



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA

CARRERA DE COMPUTACIÓN

TEMA:

**Diseño de un protipo de tutor inteligente Para aprendizaje en
lenguajes de programación**

AUTOR:

Mora Carrión, Guillermo Enrique

**Trabajo de titulación previo a la obtención del título de
INGENIERO EN CIENCIAS DE LA COMPUTACIÓN**

TUTOR:

Ing. Cornejo Gómez, Galo Enrique Mgs

Guayaquil, Ecuador

8 de marzo de 2022



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACIÓN

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por **Mora Carrión Guillermo Enrique**, como requerimiento parcial para la obtención del Título de **Ingeniero en Ciencias de la Computación**.

TUTOR

Ing. Galo Cornejo Gómez, Mgs.

Guayaquil, a los 8 días del mes de marzo del año 2022



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACION

DECLARACIÓN DE RESPONSABILIDAD

Yo, Guillermo Enrique Mora Carrión

DECLARO QUE:

El Trabajo de Titulación, **Diseño de un protipo de tutor inteligente Para aprendizaje en lenguajes de programación**, previo a la obtención del título de **Ingeniero en Ciencias de la Computación** ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, la veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 8 días del mes de marzo del año 2022

EL AUTOR

Guillermo Enrique Mora Carrión



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

**FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACION**

AUTORIZACIÓN

Yo, Guillermo Enrique Mora Carrión

Autorizo a la Universidad Católica Santiago de Guayaquil a la publicación en la biblioteca de la institución del Trabajo de Titulación: **Diseño de un protipo de tutor inteligente Para aprendizaje en lenguajes de programación**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 8 días del mes de marzo del año 2022

EL AUTOR

Guillermo Enrique Mora Carrión



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

**FACULTAD DE INGENIERÍA
CARRERA DE COMPUTACION**

REPORTE URKUND

URKUND	
Documento	Tesis Guillermo Mora.docx (D129242275)
Presentado	2022-03-02 10:41 (-05:00)
Presentado por	guillermomoracarrion@gmail.com
Recibido	galo.cornejo.ucsg@analysis.orkund.com
	4% de estas 47 páginas, se componen de texto presente en 15 fuentes.

Fecha de elaboración: 02-03-2022


Nombre del tutor y firma: GALO CORNEJO GÓMEZ

AGRADECIMIENTO

En primer lugar, agradezco a Dios por bendecirme con inteligencia, sabiduría, perseverancia y capacidad para sobreponerme a cada obstáculo que se me presentó en esta etapa universitaria.

Agradezco a mis padres por ser los pilares, la base esencial y fundamental de mi vida, que con esfuerzo, trabajo, sacrificio y tiempo me han dado la oportunidad de estudiar la carrera que siempre soñé ejercer.

Finalmente agradezco a cada persona con la que compartí momentos de enseñanza y aprendizaje en la Facultad de Ingeniería, compañeros, docentes, amigos, etc. Por formar parte de este gran camino, porque no ha hecho más que empezar.

Guillermo Mora C.

DEDICATORIA

Quiero dedicar este trabajo de titulación, con mucho cariño a mis padres, y mi hermano quienes siempre estuvieron conmigo en las buenas, en las malas y en las peores, dándome el apoyo y la motivación necesaria para lograr este objetivo tan anhelado por mí. A mi familia y amigos cercanos, Desiderio, Isabel, Allison, Alex, Rebeca, Marianella, Ricardo, Fabiola, Dome, Luis, Pamela, Arianna, entre otros que son parte de mi corazón, y que siempre creyeron en mí, y que nunca dudaron en que conseguiría esta meta.

Guillermo Mora C.

INDICE GENERAL

INDICE DE FIGURAS	XIV
RESUMEN.....	XVIII
ABSTRACT.....	XIX
INTRODUCCIÓN	2
CAPITULO I: EL PROBLEMA	4
1.1 PLANTEAMIENTO DEL PROBLEMA	4
1.2 JUSTIFICACIÓN	4
1.3 ALCANCE	5
1.4 OBJETIVOS	6
1.4.1 OBJETIVO GENERAL.....	6
1.4.2 OBJETIVOS ESPECÍFICOS.....	6
1.5 PREGUNTA DE INVESTIGACIÓN	6
CAPITULO II MARCO TEORICO, CONCEPTUAL Y LEGAL	7
2.1 MARCO TEÓRICO	7
2.1.1 INGENIERÍA DE SOFTWARE EDUCATIVO (ISE).....	7
2.1.1.1 METODOLOGIA DE INGENIERÍA DE SOFTWARE EDUCATIVO	7
2.1.2 TUTOR.....	8
2.1.2.1 CAPACIDADES DE UN TUTOR.....	8
2.1.3 LA INTELIGENCIA ARTIFICIAL.....	8
2.1.4 SISTEMA DE TUTOR INTELIGENTE (STI).....	9
2.1.4.1 ARQUITECTURA DE LOS SISTEMAS INTELIGENTES.....	11
2.1.4.1.1 MÓDULO DEL DOMINIO	12
2.1.4.1.1.1 PARÁMETROS BÁSICOS DEL SISTEMA	13
2.1.4.1.1.2 BASE DE CONOCIMIENTOS.....	13
2.1.4.1.1.3 ELEMENTOS DIDÁCTICOS	15
2.1.4.1.2 MÓDULO DEL TUTOR	16
2.1.4.1.2.1 SUB-MODULO PROTOCOLOS PEDAGOGICOS.....	16
2.1.4.1.2.1.1 ANALIZADOR DE PERFIL.....	17
2.1.4.1.2.1.2 PROTOCOLO PEDAGÓGICO	17
2.1.4.1.2.1.2.1 PROTOCOLO PEDAGÓGICO GENERAL/TRADICIONAL	18
2.1.4.1.2.1.2.2 PROTOCOLO DE APRENDIZAJE BASADO EN PROYECTOS (ABP)	18
2.1.4.1.2.1.2.3 FLIPPED CLASSROOM (AULA INVERTIDA).....	18

2.1.4.1.3 MÓDULO DEL ESTUDIANTE.....	19
2.1.4.1.4 MÓDULO DE INTERFAZ.....	20
2.1.5 APRENDIZAJE	20
2.1.5.1 PROCESO ENSEÑANZA – APRENDIZAJE.....	20
2.1.5.2 ESTILOS DE APRENDIZAJE	20
2.1.5.2.1 ACTIVOS	21
2.1.5.2.2 REFLEXIVOS.....	21
2.1.5.2.3 TEÓRICO.....	22
2.1.5.2.4 PRAGMÁTICO.....	22
2.1.5.3 TIPOS DE APRENDIZAJE	23
2.1.5.3.1 APRENDIZAJE IMPLÍCITO	23
2.1.5.3.2 APRENDIZAJE EXPLÍCITO.....	23
2.1.5.3.3 APRENDIZAJE ASOCIATIVO.....	23
2.1.5.3.4 APRENDIZAJE NO ASOCIATIVO.....	23
2.1.5.3.5 APRENDIZAJE SIGNIFICATIVO	23
2.1.5.3.6 APRENDIZAJE COOPERATIVO	24
2.1.5.3.7 APRENDIZAJE COLABORATIVO	24
2.1.5.3.8 APRENDIZAJE EMOCIONAL	24
2.1.5.3.9 APRENDIZAJE OBSERVACIONAL.....	24
2.1.5.3.10 APRENDIZAJE EXPERIENCIAL.....	25
2.1.5.3.11 APRENDIZAJE POR DESCUBRIMIENTO	25
2.1.5.3.12 APRENDIZAJE MEMORÍSTICO.....	25
2.1.5.3.13 APRENDIZAJE RECEPTIVO	25
2.1.6 LENGUAJES DE PROGRAMACIÓN.....	25
2.1.6.1 LENGUAJES DE PROGRAMACIÓN DE BAJO NIVEL.....	26
2.1.6.2 LENGUAJES DE PROGRAMACIÓN DE ALTO NIVEL.....	26
2.1.7 APRENDIZAJE DE LOS LENGUAJES DE PROGRAMACIÓN.....	27
2.1.7.1 ¿COMO SE PUEDEN APRENDER LOS LENGUAJES DE PROGRAMACION?	27
2.1.7.2 ¿QUÉ ES LO QUE SE APRENDE DE LOS LENGUAJES DE PROGRAMACION?	28
2.1.8 ENSEÑANZA DE LOS LENGUAJES DE PROGRAMACIÓN.....	29
2.1.8.1 ¿COMO SE ENSEÑAN LOS LENGUAJES DE PROGRAMACIÓN?.....	32
2.1.9 SISTEMAS ADAPTATIVOS DE EVALUACIÓN DE CONOCIMIENTO	32

2.1.9.1 APRENDIZAJE ADAPTATIVO	32
2.1.9.2 EVALUACIÓN ADAPTATIVA	32
2.1.9.3 APRENDIZAJE Y EVALUACIÓN ADAPTATIVA	33
2.1.9.3.1 MODELOS DE APRENDIZAJE ADAPTATIVO	33
2.1.9.4 SISTEMAS ADAPTATIVOS DE APRENDIZAJE Y EVALUACIÓN	34
2.1.9.4.1 EXÁMENES ADAPTATIVOS (INFORMÁTICA)	34
2.1.9.4.2 TUTORIALES ADAPTABLES O ADAPTATIVOS	34
2.2 MARCO CONCEPTUAL	34
2.2.1 HERRAMIENTAS DE PROGRAMACIÓN	34
2.2.1.1 VISUAL STUDIO 2019	34
2.2.1.2 .NET	35
2.2.1.2.1 LENGUAJES DE PROGRAMACION EN .NET	35
2.2.1.2.1.1 C#	35
2.2.1.2.1.2 F#	35
2.2.1.2.1.3 VISUAL BASIC	35
2.2.1.2.1.2 BOOTSTRAP	35
2.2.1.3 APLICACIONES WEB	36
2.2.2 BASE DE DATOS	36
2.2.2.1 MySQL	36
2.2.2.1.1 SQL	36
CAPITULO III: METODOLOGIA DE INVESTIGACION Y DESARROLLO	37
3.1 METODOLOGIA DE LA INVESTIGACION	37
3.1.1 ENFOQUE DE LA INVESTIGACION	37
3.1.1.1 INVESTIGACION CUALITATIVA	38
3.1.2 TIPO DE INVESTIGACION	39
3.1.3 METODO DE INVESTIGACION	40
3.1.3.1 MÉTODO EXPERIMENTAL	40
3.1.4 TECNICA DE INVESTIGACION	40
3.1.4.1 ENTREVISTA	40
3.1.4.2 TECNICAS DE ENTREVISTA	41
3.1.5 ANALISIS DE RESULTADOS	42
3.2 METODOLOGIA DE DESARROLLO	43
3.2.1 METODOLOGIA DE PROTOTIPADO RAPIDO	43

CAPITULO IV: DISEÑO DEL PROTOTIPO	46
4.1 ARQUITECTURA DEL PROTOTIPO	46
4.1.1 DIAGRAMA DEL PROTOTIPO	46
4.1.2 COMPONENTES Y DOCUMENTOS FUNCIONALES	46
4.1.2.1 MODULO DE DOMINIO	46
4.1.2.1.1 PARÁMETROS DEL SISTEMA	47
4.1.2.1.2 BASE DE CONOCIMIENTOS	47
4.1.2.1.3 REGLAS DE LA BASE DE CONOCIMIENTOS	49
4.1.2.1.4 RECURSOS PEDAGÓGICOS	49
4.1.2.1.5 BANCO DE PREGUNTAS	50
4.1.2.2 MODULO TUTOR	50
4.1.2.2.1 PROTOCOLO PEDAGÓGICO	50
4.1.2.2.1.1 CONSIDERACIONES	51
4.1.2.2.1.2 REGLAS DEL SISTEMA	52
4.1.2.2.1.3 MAQUINA DE INFERENCIA	53
4.1.2.2.1.4 SEGUIMIENTO	53
4.1.2.2.1.5 AGREGAR RECURSOS	53
4.1.2.2.1.6 PARÁMETROS – CONTENIDO	54
4.1.2.2.2 ANALIZADOR DEL PERFIL	54
4.1.2.2.2.1 MODELO MATEMATICO	54
4.1.2.2.2.1.1 NIVELES	54
4.1.2.2.2.1.2 VARIABLES	54
4.1.2.2.2.1.3 MODELO	54
4.1.2.2.2.1.4 EJEMPLO DE MODELO MATEMATICO	55
4.1.2.2.2.2 CONTROL DE LA REVISION DE RECURSOS Y DESARROLLO DE EJERCICIOS	56
4.1.2.3 MODELO ESTUDIANTE	56
4.1.2.3.1 DIAGNOSTICAR CONOCIMIENTOS	57
4.1.2.3.2 VISUALIZAR PROGRESO	57
4.1.2.3.3 VISUALIZACIÓN DE CONTENIDOS Y RECURSOS	57
4.1.2.3.4 DESCARGA DE RECURSOS	58
4.1.2.3.5 VISUALIZACIÓN Y DESARROLLO DE EJERCICIOS	58
4.1.2.4 INTERFACE	58

4.1.2.4.1 AUTENTICACIÓN	58
4.1.2.4.2 CARGA MASIVA DE USUARIOS	58
4.1.2.4.3 CARGAR BANCO DE PREGUNTAS	59
4.1.2.4.4 REGISTRO	59
4.1.2.4.5 CATÁLOGOS	59
4.1.3 REQUERIMIENTOS DE HARDWARE	59
4.1.3.1 SERVIDOR DE BASE DE DATOS	59
4.1.3.2 SERVIDOR WEB	59
4.1.4 REQUERIMIENTOS DE SOFTWARE	60
4.1.4.1 SERVIDOR DE BASE DE DATOS	60
4.1.4.2 SERVIDOR WEB	60
4.1.4.3 AMBIENTE DE DESARROLLO	60
4.1.5 ARQUITECTURA DE HARDWARE	60
4.1.6 ARQUITECTURA Y FRAMEWORK DE SOFTWARE	61
4.1.7 TECNOLOGIA DE LA BASE DE DATOS	62
4.2 DIAGRAMA DE FLUJO DE DATOS	63
4.2.1 DIAGRAMA DE CONTEXTO	63
4.2.1.1 DIAGRAMA DE FLUJO DE DATOS NIVEL 1	64
4.3 CASOS DE USO	64
4.3.1 CASOS DE USO INTERFASE DEL SISTEMA	64
4.3.2 CASOS DE USO MODULO DOMINIO	68
4.3.3 CASOS DE USO MODULO TUTOR	69
4.3.4 CASOS DE USO MODULO ESTUDIANTE	70
4.3.5 CASOS DE USO DEL SISTEMA	72
4.4 MODELO ENTIDAD RELACION	74
4.5 DICCIONARIO DE DATOS	74
4.6 REFINAMIENTO DEL PROTOTIPO	80
4.9.1 PRUEBAS DE PROTOTIPADO	81
4.9.1.1 PROTOTIPO VERSIÓN 1.0	82
4.9.1.2 PROTOTIPO VERSIÓN 2.0	82
4.9.1.3 PROTOTIPO VERSIÓN 3.0	82
4.10 INTERFAZ GRAFICA	83
4.10.1 LOGIN, CAMBIO DE CONTRASEÑA Y MODELO DE MENU DE OPCIONES	83

4.10.1.1 INICIO DE SESION	83
4.10.1.2 MENU DE OPCIONES	84
4.10.1.3 CAMBIAR DE CLAVE	85
4.10.2 ROL ADMINISTRADOR	86
4.10.2.1 CARGAR BANCO DE PREGUNTAS	86
4.10.2.2 REGISTRO DE USUARIOS	87
4.10.2.3 PARAMETROS DEL SISTEMA	87
4.10.2.4 UNIDAD	88
4.10.2.5 TEMA	89
4.10.2.6 CARGA MASIVA DE USUARIOS	89
4.10.3 ROL ESTUDIANTE	90
4.10.3.1 PANEL DE APRENDIZAJE DEL ESTUDIANTE	90
4.10.3.2 RESUMEN DE RESPUESTAS	92
4.10.3.3 EVALUACION DEL ESTUDIANTE	92
4.10.3.4 REVISION Y DESCARGA DE RECURSOS	93
4.10.4 ROL DEL TUTOR	94
4.10.4.1 BASE DE CONOCIMIENTOS	94
4.10.4.2 AGREGAR RECURSOS	95
4.10.4.3 SEGUIMIENTO	95
4.10.4.4 PARAMETROS DE AVANCE	96
4.11 DESCRIPCION FUNCIONAL DEL PROTOTIPO	96
CONCLUSIONES	100
RECOMENDACIONES	101
BIBLIOGRAFÍAS	102
ANEXOS	107
ANEXO #1	107
ANEXO #2	112

INDICE DE FIGURAS

Figura 1: Interacción de cada uno de los módulos para el funcionamiento de un Sistema Tutor Inteligente	10
Figura 2: Esquema de un Sistema Tutor Inteligente con cada uno de sus principales módulos ...	12
Figura 3: Syllabus de la materia Lenguajes de Programación.....	14
Figura 4: Sub-módulo de Protocolos Pedagógicos	16
Figura 5: Enfoques investigativos	38
Figura 6: Técnicas de entrevistas estructuradas y no estructuradas.....	42
Figura 7: Diagrama del Prototipo de un Sistema de Tutor Inteligente	46
Figura 13: Base de Conocimientos categorizada por niveles de los Lenguajes Imperativos	48
Figura 14: Ultimo Syllabus de la materia Lenguajes de Programación	48
Figura 8: Arquitectura del Hardware.....	61
Figura 9: Arquitectura del Hardware a detalle	62
Figura 10: Logo de MySQL:.....	62
Figura 11: Diagrama de contexto del Sistema de Tutor Inteligente	63
Figura 12: DIAGRAMA DE FLUJO DE DATOS NIVEL 1	64
Figura 15: Modelo Entidad Relación de la Base de Datos, desarrollado por el autor	74
Figura 16: Login del prototipo sistema tutor inteligente.....	84
Figura 17: Menú de las opciones.....	85
Figura 18: Interfaz de cambio de clave.....	86
Figura 19: Carga de banco de preguntas.....	87
Figura 20: Interfaz de Registro de Usuarios	87
Figura 21: Interfaz de Parámetros del Sistema	88
Figura 22: Interfaz de Ingreso de Unidad	88
Figura 23: Interfaz de Ingreso de Tema.....	89
Figura 24: Carga Masiva de usuarios	90
Figura 25: Panel de aprendizaje del estudiante sin recursos	91
Figura 26: Panel de aprendizaje del estudiante con recursos	91
Figura 27: Resumen de las respuestas de la prueba	92
Figura 28: Evaluación del estudiante.....	93
Figura 29: Interfaz de revisión y descarga de los recursos pedagógicos	94
Figura 30: Panel de Ingreso de la Base de Conocimientos	94
Figura 31: Panel de Ingreso de los Recursos	95
Figura 32: Panel de Seguimiento del Estudiante	95
Figura 33: Panel de Parametrización de los avances.....	96

INDICE DE TABLAS

Tabla 1: Entradas y salidas de los módulos del sistema de tutor inteligente	11
Tabla 2: Tipos de Recursos Didácticos	15
Tabla 3: Características de los estilos de aprendizaje del estudiante	22
Tabla 4: Recursos Digitales o Pedagógicos	49
Tabla 5: Servidor de Base de Datos de Hardware	59
Tabla 6: Servidor Web de Hardware	59
Tabla 7: Servidor Base de Datos a nivel de Software	60
Tabla 8: Servidor Web a nivel de Software	60
Tabla 9: Ambiente de Desarrollo de Software	60
Tabla 10: Sistemas y Aplicaciones de la arquitectura y Framework de software	61
Tabla 11: Caso de Uso CU-001	65
Tabla 12: Caso de Uso CU-002	65
Tabla 13: Caso de Uso CU-003	66
Tabla 14: Caso de Uso CU-004	66
Tabla 15: Caso de Uso CU-005	66
Tabla 16: Caso de Uso CU-006	67
Tabla 17: Caso de Uso CU-007	67
Tabla 18: Caso de Uso CU-008	67
Tabla 19: Caso de Uso CU-009	68
Tabla 20: Caso de Uso CU-010	68
Tabla 21: Caso de Uso CU-011	69
Tabla 22: Caso de Uso CU-012	69
Tabla 23: Caso de Uso CU-013	69
Tabla 24: Caso de Uso CU-014	70
Tabla 25: Caso de Uso CU-015	70
Tabla 26: Caso de Uso CU-016	71
Tabla 27: Caso de Uso CU-017	71
Tabla 28: Caso de Uso CU-018	72
Tabla 29: Caso de Uso CU-019	72
Tabla 30: Caso de Uso CU-020	72
Tabla 31: Caso de Uso CU-021	73
Tabla 32: Caso de Uso CU-022	73
Tabla 33: Información de los usuarios del sistema	74
Tabla 34: Información de los usuarios del sistema	75
Tabla 35: Información de los usuarios del sistema	75
Tabla 36: Rol del usuario en el sistema	75
Tabla 37: Menú dinámico de las opciones del sistema	76
Tabla 38: Información sobre los parámetros sobre el avance en los contenidos	76
Tabla 39: Parámetros del sistema	76
Tabla 40: Información de los catálogos del sistema	76
Tabla 41: Información sobre los temas de la materia	77
Tabla 42: Recursos relacionados con el contenido	77

Tabla 43: Información sobre el progreso del aprendizaje del estudiante.....	78
Tabla 44: Información sobre el progreso del aprendizaje del estudiante por contenido	78
Tabla 45: Información sobre los temas de la materia	78
Tabla 46: Información sobre los contenidos aprendidos por los usuarios	79
Tabla 47: Información sobre los recursos revisados por los usuarios.....	79
Tabla 48: Información sobre las preguntas realizadas por el estudiante	80
Tabla 49: Información sobre los intentos en realizar los cuestionarios	80
Tabla 50: Implementaciones del prototipo	81
Tabla 51: Descripción Funcional de software con cada una de sus características	97

INDICE DE ANEXOS

Anexo 1: Syllabus Nuevo Rediseño de Lenguajes de Programación del período B-2021 otorgado por el docente de la materia. Ing. José Miguel Erazo	107
Anexo 2: Entrevista al docente de la materia lenguajes de Programación - Ing. José Erazo Ayón con sus respectivas preguntas y respuestas.....	112

RESUMEN

El presente trabajo tiene como objetivo es diseñar un prototipo de sistema de tutor inteligente basado en WEB para el aprendizaje de la materia Lenguajes de programación. El prototipo ha sido diseñado como una variante de sistema tutor inteligente, dando como resultado un sistema adaptativo de evaluación de conocimiento. Se utilizó la metodología de prototipo rápido considerando cada uno de los módulos respectivos siendo estos el estudiante, dominio, tutor e interfaz.

Se realizó una entrevista al Ing. José Erazo, con el objetivo de abordar temas relacionados con la enseñanza y aprendizaje para la materia Lenguajes de Programación. Las conclusiones de la entrevista indican que la implementación del prototipo, servirá de gran manera para los estudiantes con el fin mejorar el aprendizaje pragmático de los mismos. Cabe recalcar que el estudiante es quien decide si mejorar su aprendizaje o no.

