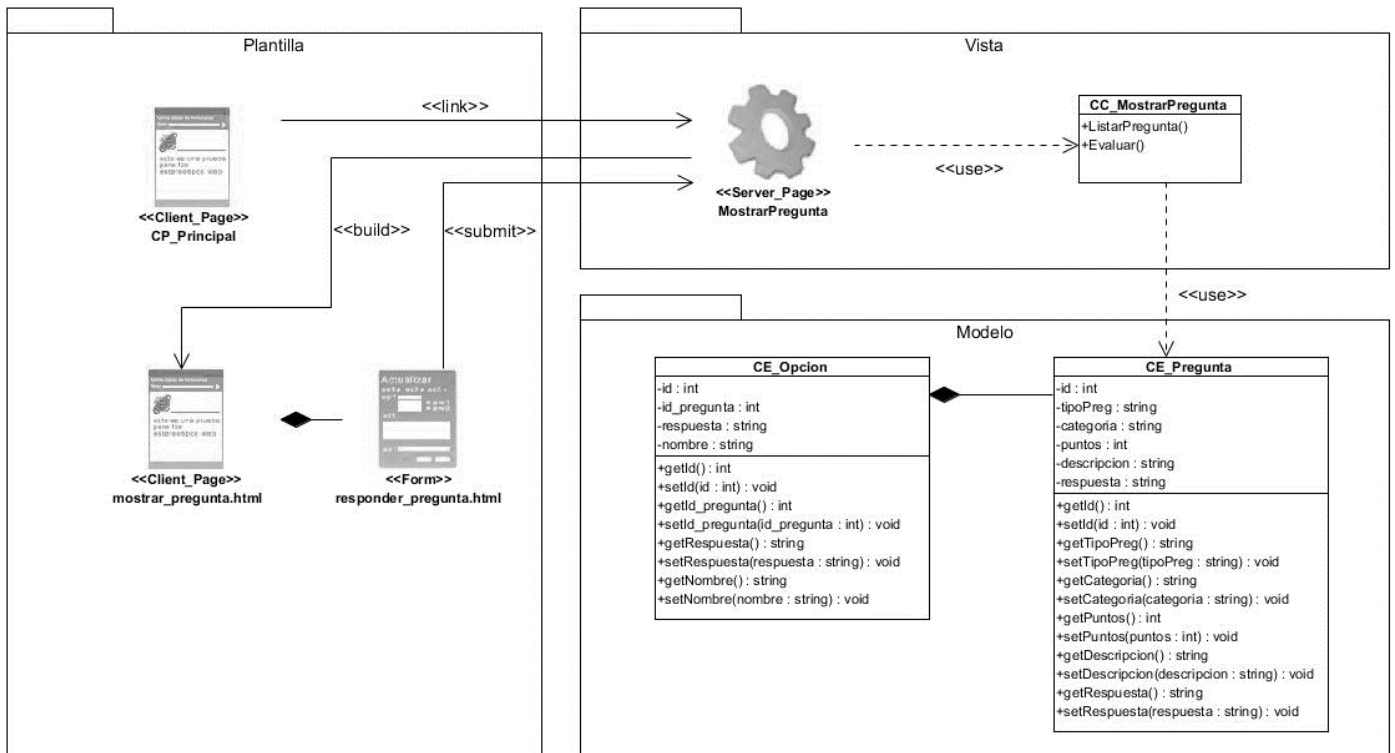


Ilustración 7: Diagrama de clases del diseño de la HU Mostrar Pregunta (Elaboración propia).

2.8 Conclusiones del capítulo

En este capítulo se han abordado los elementos del análisis y diseño de SAATI. El levantamiento de los requisitos funcionales y no funcionales obtenidos a partir del proceso de identificación de los requisitos, sirvieron de guía para desarrollar las distintas funcionalidades y de este modo satisfacer las necesidades detectadas. La utilización de los patrones de diseño permitió identificar aspectos importantes de la estructura del diseño del sistema propuesto, lo que garantizó lograr una adecuada reutilización del código para futuras versiones. Las descripciones de las historias de usuario y la elaboración del diagrama de clases del diseño posibilitaron una mejor comprensión del funcionamiento de la propuesta de solución. Adoptar la arquitectura de software Modelo-Vista-Plantilla propuesta por el *framework* de desarrollo utilizado, permitió una propicia organización del sistema a implementar. Además, se definieron las estrategias de gamificación para lograr una mejor motivación en los estudiantes.

CAPÍTULO 3. Implementación y pruebas de la propuesta de solución

3.1 Introducción

En el presente capítulo se muestra el diagrama de componentes como resultado del diseño desarrollado. La fase de implementación comprende la materialización, en forma de código, de los artefactos y las descripciones con el objetivo de conformar el producto final requerido por el cliente. Además, en este capítulo se presentan los resultados de aplicar las pruebas de software y la validación del sistema, con el objetivo de corroborar la correspondencia entre el producto y los requisitos definidos anteriormente.

3.2 Estándares de codificación

Cada programador tiene su propia forma de escribir los códigos, pero de la forma que se use depende la facilidad de que otros programadores entiendan el código y se les facilite su reutilización, de ahí se desprende la importancia de los estilos de programación, también conocidos como estándares o convenciones de código, los cuales definen un grupo de convenciones para escribir código fuente en ciertos lenguajes de programación (Abraira, 2016).

A continuación, se especifican los estándares de codificación a utilizar en la construcción de la solución:

Tabla 6: Estándares de codificación (Elaboración propia).

Tipo de estándar	Descripción
Alineación	<ul style="list-style-type: none"> ❖ Las líneas de continuación deben alinearse verticalmente con el carácter que se ha utilizado (paréntesis, llaves, corchetes). ❖ Utilizar una alineación de una tabulación para cada línea con excepción de la primera. ❖ La alineación se realizará solamente con tabulaciones, no deben utilizarse nunca los cuatro (4) espacios.
Máxima longitud de las líneas	<ul style="list-style-type: none"> ❖ Todas las líneas deben estar limitadas a un máximo de setenta y nueve (79) caracteres.

	<ul style="list-style-type: none"> ❖ Dentro de paréntesis, corchetes o llaves se puede utilizar la continuación implícita para cortar las líneas largas. ❖ En cualquier circunstancia se puede utilizar el carácter “\” para cortar las líneas largas.
Líneas en blanco	<ul style="list-style-type: none"> ❖ Separar las funciones de alto nivel y definiciones de clases con dos (2) líneas en blanco. ❖ Las definiciones de métodos dentro de una clase deben separarse por una (1) línea en blanco. ❖ Se pueden utilizar líneas en blanco, escasamente, para separar secciones lógicas.
Codificaciones	<ul style="list-style-type: none"> ❖ Utilizar la codificación UTF-8. ❖ Se pueden incluir cadenas que no correspondan a esta codificación utilizando “\x”, “\u” o “\U”.
Importación	<ul style="list-style-type: none"> ❖ Las importaciones deben estar en líneas separadas. ❖ Siempre deben colocarse al comienzo del archivo. ❖ Deben quedar agrupadas de la siguiente forma: <ol style="list-style-type: none"> 1. Importaciones de la librería estándar. 2. Importaciones terceras relacionadas. 3. Importaciones locales de la aplicación / librerías. ❖ Cada grupo de importaciones debe estar separado por una línea en blanco.
Espacios en blanco en expresiones y sentencias	<ul style="list-style-type: none"> ❖ Evitar utilizar espacios en blanco en las siguientes situaciones:

