

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



TESIS DE GRADO

**“TUTOR WEB INTELIGENTE PARA LA ENSEÑANZA DE
ARMONÍA MUSICAL”**

PARA OPTAR EL TÍTULO DE LICENCIATURA EN INFORMÁTICA MENCIÓN:
INGENIERÍA DE SISTEMAS INFORMÁTICOS

POSTULANTE: IVAR RAÚL RODRÍGUEZ GUACHALLA

TUTOR METODOLÓGICO: M. Sc. ALDO RAMIRO VALDEZ ALVARADO

ASESORA: LIC. BRÍGIDA ALEXANDRA CARVAJAL BLANCO

LA PAZ – BOLIVIA

2015



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

ÍNDICE GENERAL

CAPÍTULO I	1
MARCO INTRODUCTORIO	1
1.1 INTRODUCCIÓN	1
1.2 ANTECEDENTES	2
1.3 PLANTEAMIENTO DEL PROBLEMA	5
1.3.1 PROBLEMA CENTRAL	5
1.3.2 PROBLEMAS SECUNDARIOS	6
1.4 DEFINICIÓN DE OBJETIVOS	6
1.4.1 OBJETIVO GENERAL	6
1.4.2 OBJETIVOS ESPECÍFICOS	6
1.5 HIPÓTESIS	7
1.5.1 OPERACIONALIZACIÓN DE VARIABLES	7
1.6 JUSTIFICACIÓN	8
1.6.1 ECONÓMICA	8
1.6.2 SOCIAL	8
1.6.3 CIENTÍFICA	9
1.7 ALCANCES Y LÍMITES	9
1.7.1 ALCANCES	9
1.7.1.1 TEMPORAL	9
1.7.1.2 ESPACIAL	9
1.7.2 LÍMITES	9
1.8 APORTES	10
1.8.1 PRACTICO	10
1.8.2 TEÓRICO	10
1.9 METODOLOGÍA	10
CAPÍTULO II	12
MARCO TEÓRICO	12
2.1 INGENIERÍA DE SOFTWARE EDUCATIVO	12
2.1.1 METODOLOGÍA DE DESARROLLO MeiSE: METODOLOGÍA DE INGENIERÍA DE SOFTWARE EDUCATIVO	14
2.1.1.1 FASE CONCEPTUAL	15

2.1.1.2	ANÁLISIS Y DISEÑO INICIAL	15
2.1.1.3	PLAN DE ITERACIONES	16
2.1.1.4	DISEÑO COMPUTACIONAL	18
2.1.1.5	DESARROLLO	18
2.1.1.6	FASE DE DESPLIEGUE	19
2.2	INGENIERÍA WEB	19
2.2.1	WEBML (WEB MODELING LANGUAGE)	20
2.3	SISTEMA TUTOR INTELIGENTE	21
2.3.1	TUTOR	21
2.3.2	CARACTERÍSTICAS Y DEFINICIÓN DE UN STI	22
2.3.3	ARQUITECTURA	24
2.3.3.1	MÓDULO DEL ESTUDIANTE	24
2.3.3.2	MÓDULO DEL DOMINIO	25
2.3.3.3	MÓDULO DE INTERFACE	25
2.3.3.4	MÓDULO DEL TUTOR	26
2.4	INTELIGENCIA ARTIFICIAL	26
2.4.1	AGENTE INTELIGENTE	27
2.4.1.1	CARACTERÍSTICAS DE LOS AGENTES INTELIGENTES	28
2.4.1.2	ESTRUCTURA DE UN AGENTE INTELIGENTE	29
2.4.1.3	AGENTE PEDAGÓGICO	30
2.5	PROCESO ENSEÑANZA-APRENDIZAJE	32
2.5.1	ELEMENTOS DEL PROCESO ENSEÑANZA-APRENDIZAJE	32
2.5.2	ESTILOS DE APRENDIZAJE	34
2.5.3	PLAN DE TRABAJO PARA CADA ESTILO DE APRENDIZAJE	35
2.6	ARMONÍA MUSICAL	36
2.6.1	CONTENIDO DE ARMONÍA MUSICAL	37
2.7	PLATAFORMA TECNOLÓGICA	38
	BOOTSTRAP 3	38
CAPÍTULO III		39
MARCO APLICATIVO		39
3.1	INTRODUCCIÓN	39
3.2	ETAPAS MeISE (Metodología de Ingeniería de Software Educativo)	40
3.2.1	ETAPA DE DEFINICIÓN	40
3.2.1.1	FASE CONCEPTUAL	40
3.2.1.2	ANÁLISIS Y DISEÑO INICIAL	62
3.2.1.3	FASE PLAN DE ITERACIONES	67

3.2.2	ETAPA DE DESARROLLO	68
3.2.2.1	DISEÑO COMPUTACIONAL	68
3.2.2.2	FASE DE DESARROLLO	74
3.2.2.3	FASE DE DESPLIEGUE	77
CAPÍTULO IV		81
PRUEBA DE HIPÓTESIS		81
4.1	FORMULACIÓN DE LA HIPÓTESIS	81
4.2	TAMAÑO DE LA MUESTRA	81
4.3	TEST DE CONOCIMIENTO DE ARMONÍA MUSICAL	82
4.4	PROCEDIMIENTO	82
4.5	REGLA DE DECISIÓN	86
CAPÍTULO V		87
CONCLUSIONES Y RECOMENDACIONES		87
5.1	CONCLUSIONES	87
5.2	RECOMENDACIONES	88
BIBLIOGRAFÍA		89

ÍNDICE DE TABLAS

Tabla 1.1 Operacionalización de Variables	9
Tabla 2.1: Actividades y Artefactos de la Fase Conceptual.	17
Tabla 2.2: Actividades y Artefactos de la Fase de Análisis y Diseño Inicial.	18
Tabla 2.3: Actividades y Artefactos de la Fase de Plan de Iteraciones.	19
Tabla 2.4: Actividades y Artefactos de la Fase de Diseño Computacional	19
Tabla 2.5: Actividades y Artefactos de la Fase de Desarrollo	20
Tabla 2.6: Actividades y Artefactos de la Fase de Despliegue	21
Tabla 2.7: Los cuatro estilos de aprendizaje (Gallego, Honey y Alonso)	36
Tabla 2.8: Plan de trabajo para cada estilo de aprendizaje	38
Tabla 2.9: Tabla de herramientas a utilizar en prototipo.	40
Tabla 3.1: Análisis PAMA del Agente Pedagógico	47
Tabla 3.2: Lista de Riesgos	50
Tabla 3.3: Equipo de Trabajo del proyecto	51
Tabla 3.4: Tabla de Programación de Actividades 1	51
Tabla 3.5: Tabla de Programación de Actividades 2	52
Tabla 3.6: Tabla de Programación de Actividades 3	52
Tabla 3.7: Tabla de Programación de Actividades 4	52
Tabla 3.8: Tabla de Programación de Actividades 5	53
Tabla 3.9: Tabla de Programación de Actividades 6	53
Tabla 3.10: Tabla de Programación de Actividades 7	53
Tabla 3.11: Tabla de Programación de Actividades 8	54
Tabla 3.12: Tabla de Programación de Actividades 9	54
Tabla 3.13: Tabla de Programación de Actividades 10	55
Tabla 3.14: Modelo de actores	55
Tabla 3.15: Plan de Trabajo por iteración.	70
Tabla 4.2: Resultados Test Sin Tutor	86
Tabla 4.3: Resultados del test a ambos grupos	87

ÍNDICE DE FIGURAS

Figura 2.1: Ciclo de vida de la Metodología	16
Figura 2.2: Arquitectura tradicional de un STI	26
Figura 2.3: Representación del concepto de Agente	29
Figura 2.4: Representación del concepto de Agente	32
Figura 2.5: Elementos del proceso Enseñanza-Aprendizaje	35
Figura 3.1: MeISE, Sistema Tutor Inteligente y Agentes Inteligentes	41
Figura 3.2: Modelo Instruccional	42
Figura 3.3: Casos de uso de Gestión de Estudiantes	56
Figura 3.4: Casos de uso de Temas	60
Figura 3.5: Casos de uso de Evaluaciones	62
Figura 3.6: Prototipo de interfaz de usuario	66
Figura 3.7: Diagrama de Navegación para el botón de Aprendizaje	67
Figura 3.8: Diagrama de Navegación para el botón de Ejercitar	67
Figura 3.9: ADV para el registro de estudiantes	68
Figura 3.10: ADV para el login de Usuarios	68
Figura 3.11: ADV para la interfaz de bienvenida	69
Figura 3.12: Modelo Entidad – Relación del Sistema	71
Figura 3.13: Diagrama de Clases	72
Figura 3.14: Diagrama de Secuencia de Registro de Estudiantes	73
Figura 3.15: Diagrama de Secuencia de Visualizar Temas	73
Figura 3.16: Diagrama de Secuencia Realizar Evaluación	73
Figura 3.17: Modelo de Navegación refinado	75
Figura 3.18: Interfaz de Usuario con diseño Responsive	76
Figura 3.19: Interfaz Registro de Estudiantes	77
Figura 3.20: Interfaz Vista de temas de acuerdo al avance del Estudiante	78
Figura 3.21: Interfaz Vista de contenido de los Temas	78
Figura 3.22: Interfaz de Evaluaciones	79
Figura 3.23: Medición de calidad	81
Figura 4.1: Test de conocimientos de armonía musical	85

A DIOS.
A MIS PADRES.
Y AMIGOS.

AGRADECIMIENTOS

A Dios que me dio la oportunidad de lograr esta meta.

A mis padres, Hortencia Guachalla y Cipriano Rodríguez por su apoyo incondicional en todo momento de mi vida y mas aún en esta etapa de mi formación profesional. De igual manera a mis queridos hermanos Einar y Gudnar por su colaboración constante hacia mí.

A mi Tutor M.Sc. Aldo Valdez Alvarado y asesora Lic. Brígida Carvajal Blanco quienes gracias a sus observaciones, recomendaciones y en especial por su tiempo se pudo concluir el documento.

A todos mis amigos con quienes compartí esta aventura universitaria, gracias por su apoyo y su sincera amistad, en especial a mi grupo de amigos más cercanos, a quienes considero como mis hermanos: Cristhian Poma, Royer Escobar, Marcelo Bayón, Imanol Villarreal, Mónica Aruni.

Un agradecimiento muy especial a Cristina Quisbert con quien compartí y compartiré gratos momentos de mi vida. Gracias por tu paciencia y cariño hacia mí.

Finalmente a la prestigiosa casa de estudios Universidad Mayor de San Andrés que me cobijó en sus aulas estos 5 años de mi vida y en especial a la carrera de Informática.

RESUMEN

En el presente trabajo de investigación se propone la construcción de un Tutor Web Inteligente para la enseñanza de la materia de armonía musical dictada a estudiantes de música superior. Los cuales tendrán una herramienta que les permita aprender la materia de forma interactiva.

Se utiliza la Metodología de Ingeniería de Software Educativo (MeIse) propuesta por Antonieta Abud; la estructura de un Sistema Tutor Inteligente que cuenta con cuatro módulos; y un agente inteligente pedagógico como parte del módulo Tutor del Sistema Tutor Inteligente.

Las herramientas principales para el desarrollo del Tutor son los siguientes: El framework de desarrollo web Django, el motor de base de datos PostgreSQL, y el framework frontend Bootstrap.

Cabe destacar que este trabajo se lo realiza, no con el fin de reemplazar al profesor o docente que imparte la materia, sino, con el de fortalecer más aun las clases y de alguna manera eliminar las “lagunas” que pueden crearse en los estudiantes cuando éstos no atienden al 100% la clase o por algún motivo faltan.

Palabras clave: Tutor Inteligente, Sistema Web, Armonía Musical, Música, software educativo

ABSTRACT

In the present research the construction of an Intelligent Web Tutor for teaching of musical harmony given to students in higher music is proposed. Which will have a tool that allows them to learn the material interactively.

Methodology Software Engineering Education (Meise) proposed by Antoinette Abud is used; the structure of an Intelligent Tutor System which has four modules; and a teaching intelligent agent as part of Tutor System Intelligent Tutor module.

The main tools for the development of Tutor are: The web framework Django, the engine PostgreSQL database, and the frontend Bootstrap framework.

Note that this work is done, not to replace the teacher or teacher who teaches the subject but with even more to strengthen the classes and somehow eliminate the "gaps" that can be created in students when they do not attend the class or 100% for some reason missing.

Keywords: Intelligent Tutor, Web System, Harmony Musical, Music, educational software

CAPÍTULO I

MARCO INTRODUCTORIO

1.1 INTRODUCCIÓN

Hoy en día es evidente el incremento de la utilización de las tecnologías de la información y la comunicación en los procesos de enseñanza/aprendizaje. Las más demandadas en este ámbito son las basadas en Inteligencia Artificial (IA) como los Sistemas Tutores Inteligentes (STI), los cuales buscan emular el comportamiento de un tutor humano, es decir, poder impartir el conocimiento usando alguna forma de inteligencia para poder asistir y guiar al estudiante en su proceso de aprendizaje. Estos sistemas han demostrado sus beneficios dentro las instituciones educativas como son flexibilidad, enseñanza asíncrona y a distancia.

Un STI es un sistema de enseñanza o instrucción asistida por computadora (IAC), que utiliza técnicas de IA, principalmente para representar el conocimiento y dirigir una estrategia de enseñanza. Es capaz de comportarse como un experto, tanto en el dominio del conocimiento que enseña, mostrando al estudiante cómo aplicar dicho conocimiento, como en el dominio pedagógico, donde es capaz de diagnosticar la situación en la que se encuentra el estudiante y de acuerdo a ello ofrecer una acción o solución que le permita progresar al estudiante. (JIMÉNEZ et.al., 2005)

Por otro lado el internet, con la aparición de la Web 2.0 en el año 2004, se ha convertido en una herramienta de vital importancia no solo a nivel personal, más aún a nivel empresarial y mundial, tanto así que sería difícil imaginarse la vida sin él. La Web 2.0 nos permite la utilización y observación de páginas web con características de sociabilidad entre usuarios e interacción con la propia página, diferenciándose así con las páginas estáticas anteriores a la Web 2.0 donde el usuario era un sujeto pasivo que recibía la información o la publicaba, sin

que existieran demasiadas posibilidades para que se generara la interacción. Por lo tanto, la Web 2.0, está formada por las varias plataformas de interacción de usuario y páginas web tales como: Blogger, las redes sociales, como Facebook o Twitter, los servicios conocidos como wikis (Wikipedia) , Portales de alojamiento de fotos, audio o vídeos (Flickr, YouTube), entre otros. En este sentido, aprovechando las características tan positivas que nos ofrece la Web 2.0 expuestas anteriormente y tomando en cuenta la escasez de tutores inteligentes aplicado al área de la música, peor aún en temas un poco avanzados como es la armonía musical, se propone en el presente trabajo el diseño y desarrollo de un Sistema Web Tutor Inteligente para la enseñanza de armonía musical.

Cabe destacar que este trabajo se lo realiza, no con el fin de reemplazar al profesor o docente que imparte la materia, sino, con el de fortalecer más aun las clases y de alguna manera eliminar las “lagunas” que pueden crearse en los estudiantes cuando éstos no atienden al 100% la clase o por algún motivo faltan.

1.2 ANTECEDENTES

La Inteligencia Artificial, una rama de la computación que aparece a finales de la década de los 50, y sus técnicas han progresado a un ritmo apresurado debido a sus diversas aplicaciones en varios sectores sociales y áreas del conocimiento. La Instrucción Asistida por Computadoras es uno de ellos, recibió un gran impacto con estas técnicas, ya que se facilitan los procesos de enseñanza y se garantiza mayor eficiencia en los mismos.

Es en 1970 que Carbonell da la iniciativa de aplicar las técnicas de la Inteligencia Artificial a los sistemas IAC; a él se le atribuye la creación del primer Tutor Inteligente: el SCHOLAR (un tutor inteligente para la enseñanza de la geografía de América del Sur), aunque el término de “Sistema Tutor Inteligente” es definido posteriormente por Brown. (DUARTE, 2014).

Posterior al SCHOLAR de Carbonell, aparecieron muchos STIs más, nombraremos algunos, como el WHY de Stevens, Collins en 1977 aplicado a la Meteorología el cual utiliza como

característica el dialogo socrático; BUGGY de Brown en 1981 aplicado a la Aritmética; MENO de Wolf en 1984 aplicado a la Programación; ACTP de Anderson 1984 que es un Tutor del lenguaje de programación LISP utilizando un modelado cognitivo; SIERRA de VanLehn en 1987 aplicado a la Aritmética utilizando Predicción de errores; SHERLOCK de Lesgold, Katz en 1991 aplicado a la electrónica con la característica de un aprendizaje cognitivo; CIRCSIM de Evens, Michael, Rovick en 1996 aplicado a la Medicina, entre otros. Así surgieron los primeros STIs, cada vez más aplicando las técnicas de la Inteligencia Artificial consolidando así el concepto de Sistemas Tutores Inteligentes.

Con el paso del tiempo y la gran evolución de la tecnología sobre todo en campos como realidad virtual, multimedia, los sistemas tutores inteligentes se tornan hoy por hoy como herramientas de gran utilidad e importancia en la educación.

En 2011, Luis Von Ahn y Severin Hacker, lanzan un sitio web destinado al aprendizaje gratuito de idiomas llamado DUOLINGO que en la actualidad más de 100 millones de personas, a nivel mundial, lo utilizan. La interfaz de Duolingo permite entender cómo cada uno de los estudiantes progresa a través del curso, y analizar los resultados en gran escala. Esto a su vez permite hacer pequeños cambios a los desarrolladores del sistema y medir la velocidad en la que la gente adquiere información. Actualmente se están usando algoritmos de “aprendizaje de máquinas” (machine learning en inglés) para personalizar el plan de estudios de cada estudiante y optimizar el aprendizaje a nivel individual. Con Duolingo aprender un idioma es mucho más fácil y divertido que con los métodos tradicionales ya que utiliza tecnologías multimedia de modo que aprender idiomas puede ser visto como un juego. Un estudio independiente mostró que 34 horas de Duolingo son equivalentes a un semestre universitario completo de enseñanza de idiomas. (VON AHN, 2015).

Particularmente en la carrera de Informática de la Universidad Mayor de San Andrés, existen bastantes trabajos de investigación sobre Sistemas Tutores Inteligentes, lo que demuestra su crecida utilización en la actualidad. Mencionaremos algunos de ellos:

- Título: “SISTEMA TUTOR INTELIGENTE PARA LA ENSEÑANZA EN NIÑOS CON DISCAPACIDAD INTELECTUAL”

Autor: Gary Americo Mamani Villca

Año: 2010

Se plantea un modelo de Tutor Inteligente para niños con discapacidad intelectual, de manera que se pueda fortalecer sus conocimientos básicos adquiridos en el área de matemática y lenguaje. El tutor solo es manipulado por el mouse de modo que el niño no se complique en el manejo del mismo. En el diseño se utilizan agentes pedagógicos, el modelo de 4 etapas del modelo de enseñanza de Allesi y Tollip y el modelo en cascada con prototipado.

- Título: “TUTOR INTELIGENTE DE ESCRITURA CREATIVA (NIVEL PRIMARIO)”

Autor: Lourdes Beltran Colque

Año: 2008

Tutor que coadyuva al educando en el proceso de aprendizaje del lenguaje. Este trabajo toma en cuenta a niños que cursan el cuarto grado de primaria tanto de la ciudad de La Paz como El Alto. La metodología a emplearse fue Científica. En la prueba de hipótesis se utilizó dos tests un Pre-test y un Post-test, con una muestra poblacional de 12 personas, y se promediaron ambos casos y haciendo una simple comparación de esos resultados se puede verificar un incremento en el aprendizaje.

- Título: “TUTOR INTELIGENTE PARA EL APRENDIZAJE DE LA LECTURA DE PARTITURAS MUSICALES”

Autor: Santos Jahuirá Mendoza

Año: 2013

Presenta el desarrollo de un tutor inteligente para el aprendizaje de la lectura musical, orientado a estudiantes de últimos cursos de colegio e institutos de enseñanza musical utilizando la metodología MeIse (Metodología Ingeniería de Software Educativo) de Antonieta Abud y agentes pedagógicos. Para la prueba de hipótesis utiliza la prueba t-student con una población de 20 estudiantes.

1.3 PLANTEAMIENTO DEL PROBLEMA

1.3.1 PROBLEMA CENTRAL

Así como en todas las áreas del conocimiento la música tiene grados de dificultad, mientras se va avanzando en su estudio se torna un poco más complicado de entender, motivo por el cual muchos estudiantes pasan muchas dificultades en el transcurso de la materia. La armonía musical es un curso avanzado de música para el cual se requiere de base sólida en teoría musical, solfeo, rítmica, conocer diversos instrumentos y hasta saber obligatoriamente tocar el piano (por lo menos un nivel básico) porque a diferencia de los demás instrumentos en el piano se pueden tocar varias notas simultáneamente (acordes) lo cual es fundamental en el estudio de armonía musical.

“La música es una fuente generadora de riqueza y forma parte de las industrias culturales y de entretenimiento; si el músico no es consciente de ello, está fuera de la foto. El músico actual necesita de herramientas que lo hagan competitivo en un mundo globalizado” (COCK, 2014)

Surge entonces el problema de investigación:

¿Cómo apoyar al proceso de aprendizaje en estudiantes de la materia de armonía musical?

1.3.2 PROBLEMAS SECUNDARIOS

- Muchos estudiantes llegan a la materia con dificultades en la identificación de melodías escrita en partitura, lo cual es muy importante para el músico ya que le permite al mismo afinar el oído, desarrollar el oído interno y desarrollar un pensamiento musical. Es importante el desarrollo de estas capacidades en armonía musical, caso contrario el estudiante pasará por muchos obstáculos en el transcurso de la materia.
- Existen estudiantes que no interpretan de manera correcta ritmos avanzados, esto le impide al estudiante identificar melodías en partitura correctamente.
- El plan de trabajo de clase del profesor no se adecúa al estilo de aprendizaje de todos los estudiantes, ya que no es el mismo en todos, lo cual genera un desequilibrio de aprendizaje en el curso.
- Los estudiantes tienen pocas alternativas a la hora de querer ejercitar lo aprendido, lo cual le impedirá al estudiante un rendimiento satisfactorio en la materia.
- El exceso de teoría en la enseñanza hace que el estudiante se aburra del estudio de armonía musical.