Palabras Claves: Aprendizaje, Tutor inteligente, Métodos de enseñanza Aplicación web, sistema adaptativo de evaluación de conocimiento, aprendizaje pragmático, estudiantes, lenguaje de programación.

ABSTRACT

The objective of this work is to design an intelligent tutor system prototype based on WEB for the learning of Programming Languages. The system has been developed as a variant of the intelligent tutor system, resulting in an adaptive knowledge evaluation system. The rapid prototype methodology was used considering each of the respective modules, these being the student, domain, tutor and interface.

An interview was conducted with Ing. José Erazo, with the aim of addressing issues related to teaching and learning for the Programming Languages subject. The conclusions of the interview indicate that the implementation of the prototype will serve in a great way for students in order to improve their pragmatic learning. It should be emphasized that the student is the one who decides whether to improve their learning or not.

Keywords: Learning, adaptive knowledge evaluation system, Intelligent tutor, Teaching methods, Web application, students, pragmatic learning, and programming language.

INTRODUCCIÓN

Los avances tecnológicos han contribuido sustancialmente en la educación, entre los que tenemos la implementación de plataformas virtuales educativas y además la incorporación de prototipos de sistemas tutores inteligentes que han sido desarrollados para responder lo más aproximado posible a la tutoría humana ayudando a los estudiantes a entender cómo funcionan los lenguajes de programación de manera efectiva.

La actualidad que envuelve al ser humano evoluciona a pasos desmedidos en distintos ámbitos. El crecimiento y evolución de las nuevas tecnologías a nivel tecnológico/ informático, han provocado una transformación total de cada una de las formas de difusión, producción y consumo del conocimiento. La aparición de nuevas tecnologías tales como son: la televisión digital, la introducción de las computadoras en los hogares, el acceso a internet, telefonía móvil ha provocado nuevas necesidades formativas y de conocimientos en las personas.

Por lo que las personas de hoy en día, requieren de nuevas habilidades y conocimientos especialmente los estudiantes universitarios, para poder desarrollarse en el campo laboral.

El aprendizaje asistido por computadora se originó por los años 60 por medio de investigaciones realizadas en universidades y han ido mejorando a través de los tiempos teniendo un mayor alcance con la aparición del internet.

Los sistemas tutores inteligentes comenzaron teniendo un mayor desarrollo alrededor de los años 80 y fueron diseñados con la gran intención de impartir conocimientos fundamentándose en formas de inteligencia artificial para poder así, guiar al estudiante en el proceso de aprendizaje (Urretavizcaya, 2001) en este caso en Lenguajes de programación.

El propósito es mostrar un comportamiento equivalente o parecido al de un profesor o también llamado tutor humano, y este como tal que se pueda adaptar al

comportamiento del estudiante, identificando así la forma en el que este resuelva un problema y brindarle la ayuda cuando cometa errores.

Los sistemas tutores inteligentes son aplicaciones que ayudan a la enseñanza y aprendizaje por medio de la interacción entre el tutor y el estudiante utilizando diferentes dispositivos electrónicos y el intercambio de información.

Los sistemas adaptativos de evaluación de conocimiento es la personalización a nivel educativo de múltiples técnicas de aprendizaje, tras un proceso de diferenciación que identifica las necesidades específicas del estudiante y ofrece diferentes posibilidades. Un elemento muy destacable de un modelo adaptativo de evaluación de conocimiento es la medición del progreso, es decir, la evaluación que por sus características es adaptativa. Siendo este un recurso muy importante para el componente de trabajo autónomo del estudiante.

La implementación del prototipo contribuirá al aprendizaje de los lenguajes de programación de los estudiantes de Ingeniería en Ciencias de la computación de la Universidad Católica Santiago de Guayaquil.

El presente trabajo tiene como propósito brindar un medio por el cual el “aprendizaje” sea propicio y apoye la enseñanza y el aprendizaje del estudiante, que permita la corrección de errores del mismo en el proceso de enseñanza-aprendizaje, todo esto gracias a la construcción de un sistema tutor inteligente.

La mayor y gran importancia del presente trabajo es exponer y dar a conocer que la aplicación de la informática en el área educativa en nuestro medio es muy útil al mismo tiempo es una herramienta valiosa y aplicable tanto para el docente como para el estudiante.

CAPITULO I: EL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA

En el proceso de enseñanza y aprendizaje de la asignatura Lenguajes de Programación de la UCSG, el estudiante contaba con interacciones con el docente a nivel presencial, realizando actividades en conjunto con un seguimiento más cercano debido al acto presencial, adicional a esto, el estudiante podía recurrir a una ayudantía de catedra en el caso de requerirlas, todo esto hasta antes de la pandemia.

El método de aprendizaje estudiante-docente es una técnica si bien la cual favorece y facilita el aprendizaje activo del estudiante, no siempre será el mejor de los métodos a comparación con el grupal (Feng et al., 1991). Ahora con la llegada de la pandemia, el aprendizaje presencial es prácticamente nulo, los estudiantes siguen teniendo un contacto con el docente vía online a través de las revisiones de recursos pedagógicos y tecnológicos que el docente pueda sugerir, para que así el estudiante pueda investigar y revisar, todo esto en el horario de clases.

Adicional a esto, el estudiante no tiene un recurso pedagógico diferente, por lo que se propone crear un prototipo de sistema de tutor inteligente con el propósito es mostrar un comportamiento equivalente o parecido al de un profesor o también llamado tutor humano, y este como tal que se pueda adaptar al comportamiento del estudiante, identificando así la forma en el que este resuelva un problema y brindarle la ayuda cuando cometa errores.

El sistema propuesto permitirá evaluar el aprendizaje de los estudiantes de manera continua y además poder identificar en que tema el estudiante tiene deficiencia, reforzarlo por medio de recursos pedagógicos acordes a su estilo de aprendizaje para así lograr obtener los conocimientos necesarios y dominar el tema y el contenido respectivo.

1.2 JUSTIFICACIÓN

El STI, permitirá al estudiante involucrarse más en el aprendizaje, si este así lo desea, a través de las revisiones de recursos pedagógicos y tecnológicos que el docente pueda sugerir, para que así el estudiante pueda investigar y revisar,

facilitando así su aprendizaje pragmático. El prototipo de sistema de tutor inteligente orientado a una variante del mismo denominado sistema adaptativo de evaluación de conocimiento, a implementarse, será el primero en la UCSG según el repositorio de la misma (UCSG, 2021), y el segundo en la ciudad de Guayaquil.

Esta investigación aportará con conocimientos y material significativo sobre la importancia de la implementación de un sistema tutor inteligente en la carrera de Ingeniería en Computación de la UCSG, los resultados de este prototipo podrán ser utilizados para elaborar futuras propuestas tecnológicas de mejora al sistema educativo, no solo de la UCSG, sino de todo el Ecuador.

El sector a beneficiar serán los estudiantes de la materia Lenguajes de Programación de la carrera Ingeniería en Computación que deseen mejorar sus conocimientos en proceso de enseñanza-aprendizaje de la carrera ya mencionada. Según (Hernández & Rengifo, 2015) a lo largo de la última década, ha surgido la gran necesidad de implementar nuevas estrategias que puedan contribuir al enriquecimiento del proceso enseñanza-aprendizaje, todo esto se debe al desarrollo tecnológico de la información y de las comunicaciones. Sabiendo así que han surgido diversas herramientas que apoyan la educación, siendo una de estos, los sistemas de tutores inteligentes.

Durante el proceso realizado se consiguió que el sistema tenga la capacidad de evaluar y retroalimentar al estudiante, para así ayudarlo a mejorar de gran manera el manejo de conocimientos en las distintas áreas que este tenga debilidades para así fortalecer su rendimiento académico a través del sistema de tutor inteligente de la materia Lenguajes de Programación.

1.3 ALCANCE

En la presente investigación se pretende diseñar un prototipo de sistema de tutor inteligente basado en WEB, siendo capaz de complementar la enseñanza de los distintos contenidos de la materia Lenguajes de Programación de la UCSG. El sistema tutor inteligente servirá de apoyo al estudiante en la asignatura ya mencionada como un tutor extracurricular para así poder afianzar los contenidos explicados.

Se desarrollarán las funcionalidades necesarias de los distintos módulos que integran al tutor inteligente en un entorno web, basados en los estilos de aprendizaje del estudiante, con la inclusión de ya sean documentos, ejercicios, líneas de código, entre otros.

Se implementará una variante denominada sistema adaptativo de evaluación de conocimiento, esto contribuirá al componente de trabajo del estudiante con objetivo general de mejorar la enseñanza y el aprendizaje de los Lenguajes de Programación.

Se abarcará solo un tema, siendo este el estudio de los Lenguajes Imperativos, con sus respectivos contenidos desde la Introducción hasta el manejo de Lenguaje de Programación Multi-paradigma, al final de la revisión de cada contenido se realizará una evaluación para evaluar su aprendizaje.

Para finalizar se realizarán las respectivas pruebas y ajustes del prototipo con el objetivo de tener una implementación efectiva.

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERAL

Diseñar un prototipo de sistema de tutor inteligente basado en WEB para el aprendizaje de la materia Lenguajes de Programación.

1.4.2 OBJETIVOS ESPECÍFICOS

- Diseñar los módulos de Dominio, Tutor y Estudiante que conforman el prototipo de un Sistema Tutor Inteligente.
- Desarrollar un prototipo funcional con sus respectivos módulos basados en la WEB
- Diseñar la funcionalidad de un prototipo de sistema tutor inteligente orientado a los sistemas adaptativos de evaluación del conocimiento.
- Realizar las pruebas del prototipo para validar su funcionalidad.

1.5 PREGUNTA DE INVESTIGACIÓN

El diseño de un prototipo de Sistema Tutor Inteligente, ¿Es un buen recurso tecnológico en el proceso de enseñanza-aprendizaje para los estudiantes de la asignatura Lenguajes de Programación?

CAPITULO II MARCO TEORICO, CONCEPTUAL Y LEGAL

Al llegar a esta sección, encontraremos toda la información recolectada de la investigación teórica en cuanto a procesos y conceptos generales, que se requieren para la puesta en marcha del tutor virtual presentado, así como también conceptos relacionados con la modalidad de trabajo a usar en el aplicativo web.

Más adelante se encuentra lo que es el marco conceptual en donde se redactaran temas técnicos, metodologías usadas y las herramientas necesarias para el tutor virtual a desarrollar.

2.1 MARCO TEÓRICO

2.1.1 INGENIERÍA DE SOFTWARE EDUCATIVO (ISE)

Según Galvis (2000) la ingeniería de software educativo (ISE) es una de las tantas ramas de la ingeniería de software la cual está encargada de mejorar el desarrollo de distintas aplicaciones computacionales que tienen como gran fin en común, en este caso, implementar procesos de aprendizajes desde las instituciones educativas superiores siendo orientado este proyecto a la UCSG.

El término "Ingeniería de Software Educativo", es una referencia total, así del mismo modo es una gran guía para el desarrollo del software en este caso el educativo.

2.1.1.1 METODOLOGIA DE INGENIERÍA DE SOFTWARE EDUCATIVO

La metodología de ingeniería de software, la misma fue publicada en el año 1991, menciona que ofrece varios mecanismos ya sean los de: Análisis, Diseño educativo, Comunicacional, Prueba X piloto y entre otras, para que así, cada vez que se fundamentan en cada uno de los principios ya sean educativos, comunicacionales y de tecnología educativa tengan validez comprobada y sobre todo dolida a la hora de impartir clases.

Sin embargo, desde la otra perspectiva, en este caso de la informática computacional da a entender que no ha evolucionado, por lo que abarca enriquecerla tomando en consideración los multi-avances tecnológicos en el diseño y desarrollo de la computación que se han logrado y que se seguirá incrementando con el correr de los años (Galvis, 1994).

2.1.2 TUTOR

Según (Aretio, 2001), la palabra tutor, hace referencia a la imagen de quien ejerce ayuda, o tutela, de otra persona a falta o necesitada de conocimientos. En el área de educación a distancia, su característica fundamental es la de promover el desarrollo del estudio autónomo, es decir, es un orientador del aprendizaje del estudiante sin la presencia del docente en circunstancias normales.

2.1.2.1 CAPACIDADES DE UN TUTOR

Según (García Aretio, 1994) las principales capacidades que debe tener un tutor son:

- Brindar las posibilidades de las instituciones sociales como objeto de aprendizaje y del medio ambiente social.
- Asesorar al alumno en la organización de su syllabus (objetivos, materias, contenidos, recursos, talleres y actividades).
- Dominar ciertas técnicas y habilidades para así poder tratar de forma específica los contenidos estos pueden ser: escritos, videos, audios, informáticos, que a su vez están integrados dentro de las técnicas del diseño curricular. Utilizar como instrumento las posibilidades del lenguaje total.

2.1.3 LA INTELIGENCIA ARTIFICIAL

Se la conoce con el nombre de inteligencia artificial a las máquinas, sistemas, computadores, ordenadores que imitan la inteligencia humana para poder realizar tareas, actividades y asignaciones que pueden mejorar considerablemente a partir de los datos y la información que se logre recopilar.

En otras palabras esta es conocida por «la habilidad de los computadores para poder hacer actividades que normalmente requieren inteligencia humana»(Rouhiainen, 2018).

La Inteligencia artificial o también llamada por su abreviatura IA se manifiesta de varias formas. Algunos ejemplos conocidos son:

Los chat-bots que son los utilizan IA para comprender con gran rapidez los problemas de los clientes y proporcionar respuestas más eficientes para ellos.

Podemos concluir que, la inteligencia artificial es aquella disciplina encargada de poder construir procesos únicos y distintivos, estos al ser ejecutados sobre una arquitectura física provocan una reacción y un resultado que maximiza una medida de rendimiento fija, basándose así en la cadena de entradas percibidas y en el conocimiento recopilado en la arquitectura deseada por el usuario final.

Por último, los motores de recomendación, estos pueden proporcionar recomendaciones automatizadas para programas de televisión según los hábitos de visualización de los usuarios, más conocidos como los algoritmos de búsqueda (Oracle México, 2020).

2.1.4 SISTEMA DE TUTOR INTELIGENTE (STI)

Es un sistema de enseñanza que es asistido por computadora, este a su vez utiliza distintas técnicas de inteligencia artificial (AI), principalmente para poder dirigir una estrategia de enseñanza y representar el conocimiento necesario; este como tal es capaz de tener comportamientos de nivel experto, ya sea en el dominio del conocimiento, el cual enseña (mostrando al estudiante cómo poder aplicar dicho conocimiento), como en el dominio pedagógico, en donde es capaz de determinar su diagnóstico y a su dar una evaluación respectiva la situación en la que se encuentra el estudiante, como resultado de esto, se ofrecerá una acción o solución que le permita progresar en el aprendizaje respectivo (*Sistema Tutores Inteligentes - EcuRed*, 2018).

Estos sistemas como tal, representan herramientas pedagógicas más evolucionadas y por ende proporcionan más experiencias de aprendizaje individualizado para el estudiante, lo que a su vez permite cada uno de los procesos de enseñanza sean más acoplables a todas las necesidades específicas o al nivel de aprendizaje de cada estudiante, esto se logrará por medio de la construcción y

del análisis que reflejen de manera individual y colectiva, los aspectos cognoscitivos y de comportamiento (Rodríguez Chávez & Rodríguez Chávez, 2021).

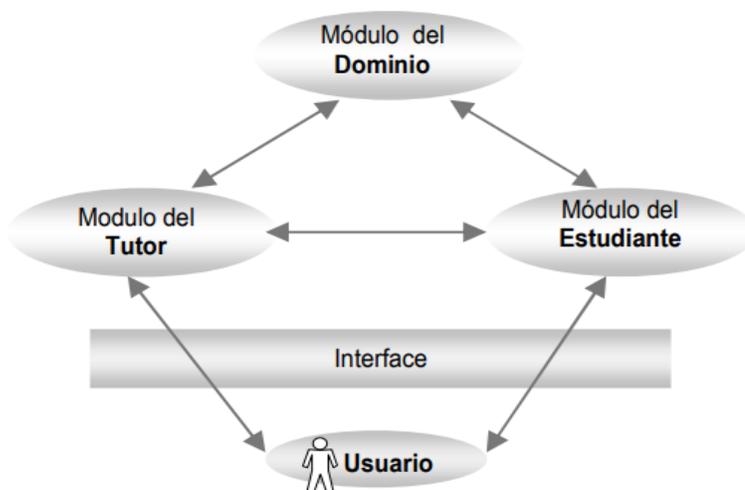


Figura 1: Interacción de cada uno de los módulos para el funcionamiento de un Sistema Tutor Inteligente

Fuente: (Cataldi, 2004)

La figura 1 muestra la interacción entre los módulos básicos, que son modulo del dominio, modulo del Tutor y Modulo del Estudiante. Los sistemas de tutores inteligentes son capaces de determinar lo que el estudiante conoce y a la vez como progresa su aprendizaje, por lo que la enseñanza, se puede acoplar según sean las necesidades del estudiante, sin la presencia de un tutor “humano” en este caso, el docente.

En la tabla 1 se mostrará detalladamente un ejemplo de cada sección del tutor inteligente, en donde se aprecia como el módulo del dominio facilita tanto al estudiante como al tutor, cada una de las actividades y contenidos que son fundamentales y necesarios para poder llevar a cabo los procesos de enseñanza y aprendizaje y cada actualización del módulo del estudiante, y de las actividades (retroalimentación, feedback y preguntas) de los módulos respectivos.

También podemos destacar lo que sucede con el modulo del estudiante, este como tal proporcionará y facilitará al tutor el conocimiento adquirido por el estudiante y a su vez los errores que este haya tenido en el proceso de aprendizaje. Del mismo

modo, el modulo del tutor va a proporcionar al estudiante cada uno de los errores ya mencionados que se hayan cometido por este, para así poder actualizar el modulo respectivo.

Tabla 1: Entradas y salidas de los módulos del sistema de tutor inteligente

Entrada/Salida	Modulo del dominio	Modulo del estudiante	Modulo del Tutor
Modulo del dominio	*****	Se mostrarán contenidos u actividades (ej. objetivos, materias, lecciones, consultas, etc.)	Se mostrarán contenidos u actividades (ej. objetivos, materias, lecciones, consultas, etc.)
Modulo del estudiante	*****	*****	Conocimiento del estudiante y errores que se hayan presentado anteriormente
Modulo del Tutor	*****	Se mostrarán cada uno de los errores cometidos por el estudiante	*****

Fuente: (Ferreira et al., 2012)

2.1.4.1 ARQUITECTURA DE LOS SISTEMAS INTELIGENTES

Todos los sistemas de tutores inteligentes poseen una arquitectura muy distintiva, estas permiten que el proceso de enseñanza-aprendizaje del estudiante sea interactivo/participativo a través de los módulos ya mencionados.

Los STI permiten la emulación de un tutor real/humano para determinar su metodología de enseñanza través de un módulo del dominio, el cual es el que define el

dominio del conocimiento, el módulo del estudiante, el cual es el que se encarga de delimitar el conocimiento del estudiante en cada tópico durante la sesión de estudio, un módulo del tutor el cual es el que genera las interacciones de aprendizaje basadas en cada una de las discrepancias entre el docente y el estudiante, por último, la interfaz con la cual el usuario permite la interacción del estudiante con un sistema de tutor inteligente de una manera más eficiente podemos tomar como ejemplo el conocimiento sobre cómo presentar los contenidos a revisar de una materia en específico.

Para la interfaz gráfica se siguen los principios del diseño, implementación y evaluación de sistemas computacionales, estos como tal interactivos para poder ser utilizados por humanos como se ve en la siguiente figura.

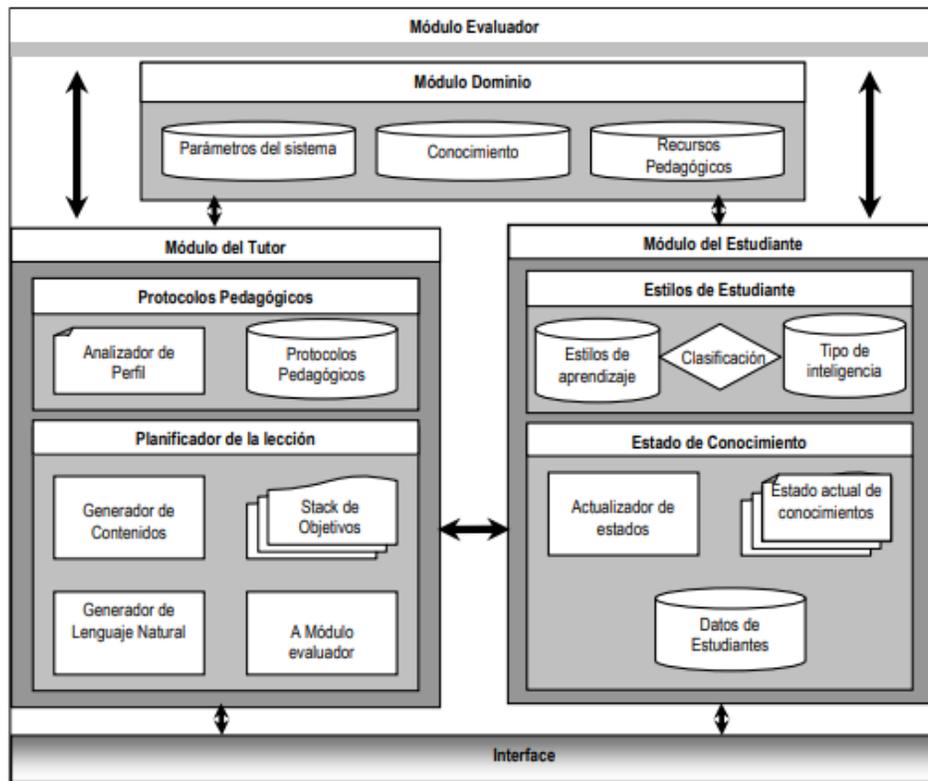


Figura 2: Esquema de un Sistema Tutor Inteligente con cada uno de sus principales módulos

Fuente: (Cataldi, 2004)

2.1.4.1.1 MÓDULO DEL DOMINIO

El Módulo del Dominio consiste en la creación de un sistema experto capaz de inferir conocimiento para verificar evaluaciones y recomendar contenido, según

la planificación de los temas a desarrollar. Básicamente es el representa las explicaciones y el conocimiento que domina el docente en calidad de experto en su área; es por ello necesario desarrollar un Sistema Experto para complementar esta peculiaridad en el STI.

Es el que se define como “El módulo dominio o del experto, el cual da a conocer al usuario sobre la materia o curso que se impartirá. Entre los sub-módulos del sistema de tutor inteligente están los siguientes:

2.1.4.1.1.1 PARÁMETROS BÁSICOS DEL SISTEMA

Los parámetros básicos para que un sistema pueda funcionar se deberán almacenar en una base de datos, así como también las contraseñas para los profesores, para que éstos puedan acceder a distintos menús particulares, donde pueda mostrar el desarrollo de cada una de las sesiones pedagógicas que tenga el sistema.