Comentarios

1. Inmediatamente dentro de paréntesis, corchetes y llaves.
 2. Inmediatamente antes de una coma, un punto y coma o dos puntos.
 3. Inmediatamente antes del paréntesis que comienza la lista de argumentos en la llamada a una función.
 4. Inmediatamente antes de un corchete que empieza una indexación.
 5. Más de un espacio alrededor de un operador de asignación (u otro) para alinearlos con otro.
- ❖ Deben rodearse con exactamente un espacio los siguientes operadores binarios:
 1. Asignación (=).
 2. Asignación de aumentación (+=, -=, etc.).
 3. Comparación (==, <, >, >=, <=, !=, <>, in, not in, is, is not).
 4. Expresiones lógicas (and, or, not).
 - ❖ Si se utilizan operadores con prioridad diferente se aconseja rodear con espacios a los operadores de menor prioridad.
 - ❖ No utilizar espacios alrededor del igual (=) cuando es utilizado para indicar un argumento de una función o un parámetro con un valor por defecto.
 - ❖ Los comentarios deben ser oraciones completas.
 - ❖ Si un comentario es una frase u oración su primera palabra debe comenzar con mayúscula a menos que sea un identificador que comience con minúscula.
 - ❖ Nunca cambiar las minúsculas y mayúsculas en los identificadores de clases, objetos, funciones, etc.

	<ul style="list-style-type: none"> ❖ Si un comentario es corto el punto final puede omitirse.
Comentarios en bloque	<ul style="list-style-type: none"> ❖ Deben estar alienados al mismo nivel que el código a comentar. ❖ Cada línea de un comentario en bloque comienza con un numeral (#) y un espacio en blanco.
Comentarios en la misma línea	<ul style="list-style-type: none"> ❖ Se recomienda utilizarlos escasamente. ❖ Se debe definir comenzando por un numeral (#) seguido de un espacio en blanco. ❖ Deben ubicarse en la misma línea que se desea comentar.
Cadenas de documentación	<ul style="list-style-type: none"> ❖ Deben quedar documentados todos los módulos, funciones, clases y métodos públicos. ❖ Para definir una cadena de documentación debe quedar encerrada dentro de ("""). ❖ Los (""") que finalizan una cadena de documentación deben quedar en una línea a no ser que la cadena sea de una sola línea.
Convenciones de nombramiento	<ul style="list-style-type: none"> ❖ Nunca se deben utilizar como simples caracteres para nombres de variables los caracteres ele minúscula "l", o mayúscula "O", ele mayúscula "L" ya que en algunas fuentes son indistinguibles de los números uno (1) y cero (0). ❖ Los módulos deben tener un nombre corto y en minúscula.

- | | |
|--|---|
| | <ul style="list-style-type: none">❖ Los nombres de clases deben utilizar la convención “CapWords” (palabras que comienzan con mayúsculas).❖ Los nombres de las excepciones deben estar escrito también en la convención “CapWords” utilizando el sufijo “Error”.❖ Los nombres de las funciones deben estar escritos en minúscula separando las palabras con un guion bajo “_”.❖ Las constantes deben quedar escritas con letras mayúsculas separando las palabras por un guion bajo (_). |
|--|---|

3.3 Diagrama de componentes

El diagrama de componentes está incluido en *UML* clasificado como diagrama de estructura. Por lo tanto, representa de forma estática el sistema de información. Proporcionan una vista de alto nivel de los componentes dentro de un sistema. Los componentes pueden ser un componente de software, como una base de datos o una interfaz de usuario; o un componente de hardware como un circuito, microchip o dispositivo; o una unidad de negocio como un proveedor, nómina o envío (DiagramasUML, 2019).

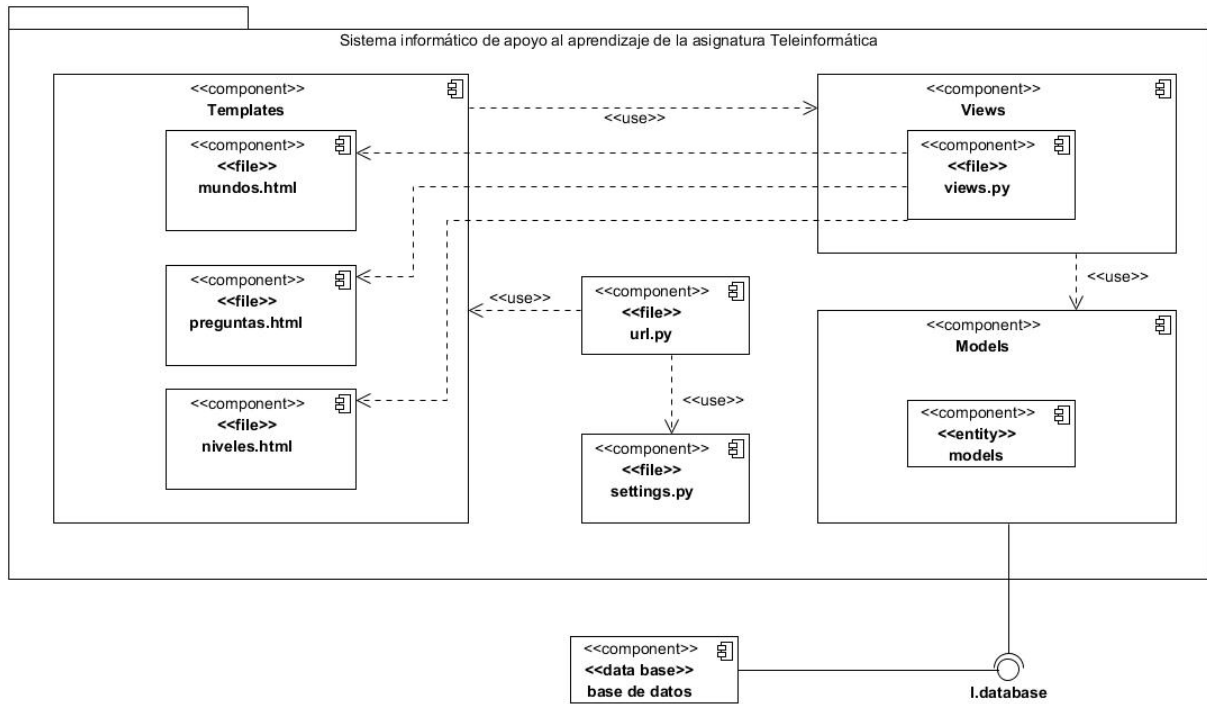


Ilustración 8: Diagrama de Componentes (Elaboración propia)

En el diagrama anterior muestra los componentes que componen SAATI, donde los componentes *views* y *templates*, contienen todos los ficheros que forman parte del mismo.

A continuación, en la Tabla 7, se describen los elementos que conforman al diagrama de componentes de la ilustración 8:

Tabla 7: Descripción de los ficheros que componen los componentes (Elaboración propia)

	Ficheros	Descripción
Templates	<i>menu.html</i>	Este fichero muestra la página principal del sistema, la cual permite el acceso a las diferentes partes del mismo.