1.4 DEFINICIÓN DE OBJETIVOS

1.4.1 OBJETIVO GENERAL

Diseñar e implementar un Tutor Web Inteligente que apoye al proceso de enseñanza en la materia de armonía musical utilizando tecnologías multimedia.

1.4.2 OBJETIVOS ESPECÍFICOS

- Brindar al estudiante un banco de melodías asociadas a su escritura en un pentagrama para su posterior identificación en varios niveles de dificultad.

- Facilitar al estudiante un banco de ritmos para su posterior identificación en varios niveles de dificultad.
- Aplicar un test, a cada estudiante, para determinar su estilo de aprendizaje y de acuerdo a ello orientar el tutor adaptado al estudiante y guardar cada avance del mismo.
- Proporcionar al tutor inteligente de ejemplos y ejercicios audiovisuales, clasificados por tema, de manera que el estudiante use el sentido del oído para la resolución de los mismos.
- Minimizar el contenido teórico del tutor inteligente, sin que ésta pierda su esencia, y dar más énfasis en la práctica del estudiante.

1.5 HIPÓTESIS

El Tutor Web Inteligente para la enseñanza de Armonía Musical utilizando tecnologías multimedia mejora el rendimiento educativo en estudiantes de la materia de armonía musical en un 15%.

1.5.1 OPERACIONALIZACIÓN DE VARIABLES

VARIABLE	INDICADORES	DEFINICIÓN
DEPENDIENTE: El rendimiento educativo en estudiantes de la materia de Armonía Musical mejora en un 15%.	Nivel de aceptación de los estudiantes. Aprendizaje. Calificación o nota obtenida en la materia. Repetitividad en la materia.	Capacidad, logro de los objetivos y aprendizajes que posee el alumno
INDEPENDIENTE: El Tutor Web Inteligente	Test de usabilidad. Formas de evaluación. Modelo educativo. Pruebas de Calidad	Herramienta tecnológica de apoyo en la educación que actúa como un tutor o profesor particular, utiliza como recurso principal

1.6 JUSTIFICACIÓN

1.6.1 ECONÓMICA

En el ámbito económico el Tutor Inteligente al ser implementado en la web y de acceso gratuito ahorrará a los usuarios costos de compra de textos, libros, por otro lado también en la instrucción en alguna institución musical, aunque siempre es recomendable la supervisión de un docente quien despeja cualquier duda que se pueda generar en el estudiante en su proceso de aprendizaje.

Al utilizar tecnologías multimedia el usuario estudiante evitará la compra de material adicional al tutor como ser CDs, DVDs, otros. Tan solo se necesitará acceso a internet y una salida de audio en su equipo (parlantes, audífonos).

Por otro lado para la implementación del prototipo, se lo realizará con herramientas de software libre y gratuitas como son PostgreSQL, Django, Bootstrap. Por lo que no se realizarán gastos económicos considerables.

1.6.2 SOCIAL

El Tutor Inteligente será una herramienta orientada especialmente para estudiantes de música de cualquier institución superior de música, de igual manera a los docentes de música superior para así darle mayor dinamismo a la cátedra y finalmente toda aquella persona que desee aprender armonía musical o desee ampliar su conocimiento en música en el caso de los músicos aficionados.

Los estudiantes de música experimentarán una motivación hacia el estudio del mismo con mayor énfasis, también a expertos en el tema quienes podrán aportar en el mejoramiento del sistema y la utilización del mismo.

1.6.3 CIENTÍFICA

Se apoyará e incentivará al desarrollo de tutores inteligentes para más áreas del conocimiento y también la creación de herramientas de desarrollo especialmente para el área de la música. De igual manera aumentará el interés sobre la utilización de tecnologías multimedia en el desarrollo de sistemas tutores inteligentes ya que le da mayor dinamismo al mismo.

1.7 ALCANCES Y LÍMITES

1.7.1 ALCANCES

1.7.1.1 TEMPORAL

Esta investigación está destinada a personas de toda edad que deseen aprender o reforzar sus conocimientos en armonía musical, más aún a estudiantes de instituciones superiores de música, cabe aclarar que, para estudiar armonía musical es necesario previos conocimientos en teoría musical como ser: las notas musicales, figuras musicales, solfeo, rítmica, lectura de partituras musicales, resolver dictados rítmicos y melódicos, entre otros.

1.7.1.2 ESPACIAL

Esta investigación está destinada especialmente para estudiantes de armonía musical en instituciones superiores de música. Particularmente el caso de estudio será en: Escuela Nacional de Música “Luis Felipe Arce” en el curso de 1ro Superior.

1.7.2 LÍMITES

- Se contemplará la Armonía Clásica y no así Armonía Moderna, cabe destacar que la diferencia entre ambas radica en el estilo de música en que se lo aplica, Clásica en la música clásica y la Moderna en estilos como el Jazz, Bossa, Pop, Blues.
- El idioma del Tutor Inteligente será únicamente el español.

- Existen limitaciones naturales como son la ceguera y la sordera los cuales impedirán a estas personas la usabilidad del tutor inteligente.
- El usuario deberá tener conocimientos básicos de teoría musical.

1.8 APORTES

1.8.1 PRACTICO

El aporte de la investigación es un prototipo de Tutor Inteligente de acceso vía web utilizando tecnologías multimedia para la enseñanza de armonía musical, el cual coadyuvará en el proceso de enseñanza donde están involucrados estudiantes y docentes.

1.8.2 TEÓRICO

En el desarrollo del tutor inteligente se utilizará la metodología MeISE (Metodología de Ingeniería de Software Educativo propuesta por M.Sc. Antonieta Abud), nos apoyaremos con algunos diagramas de WebML y para fortalecer las decisiones del Tutor se utilizará agentes pedagógicos.

1.9 METODOLOGÍA

La metodología de investigación utilizada será el Método Científico el cual es capaz de proporcionarnos una respuesta a nuestra interrogante que es nuestra hipótesis planteada, su proceso, según Francis Bacon, consta de 6 pasos:

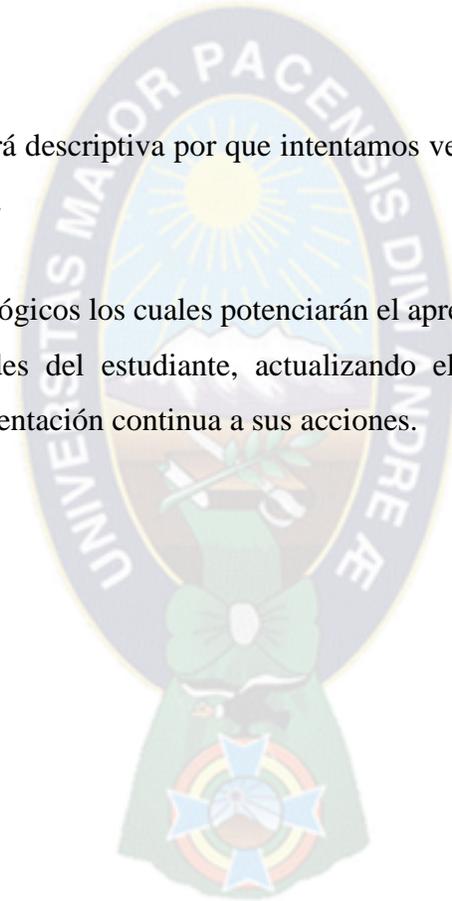
- *Observación:* Se realiza a través de la percepción de nuestros sentidos (vista, oído, tacto, olfato, gusto).
- *Inducción:* Se plantea un problema de investigación de acuerdo a la observación realizada.
- *Hipótesis:* Se plantea una suposición de acuerdo a los hechos observados.

- *Probar la hipótesis:* Procesos mediante la experimentación para tratar de comprobar la hipótesis.
- *Demostración o refutación de la hipótesis:* Se muestra si la hipótesis es posible o no.
- *Conclusiones:* De acuerdo a los pasos anteriores se escribe una conclusión.

Para la construcción del modelo y prototipo se utilizará la metodología de Ingeniería de Software Educativo (MeISE) de Antonieta Abud Figueroa, apoyado con WebML para el modelado Web del tutor.

El tipo de investigación será descriptiva por que intentamos verificar con datos estadísticos nuestra hipótesis planteada.

Se utilizarán agentes pedagógicos los cuales potenciarán el aprendizaje de modo que el tutor se adapte a las necesidades del estudiante, actualizando el estado actual del entorno, proporcionando una realimentación continua a sus acciones.



2.1 INGENIERÍA DE SOFTWARE EDUCATIVO

El software ha asistido a la educación de forma paralela a su evolución y a la del hardware; en principio mediante la digitalización de lecciones que los docentes impartían a estudiantes, valiéndose de los recursos multimedia que ofrece el computador, y que promueven, incentivan el placer y la curiosidad por el estudio de temas presentados en soportes multimedia. El desarrollo de programas informáticos, constituye un proceso, siendo la ingeniería del software la disciplina que comprende todos los aspectos de su producción. A través de la ingeniería del software se establecen prácticas efectivas para desarrollar y entregar un software adaptado al propósito que impulsa su desarrollo; esta utilidad se verá reflejada a través de atributos como la funcionalidad y el rendimiento que ofrezca al usuario, por otra parte, el software también debe ser confiable, fácil de mantener y de utilizar. (MARCANO, 2014)

El proceso de producción de software incluye un conjunto de actividades que conllevan al producto final. Se pueden definir cuatro actividades fundamentales: la especificación del software, para identificar el alcance del producto a producir; el desarrollo, donde se diseña el programa; la validación, que consiste en revisar el software para asegurar su funcionamiento en correspondencia con las exigencias de su desarrollo; y la evolución, en donde se modifica el software para actualizarlo, adaptándolo a los posibles cambios que se requieran para garantizar su funcionamiento. Sin embargo, según sea la naturaleza del software, se deben incorporar particulares procesos de desarrollo, que se ajusten a determinadas especificaciones que deben plantearse en medio de la tarea de diseño, de tal forma que se pueda crear una arquitectura de información adecuada; tal es el caso del software educativo,

cuyo contenido es netamente didáctico, dinámico, dirigido docentes y estudiantes en medio del proceso de enseñanza aprendizaje. Este tipo de software amerita un proceso de desarrollo que apuntale hacia lo evolutivo, de forma tal que las funciones de especificación de requisitos, el desarrollo, así como la validación, se entrelacen; bajo este concepto, al emprender el proceso de producción de software, se desarrolla un sistema inicial y se refina para adoptar en su totalidad las directrices de sus usuarios, en este caso en particular se refiere a docentes y estudiantes. (SOMMERVILLE, 2007)

A la hora de construir un software educativo nos topamos con el problema de qué metodología seguir, de modo tal que nos asegure la calidad del mismo. Se requiere incluir en su diseño criterios que favorezcan la comprensión del contenido por parte del alumno; es decir, deben apoyarse en bases psicopedagógicas sobre el aprendizaje, así como en los principios básicos de la ingeniería de software que permitan concretar el desarrollo de la aplicación en forma exitosa.

Las metodologías convencionales de ingeniería de software generalmente abarcan actividades para la obtención de los requisitos, el diseño del sistema (diseño preliminar y diseño detallado), la construcción, las pruebas, la instalación y el mantenimiento del producto de software, todas enfocadas a atender los aspectos técnicos del producto y no se ocupan de los aspectos de la calidad didáctica, por lo que es necesario adaptarlos para que incluyan actividades orientadas a atender las características didácticas. (ABUD, 2009)

Maria Antonieta Abud nos propone una metodología de desarrollo de software educativo con un enfoque iterativo e incluye aspectos computacionales, pedagógicos y de comunicación, llamado MeISE, el cual se utilizará en el desarrollo de nuestro Tutor Web Inteligente.

2.1.1 METODOLOGÍA DE DESARROLLO MeISE: METODOLOGÍA DE INGENIERÍA DE SOFTWARE EDUCATIVO

La Metodología de Ingeniería de Software Educativo MeISE propone un ciclo de vida dividido en dos etapas. En la primera etapa se contempla la definición de requisitos y el análisis y diseño preliminar, durante los cuales se determinan en forma global las características que se pretende alcanzar con el producto, los requisitos pedagógicos, de comunicación y la arquitectura sobre la cual se construirá el software, y se termina con un plan de iteraciones las cuales se programan teniendo cuidado de que el producto que se libera al término de cada una está didácticamente completo, es decir que cubre completamente algunos de los objetivos didácticos del software. Una vez establecidos estos lineamientos, inicia la segunda etapa, en la cual se procede a desarrollar el producto, de modo que el equipo toma cada iteración, la diseña, la construye, la prueba y la implementa, evaluando al final la conveniencia de proseguir con subsecuentes iteraciones hasta obtener un producto completo.

Las fases propuestas para la etapa de definición son: la fase conceptual, durante la cual se identifican los requerimiento del sistema, se conforma el equipo de trabajo y se elabora el plan de desarrollo; la fase de análisis y diseño inicial, en la que se propone la arquitectura que servirá de base para la solución del problema y se establecen las características pedagógicas y de comunicación que regirán el desarrollo del software; finalmente la fase de plan de iteraciones, en la cual se divide el proyecto en partes funcionales que permitan mejor control en su desarrollo.

En la etapa de desarrollo se tienen: la fase de diseño computacional, en la que se realizará un diseño computacional detallado de un incremento específico del software; la fase de desarrollo, durante la cual se implementa la arquitectura en forma incremental (iteración por iteración); y la fase de despliegue, donde se realiza la transición del producto ejecutable al usuario final.

Estas tres últimas etapas se repiten iterativamente para cada incremento del software.

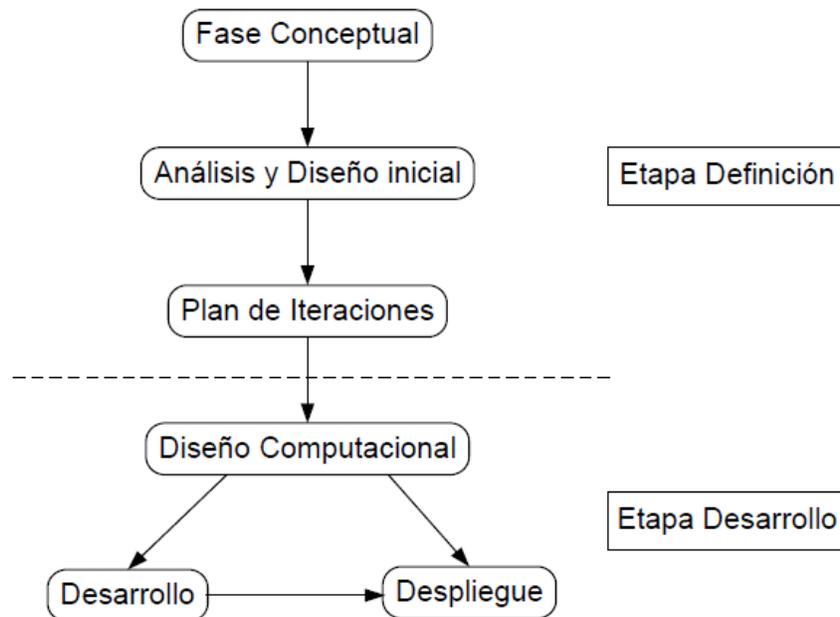


Figura 2.1: Ciclo de vida de la Metodología

Fuente: (ABUD, 2009)

3.1.1.1 FASE CONCEPTUAL

Esta etapa inicia con una investigación sobre los requerimientos que se cubrirán con el producto a desarrollar, delimitando su alcance. Se desarrolla el plan del proyecto, se evalúan riesgos y se establecen los criterios de éxito. En la tabla 2.1 se muestran las actividades a realizar y los artefactos que se generan en esta fase.

3.1.1.2 ANÁLISIS Y DISEÑO INICIAL

En la fase de análisis y diseño inicial se analiza el dominio del problema y se establece la arquitectura del sistema. En este punto se describen a detalle los requisitos del Software y las características educativas y de comunicación que el producto debe contemplar. En la tabla 2.2 se detallan estas actividades.

ACTIVIDAD	ARTEFACTO
Analizar las necesidades educativas	<u>Modelo instruccional</u> (incluye temática a atender, objetivos, conocimientos previos, fuentes de información, modelo educativo a utilizar, elementos de motivación y formas de evaluación) <u>Glosario</u> (descripción de los términos que pueden causar confusión o duda)
Revisar alternativas de solución	<u>Estudio de alternativas</u> (establece las diferentes alternativas que se tienen para el desarrollo del software, se determina el tipo de modelo educativo y se justifica la elección)
Elaborar un estudio de riesgos	<u>Lista de riesgos</u> (establece los riesgos relativos al desarrollo y a los aspectos pedagógicos y la forma de atenderlos)
Conformar del equipo de trabajo y el plan inicial de desarrollo	<u>Plan Inicial</u> (se conforma el equipo de trabajo, se elabora la programación de actividades, se asignan responsables a cada una y se determinan los tiempos estimados para llevarlas a cabo)
Identificar la funcionalidad que se pretende alcanzar con el software	<u>Modelo de actores</u> (identifica los tipos de usuario que utilizarán el software y describe sus características) <u>Modelo de casos de uso</u> (establece un modelo general de las funciones que cubrirá el sistema a través de diagramas de casos de uso y su especificación)
Establecer los criterios de medición de calidad del proceso, considerando aspectos tanto técnicos como pedagógicos	<u>Modelo de aceptación</u> (incluye las características mínimas que deben cumplirse para que el producto se acepte)

Tabla 2.1: Actividades y Artefactos de la Fase Conceptual.

Fuente: (ABUD, 2009)

3.1.1.3 PLAN DE ITERACIONES

Una vez identificados los requisitos a cubrir con el software se procede a analizar cuántos subproductos funcionales pueden producirse de modo que se puedan liberar partes operativas del sistema final, con el objetivo de llevar un mejor control en el desarrollo. Una vez identificados los incrementos se priorizan y se colocan con mayor prioridad aquellos que cubren los conocimientos base. En la tabla 2.3 se muestran los resultados de esta fase.

ACTIVIDAD	ARTEFACTO
Identificar los requisitos funcionales y no funcionales que se cubrirán con el software	<u>Modelo de requisitos</u> (Se determinan los requisitos que debe cumplir el software en cuanto a funcionalidad, comunicación, interfaz y docencia.)
Establecer la arquitectura del software	<u>Descripción de la arquitectura</u> (establecer la arquitectura base sobre la cual se desarrollará el software; se debe considerar que dicha arquitectura sea capaz de atender adecuadamente las tareas de aprendizaje que se van a manejar)
Elaborar el diseño educativo	<u>Modelo educativo</u> (Se definen el objetivo terminal y los subobjetivos, y en base a éstos se establecen las tareas de aprendizaje apegadas al tipo de modelo educativo)
Elaborar el diseño de comunicación general del producto	<u>Modelo de interfaz</u> (diseño de las zonas de comunicación y pantallas que se seguirán a lo largo del desarrollo) <u>Modelo de navegación</u> (diseño de los caminos de navegación generales que se presentarán al usuario) <u>Prototipo de la interfaz de usuario</u> (establecer las plantillas de diseño que se seguirán a lo largo del desarrollo)

Tabla 2.2: Actividades y Artefactos de la Fase de Análisis y Diseño Inicial.

Fuente: (ABUD, 2009)

ACTIVIDAD	ARTEFACTO
Diseñar las iteraciones de forma que las versiones ejecutables cubran objetivos didácticos bien planeados, de acuerdo a la secuencia de temas.	<u>Plan de iteraciones</u> (dividir el desarrollo en iteraciones, cuidando de que cada iteración cubre requisitos y objetivos educativos completos)
Priorizar las iteraciones, de modo que las que contienen conocimientos básicos que se requieren como base para aprendizajes posteriores se ejecuten primero.	<u>Lista de Iteraciones Priorizadas</u> (ordenar las iteraciones programadas de forma lógica de acuerdo a los contenidos)

Tabla 2.3: Actividades y Artefactos de la Fase de Plan de Iteraciones.

Fuente: (ABUD, 2009)

3.1.1.4 DISEÑO COMPUTACIONAL

Para cada iteración se debe elaborar el diseño computacional detallado, de modo que sirva de base para el desarrollo. Los artefactos y actividades propios de este paso se muestran en la tabla 2.4.

ACTIVIDAD	ARTEFACTO
Realizar el plan de trabajo de la iteración	<u>Plan de trabajo</u> (se determinan las tareas que se realizarán en el diseño del software, se asignan a los miembros del equipo y se calendarizan)
Elaborar el diseño computacional	<u>Modelo de diseño</u> (detallar el diseño a través de diagramas de clases y secuencia, incluir la descripción de clases y métodos; para los desarrollos que requieren bases de datos, incluir la especificación de diccionario de datos y diagramas entidad relación)
Refinar el diseño de navegación	<u>Modelo de navegación refinado</u> (diseñar los caminos de navegación específicos para la iteración en desarrollo)
Refinar prototipo de interfaz	<u>Modelo de interfaz usuario</u> (desarrollar las pantallas específicas para los elementos de la iteración en desarrollo)

Tabla 2.4: Actividades y Artefactos de la Fase de Diseño Computacional

Fuente: (ABUD, 2009)

3.1.1.5 DESARROLLO

Se desarrolla en esta fase el producto, implementando la arquitectura de manera que se obtiene una versión del software lista para que sea utilizada por los usuarios finales. En la tabla 2.5 se incluyen sus elementos a detalle.

ACTIVIDAD	ARTEFACTO
Desarrollar los componentes	<u>Modelo de desarrollo</u> (Determinar los componentes a desarrollar y documentarlos.)
Probar los componentes	<u>Modelo de pruebas unitarias</u> (Realizar pruebas de los componentes contra los criterios previamente establecidos. Estas pruebas deben incluir las pruebas del diseño instruccional)
Integrar al desarrollo previo	<u>Modelo de Integración</u> (establecer un plan para incorporar el nuevo desarrollo a la liberación previa si es el caso)
Realizar pruebas de integración	<u>Pruebas de integración</u> (realizar pruebas para verificar que la incorporación del nuevo incremento no ha inducido fallas al sistema)

Tabla 2.5: Actividades y Artefactos de la Fase de Desarrollo

Fuente: (ABUD, 2009)

3.1.1.6 FASE DE DESPLIEGUE

En la fase de despliegue se realiza la transición del producto a los usuarios. Aquí se culmina con una versión ejecutable del producto. Las actividades y artefactos de esta fase se describen en la tabla 2.6. Al finalizar esta etapa se evalúa la conveniencia de continuar los desarrollos, y en su caso regresar a la etapa de diseño computacional para continuar con el siguiente incremento.