A su vez, se almacenan los datos para que el sistema funcione de forma remota, esto se debe a no existe la necesidad de que los módulos se puedan encontrar físicamente en un mismo host, sino, que estos como tal puedan estar distribuidos a través de una red. Existen otros parámetros a nivel interno, tales como las contraseñas de los usuarios y otros datos que requieran el sistema poder operar con normalidad, pero que a la vez no se encuentran almacenados y recopilados en ninguno de los otros sub-módulos por lo que entran dentro de esta categoría y por lo tanto estos, se almacenan como un parámetro más para el funcionamiento del sistema.

Dentro de los parámetros del sistema se tiene:

- Ruta de almacenamiento de los recursos pedagógicos
- Tamaño máximo de los archivos a ser cargados al servidor
- Periodo Lectivo
- Secuenciales
- Etc.

2.1.4.1.1.2 BASE DE CONOCIMIENTOS

Son los elementos o contenidos del tema que deben cargarse por un experto en el sistema de tutor inteligente, en este caso el Syllabus. Esto puede darse a través de

conceptos básicos, preguntas, ejercicios, problemas y las distintas relaciones que pueda sugerir el docente. Estos como tal pueden encontrarse traducidos y disponibles en varios idiomas y con diferentes formas de acceso al conocimiento (mapa de conceptos o Syllabus), independientemente del protocolo pedagógico que el planificador de la lección seleccione para desarrollar el concepto seleccionado, a su vez se espera que todos los datos a disposición del generador de contenidos para así poder generar un currículo o Syllabus de la materia en este caso Lenguajes de Programación.

LENGUAJES IMPERATIVOS		
TEMAS	CONTENIDOS	No. DE HORAS PARA CADA TEMA
LENGUAJES IMPERATIVOS	INTRODUCCIÓN	02:00
LENGUAJES IMPERATIVOS	MANEJO DE DATOS Y TIPOS	02:00
LENGUAJES IMPERATIVOS	ASIGNACIONES Y EXPRESIONES	02:00
LENGUAJES IMPERATIVOS	FLUJOS DE CONTROL	02:00
LENGUAJES IMPERATIVOS	COMPONENTES DE UN PROGRAMA IMPERATIVO	02:00
LENGUAJES IMPERATIVOS	EJEMPLOS DE PROGRAMAS IMPERATIVOS	02:00
LENGUAJES IMPERATIVOS	USO DE LENGUAJES DE PROGRAMACIÓN PREPROCESADOR DE HYPER TEXTO	04:00
LENGUAJES IMPERATIVOS	MANEJO DE LENGUAJE DE PROGRAMACIÓN MULTIPARADIGMA	04:00

Figura 3: Syllabus de la materia Lenguajes de Programación

Fuente: Ing. José Erazo, docente de la materia Lenguajes de Programación de la UCSG

En la figura # 3 se observa el syllabus de la materia Lenguajes de Programación (ANEXO#1) con los temas de estudios correspondientes, siendo “Lenguajes Imperativos” el tema a abordar correspondiente a la Unidad #1 de estudio con sus respectivos contenidos a tratar.

Los contenidos mostrados servirán como guía oficial de la materia Lenguajes de Programación para el aprendizaje secuencial del estudiante con las respectivas evaluaciones al final de cada capítulo o contenido finalizado, es decir, el sistema hará un seguimiento al aprendizaje del estudiante.

2.1.4.1.1.3 ELEMENTOS DIDÁCTICOS

En un sentido amplio se entiende por recursos pedagógicos cualquier instrumento u objeto que pueda servir como recurso para que, mediante su manipulación, observación o lectura se ofrezcan oportunidades para aprender algo, o bien con su uso se intervenga en el desarrollo de alguna función de la enseñanza (Sacristan, 1990).

Es decir, los materiales comunican contenidos para su aprendizaje y pueden servir para estimular y dirigir el proceso de enseñanza-aprendizaje, total o parcialmente.

Son las imágenes, videos, sonidos, es decir material multimedia que se requiere para facilitarle al alumno apropiarse de conocimiento en la sesión pedagógica. También se puede incluir una biblioteca con ejercicios y ejemplos, con las respuestas para su autocomprobación, así como cualquier otro recurso didáctico que permita al alumno acceder a ese conocimiento, es decir desde la idea de nuevas puertas de acceso al mismo (Edith, 1993).

Tabla 2: Tipos de Recursos Didácticos

Tipos de Recursos Digitales	Descripción
Presentaciones de Power Point	Son recursos pedagógicos que proporcionan bases de consulta para cada tema del portal web. Cada presentación contiene un conjunto de diapositivas con explicaciones y ejemplos del tema respectivo
Tareas, Talleres	Son actividades integradoras para realizarse por los estudiantes
Soluciones o códigos	Son recursos que muestran la solución a un ejemplo práctico o taller que se realizó en clases. Estos recursos muestran código en archivos de texto.

Fuente: (Carlos R. Jaimez-González, 2016)

Según la tabla #2 podemos apreciar los distintos recursos que puede utilizar un STI para la enseñanza de una materia en concreto en este caso la de Lenguajes de Programación

2.1.4.1.2 MÓDULO DEL TUTOR

Como su nombre lo indica es quien tutora o acompaña permanentemente y quien orienta al estudiante, pero este caso, es aquel que define la estrategia, las reglas y proceso a aplicarse para proveer la información al estudiante basado en la información recibida por los otros módulos. Contiene como dos grandes componentes: los protocolos pedagógicos y planificadores de la lección.

Para la creación y presentación del material de enseñanza se pueden tomar algunos principios prácticos:

- Inducción que motive al estudiante a aprender.
- Comunicación efectiva, donde se incluye la sencillez en los términos, el uso apropiado de tecnicismos, y la capacidad de parafraseo.
- El uso efectivo de apoyos visuales, con un propósito definido, y apareciendo en el momento apropiado.
- La variación del estímulo

2.1.4.1.2.1 SUB-MÓDULO PROTOCOLOS PEDAGÓGICOS

En la Figura #4 se puede observar el esquema principal del sub-módulo de los protocolos pedagógicos, en concreto, con sus dos componentes básicos. Éste interactúa, con el resto de los sub-módulos del módulo del tutor y a su vez es el que realiza cada una de las peticiones de los datos, más específicamente al módulo del estudiantil, para así poder averiguar cuál es el perfil del mismo y saber cuál es su estado de conocimientos.

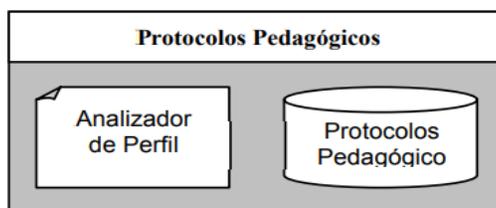


Figura 4: Sub-módulo de Protocolos Pedagógicos

Fuente: (Salgueiro, 2001)

El sub-módulo de protocolos pedagógicos está conformado de un analizador de perfil, el cual determina a partir del perfil del estudiante recopilado y almacenado en el módulo del estudiante, cuál será el mejor protocolo pedagógico a utilizar en cada una de las sesiones pedagógicas. El sistema cuenta con una base de protocolos pedagógicos cuyo uso estará subordinado a la disponibilidad de los contenidos en el módulo de conocimientos, pero la lección siempre se puede generar para alguno de los protocolos.

2.1.4.1.2.1.1 ANALIZADOR DE PERFIL

Este apartado es aquel que debe buscar y determinar, cual es el mejor protocolo pedagógico que se encuentre en el tutor y que esté disponible, para que así el estudiante, pueda conseguir la sesión pedagógica deseada. En este caso la eficacia resultante estará influida en gran medida por la precisión en la selección del protocolo seleccionado.

Se utilizarán las planillas de estilos de aprendizaje recreadas de Felder (1990) que son herramientas para la toma de datos de una validez muy alta y confiabilidad garantizada, que a su vez que han demostrado una gran aptitud a la hora de clasificación de los estilos por su autor y otros investigadores que las utilizaron. La idea de estas planillas es obtener el estilo a través del cual el alumno aprende de un modo más efectivo.

Entre las opciones disponibles, se seleccionó esta planilla de estilos de aprendizaje ya que la misma se había aplicado en investigaciones acerca de los estilos de aprendizaje de los estudiantes de Ingeniería en Ciencias de la Computación respondiendo de un modo óptimo. Según Felder (1990) los estilos se trabajan de a pares y se los puede definir de acuerdo a la:

- **Forma de procesar la información:** Tareas Activas – Tareas de Reflexión.
- **Forma de percibir la información:** Sensorial – Intuitiva (racional).
- **Forma de presentar la información:** Visual – Verbal.
- **Forma del proceso del aprendizaje:** Secuencial – Global.

2.1.4.1.2.1.2 PROTOCOLO PEDAGÓGICO

Los protocolos pedagógicos son instrucciones a seguir, los cuales establecen e indican cuando debe iniciarse un proceso, los pasos que se deben de seguir y las

decisiones respectivas a tomar según determinadas circunstancias a lo largo del proceso de aprendizaje para solucionar y llenar los vacíos académicos.

Como protocolos pedagógicos a nivel presencial podemos destacar los siguientes:

2.1.4.1.2.1.2.1 PROTOCOLO PEDAGÓGICO GENERAL/TRADICIONAL

Es aquel que se realiza de manera tradicional, en donde clases comienzan y el docente imparte sus conocimientos, el estudiante escucha y toma apuntes en sus cuadernos respectivos.

Posterior a eso se realizan talleres o lecciones y se da una retro-alimentación respectiva.

2.1.4.1.2.1.2.2 PROTOCOLO DE APRENDIZAJE BASADO EN PROYECTOS (ABP)

Este protocolo pedagógico permite a los estudiantes adquirir conocimientos respectivos y a su vez generan competitividades a través de la elaboración y desarrollo de proyectos que puedan dar una respuesta a problemáticas de la vida real. Todo esto, partiendo de un problema en concreto y sobre todo real.

Esta metodología asegura procesos de aprendizaje mucho más eficaces, prácticos y didácticos, a la par, permite al estudiante desarrollar competencias complejas como el pensamiento crítico, la comunicación, la colaboración y la resolución de problemas de distinta índole (Elige Educar, 2021).

2.1.4.1.2.1.2.3 FLIPPED CLASSROOM (AULA INVERTIDA)

En este modelo de protocolo pedagógico, cada uno de los elementos tradicionales o regulares que se imparten por el docente en el aula de clases, se invierten. El material educativo es estudiado en casa por los estudiantes, posterior a esto, se trabajan en el aula de clases.

El objetivo principal de este protocolo es controlar y optimizar el tiempo que se use en clases regulares para poder dedicarlo a atender las necesidades especiales de cada estudiante y del mismo modo, al desarrollo de proyectos cooperativos.

En el caso del protocolo del tutor inteligente, varia debido a la virtualidad y la no presencia de un docente como tal, por lo que este debe indicar:

1. Cuando se considera que hay que poner en marcha el protocolo en qué fase del curso, y con qué dificultad
2. Considerar cuales son los pasos a seguir para mejorar acorde al nivel y estilo de aprendizaje
3. Si las medidas anteriores no solucionan los vacíos que tiene el estudiante, se puede realizar una evaluación psicopedagógica para conocer puntos fuertes y débiles.
4. Aplicar medidas extraordinarias
5. Etc.

Para los proyectos de sistemas de tutores inteligentes basados en métodos de aprendizaje-enseñanza, lo primero que se debe realizar es una prueba de diagnóstico al estudiante para evaluar los conocimientos que este o esta posea.

El estudiante para poder pasar o avanzar de contenido en el tema a tratar, debe de realizar una evaluación respecto al contenido que haya revisado.

De sacar una calificación menor a la nota mínima por ej. 7/10, siendo esta la calificación mínima para aprobar, el sistema enviará recursos pedagógicos relacionados al estilo de aprendizaje que tenga asignado el estudiante.

Si el estudiante persiste reprobando, se realizará una evaluación psicopedagógica para conocer puntos fuertes y débiles.

2.1.4.1.3 MÓDULO DEL ESTUDIANTE

Este módulo como tal, tiene como objetivo ser diseñado para diagnósticos de cada uno de los conocimientos previos, capacidades y necesidades del estudiante. Se establece un perfil de acuerdo a sus fortalezas, debilidades y comprensión de contenidos que se adquirieron en horas de clases regulares.

Una vez establecidos, se puede seleccionar y generar a través de reglas y código programado la mejor información que el estudiante necesita recibir para

progresar en la adquisición de más conocimientos. Esto es continuo y se actualiza en cada interacción con el tutor respectivamente.

2.1.4.1.4 MÓDULO DE INTERFAZ

Es la interfaz de interacción entre el sistema tutor inteligente y el estudiante real, que se encarga de presentar el material de dominio y cualquier otro elemento didáctico de la manera correcta. La interface, que permite a los usuarios: el estudiante, el instructor y el desarrollador del sistema, para su desarrollo se pueden seguir los criterios ergonómicos basados en el estándar ISO 309241 para Human Computer Interface.

Es la interface de interacción entre el STI y el estudiante real, que se encarga de presentar el material del dominio y cualquier otro elemento didáctico de la manera correcta La interfaz, que permite a los usuarios interactuar con el sistema (Cataldi, 2004).

2.1.5 APRENDIZAJE

Se entiende por aprendizaje al proceso a través del cual el ser humano adquiere o modifica sus habilidades, destrezas, conocimientos o conductas, como fruto de la experiencia directa, el estudio, la observación, el razonamiento o la instrucción.

2.1.5.1 PROCESO ENSEÑANZA – APRENDIZAJE

Se entiende el proceso de enseñanza-aprendizaje como un fenómeno universal requerido para la continuidad cultural, a través del cual una generación prepara a otra que le sucede.

Fundamentalmente, el objetivo principal es producir un cambio que puede ser de la ignorancia al saber, de reconstruir nuevos conocimientos a partir de lo previamente conocido. En otras palabras, podemos decir que es un proceso socio-cultural, a través del cual se comparten significados entre individuos (Perales, 2000).

2.1.5.2 ESTILOS DE APRENDIZAJE

El concepto estilos de aprendizaje se refiere a aquellas estrategias preferidas por los estudiantes y que se relacionan con distintas formas de recopilar, interpretar, organizar y pensar sobre la nueva información a la cual se va a tratar.

El estilo de aprendizaje consiste en definitiva en cómo nuestra mente procesa la información, cómo es influida por las percepciones de cada individuo, con el fin de alcanzar aprendizajes eficaces y significativos. Por ejemplo, cuando se aprende un nuevo concepto, algunos estudiantes se centran en los detalles, otros en los aspectos lógicos, otros prefieren hacerlo leyendo o llevándolos a la práctica a través de actividades

Una de las mejores capacidades que tiene el ser humano es la capacidad para aprender. Aprende espontáneamente, aprende de los errores, como gran consecuencia de esto, solucionar los problemas respectivos, para poder tener una vida mejor.

A medida que se han desarrollado las ciencias y las tecnologías, se ha tratado de comprender cómo aprendemos y cómo podemos intervenir para mejorar los procesos que conllevan a la adquisición de nuevas conductas, nuevos comportamientos, nuevas habilidades, o en términos más actuales, nuevas competencias.

Para Honey y Mumford (2006) los estilos de aprendizaje son cuatro:

2.1.5.2.1 ACTIVOS

Son aquellas personas que tienen predominancia en buscar en nuevas experiencias. Mentalmente son muy abiertos, no son para nada escépticos, se crecen con los desafíos que suponen nuevos aprendizajes y sobre todo experiencias, y se aburren con los largos plazos, en otras palabras, clases largas.

A la vez son personas que les gusta trabajar en grupo, se sienten muy confortables al hacerlo por lo que se involucran activamente en cada actividad que les sea asignada. Ej.: estudiantes que les guste hacer/resolver ejercicios de programación, hacer un puzle.

2.1.5.2.2 REFLEXIVOS

Los reflexivos, así como los activos aprenden también con las nuevas experiencias, pero en cambio a estos, no les gusta estar directamente implicados en ellas. Reúnen datos, que los analizan con detenimiento antes de llegar a alguna conclusión determinada.

Así mismo, Disfrutan observando la actuación de los demás, escuchándolos, pero sin intervenir hasta que se han adueñado de la situación. Ej. Estudiantes que revisen documentación mediante enlaces, comentar una noticia del periódico.

2.1.5.2.3 TEÓRICO

Los teóricos son los que aprenden mucho mejor cuando las cosas que se les enseñan forman parte de un sistema, modelo, teoría o concepto. Les gusta analizar y sintetizar. Para ellos si algo es lógico, es bueno. Ej. Estudiantes que sean más afines a la revisión de material documentado, ya sean PDF, presentaciones de diapositivas, imágenes, entre otros, como otro ejemplo podemos poner un juego de razonamiento lógico, con definiciones claras.

2.1.5.2.4 PRAGMÁTICO

El punto fuerte de los pragmáticos es la aplicación práctica de las ideas. Descubren el aspecto positivo de las nuevas ideas y aprovechan la primera oportunidad que se les presenta para así poder experimentarlas.

Tienden a ser muy impacientes cuando hay personas que teorizan. Ej. Estudiantes son más afines a realizar evaluaciones, imparte una clase.

En la tabla #3 veremos algunas características de los estilos de aprendizaje que tendrían los estudiantes.

Tabla 3: Características de los estilos de aprendizaje del estudiante

CARACTERÍSTICAS DE LOS ESTILOS DE APRENDIZAJE			
Estudiante Activo	Estudiante Reflexivo	Estudiante Pragmático	Estudiante Teórico
Mente abierta	Prudente	Actúa por cuenta propia	Lógicos
No es prejuicioso/a	Receptivo	Impaciente	Perfeccionistas
Entusiasta	Analítico	Realista	Objetivos
Creativo	Observador	Directo	Racionales
Arriesgado	Argumenta	Organizador	Analíticos
Inventor/ Innovador	Detallista	Eficaz	Críticos
Líder	Investigador	Practico	Metódicos

Fuente: (Carrillo, 2014)

2.1.5.3 TIPOS DE APRENDIZAJE

Los estilos de aprendizaje son todos aquellos rasgos cognitivos y fisiológicos por los que los estudiantes perciben e interactúan dentro cada uno de los procesos de aprendizaje. Entre estos según (Alejandro Rodríguez Puerta, 2019) estos son:

2.1.5.3.1 APRENDIZAJE IMPLÍCITO

Es aquel que se compone en un aprendizaje habitualmente no intencional/voluntario y donde el estudiante (este caso es la persona en proceso de aprendizaje, no es consciente de lo que se aprende en el proceso) (Alejandro Rodríguez Puerta, 2019) .

2.1.5.3.2 APRENDIZAJE EXPLÍCITO

Este se caracteriza porque el estudiante tiene toda la intención de aprender y es muy consciente de lo que se aprende.

Como ejemplo decimos que podemos adquirir información sobre personas, objetos y lugares, a cualquier hora y en cualquier momento (Alejandro Rodríguez Puerta, 2019).

2.1.5.3.3 APRENDIZAJE ASOCIATIVO

Es un proceso por el cual, el estudiante aprende a asociar entre uno o dos estímulos y un comportamiento determinado (Alejandro Rodríguez Puerta, 2019).

2.1.5.3.4 APRENDIZAJE NO ASOCIATIVO

Se basa en un cambio en nuestra respuesta ante un estímulo en concreto, el cual se presenta de forma prologada y repetitiva. Como ejemplo podemos tener el hecho de que, cuando alguien vive cerca de una discoteca o un bar, al principio será incómodo y molesto por el ruido que puede ocasionar.

Con el pasar del tiempo, tras haberse adaptado a este hecho, no se notará el ruido, teniendo como conclusión que la persona se adaptó a este cambio (Alejandro Rodríguez Puerta, 2019).

2.1.5.3.5 APRENDIZAJE SIGNIFICATIVO

Este tipo de aprendizaje se caracteriza porque el estudiante recoge la información, la selecciona, organiza y establece relaciones con el conocimiento que ya tenía

previamente. En otras palabras, es cuando una persona relaciona la información nueva con la que ya posee (Alejandro Rodríguez Puerta, 2019).

2.1.5.3.6 APRENDIZAJE COOPERATIVO

Este tipo permite que cada alumno aprenda, no de manera individual, sino junto a sus compañeros de clase. Por lo general se realizan en las aulas de clases a nivel presencial y los grupos de estudiantes no suelen superar los cinco miembros. El docente es quien forma los grupos y por ende quien los guía, distribuyendo roles y las funciones que cada estudiante debe ejercer.

2.1.5.3.7 APRENDIZAJE COLABORATIVO

El aprendizaje colaborativo es similar al aprendizaje cooperativo. Ahora bien, el primero se diferencia del segundo en el grado de libertad con la que se constituyen y funcionan los grupos.

En este tipo de aprendizaje, son los profesores o educadores quienes proponen un tema o problema y los estudiantes toman la decisión respectiva del cómo abordarlo (Alejandro Rodríguez Puerta, 2019).

2.1.5.3.8 APRENDIZAJE EMOCIONAL

Significa poder conocer y gestionar cada una de las emociones de una forma más eficiente. Este aprendizaje puede aportar muchos beneficios a nivel psicológico y sobre todo, este influye positivamente en nuestro bienestar personal, mejora las relaciones interpersonales, y a su vez favorece el desarrollo personal (Alejandro Rodríguez Puerta, 2019).

2.1.5.3.9 APRENDIZAJE OBSERVACIONAL

A este aprendizaje se lo conoce como aprendizaje del modelado, este como tal se puede basar en una situación social en donde al menos participan dos personas: el modelo (siendo este, la persona de la que se aprende) y el sujeto o persona, es aquel que realiza las respectivas observaciones de dicha conducta, y las aprende (Alejandro Rodríguez Puerta, 2019).

2.1.5.3.10 APRENDIZAJE EXPERIENCIAL

Es el aprendizaje que se produce, como su propio nombre indica, a través de la experiencia. Siendo esta, una manera muy fuerte de aprender. En concreto podemos decir que cuando hablamos de aprender y mejorar los errores, nos podemos referir al aprendizaje producido por experiencia propia (Alejandro Rodriguez Puerta, 2019).

2.1.5.3.11 APRENDIZAJE POR DESCUBRIMIENTO

Hace referencia al aprendizaje activo, este se basa en que la persona al contrario de aprender los contenidos de forma pasiva. Aprende descubriendo, relacionando y reordenando cada concepto para así, poder adaptarlos a su esquema cognitivo (Alejandro Rodriguez Puerta, 2019).

2.1.5.3.12 APRENDIZAJE MEMORÍSTICO

Significa aprender y fijar en la memoria distintos conceptos sin entender lo que significan, por lo que no realiza un proceso de significación. Es un tipo de aprendizaje que se lleva a cabo como una acción mecánica y repetitiva (Alejandro Rodriguez Puerta, 2019).

2.1.5.3.13 APRENDIZAJE RECEPTIVO

Con este tipo de aprendizaje denominado aprendizaje receptivo la persona recibe el contenido que ha de internalizar. Es un tipo de aprendizaje impuesto, pasivo.