Views	<i>mundos.html</i>	Este fichero muestra los mundos asociados a los temas de la asignatura Teleinformática, estos contienen los niveles que deberá vencer el estudiante.
	<i>niveles.html</i>	Este fichero muestra los niveles pertenecientes a un mundo seleccionado, estos contienen las preguntas que deberá resolver el estudiante.
	<i>preguntas.html</i>	Este fichero muestra la interfaz donde el estudiante responde las preguntas y es evaluado.
	<i>modal_nivel.html</i>	Este fichero muestra una ventana emergente cuando se selecciona el nivel correspondiente, donde se puede acceder a la tabla de posiciones de dicho nivel o comenzar el nivel.
	<i>Mundos.py</i>	Este fichero es el encargado de enviar a la plantilla del mundo toda la información del mismo.
	<i>Niveles.py</i>	Este fichero es el encargado de enviar a la plantilla de niveles la información de los mismos e información de los modelos necesarios que se muestran en dicha plantilla, así como redirigir a la plantilla del modal cuando este se levanta.
	<i>Preguntas.py</i>	Este fichero se encarga de enviar a la plantilla las preguntas relacionadas con un nivel determinado que pertenecen a un mundo y toda la información de los modelos que son necesarios para mostrar en la plantilla.

3.4 Diagrama de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Un nodo es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional, que generalmente tiene memoria y a menudo, capacidad de procesamiento. Los nodos se utilizan para modelar la topología del hardware sobre el que se ejecuta el sistema. Representan un procesador o un dispositivo sobre el que se pueden desplegar los componentes. La relación entre nodo y componente que despliega puede mostrarse con una relación de dependencia (Prefacio, 2000). La siguiente ilustración representa la vista de despliegue del sistema.



Ilustración 9: Representación del Diagrama de Despliegue (Elaboración propia).

PC Cliente: Se refiere a las estaciones de trabajo que realizan las peticiones al servidor de aplicaciones donde está hospedado SAATI, mediante un navegador web utilizando el protocolo de comunicación *HTTPS* por el puerto 443.

Servidor Web: Es el encargado de brindar la interfaz de la plataforma para que los usuarios puedan hacer uso de esta, almacena todo el código fuente del sistema y se comunica por medio de los protocolos *TCP* con el servidor de bases de datos.

Sistema Gestor de Bases de Datos: Almacena toda la información que brinda la plataforma hospedada en el servidor de aplicaciones. La información es obtenida o modificada en dependencia del nivel de privilegio del usuario que realiza la petición. La comunicación con el servidor de aplicaciones es a través del protocolo *TCP* empleando el puerto 5432.

3.5 Interfaz gráfica de usuario

Una vez finalizado el desarrollo del software es posible visualizar las pantallas principales de SAATI, donde se observa el resultado obtenido durante la implementación de las historias de usuarios descritas en el capítulo anterior.

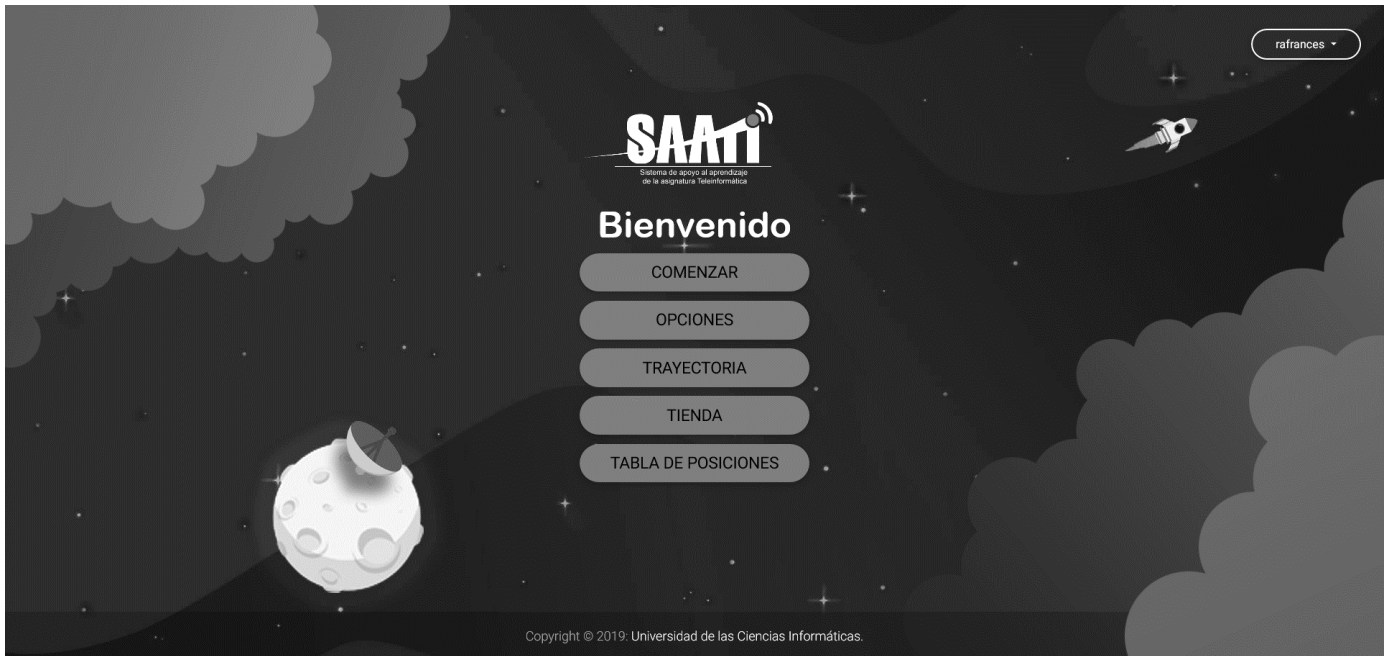


Ilustración 10: Interfaz principal del sistema (Elaboración propia)