ACTIVIDAD	ARTEFACTO
Entregar producto al usuario	<u>Producto</u> (Se debe entregar el producto debidamente empacado, etiquetado y con información sobre su contenido, aplicación, población objetivo y requerimientos de instalación) <u>Manual de Usuario</u> (Debe contener información detallada de cómo utilizar el software) <u>Manual de Instalación</u> (información de los requerimientos para su funcionamiento y procedimiento de instalación)
Evaluar las características de calidad y satisfacción de los usuarios	<u>Aceptación del Usuario</u> (realizar pruebas con los usuarios finales y comprobar su grado de satisfacción y efectividad del software)
Evaluar la conveniencia de continuar con otro incremento al producto	<u>Evaluación de despliegue</u> (analizar los resultados de la prueba de aceptación del usuario y determinar si es conveniente seguir con otra iteración)

Tabla 2.6: Actividades y Artefactos de la Fase de Despliegue

Fuente: (ABUD, 2009)

3.2 INGENIERÍA WEB

“Es el proceso utilizado para crear, implantar y mantener aplicaciones y sistemas Web de alta calidad.” (RANGEL et. al, 2001)

Esta breve definición nos lleva a abordar un aspecto clave de cualquier proyecto como es determinar qué tipo de proceso es más adecuado en función de las características del mismo. El desarrollo de aplicaciones Web posee determinadas características que lo hacen diferente

del desarrollo de aplicaciones o software tradicional y sistemas de información. La ingeniería de la Web es multidisciplinar.

Los sistemas y aplicaciones (WebApps) basados en Web hacen posible que una población extensa de usuarios finales dispongan de una gran variedad de contenido y funcionalidad. La ingeniería web no es un clónico perfecto de la ingeniería del software, pero toma prestado muchos de los conceptos y principios básicos de la ingeniería del software, dando importancia a las mismas actividades técnicas y de gestión. Existen diferencias sutiles en la forma en que se llevan a cabo estas actividades, pero la filosofía primordial es idéntica dado para el desarrollo de un sistema basado en computadora. (PRESSMAN, 2002)

3.2.1 WEBML (WEB MODELING LANGUAGE)

Es una notación para especificar complejos sitios Web en el ámbito conceptual, que permite apoyar las actividades del diseño de estos, a partir de su descripción desde distintos puntos de vista como son el conceptual, el navegacional y el de presentación, entre otros.

WebML consta de 4 modelos:

- Modelo de Datos
- Modelo de Hipertexto
- Modelo de Gestión de Contenido
- Modelo de Presentación

De los cuales sólo utilizaremos dos modelos: Modelo de Datos y de Hipertexto

MODELO DE DATOS

El modelo de datos representa las diferentes tablas de datos y sus relaciones que son necesarias para una aplicación Web concreta. Se pueden utilizar:

- Diagramas de Entidad-Relación (E-R) que muestran todas las tablas, los diferentes campos de cada tabla, y las relaciones entre ellas.
- Diagramas UML de clases que pueden representar la misma información que un diagrama de Entidad - Relación (por lo que puede usarse de manera equivalente), e incluso información adicional sobre el modelo de datos

MODELO DE HIPERTEXTO

Cada hipertexto describe una vista del sitio.

- Modelo de navegación. Representa los enlaces entre las diferentes páginas y sus elementos de contenido.

3.3 SISTEMA TUTOR INTELIGENTE

3.3.1 TUTOR

Es una figura académica fundamental en el proceso formativo, gracias al conocimiento individual que tiene del educando. Desarrolla tres cometidos básicos:

- a) Orienta y asesora al alumno en la elaboración de su plan de trabajo personal y la selección de sus actividades académicas; planifica, coordina, motiva y potencia el trabajo del educando, evitándole obstáculos y desarrollando sus capacidades intelectuales y de investigación a lo largo de todo el proceso formativo.
- b) Ejerce una acción docente, a fin de desarrollar la pericia y sirve como modelo para la adquisición del conocimiento experto por el educando. Transmite conocimientos, estrategias para abordar problemas, enseña procedimientos, criterios y métodos, favorece el desarrollo del proyecto de investigación del alumno, así como las actitudes y valores necesarios para el trabajo académico.

- c) Socializa al alumno y/o lo integra al grupo de investigación y a los medios académicos, conduce al educando a trabajar con otros tutores organizados en una red tutorial, lo prepara para el trabajo en equipo y la vida académica.

(SANDOVAL, 2014)

3.3.2 CARACTERÍSTICAS Y DEFINICIÓN DE UN STI

Guardia Robles resume un conjunto de características que deben cumplir todos los Sistemas Tutores Inteligentes (STI):

- Deben ser “inteligentes” en comparación con los sistemas tradicionales de instrucción por computadora (CAI), siendo el diferencial de inteligencia los métodos de la rama de la Inteligencia Artificial.
- Deben poseer la capacidad tanto para resolver el problema que se le presenta a un estudiante como también la capacidad de explicar cómo lo resolvió.
- Como en los CAI tradicionales, permiten una mayor individualización en la instrucción, llegando más lejos, a través del entendimiento de las metas y creencias del estudiante.
- Se usan técnicas de Inteligencia Artificial para planeación, optimización y búsquedas, dejando que el sistema decida el orden de presentación del contenido al alumno.
- La interacción puede ser muy variada en un STI: desde sistemas pasivos (que esperan para que el alumno realice una acción), hasta los que constantemente presentan nueva información (tutor oportunista), con casos intermedios en los que se enseña un concepto en un momento determinado o solo cuando el alumno lo pide.
- Utilizan nuevas tecnologías, con los ejemplos de interfaces orientadas a la utilización de multimedia y del WWW.
- No basta con indicarle un error al estudiante, el sistema debe hacer hipótesis basadas en el historial de errores del alumno y detectar la fuente del problema.

Con estas consideraciones, Robles, presenta una definición para los tutores inteligentes: “Un Sistema Tutor Inteligente es un sistema de enseñanza asistida por computadora, que utiliza técnicas de Inteligencia Artificial, principalmente para representar el conocimiento y dirigir una estrategia de enseñanza; y es capaz de comportarse como un experto, tanto en el dominio del conocimiento que enseña (mostrando al alumno cómo aplicar dicho conocimiento), como en el dominio pedagógico, donde es capaz de diagnosticar la situación en la que se encuentra el estudiante y de acuerdo a ello ofrecer una acción o solución que le permita progresar en el aprendizaje.” (ROBLES, 1993)

Villareal, por otra parte, plantea que los Sistemas Tutores Inteligentes simulan a un tutor autoritario que posee una estrategia de enseñanza de los conceptos del dominio del tipo uno a uno. Además es un experto en un dominio de conocimiento determinado y actúa como guía, tutor o entrenador. Este tutor debe poder adaptarse a las necesidades, que surgen a lo largo de la interacción en una sesión de tutelado, del estudiante alumno. (VILLAREAL, 2003)

Los primeros STI basados en ambientes Web utilizaban una interface soportada por cualquier explorador que les proporcionaba la ventaja de convertirlos en multiplataforma y no se requería ninguna instalación de componentes en el servidor en el que se debía utilizar el STI. Pero, esta aproximación estaba limitada a las características intrínsecas de la red Internet y de los protocolos que en ésta se desarrollan. Actualmente, gracias a las herramientas que nos permite utilizar la Web 2.0 podemos diseñar STI más poderosos que los primeros basados en Web.

En los últimos años el Internet se transformó no sólo en una fuente de intercambio de información, sino también en una fuente alternativa y el medio para realizar un sinnúmero de acciones. La educación a distancia no se quedó atrás en el uso de esta nueva tecnología y se integró a Internet como un medio para mejorar la calidad educativa y el perfeccionamiento a

distancia, donde convergen un gran número de estudiantes de diferentes edades y de todos los perfiles socio-culturales. (LAGE y CATALDI, 2014)

La arquitectura y diseño de los STI, entonces, debe soportar el procesamiento distribuido, para poder utilizar la red de una manera eficiente, ya sea esta la Internet o cualquier otra red. Pasemos entonces a definir lo que es un Sistema Tutor Inteligente, su arquitectura y componentes.

3.3.3 ARQUITECTURA

Los STI tienen como principal objetivo impartir la enseñanza de un contenido dado un dominio en la forma más adecuada a las necesidades individuales del alumno. Estos sistemas se basan en una arquitectura compuesta por tres grandes módulos: el módulo del tutor, el módulo del alumno y el módulo del dominio (VILLARREAL et al., 2001). Podría agregarse un cuarto módulo denominado el módulo de interface (GONZALES, 2004).

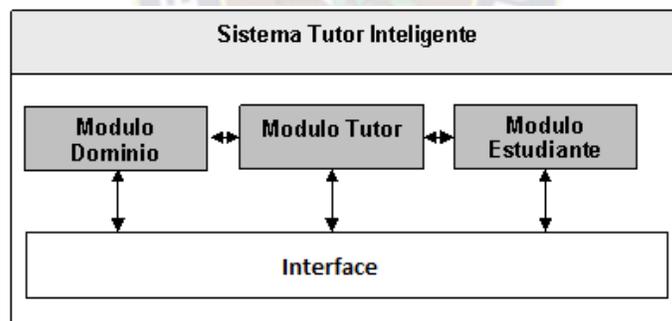


Figura 2.2: Arquitectura tradicional de un STI
Fuente: (CATALDI, 2006)

3.3.3.1 MÓDULO DEL ESTUDIANTE

La función central del módulo del estudiante, para Sierra se puede resumir en el siguiente párrafo: “Para realizar inferencias acerca de lo que el estudiante conoce, es necesario interpretar o hacer presunciones en base a las acciones del estudiante. Estas interpretaciones constituyen la parte central en el desarrollo del modelo del estudiante en el diseño de un

Sistema Tutor Inteligente.” (SIERRA et al., 2003) .Por lo tanto, es importante conocer el estado y el estilo de aprendizaje de cada uno de los alumnos para que el módulo tutor pueda tomar las decisiones pedagógicas correctas. Ésta es una de las salidas más importantes del módulo del estudiante y una de las entradas del módulo tutor. El módulo tutor, luego, realimentará las técnicas utilizadas, los temas expuestos en el dominio y los resultados obtenidos para que se pueda actualizar la representación del estado del conocimiento del alumno.

3.3.3.2 MÓDULO DEL DOMINIO

Para poder construir el modelo de un proceso, debe ser posible descomponerlo en sus partes constitutivas. Es decir, el mecanismo a ser modelado debe tener partes identificables en las cuales pueda ser descompuesto. Por lo tanto para el caso del módulo del dominio, se debe descomponer el conocimiento de los temas para los cuales el sistema deberá ser experto. Entre ellos se encuentra el conocimiento dependiente del dominio, compuesto por las definiciones, los conceptos fundamentales y las agrupaciones de conceptos que forman los temas. También existe el conocimiento independiente del dominio, el cual se compondrá de los diferentes parámetros del sistema que se requieren para el correcto funcionamiento del mismo y por último se encuentran los elementos didácticos que se utilizan para mejorar y facilitar la experiencia del proceso enseñanza/aprendizaje.

3.3.3.3 MÓDULO DE INTERFACE

La interface se encarga de presentar el contenido de la sesión de tutoría en forma homogénea a lo largo de toda la curricula, presentando en tiempo y forma los elementos pedagógicos que utiliza cada sesión, como pueden ser ejercitación, material multimedia, texto, etc. Si bien la interface existe en todos los Sistemas Tutor Inteligente, no se la considera un módulo fundamental, ya que se centra más en el aspecto gráfico y no en los contenidos.

Se puede generar una interface que responda a la adecuación del usuario, donde éste establezca parámetros, como la letra, el tamaño de la letra, el color de fondo, etc., para crear un ambiente de trabajo donde se sienta más cómodo y mejorando la sesión educativa. El diseño es muy importante para la interface, ya que depende en gran parte de los usuarios.

3.3.3.4 MÓDULO DEL TUTOR

La idea principal de este módulo es la de utilizar la computadora como “herramienta de aprendizaje”, buscando automatizar el proceso de enseñanza [VILLAREAL et al., 2001]. Esto motivó los primeros esfuerzos en la década del sesenta en brindar soporte a la los docentes. Desde aquella época hasta la actualidad se han realizado cambios en el paradigma utilizado en la educación, partiendo de un estilo fuertemente conductista hasta el día de hoy donde el concepto de “impartir” conocimientos es reemplazado por el de “descubrir” o “construir” el conocimiento de forma significativa.

Por lo tanto, el software reproducía la forma de enseñanza que prevalecía en el sistema educacional de esa época. Una parte del contenido del dominio era presentada en una o más pantallas (orientadas solo a caracteres, sin gráficos u otros elementos didácticos) y la interacción se limitaba a presionar la tecla Enter para cambiar de pantalla. Esta forma de enseñanza recibió el nombre de Computer Assisted Instruction (CAI).

3.4 INTELIGENCIA ARTIFICIAL

A McCarthy se le debe el concepto de inteligencia artificial (IA), que él acuñó en 1956 durante la conferencia de Dartmouth, el cual decía que la Inteligencia Artificial es: “Es la ciencia e ingenio de hacer máquinas inteligentes, especialmente programas de cómputo inteligentes” (MCCARTHY, 1956)

Ponce (2010) para definir lo que es la Inteligencia Artificial, primeramente se plantea la pregunta: “¿Qué es la Inteligencia?”. Destaca algunas propiedades generales que presenta la inteligencia humana, como ser la habilidad de enfrentar nuevas situaciones, la habilidad de resolver problemas, de responder preguntas, elaborar planes, entre muchos. Por otro lado considera a la IA como un dialecto simbólico constituido por cadenas de caracteres que representan conceptos del mundo real, basándose en que el hombre desde sus inicios representaba el mundo real mediante símbolos, los cuales constituyen la base del lenguaje humano.

Considerando lo anterior, Ponce nos brinda la siguiente definición: “La IA es una rama de las ciencias computacionales que se ocupa de los símbolos y métodos no algorítmicos para la resolución de problemas.” (PONCE, 2010)

3.4.1 AGENTE INTELIGENTE

Un agente inteligente, es una entidad capaz de percibir su entorno, procesar tales percepciones y responder o actuar en su entorno de manera racional, es decir, de manera correcta y tendiendo a maximizar un resultado esperado. Es capaz de percibir su medioambiente con la ayuda de sensores y actuar en ese medio utilizando actuadores (elementos que reaccionan a un estímulo realizando una acción). (RUSSELL Y NORVIG, 2004)

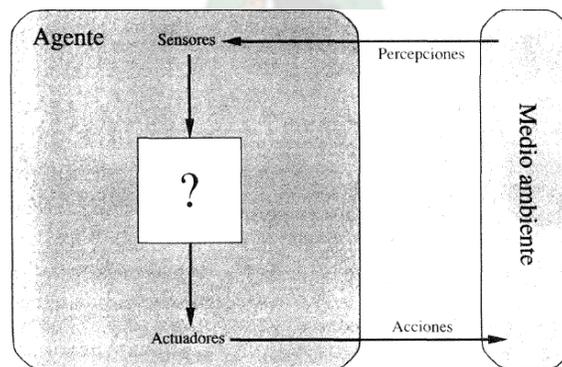


Figura 2.3: Representación del concepto de Agente
Fuente: (RUSSELL Y NORVIG, 2004)

En este contexto la racionalidad es la característica que posee una elección de ser correcta, más específicamente, de tender a maximizar un resultado esperado. Este concepto de racionalidad es más general y por ello más adecuado que inteligencia (la cual sugiere entendimiento) para describir el comportamiento de los agentes inteligentes. Por este motivo es mayor el consenso en llamarlos agentes racionales (RUSSELL Y NORVIG, 2004)

Podemos considerar, entonces, a humanos como agentes inteligentes ya que se amolda al concepto. Sin embargo, a lo largo del desarrollo de la presente tesis se hablará específicamente de agentes software que básicamente son programas con capacidad de actuar de manera independiente de acuerdo a su entorno, con el fin de lograr los objetivos para los cuales fueron diseñados.

3.4.1.1 CARACTERÍSTICAS DE LOS AGENTES INTELIGENTES

Wooldridge (1999) describe características básicas de los agentes inteligentes:

- **Reactividad:** Mediante la cual los agentes, al percibir su entorno, tienden a reaccionar apropiadamente a los cambios que ocurren en él.
- **Pro-actividad:** Se fundamenta en la idea de que los agentes no simplemente actúan en respuesta a su entorno, sino que son capaces de exhibir un comportamiento orientado al objetivo, pero de una manera en la cual son ellos los que toman la iniciativa.
- **Habilidad social:** Se dice que los agentes requieren de comunicarse, e interactuar con otros agentes, para la consecución de sus objetivos, incluso en algunos casos deben ser capaces de negociar, y cooperar con ellos.

Las cualidades que tendría un agente inteligente ideal, según James A. Hendler, serían:

- **Comunicativo:** el agente debe entender las necesidades, objetivos y preferencias del usuario para que éste pueda realizar su función correctamente. El agente también debe poder comunicarse con el entorno mediante representaciones compartidas de conocimiento.
- **Capaz:** El agente no sólo debe proporcionar una información, sino también un servicio, es decir, debe tener capacidad para hacer cosas. Por ejemplo, si se precisa un artículo de revista y ésta es de pago, el agente debe ser capaz de encontrar el artículo, informar del precio, dar el número de tarjeta de crédito, etc.
- **Autónomo:** el agente, además de comunicarse, debe poder interactuar con el entorno, tomando decisiones y actuando por sí solo, limitando sus acciones según el nivel de autonomía permitida por el usuario.
- **Adaptativo:** Debe ser capaz de aprender del entorno: preferencias de usuarios, fuentes de información y de otros agentes.

En ambos casos los autores coinciden casi en todo cuando hablan de las características que un agente debería tener, pero podríamos quedarnos con las características que propone Hendler ya que éste nos da una visión más clara de lo que debería considerarse a la hora de diseñar un agente, incluye además, el concepto de “Autonomía”.

3.4.1.2 ESTRUCTURA DE UN AGENTE INTELIGENTE

Un agente inteligente tiene una estructura básica que consiste en un programa que se ejecuta sobre una arquitectura.

Agente=Programa agente + Arquitectura

- El programa agente será una función que implementará la transformación de secuencias de percepciones en acciones.
- La arquitectura será una computadora que se ocupará de que las percepciones lleguen al programa y las acciones lleguen a los efectores.

La IA se ocupa del diseño del programa agente. Antes de diseñar un programa agente es necesario conocer los distintos elementos que caracterizan al agente:

- Percepciones posibles.
- Acciones posibles.
- Medida de desempeño u objetivos a lograr.
- Tipo de entorno en el que va a operar

El programa agente tiene la siguiente estructura:

```

Función ESQUELETO-AGENTE(percepción) return acción
  Static: memoria

memoria ← ACTUALIZACION-MEMORIA(memoria,percepción)
acción ← ESCOGER LA MEJOR OPCION(memoria)
memoria ← ACTUALIZACION-MEMORIA(memoria,acción)

return acción

```

Figura 2.4: Representación del concepto de Agente
Fuente: (RUSSELL Y NORVIG, 2004)

3.4.1.3 AGENTE PEDAGÓGICO

Un Agente Pedagógico (AP) es una aplicación de agentes de software en entornos de aprendizaje de algún dominio de conocimientos específico. Esencialmente, un AP actúa como la personificación de un tutor en un entorno virtual de aprendizaje, entorno que puede

ser textual, bidimensional o tridimensional (de escritorio o de inmersión). La personificación de este tutor puede o no ser humana, pero resulta vital que posea características antropomorfas que permitan la comunicación verbal y no-verbal entre el tutor y el estudiante, de manera de motivar a éste, de capturar su atención y de facilitar la comprensión de conceptos, logrando así el cumplimiento de los objetivos educacionales.

Se considera, además, que un AP es inteligente en el mismo contexto de los sistemas tutoriales inteligentes: por la utilización de técnicas de Inteligencia Artificial (IA) para la representación de conocimiento y la explicación de su razonamiento. (TRONCOSO, 2005)

Los AP heredan todas las dificultades de implementación tanto de los agentes como del software educativo. Si además se utiliza una representación animada para mostrarlo, aparecen problemas nuevos. En primer lugar, los agentes deben mostrar un comportamiento coherente, coordinando su comportamiento con el de otros agentes, y respondiendo de forma lógica a los estímulos de su entorno, incluyendo dentro de éstos a las acciones del usuario. Además, necesitan poseer el conocimiento sobre el dominio que el estudiante está aprendiendo. En general los agentes comunes tienen cierto grado de inteligencia que les permite desenvolverse en su entorno para conseguir sus objetivos. En el caso de los agentes pedagógicos esa inteligencia no consiste en poder resolver los ejercicios que deben solucionar los estudiantes, sino ser capaz de explicar cómo se resuelven, dando consejos y ayuda contextualizada. Esto requiere una profunda comprensión de las relaciones entre cada una de las acciones necesarias para solucionar el problema. (JOHNSON, 1998)

Resumiendo lo anteriormente expuesto sobre los Agentes Pedagógicos podemos mencionar lo siguiente:

- Un AP es un Agente Inteligente aplicado a entornos de aprendizaje de algún área del conocimiento.
- Gracias a los AP los CAI (Instrucción Asistida por Computador) pasarían a ser ITS (Sistema Tutor Inteligente).

- Un AP puede tener una representación animada el cual debería responder de forma lógica a los estímulos de su entorno, incluyendo dentro de éstos a las acciones del usuario.

3.5 PROCESO ENSEÑANZA-APRENDIZAJE

Se aborda el proceso mediante el cual se comunican o transmiten conocimientos especiales o generales sobre una materia, sus dimensiones en el fenómeno del rendimiento académico a partir de los factores que determinan su comportamiento.

El proceso enseñanza-aprendizaje constituye un verdadero par dialéctico en el cual el primer componente debe organizarse y desarrollarse de manera tal que facilite la apropiación del conocimiento de la realidad objetiva que, en su interacción con un sustrato material neuronal, asentado en el subsistema nervioso central del individuo, permitirá que en el menor tiempo y con el mayor grado de eficiencia y eficacia posibles, el establecimiento de los engramas sensoriales, aspectos intelectivos y motores necesarios para que el reflejo se materialice y concrete. (ALFONSO, 2003)

3.5.1 ELEMENTOS DEL PROCESO ENSEÑANZA-APRENDIZAJE

El concepto del proceso enseñanza-aprendizaje implica que hay un sujeto que conoce (el que puede enseñar), y otro que desconoce (el que puede aprender).