En el entorno de clases ocurre este hecho cuando el estudiante, sobre todo por la explicación del docente, ya sea material impreso o la información audiovisual, solo necesita comprender el contenido para poder reproducirlo (Alejandro Rodriguez Puerta, 2019).

2.1.6 LENGUAJES DE PROGRAMACIÓN

Sabemos que un lenguaje de programación es una herramienta que permite desarrollar software o programas para computadora.

Los lenguajes de programación son empleados para diseñar e implementar programas encargados de definir y administrar el comportamiento de los dispositivos físicos y lógicos de una computadora (CUAED, 2016). Lo anterior mencionado se logra mediante la creación e implementación de distintos algoritmos de precisión que se

utilizan como una forma de comunicación humana/real con la computadora.

A grandes rasgos, un lenguaje de programación se conforma de una serie de símbolos y reglas de sintaxis y semántica que definen la estructura principal del lenguaje y le dan un significado a sus elementos y expresiones.

Mediante el lenguaje respectivo, se comunican el programador y la máquina, permitiendo así especificar, de forma concreta y precisa, teniendo aspectos tales como:

- ✓ Cuáles son los datos que debe operar un software en específico
- ✓ Cómo deben ser transmitidos o almacenados los datos
- ✓ Las acciones respectivas que debe tomar el software dependiendo de las circunstancias variables que se presenten.

En otras palabras, podemos decir el lenguaje de programación es un sistema estructurado de comunicación, el cual está conformado por conjuntos de símbolos, palabras claves, reglas semánticas y sintácticas que permiten el entendimiento entre un programador y una máquina (Rock Content, 2019).

Los lenguajes de programación son la base fundamental para poder construir todas las aplicaciones digitales utilizadas hoy en día, a su vez estos se clasifican en dos tipos, siendo estos: lenguaje de bajo nivel y de alto nivel.

2.1.6.1 LENGUAJES DE PROGRAMACIÓN DE BAJO NIVEL

Este lenguaje base sirve de interfaz y crea un vínculo único e inseparable entre el hardware y el software, siendo así totalmente orientado a máquina. Además, este ejerce un control directo sobre el equipo y la estructura física. Para poder aplicarlo adecuadamente es necesario que el programador tenga altos conocimientos del hardware (Mathieu Mihaela Juganaru, 2016).

2.1.6.2 LENGUAJES DE PROGRAMACIÓN DE ALTO NIVEL

Tienen como objetivo facilitar el trabajo del programador, ya que utilizan unas instrucciones más fáciles de entender.

Además, el lenguaje de alto nivel permite escribir códigos mediante idiomas que conocemos (español, inglés, etc.) y luego, para ser ejecutados, se traduce al lenguaje de máquina mediante traductores o compiladores (Mathieu Mihaela Juganaru, 2016).

2.1.7 APRENDIZAJE DE LOS LENGUAJES DE PROGRAMACIÓN

En el campo de la programación, todo principiante se debe enfrentar a distintas interrogantes, independientemente de los métodos o herramientas a utilizar si se trata de codificar o programar. El no tener experiencia o conocimiento alguno puede generar grandes dificultades a la hora de tener una visión general de los lenguajes de Programación.

Para aprender programación desde cero, se recomienda elegir el lenguaje de programación que el estudiante crea adecuado para estudiar. En concreto, recomienda elegir un lenguaje que sea agradable y fácil para comenzar. Un ejemplo podría ser, HTML o CSS, estos se usan principalmente para aplicaciones web básicas. Si el estudiante busca por lenguajes de programación más serios para aprender cómo programar, Python podría ser perfecto.

2.1.7.1 ¿COMO SE PUEDEN APRENDER LOS LENGUAJES DE PROGRAMACION?

Las habilidades de programación son de naturaleza amplia; la programación como tal abarca nociones muy variadas tales como la estrategia de solución de problemas y la inteligencia emocional.

Cuando se aprenden los lenguajes de programación hay varias cosas que se deben comprender a la perfección:

- Aprender los fundamentos de ciencias computacionales, para ello existen muchos recursos. Para poder aprender programación desde cero, se necesita tener una clara idea de cómo funcionan los computadores.
- Entender cómo funcionan las redes. Es importante saber cómo diferentes computadoras se pueden comunicar entre ellas en varios tipos de redes. Esto

podrá facilitar de gran manera al estudiante, el entendimiento de conceptos más complejos en futuros cursos de programación como tal.

- Familiarizarse con la línea de comando.
- Después de todo lo mencionado, es importante saber cómo crear una página web y conseguir un dominio. Esto no te será un gran desafío, pero es esencial para el aprendizaje de todo estudiante para poder convertirse en un desarrollador web (Laura M., 2017).

En cuanto a las habilidades debemos destacar las siguientes:

- Establecer planes detallados/plan detallado: El programador al enfrentarse a la solución del problema, luego de haberlo comprendido en su totalidad debe proponer los pasos a nivel lógico que se deben seguir para así poder obtener una solución en concreto.
- Conocer de gran manera el lenguaje de programación en el que se desea desarrollar: Para poder solucionar el problema el programador debe conocer que características puede ofrecer el lenguaje en el que se piensa desarrollar.
- “Debugging la app”. Esto se hace para confirmar lo que hace el programa o en su defecto lo que no hace, de este modo se puede interpretar los mensajes de salidas de erros y logs para así poder corregirlos y a nivel general realizar rastreos del flujo de la app (ACTIVE, 2015).

2.1.7.2 ¿QUÉ ES LO QUE SE APRENDE DE LOS LENGUAJES DE PROGRAMACION?

Por lo general los cursos de lenguajes de programación buscan el desarrollo de competencias generales y concretas en el estudiante, siendo estas:

- Capacidad de analizar y sintetizar.
- Capacidades para aprender, modernizarse y actualizarse.
- Capacidad de solucionar distintos problemas.
- Capacidad de proponer pasos o secuencias lógicas para resolver un problema.
- Capacidad de diseñar e implementar algoritmos eficientes en un lenguaje de programación.

Todo esto con el objetivo de que el estudiante pueda desarrollar habilidades y destrezas que sirvan para su carrera estudiantil y que mejor manera que mirarlo a largo plazo, siendo así en el tan conocido campo laboral, que en este siglo 21 abre muchas puertas para los programadores y desarrolladores, debido a su gran demanda en la sociedad de hoy en día.

No cualquiera puede ser programador o adquirir habilidades para poder serlo, para ello cada una de las habilidades mencionada anteriormente deben desarrollarse a lo largo de su etapa de estudio en la carrera escogida. Todas las habilidades tienen puntos a mejorar y puntos a corregir, todo esto es parte del aprendizaje, el estudiante debe poner de sí mismo gran esfuerzo y sobre todo dedicarse a aprender, desarrollar y mejorar cada uno de estos aspectos.

2.1.8 ENSEÑANZA DE LOS LENGUAJES DE PROGRAMACIÓN

El objetivo principal de la enseñanza de la programación en la formación inicial del docente de informática en general es resolver problemas, con el empleo de un lenguaje de programación (Díaz Tejera et al., 2018). Para que esto se cumpla, se necesita que el estudiante desarrolle el sistema de habilidades específicas de la programación y se apropie del sistema de conocimientos y procedimientos en correspondencia con las técnicas de programación y cada una de las herramientas de desarrollo en estudio.

Los estudiantes de ciencias computacionales o de sistemas computacionales, sea cual sea su mayor fortaleza en la carrera, ya sean estos los programadores y desarrolladores, administradores de base de datos, ciencias de datos, entre otros. Todos deben adquirir conocimientos generales de matemáticas, cálculo, estadística, diseño, química, física, construcción de software, entre varios otros servirán para orientar al estudiante en su carrera profesional.

Estas asignaturas fuertes hacían que muchos estudiantes nuevos, atraídos las nuevas tecnologías y la computación, se sintieran incapaces de lograr avanzar y escogieran otra carrera. Gracias a esto la enseñanza de los lenguajes de programación ha cambiado en su totalidad, transformándose en una carrera mucho más accesible y sobre todo necesaria (Pinion, 2020).

Los expertos recomiendan iniciar la enseñanza de programación desde edades tempranas siendo estas de nivel infantil-juvenil dado los beneficios que les aporta, aunque para ello es necesario crear un contexto propicio, seleccionar un lenguaje y entorno de programación adecuados y disponer de unos contenidos en forma de tareas a resolver o desafíos a los que enfrentarse escalados en dificultad.

Según (Edacom, 2015) proveedor de soluciones tecnológicas que trabaja con instituciones comprometidas con la innovación que buscan integrar programas de educación tecnológica en México, recalcó que el aprendizaje de la programación a temprana edad es hoy más importante, debido a que esta disciplina desarrolla grandes habilidades que representan una mayor ventaja en muchos otros aspectos de la vida, debido a que en el futuro, la mayoría de los trabajos se realizarán de manera virtual o digitalmente.

Además de otros beneficios en la enseñanza de la programación para el desarrollo y enseñanza para los jóvenes de hoy en día, según (Observatorio Digital, 2020) recalca:

- ✓ Aprender a programar mejora su razonamiento, les facilita el aprendizaje de las matemáticas y la capacidad de resolver problemas.
- ✓ Favorece su creatividad.
- ✓ Aumenta su confianza y autoestima.
- ✓ Por su naturaleza, obliga a mantener la concentración.
- ✓ Desarrollar y estructurar competencias como la lógica.
- ✓ Fomenta el trabajo en equipo.

En la enseñanza de los lenguajes de programación, ha sido muy recurrente el estudio del proceso de formación y desarrollo de las habilidades. Pudiendo así destacar, las investigaciones realizadas por (Alea, Miqueo, & Aguilar, 2005) y (Díaz & Crespo, 2010) quienes demuestran la existencia de una tendencia para ajustar cada una de las habilidades de la computación a la manipulación de software, ya sean estas de carácter general o específicos.

Sin embargo, los lenguajes de programación cuentan con disciplinas en las que no es suficiente el desarrollar habilidades en la manipulación de software, como el campo de la de desarrollo de software o más conocido como el campo de programadores. El

objetivo esencial de la enseñanza y aprendizaje de los lenguajes de programación es el uso de los computadores con el fin de resolver problemas, a través de los recursos que brindan los diferentes lenguajes de programación.

Durante el proceso de formación y desarrollo de habilidades de computación de los lenguajes de la programación, resulta muy destacable el uso de medios de enseñanza. Siendo el caso de los métodos tradicionales (pizarrón, material impreso, láminas, folletos) permitiendo así al estudiante desarrollar habilidades lógicas y concretas de los lenguajes de programación, vinculadas así con la habilidad informática general para resolver poder problemas con computadoras (Díaz Tejera et al., 2018).

Entre ellos, podemos destacar el análisis de los problemas, la identificación de conceptos y procedimientos a emplear y la descripción de los pasos principales para elaborar su solución. Mientras que la utilización de la computadora y, en especial del software seleccionado como objeto de estudio y aprendizaje, además de contribuir a desarrollar las habilidades antes mencionadas.

Facilitando así el desarrollo de otras habilidades específicas de la programación, que solo pueden lograrse con la interacción entre el estudiante y un entorno de desarrollo integrado. Pudiendo tomar así como criterio la existencia de una gran variedad de tipos de estrategias didácticas, (Ortiz de Zevallos, 2014) propone cuatro tipos para el desarrollo de la identidad cultural para la educación superior:

- Estrategias para generar o activar conocimientos previos.
- Estrategias para facilitar la adquisición del conocimiento.
- Estrategias para el desarrollo de contenidos procedimentales, habilidades cognitivas y habilidades psicomotrices.
- Estrategias para la adquisición de actitudes, valores y normas

El análisis de la caracterización de estos tipos de estrategias didácticas lleva a las autoras a plantear que, para la adquisición de habilidades de programación en la formación de profesores de Lenguajes de Programación, también pueden diseñarse estrategias que se correspondan con estos mismos tipos. Para esto solo sería necesario, emplear técnicas y procedimientos en correspondencia con los objetivos que se quieran alcanzar.

2.1.8.1 ¿COMO SE ENSEÑAN LOS LENGUAJES DE PROGRAMACIÓN?

La enseñanza de los lenguajes de programación en los sistemas educativos actuales toma al grupo de los estudiantes como una entidad con una dinámica incomparable o en este caso única, pasando tema tras tema, sin garantizar en su totalidad que cada estudiante haya alcanzado un estado de manejo del tópico o tema que le de las herramientas suficientes y necesarias para poder avanzar y comenzar con el siguiente tema

Esto como tal causa que paulatinamente el estudiante se vea superado por el contenido y las actividades que se realicen en la materia y a partir de esta premisa, termine de darse por vencido y no volver a ver un curso de programación como tal.

2.1.9 SISTEMAS ADAPTATIVOS DE EVALUACIÓN DE CONOCIMIENTO

2.1.9.1 APRENDIZAJE ADAPTATIVO

El aprendizaje adaptativo es la personalización a nivel educativo de múltiples técnicas a nivel de aprendizaje, tras un proceso de diferenciación que identifica las necesidades específicas del estudiante y ofrece diferentes posibilidades (LAEDU.DIGITAL, 2020).

Se dice que el “aprendizaje adaptativo” obtuvo una gran aceptabilidad de la población en la década de 1970, con el surgimiento del área de la Inteligencia Artificial. La premisa básica fue adaptar el proceso educativo a las fortalezas y debilidades de cada estudiante, sin embargo, en esa época su aplicación no se pudo desarrollar ni expandir a nivel mundial debido a los altos costos y al tamaño de los equipos de cómputo que eran necesarios para la adecuada implementación en aquella época.

2.1.9.2 EVALUACIÓN ADAPTATIVA

En un modelo adaptativo es muy importante es la medición del progreso como elemento, en otras palabras, la evaluación a tomar por sus características es adaptativa. Como antecedente, tenemos que la evaluación adaptativa computarizada (CAT) fue un descubrimiento sucesor de una serie de aplicaciones exitosas que surgieron a partir del año 1905 con el desarrollo del primer examen adaptativo de Alfred Binet: Binet IQ Test (Verywell Mind, 2018).

Las pruebas adaptativas están compuestas por ítems o elementos seleccionados de una colección (banco de ítems). Los elementos se seleccionan para que coincidan con el nivel de habilidad (o aptitud) estimado de la persona: si tiene éxito en un elemento, el

siguiente será un poco más difícil; si falla, el próximo será un poco más fácil. La prueba finaliza cuando la habilidad o aptitud del sujeto alcanza el objetivo planteado (Subitus, 2019).

2.1.9.3 APRENDIZAJE Y EVALUACIÓN ADAPTATIVA

Con la evolución de las TIC, el análisis de datos moderno y la investigación emergente sobre el aprendizaje, hoy en día, nos enfrentamos a la oportunidad tan esperada de poder construir sistemas de aprendizaje y evaluación adaptativos.

Estos sistemas ahora se utilizan en una variedad de entornos para enseñar y capacitar de manera más efectiva.

2.1.9.3.1 MODELOS DE APRENDIZAJE ADAPTATIVO

Los proveedores que ofrecen soluciones de aprendizaje adaptativo a menudo trabajan en diferentes áreas de la investigación académica, incluidos los sistemas de tutoría inteligente, el aprendizaje automático, la memoria y las teorías de la carga cognitiva, entre otros. Además, los sistemas de aprendizaje adaptativo a menudo se dividen en categorías o modelos con varios nombres. Sin embargo, podemos establecer que, independientemente de estas variaciones, existen dos modelos generales, aunque no excluyentes entre sí.

- **Movido por el contenido:** Este modelo se basa en monitorear el desempeño, las interacciones y los metadatos que se generan a partir de la interacción de los estudiantes con el contenido. Esta información y su relación con los objetivos de aprendizaje se concentran en un tablero o panel que el docente consulta para identificar lo que necesita ser ajustado, cambiado o modificado en la instrucción; contenidos o rutas de aprendizaje (Acuña, 2019).
- **Impulsado por la revisión/evaluación:** Suele ser el modelo con el que comúnmente se asocia el aprendizaje adaptativo. En este enfoque, el sistema hace ajustes de forma dinámica y casi en tiempo real a la instrucción, los recursos de aprendizaje y las trayectorias de los cursos en función de la evaluación continua del desempeño y el dominio de los estudiantes. Aquí no es necesaria la intervención del docente (Acuña, 2019).

2.1.9.4 SISTEMAS ADAPTATIVOS DE APRENDIZAJE Y EVALUACIÓN

Existen varias formas de aplicar el aprendizaje adaptativo, que van desde un nivel básico, como los sistemas de respuesta de la audiencia o también conocidos como los “clickers”, estos son que le permiten a la persona que expone su trabajo ajustar su presentación en tiempo real en función de los comentarios que recibe de su audiencia, hasta sistemas más sofisticados que ajustan al tipo de preguntas que se le hacen al alumno en función de sus respuestas previas, como es el caso del TOEFL por ordenador.

En la actualidad, los sistemas más importantes y significativos en educación adaptativa son los siguientes:

2.1.9.4.1 EXÁMENES ADAPTATIVOS (INFORMÁTICA)

Las pruebas adaptativas basadas en computadora (CAT) son un tipo de evaluación que se construye utilizando algoritmos complejos que le permiten adaptar y producir pruebas óptimas para cada estudiante. La prueba se centra en proporcionar los ítems más adecuados al nivel de la persona (FastTest, 2013). FastTest afirma que estos exámenes ofrecen los siguientes beneficios:

- Pruebas mucho más cortas (reducción entre 50 y 90 por ciento del tiempo)
- Calificaciones mucho más precisas
- Mayor motivación en el estudiante.
- Mayor confiabilidad de la prueba

2.1.9.4.2 TUTORIALES ADAPTABLES O ADAPTATIVOS

Son Sistemas Tutoriales Inteligentes (ITS) con los que los estudiantes suelen interactuar a través de una simulación enfocada en un objetivo-tarea mientras son monitorizados y controlados (Alberto, 2015)..

2.2 MARCO CONCEPTUAL

2.2.1 HERRAMIENTAS DE PROGRAMACIÓN

2.2.1.1 VISUAL STUDIO 2019

Microsoft Visual Studio es un IDE creado por Microsoft y utilizado para diferentes tipos de desarrollo de software, como programas informáticos, sitios web, aplicaciones web, servicios web y aplicaciones móviles (Incredibuild, 2021). Contiene herramientas

de finalización, compiladores y otras características para facilitar el proceso de desarrollo de software en los distintos frameworks de desarrollo, en este caso tomaremos el ejemplo de .NET.

2.2.1.2 .NET

.NET es una plataforma de código abierto para desarrolladores, multiplataforma y sobre todo gratuita para así poder crear muchos tipos diferentes de aplicaciones. En este caso .NET será el IDE en el cual se va a desarrollar el código del tutor web

Con .NET, puede utilizar varios idiomas, editores y bibliotecas para crear aplicaciones web, móviles, de escritorio, juegos e IoT (Microsoft, 2017).

2.2.1.2.1 LENGUAJES DE PROGRAMACION EN .NET

Puede escribir aplicaciones .NET en C #, F # o Visual Basic.

2.2.1.2.1.1 C#

C# es un lenguaje de programación simple, moderno, orientado a objetos y con seguridad de tipos. Este lenguaje será la base para el programa a realizar.

2.2.1.2.1.2 F#

F # es un lenguaje de programación que facilita la escritura de código conciso, robusto y eficaz.

2.2.1.2.1.3 VISUAL BASIC

Visual Basic es un lenguaje accesible con una sintaxis simple para crear aplicaciones orientadas a objetos y seguras de tipos. Visual servirá para la parte de desarrollo relacionado con ASP (Microsoft, 2017).

2.2.1.2.1.2 BOOTSTRAP

Bootstrap, es un framework 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 pueda visualizar(Mark Otto, Jacob Thornton, and Bootstrap, 2020). 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.

El beneficio de poder usar responsive design en un sitio web, es principalmente que el sitio web se adapta automáticamente al dispositivo desde donde se acceda. Lo que se usa con más frecuencia, también podemos usar el módulo CSS3 que es el que permite la representación de contenido para adaptarse a condiciones como la resolución de la pantalla y si trabajas las dimensiones de tu contenido en porcentajes, en ese caso puedes tener una web muy fluida capaz de adaptarse a casi cualquier tamaño de forma automática (ARWEB Agencia Digital Costa Rica, 2014).

2.2.1.3 APLICACIONES WEB

Las aplicaciones web son aquellas que como su nombre lo indica se ejecutan en la web o internet. Esto quiere decir que los archivos, datos o información con los que el usuario trabaja serán almacenados y procesados dentro de la web (Mora, 2002).

En otras palabras, toda la información se guardará en los grandes servidores de internet, siendo un claro ejemplo el almacenamiento en la nube, y a partir de esto se nos enviará a nuestro equipo o dispositivo cada dato que sea requerido o solicitado en ese preciso momento, quedando así una copia en nuestro equipo o PC de forma temporal.

2.2.2 BASE DE DATOS

2.2.2.1 MySQL

MySQL Database Service es un servicio como tal de base de datos el cual es completamente administrado para poder implementar aplicaciones nativas en la nube. HeatWave, un acelerador de consultas integrado y de alto rendimiento, aumenta el rendimiento de MySQL en 5400x (Oracle, 2022).

2.2.2.1.1 SQL

SQL es un lenguaje de programación de propósito especial diseñado para manejar datos en un sistema de administración de bases de datos relacionales. Un servidor de base de datos es un programa informático que proporciona servicios de base de datos a otros programas o computadoras, según lo define el modelo cliente-servidor. Por lo tanto, un servidor SQL es un servidor de base de datos que implementa el lenguaje de consulta estructurado (SQL).

CAPITULO III: METODOLOGIA DE INVESTIGACION Y DESARROLLO

En este capítulo se hablará sobre los temas que rodean al enfoque investigativo del proyecto “Tutor Inteligente”, en otras palabras, sobre la descripción de la investigación utilizada para el levantamiento de la información pertinente. Se presentará la respectiva metodología de investigación requerida a partir de un análisis previo para poder determinar cuál sería la mejor metodología a realizar, seguido de la debida técnica a utilizar para la compilación/recolección de los requerimientos necesarios para el levantamiento de información deseado. Para finalizar se incluirán todos y cada uno de los conceptos que tengan relación con la metodología de programación usada para el desarrollo del aplicativo web.

“La investigación científica se distingue por justificar sus conocimientos, por dar constancia de sus verdades, aunque éstas sean precarias y temporales, por eso es fundamentada, porque puede argumentar lo que para el investigador en ese momento histórico es cierto y se establece a través de los métodos de investigación, porque el investigador sigue procedimientos, desarrolla su tarea basándose en un plan previo, que se va ajustando en la práctica.”(Davila & Davila, 2000) .

3.1 METODOLOGIA DE LA INVESTIGACION

Bajo el término “metodología de investigación” hace referencia a todas y cada una de las decisiones que el investigador tomará para alcanzar sus objetivos planteados, las mismas que se enfocarán en aspectos como el diseño de la investigación, las estrategias a utilizar, las muestras a estudiar, los métodos empleados para recoger y analizar los datos, las técnicas seleccionadas para el análisis de la información y los criterios para incrementar la calidad del trabajo, entre otras (Binda & Balbastre-Benavent, 2013).