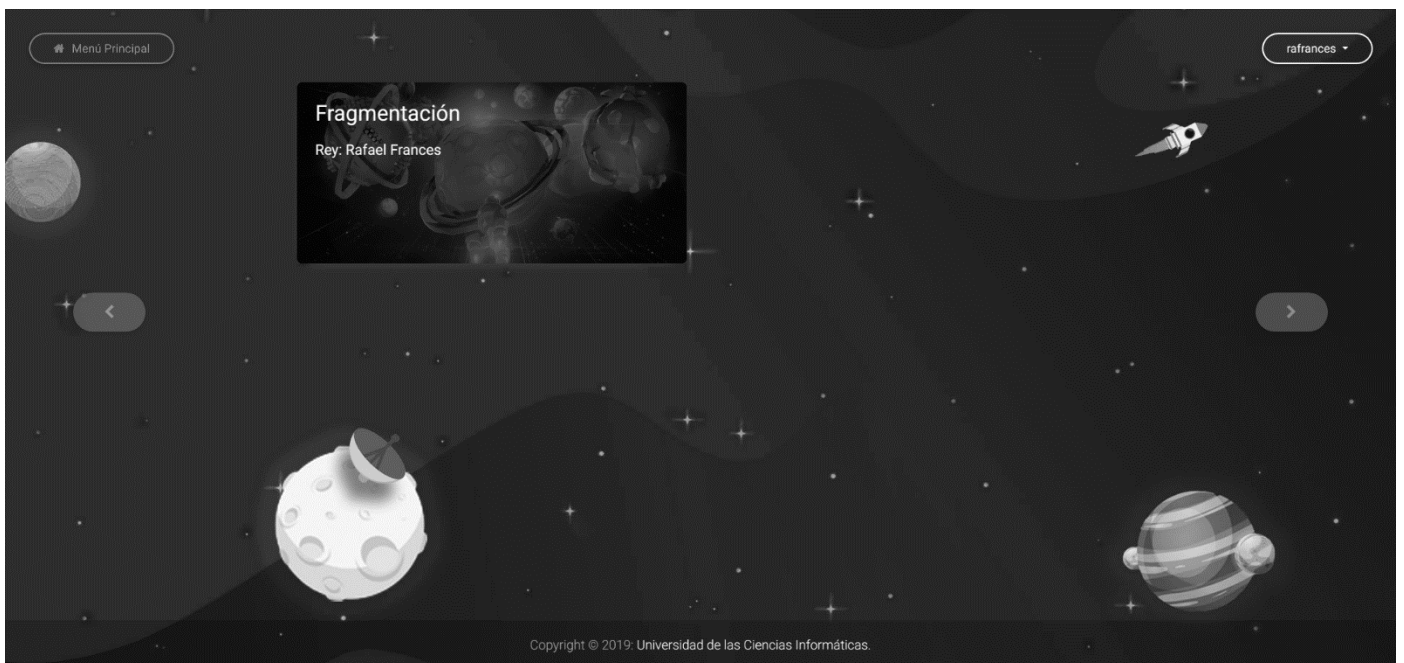


Ilustración 11: Interfaz de selección mundos (Elaboración propia)

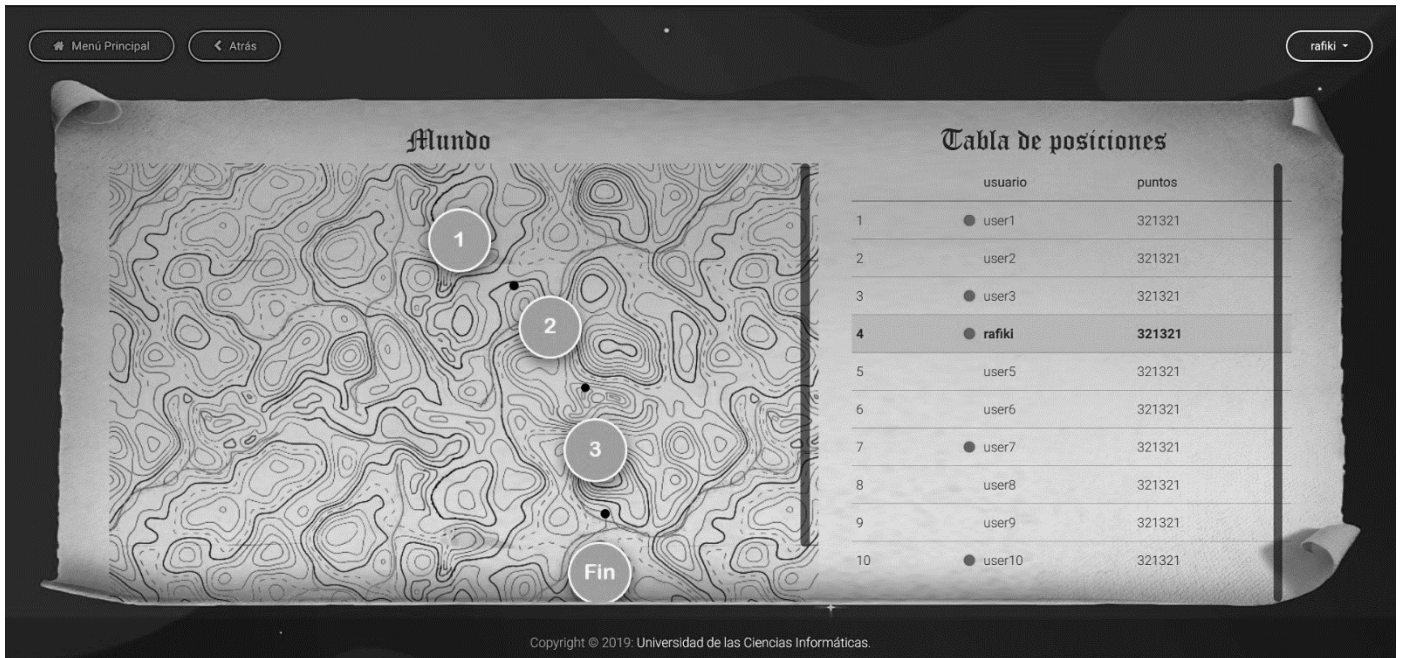


Ilustración 12: Interfaz de selección del nivel (Elaboración propia)



Ilustración 13: Interfaz de realización de ejercicios (Elaboración propia)

3.6 Pruebas de software

El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales, es decir, la realización de las pruebas para la detección de errores. Además, son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa (Pressman, 2015).

3.6.1 Pruebas de aceptación

Según la metodología *AUP* variación UCI, es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales fue construido.

Con el objetivo de evaluar la solución implementada se utilizó la técnica de ladov. Esta técnica evalúa el nivel de satisfacción del usuario, de esta forma se conoce si los componentes realizados cumplen las expectativas de los clientes; está conformada por cinco preguntas: 3 cerradas y 2 abiertas (efdeportes, 2002).

La técnica de ladov constituye una vía indirecta para el estudio de la satisfacción, debido a que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre las preguntas cerradas (preguntas 1, 2 y 3) que se intercalan dentro de un cuestionario (ver Anexo 2) y cuya relación el sujeto desconoce. Estas tres preguntas se relacionan a través de lo que se denomina el "Cuadro Lógico de ladov" (efdeportes, 2002). El cual se muestra en la siguiente tabla:

Tabla 8: Cuadro Lógico de ladov (Elaboración Propia)

	1. ¿Considera usted que es posible obtener mejores resultados en la asignatura Teleinformática sin una adecuada motivación de los estudiantes?		
	No	No sé	Sí
3. ¿Satisface sus necesidades,	2. ¿Utilizaría usted la aplicación propuesta a la hora de estudiar y aprender Teleinformática?		

CAPITULO 3. Implementación y pruebas de la propuesta de solución

según el rol que desempeña, en la docencia con respecto a la asignatura Teleinformática, el sistema propuesto?	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me satisface mucho.	1	2	6	2	2	6	6	6	6
No me satisface tanto.	2	2	3	2	3	3	6	3	6
Me da lo mismo.	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me satisface.	6	3	6	3	4	4	3	4	4
No me satisface nada.	6	6	6	6	4	4	6	4	5
No sé qué decir.	2	3	6	3	3	3	6	3	4

El número resultante de la interrelación de las tres preguntas nos indica la posición de cada sujeto en la escala de satisfacción.

La escala de satisfacción es la siguiente:

- (1). Clara satisfacción
- (2). Más satisfecho que insatisfecho
- (3). No definida
- (4). Más insatisfecho que satisfecho

(5). Clara insatisfacción

(6). Contradictoria.

Para medir el grado de satisfacción se tomó una muestra de 10 estudiantes del tercer año de la Facultad 1. Esta técnica también permite obtener el índice de satisfacción grupal (ISG), para lo cual se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1 como se presenta en la siguiente Tabla.