- El que puede enseñar, quiere enseñar y sabe enseñar (EL PROFESOR)
- El que puede aprender quiere y sabe aprender (EL ALUMNO o ESTUDIANTE).

Ha de existir pues una disposición por parte de alumno y profesor. Aparte de estos agentes, están los contenidos, esto es, lo que se quiere enseñar o aprender (ELEMENTOS CURRICULARES) y los procedimientos o instrumentos para enseñarlos o aprenderlos (MEDIOS). Cuando se enseña algo es para conseguir alguna meta (OBJETIVOS). Por otro

lado, el acto de enseñar y aprender acontece en un marco determinado por ciertas condiciones físicas, sociales y culturales (CONTEXTO). (ESCUADERO, 1999)

En la figura 2.5 se ilustra lo planteado por Escudero sobre los elementos del proceso Enseñanza – Aprendizaje.

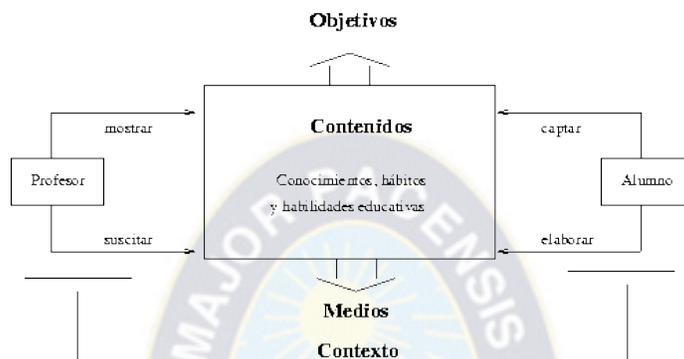


Figura 2.5: Elementos del proceso Enseñanza-Aprendizaje
Fuente: (ESCUADERO, 1999)

De acuerdo con lo expuesto, podemos considerar que el proceso de enseñar es el acto mediante el cual el profesor muestra o suscita contenidos educativos (conocimientos, hábitos, habilidades) a un alumno, a través de unos medios, en función de unos objetivos y dentro de un contexto. El proceso de aprender es el proceso complementario de enseñar. Aprender es el acto por el cual un alumno intenta captar y elaborar los contenidos expuestos por el profesor, o por cualquier otra fuente de información. Él lo alcanza a través de unos medios (técnicas de estudio o de trabajo intelectual). Este proceso de aprendizaje es realizado en función de unos objetivos, que pueden o no identificarse con los del profesor y se lleva a cabo dentro de un determinado contexto.

Cada persona presenta una manera diferente de pensar y aprender, es decir, un estilo particular de aprendizaje. Por este motivo, en cualquier ámbito en el que se realice un aprendizaje, y no sólo en el plano educativo, existe un gran interés por conocer la forma en

que cada individuo lo realiza. En esta línea, el estudio de los estilos de aprendizaje ha alcanzado una gran relevancia (HONEY et. al., 2003)

3.5.2 ESTILOS DE APRENDIZAJE

“Los estilos de aprendizaje son los rasgos cognitivos, afectivos y fisiológicos que sirven como indicadores relativamente estables, de cómo los discentes perciben, interaccionan y responden a sus ambientes de aprendizaje.” (HONEY et. al., 2003)

Se distinguen cuatro estilos: activo, reflexivo, teórico y pragmático. Para estos autores el estilo puede variar según la situación de aprendizaje y argumentan que hay individuos que presentan cualidades mixtas, esto es, la unión de varios estilos.

La siguiente tabla sintetiza las principales características que tiene los estudiantes según sea su estilo de aprendizaje:

ESTILO	CARACTERÍSTICAS
ACTIVO	Se implican en las nuevas experiencias. Se crecen ante los desafíos. Disfrutan participando en tareas grupales No les agradan los plazos largos
REFLEXIVO	Observan las experiencias desde diversas perspectivas. Son prudentes a la hora de tomar decisiones. Escuchan a los demás y no intervienen hasta que se hacen dueños de la situación.
TEÓRICO	Adaptan las observaciones dentro de teorías lógicas y complejas. Tienden a ser perfeccionistas. Buscan la racionalidad y la objetividad.
PRAGMÁTICO	Les gusta la aplicación práctica de las ideas. Ven los aspectos positivos de las nuevas ideas y las experimentan en cuanto pueden. Son realistas y actúan con rapidez.

Tabla 2.7: Los cuatro estilos de aprendizaje (Gallego, Honey y Alonso)
Fuente: (HONEY et. al.; 2009)

3.5.3 PLAN DE TRABAJO PARA CADA ESTILO DE APRENDIZAJE

A la hora de diseñar actividades para cada estilo de aprendizaje hay que tener en cuenta las siguientes consideraciones:

En el estilo activo, los alumnos tienden a implicarse íntegramente y sin prejuicios en experiencias novedosas y actuales que signifiquen retos en la realización y consecución de objetivos. Las actividades cuyo proceso dura mucho tiempo terminan por cansarles y prefieren aquellas tareas que no requieran largos plazos de ejecución. Disfrutan con el trabajo en equipo siendo ellos el centro.

En el estilo reflexivo, los alumnos recogen todos los datos posibles, y, tras un minucioso análisis, toman una decisión, e incluso no dudan en retroceder para volver a repensar sobre ella. Son prudentes, les gusta estudiar todas las facetas de una cuestión y considerar todas las posibles implicaciones derivadas antes llevar su gestión. No son partidarios de participar activamente en las reuniones, prefieren mantenerse a la expectativa observando y analizando las conductas y expresiones de los demás.

En el estilo teórico, los alumnos abordan los problemas de manera vertical y por fases lógicas y no se dan por satisfechos hasta que estiman que han llegado a la perfección o a ser el mejor. Ofrecen resistencia a trabajar en grupo, a no ser que consideren que los componentes sean de su mismo nivel intelectual. Tienden a ser perfeccionistas y no se encuentran satisfechos cuando no existe organización o los elementos no se articulan según la lógica racional. Se interesan por todo lo relacionado con sistemas de pensamiento, modelos teóricos, principios generales y mapas conceptuales.

En el estilo pragmático, los alumnos son inquietos, les gusta actuar y manipular rápidamente con aquellos proyectos o actividades que les atraen. Se inquietan ante discursos teóricos y exposiciones magistrales que no van acompañados de demostraciones o aplicaciones. Se

impacientan en los debates y discusiones teóricos de larga duración donde no aprecian nada tangible. (HERNÁNDEZ et. al., 2009)

ESTILO	ACTIVIDADES
ACTIVO	<ul style="list-style-type: none"> - Participación en clases en las que primen las actividades prácticas. - Consulta de bibliografía. - Asistencia a clases con un enfoque lúdico - Empleo de las TIC (vídeo, audio, fotografía, internet, aplicaciones informáticas, etc.) para la realización de Tareas. - Exámenes tipo test, estudio grupal; para pruebas de desarrollo, estudio individual. - Puesta en práctica de lo aprendido en clase. - Interacción profesor-alumno en clases dinámicas.
REFLEXIVO	<ul style="list-style-type: none"> - Asistencia a clase donde el profesor ponga variados ejemplos que permitan la reflexión. - Trabajos con TIC y búsqueda bibliográfica. - Búsqueda en internet de información útil y relevante. - Estudio de forma individual para realizar el repaso con otros compañeros. - Empleo de las TIC como complemento de los libros. - Memorización mediante técnicas nemotécnicas: reescribir lo leído y analizarlo varias veces; leer en voz alta repetidamente lo que hay que estudiar; hacer frases con palabras difíciles y acrósticos. - Reflexión sobre todo lo aprendido.
TEÓRICO	<ul style="list-style-type: none"> - Estudio individual para concentrarse mejor. - Resolución de problemas y de ejercicios conflictivos. - Realización de actividades concretas y claras, evitando el exceso de información redundante. - Participación en actividades que permitan la búsqueda de información en internet, más que en la biblioteca. - Utilización de plataformas educativas online, como Moodle.
PRAGMÁTICO	<ul style="list-style-type: none"> - Utilización de repeticiones para memorizar. - Estudio con resúmenes y esquemas claros elaborados por ellos mismos. - Actividades motivadoras y variadas, que les animen a estudiar. - Lectura de libros de forma voluntaria. - Trabajo individual y voluntario. - Realización de actividades que relacionen la teoría con la práctica.

Tabla 2.8: Plan de trabajo para cada estilo de aprendizaje

Fuente: (MOYA et. al.; 2009)

3.6 ARMONÍA MUSICAL

Estudia la asociación de varios sonidos musicales simultáneos para la formación de acordes y su disposición en sucesiones naturales y ordenadas. La Armonía es una parte de la materia de composición musical, definida por Schoenberg, como la enseñanza de los sonidos

simultáneos (acordes) y de sus posibilidades de encadenamiento, teniendo en cuenta sus valores arquitectónicos, melódicos y rítmicos, y sus relaciones de equilibrio. (SCHOENBERG, 1979).

Mencionamos anteriormente, al plantear el problema de investigación, que para la materia de armonía musical se debe tener sólidos conocimientos de teoría básica de la música, rítmica y solfeo. Definiremos a continuación lo que es la Rítmica y Solfeo, además de brindar el contenido de la materia de armonía musical (Temas que tomaremos en cuenta en el Tutor Inteligente).

- **RÍTMICA**

También llamado solfeo rítmico, es la técnica de entonar una melodía basándose únicamente en el ritmo del mismo, utilizando generalmente las manos.

- **SOLFEO**

También llamado solfeo melódico, es la técnica o método de entrenamiento musical, que consiste en entonar una melodía pronunciando los nombres de las notas musicales tomando en cuenta no solamente el ritmo sino, de manera principal, en los sonidos de las notas.

3.6.1 CONTENIDO DE ARMONÍA MUSICAL

El contenido de la materia de armonía musical es demasiado extenso, por lo que no tomaremos en cuenta todo, sino 5 temas iniciales de la materia de vital importancia:

- I. Escalas**
- II. Intervalos**
- III. Triadas y Cuatriadas**
- IV. Funciones Armónicas**
- V. Progresiones Armónicas**

3.7 PLATAFORMA TECNOLÓGICA

HERRAMIENTA	DESCRIPCIÓN
TECNOLOGÍA MULTIMEDIA	La tecnología multimedia es aquella que combina el uso de archivos de diferentes tipos de origen, como imágenes, vídeos, animaciones, voces, música, y textos. Su utilización se ha convertido en un uso estandarizado en internet y en las nuevas tecnologías en general, ganando una importante aceptación entre los usuarios debido a la interactividad que produce mediante la utilización de recursos de calidad.
DJANGO 1.8	Django es un framework de desarrollo Web que ahorra tiempo y hace que el desarrollo Web sea divertido. Utilizando Django puedes crear y mantener aplicaciones Web de alta calidad con un mínimo esfuerzo.
POSTGRESQL 9.4	PostgreSQL es un potente sistema de base de datos objeto-relacional de código abierto. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de fiabilidad e integridad de datos. Se ejecuta en los principales sistemas operativos que existen en la actualidad como: <ul style="list-style-type: none"> • Linux • UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) • Windows
SUBLIME TEXT 2	Es un editor de texto y editor de código fuente está escrito en C++ y Python para los plugins. Desarrollado originalmente como una extensión de Vim, con el tiempo fue creando una identidad propia, por esto aún conserva un modo de edición tipo vi llamado Vintage mode.
JAVASCRIPT	JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.
BOOTSTRAP 3	Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “responsive design” o diseño adaptativo.

Tabla 2.9: Tabla de herramientas a utilizar en prototipo

Fuente: Propia

4.1 INTRODUCCIÓN

A continuación detallaremos la estructura general del Tutor Web Inteligente propuesto, fusionando la arquitectura de los STI con el Agente Pedagógico y por ultimo con la metodología de desarrollo de software educativo MeISE como se observa en la figura 3.1, obteniendo así un producto de calidad.

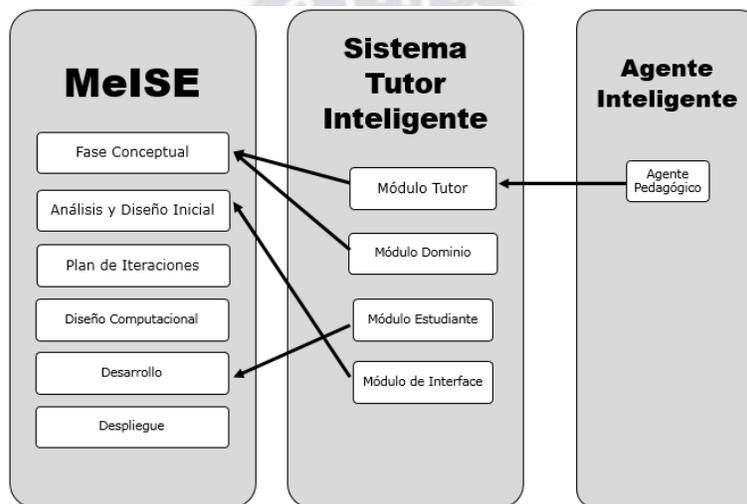


Figura 3.1: MeISE, Sistema Tutor Inteligente y Agentes Inteligentes

De la figura anterior podemos resumir lo siguiente:

- El diseño del agente pedagógico se lo realizará en el momento de diseñar el Módulo Tutor del STI como apoyo al mismo, ya que es éste el módulo el encargado de cumplir la función de Tutor o Enseñante.
- El Módulo Tutor será diseñado en la fase Conceptual de la metodología MeISE, específicamente en la actividad de Análisis de las necesidades Educativas.
- El Módulo de Dominio será diseñado en la fase Conceptual de la metodología MeISE, igualmente en la actividad de Análisis de las necesidades Educativas.

- El Módulo de Estudiante será elaborado en la fase de Diseño Computacional.
- El Módulo de Interface estará inmiscuido en la fase de Análisis y Diseño Inicial en la actividad de Diseño de Comunicación del producto.

4.2 ETAPAS MeISE (Metodología de Ingeniería de Software Educativo)

4.2.1 ETAPA DE DEFINICIÓN

4.2.1.1 FASE CONCEPTUAL

A. ANALIZAR LAS NECESIDADES EDUCATIVAS

- i. **MODELO INSTRUCCIONAL:** Incluye temática a atender, objetivos, conocimientos previos, fuentes de información, modelo educativo a utilizar, elementos de motivación y formas de evaluación.

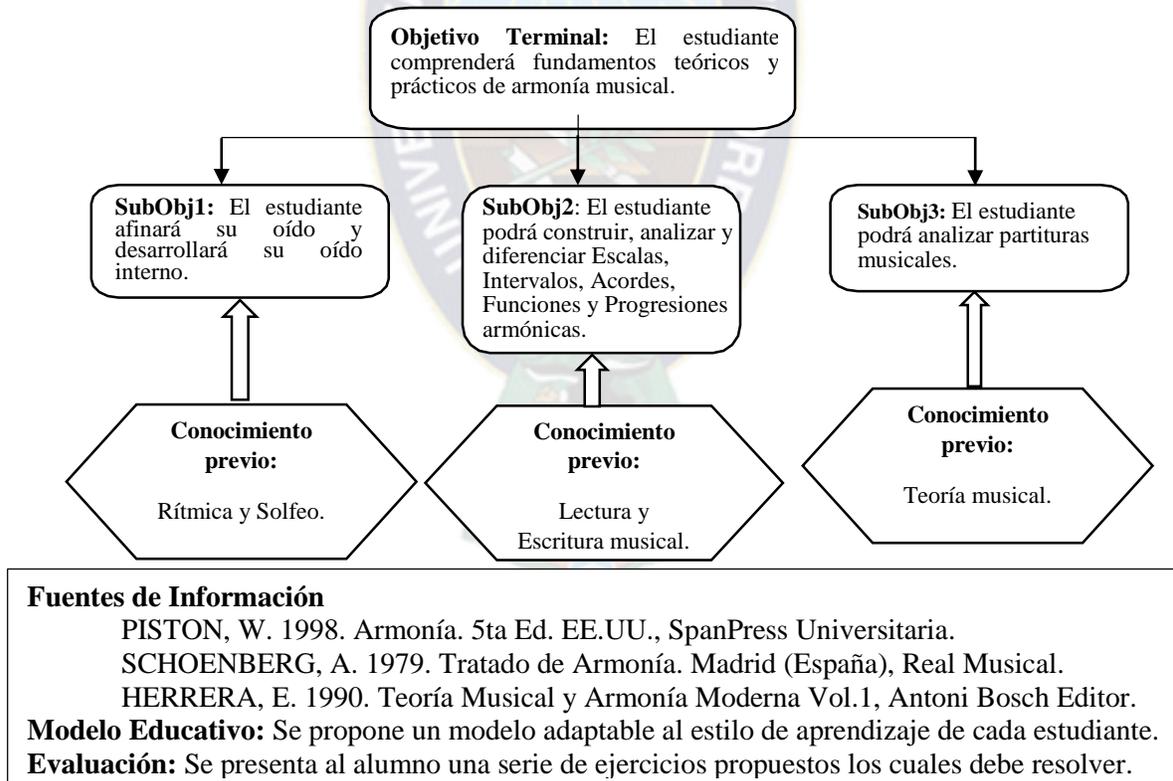


Figura 3.2: Modelo Instruccional

ii. GLOSARIO: Descripción de los términos que pueden causar confusión o duda

A continuación se definirá la terminología que se emplea en el desarrollo del Tutor Web Inteligente.

- Acorde: Combinación de dos o más notas que se ejecutan simultáneamente.
- Armadura: Indicación de la tonalidad predominante de una pieza musical. Está indicada por uno o más sostenidos o bemoles.
- Escala: Serie de notas que suben o bajan su altura en forma sucesiva y que poseen una separación interválica constante.
- Nota: Símbolo que representa un sonido aislado de altura y duración determinados.

iii. DISEÑO MÓDULO DOMINIO (STI)

Para conseguir impartir conocimiento, se debe contar con un dominio profundo del contenido a enseñar, es decir se debe contar con conocimiento profundo de la materia. Como se mencionó en el marco teórico sobre los módulos de un STI, el modelo de dominio se encontrará conocimiento detallado y específico acerca de los temas que brindara el Tutor Inteligente, estos son estructurados pedagógicamente y se cuenta con el soporte de un experto en el área, es decir, un experto con dominio de conocimiento en el campo de la música aporta con sus conocimiento para la elaboración del módulo de dominio.

En el módulo de dominio entonces se encontrara contenido de cada tema de armonía musical. El presente tutor inteligente considerará el siguiente plan de trabajo de la materia de armonía musical:

I. INTRODUCCIÓN

OBJETIVO.- Identificar el perfil de aprendizaje del estudiante, aplicando para ello, el Test De Estilos De Aprendizaje (Honey). De acuerdo a ello el tutor se adaptará al estudiante para un mejor aprendizaje.

II. ESCALAS

OBJETIVO.- El estudiante podrá construir y diferenciar los tipos de escalas tanto de forma auditiva como escrita utilizando la armadura correspondiente a la escala.

CONTENIDO

- Escala mayor
- Escala menor
- Escala menor natural
- Escala menor armónica
- Escala menor melódica
- Escala cromática
- Armadura y tonalidad
- Material de Apoyo

EVALUACIÓN

Ejercicios teóricos y auditivos:

- El estudiante tendrá que identificar el tipo de escala que escucha.
- Dado el sonido del primer grado de una escala identificar el n-ésimo grado.
- Verificar si una escala está escrita correctamente o no, de acuerdo a su estructura.

III. INTERVALOS

OBJETIVO.- El estudiante podrá construir y diferenciar los tipos de intervalos que existen tanto de forma auditiva como escrita.

CONTENIDO

- Clasificación
- Relación entre intervalos
- Inversión de Intervalos
- Material de Apoyo

EVALUACIÓN

Ejercicios teóricos y auditivos:

- El estudiante tendrá que identificar el tipo de intervalo que escucha.
- Construir a partir de una nota el intervalo pedido.
- Verificar si un intervalo está escrito correctamente o no, de acuerdo a su estructura.

IV. TRIADAS Y CUATRIADAS

OBJETIVO.- El estudiante podrá construir y diferenciar los tipos de triadas y cuatriadas tanto de forma auditiva como escrita.

CONTENIDO

- TRIADAS
- Clasificación
- Cifrado
- Inversión de triadas
- Ejercicios (teórico, auditivo)
- CUATRIADAS
- Clasificación y cifrado
- Material de Apoyo

EVALUACIÓN

Ejercicios teóricos y auditivos:

- El estudiante tendrá que identificar el tipo de triadas o cuatriada que escucha.
- Identificar además del tipo de triada o cuatriada, la inversión en la cual se encuentra de forma auditiva.
- Verificar si una triada o cuatriada está escrito correctamente o no, de acuerdo a su estructura.
- Ejercicios (teórico, auditivo)

V. FUNCIONES ARMÓNICAS

OBJETIVO.- El estudiante podrá identificar cualquier grado los tipos de triadas y cuatriadas tanto de forma auditiva como escrita.

CONTENIDO

- Grados de la Escala
- Tónica
- Dominante y subdominante
- Identificación del I, IV y V
- Ejercicios

VI. PROGRESIONES ARMÓNICAS

OBJETIVO.- El estudiante podrá analizar una progresión armónica, identificando la tonalidad y los grados utilizados.

CONTENIDO

- Movimiento entre fundamentales
- Análisis de una progresión
- Ejercicios de análisis de progresiones

iv. DISEÑO DEL MÓDULO TUTOR CON EL AGENTE PEDAGÓGICO

El diseño de este módulo, es donde se implementa el agente pedagógico para una mejor enseñanza al estudiante.

DISEÑO DEL AGENTE

En el módulo Tutor se plantean, los métodos y técnicas instruccionales y técnica de asesoría, estos estarán controlados por un agente pedagógico, este cumple la tarea del tutor o educador, es así que se introducirá su diseño como parte del módulo.

CONSTRUCCIÓN DEL AGENTE PEDAGÓGICO

Para la construcción del agente pedagógico, como ya anteriormente se dijo que este cumple con la labor de tutor o educador, se toma como base lo que plantea que este consta de dos procesos básicos:

- **Definición de objetivos del agente.-** El objetivo fundamental del agente es la de seleccionar los contenidos, actividades y estrategias que se usaran para enseñar al estudiante, para eso contará con la ayuda del test de estilos de aprendizaje aplicado a cada uno de los alumnos al principio del Tutor.
- **Adquisición y representación del conocimiento.-** Antes de proceder al diseño del programa agente, es necesario contar con una idea bastante precisa de PAMA de todo agente inteligente, en este caso los elementos propios que debe contener el Tutor Inteligente para la enseñanza de Armonía Musical (P= Percepciones, A= Acciones, M= Meta, A= Ambiente).