3.1.1 ENFOQUE DE LA INVESTIGACION

La investigación científica se concibe como un conjunto de procesos sistemáticos y empíricos que se aplican al estudio de un fenómeno; es dinámica, cambiante y evolutiva. Se puede manifestar de tres formas: cuantitativa, cualitativa y mixta (Davila & Davila, 2000). El enfoque mixto implica combinar las dos primeras (cuantitativa y cualitativa).

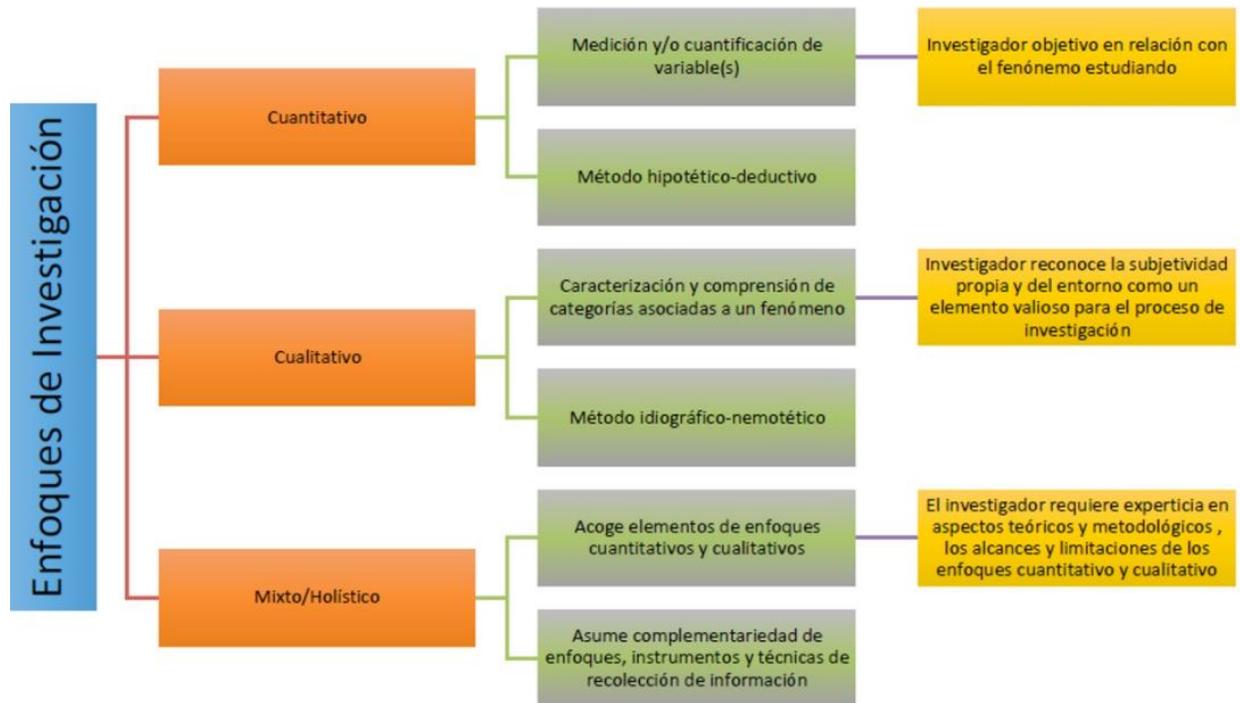


Figura 5: Enfoques investigativos

Fuente: (Davila & Davila, 2000)

3.1.1.1 INVESTIGACION CUALITATIVA

“La investigación cualitativa estudia la realidad en su contexto natural, tal y como sucede, intentando sacar sentido de, o interpretar, los fenómenos de acuerdo con los significados que tienen las personas implicadas” (Herrera, 2017).

En un tipo de investigación cualitativa se pretende utilizar técnicas para obtener datos mediante la interacción con las personas implicadas.

En el presente proyecto utiliza la investigación con un enfoque cualitativo ya que del mismo utiliza técnicas para la obtención masiva de datos y a la vez es muy útil para desarrollar visiones sistémicas del problema en concreto, tomando en cuenta todos y cada uno de los aspectos posibles de un fenómeno. Es decir, se puede trabajar con mayores detalles a la hora de realizar la investigación tomando en cuenta las múltiples observaciones amplias y panorámica a la hora de su desarrollo.

Otro factor que resalta al enfoque cualitativo es la flexibilidad que este pueda tener, siendo así posible la realización de cambios en la estructura de la investigación, cosa que no es posible en el enfoque cuantitativo.

3.1.2 TIPO DE INVESTIGACION

El tipo de investigación que se utilizará para este Trabajo de Integración Curricular es de tipo descriptivo, esta como tal a que se refiere al diseño de la investigación, creación de preguntas y análisis de datos que se llevarán a cabo sobre el tema. Se conoce como método de investigación observacional porque ninguna de las variables que forman parte del estudio está influenciada. Es un proceso muy importante tanto en investigación cuantitativa como la cualitativa. Otras, como la investigación exploratoria, se basan en ella para llevarse a cabo.

Entre los tipos de investigación descriptiva, tenemos:

- **Método de observación:** Siendo este el más eficiente para poder llevar a cabo la investigación descriptiva. Se puede utilizar tanto en la observación cuantitativa (recopilación de datos tales en números y valores) como la cualitativa (esta como tal mide características de los elementos a investigar).
- **Método de estudio de caso:** Implican un análisis profundo y el estudio de individuos o grupos. Suele conducir a una hipótesis y amplía el alcance de la investigación de un fenómeno.
- **Encuesta de Investigación:** Una herramienta muy popular para la investigación de mercado y en aulas (Economipedia, 2018).

Como paso uno, para realizar la debemos plantear de forma concisa y concreta las preguntas de investigación. En otras palabras, debemos tener claro lo que buscamos.

Luego escoger el método y que indicadores se van a utilizar para así identificar los que nos servirán durante toda la investigación, cabe recalcar que para conocer este paso hay que conocerlos de gran manera.

Como tercer paso tenemos que analizar, en otras palabras, poner en práctica lo ya mencionado anteriormente. En este caso hay que hacer el trabajo de campo, en donde

debemos medir, resumir, dividir, clasificar y, en definitiva, describir. Para finalizar y como último punto debemos interpretar, con los datos en mano, se debe entender que se muestra.

3.1.3 METODO DE INVESTIGACION

3.1.3.1 MÉTODO EXPERIMENTAL

El método de investigación del sistema tutor inteligente, utilizará el método experimental. Ya que, en el mismo, el investigador manipula una o más variables de estudio, para así poder controlar el aumento o disminución de esas variables y su efecto en las conductas observadas. En otras palabras, un experimento investigativo consiste en hacer un cambio en el valor de una variable, siendo esta una independiente, y observar su efecto en otra variable (variable dependiente).

Esto se lleva a cabo en condiciones rigurosamente controladas, con el fin de describir de qué modo o por qué causa se produce una situación o acontecimiento particular (Bastar, 2012). Los métodos experimentales son los adecuados para poner a prueba hipótesis de relaciones causales, un ejemplo simple de una investigación experimental es una prueba de laboratorio. Siempre que la investigación se realice bajo condiciones científicamente aceptables, se califica como una investigación experimental.

Como ventajas podemos decir que:

- ✓ Los investigadores tienen un control más fuerte sobre las variables para obtener los resultados deseados.
- ✓ El sujeto o la industria no es un criterio para la investigación experimental debido a que cualquier industria puede implementarlo con fines de investigación.
- ✓ Los resultados son extremadamente específicos.
- ✓ La causa y el efecto de una hipótesis puede dar pie a que los investigadores puedan analizar mayores detalles (QuestionPro, 2018).

3.1.4 TECNICA DE INVESTIGACION

3.1.4.1 ENTREVISTA

La entrevista como tal es una técnica de mucha utilidad en gran parte de investigaciones cualitativas con el propósito general de recopilar información; ésta en

concreto se define como un diálogo o comúnmente conocido como conversación con un fin determinado propuesto, diferenciándose del simple hecho de solo hablar. Según define Canales (Canales, 2006) "la comunicación interpersonal establecida entre el investigador y el sujeto de estudio, a fin de obtener respuestas verbales a las interrogantes planteadas sobre el problema propuesto". A partir de esto Heinemann propone sustentarla y complementarla con el uso de otro tipo de estímulos ya sea el contacto visual con el claro objetivo de obtener información útil para poder resolver la pregunta del tema central propuesto a desarrollar (Díaz-Bravo et al., 2013).

La información se obtiene de fuentes primarias, siendo en este caso una entrevista al Ing. José Miguel Erazo Ayón, docente de la materia de Lenguajes de Programación. "Las técnicas específicas de la investigación de campo tienen como finalidad recoger y registrar ordenadamente los datos relativos al tema escogido como objeto de estudio. Equivalen, por tanto, a instrumentos que permiten controlar los fenómenos" (Baena Paz, 2014, p. 12).

En la presente investigación se recopilaron los siguientes tópicos:

- ✓ Tipos de Aprendizaje
- ✓ Protocolos de Enseñanza-Aprendizaje
- ✓ Programa de estudio
- ✓ Evaluaciones

3.1.4.2 TECNICAS DE ENTREVISTA

Realizar entrevistas es muy útil y ventajoso sobre todo en estudios descriptivos y en fases de exploración, así como también para diseñar herramientas de recolección de datos (en este caso en concreto, la entrevista para la investigación cualitativa, independientemente del modelo que se desee usar.

Esta se determina por los siguientes elementos: tiene como propósito claro y conciso, el cual es obtener la información concreta en relación con un tema determinado; se pretende conseguir los significados que los informantes atribuyen a los temas en cuestión; el entrevistador debe mantener una actitud activa durante el desarrollo de la entrevista, en la que la interpretación sea continua con la finalidad de obtener una

compresión profunda del discurso del entrevistado) (Díaz-Bravo et al., 2013). Con frecuencia la entrevista se complementa con otras técnicas de acuerdo a la naturaleza específica de la investigación.

	ENTREVISTA ESTRUCTURADA	ENTREVISTA NO ESTRUCTURADA
VENTAJAS	<ul style="list-style-type: none"> • tiene una buena elaboración para facilitar su respuesta • fácil de manejar y examinar • evalúa tanto respuestas como preguntas • el entrevistador se prepara previamente para que salga bien la entrevista • las entrevistas son más fáciles de desarrollar por lo que son más cortas 	<ul style="list-style-type: none"> • hay mayor facilidad de realizar preguntas abiertas • el entrevistador puede obtener información sobre temas más sensibles • se puede alargar la entrevista y hacer preguntas sobre temas que no se trataron • se pueden omitir las preguntas o respuestas que crea el entrevistador no son importantes
DESVENTAJAS	<ul style="list-style-type: none"> • tiene gran valor el realizar una entrevista estructurada • el entrevistador puede que intimide o haga que la respuesta no sea tan fresca • no se puede obtener tanta información personal como es deseado 	<ul style="list-style-type: none"> • puede utilizarse mal el tiempo que se toma para la entrevista • se requiere mayor tiempo para esta entrevista

Figura 6: Técnicas de entrevistas estructuradas y no estructuradas

Fuente: (Cuadros comparativos entre entrevista y encuesta | Cuadro Comparativo, 2018)

Teniendo esto en cuenta el método cualitativo es parte fundamental en este trabajo, se realizó una entrevista al Ing. José Miguel Erazo docente de la materia Lenguajes de Programación de la Universidad Católica Santiago de Guayaquil con el objetivo de estudiar los distintos recursos académicos a utilizar, protocolos pedagógicos o de estudio, formas de llevar las clases para que sean interactivas.

3.1.5 ANALISIS DE RESULTADOS

Una vez realizada la entrevista, se procedió a analizar cada una de las respuestas del Ing. José Miguel Erazo, docente de la materia Lenguajes de Programación de la Universidad Católica Santiago de Guayaquil, con la finalidad de poder analizar temas

como los tipos de aprendizaje, los protocolos de enseñanza, programas de estudio y evaluaciones para poder establecer un patrón que regularmente se sigue, y que a su vez se pueda utilizar o mejorar en el sistema de tutor inteligente que se plantea en el presente proyecto. En el ANEXO#2 se detalla la entrevista con el docente.

Con la información obtenida de la entrevista se determinó que en el proceso de enseñanza y aprendizaje de la materia Lenguajes de programación sigue un protocolo de pedagogía regular, o también conocido como enseñanza general en donde el docente y el estudiante son los protagonistas ya sea a nivel virtual o presencial.

Varían de cierta forma con la asignación de mini-proyectos y realización de cursos de auto-aprendizaje tal y como lo es Solo-Learn en donde el estudiante al aprobar dicho curso, se le otorgara un certificado que, por lo general según lo mencionado por el Ing. José Erazo, es considerado como una nota.

Otro aspecto a destacar, es el hecho de que la materia cuente con ayudantías de catedra, esto no significa que el estudiante aprenda, ya que depende netamente del estudiante si desea mejorar sus conocimientos o no.

Como resultado del análisis de la entrevista realizada, se llegó a la conclusión de que es necesaria una implementación del proceso de aprendizaje del estudiante, a través de un sistema de tutor inteligente el cual será desarrollado, empleando una página web que pueda facilitar la enseñanza y el aprendizaje con estilo de aprendizaje pragmático, a fin de promover aún más, el auto aprendizaje. El tutor como tal podría servir de manera complementaria a la formación que se lleva en la carrera, o como una complementación de las ayudantías de catedra. A su vez serviría de gran ayuda para fortalecer temas, tópicos y vacíos que tenga el estudiante en su aprendizaje de los lenguajes de programación como tal.

3.2 METODOLOGIA DE DESARROLLO

3.2.1 METODOLOGIA DE PROTOTIPADO RAPIDO

El prototipo es uno de los medios usados en diseño para hacer avanzar la definición del producto y demostrar sus capacidades.

Mientras que el prototipo rápido es el proceso que permite producir un prototipo muy rápidamente (de media entre el 1 y el 5% del tiempo total del diseño).

El prototipo rápido, salvo casos muy particulares en los que se busca mucha representatividad (materiales, funcionalidades) tiene un coste pequeño (de media entre el 0.1 y el 1% del coste total del diseño).

Estos como tal tienen las siguientes características:

- Los potenciales usuarios interactúan con este prototipo y dan su información al diseñador en fases tempranas del diseño (lo que contribuye a definir mejor los requisitos del producto).
- El diseñador hace pruebas funcionales y manipulaciones lo que le lleva a comprender mejor el producto y darse cuenta anticipada de posibles errores.

Este método está asociado a la idea de desarrollar diferentes conceptos propuestos mediante prototipos de software o hardware, para su posterior evaluación. El desarrollo de la simulación o prototipado del sistema futuro puede ser de gran ayuda, permitiendo a los usuarios visualizar el sistema e informar sobre el mismo pudiéndose utilizar para aclarar opciones sobre los requerimientos de usuario y para especificar detalles de la interfaz de usuario a incluir en el sistema futuro (Jacobs & Reid, 1992).

Para definir el prototipo rápido se debe seguir los siguientes pasos

1. Definir y concretar los requisitos del proyecto

Durante esta fase, las partes implicadas deben trabajar juntas para definir y concretar los requisitos del proyecto, como los objetivos, las expectativas, los plazos y el presupuesto. Una vez que se hayan establecido y definido claramente todos los aspectos de los requisitos del proyecto, es el momento de pasarlos a la dirección para que los aprueben (Bruno Pelaez, 2018).

2. Comenzar a diseñar los prototipos

En cuanto se haya terminado de definir el alcance del proyecto se puede comenzar la fase de desarrollo. Los diseñadores y desarrolladores trabajarán estrechamente con los clientes con el objetivo de crear y mejorar los prototipos que ya están en marcha hasta que el producto final esté preparado.

3. Recopilación de las opiniones del usuario

En esta fase, los prototipos y los sistemas beta se convierten en modelos de trabajo. Entonces, los desarrolladores recogen la información que proporcionan los usuarios para corregir y mejorar los prototipos y crear un producto con la mayor calidad posible.

4. Pruebas, pruebas y más pruebas

En esta fase es necesario poner a prueba el producto de software y asegurarse de que todos los engranajes funcionan juntos sin problemas para satisfacer las expectativas del cliente. Se sigue incorporando la opinión del cliente a medida que el código se pone a prueba una y otra vez hasta que su funcionamiento sea impecable.

5. Presentación del sistema

Esta es la fase anterior al lanzamiento del producto terminado. En ella se integran la conversión de datos y la formación del usuario (Bruno Pelaez, 2018).

CAPITULO IV: DISEÑO DEL PROTOTIPO

4.1 ARQUITECTURA DEL PROTOTIPO

4.1.1 DIAGRAMA DEL PROTOTIPO

El diseño del prototipo Tutor Inteligente consta de tres módulos, siendo estos: Módulo del tutor, Módulo del estudiante y módulo del Dominio, estos están interconectados entre sí por medio de una interface, en el diagrama se muestra los componentes de cada uno de ellos, a continuación, se mostrará el detalle en la siguiente figura:

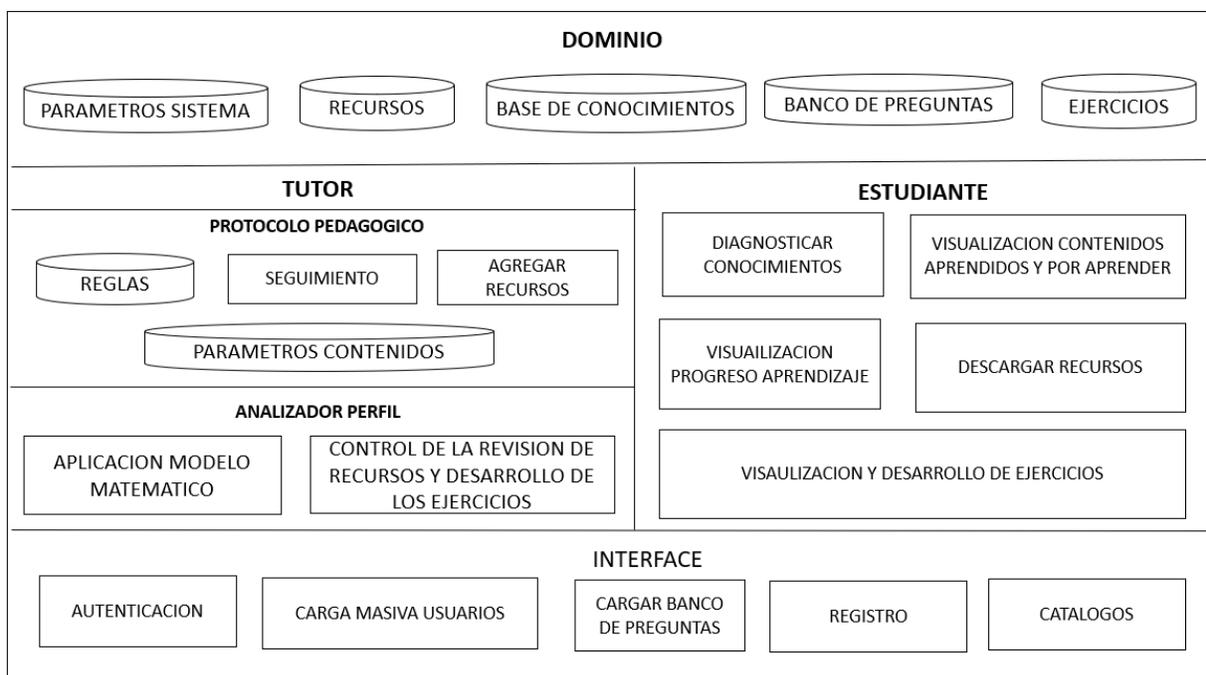


Figura 7: Diagrama del Prototipo de un Sistema de Tutor Inteligente

Fuente: Creación Propia

4.1.2 COMPONENTES Y DOCUMENTOS FUNCIONALES

El diseño del prototipo de un sistema tutor inteligente consta de 3 módulos principales:

4.1.2.1 MODULO DE DOMINIO

Este módulo permitirá organizar la base de Conocimientos de la materia Lenguaje de Programación, administrar recursos pedagógicos que servirán para el aprendizaje del

estudiante, gestionar los parámetros del sistema y gestionar el banco de preguntas que serán utilizadas en las evaluaciones de conocimientos.

A continuación, el detalle de los componentes de este módulo:

4.1.2.1.1 PARÁMETROS DEL SISTEMA

Por medio de este componente el administrador puede configurar diferentes variables que se utilizan en la operación del prototipo:

1. Variables para utilizar en el modelo matemático
 - Porcentaje del Valor por Revisión de Recursos (VPPR)
 - Porcentaje del Valor por Evaluaciones (VPRC)
 - Valor de Ponderación (VBP)
 - Valor de Aprobación del Nivel de Conocimiento (PR)
2. Cantidad de preguntas que se mostraran en las evaluaciones realizadas por los estudiantes
3. Tamaño de los recursos (archivos) a cargar al Sistema

4.1.2.1.2 BASE DE CONOCIMIENTOS

La organización de la base de conocimientos configurado en el prototipo de un sistema tutor inteligente es la siguiente:

1. **Unidad:** Estructura principal del conocimiento
 - 1.1 **Tema:** Componente secundario del conocimiento
 - 1.1.1 **Contenidos:** Corresponden a los subtemas
 - 1.1.1.1 **Recursos:** Son los documentos, videos, enlaces que son cargados al sistema para la revisión de los estudiantes

Para un eficiente proceso de enseñanza y aprendizaje de los contenidos de la materia de Lenguaje de Programación se establecieron Niveles.

En la figura #13, se podrá apreciar la estructura Unidad – Tema – Contenidos – Recursos asociadas a los respectivos niveles y recursos pedagógicos para poder avanzar de nivel.

BASE DE CONOCIMIENTOS					
	Unidad	Lenguajes Imperativos			
	Tema	Lenguajes Imperativos			
Niveles	BASICO	Contenido por niveles	INTRODUCCION A LOS LENGUAJES IMPERATIVOS	Recursos pedagógicos	Introduccion_a_la_programacion_imperativ.pdf
			MANEJO DE DATOS Y TIPOS		Introduccion_tipos_datos.pdf
			ASIGNACIONES Y EXPRESIONES		Programacion_con_Asignaciones_III.pdf
	INTERMEDIO		FLUJOS DE CONTROL		Lenguaje_imperativo_flujos.pdf
			COMPONENTES DE UN PROGRAMA IMPERATIVO		Lenguaje_imperativo_comp.pdf
	AVANZADO		USO DE LENGUAJES DE PROGRAMACION PREPROCESADOR DE HYPER TEXTO		teoria_php.pdf
			EJERCICIOS DE PROGRAMACION PREPROCESADOR DE HYPER TEXTO		ejercicios_php.pdf

Figura 8: Base de Conocimientos categorizada por niveles de los Lenguajes Imperativos

Para la prueba del prototipo se tomó como referencia para la enseñanza y aprendizaje, la unidad y tema: LOS LENGUAJES IMPERATIVOS, con sus respectivos contenidos que se pueden apreciar en la figura 14 y en el ANEXO #1.