Tabla 9: Índices para el cálculo de satisfacción grupal (efdeportes, 2002)

Índice	Escala
+1	Máximo de satisfacción
0.5	Más satisfecho que insatisfecho
0	No definido y contradictorio
-0.5	Más insatisfecho que satisfecho
-1	Máxima insatisfacción

La satisfacción grupal se calcula mediante la siguiente expresión:

$$ISG = \frac{A(+1) + B(+0.5) + C(0) + D(-0.5) + E(-1)}{N}$$

Donde:

- A, B, C, D, E representan el número de sujetos con índice individual 1, 2, 3 o 6, 4, 5
- N representa el número total de sujetos del grupo

Esto permite reconocer las categorías grupales:

- Insatisfacción: desde (-1) hasta (-0,5)

CAPITULO 3. Implementación y pruebas de la propuesta de solución

- Contradictorio: desde (-0,49) hasta (+0,49)
- Satisfacción: desde (+0,5) hasta (1)

Luego se aplicó la técnica calculando el índice de satisfacción grupal (ISG) obteniendo como resultado 0.80 aproximadamente, lo que significa una clara satisfacción con el desarrollo de SAATI.

$$ISG = \frac{7(+1) + 2(+0.5) + 1(0) + 0(-0.5) + 0(-1)}{10} = 0.80$$

Por otra parte, se le entregó la aplicación al cliente, quien emitió su criterio a través de una carta de aceptación (ver Anexo 4), para validar que el sistema satisface sus necesidades. Para respaldar dicho criterio, se solicitó la colaboración de un grupo de especialistas de la asignatura Teleinformática, los cuales se relacionan a continuación:

- Alexander López Pupo

Profesor del Dpto. de Programación, en la disciplina Sistemas Digitales. Imparte las asignaturas “Arquitectura de Computadoras” y “Teleinformática”, esta última con 6 años de experiencia. Forma parte del actual claustro de Teleinformática en la Facultad 1 de la UCI.

Categoría Docente: Asistente

Categoría Científica: -

- Ailyn Gutiérrez Ferrera

Profesora del Dpto. de Programación, en la disciplina Sistemas Digitales, ocupa el cargo de Jefa de Dpto. Imparte las asignaturas “Arquitectura de Computadoras” y “Teleinformática”, esta última con 6 años de experiencia. Forma parte del actual claustro de Teleinformática en la Facultad 1 de la UCI.

Categoría Docente: Asistente

Categoría Científica: Máster

- Pedro Randy Rodríguez Rivera

Profesor del Dpto. de Programación, en la disciplina Sistemas Digitales. Imparte las asignaturas “Arquitectura de Computadoras” y “Teleinformática”, esta última con 3 años de experiencia. Forma parte del actual claustro de Teleinformática en la Facultad 1 de la UCI.

Categoría Docente: Asistente

Categoría Científica: -

- Ruth Yurina Vega

Profesora del Dpto. de Programación, en la disciplina Sistemas Digitales. Imparte las asignaturas “Redes y Seguridad Informática” y “Configuración de Equipos de Interconexión de Redes”, con 5 años de experiencia en ambas. Los objetivos de estas asignaturas son continuidad directa de los objetivos evaluados en Teleinformática.

Categoría Docente: Asistente

Categoría Científica: Máster

- Nestor Ariel Delgado Pacheco

Profesor del Dpto. de Programación, en la disciplina de Programación y también en la de Sistemas Digitales. Imparte las asignaturas “Programación 2” y “Redes y Seguridad Informática”, esta última con 4 años de experiencia. Los objetivos de la asignatura “Redes y Seguridad Informática” son continuidad directa de los objetivos evaluados en Teleinformática.

Categoría Docente: Instructor.

Categoría Científica: -

3.6.2 Pruebas de Carga y Estrés

La carga de trabajo se refiere a la capacidad máxima que tiene un servidor web, para atender a un conjunto de usuarios de manera simultánea. Por ello, las actividades de esta etapa tienen relación con comprobar, de manera anticipada, el funcionamiento que tendrá el servidor del Sitio Web cuando esté en plena operación (Díaz, y otros, 2014).

Las pruebas en este caso consisten en simular una carga de trabajo similar y superior a la que tendrá cuando el sitio esté funcionando, con el fin de detectar si el software instalado cumple con los requerimientos de muchos usuarios simultáneos y también si el hardware es capaz de soportar la cantidad de visitas esperadas (Serna, y otros, 2015).

Resultados de las pruebas de carga y estrés

Para realizar las pruebas de carga y estrés se efectuó un análisis de la cantidad de usuarios en la Universidad. Teniendo en cuenta que el sistema está pensado para el apoyo de la asignatura Teleinformática, y que la matrícula del 3er año es de cerca de 700 estudiantes, se desarrollaron las pruebas para un total de 700, 1000 y 2000 usuarios concurrentes.

Las pruebas se desarrollaron con el apoyo de la herramienta *Apache JMeter 2.12*, en la que se simuló el

entorno donde debe interactuar el sistema para obtener la información más correcta acerca del comportamiento y resultados en general. Por lo que fue elegido un ambiente con las siguientes características:

Software:

- ❖ Sistema Operativo: Windows 10 Arquitectura de 64bits
- ❖ Sistema Gestor de Bases de Datos: PostgreSQL 11.0.2
- ❖ Servidor Web: Apache v2.4
- ❖ Máximo de hilos concurrentes: entre 700 y 1000

Hardware:

- ❖ Microprocessor: Intel Core (TM) i3 – 2120M CPU @ 3.30 GHz
- ❖ Memoria RAM: 4GB
- ❖ Tarjeta de Red: Ethernet 10/100 Mbps

A continuación, se describen las variables que miden el resultado de las pruebas de carga y estrés realizadas al sistema:

#Muestras: Cantidad de peticiones realizadas para cada *URL*.

Media: Tiempo promedio en milisegundos en el que se obtienen los resultados.

Mín.: Tiempo mínimo que demora un hilo en acceder a una página.

Máx.: Tiempo máximo que demora un hilo en acceder a una página.

% Error: Por ciento de error de las páginas que no se llegaron a cargar de manera satisfactoria.

Rendimiento: El rendimiento se mide en cantidad de solicitudes por segundo.

Kb/s: Velocidad de carga de las páginas.

Resultados de las pruebas de Carga y Estrés.

Como se muestra en la Tabla 10, se simularon las peticiones realizadas al sistema por un total de 700, 1000 y 2000 usuarios simultáneamente, realizando hasta 5 peticiones por segundo, obteniéndose los siguientes

CAPITULO 3. Implementación y pruebas de la propuesta de solución

resultados:

Tabla 10: Resultados de las pruebas de carga y estrés (Elaboración propia)

Cant. de Hilos	#Muestras	Media	Mín.	Máx.	%Error	Rendimiento	Kb/s
700	1016	1433	42	3103	0.00	89.14	1003
1000	2400	1134	118	4187	0.04	78.25	1500
2000	2900	1012	193	4324	0.28	67.25	2100

Las pruebas realizadas muestran que el sistema es capaz de responder a 1016 peticiones de 700 usuarios conectados simultáneamente en un tiempo promedio de 1433 milisegundos (1.4 segundos aproximadamente) con 0 % de error y una velocidad de carga de 1003 milisegundos (1 segundo aproximadamente), esto evidencia que el sistema puede procesar la carga esperada para esta cantidad de usuarios.

Por otra parte, se realizaron 2400 peticiones iniciadas por 1000 usuarios y en este caso el sistema respondió en 1134 milisegundos (1.1 segundos aproximadamente) como tiempo promedio y una velocidad de carga de 1500 milisegundos (1.5 segundos aproximadamente). Esto demuestra que el sistema es capaz de procesar la carga esperada para este número de usuarios, aunque falló en un 0.04% de las peticiones.

Por último, y con el objetivo de analizar el comportamiento del sistema en condiciones extremas, se realizó una prueba de estrés para un total de 2000 usuarios conectados simultáneamente. En este caso, el sistema responde a las 2900 peticiones en un tiempo promedio de 1012 milisegundos (1 segundo aproximadamente), pero falla el 0.28 % de las peticiones, con una velocidad de carga de 2100 milisegundos (2.1 segundos aproximadamente).