TABLA DEL PAMA

TIPO DE AGENTE	PERCEPCIONES	ACCIONES	METAS	AMBIENTE
Agente Pedagógico	- Datos del Estudiante - Test de Estilos de Aprendizaje - Selección de Temas - Respuestas del Estudiante a las Pruebas	- Instrucciones - Contenido - Pruebas - Ejemplos - Sugerencias - Resultados	- Motivar el aprendizaje de Armonía Musical - Comunicar avances al enseñante	Estudiantes de Instituciones Superiores de Música

Tabla 3.1: Análisis PAMA del Agente Pedagógico

El algoritmo para la representación de conocimiento es:

```
If Estudiante registrado Then
  Ingresa
  Verifica Avance
  Habilita temas
  Estudiante elige tema
  VerificaEstilo
  Muestra Tema y actividades
If Estudiante ve_tema Then
  Habilita prueba
  Estudiante Ingresa respuestas
  If respuestas correctas then
    Respuesta correcta
  Else
    Respuesta incorrecta
  End if
Else
  Alumno no ingresa prueba
End if
Else
  Registrar Estudiante
End if
```

B. REVISAR ALTERNATIVAS DE SOLUCIÓN

- i. ESTUDIO DE ALTERNATIVAS:** Establece las diferentes alternativas que se tienen para el desarrollo del software, se determina el tipo de modelo educativo y se justifica la elección.

MODELO EDUCATIVO

Analizando las diferentes propuestas de modelos educativos que existen actualmente se optó por el modelo constructivista, el cual ve al estudiante como el protagonista y constructor de su propio aprendizaje, por otra parte ve al docente o tutor como un orientador o facilitador

que comparte su saber con sus estudiantes. Se vio otras alternativas de modelos educativos como el conductismo pero en la actualidad ya no se aplica.

Reforzando al modelo constructivista mencionado anteriormente se estudió la teoría sobre los estilos de aprendizaje y ver la forma de aplicarlo en el sistema. Ya que los estudiantes se diferencian uno del otro en cuanto a sus estilos de aprendizajes, se pensó en un Tutor que se adapte al estilo de aprendizaje de cada estudiante, lo cual será posible gracias a un test previo que se le hará al estudiante al principio del curso. Se tomará en cuenta los 4 tipos de estilos de aprendizaje (propuestos por Gallego et. al.): activo, reflexivo, teórico y pragmático.

JUSTIFICACIÓN

Se eligió el modelo constructivista ya que se adapta más a lo que es el aprendizaje a través de software, tomando en cuenta que no se quiere reemplazar el rol de tutor al profesor sino el de apoyar al proceso de enseñanza – aprendizaje, de tal manera de fomentar al estudiante a ser el protagonista y activo en el aula, que es lo que se quiere según el modelo constructivista, a diferencia del modelo conductista el cual ve al alumno como mero receptor del saber, pasivo.

Por otra parte se consideró la aplicación de la teoría de estilos de aprendizaje en nuestro sistema porque no todos los estudiantes se adaptan al plan educativo que propone el docente de la materia, es por ese motivo que se ha decidido diseñar e implementar un Software Educativo (Tutor Inteligente) que se adapte, o bien, se acomode al estilo de aprendizaje de cada uno de los estudiantes.

C. ELABORAR UN ESTUDIO DE RIESGOS

- i. LISTA DE RIESGOS:** establece los riesgos relativos al desarrollo y a los aspectos pedagógicos y la forma de atenderlos.

A continuación se muestran algunos de los riesgos que podrían presentarse durante y posterior al desarrollo del Tutor Web en cuanto a aspectos pedagógicos y cómo se planea atenderlos.

RIESGOS RELATIVOS AL DESARROLLO	ASPECTOS PEDAGÓGICOS	FORMA DE ATENDERLOS
No considerar los conocimientos previos requeridos a cada tema.	El estudiante no logra entender el tema y por lo tanto reprueba.	El estudiante usuario no podrá continuar con el avance de la materia si no termina el tema anterior.
Las instituciones de enseñanza musical normalmente no cuentan con equipos de computación como recurso de enseñanza.	Los estudiantes no cuentan con un software educativo que les motive.	El diseño del prototipo del Tutor tendrá un diseño responsivo, de tal manera que el estudiante pueda acceder, desde su teléfono inteligente, al Tutor.
El Tutor tenga errores en el contenido de los temas expuestos.	El estudiante aprenda de manera incorrecta los conceptos explicados en el Tutor.	Revisar el contenido del Tutor con un profesional en música y hacer pruebas en alguna institución superior de música.

Tabla 3.2: Lista de Riesgos

D. CONFORMAR EL EQUIPO DE TRABAJO Y EL PLAN INICIAL DE DESARROLLO

- i. PLAN INICIAL:** se conforma el equipo de trabajo, se elabora la programación de actividades, se asignan responsables a cada una y se determinan los tiempos estimados para llevarlas a cabo.

EQUIPO TÉCNICO DEL PROYECTO

El equipo técnico del proyecto es el responsable de la gestión y coordinación técnica del Proyecto.

A continuación se detalla el equipo de trabajo para el siguiente proyecto:

CARGO	RESPONSABLE	TAREA
Jefe de Proyecto	Univ. Ivar Raúl Rodríguez Guachalla	Responsable del proyecto
Coordinador de actividades	Lic. Teresa Gutiérrez	Coordinar las reuniones y supervisar el avance del proyecto en la Escuela Nacional de Música "Luis Felipe Arce"
Equipo de desarrollo	Univ. Ivar Raúl Rodríguez Guachalla	Responsable del desarrollo del proyecto (análisis, diseño e implementación)

Tabla 3.3: Equipo de Trabajo del proyecto

PROGRAMACIÓN DE ACTIVIDADES

Se detallará cuidadosamente la programación de actividades agrupadas por fases, el responsable de cada actividad y el tiempo estimado de las mismas.

En la Tabla 3.4 se puede apreciar las actividades con fecha de inicio 4 de Agosto y finalización 14 de Agosto.

Fecha de Inicio: 4 de Agosto de 2015	
Fecha de Finalización: 14 de Agosto de 2015	
Responsable: Ivar Raúl Rodríguez Guachalla	
Etapas a la que pertenece: Etapa de Definición	
Artefactos a desarrollar	
FASE	ACTIVIDAD
Conceptual	Modelo Instruccional Glosario Estudio de Alternativas Lista de Riesgos Plan Inicial

Tabla 3.4: Tabla de Programación de Actividades 1

En la Tabla 3.5 se puede apreciar las actividades con fecha de inicio 17 de Agosto y finalización 24 de Agosto.

Fecha de Inicio: 17 de Agosto de 2015	
Fecha de Finalización: 24 de Agosto de 2015	
Responsable: Ivar Raúl Rodríguez Guachalla	
Etapa a la que pertenece: Etapa de Definición	
Artefactos a desarrollar:	
FASE	ACTIVIDAD
Conceptual	Modelo de Actores Modelo de Casos de Uso Modelo de Aceptación

Tabla 3.5: Tabla de Programación de Actividades 2

En la Tabla 3.6 se puede apreciar las actividades con fecha de inicio 25 de Agosto y finalización 31 de Agosto.

Fecha de Inicio: 25 de Agosto de 2015	
Fecha de Finalización: 31 de Agosto de 2015	
Responsable: Ivar Raúl Rodríguez Guachalla	
Etapa a la que pertenece: Etapa de Definición	
Artefactos a desarrollar:	
FASE	ACTIVIDAD
Análisis y Diseño Inicial	Modelo de Requisitos Descripción de la Arquitectura Modelo Educativo

Tabla 3.6: Tabla de Programación de Actividades 3

En la Tabla 3.7 se puede apreciar las actividades con fecha de inicio 01 de Septiembre y finalización 11 de Septiembre.

Fecha de Inicio: 01 de Septiembre de 2015	
Fecha de Finalización: 11 de Septiembre de 2015	
Responsable: Ivar Raúl Rodríguez Guachalla	
Etapa a la que pertenece: Etapa de Definición	
Artefactos a desarrollar:	
FASE	ACTIVIDAD
Análisis y Diseño Inicial	Modelo de Interfaz Modelo de Navegación Prototipo de la Interfaz de Usuario

Tabla 3.7: Tabla de Programación de Actividades 4

En la Tabla 3.8 se puede apreciar las actividades con fecha de inicio 14 de Septiembre y finalización 18 de Septiembre.

Fecha de Inicio: 14 de Septiembre de 2015	
Fecha de Finalización: 18 de Septiembre de 2015	
Etapa a la que pertenece: Etapa de Definición	
Responsable: Ivar Raúl Rodríguez Guachalla	
Artefactos a desarrollar:	
FASE	ACTIVIDAD
Plan de Iteraciones	Plan de Iteraciones Lista de Iteraciones Priorizadas

Tabla 3.8: Tabla de Programación de Actividades 5

En la Tabla 3.9 se puede apreciar las actividades con fecha de inicio 21 de Septiembre y finalización 30 de Septiembre.

Fecha de Inicio: 21 de Septiembre de 2015	
Fecha de Finalización: 30 de Septiembre de 2015	
Etapa a la que pertenece: Etapa de Desarrollo	
Responsable: Ivar Raúl Rodríguez Guachalla	
Artefactos a desarrollar:	
FASE	ACTIVIDAD
Diseño Computacional	Plan de Trabajo Modelo de Diseño Modelo de Navegación refinado

Tabla 3.9: Tabla de Programación de Actividades 6

En la Tabla 3.10 se puede apreciar las actividades con fecha de inicio 01 de Octubre finalización 09 de Octubre.

Fecha de Inicio: 1 de Octubre de 2015	
Fecha de Finalización: 9 de Octubre de 2015	
Etapa a la que pertenece: Etapa de Desarrollo	
Responsable: Ivar Raúl Rodríguez Guachalla	
Artefactos a desarrollar:	
FASE	ACTIVIDAD
Diseño Computacional y Desarrollo	Modelo de Navegación refinado Modelo de Interfaz de Usuario Modelo de Desarrollo

Tabla 3.10: Tabla de Programación de Actividades 7

En la Tabla 3.11 se puede apreciar las actividades con fecha de inicio 12 de Octubre y finalización 26 de Octubre.

Fecha de Inicio: 12 de Octubre de 2015	
Fecha de Finalización: 26 de Octubre de 2015	
Etapa a la que pertenece: Etapa de Desarrollo	
Responsable: Ivar Raúl Rodríguez Guachalla	
Artefactos a desarrollar:	
FASE	ACTIVIDAD
Desarrollo	Modelo de Desarrollo Modelo de Pruebas Unitarias Modelo de Integración Pruebas de Integración

Tabla 3.11: Tabla de Programación de Actividades 8

En la Tabla 3.12 se puede apreciar las actividades con fecha de inicio 27 de Octubre y finalización 12 de Noviembre.

Fecha de Inicio: 27 de Octubre de 2015	
Fecha de Finalización: 12 de Noviembre de 2015	
Etapa a la que pertenece: Etapa de Desarrollo	
Responsable: Ivar Raúl Rodríguez Guachalla	
Artefactos a desarrollar:	
FASE	ACTIVIDAD
Diseño Computacional	Modelo de diseño refinado Modelo de navegación refinado Modelo de interfaz de usuario
Desarrollo	Modelo de desarrollo refinado Modelo de integración refinado Pruebas de integración refinadas
Despliegue	Producto Manual de Usuario

Tabla 3.12: Tabla de Programación de Actividades 9

En la Tabla 3.13 se puede apreciar las actividades con fecha de inicio 13 de Noviembre finalización 23 de Noviembre.

Fecha de Inicio: 13 de Noviembre de 2015	
Fecha de Finalización: 23 de Noviembre de 2015	
Etapa a la que pertenece: Etapa de Desarrollo	
Responsable: Ivar Raúl Rodríguez Guachalla	
Artefactos a desarrollar:	
FASE	ACTIVIDAD
Diseño Computacional	Modelo de diseño refinado Modelo de navegación refinado Modelo de interfaz de usuario
Desarrollo	Modelo de desarrollo refinado Modelo de integración refinado Pruebas de integración
Despliegue	Producto refinado Manual de Usuario corregido Correcciones de las pruebas

Tabla 3.13: Tabla de Programación de Actividades 10

E. IDENTIFICAR LA FUNCIONALIDAD QUE SE PRETENDE ALCANZAR CON EL SOFTWARE

- i. **MODELO DE ACTORES:** identifica los tipos de usuario que utilizarán el software y describe sus características.

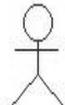
ACTORES	DESCRIPCIÓN
 ADMINISTRADOR (EXPERTO)	<ul style="list-style-type: none"> Realiza Altas, Bajas y Modificaciones de los temas, pruebas, ejercicios y registro de usuarios (estudiantes). Generar reportes
 USUARIO (ESTUDIANTE)	<ul style="list-style-type: none"> Visualiza los temas desbloqueados de acuerdo al avance que éste tenga. Resuelve pruebas por tema para desbloquear los temas siguientes. Ejercita temas avanzados con ejercicios prácticos.

Tabla 3.14: Modelo de actores

ii. **MODELO DE CASOS DE USO:** establece un modelo general de las funciones que cubrirá el sistema a través de diagramas de casos de uso y su especificación.

• **DIAGRAMA DE CASOS DE USO: GESTIÓN DE ESTUDIANTES**

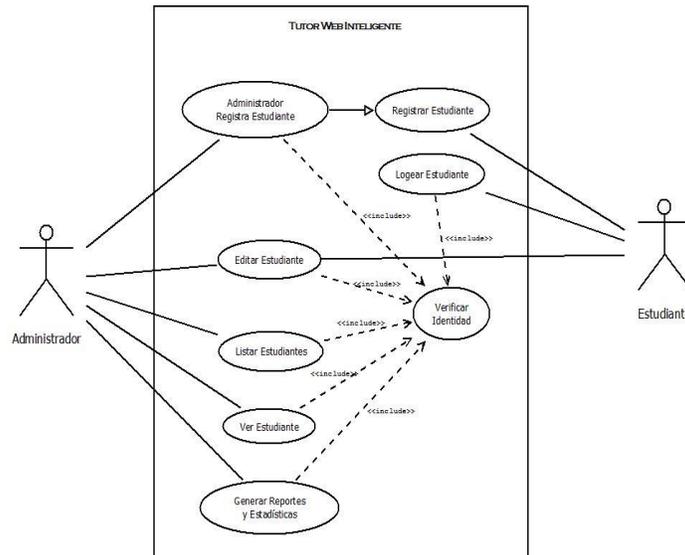


Figura 3.3: Casos de uso de Gestión de Estudiantes

✓ **CASO DE USO: REGISTRAR ESTUDIANTE.**

Descripción: El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se desee registrar un estudiante.

Precondición: El estudiante no está registrado en el sistema.

Secuencia Normal:

1. El sistema solicita los siguientes datos: e-mail, Nombres, Apellidos, Contraseña.
2. El **estudiante** proporciona los datos solicitados.
3. El sistema comprueba y almacena los datos suministrados.

Postcondición: Los datos del **estudiante** quedan almacenados en el sistema.

Excepciones:

Si el estudiante ya está registrado, se muestra un mensaje de alerta indicando la excepción.

✓ **CASO DE USO: ADMINISTRADOR REGISTRA ESTUDIANTE.**

Descripción: El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador desee registrar un estudiante.

Precondición: El estudiante a registrar no está en la base de datos del sistema.

Secuencia Normal:

1. El sistema solicita los siguientes datos: e-mail, Nombres, Apellidos, Contraseña.
2. El **administrador** proporciona los datos del estudiante.
3. El sistema comprueba y almacena los datos suministrados.

Postcondición: Los datos del **estudiante** quedan almacenados en el sistema.

Excepciones: Si el estudiante ya está registrado, se realiza el caso de uso Editar Estudiante.

✓ **CASO DE USO: LOGEAR USUARIO.**

Descripción: El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un **estudiante** desee entrar al sistema.

Precondición: El estudiante está registrado en el sistema.

Secuencia Normal:

1. El sistema solicita los siguientes datos: e-mail, contraseña.

2. El **estudiante** proporciona los datos solicitados.
3. El sistema comprueba y logea al estudiante.

Postcondición: El **estudiante** queda logeado en el sistema.

Excepciones:

Si el **estudiante** no está registrado, el sistema muestra un mensaje de error, a continuación el caso de uso queda sin efecto.

✓ **CASO DE USO: EDITAR USUARIO.**

Descripción: El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se solicita la modificación de un usuario, o durante los siguientes casos de uso: ADMINISTRADOR REGISTRA USUARIO.

Precondición: El estudiante existe en el sistema.

Secuencia Normal:

1. Se realiza el caso de uso **Logear usuario**.
2. El sistema muestra los datos actuales del **estudiante** para su modificación
3. El sistema almacena los datos suministrados.

Postcondición: Los datos del **estudiante** quedan modificados en el sistema.

✓ **CASO DE USO: VER ESTUDIANTE.**

Descripción: El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el **administrador** solicite ver los datos de un **estudiante**.

Precondición: El **estudiante** está registrado en el sistema.

Secuencia Normal:

1. El actor **Administrador** selecciona un estudiante para ver los datos.
2. El sistema muestra todos los datos existentes para ese estudiante.

Postcondición: Los datos del **estudiante** son mostrados al **administrador**.

✓ **CASO DE USO: LISTAR ESTUDIANTES.**

Descripción: El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el **administrador** solicite listar todos los **estudiantes** registrados en el sistema.

Precondición: Existe al menos un **estudiante** registrado en el sistema.

Secuencia Normal:

1. El Administrador solicita que se muestre una lista de todos los estudiantes registrados en el sistema.
2. El sistema muestra una lista con todos los estudiantes registrados.

Postcondición: Se muestra una lista con los datos de los estudiantes registrados.

✓ **CASO DE USO: GENERAR REPORTES Y ESTADÍSTICAS.**

Descripción: El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el **administrador** solicite mostrar reportes y estadísticas de los estudiantes registrados en el sistema.

Precondición: Existe al menos un **estudiante** registrado en el sistema.

Secuencia Normal:

1. El Administrador solicita que se muestre una lista de todos los estudiantes registrados en el sistema y su avance general o bien un reporte general.
2. El sistema muestra una lista detallada con todos los estudiantes registrados y su respectivo avance, los mejores estudiantes. En el caso de reporte general el sistema

mostrará porcentajes generales de todos los estudiantes: Porcentaje y Número de estudiantes agrupados por estilo de aprendizaje, Porcentaje de estudiantes por tema.

Postcondición: Se muestra una variedad de reportes y estadísticas.

- **DIAGRAMA DE CASOS DE USO: TEMAS**

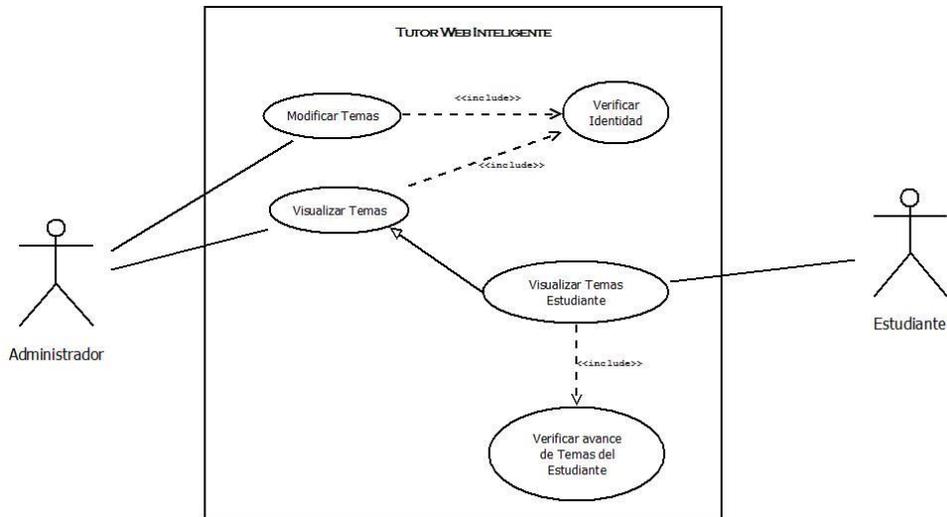


Figura 3.4: Casos de uso de Temas

✓ **CASO DE USO: MODIFICAR TEMAS.**

Descripción: El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador modificar temas.

Precondición: El usuario está logeado como Administrador.

Secuencia Normal:

1. El **Administrador** selecciona la opción modificar tema.
2. El Sistema muestra todos los temas para que se elija uno de ellos.
3. El **Administrador** selecciona el tema que desee modificar.
4. El sistema muestra un editor HTML del tema para su posterior modificación.

Postcondición: El Sistema guarda los cambios realizados del código HTML.

✓ **CASO DE USO: VISUALIZAR TEMAS.**

Descripción: El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el Administrador desee revisar algún tema en específico.

Precondición: El usuario está logeado como Administrador.

Secuencia Normal:

1. El **Administrador** selecciona un tema para visualizarlo (puede ver todos los temas).
2. El sistema muestra el tema seleccionado.

Postcondición: El sistema muestra el tema seleccionado por el administrador para su posterior revisión.

✓ **CASO DE USO: VISUALIZAR TEMAS ESTUDIANTE.**

Descripción: El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un estudiante desee visualizar algún tema en específico.

Precondición: El usuario está logeado como Estudiante.

Secuencia Normal:

1. El sistema muestra los temas habilitados y no habilitados para la visualización y avance del estudiante.
2. El **Estudiante** selecciona el tema que desea avanzar.

3. El sistema verifica el avance de temas del estudiante, si le corresponde avanzar ese tema o es un tema ya avanzado anteriormente, muestra el tema seleccionado, si no deniega el acceso al tema.

Postcondición: El sistema muestra el tema seleccionado por el estudiante o deniega el acceso si el tema no le corresponde aún.