LENGUAJES IMPERATIVOS		
TEMAS	CONTENIDOS	No. DE HORAS PARA CADA TEMA
LENGUAJES IMPERATIVOS	INTRODUCCIÓN	02:00
LENGUAJES IMPERATIVOS	MANEJO DE DATOS Y TIPOS	02:00
LENGUAJES IMPERATIVOS	ASIGNACIONES Y EXPRESIONES	02:00
LENGUAJES IMPERATIVOS	FLUJOS DE CONTROL	02:00
LENGUAJES IMPERATIVOS	COMPONENTES DE UN PROGRAMA IMPERATIVO	02:00
LENGUAJES IMPERATIVOS	EJEMPLOS DE PROGRAMAS IMPERATIVOS	02:00
LENGUAJES IMPERATIVOS	USO DE LENGUAJES DE PROGRAMACIÓN PREPROCESADOR DE HYPER TEXTO	04:00
LENGUAJES IMPERATIVOS	MANEJO DE LENGUAJE DE PROGRAMACIÓN MULTIPARADIGMA	04:00

Figura 9: Ultimo Syllabus de la materia Lenguajes de Programación

Fuente: Ing. José Erazo, docente de la materia Lenguajes de Programación de la UCSG

4.1.2.1.3 REGLAS DE LA BASE DE CONOCIMIENTOS

Las reglas son utilizadas para controlar y establecer el comportamiento de la base de conocimientos, a continuación, el detalle:

1. Los contenidos de la materia están jerarquizados por Unidad y Tema
2. Los contenidos están configurados en base a un Syllabus
3. Los niveles de conocimientos están catalogados con Básico, Intermedio y Avanzado.
4. Todos los estudiantes inician por el nivel Básico.
5. Los contenidos están configurados por un orden secuencial de aprendizaje.
6. Para llevar un eficiente seguimiento del aprendizaje los contenidos deben tener un valor de ponderación.
7. La suma de los valores o puntajes por todos los contenidos configurados por unidad deben sumar 100 puntos.
8. El valor o puntaje de ponderación debe estar relacionado por el nivel de complejidad.

4.1.2.1.4 RECURSOS PEDAGÓGICOS

Son los contenidos o el material proporcionado por el docente para la enseñanza y aprendizaje de la materia Lenguajes de Programación, pudiendo ser estos: archivos .pdf, videos, planteamiento de ejercicios, ejemplos de código que se puedan realizar.

Siendo estos los mencionados en la tabla 4, los tipos de recursos a utilizar serian, archivos pdf, videos, links, entre otros.

Tabla 4: Recursos Digitales o Pedagógicos

Tipos de Recursos Digitales	Descripción
Presentaciones de Power Point	Son recursos pedagógicos que proporcionan bases de consulta para cada tema del portal web. Cada presentación contiene un conjunto de

	diapositivas con explicaciones y ejemplos del tema respectivo
Tareas, Talleres	Son actividades integradoras para realizarse por los estudiantes
Soluciones o códigos	Son recursos que muestran la solución a un ejemplo práctico o taller que se realizó en clases. Estos recursos muestran código en archivos de texto.
Archivos o documentos	Como sus nombres lo indican son documentos, archivos, artículos que se pueden leer, analizar

Estos recursos pedagógicos y ejercicios van a estar relacionados directamente con los contenidos

4.1.2.1.5 BANCO DE PREGUNTAS

En el banco de preguntas contendrán preguntas objetivas relacionadas entre cada uno de los contenidos a través de los respectivos niveles, siendo estos: básico, intermedio y avanzado.

Cada pregunta tendrá 4 opciones de respuesta donde una es correcta y 4 incorrectas. Dentro del banco de preguntas se encuentra información relacionada a los recursos a revisar y la retroalimentación que se presenta al estudiante al responder de manera incorrecta.

4.1.2.2 MODULO TUTOR

Este módulo estará conformado por dos sub-módulos, siendo el primero el protocolo pedagógico y el segundo el analizador del perfil.

4.1.2.2.1 PROTOCOLO PEDAGÓGICO

En este sub-módulo se encuentran cada una las reglas que se deben cumplir para poder determinar el estado del aprendizaje del estudiante. Así mismo el poder controlar el seguimiento del estudiante.

4.1.2.2.1.1 CONSIDERACIONES

Para controlar el avance en el aprendizaje se establecieron las siguientes condiciones dentro de este módulo las cuales se describen a continuación:

1. Los contenidos registrados están organizados de manera secuencial y en niveles.
2. La parte inteligente del prototipo es revisar lo registrado en la base de datos sobre los avances del aprendizaje del estudiante y solo presentar los contenidos con sus respectivos recursos y ejercicios asociados de acuerdo al nivel de conocimiento para ser revisados y ejecutados. Luego de controlar si se realizó la revisión de los recursos y ejecución de los ejercicios el prototipo le habilita automáticamente la evaluación correspondiente.
3. El progreso del aprendizaje es por medio de grafica de barras con los respectivos contenidos aprendidos y un porcentaje de avance con los ejercicios realizados.
4. Para medir el conocimiento de los contenidos se deben realizar evaluaciones.
5. Las evaluaciones se realizan por las preguntas cargadas en lote.
6. Las preguntas están relacionadas por unidad, tema y contenido.
7. Para poder rendir las evaluaciones los estudiantes deberán revisar los recursos pedagógicos y realizar los ejercicios propuestos.
8. Existe un porcentaje de revisión y porcentaje de realización de ejercicios que deben cumplir los estudiantes para poder ejecutar las evaluaciones.
9. Al concluir la evaluación el prototipo presentara un resumen de preguntas correctas e incorrectas, además indicara los recursos a ser revisados para las respuestas que no fueron correctas.
10. El porcentaje de revisión y desarrollo de ejercicios es diferente para cada contenido. Estos parámetros son configurados por el tutor.
11. La actualización del perfil el prototipo lo realiza mediante la aplicación de un modelo matemático el cual determina si el estudiante ha revisado los

recursos, ejecutado los ejercicios y ha realizado correctamente las evaluaciones.

12. Para completar el aprendizaje de la unidad aprobar los niveles desde el básico hasta el Avanzado.

4.1.2.2.1.2 REGLAS DEL SISTEMA

1. Si nivel conocimiento es Básico o Intermedio o Avanzado entonces muestra los contenidos, recursos y ejercicios asociados al nivel.
2. Si recurso revisados y ejercicio ejecutado por contenido entonces actualizar avance detallado.
3. Si se cumplió la totalidad el avance detallado de los recursos y ejercicios ejecutados entonces se avanza al nuevo contenido.
4. Si porcentaje total de Recursos revisados y ejercicios ejecutados por contenido son mayores o iguales al porcentaje Total de Recursos y Ejercicios propuestos entonces habilitar evaluación, caso contrario el sistema le habilita los recursos y ejercicios que tiene que revisar y ejecutar. Cabe indicar que los porcentajes de revisión y ejecución son parametrizables y deben ser configurados por cada contenido.
5. Si no se completa el porcentaje de respuestas correctas entonces el prototipo presenta una retroalimentación que corresponde a los recursos, caso contrario el estudiante debe nuevamente revisar los recursos para rendir una nueva evaluación.
6. Si Porcentaje de respuestas correctas alcanzadas por el estudiante es mayor o igual al Porcentaje de Respuestas Correctas por Contestar entonces actualizar el nivel de aprendizaje del estudiante caso contrario el estudiante se mantiene en el mismo nivel y deberá seguir rindiendo las evaluaciones.
7. Si repuestas correctas son mayores o iguales al porcentaje respectivo y nivel de conocimiento es avanzado entonces el Estudiante conoce totalmente la Unidad y Tema caso contrario deberá seguir rindiendo las evaluaciones.

4.1.2.2.1.3 MAQUINA DE INFERENCIA.

La máquina de inferencia del sistema tutor, debe establecer comunicación con el módulo tutor y ejecutar las inferencias sobre cada respuesta del estudiante, tomando en consideración no solo la respuesta como tal, sino también las interacciones del estudiante (traza de seguimiento) con el sistema durante todo el aprendizaje. De esta manera si un estudiante no alcanza el nivel de conocimiento requerido para ser promovido al siguiente, el sistema tutor a través de la máquina de inferencia y tomando como referencia la traza, podría disminuir los niveles de dificultad presentando al estudiante alternativas para la comprensión de los contenidos. La máquina de inferencia debe tener en cuenta el progreso del estudiante en función de respuestas correctas y ponderaciones, para que según las reglas de cada nivel, pueda realizar las acciones de promoción al siguiente nivel, degradarlo de nivel o mantener al estudiante en el mismo nivel. Este componente no debe ser diseñado e implementado en un próximo trabajo.

4.1.2.2.1.4 SEGUIMIENTO

El tutor podrá visualizar los avances que tiene el estudiante en relación a los contenidos que va aprobando.

4.1.2.2.1.5 AGREGAR RECURSOS

El tutor debe adicionar los recursos que deben revisar los alumnos para adquirir el conocimiento de los contenidos de la materia. Consideraciones en la carga de los recursos:

- Los recursos están relacionados por Unidad – Tema – Contenido.
- Los recursos se los puede categorizar por tipo, a continuación, el detalle de los diferentes tipos:
 1. Imagen
 2. Video
 3. Enlace
 4. Presentación
 5. Documento PDF
 6. Documento WORD
 7. Ejercicio

- Para subir ejercicios propuestos se deberá ingresar el recurso tipo ejercicio (7)

4.1.2.2.1.6 PARÁMETROS – CONTENIDO

Son parámetros que se deben cumplir para poder determinar el estado del aprendizaje del estudiante, así mismo el poder diagnosticar el nivel y el avance del aprendizaje que tiene el estudiante.

Los parámetros de avance por contenido están expresados en porcentaje y divididos en dos segmentos:

- Porcentaje de Revisión de Recursos (PPRR)
- Porcentaje de Revisión de Ejercicios (PRE)

4.1.2.2.2 ANALIZADOR DEL PERFIL

Para la actualización del perfil del estudiante se debe aplicar el modelo matemático. Este modelo revisa de acuerdo a su configuración si el estudiante está apto de avanzar el nivel.

Este sub-módulo contiene los siguientes componentes, el primero es el modelo matemático y Control Revisión de Contenidos y Desarrollo de Ejercicios

4.1.2.2.2.1 MODELO MATEMATICO

A continuación, se expondrá el modelo matemático implementado en el Sistema Tutor Inteligente para la categorización del nivel de conocimiento que posee el estudiante en los Lenguajes Imperativos de la materia de Lenguaje de Programación.

4.1.2.2.2.1.1 NIVELES

1. B=Básico
2. I=Intermedio
3. A=Avanzado

4.1.2.2.2.1.2 VARIABLES

RR=Recursos Revisados

RC=Respuestas Correctas

4.1.2.2.2.1.3 MODELO

VPPRR=Valor del porcentaje de ponderación de la revisión de recursos

VPRC=Valor del porcentaje de ponderación de las respuestas correctas

PRR=Puntaje Recursos Revisados

PRC=Puntaje Respuestas Correctas

FRR=Factor Recursos Revisados

FRC=Factor Respuestas Correctas

PR=Puntaje Requerido

NTRXR=Número Total de Recursos por Revisar

NTPXC=Número Total de Preguntas por Contestar

NTRR=Número Total de Recursos Revisados

NTPC=Número Total de Preguntas Contestadas

VBP=Valor base de Ponderación

$FRR = VBP / NTRXR$

$FRC = VBP / NTRXC$

$PRR = NTRR * FRR$

$PRC = NTRC * FRC$

$VPRR * PRR + VPRC * PRC \geq PR$

4.1.2.2.1.4 EJEMPLO DE MODELO MATEMATICO

Como ejemplo para el modelo matemático, podemos tomar: el Valor del porcentaje de ponderación de la revisión de recursos, Valor del porcentaje de ponderación de las respuestas correctas y un puntaje requerido parametrizable para dar a conocer la aprobación o no del estudiante.

VPRR=40%

VPRC=60%

PR=7

NTRXR=10

NTRXC=8

NTRR=7

NTRC=7

VBP=10

FRR=10/ 10 = 1

FRC=10/ 8=1.25

PRR=7*1=7

PRC=7*1.25 = 8.75

VPRR*PRR + VPRC*PRC >= PR

0.4*7 + 0.6*8,75 >=7

2.8 + 5.25 >=7

8.05 >=7 por lo tanto el Nivel Aprobado

4.1.2.2.2 CONTROL DE LA REVISION DE RECURSOS Y DESARROLLO DE EJERCICIOS

El avance del aprendizaje está dado por la revisión de los contenidos y por la ejecución de los ejercicios propuestos por el tutor.

Para el control del avance el prototipo lo realiza por medio de la parametrización de la revisión de contenidos y ejercicios.

La parametrización del control está dada en porcentaje. Este control se lo realiza para poder habilitar la opción de rendir los exámenes por parte del estudiante.

4.1.2.3 MODELO ESTUDIANTE

En este módulo se permite poder realizar las pruebas para medir el aprendizaje, visualizar los contenidos y avance del aprendizaje

Así también el sistema tutor inteligente en el módulo del estudiante contiene cinco componentes diferentes opciones que permiten la interacción de los módulos entre sí, estos son:

4.1.2.3.1 DIAGNOSTICAR CONOCIMIENTOS

El estudiante puede realizar las evaluaciones para medir el grado de conocimientos que ha adquirido en base a las revisiones de los recursos y desarrollo de los ejercicios.

El prototipo contiene las siguientes consideraciones para realizar las evaluaciones:

- Las cantidades de preguntas a resolver por parte del estudiante son parametrizables.
- Por cada pregunta el prototipo presenta cuatro opciones, de las cuales el estudiante debe escoger una opción que se considere correcta.
- Al final de responder las preguntas se presentará un resumen de intentos y el detalle de las respuestas correctas e incorrectas y presentará además una retroalimentación indicando los recursos que deberá revisar de nuevo en aquellas respuestas que fueron incorrectas.

4.1.2.3.2 VISUALIZAR PROGRESO

Mediante esta opción el estudiante podrá visualizar el avance en el aprendizaje de los contenidos. El progreso esta dado de manera general y de manera detallada.

El avance general estará dado por el puntaje acumulado por los contenidos revisados y el avance detallado estará dado por cada uno de los contenidos revisado por el estudiante.

4.1.2.3.3 VISUALIZACIÓN DE CONTENIDOS Y RECURSOS

El estudiante podrá visualizar los contenidos y recursos aprendidos y por aprender por niveles.

El estudiante podrá visualizar los contenidos por aprender y los contenidos aprendidos, así mismo el prototipo presentara mediante un panel cuales son los recursos revisados y por revisar.

El prototipo presentara los contenidos a ser revisados mediante la configuración realizada en la Base de Conocimientos previamente.

4.1.2.3.4 DESCARGA DE RECURSOS

El estudiante para revisar los recursos deberá descargarlos, estos los utilizará para prepararse y poder rendir las evaluaciones y obtener los conocimientos necesarios para dominar la materia.

Los recursos a ser descargados están organizados por Unidad – Tema y Contenido.

En el caso de los ejercicios el estudiante podrá visualizar el ejercicio y desarrollarlo. Una vez desarrollado el ejercicio se lo considerara como revisado,

4.1.2.3.5 VISUALIZACIÓN Y DESARROLLO DE EJERCICIOS

El desarrollo de los ejercicios es una variable que el estudiante debe cumplir para poder rendir las evaluaciones y avanzar de nivel. Este desarrollado debe ser ejecutado por el estudiante para poder rendir las evaluaciones y avanzar de nivel.

4.1.2.4 INTERFACE

Dentro de este módulo se encuentran componentes que permiten la interacción de los diferentes usuarios que utilizan el Sistema Tutor Inteligente, los componentes se detallan a continuación:

4.1.2.4.1 AUTENTICACIÓN

Los usuarios para acceder al sistema deben contar con una cuenta y con una clave. Así mismo el usuario debe tener asignado el respectivo rol. Los roles que tiene el sistema son:

- a. Estudiante
- b. Tutor
- c. Administrador.

De acuerdo al rol se mostrarán las diferentes opciones que contiene el sistema.

4.1.2.4.2 CARGA MASIVA DE USUARIOS

Mediante este proceso el sistema registra los usuarios que van a utilizar el Sistema tutor Inteligente. Los usuarios que se van a cargar en el sistema son aquellos que tienen el rol de estudiante.

4.1.2.4.3 CARGAR BANCO DE PREGUNTAS

Este proceso consiste en cargar de manera masiva las preguntas que van a ser utilizadas para las evaluaciones que deben realizar los estudiantes para identificar si poseen los conocimientos necesarios para ir aprobando los niveles de aprendizaje que contiene el sistema tuto inteligente.

4.1.2.4.4 REGISTRO

Esta opción permite registrar a los usuarios de manera específica, aquellos estudiantes que no fueron registrados de manera masiva.

4.1.2.4.5 CATÁLOGOS

La organización del conocimiento es por medio de Unidad, Temas, Contenidos y Recursos, por lo que se deben registrar las unidades y los temas que se van a enseñar a los estudiantes

4.1.3 REQUERIMIENTOS DE HARDWARE

Para el diseño del prototipo del sistema tutor inteligente se han identificado los requerimientos mínimos de hardware que debería implementarse para el correcto funcionamiento de la solución, a continuación, se detallan los componentes físicos que debería tener instalado en cada uno de los equipos tecnológicos:

4.1.3.1 SERVIDOR DE BASE DE DATOS

Tabla 5: Servidor de Base de Datos de Hardware

PROCESADOR	64 BITS DE 1.4 GHZ
MEMORIA	4 GB
ALMACENAMIENTO	128 GB LIBRE
TARJETA DE RED	100 Mbit/s a 1000 Mbit/s

4.1.3.2 SERVIDOR WEB

Tabla 6: Servidor Web de Hardware

PROCESADOR	64 BITS DE 1.4 GHZ
MEMORIA	4 GB
ALMACENAMIENTO	128 GB LIBRE

TARJETA DE RED	100 Mbit/s a 1000 Mbit/s
VELOCIDAD DE INTERNET	20 mbps

4.1.4 REQUERIMIENTOS DE SOFTWARE

4.1.4.1 SERVIDOR DE BASE DE DATOS

Tabla 7: Servidor Base de Datos a nivel de Software

SISTEMA OPERATIVO	WINDOWS SERVER 2016
MOTOR DE BASE DE DATOS	Mysql versión 5.7

4.1.4.2 SERVIDOR WEB

Tabla 8: Servidor Web a nivel de Software

SISTEMA OPERATIVO	WINDOWS SERVER 2016
SERVICIO	INTERNET INFORMATION SERVICES

4.1.4.3 AMBIENTE DE DESARROLLO

Tabla 9: Ambiente de Desarrollo de Software

LENGUAJE DE PROGRAMACION	VISUAL STUDIO 2017 COMMUNITY
ENTORNO DE DISEÑO	DEVEXPRESS
ENTORNO BASE DE DATOS	MYSQL QUERY BROWSER

4.1.5 ARQUITECTURA DE HARDWARE

Para la correcta ejecución del sistema tutor inteligente se implementará los siguientes equipos:

- Servidor WEB: Servicios y recursos compartidos
- Servidor Base Datos: Almacenamiento de la información de la solución respectiva

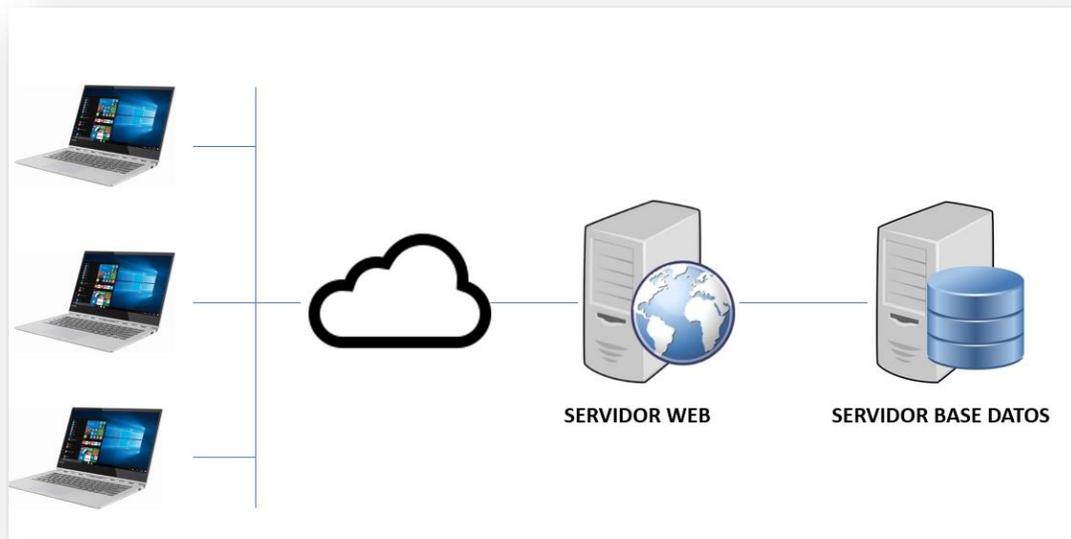


Figura 10: Arquitectura del Hardware

4.1.6 ARQUITECTURA Y FRAMEWORK DE SOFTWARE

Para la correcta ejecución del prototipo del sistema tutor inteligente se implementará los siguientes sistemas y aplicaciones:

Tabla 10: Sistemas y Aplicaciones de la arquitectura y Framework de software

Servidor WEB	Sistema Operativo Windows Server 2016
	Apache
Servidor Base Datos	Sistema Operativo Windows Server 2016
	Motor de Base de Datos Mysql versión 5.7

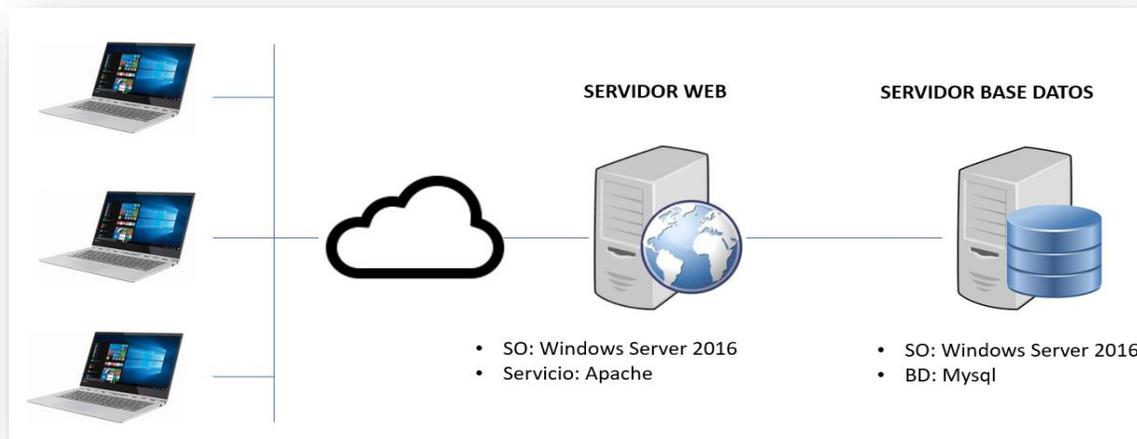


Figura 11: Arquitectura del Hardware a detalle

4.1.7 TECNOLOGIA DE LA BASE DE DATOS

El prototipo de sistema tutor inteligente tendrá como base de datos transaccional MySQL donde se registrarán los datos necesarios para la ejecución del sistema. Será una base de gran utilidad ya que trabaja con soporte para múltiples procesadores, soporta hasta 16 índices por tabla, es gratuita y de código abierto y sobre todo para las opciones de cliente donde se conectará con gran facilidad a MySQL a través de comandos TCP (Educacao, 2020). Entre otras funcionalidades que hacen de esta base de datos un lugar de fácil acceso y uso.