El tiempo promedio de respuesta de aplicaciones de este tipo no debe superar los 3000 milisegundos (3 segundos). Teniendo en cuenta estos resultados se evidencia que el sistema es un 100% fiable.

3.7 Conclusiones del capítulo

En este capítulo se abordaron los elementos de la implementación de SAATI, así como las pruebas realizadas al mismo y los resultados obtenidos. La confección del diagrama de componentes permitió una

mejor comprensión de la estructura de los componentes de la propuesta de solución. El empleo de un estándar de codificación permitió garantizar la alta calidad, minimización de errores y mayor limpieza en el código fuente, permitiendo pueda ser mantenido fácilmente y reutilizado por otros desarrolladores que lo necesiten. La implementación de la propuesta de solución facilitó la obtención de una aplicación funcional lista para su uso. Por último, la detección temprana de errores, mediante la aplicación de pruebas permitió validar que el sistema funciona como inicialmente fue diseñado, que no se incurrió en la violación de ningún requisito y la implementación fue correcta.

CONCLUSIONES GENERALES

El presente Trabajo de Diploma finaliza con el desarrollo de un Sistema informático, el cual sirve de apoyo al proceso de enseñanza-aprendizaje de la asignatura Teleinformática en la carrera Ingeniería en Ciencias Informáticas. Una vez cumplidos los objetivos trazados en la investigación, se concluye lo siguiente:

- ❖ El sistema fue desarrollado como tecnología para el aprendizaje, proveyendo al estudiante una herramienta para facilitar la práctica en las actividades de auto preparación, y al profesor, para guiar, controlar y evaluar estas actividades.
- ❖ El uso de estrategias de gamificación permitió presentar los ejercicios de la asignatura Teleinformática de una manera más dinámica, interactiva, competitiva y motivadora para los estudiantes, así como incrementar la dificultad de los mismos de forma progresiva.
- ❖ La incorporación de un ranking y un premio simbólico agregó un elemento de competencia a la experiencia lúdica en el sistema.
- ❖ La combinación de conocimientos adquiridos de las distintas áreas del conocimiento como son la programación, ingeniería y gestión de software, bases de datos, entre otras permitió el análisis, diseño e implementación de la propuesta de solución.
- ❖ La validación a través del diseño y aplicación de pruebas, permitió comprobar el funcionamiento correcto de SAATI, a partir del cumplimiento de los requisitos definidos por el cliente.

RECOMENDACIONES

Para dar continuidad a la presente investigación, el autor recomienda:

- ❖ El despliegue de la aplicación en la UCI, para que pueda ser utilizado por los profesores y estudiantes del 3er año de la carrera Ingeniería en Ciencias Informáticas.
- ❖ La incorporación de otras asignaturas.
- ❖ La diversificación de los tipos de ejercicios que se muestran en la aplicación.

REFERENCIAS

- Abraira, J. A. 2016.** *Sistema interactivo-experimental para el proceso de enseñanza-aprendizaje de la Matemática Discreta en la carrera Ingeniería en Ciencias Informáticas.* 2016.
- Academic. 2012.** Enciclopedia Universal: . [En línea] 2012. [Citado el: 12 de abril de 2019.] http://enciclopedia_universal.esacademic.com/19671/Modelado_del_Software.
- Alegsa, L. 2019.** Definición de Sistema Informático. [En línea] 2019. [Citado el: 22 de enero de 2019.] http://www.alegsa.com.ar/Dic/sistema_informatico.php.
- Arteta, N.** Hipertextual. [En línea] [Citado el: 24 de enero de 2019.] <https://hipertextual.com/archivo/2015/01/que-es-gamificacion/>.
- Autores, Colectivo de. 2004.** *Pedagogía.* La Habana : Editorial Pueblo y Educación, 2004.
- Bermúdez, G. S., Jiménez, I. M., y Rodríguez, L. R. 2013.** Uso de Patrones de Diseño: Un caso Práctico. *Journal of Tropical Engineering* . 22.2, 2013.
- Cabello, Maria Victoria Nevado. 2010.** *Introducción a las Bases de Datos Relacionales.* Madrid : Vision Libros, 2010. ISBN: 978-84-9983-617-1.
- Canós, J. y Letelier, P. y Penadés, M. C. 2003.** *Metodologías ágiles en el desarrollo de software.* Valencia : s.n., 2003.
- Carreño, C. S. Gonzalez y A. M. 2015.** *Técnicas de gamificación aplicadas en la docencia de Ingeniería Informática. ReVision 8.1.* 2015.
- CMS. 2008.** Selecting a development approach. [En línea] 2008. [Citado el: 22 de enero de 2019.] <http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-InformationTechnology/XLC/Downloads/SelectingDevelopmentApproach.pdf>.
- Condori, J. L. 2012.** *Python - Django Framework de desarrollo web para perfeccionistas Basado en el Modelo MTV.* 2012.
- Cortéz M. I., Hernández C. R. y Cabrera M. M. 2017.** *Sistema informático para la administracion de riesgos en proyectos.* s.l. : Universidad y Sociedad, 2017.
- Delgado Fernández, Marianela y Solano González, Arlyne. 2009.** *ESTRATEGIAS DIDÁCTICAS*

REFERENCIAS

CREATIVAS EN ENTORNOS. Costa Rica : s.n., 2009. ISSN 1409-4703.

DiagramasUML. 2019. Diagrama de Componentes . [En línea] 2019. [Citado el: 25 de abril de 2019.] <https://diagramasuml.com/componentes/>.

Díaz, J.A. y Dosting, A. 2014. *Gestión de empresas de desarrollo de software a partir de aplicaciones de tipo código abierto*. 2014.

Diaz, V. M. 2015. *La gamificación educativa. Una alternativa para la enseñanza creativa*. . s.l. : Digital Education Review, 2015.

e-ABC. 2017. e-ABC. [En línea] 2017. [Citado el: 18 de enero de 2019.] <http://www.eabclearning.com/definicion-learning>.

2017. Ecured. [En línea] 28 de junio de 2017. https://www.ecured.cu/Requisitos_no_funcionales.

efdeportes. 2002. efdeportes. [En línea] 2002. [Citado el: 25 de mayo de 2019.] <https://www.efdeportes.com/efd47/iadov.htm>.

Escalante, L. C. 2016. *Aplicación de Patrones de Diseño para garantizar Alta Flexibilidad en el Software*. 2016.

Espinosa, R. S. 2016. *Presentación. Juegos digitales y gamificación aplicados en el ámbito de la educación*. 2016.

FOROUZAN, Behrouz A. 2007. *Transmisión de datos y redes de comunicaciones*. . Madrid, España : McGraw-Hill, 2007. ISBN:9788448156176.

Foundation, Apache Software. 2011. The Apache HTTP Server Project. [En línea] 2011. [Citado el: 23 de enero de 2019.] <http://httpd.apache.org/>.

Gauchat, J. D. 2014. *El gran libro de HTML5*. s.l. : Editorial Tomera Ltda, 2014.

GAVARRI, Sandra Lorena. 2016. *El aprendizaje de lenguas extranjeras mediado por las TIC: aprender Inglés con Duolingo*. s.l. : El Toldo de Astier, 2016.