- **DIAGRAMA DE CASOS DE USO: EVALUACIONES**

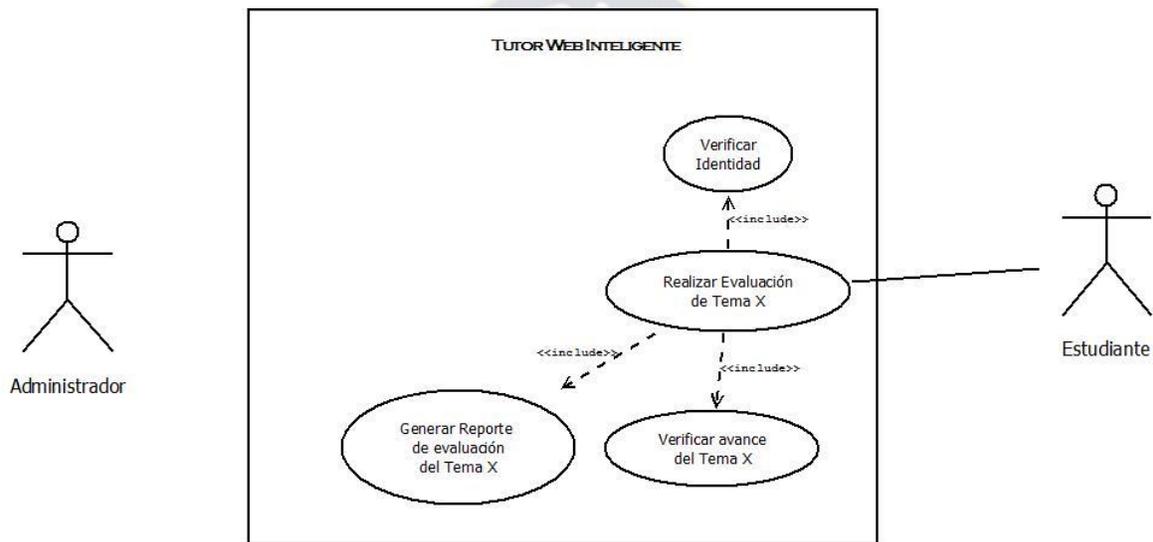


Figura 3.5: Casos de uso de Evaluaciones

- ✓ **CASO DE USO: REALIZAR EVALUACIÓN DEL TEMA X.**

Descripción: El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un estudiante desee resolver la evaluación del tema avanzado.

Precondición: El estudiante está identificado.

Secuencia Normal:

1. El **Estudiante** selecciona la evaluación que desea realizar.

2. El sistema verifica si es que le corresponde al estudiante dar la evaluación seleccionada, si es que no le corresponde le deniega el acceso y si le corresponde le muestra la evaluación.
3. El estudiante realiza la evaluación.
4. El sistema califica la evaluación y le muestra el puntaje adquirido al estudiante.

Postcondición: El sistema almacena el puntaje adquirido por el estudiante, guarda su avance y desbloquea el siguiente tema.

F. ESTABLECER LOS CRITERIOS DE MEDICIÓN DE CALIDAD DEL PROCESO, CONSIDERANDO ASPECTOS TANTO TÉCNICOS COMO PEDAGÓGICOS.

- i. **MODELO DE ACEPTACIÓN:** incluye las características mínimas que deben cumplirse para que el producto se acepte.

Desde el punto de vista técnico, se requiere que el tutor web sea de acceso gratuito desde cualquier navegador web. En cuanto a su interfaz, ésta debe ser intuitiva y estandarizada de manera tal que el usuario centre su atención en el estudio de la asignatura de forma natural y que la ayuda de la aplicación, resulte conveniente y útil a la hora de resolver inquietudes.

En cuanto al aspecto pedagógico, se busca facilitar a estudiantes de Armonía Musical de instituciones superiores de música el acceso a contenido instruccional, utilizando multimedios. Además, se busca afianzar al estudiante al uso de la tecnología en el proceso de enseñanza - aprendizaje.

ASPECTOS TÉCNICOS

Se listarán los aspectos técnicos mínimos que se deben cumplir a la hora de querer utilizar el Tutor Web, basándonos en las herramientas usadas para el prototipo.

- Navegador Web: Se recomienda los navegadores Chrome, Firefox y Opera debido a son muy compatibles con las funcionalidades que nos brinda el framework Bootstrap 3.
- Una salida de audio del computador ya que será indispensable en el momento de resolver los ejercicios.

4.2.1.2 ANÁLISIS Y DISEÑO INICIAL

A. IDENTIFICAR LOS REQUISITOS FUNCIONALES Y NO FUNCIONALES QUE SE CUBRIRÁN CON EL SOFTWARE.

- i. **MODELO DE REQUISITOS:** Se determinan los requisitos que debe cumplir el software en cuanto a funcionalidad, comunicación, interfaz y docencia.

REQUERIMIENTOS DE ADMINISTRADOR

- Registrar nuevos estudiantes (usuarios)
- Eliminar y editar estudiantes
- Revisar reportes de todos los estudiantes
- Editar temas, actividades y evaluaciones

REQUERIMIENTOS DE ESTUDIANTE

- Acceso a los temas de acuerdo a su avance
- Realizar el test de estilos de aprendizaje
- Actividades de acuerdo a su estilo de aprendizaje
- Resolver la evaluación del tema avanzado
- Realizar los ejercicios propuestos
- Acceso a un glosario de la materia

B. ESTABLECER LA ARQUITECTURA DEL SOFTWARE.

- DESCRIPCIÓN DE LA ARQUITECTURA:** establecer la arquitectura base sobre la cual se desarrollará el software; se debe considerar que dicha arquitectura sea capaz de atender adecuadamente las tareas de aprendizaje que se van a manejar.

La arquitectura base sobre la que se desarrollará el Tutor Web Inteligente será la arquitectura misma de un Tutor Inteligente basado en 4 módulos visto en el marco teórico:

Módulo del Alumno: Este módulo debe representar el estado inicial del alumno y sus características individuales.

Módulo del Tutor: Este módulo posee el conocimiento sobre las estrategias y tácticas de enseñanza para poder seleccionarlas en función de las características del alumno, que están almacenadas en el módulo del alumno.

Módulo del Dominio: Este módulo posee el conocimiento de la materia formado por las reglas de producción, estereotipos, etc. De aquí el módulo tutor obtiene el conocimiento que debe enseñar.

Módulo de Interface: Es la interface de interacción entre el STI y el alumno real, que se encarga de presentar el material del dominio y cualquier otro elemento didáctico de la manera correcta.

C. ELABORAR EL DISEÑO EDUCATIVO

- i. MODELO EDUCATIVO:** Se definen el objetivo terminal y los subobjetivos, y en base a éstos se establecen las tareas de aprendizaje apegadas al tipo de modelo educativo.

En el punto de Estudio de Alternativas de la fase conceptual se definió un modelo educativo basado en el modelo educativo constructivista y en los estilos de aprendizaje, logrando un Tutor adaptable al estudiante.

D. ELABORAR EL DISEÑO DE COMUNICACIÓN GENERAL DEL PRODUCTO

- i. MODELO DE INTERFAZ:** Diseño de las zonas de comunicación y pantallas que se seguirán a lo largo del desarrollo.

El diseño de la interfaz es de vital importancia ya que con ella se logra la comunicación entre usuario final y la máquina. Distinguiremos dos tipos de interfaz de usuario: Interfaz Estudiante e Interfaz Administrador. Ambos tienen el mismo diseño de interfaz, solo que el administrador tiene más opciones en el menú, opciones descritas en los casos de uso.

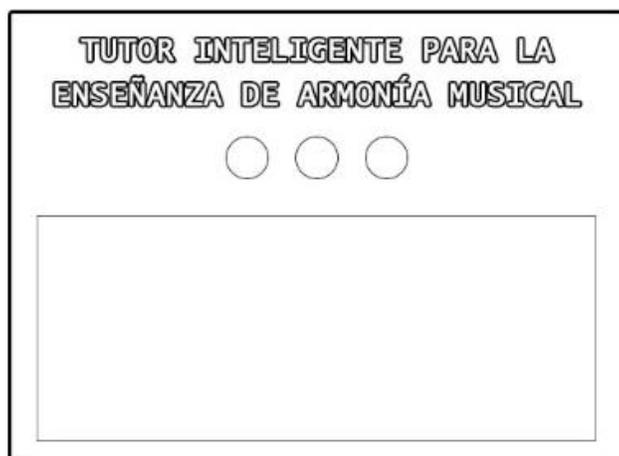


Figura 3.6: Prototipo de interfaz de usuario

ii. MODELO DE NAVEGACIÓN: Diseño de los caminos de navegación generales que se presentarán al usuario.



Figura 3.7: Diagrama de Navegación para el botón de Aprendizaje



Figura 3.8: Diagrama de Navegación para el botón de Ejercitar

- iii. **PROTOTIPO DE LA INTERFAZ DE USUARIO:** Establecer las plantillas de diseño que se seguirán a lo largo del desarrollo.

Se utilizará las vistas de datos abstractos (ADV) de la metodología OOHDM, el cual detalla la interfaz de usuarios del software de manera que se observa la posible organización y comportamiento final que tendrán.

- ✓ ADV para el Registro de Estudiantes

El diagrama muestra una interfaz de usuario para el registro de estudiantes. El título es "ADV: REGISTRO DE ESTUDIANTES". Hay cuatro campos de entrada de texto, cada uno con un label y el tipo de dato "Varchar": "e-mail : Varchar", "nombres : Varchar", "apellidos : Varchar" y "contraseña : Varchar". Debajo de estos campos hay un botón rectangular gris con el texto "REGISTRAR".

Figura 3.9: ADV para el registro de estudiantes

- ✓ ADV para el Login de usuario

El diagrama muestra una interfaz de usuario para el login de usuario. El título es "ADV: Login de Usuario". Debajo del título hay un subtítulo: "TUTOR WEB INTELIGENTE PARA LA ENSEÑANZA DE ARMONÍA MUSICAL". Hay dos campos de entrada de texto: "E-MAIL" y "CONTRASEÑA". Debajo de estos campos hay un botón rectangular gris con el texto "ENTRAR".

Figura 3.10: ADV para el login de Usuarios

- ✓ ADV para la interfaz de Bienvenida

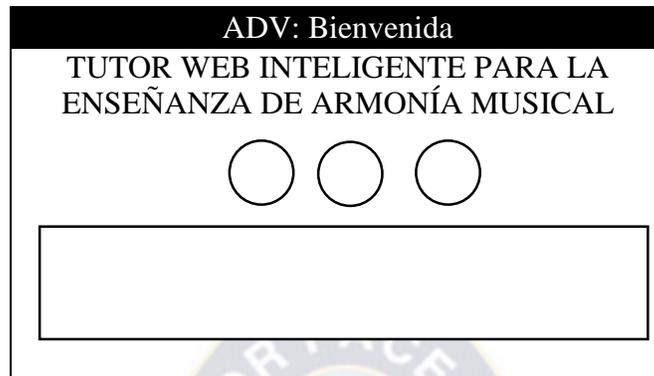


Figura 3.11: ADV para la interfaz de bienvenida

4.2.1.3 FASE PLAN DE ITERACIONES

A. PLAN DE ITERACIONES PRIORIZADAS

Se dividió el desarrollo en tres iteraciones en base a los casos de uso planteados anteriormente y los requerimientos que ameriten los mismos, cuidando de que cada una cubriera requisitos y objetivos completos.

ITERACIÓN #1:

- Diseño e implementación de una base de datos que responda a los requerimientos de todos casos de uso.
- Definir la plantilla HTML con la que se trabajará.
- Implementar los casos de uso de Gestión de Estudiantes.
 - Registro de estudiantes
 - Administrador registra estudiantes
 - Logueo de usuarios
 - Edición de datos de usuario
 - Ver datos estudiantes

- Listar estudiantes
- Generación de reportes y estadísticas

ITERACIÓN #2

- Implementar los casos de uso de Temas.
 - Modificar temas.
 - Visualizar temas
 - Visualizar temas estudiante

ITERACIÓN #3

- Implementar los casos de uso de Evaluaciones.
 - Estudiante realiza evaluación del tema x.

4.2.2 ETAPA DE DESARROLLO

4.2.2.1 DISEÑO COMPUTACIONAL

A. REALIZAR EL PLAN DE TRABAJO DE LA ITERACIÓN

- i. PLAN DE TRABAJO:** se determinan las tareas que se realizarán en el diseño del software, se asignan a los miembros del equipo y se calendarizan.

ITERACIÓN	FECHA INICIO	FECHA FIN	RESPONSABLE
ITERACIÓN 1	21-09-2015	26- 10-2015	Ivar Rodríguez Guachalla
ITERACIÓN 2	27-10-2015	12-10-2015	Ivar Rodríguez Guachalla
ITERACIÓN 3	13-10-2015	23-10-2015	Ivar Rodríguez Guachalla

Tabla 3.15: Plan de Trabajo por iteración.

B. ELABORAR EL DISEÑO COMPUTACIONAL

- i. **MODELO DE DISEÑO:** detallar el diseño a través de diagramas de clases y secuencia, incluir la descripción de clases y métodos; para los desarrollos que requieren bases de datos, incluir la especificación de diccionario de datos y diagramas entidad – relación.

MODELO ENTIDAD – RELACIÓN

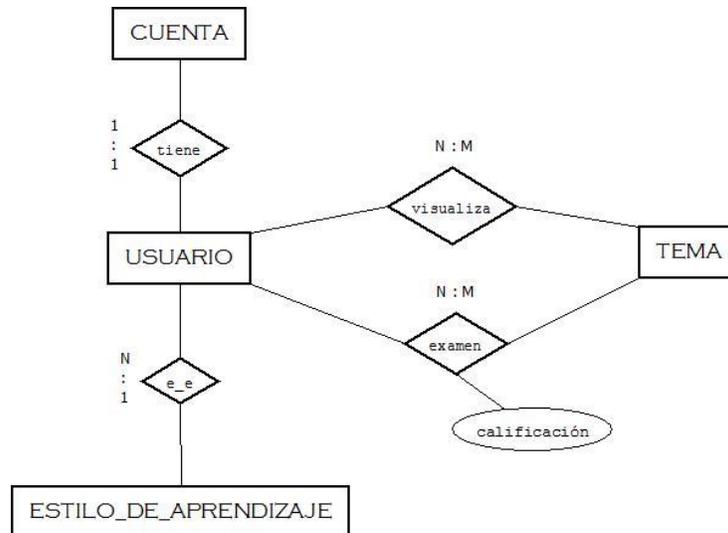


Figura 3.12: Modelo Entidad – Relación del Sistema

ATRIBUTOS

A continuación se listarán los atributos por entidad del diagrama anterior:

CUENTA(**id_cuenta**, password, nivel)

USUARIO(**id_usuario**, email, nombres, apellidos)

TEMA(**id_tema**, titulo)

ESTILO_DE_APRENDZAJE(**id_estilo**, descripcion)

MODELO RELACIONAL

TEMA

idTema	titulo
---------------	--------

ESTILO_DE_APRENDIZAJE

idEstilo	descripcion
-----------------	-------------

USUARIO

idUsuario	email	nombres	apellidos	<u>idEstilo</u>
------------------	-------	---------	-----------	-----------------

CUENTA

idCuenta	password	nivel	<u>idUsuario</u>
-----------------	----------	-------	------------------

visualiza

idUsuario	idTema
------------------	---------------

examen

idUsuario	idTema	calificacion
------------------	---------------	--------------

DIAGRAMA DE CLASES

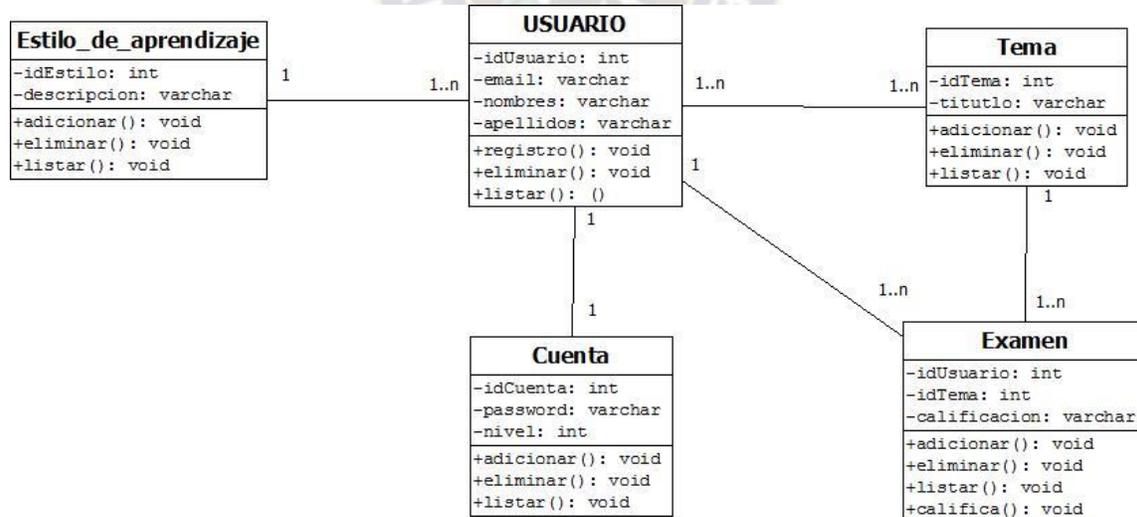


Figura 3.13: Diagrama de Clases

DIAGRAMA DE SECUENCIA: REGISTRO DE ESTUDIANTES

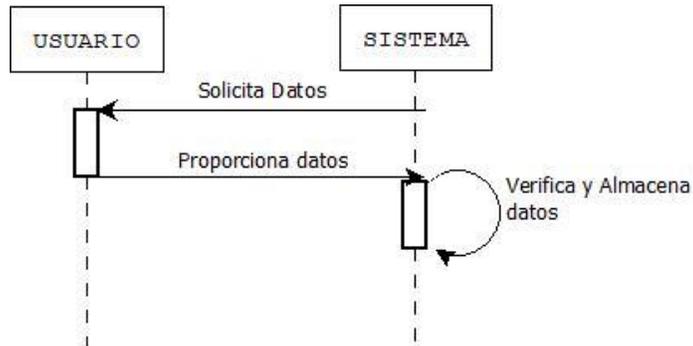


Figura 3.14: Diagrama de Secuencia de Registro de Estudiantes

DIAGRAMA DE SECUENCIA: VISUALIZAR TEMAS

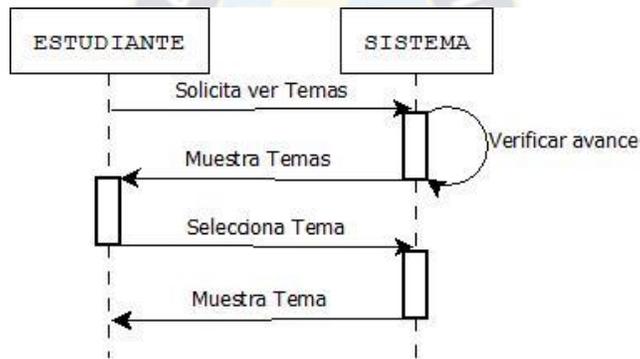


Figura 3.15: Diagrama de Secuencia de Visualizar Temas

DIAGRAMA DE SECUENCIA: VISUALIZAR TEMAS

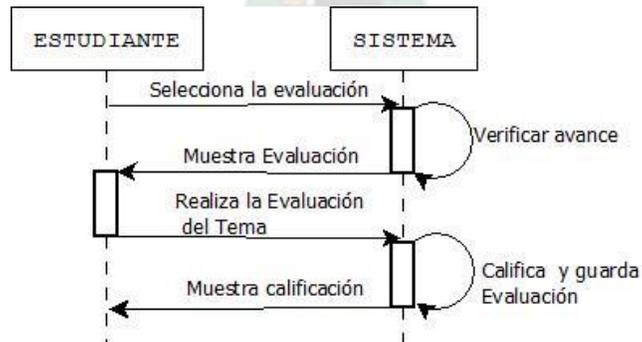


Figura 3.16: Diagrama de Secuencia Realizar Evaluación

DICCIONARIO DE DATOS

Tabla: TEMA				
Llave	Nombre Campo	Tipo	Tamaño	Descripción
PK	IdTema	Numérico	2	Almacena código de tema
	Titulo	Texto	30	Almacena el título del tema

Tabla: ESTILO_DE_APRENDIZAJE				
Llave	Nombre Campo	Tipo	Tamaño	Descripción
PK	IdEstilo	Numérico	2	Almacena código del Estilo de Aprendizaje
	Descripción	Texto	30	Almacena el nombre del estilo de aprendizaje

Tabla: USUARIO				
Llave	Nombre Campo	Tipo	Tamaño	Descripción
PK	IdUsuario	Numérico	4	Almacena código del Usuario (Estudiante o Administrador)
	Email	Texto	30	Almacena el email del Usuario
	Nombres	Texto	20	Almacena el Nombre del Usuario
	Apellidos	Texto	28	Almacena el Nombre del Usuario
FK	idEstilo	Numérico	1	Almacena código del Estilo de aprendizaje que se adapta al usuario

Tabla: CUENTA				
Llave	Nombre Campo	Tipo	Tamaño	Descripción
PK	IdCuenta	Numérico	4	Almacena código de la cuenta del usuario
	Password	Texto (Hash)	30	Almacena la contraseña de la cuenta del usuario
	Nivel	Numérico	1	Almacena el nivel de acceso
FK	idUsuario	Numérico	4	Almacena el código del Usuario al que pertenece la cuenta

C. REFINAR EL DISEÑO DE NAVEGACIÓN

- i. **MODELO DE NAVEGACIÓN REFINADO:** Diseñar los caminos de navegación específicos para la iteración en desarrollo.