Figura 12: Logo de MySQL:

Fuente: (Oracle, 2022)

En la base de datos se registrar la información en las siguientes tablas:

1. TB_AUDITORIA
2. TB_CATALOGO
3. TB_CONTENIDO
4. TB_CURSO
5. TB_ESTUDIANTE_CLASE
6. TB_EVALUACION
7. TB_MENU
8. TB_PARAMETRO
9. TB_TEMARIO
10. TB_USUARIO

4.2 DIAGRAMA DE FLUJO DE DATOS

4.2.1 DIAGRAMA DE CONTEXTO

Mediante el siguiente diagrama se detallan el flujo de los datos del sistema tutor inteligente correspondientes a los perfiles de Estudiantes – Tutor – Administrador:

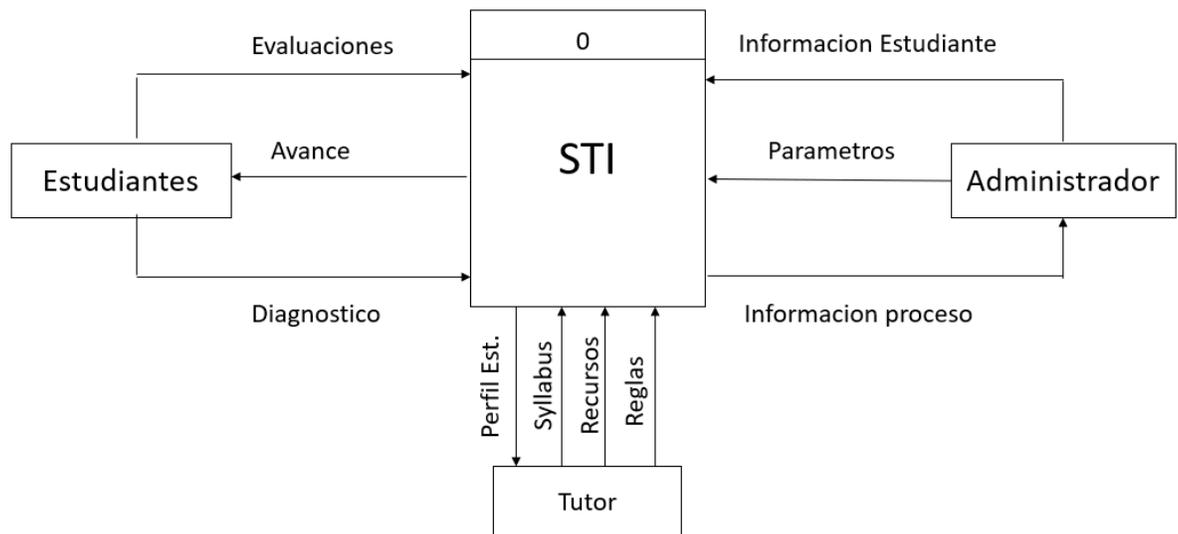


Figura 13: Diagrama de contexto del Sistema de Tutor Inteligente

La parte del tutor contendrá el syllabus que es el documento por oficial proporcionado por el docente, los recursos que son proporcionados por el tutor mismo, las reglas que son como parámetros que se deben seguir.

4.2.1.1 DIAGRAMA DE FLUJO DE DATOS NIVEL 1

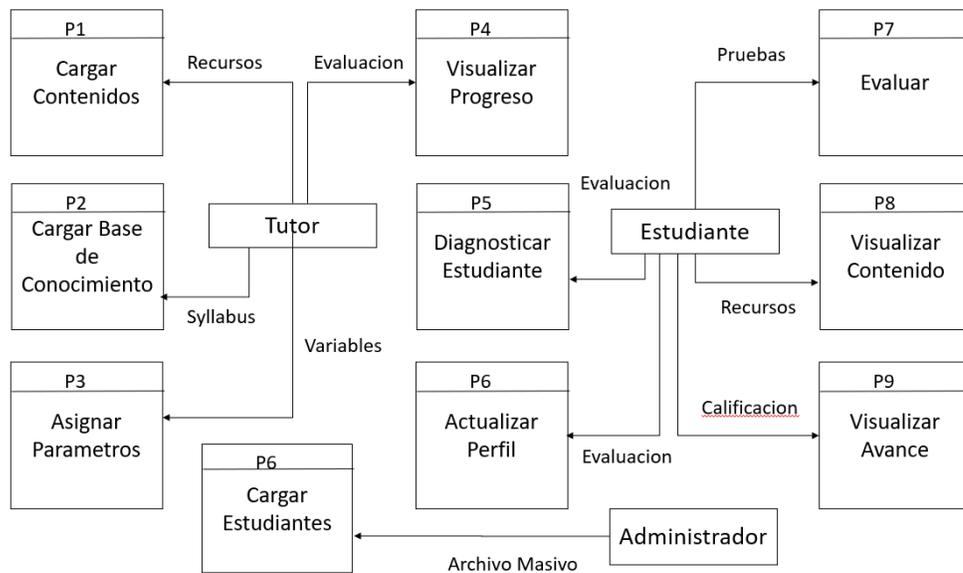


Figura 14: DIAGRAMA DE FLUJO DE DATOS NIVEL 1

En el siguiente flujo, se muestran cada uno de los apartados que contendrá el sistema del tutor

4.3 CASOS DE USO

A continuación, se presentan los casos de usos correspondientes a los roles de estudiante, tutor y administrador

4.3.1 CASOS DE USO INTERFASE DEL SISTEMA

CASO DE USO: CU-001

En este caso el usuario ingresa su usuario y contraseña y el sistema de tutor inteligente valida si existe registrado el usuario que previamente ingresó por defecto el administrador.

Tabla 11: Caso de Uso CU-001

Código	CU-001
Descripción	Autenticación de Credenciales
Actor	Estudiante – Tutor – Administrador
Objetivo	Validar las credenciales para ingreso al sistema
Detalle	El usuario ingresa su usuario y contraseña y el sistema valida si existe registrado
Secuencia	<ul style="list-style-type: none">• Ingreso de credenciales• Acceso al menú de opciones del sistema

CASO DE USO: CU-002

En este caso el protagonista es el administrador, el cual es el que se encarga de registrar a cada uno de los estudiantes y al tutor de la materia respectiva con sus usuarios y contraseñas debidos.

Tabla 12: Caso de Uso CU-002

Código	CU-002
Descripción	Registro de usuarios en el Sistema
Actor	Administrador
Objetivo	Registrar los usuarios para el uso del sistema
Detalle	El Administrador es el encargado en registrar a los estudiantes y al tutor de la materia lenguaje de programación
Secuencia	<ul style="list-style-type: none">• Registro de la información• Asignación de las opciones del sistema en base al rol

CASO DE USO: CU-003

En este caso cada usuario puede cambiar la contraseña a gusto, independientemente de la que se le haya creado por defecto, si este la desea usar o no. En caso de presentarse problemas se recurrirá al administrador para que realice los cambios respectivos.

Tabla 13: Caso de Uso CU-003

Código	CU-003
Descripción	Cambiar contraseña
Actor	Administrador – Estudiante – Tutor
Objetivo	Cambiar la contraseña en el sistema
Detalle	Los usuarios podrán realizar el cambio de contraseña
Secuencia	<ul style="list-style-type: none">• Ingreso información para identificar al usuario• Cambio de contraseña

CASO DE USO: CU-004

Tabla 14: Caso de Uso CU-004

Código	CU-004
Descripción	Carga masiva de información de estudiantes
Actor	Administrador
Objetivo	Registro masivo de estudiantes
Detalle	El administrador carga la información de los estudiantes mediante un archivo Excel
Secuencia	<ul style="list-style-type: none">• Seleccionar el archivo• Carga en lote los estudiantes al sistema

CASO DE USO: CU-005

Tabla 15: Caso de Uso CU-005

Código	CU-005
Descripción	Carga Banco de Preguntas
Actor	Administrador
Objetivo	Registro de preguntas para las evaluaciones
Detalle	El administrador carga la información de las preguntas mediante un archivo Excel
Secuencia	<ul style="list-style-type: none">• Seleccionar el archivo

	<ul style="list-style-type: none"> • Carga en lote las preguntas
--	---

CASO DE USO: CU-006

Tabla 16: Caso de Uso CU-006

Código	CU-006
Descripción	Menú dinámico de Opciones del sistema
Actor	Sistema
Objetivo	Mostrar las opciones de acuerdo al perfil del usuario
Detalle	El sistema mostrará de manera dinámica las opciones en el sistema en base al perfil del usuario
Secuencia	<ul style="list-style-type: none"> • Ingreso de Credenciales • Carga de opciones al menú

CASO DE USO: CU-007

Tabla 17: Caso de Uso CU-007

Código	CU-007
Descripción	Unidad
Actor	Administrador
Objetivo	Creación de las unidades de estudio
Detalle	El administrador registra la información de las unidades
Secuencia	<ul style="list-style-type: none"> • Registro de la Unidad

CASO DE USO: CU-008

Tabla 18: Caso de Uso CU-008

Código	CU-008
Descripción	Tema
Actor	Administrador
Objetivo	Creación de los temas
Detalle	El administrador registra la información de los temas

Secuencia	<ul style="list-style-type: none"> • Registro de los temas
-----------	---

4.3.2 CASOS DE USO MODULO DOMINIO

CASO DE USO: CU-009

Tabla 19: Caso de Uso CU-009

Código	CU-009
Descripción	Subir Recursos Pedagógicos
Actor	Tutor
Objetivo	Subir al sistema los recursos pedagógicos
Detalle	El sistema permitirá al tutor cargar al sistema los recursos pedagógicos que serán utilizados para la enseñanza de lenguaje de programación
Secuencia	<ul style="list-style-type: none"> • Seleccionar el recurso de acuerdo al tipo • Cargar al sistema

CASO DE USO: CU-010

Tabla 20: Caso de Uso CU-010

Código	CU-010
Descripción	Registro de Parámetros del sistema
Actor	Administrador
Objetivo	Registrar los parámetros básicos al sistema para su ejecución
Detalle	Mediante esta opción el administrador deberá registrar los parámetros para la operación del sistema tutor inteligente
Secuencia	<ul style="list-style-type: none"> • Ingresar parámetros del sistema • Registro de los parámetros del sistema

CASO DE USO: CU-011

Tabla 21: Caso de Uso CU-011

Código	CU-011
Descripción	Subir Ejercicios Propuestos
Actor	Tutor
Objetivo	Subir al sistema los ejercicios propuestos
Detalle	El sistema permitirá al tutor cargar al sistema los ejercicios que deben desarrollar los estudiantes
Secuencia	<ul style="list-style-type: none">• Seleccionar el ejercicio y en tipo de recurso se debe seleccionar ejercicios• Cargar al sistema

4.3.3 CASOS DE USO MODULO TUTOR

CASO DE USO: CU-012

Tabla 22: Caso de Uso CU-012

Código	CU-012
Descripción	Visualización del Progreso
Actor	Tutor
Objetivo	Visualizar el avance del progreso del aprendizaje del estudiante
Detalle	Mediante esta opción el tutor podrá visualizar el avance del aprendizaje de la materia por parte de un estudiante específico
Secuencia	<ul style="list-style-type: none">• Ingresar el código del estudiante• Visualización del progreso

CASO DE USO: CU-013

Tabla 23: Caso de Uso CU-013

Código	CU-013
Descripción	Base de Conocimiento

Actor	Tutor
Objetivo	Registrar los contenidos de la materia
Detalle	Mediante esta opción el tutor registrar en el sistema los conocimientos de la materia a enseñar
Secuencia	<ul style="list-style-type: none"> • Iniciar Sesión • Registro de contenidos

4.3.4 CASOS DE USO MODULO ESTUDIANTE

CASO DE USO: CU-014

Tabla 24: Caso de Uso CU-014

Código	CU-014
Descripción	Visualizar el Avance del aprendizaje del Estudiante
Actor	Estudiante
Objetivo	Mostrar los contenidos aprendidos y por aprender del estudiante
Detalle	Mediante esta opción estudiante podrá visualizar el avance del aprendizaje
Secuencia	<ul style="list-style-type: none"> • Iniciar Sesión • Visualizar avance en el aprendizaje

CASO DE USO: CU-015

Tabla 25: Caso de Uso CU-015

Código	CU-012
Descripción	Evaluación realizada por el estudiante
Actor	Estudiante
Objetivo	Ejecución de las pruebas para medir los conocimientos adquiridos por el estudiante
Detalle	Los estudiantes luego de la revisión de los recursos pedagógicos deberán realizar las pruebas

Secuencia	<ul style="list-style-type: none"> • Ejecución de las pruebas • Presentación de la calificación
-----------	---

CASO DE USO: CU-016

Tabla 26: Caso de Uso CU-016

Código	CU-016
Descripción	Visualización y descarga de los recursos pedagógicos
Actor	Estudiante
Objetivo	Descargar el material de revisión y estudio
Detalle	Los estudiantes podrán visualizar y descargar los recursos pedagógicos para su revisión. Los contenidos se presentaran el base al estilo de aprendizaje que posee el estudiante
Secuencia	<ul style="list-style-type: none"> • Visualización de los contenidos • Descarga de los contenidos

CASO DE USO: CU-017

Tabla 27: Caso de Uso CU-017

Código	CU-017
Descripción	Visualización y desarrollo de los ejercicios propuestos
Actor	Estudiante
Objetivo	Desarrollar el ejercicio propuesto por el tutor
Detalle	Los estudiantes podrán visualizar y desarrollar el ejercicio propuesto.
Secuencia	<ul style="list-style-type: none"> • Visualización del ejercicio • Desarrollo del ejercicio

CASO DE USO: CU-018

Tabla 28: Caso de Uso CU-018

Código	CU-018
Descripción	Visualización del progreso del Aprendizaje
Actor	Estudiante
Objetivo	Visualización del progreso de aprendizaje de los contenidos de la materia
Detalle	Los estudiantes podrán visualizar el estado del aprendizaje
Secuencia	<ul style="list-style-type: none">• Visualizar el progreso del aprendizaje

CASO DE USO: CU-019

Tabla 29: Caso de Uso CU-019

Código	CU-019
Descripción	Visualización las respuestas de los cuestionarios
Actor	Estudiante
Objetivo	Visualización de las respuestas de los cuestionarios
Detalle	Los estudiantes podrán visualizar las respuestas de los cuestionarios
Secuencia	<ul style="list-style-type: none">• Visualizar las respuestas de los cuestionarios

4.3.5 CASOS DE USO DEL SISTEMA

CASO DE USO: CU-020

Tabla 30: Caso de Uso CU-020

Código	CU-020
Descripción	Control de Revisión y Desarrollo de Ejercicios
Actor	Sistema
Objetivo	Controlar mediante los parámetros de revisión y desarrollo de Ejercicios configurados por el tutor

Detalle	El sistema verifica que el estudiante reviso los recursos y los ejercicios desarrollados por el estudiante y si cumple le permite rendir las evaluaciones
Secuencia	<ul style="list-style-type: none"> • Revisión de recursos • Revisión de desarrollo de ejercicios • Presenta la evaluación a ser realizada por el estudiante

CASO DE USO: CU-021

Tabla 31: Caso de Uso CU-021

Código	CU-021
Descripción	Actualizar el perfil del estudiante
Actor	Sistema
Objetivo	Actualizar el nivel de aprendizaje del estudiante en base al aprendizaje de las materias
Detalle	Luego que el estudiante realice las evaluaciones el estudiante se le actualiza el nivel
Secuencia	<ul style="list-style-type: none"> • Evaluaciones • Actualizar el perfil

CASO DE USO: CU-022

Tabla 32: Caso de Uso CU-022

Código	CU-022
Descripción	Modelo Matemático
Actor	Sistema
Objetivo	Actualizar el nivel de aprendizaje del estudiante en base a los parámetros de revisión de recursos y evaluaciones
Detalle	Luego que el estudiante realice las evaluaciones se ejecuta el modelo matemático y actualiza el nivel
Secuencia	<ul style="list-style-type: none"> • Evaluaciones

- Ejecuta modelo matemático

4.4 MODELO ENTIDAD RELACION

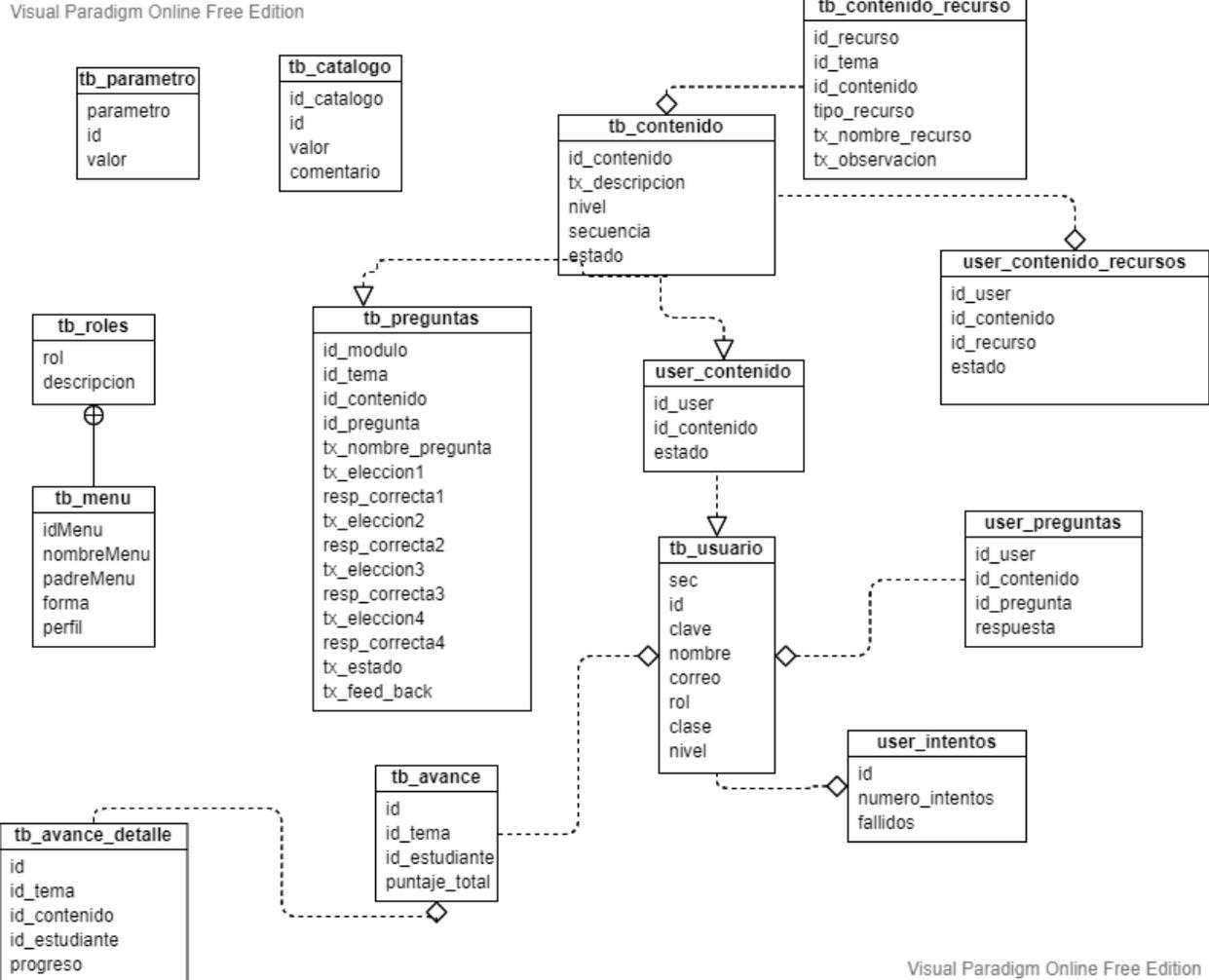


Figura 15: Modelo Entidad Relación de la Base de Datos, desarrollado por el autor

4.5 DICCIONARIO DE DATOS

TB_TEMA: Información de los usuarios del sistema

Tabla 33: Información de los usuarios del sistema

Campo	Descripción
Id_unidad	Código de la Unidad
Id_tema	Código del tema

No_tema	Nombre del tema
Estado	Estado del tema

TB_UNIDAD: Información de los usuarios del sistema

Tabla 34: Información de los usuarios del sistema

Campo	Descripción
Id_unidad	Código de la Unidad
No_unidad	Nombre de la Unidad

TB_USUARIO: Información de los usuarios del sistema

Tabla 35: Información de los usuarios del sistema

Campo	Descripción
Sec	Numero secuencial auto incremental
Id	Código de Usuario
Clave	Contraseña del usuario
Nombre	Nombre del usuario
Correo	Correo del usuario
Rol	Perfil del usuario : Estudiante – Tutor – Administrador
Clase	Código de la clase al que pertenece el usuario
Nivel	Nivel Básico – Intermedio - Avanzado

TB_ROL: Rol del usuario en el sistema

Tabla 36: Rol del usuario en el sistema

Campo	Descripción
Rol	Código del rol
Descripción	Descripción del rol del usuario : Estudiante – Tutor - Administrador

TB_MENU: Menú dinámico de las opciones del sistema

Tabla 37: Menú dinámico de las opciones del sistema

Campo	Descripción
Id	Código de menú
Descripción	Descripción del menú
Padre_menu	Identificador del código del padre del menú
Forma	Pantalla de la opción correspondiente
Rol	Perfil del usuario : Estudiante – Tutor – Administrador

TB_PARAM_CONTENIDO: Información sobre los parámetros sobre el avance en los contenidos

Tabla 38: Información sobre los parámetros sobre el avance en los contenidos

Campo	Descripción
Id_unidad	Código de la unidad
Id_Tema	Identificador del tema
Id_param	Código del parámetro
Identificador	Identificador del parámetro
Valor	Valor del parámetro
Txt_descripcion	Descripción del parámetro

TB_PARAMETRO: Parámetros del sistema

Tabla 39: Parámetros del sistema

Campo	Descripción
Parámetro	Código del parámetro
Id	Identificador del parámetro
Valor	Valor del parámetro

TB_CATALOGO: Información de los catálogos del sistema

Tabla 40: Información de los catálogos del sistema

Campo	Descripción
--------------	--------------------

IdCatalogo	Código de la tabla de catálogos
Id	Identificador de la tabla de catálogos
Valor	Valor de la tabla de catalogo
Comentario	Comentario sobre la información del catalogo