Guaña-Moya, Edison Javier, Llumiquinga-Quispe, Sylvia del Rosario y Ortiz-Remache, Ketty Jadira. 2015. *Caracterización de entornos virtuales de enseñanza aprendizaje (EVEA) en la educación virtual*. Ecuador : Ciencias Holguín, 2015. ISSN 1027-2127.

REFERENCIAS

- Guerrero, C. A., y Johanna M. Suárez, L. E. 2013.** *Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web.* s.l. : Informacion tecnologica, 2013.
- Gutiérrez, E. 2009.** *Java Script: conceptos básicos y avanzados.* Barcelona, España : Informática Técnica, 2009.
- Imelda, Nandina y Martinez Carod. 2011.** *Ponderación de Requisitos de Software usando Técnicas Cognitivas y Orientación por Objetos.* Argentina : s.n., 2011.
- 2017.** Intercambios Virtuales. [En línea] 28 de junio de 2017. <http://www.intercambiosvirtuales.org/tag/wolfram-research-mathematica>.
- JetBrains. 2016.** JetBrains. [En línea] 2016. [Citado el: 19 de noviembre de 2018.] [https://www.jetbrains.com/pycharm/..](https://www.jetbrains.com/pycharm/)
- Larman., Craig. 2003.** *Modelo del Dominio.* s.l. : Prentice Hall, 2003.
- Martinez, Iker Ros. 2008.** *Moodle, la plataforma para la enseñanza y organizacion escolar.* s.l. : Ikastorratza, e-Revista de Didáctica 2, 2008. issn: 1988-5911.
- Martinez, Juan Carlos, Guzmán, Lorena Alonso y Alarcón, Victor Manuel: Gómez, Claudia. 2013.** *Diagramas de navegación en aplicaciones web.* s.l. : Vinculos, 2013.
- Martínez, R. 2010.** PostgreSQL. [En línea] 2010. [Citado el: 4 de diciembre de 2018.] http://www.postgresql.org.es/sobre_postgresql..
- Moreno Cadavid, Julián y Montoya Gómez, Luis F. 2015.** *Uso de un entorno virtual de aprendizaje ludificado como estrategia didáctica en un curso de pre-cálculo: Estudio de caso en la Universidad Nacional de Colombia.* Medellín, Colombia. : RISTI no.16 Porto dez., 2015. ISSN 1646-9895.
- Navarro, Ruben Edel. 2009.** *LAS NUEVAS TECNOLOGÍAS PARA EL APRENDIZAJE: ESTADO DEL ARTE.* Veracruz, Mexico : Editorial Pearson, 2009. ISBN: 978-970-26-1213-1.
- Ortiz, Kadir Hector. 2008.** *PLATAFORMA PARA EL CONTROL DEL USO DE SOFTWARES EDUCATIVOS.* Cienfuegos, Cuba : s.n., 2008.
- Paradigm, Visual. 2013.** *Visual Paradigm for UML.* 2013.

REFERENCIAS

- Park, D. y Stefănescu, A. y Roşu, G. 2015.** *KJS: A complete formal semantics of JavaScript*. s.l. : ACM SIGPLAN Notices., 2015.
- Prefacio, X. V. 2000.** *Parte 1: El Proceso Unificado de Desarrollo de Software*. 2000.
- Pressman, R. 2010.** *Ingeniería de Software, un enfoque práctico*. New York : s.n., 2010.
- Python-Django. Framework de desarrollo web para perfeccionistas basado en el modelo MTV.* **Condori Ayala, J. L. 2012.** 2012.
- Rivero, F. V. Sanchez y C. A. 2015.** *Diseño e implementación de una estrategia de gamificación en una plataforma virtual de educación*. s.l. : Fides et Ratio-Revista de Difusión cultural y científica de la Universidad La Salle en Bolivia., 2015.
- Sebastián, F. J. 2015.** *Desarrollo Web del Grupo Flulng adaptado a los nuevos estándares HTML5 y CSS3*. 2015.
- Serna, E. y Bedoya, i. A. y Lopez, K. 2015.** *Nivel de madurez y grado de satisfacción de las herramientas libres para pruebas funcionales*. s.l. : Scientia et technica, 155-161, 2015.
- Sevilla V. d, Fernandez M. A. y Díaz M. J. 2016.** *Introducción práctica a la programación con python*. 2016.
- Sommerville, I. 2005.** *Technology & Engineering*. 2005.
- Sommerville, I. 2011.** *Ingeniería de Software*. 2011.
- Soni, R. 2016.** *Introduction to Nginx Web Server*. s.l. : Nginx. Apress I, 1 -15., 2016.
- UCI. 2018.** UCI. [En línea] 2018. [Citado el: 28 de abril de 2019.] <https://www.uci.cu/estudios/pregrado>.
- Visconti, Marcello y Astudillo, Hernan. 2012.** *Fundamentos de Ingeniería de Software*. 2012.

ANEXOS

Anexo 1: Historias de Usuario

Tabla 11: Historia de usuario Registrar usuario (Elaboración propia)

Historia de Usuario	
Número: HU_1	Usuario: Usuario
Nombre de historia: Registrar usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo
Programador Responsable: Rafael Frances León	
Descripción: El usuario debe registrarse en el sistema mediante un formulario correspondiente, para poder acceder al sistema	

Tabla 12: Historia de usuario Autenticar usuario (Elaboración propia)

Historia de Usuario	
Número: HU_2	Usuario: Usuario
Nombre de historia: Autenticar usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo
Programador Responsable: Rafael Frances León	
Descripción: El usuario puede acceder al sistema luego de estar registrado, introduciendo en un pequeño formulario, su nombre de usuario y la contraseña	

Tabla 13: Historia de usuario Mostrar mundos (Elaboración propia)

Historia de Usuario	
Número: HU_3	Usuario: Estudiante
Nombre de historia: Mostrar mundos	

ANEXOS

Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Programador Responsable: Rafael Frances León	
Descripción: El estudiante autenticado puede ver los diversos mundos correspondientes a los temas de la asignatura Teleinformática, puede desplazarse entre los mundos y entrar en cualquiera de ellos para enfrentarse a los niveles que este contiene	

Tabla 14: Historia de usuario Mostrar niveles (Elaboración propia)

Historia de Usuario	
Número: HU_4	Usuario: Estudiante
Nombre de historia: Mostrar niveles	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Programador Responsable: Rafael Frances León	
Descripción: El estudiante puede ver los niveles que tiene un mundo, y comenzar a resolver los ejercicios que estos tienen, ordenadamente	

Tabla 15: Historia de usuario Mostrar preguntas (Elaboración propia)

Historia de Usuario	
Número: HU_5	Usuario: Estudiante
Nombre de historia: Mostrar preguntas	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Programador Responsable: Rafael Frances León	
Descripción: El estudiante puede ver las preguntas contenidas en un nivel, el sistema muestra las preguntas una a la vez con la opción de desplazarse por las preguntas con una paginación	

ANEXOS

Tabla 16: Historia de usuario Mostrar tabla de posiciones global (Elaboración propia)

Historia de Usuario	
Número: HU_6	Usuario: Estudiante
Nombre de historia: Mostrar tabla de posiciones global	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Programador Responsable: Rafael Frances León	
Descripción: El estudiante puede acceder a una tabla de posiciones donde se encuentran todos los estudiantes que han interactuado con el sistema, puede observar la puntuación de cada estudiante y seleccionar si desea visualizar la puntuación ordenada ascendente o descendientemente además puede realizar búsquedas dentro de la tabla	

Tabla 17: Historia de usuario Mostrar tabla de posiciones del mundo (Elaboración propia)