Se vio la necesidad de ampliar y refinar el diagrama de navegación anterior de tal manera de lograr los objetivos propuestos, ya que el anteriormente propuesto no cumplía a cabalidad con los todos los objetivos del proyecto. En la figura 3.17 se observa el diagrama de navegación refinado:

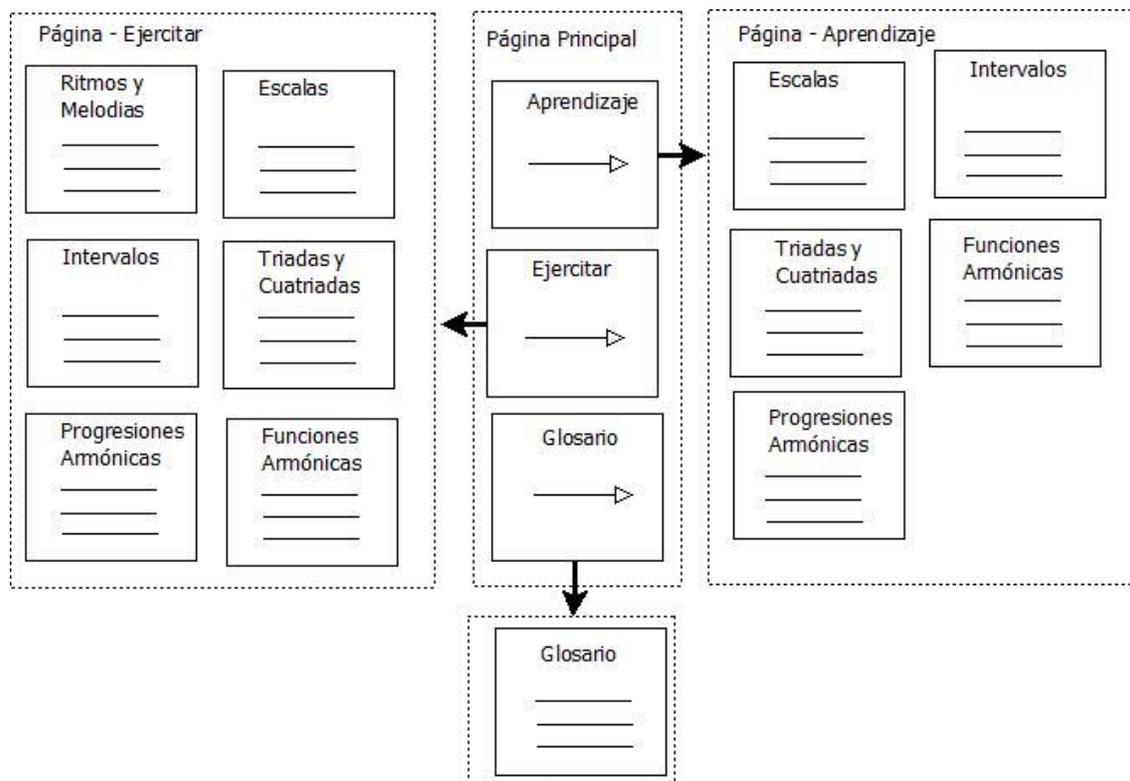


Figura 3.17: Modelo de Navegación refinado

D. REFINAR PROTOTIPO DE INTERFAZ

i. MODELO DE INTERFAZ USUARIO (desarrollar las pantallas específicas para los elementos de la iteración en desarrollo)

No se presenta ninguna refinación de la interfaz de usuario propuesta dado que no se presentó inconvenientes ni restricciones de uso con la dicha interfaz.

4.2.2.2 FASE DE DESARROLLO

A. DESARROLLAR LOS COMPONENTES

- i. **MODELO DE DESARROLLO:** Determinar los componentes a desarrollar y documentarlos.

ITERACIÓN 1

- Se definió una plantilla HTML con diseño responsivo en base a Bootstrap 3



Figura 3.18: Interfaz de Usuario con diseño Responsive

- Se implementó el diseño de base datos en base a la definición de los modelos de la aplicación en django, para su posterior sincronización con el motor de base de datos PostgreSQL de manera automática.
- Se desarrolló los casos de uso de Gestión de Estudiantes. A continuación se mostrarán algunas capturas de pantalla de los mismos.

Registro de usuarios Estudiantes

La interfaz de registro de Estudiantes se ve de la siguiente figura, consta de los campos correo electrónico, nombres, apellidos y contraseña.

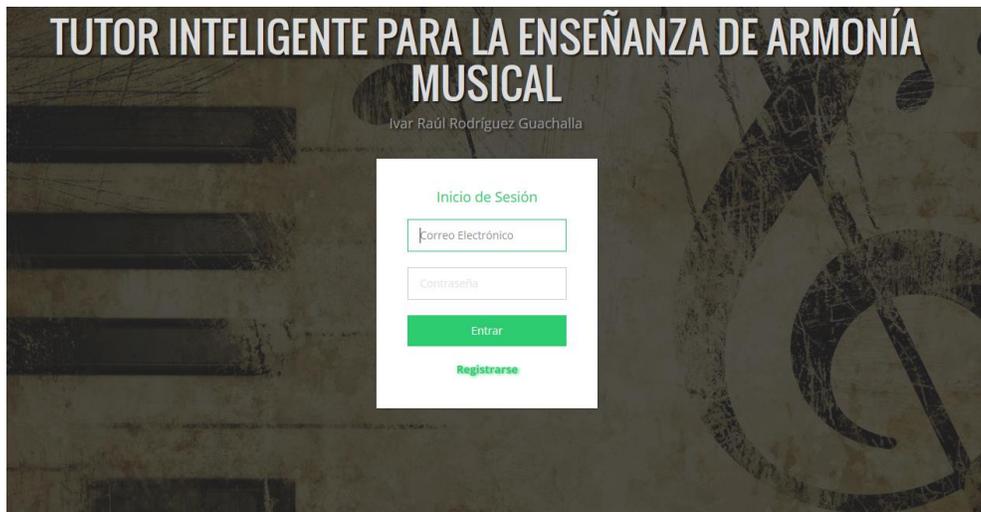


Figura 3.19: Interfaz Registro de Estudiantes

ITERACIÓN 2

- Se desarrolló los casos de uso de Temas. A continuación se mostrarán algunas capturas de pantalla de los mismos.

Visualización de Temas

Cuando el estudiante hace click en la opción de aprendizaje del menú principal, se despliega otro menú, donde están todos los temas, actividades y evaluaciones. Cabe destacar que conforme va avanzando el estudiante se van desbloqueando los temas, actividades y evaluaciones siguientes.



Figura 3.20: Interfaz Vista de temas de acuerdo al avance del Estudiante



Figura 3.21: Interfaz Vista de contenido de los Temas

ITERACIÓN 3

- Se desarrolló los casos de uso de Evaluaciones.

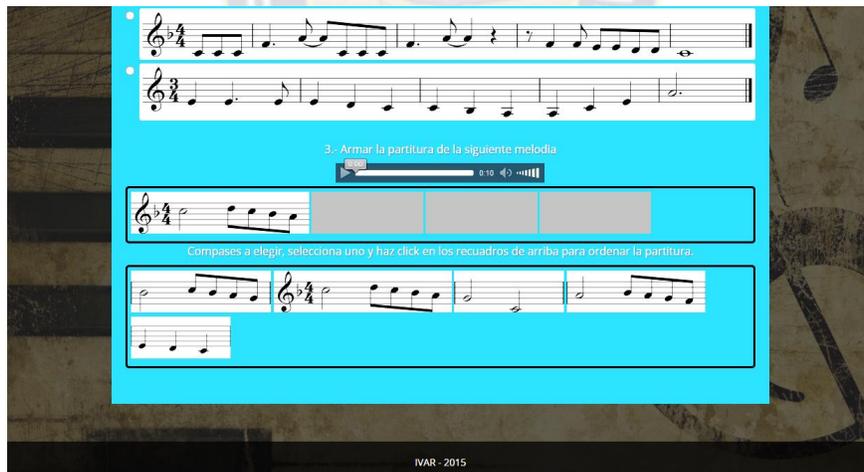


Figura 3.22: Interfaz de Evaluaciones

B. PROBAR LOS COMPONENTES

- i. **MODELO DE PRUEBAS UNITARIAS:** Realizar pruebas de los componentes contra los criterios previamente establecidos. Estas pruebas deben incluir las pruebas del diseño instruccional.

Se realizaron pruebas unitarias constantes por caso de uso: Gestión de Estudiante, Temas y Evaluaciones.

C. INTEGRAR AL DESARROLLO PREVIO

- i. MODELO DE INTEGRACIÓN:** establecer un plan para incorporar el nuevo desarrollo a la liberación previa si es el caso.

En nuestro caso no se incorporó un nuevo desarrollo, por lo cual un modelo de integración no es necesario.

D. REALIZAR PRUEBAS DE INTEGRACIÓN

- i. PRUEBAS DE INTEGRACIÓN**

No se realizarán pruebas de integración debido a que, como anteriormente explicamos, no tendremos un modelo de integración.

4.2.2.3 FASE DE DESPLIEGUE

A. ENTREGAR PRODUCTO AL USUARIO

- i. PRODUCTO:** Se debe entregar el producto debidamente empacado, etiquetado y con información sobre su contenido, aplicación, población objetivo y requerimientos de instalación.

En nuestro caso como el tutor desarrollado no es para un cliente, sino para varios usuarios estudiantes de música, la entrega del producto se lo realizará en línea, es decir, que una vez terminado se lo cargará a un servidor web para su posterior acceso desde internet.

- ii. MANUAL DE USUARIO:** Debe contener información detallada de cómo utilizar el software.

El manual de usuario estará incluido en un enlace dentro el mismo Tutor Web.

iii. **MANUAL DE INSTALACIÓN:** información de los requerimientos para su funcionamiento y procedimiento de instalación.

En este caso no es necesario un manual de instalación, ya que éste es un sistema web.

B. EVALUAR LAS CARACTERÍSTICAS DE CALIDAD Y SATISFACCIÓN DE LOS USUARIOS

MEDICIÓN DE CALIDAD

Para la medición de calidad de software se utilizó la herramienta SEO “SeoQuake”, el cual es una extensión para navegadores web de SEO dirigida principalmente a ayudar a los webmasters que se ocupan de la optimización de los motores de búsqueda y la promoción en Internet de sitios web. SeoQuake permite obtener e investigar muchos otros parámetros de calidad.

Los resultados obtenidos se observan en la figura 3.23, cabe destacar que se trabajó de manera local en el servidor de desarrollo de Django y se publicó en la IP: 172.29.160.182:8000

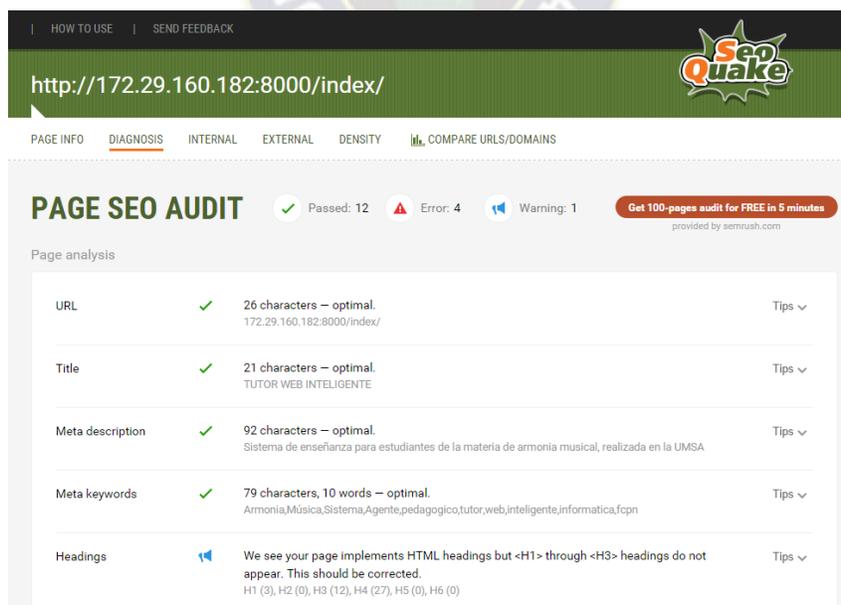


Figura 3.23: Medición de calidad

El diagnóstico de esta herramienta toma en cuenta varios aspectos de sitios web y los analiza para ver si es que existen y estén realizados de manera óptima, entre ellos: la URL, Título, Descripción <meta>, palabras clave <meta>, imágenes, no utilización de Flash y Frames, especificación del idioma, entre otros.

En el diagnóstico se puede observar un resultado de 12 aspectos aceptados; 4 con error; 1 con advertencia. Por lo que podemos decir que el prototipo implementado tiene un nivel aceptable de calidad.

ACEPTACIÓN DEL USUARIO: Realizar pruebas con los usuarios finales y comprobar su grado de satisfacción y efectividad del software.

Se realizó un test de 10 preguntas sobre la usabilidad y satisfacción del prototipo Tutor Web Inteligente (VER ANEXOS) a dos docentes de la materia de armonía musical con la finalidad de hallar el grado de aceptación que presenta el Tutor Web Inteligente.

Se utilizará la escala de Likert, el cual nos permite medir actitudes y conocer el grado de aceptación del encuestado con algunas afirmaciones que le propongamos.

Se usará una escala de cuatro parámetros como son:

- Malo = 1
- Regular = 2
- Bueno = 3
- Muy bueno = 4

La puntuación en la escala de Likert se obtiene sumando los valores obtenidos respecto a cada pregunta, denominado también por ello escala aditiva.

La siguiente tabla muestra el número de marcas en la columna de: muy bueno, bueno, regular y malo. Los resultados fueron los siguientes:

# Marcados	MUY BUENO	BUENO	REGULAR	MALO
DOCENTE 1	5 [marcas]	5 [marcas]		
DOCENTE 2	5 [marcas]	5 [marcas]		

Por ejemplo el docente 1 marcó como MUY BUENO a cinco afirmaciones del test y como BUENO igual a cinco.

- Para ambos Docentes se tiene el mismo resultado, cabe aclarar que los test tienen distintas apreciaciones.

$$\begin{aligned}
 \# \text{ de preguntas con apreciación MUY BUENO:} & \quad 5 \times 4 = 20 \\
 \# \text{ de preguntas con apreciación BUENO:} & \quad 5 \times 3 = 15 \\
 \# \text{ de preguntas con apreciación REGULAR:} & \quad 0 \times 2 = 0 \\
 \# \text{ de preguntas con apreciación MALO:} & \quad 0 \times 1 = 0 \\
 \text{Total} & = 35
 \end{aligned}$$

Con ello se calcula el grado de aceptación:

$$x = \frac{\text{puntuacion}}{\#preguntas} = \frac{35}{10} = 3,5$$

En porcentaje:

$$x_{\%} = \frac{3,5}{4} \times 100 = 87,5 \%$$

Cómo se tomó de muestra a dos docentes a partir de la encuesta, con iguales número apreciaciones por parámetro, el resultado no difiere. Por lo tanto se llega a un grado de aceptabilidad de un 87,5 %.

CAPÍTULO IV

PRUEBA DE HIPÓTESIS

Para probar la hipótesis planteada en el Capítulo I se utilizará el Método de Pruebas t-student de dos muestras independientes, ya que es de tipo cuantitativa y nuestro espacio muestral es pequeño.

5.1 FORMULACIÓN DE LA HIPÓTESIS

Debemos formular una hipótesis nula (H_0) y una Hipótesis alternativa (H_1) para luego del análisis aceptar o rechazar la Hipótesis Nula.

H_0 : El Tutor Web Inteligente para la enseñanza de Armonía Musical utilizando tecnologías multimedia no mejora el rendimiento educativo en estudiantes de la materia de armonía musical en un 15%.

H_1 : El Tutor Web Inteligente para la enseñanza de Armonía Musical utilizando tecnologías multimedia mejora el rendimiento educativo en estudiantes de la materia de armonía musical en un 15%.

5.2 TAMAÑO DE LA MUESTRA

Se presentó un mismo test de conocimientos en armonía musical a dos grupos de estudiantes los cuales se distinguen por:

- a) Estudiantes con conocimientos en armonía musical que realizaron el test sin antes utilizar el Tutor Web Inteligente.

- b) Estudiantes con conocimientos en armonía musical que realizaron el test con previa utilización del Tutor Web Inteligente.

Cada grupo constó de 15 personas, por lo que el tamaño muestral es de 30 personas.

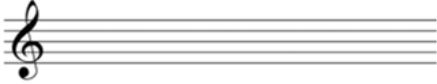
5.3 TEST DE CONOCIMIENTO DE ARMONÍA MUSICAL

El test que se realizó para medir los conocimientos de los estudiantes con y sin Tutor previo fue el siguiente:

TEST DE CONOCIMIENTO DE ARMONÍA MUSICAL

I. ESCALAS

1. Construya la escala de sol menor armónica.



2. Escuche la siguiente escala e identifique el tipo de escala.
Resp.

II. INTERVALOS

1. Construya el intervalo de 6m ascendente a partir de Fa#



2. Escuche el siguiente intervalo e identificar cual es.
Resp.

III. TRIADAS Y CUATRIADAS

1. Construya el acorde de Sol Mayor



2. Escuche el siguiente acorde e identifique qué tipo es y su grado de inversión (si lo tiene)
Resp.

IV. FUNCIONES ARMÓNICAS

1. ¿Cómo se llama el 7mo grado de la escala?
Resp.
2. ¿Cuál es el 5to grado de la escala de mi menor?
Resp.

V. PROGRESIONES ARMÓNICAS

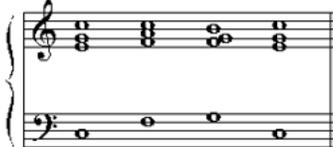


Figura 4.1: Test de conocimientos de armonía musical

La puntuación de cada pregunta del test es de 10 pts a excepción de la última que tiene el valor 20 pts. Siendo un total de 100 pts sobre el cual se evaluará el test.

5.4 PROCEDIMIENTO

El primer grupo de estudiantes, los cuales dieron el test sin utilizar el tutor web, Resultados del test mostrados en la tabla 5.2:

# ESTUDIANTE	CALIFICACIÓN	PREGUNTAS CONTESTADAS
1	60	6
2	85	9
3	65	7
4	45	5
5	75	8
6	70	7
7	60	6
8	60	6
9	60	6
10	75	8
11	65	6
12	60	5
13	65	7
14	70	6
15	70	7

Tabla 4.2: Resultados Test Sin Tutor

Tanto en la tabla 5.2 como en la 5.3 la calificación se realizó según la valoración de cada pregunta expuesta en el test, existieron casos en los cuales hubo preguntas parcialmente respondidas, éstas fueron calificadas sobre la mitad.

En la tabla 4.3 observamos los resultados obtenidos por los estudiantes que utilizaron el Tutor Web Inteligente. El número de estudiantes al igual que en el primer grupo es de 15.

# ESTUDIANTE	CALIFICACIÓN	PREGUNTAS CONTESTADAS
1	100	9
2	90	8
3	95	9
4	85	8
5	85	8
6	95	9
7	85	8
8	85	8
9	95	9
10	80	9
11	85	8
12	100	9
13	90	8
14	95	9
15	80	8

Tabla 4.3: Resultados del test con Tutor

Estadístico a utilizar:

$$t_0 = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{s^2 \times \left[\frac{1}{n_1} + \frac{1}{n_2} \right]}}$$

Con $n_1 + n_2 - 2$ grados de libertad.

Varianza común estimada:

$$s^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{(n_1 + n_2 - 2)}$$

Donde: n_1 y n_2 =tamaños muestrales

\bar{x}_1 y \bar{x}_2 = Promedio de las muestras

CALCULANDO PROMEDIOS

$$\bar{x}_1 = \frac{\sum_1^n x_i}{n} = \frac{460}{7} = 65,66666667$$

$$\bar{x}_2 = \frac{\sum_1^n x_i}{n} = \frac{635}{7} = 89,66666667$$

CALCULANDO VARIANZAS

$$s_1^2 = \frac{\sum(x_i - \bar{x})^2}{n - 1} = \frac{1193,33333}{14} = 85,2380952$$

$$s_2^2 = \frac{\sum(x_i - \bar{x})^2}{n - 1} = \frac{623,333333}{14} = 44,5238095$$

VARIANZA COMÚN ESTIMADA

$$s^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{(n_1 + n_2 - 2)} = \frac{(14)85,2380952 + (14)44,5238095}{28} = 64,8809524$$

CÁLCULO DEL ESTADÍSTICO t

$$t_{calculado} = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{s^2 \times \left[\frac{1}{n_1} + \frac{1}{n_2}\right]}} = \frac{65,66 - 89,66}{\sqrt{64,8809524 \times \left(\frac{1}{15} + \frac{1}{15}\right)}} = -8,15987048$$

Se toma un nivel de significancia o de confianza: 0.05 con grados de libertad: $15+15-2=28$

Ahora que tenemos todos los resultados vemos la tabla de la t Student para evaluar nuestros resultados, se busca el valor el cual vamos a comparar, el que calculamos basándonos en el nivel de confianza elegido (0.05) y así también los grados de libertad “gl” igual a 28.

Al buscar en la tabla “t” de Student tenemos

gl	$\alpha/2$ (0.01)
28	$t_{\text{tabulado}} = 2,467$

5.5 REGLA DE DECISIÓN

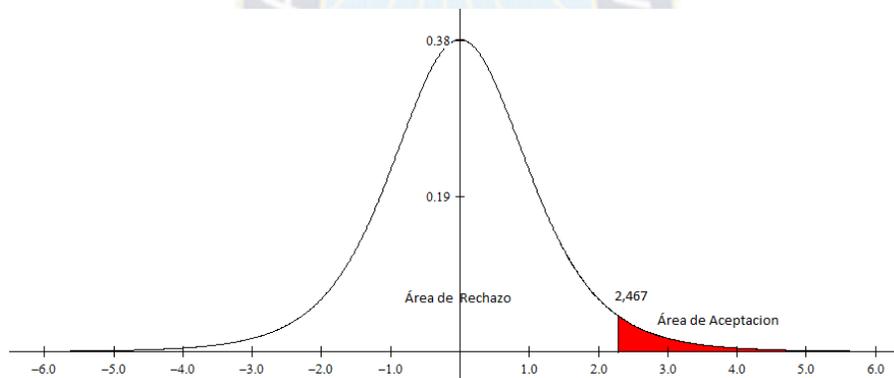


Fig 4.1: Gráfica t-student para prueba de hipótesis

Cómo $t_{\text{calculado}} < t_{\text{tabulado}} \Rightarrow -8,1598 < 2,467$ se rechaza H_0 y se acepta la Hipótesis alternativa.

CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

Terminado los capítulos de la presente investigación, se concluye que se logró alcanzar los objetivos planteados.

Se implementó un prototipo Tutor Web Inteligente, el cual ayuda al proceso de enseñanza en la materia de Armonía musical utilizando tecnologías multimedia, logrando éste un grado de aceptación del 87.5%.