Historia de Usuario	
Número: HU_7	Usuario: Estudiante
Nombre de historia: Mostrar tabla de posiciones del mundo	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Programador Responsable: Rafael Frances León	
Descripción: Una vez dentro del mundo, el estudiante puede observar una tabla ordenada con los estudiantes que participan en un mundo y su respectiva puntuación	

Tabla 18: Historia de usuario Mostrar tabla de posiciones de nivel (Elaboración propia)

Historia de Usuario	
Número: HU_8	Usuario: Estudiante
Nombre de historia: Mostrar tabla de posiciones de nivel	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio

Programador Responsable: Rafael Frances León

Descripción: El estudiante puede observar una tabla ordenada con los estudiantes que participan en un nivel seleccionado y su respectiva puntuación

Tabla 19: Historia de usuario Evaluar respuesta (Elaboración propia)

Historia de Usuario	
Número: HU_9	Usuario: Estudiante
Nombre de historia: Evaluar respuesta	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Programador Responsable: Rafael Frances León	
Descripción: luego de terminar la pregunta el estudiante presiona el botón evaluar y el sistema califica su respuesta y le otorga la puntuación correspondiente	

Tabla 20: Historia de usuario Mostrar logros de nivel

Historia de Usuario	
Número: HU_10	Usuario: Estudiante
Nombre de historia: Mostrar logros del nivel	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Programador Responsable: Rafael Frances León	
Descripción: al concluir todas las preguntas de un nivel, se le muestra al estudiante un resumen de las preguntas resueltas por categoría, la puntuación y créditos alcanzados en dicho nivel y la bonificación por tiempo que obtiene.	

Tabla 21: Historia de usuario Mostrar puntuación acumulada

Historia de Usuario

ANEXOS

Número: HU_11	Usuario: Estudiante
Nombre de historia: Mostrar puntuación acumulada	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Programador Responsable: Rafael Frances León	
Descripción: cuando el estudiante se encuentra resolviendo un nivel, puede observar en una parte de la pantalla la puntuación acumulada que ha obtenido.	

Tabla 22: Historia de usuario Mostrar créditos acumulados (Elaboración propia)

Historia de Usuario	
Número: HU_12	Usuario: Estudiante
Nombre de historia: Mostrar créditos acumulados	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Programador Responsable: Rafael Frances León	
Descripción: cuando el estudiante se encuentra resolviendo un nivel, puede observar en una parte de la pantalla la cantidad de créditos acumulados que ha obtenido.	

Tabla 23: Historia de usuario Mostrar comodines disponibles (Elaboración propia)

Historia de Usuario	
Número: HU_13	Usuario: Estudiante
Nombre de historia: Mostrar comodines disponibles	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Programador Responsable: Rafael Frances León	
Descripción: cuando el estudiante se encuentra resolviendo un nivel, puede observar en una parte de la pantalla los comodines de los cuales dispone para utilizar durante el nivel.	

ANEXOS

Tabla 24: Historia de usuario Utilizar comodín (Elaboración propia)

Historia de Usuario	
Número: HU_14	Usuario: Estudiante
Nombre de historia: Utilizar comodín	
Prioridad en negocio: Media	Riesgo en desarrollo: Alto
Programador Responsable: Rafael Frances León	
Descripción: cuando el estudiante se encuentra resolviendo un nivel, puede hacer uso de los comodines adquiridos en la tienda para desbloquear algún obstáculo del nivel y continuar avanzando.	

Tabla 25: Historia de usuario Activar audio de fondo (Elaboración propia)

Historia de Usuario	
Número: HU_15	Usuario: Estudiante
Nombre de historia: Activar audio de fondo	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Programador Responsable: Rafael Frances León	
Descripción: el estudiante tiene la opción de activar un audio de fondo en el menú principal, que se reproduce en toda la aplicación.	

Tabla 26: Historia de usuario Desactivar audio de fondo (Elaboración propia)

Historia de Usuario	
Número: HU_16	Usuario: Estudiante
Nombre de historia: Desactivar audio de fondo	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo

ANEXOS

Programador Responsable: Rafael Frances León

Descripción: el estudiante tiene la opción de desactivar un audio de fondo en el menú principal, que se reproduce en toda la aplicación.

Anexo 2: Cuestionario para evaluar la satisfacción de SAATI

Lee con cuidado y responde:

1. ¿Considera usted que es posible obtener mejores resultados en la asignatura Teleinformática sin una adecuada motivación de los estudiantes?

Si_____ No_____ No sé_____

2. ¿Utilizaría usted la aplicación propuesta a la hora de estudiar y aprender Teleinformática?

Si_____ No_____ No sé_____

3. ¿Satisface sus necesidades, según el rol que desempeña, en la docencia con respecto a la asignatura Teleinformática, el sistema propuesto?

Si_____ No_____ No sé_____

4. ¿Qué es lo que más te ayuda a estudiar Teleinformática?

5. ¿Valora que te parece la aplicación propuesta?

ANEXOS

Anexo 3: Reportes de evaluación final de la asignatura Teleinformática

Tabla 27: Reporte de evaluación final Facultad 1. 2016-2017

Teleinformática											
Facultad	Grupo	Mat. Actual	2 Ptos	3 Ptos	4 Ptos	5 Ptos	Aprobados	% Aprobados	% Calidad	Promedio	NE
F1	1301	17	3	6	8	0	14	82.35	47.06	3.29	0
	1302	17	5	7	4	1	12	70.59	29.41	3.05	0
	1303	18	8	5	5	0	10	55.56	27.78	2.83	0
	14AA	1	1	0	0	0	0	0	0	2	0
	Total	53	17	18	17	1	36	67.92	33.96	3.03	0

Tabla 28: Reporte de evaluación final 2017-2018 (Elaboración propia)

Teleinformática											
Universidad	Facultad	Mat. Actual	2 Ptos	3 Ptos	4 Ptos	5 Ptos	Aprobados	% Aprobados	% Calidad	Promedio	NE
UCI	Facultad 1	114	25	60	19	10	89	78.07	25.44	3.12	0
	Facultad 2	84	17	54	6	7	67	79.76	15.48	3.03	0
	Facultad 3	124	26	60	25	12	97	78.23	30.08	3.18	1
	Facultad 4	69	25	30	12	2	44	63.77	20.29	2.86	0
	Facultad de Ciencias y Tecnologías Computacionales	86	29	24	27	6	57	66.28	38.37	3.11	0
	Total	477	122	228	89	37	354	74.21	26.47	3.08	1

Anexo 4: Acta de aceptación

	<h2>Acta de aceptación</h2>
<h3>ACTA DE ACEPTACIÓN</h3>	
<p>En cumplimiento del Convenio de colaboración establecido entre el Departamento de Programación de la Facultad 1 y el estudiante Rafael Angel Francés León de la Facultad 1, ambos de la Universidad de las Ciencias Informáticas, se hace entrega de los productos que se relacionan a continuación:</p>	
<ul style="list-style-type: none">• Aplicación web “SAATI. Sistema Informático de apoyo al aprendizaje de la asignatura Teleinformática”.	
<p>La Parte Cliente, luego de haber revisado los productos de trabajo determina que se Aceptan.</p>	
Entrega	Recibe
Nombre y apellidos: Rafael Angel Francés León	Nombre y apellidos: Ailyn Gutiérrez Ferrera
Cargo: Estudiante	Cargo: Jefe de Departamento
	
	Fecha: 28/05/2019