- Se logró brindar al estudiante un banco de melodías y ritmos asociadas a su escritura en un pentagrama para su posterior identificación en varios niveles de dificultad, utilizando la herramienta de escritura musical Sibelius 7.5, el cual nos da la facilidad de escribir música y exportar el mismo tanto en audio como partitura en varios formatos: pdf, png, entre otros.
- El prototipo se adapta al estilo de aprendizaje de cada estudiante, para lo cual se aplicó un test de estilos de aprendizaje al iniciar el Tutor Web Inteligente.
- Se alcanzó a realizar ejemplos y ejercicios audiovisuales clasificados por tema de manera que el estudiante use el sentido del oído para la resolución de los mismos.
- Se logró minimizar el contenido teórico de la materia, para aplicarlo al tutor inteligente de manera que se da énfasis en la práctica del estudiante. Esto gracias a la ayuda de un experto en el área musical.

6.2 RECOMENDACIONES

- Es importante que el estudiante, para la utilización del Tutor Web, tenga un conocimiento básico en teoría musical, ya que éste utiliza terminología musical suponiendo que el estudiante los conoce.
- Utilizar navegadores web no obsoletas, compatibles con HTML5.
- Para una mejor interacción con el usuario, en futuras investigaciones en este campo de estudio (música), se puede tomar en cuenta, la posibilidad de utilizar herramientas web como ser librerías JavaScript que permitan la generación de partituras a partir de un script, el uso de otras técnicas de Inteligencia Artificial en el campo de la educación, entre otros.



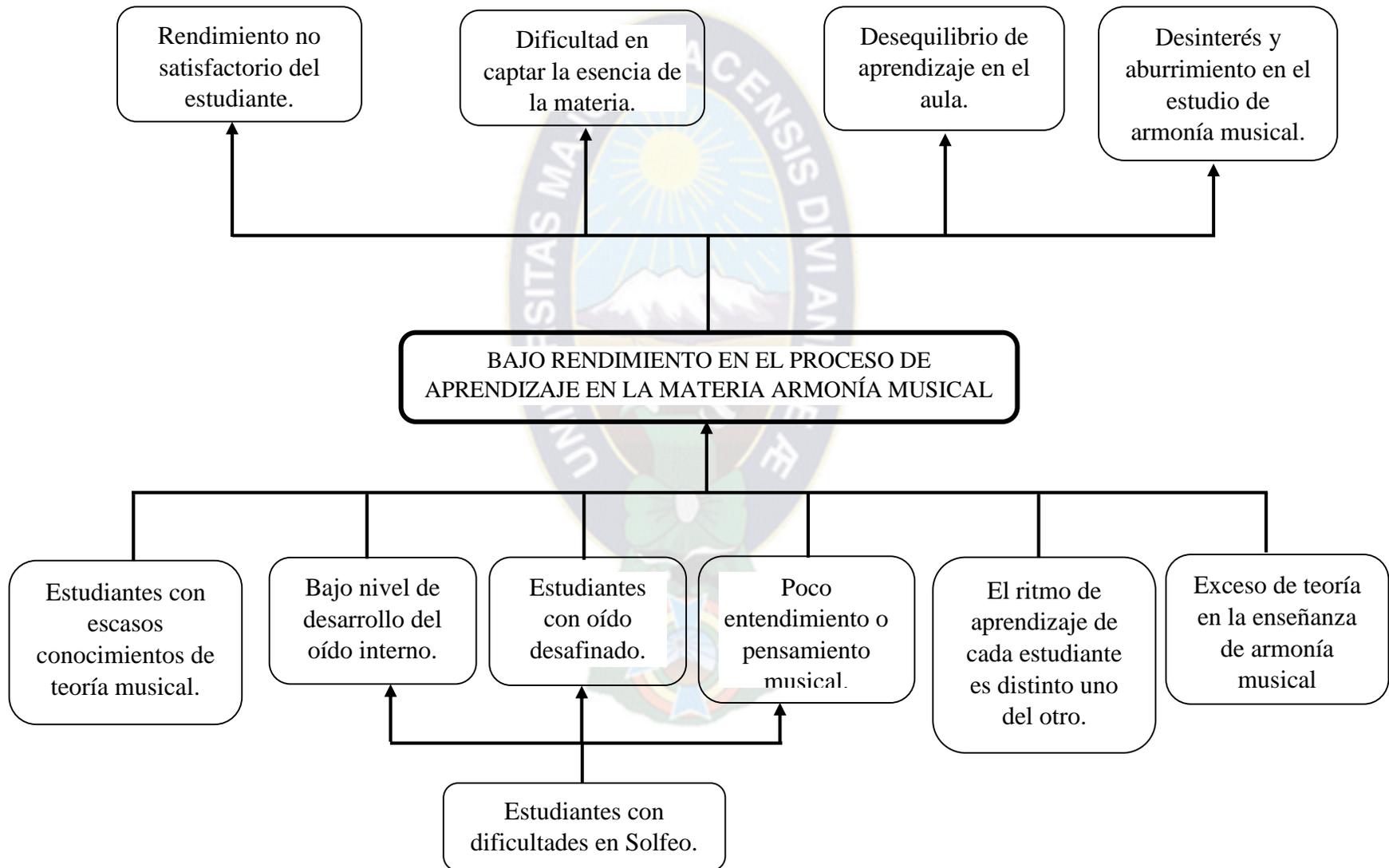
BIBLIOGRAFÍA

- CHOQUE, G. Agentes Pedagógicos Para La Enseñanza De Las Redes Neuronales. Revista Científica CEPIES, 2009, 1(1):53-64.
- DUARTE, Y. y LEGRÁ, Y. 2014. Los Sistemas Tutores Inteligentes. Revista TINO (41): 6-8.
- GONZALES, S. 2004. Sistema Tutor Inteligente Arquitectura General de Sistema Tutor Inteligente.
- HENANDEZ, J. Revista Estilos de Aprendizaje, nº4, Vol 2, Octubre de 2009
- JONNATHAN. 2012. Inteligencia Artificial en la Educación. [en línea] <http://www.monografias.com/trabajos93/inteligencia-artificial-educacion/inteligencia-artificial-educacion.shtml> [consulta: 10 de mayo 2015]
- LAGE, F & CATALDI, Z. 2014. Modelo de sistema tutor distribuido para educación a distancia.
- LOAIZA, R. De la información a la Informática [En línea], [Consultado 27 de septiembre de 2015] , disponible en http://www.bvs.sld.cu/revistas/san/vol2_2_98/san15298.htm
- ORANTES, A. 2002. Educación y Computación. Venezuela. Universidad Central de Venezuela.
- PISTON, W. 1998. Armonía. 5ta Ed. EE.UU., SpanPress Universitaria.

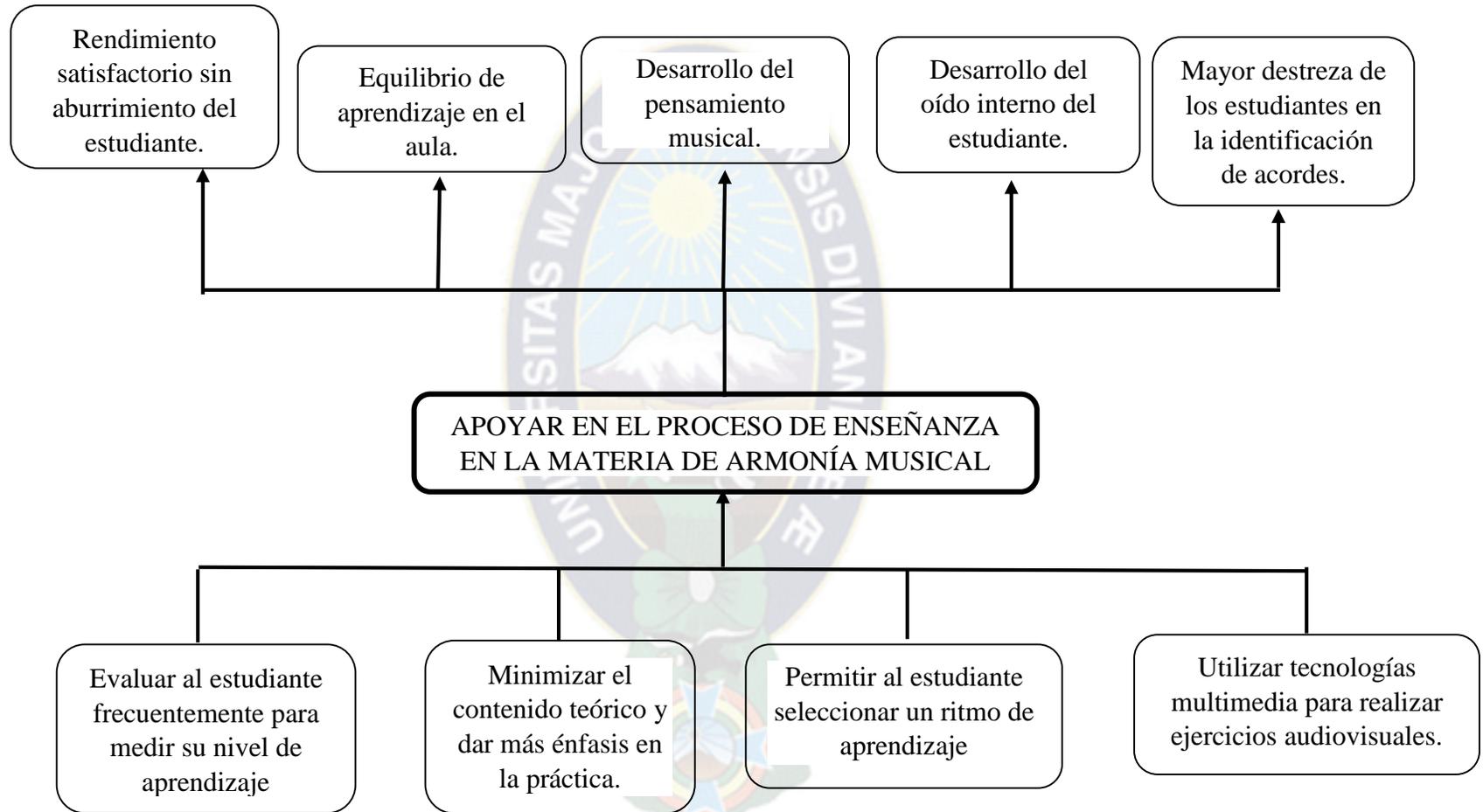
- RUSSELL S. y NORVIG, P. 2004 Inteligencia Artificial un Enfoque Moderno 2da Ed., Pearson Prentice Hall
- SCHOENBERG, A. 1979. Tratado de Armonía. Madrid (España), Real Musical.
- PONCE, P. 2010. Inteligencia Artificial con aplicaciones a la ingeniería, AlfaOmega.
- TRONCOSO, B. 2004. Aplicaciones de Agentes Pedagógicos en Entornos Virtuales para la Enseñanza.



ANEXO A: ÁRBOL DE PROBLEMAS



ANEXO B: ÁRBOL DE OBJETIVOS



ANEXO C: MARCO LÓGICO

RESUMEN NARRATIVO	INDICADORES	MEDIOS DE VERIFICACIÓN	SUPUESTOS
<p>FIN</p> <p>Diseñar e implementar un Tutor Web Inteligente que apoye al proceso de enseñanza en la materia de armonía musical utilizando tecnologías multimedia.</p>	Profesores y estudiantes interesados en el STI comparando con el contenido de la materia y brindarán sus críticas al mismo, así enriquecer el contenido del STI.	Entrevistas. Informes de Avance. Fotografías.	Se tiene información sobre las instituciones que hacen uso de las TIC en la enseñanza.
<p>PROPÓSITO</p> <p>Apoyar al proceso de enseñanza de la materia de Armonía Musical de manera que se mejore el rendimiento académico de los estudiantes.</p>	La finalización del diseño e implementación del STI como recurso didáctico en la enseñanza de la Armonía Musical, será hasta la fecha 12/11/15.	Avances del proyecto publicados en un blog. Presentación de prototipos.	Existen tesis elaboradas acerca de STI para diferentes áreas del conocimiento y se cuenta con herramientas para ello.
<p>PRODUCTOS</p> <ul style="list-style-type: none"> • Preparación en la materia de Armonía Musical. • Tutor inteligente implementado bajo plataforma Web con ejercicios audiovisuales utilizando tecnologías multimedia. 	<ul style="list-style-type: none"> • Lectura de libros y tutoriales para el aprendizaje de Armonía Musical. 31/07/15 • Desarrollo de los módulos del STI, entorno gráfico y pruebas. 12/11/15 	Encuestas. Libros utilizados durante el proyecto. Informes de evaluación en el aula.	Se tiene información en internet sobre la materia de Armonía Musical, pero la mejor es de libros porque es más completo.
<p>ACTIVIDADES</p> <ul style="list-style-type: none"> • Investigar el contenido de la materia de Armonía Musical y requisitos previos al mismo. • Investigar y diseñar la estructura de los STI. • Diseñar los ejercicios audiovisuales del STI. • Implementar prototipos de STI con una interfaz web. • Presentación del STI a estudiantes de la Escuela Nacional de Música “Luis Felipe Arce” que están cursando la materia. 	<ul style="list-style-type: none"> • 6 días 1600 Bs. • 20 días 2000 Bs. • 12 días 1500 Bs. • 40 días 3000 Bs. • 2 días 900 Bs. 	Facturas de impresiones de informes, viáticos, materiales de escritorio. Factura de acceso a internet. Fotocopias de documentos e información.	Disponibilidad de bibliografía para la enseñanza de las herramientas tecnológicas. Recopilar la información necesaria para la redacción de la tesis de forma adecuada.
	80 días 9000 Bs.		

ANEXO D: CORRIENDO EL SERVIDOR DE DESARROLLO DE DJANGO

```
C:\Users\IVAR\Documents\python\twi_armonia>python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
December 08, 2015 - 18:29:49
Django version 1.8.3, using settings 'twi_armonia.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[08/Dec/2015 18:31:28]"GET / HTTP/1.1" 302 0
[08/Dec/2015 18:31:28]"GET /login/ HTTP/1.1" 200 3163
[08/Dec/2015 18:31:29]"GET /static/bootstrap/css/bootstrap.min.css HTTP/1.1" 200 113498
[08/Dec/2015 18:31:29]"GET /static/css/font-awesome.min.css HTTP/1.1" 200 21984
[08/Dec/2015 18:31:29]"GET /static/js/jquery.appear.js HTTP/1.1" 200 3337
[08/Dec/2015 18:31:29]"GET /static/css/animate.css HTTP/1.1" 200 67721
[08/Dec/2015 18:31:29]"GET /static/js/jquery-2.1.1.min.js HTTP/1.1" 200 84245
[08/Dec/2015 18:31:29]"GET /static/js/contact_me.js HTTP/1.1" 200 2928
[08/Dec/2015 18:31:29]"GET /static/css/login.css HTTP/1.1" 200 2166
[08/Dec/2015 18:31:29]"GET /static/bootstrap/js/bootstrap.min.js HTTP/1.1" 200 35601
[08/Dec/2015 18:31:29]"GET /static/js/modernizr.custom.js HTTP/1.1" 200 29816
[08/Dec/2015 18:31:29]"GET /static/css/style.css HTTP/1.1" 200 19756
[08/Dec/2015 18:31:30]"GET /static/images/fondo_armonia.jpg HTTP/1.1" 200 567018
[08/Dec/2015 18:31:31]"GET /favicon.ico HTTP/1.1" 404 4069
[08/Dec/2015 18:31:31]"GET /favicon.ico HTTP/1.1" 404 4069
```

Ready



ANEXO E: EJERCICIOS DE EVALUACIONES DEL TUTOR

4.- ¿A qué escala le pertenece esta armadura?



- Escala de Re Mayor
- Escala de Sol menor
- Escala de Mi Mayor

5.- Armar la partitura de la escala de Mi menor

Compases a elegir, selecciona uno y haz click en los recuadros de arriba para ordenar la partitura.



Calcular



ANEXO F: INTERFAZ DE LAS EVALUACIONES EN DISPOSITIVO MÓVIL

0:00 0:06

- Escala Mayor
- Escala menor Armónica
- Escala menor Natural

2.- Escuche la escala e identifique el tipo de escala

0:00 0:06

- Escala Mayor
- Escala menor Armónica
- Escala menor Natural

3.- Escuche la escala y seleccionar la partitura a la que corresponde

0:00 0:10

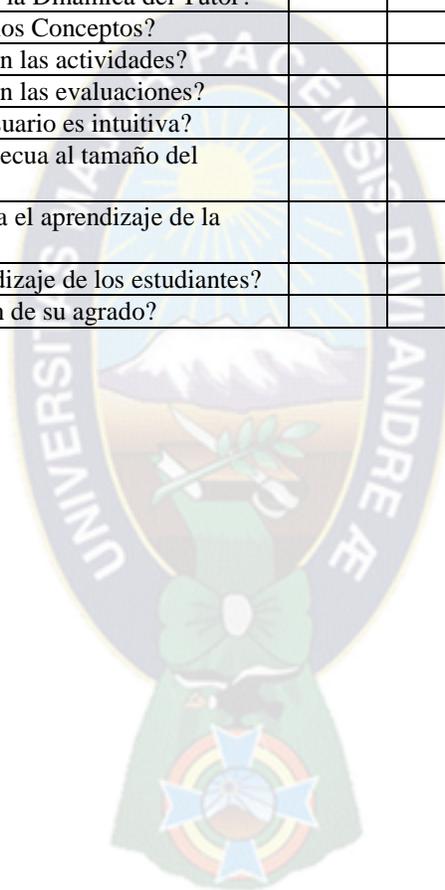
-
-
-



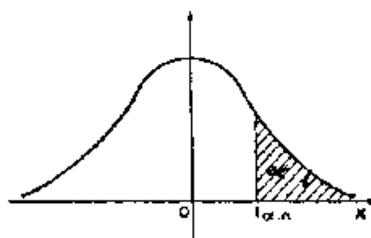
ANEXO G: TEST DE EVALUACIÓN DE USABILIDAD DEL TUTOR

Evaluación al Tutor Web Inteligente

	Muy Bueno	Bueno	Regular	Malo
1 ¿El Tutor contiene Teoría y Evaluación?				
2 ¿Cómo le pareció la Dinámica del Tutor?				
3 ¿Están correctos los Conceptos?				
4 ¿Qué le parecieron las actividades?				
5 ¿Qué le parecieron las evaluaciones?				
6 ¿La interfaz de usuario es intuitiva?				
7 ¿La interfaz se adecua al tamaño del dispositivo?				
8 ¿Es un apoyo para el aprendizaje de la materia?				
9 ¿Motiva el aprendizaje de los estudiantes?				
10 ¿Los colores son de su agrado?				



ANEXO H: TABLA DE DISTRIBUCIÓN T - STUDENT



$\alpha/2$ gl	0,40	0,30	0,20	0,10	0,050	0,025	0,010	0,005	0,001	0,0005
1	0,325	0,727	1,376	3,078	6,314	12,71	31,82	63,66	318,3	636,6
2	0,289	0,617	1,061	1,886	2,920	4,303	6,963	9,925	22,33	31,60
3	0,277	0,584	0,978	1,638	2,353	3,182	4,541	5,841	10,22	12,94
4	0,271	0,569	0,941	1,533	2,132	2,776	3,747	4,604	7,173	8,610
5	0,267	0,559	0,920	1,476	2,015	2,571	3,365	4,032	5,893	6,859
6	0,265	0,553	0,906	1,440	1,943	2,447	3,143	3,707	5,208	5,959
7	0,263	0,549	0,896	1,415	1,895	2,365	2,998	3,499	4,785	5,405
8	0,262	0,546	0,889	1,397	1,860	2,306	2,896	3,355	4,501	5,041
9	0,261	0,543	0,883	1,383	1,833	2,262	2,821	3,250	4,297	4,781
10	0,260	0,542	0,879	1,372	1,812	2,228	2,764	3,169	4,144	4,587
11	0,260	0,540	0,876	1,363	1,796	2,201	2,718	3,106	4,025	4,437
12	0,259	0,539	0,873	1,356	1,782	2,179	2,681	3,055	3,930	4,318
13	0,259	0,538	0,870	1,350	1,771	2,160	2,650	3,012	3,852	4,221
14	0,258	0,537	0,868	1,345	1,761	2,145	2,624	2,977	3,787	4,140
15	0,258	0,536	0,866	1,341	1,753	2,131	2,602	2,947	3,733	4,073
16	0,258	0,535	0,863	1,337	1,746	2,120	2,583	2,921	3,686	4,015
17	0,257	0,534	0,863	1,333	1,740	2,110	2,567	2,898	3,646	3,965
18	0,257	0,534	0,862	1,330	1,734	2,101	2,552	2,878	3,611	3,922
19	0,257	0,533	0,861	1,328	1,729	2,093	2,539	2,861	3,579	3,883
20	0,257	0,533	0,860	1,325	1,725	2,086	2,528	2,845	3,552	3,850
21	0,257	0,532	0,859	1,323	1,721	2,080	2,518	2,831	3,527	3,819
22	0,256	0,532	0,858	1,321	1,717	2,074	2,508	2,819	3,505	3,792
23	0,256	0,532	0,858	1,319	1,714	2,069	2,500	2,807	3,485	3,767
24	0,256	0,531	0,857	1,318	1,711	2,064	2,492	2,797	3,467	3,745
25	0,256	0,531	0,856	1,316	1,708	2,060	2,485	2,787	3,450	3,725
26	0,256	0,531	0,856	1,315	1,706	2,056	2,479	2,779	3,435	3,707
27	0,256	0,531	0,855	1,314	1,703	2,052	2,473	2,771	3,421	3,690
28	0,256	0,530	0,855	1,313	1,701	2,048	2,467	2,763	3,408	3,674
29	0,256	0,530	0,854	1,311	1,699	2,045	2,462	2,756	3,396	3,659
30	0,256	0,530	0,854	1,310	1,697	2,042	2,457	2,750	3,385	3,646
40	0,255	0,529	0,851	1,303	1,648	2,021	2,423	2,704	3,307	3,551
50	0,255	0,528	0,849	1,298	1,676	2,009	2,403	2,678	3,262	3,495
60	0,254	0,527	0,848	1,296	1,671	2,000	2,390	2,660	3,232	3,460
80	0,254	0,527	0,846	1,292	1,664	1,990	2,374	2,639	3,195	3,415
100	0,254	0,526	0,845	1,290	1,660	1,984	2,365	2,626	3,174	3,389
200	0,254	0,525	0,843	1,286	1,653	1,972	2,345	2,601	3,131	3,339
500	0,253	0,525	0,842	1,283	1,648	1,965	2,334	2,586	3,106	3,310
∞	0,253	0,524	0,842	1,282	1,645	1,960	2,326	2,576	3,090	3,291