TB_CONTENIDO: Información sobre los temas de la materia

Tabla 41: Información sobre los temas de la materia

Campo	Descripción
Id	numero secuencial auto-incremental
Id_Unidad	Código de la Unidad
IdTema	Identificador del tema
Contenido	Identificador del contenido
Ponderación	Calificación del tema
Secuencia	Orden de los temas
Estado	Estado del tema

TB_CONTENIDO_RECURSO: Recursos relacionados con el contenido

Tabla 42: Recursos relacionados con el contenido

Campo	Descripción
Id	numero secuencial auto-incremental
Id_Unidad	Código de la Unidad
IdTema	Identificador del tema
idContenido	Identificador del contenido
idRecurso	Identificador del recurso
NombreRecurso	Nombre del recurso
Observacion	Observación del recurso

TB_AVANCE: Información sobre el progreso del aprendizaje del estudiante

Tabla 43: Información sobre el progreso del aprendizaje del estudiante

Campo	Descripción
Id	numero secuencial auto-incremental
Id_Unidad	Código de la Unidad
IdTema	Identificador del tema
idEstudiante	Identificador del estudiante
PuntajeTotal	Valor del puntaje alcanzado por el estudiante

TB_AVANCE_DETALLE: Información sobre el progreso del aprendizaje del estudiante por contenido

Tabla 44: Información sobre el progreso del aprendizaje del estudiante por contenido

Campo	Descripción
Id	numero secuencial auto-incremental
Id_Unidad	Código de la Unidad
IdTema	Identificador del tema
idContenido	Identificador del contenido
idEstudiante	Identificador del estudiante
PuntajeTotal	Valor del puntaje alcanzado por el estudiante

TB_PREGUNTAS: Información sobre los temas de la materia

Tabla 45: Información sobre los temas de la materia

Campo	Descripción
Id_Unidad	Código de la Unidad
IdTema	Identificador del tema
idContenido	Identificador del contenido
idPregunta	Identificador de la pregunta
nombrePregunta	Nombre de la pregunta

Id_recurso	Código del recurso que debe presentarse si responde incorrectamente
EnunciadoPregunta	Descripción de la pregunta
Ponderacion	Puntaje de la pregunta
Eleccion1	Alternativa de elección de la respuesta
RespuestaCorrecta	Es o no es la respuesta correcta a la pregunta
Eleccion2	Alternativa de elección de la respuesta 2
RespuestaCorrecta2	Es o no es la respuesta correcta a la pregunta 2
Eleccion2	Alternativa de elección de la respuesta 3
RespuestaCorrecta3	Es o no es la respuesta correcta a la pregunta 3
Eleccion4	Alternativa de elección de la respuesta 4
RespuestaCorrecta4	Es o no es la respuesta correcta a la pregunta 4
Estado	Estado del registro Ingresado / Anulado
Feedback	Retro-alimentación de las respuestas

USER_CONTENIDO: Información sobre los contenidos aprendidos por los usuarios

Tabla 46: Información sobre los contenidos aprendidos por los usuarios

Campo	Descripción
Id_Unidad	Código de la Unidad
IdTema	Código del tema
usuario	Usuario
Contenido	Identificador del contenido
Estado	Estado del tema

USER_CONTENIDO_RECURSOS: Información sobre los recursos revisados por los usuarios

Tabla 47: Información sobre los recursos revisados por los usuarios

Campo	Descripción
Usuario	Usuario

Unidad	Código de la Unidad
Tema	Código del tema
Contenido	Identificador del contenido
Recurso	Recurso
Estado	Estado del tema

USER_PREGUNTAS: Información sobre las preguntas realizadas por el estudiante

Tabla 48: Información sobre las preguntas realizadas por el estudiante

Campo	Descripción
Usuario	Usuario
Unidad	Código de la Unidad
Tema	Código del tema
Contenido	Identificador del contenido
Pregunta	Código de la pregunta
Respuesta	Información de la respuesta realizada
Estado	Estado del tema

USER_INTENTOS: Información sobre los intentos en realizar los cuestionarios

Tabla 49: Información sobre los intentos en realizar los cuestionarios

Campo	Descripción
Usuario	Usuario
Intento	Numero de Intentos

4.6 REFINAMIENTO DEL PROTOTIPO

Para el diseño del prototipo de; sistema tutor inteligente se dividieron las opciones en base a los perfiles de los usuarios. El desarrollo se lo realizo siguiendo la metodología del prototipado donde se presentaron los avances de lo desarrollado por fases.

Luego de la identificación de las opciones se seleccionaron aquellas que permitían probar el flujo del aprendizaje de los estudiantes.

A continuación, el detalle de las opciones desarrolladas para la implementación del prototipo del Sistema Tutor Inteligente.

Tabla 50: Implementaciones del prototipo

Perfil	Opción
Tutor	Base de Conocimiento
	Agregar Recursos
	Seguimiento
Administrador	Carga Masiva de Usuarios
	Registro de usuario
	Parámetros del Sistema
	Cargar Banco de Preguntas
Estudiante	Visualizar progreso
	Visualizar Resumen Preguntas
	Diagnosticar
	Descargar Recursos
	Panel de Avance
Interface	Cambiar Clave
	Menú Dinámico
	Algoritmo Matemático - Actualiza Perfil
	Iniciar Sesión

4.9.1 PRUEBAS DE PROTOTIPADO

Para efectuar el proceso de prototipado se realizaron distintas versiones las cuales fueron presentadas y evaluadas por el experto, cada cambio y/o mejora que se deseó realizar fue analizada, con conclusiones, observaciones dadas por el experto hasta llegar a la versión final. Siendo esta una variante de sistema tutor inteligente, sistema adaptativo de evaluación de conocimiento.

A continuación, el detalle de las versiones presentadas al tutor para su revisión:

4.9.1.1 PROTOTIPO VERSIÓN 1.0

- Ingreso de Preguntas y las opciones de respuestas
- Registro de Usuarios al Sistema
- No existía una estandarización en las pantallas
- Base de Datos SQL Server

4.9.1.2 PROTOTIPO VERSIÓN 2.0

- Carga masiva de Preguntas y las opciones de respuestas
- Carga Masiva de Usuarios al Sistema
- Cambio de diseño en las pantallas del estudiante
- Base de Datos Mysql
- Implementación del prototipo en la Nube
- Control de la realización de las evaluaciones
- Presentar las preguntas en forma aleatoria al momento de realizar las evaluaciones
- Desarrollo del modelo matemático que controle el aprendizaje del estudiante

4.9.1.3 PROTOTIPO VERSIÓN 3.0

1. Cambio en la estructura de la organización del conocimiento (Unidad – Tema – Contenido – Ejercicios)
2. Estandarización de todas las pantallas del sistema Tutor Inteligente
3. Controlar el Avance del aprendizaje en base a los ejercicios resueltos
4. Asociar el código de los recursos para la retroalimentación de las respuestas incorrectas.
5. Mejorar la seguridad de las pantallas que conforman el Sistema
6. Semaforización de los contenidos aprendidos y por aprender
7. Semaforización de los recursos revisados y por revisar

4.10 INTERFAZ GRAFICA

A continuación, se detalla la interfaz gráfica para la operación de los diferentes usuarios que interactúan en el prototipo del sistema tutor inteligente STI.

Los perfiles de usuario que interactúan con el STI son:

1. Administrador
2. Tutor
3. Estudiante

Así mismo el sistema cuenta con demás opciones para realizar las transacciones utilitarias para su operación, a continuación, el detalle:

1. Cambiar Clave
2. Carga Masiva de Estudiantes
3. Registro de Usuario
4. Menú
5. Parámetros del Sistema

4.10.1 LOGIN, CAMBIO DE CONTRASEÑA Y MODELO DE MENU DE OPCIONES

4.10.1.1 INICIO DE SESION

Por medio de esta opción permite acceder al prototipo. El usuario deberá ingresar el usuario y la clave asignada. Los usuarios están cargados previamente por medio de la carga masiva realizador por el administrador del sistema

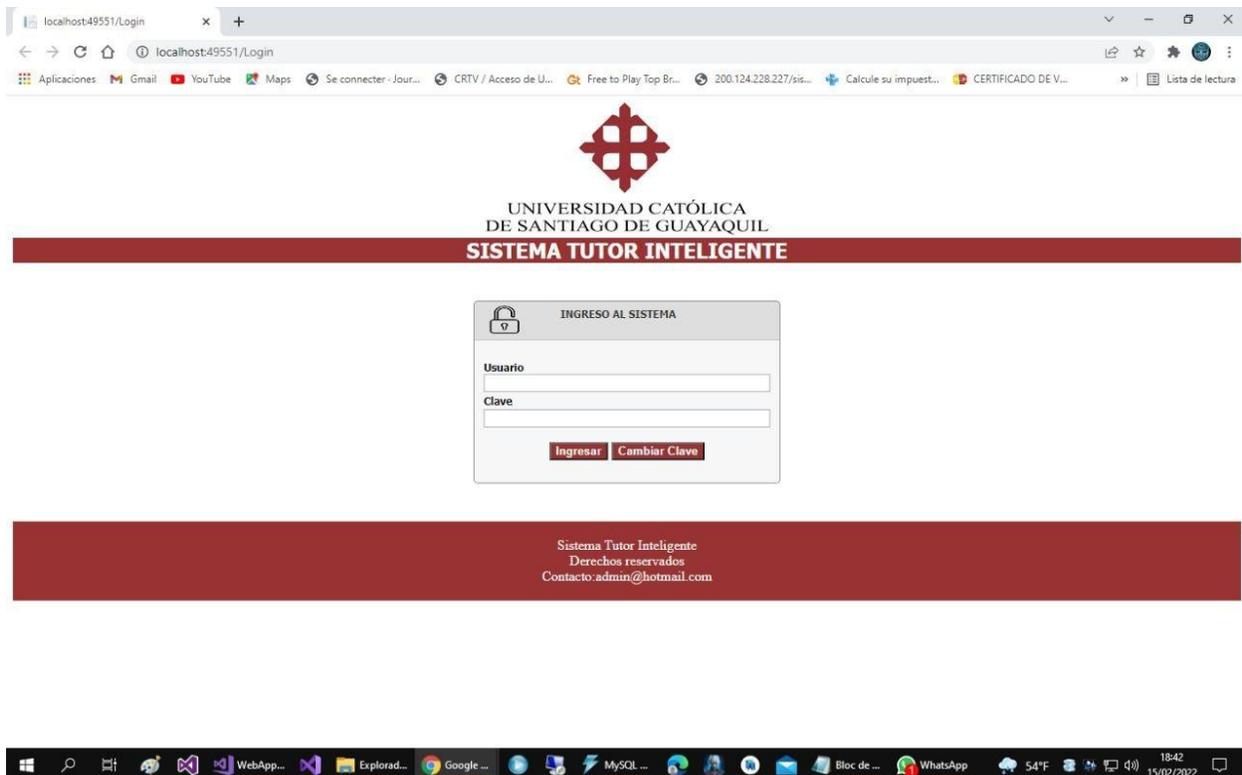


Figura 16: Login del prototipo sistema tutor inteligente

4.10.1.2 MENU DE OPCIONES

Por medio de esta pantalla los usuarios que tienen el rol de administrador y de tutor podrán acceder a las opciones que tienen asignadas para poder interactuar con el sistema tutor inteligente. El menú es dinámico es decir que le presentara solo las opciones que corresponden de acuerdo al perfil del usuario.

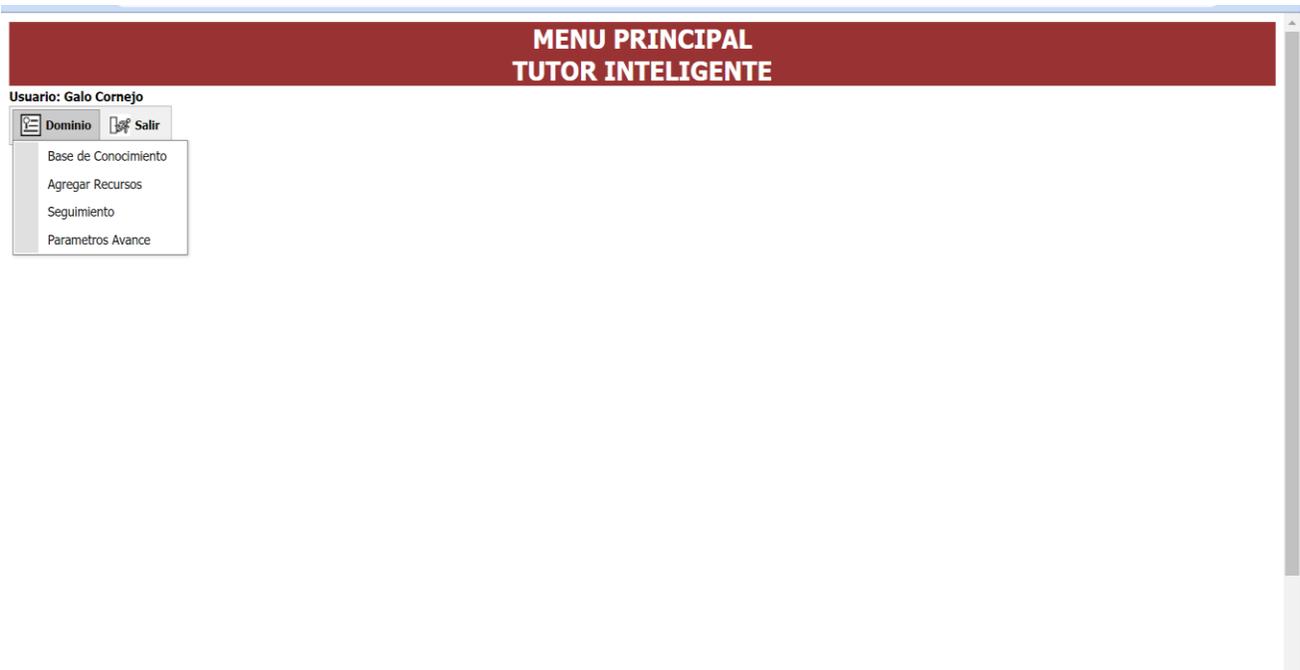


Figura 17: Menú de las opciones

4.10.1.3 CAMBIAR DE CLAVE

Por medio de esta opción el usuario debe cambiar la contraseña con el fin de mejorar la seguridad en el acceso al sistema.

Debe ingresar la contraseña actual y una nueva contraseña

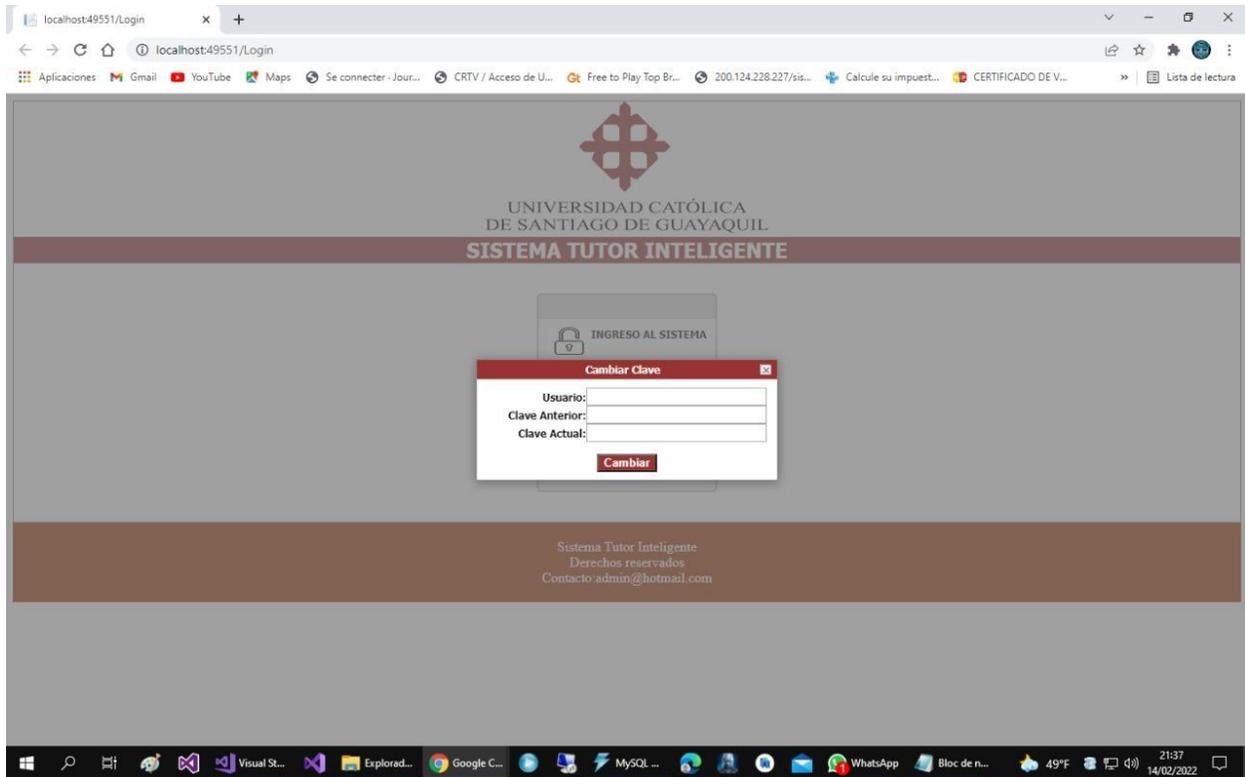


Figura 18: Interfaz de cambio de clave

4.10.2 ROL ADMINISTRADOR

4.10.2.1 CARGAR BANCO DE PREGUNTAS

El aprendizaje de la materia de Lenguaje de programación consta de varios contenidos, estos deben ser ingresados por el Tutor. Para el correcto aprendizaje de la materia los contenidos deben ser configurados secuencialmente y siguiendo el flujo correspondiente.

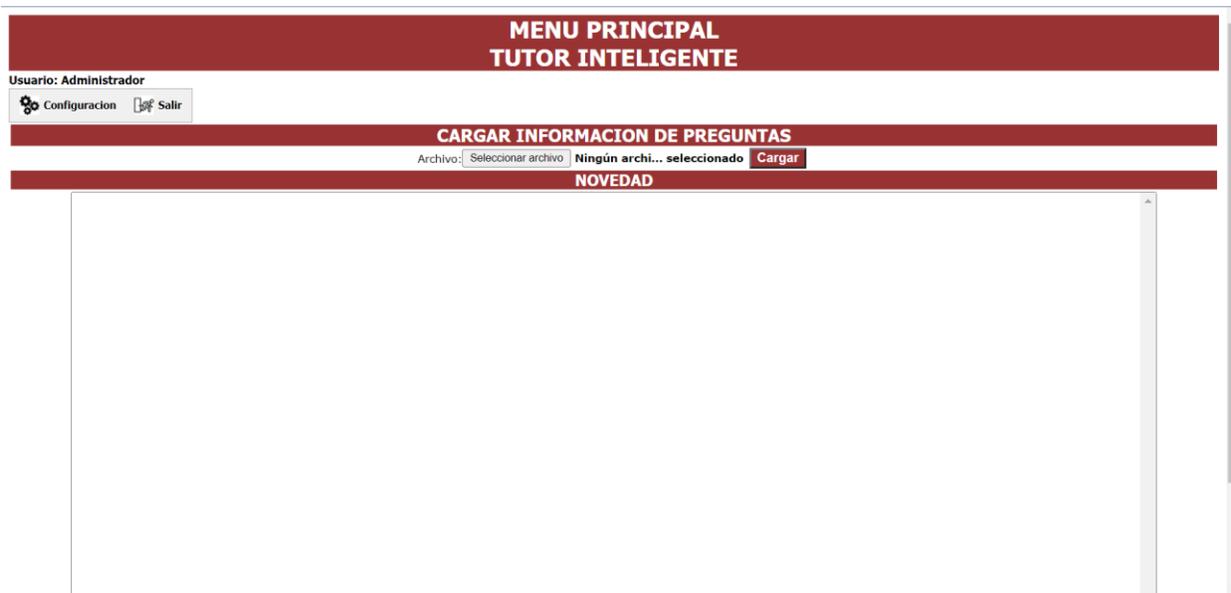


Figura 19: Carga de banco de preguntas

4.10.2.2 REGISTRO DE USUARIOS

Mediante esta opción el administrador puede ingresar a los usuarios que van a utilizar el sistema tutor inteligente



Figura 20: Interfaz de Registro de Usuarios

4.10.2.3 PARAMETROS DEL SISTEMA

Mediante esta opción el Administrador puede configurar los parámetros que utiliza el sistema para operar el sistema.

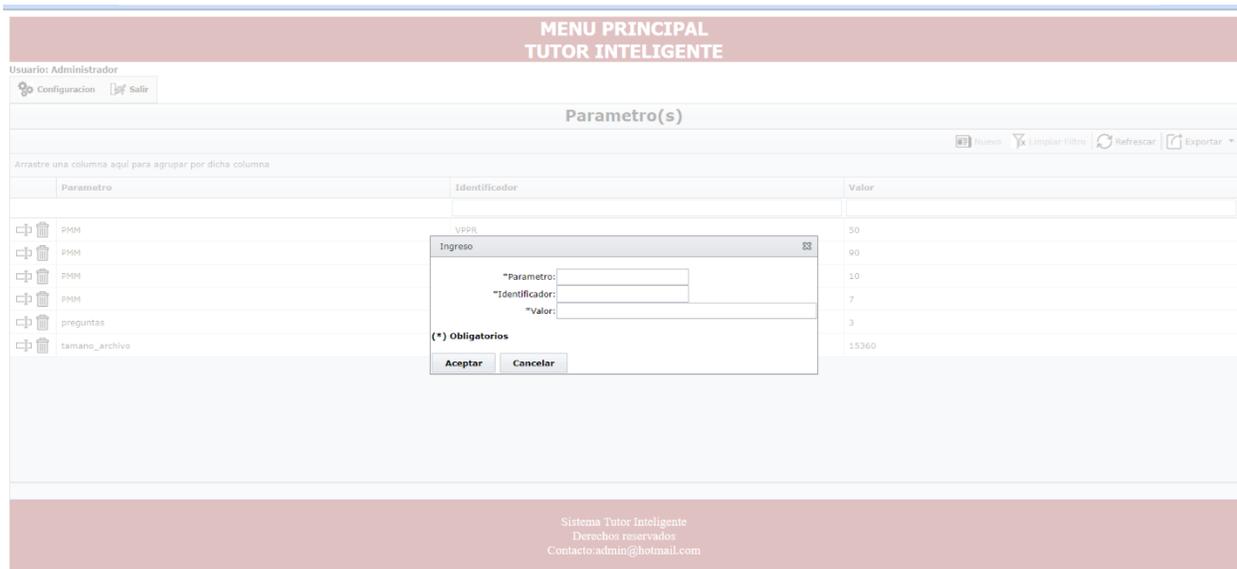


Figura 21: Interfaz de Parámetros del Sistema

4.10.2.4 UNIDAD

Opción mediante el cual el Administrador del sistema puede dar mantenimiento a las unidades que conforman la base de conocimientos.

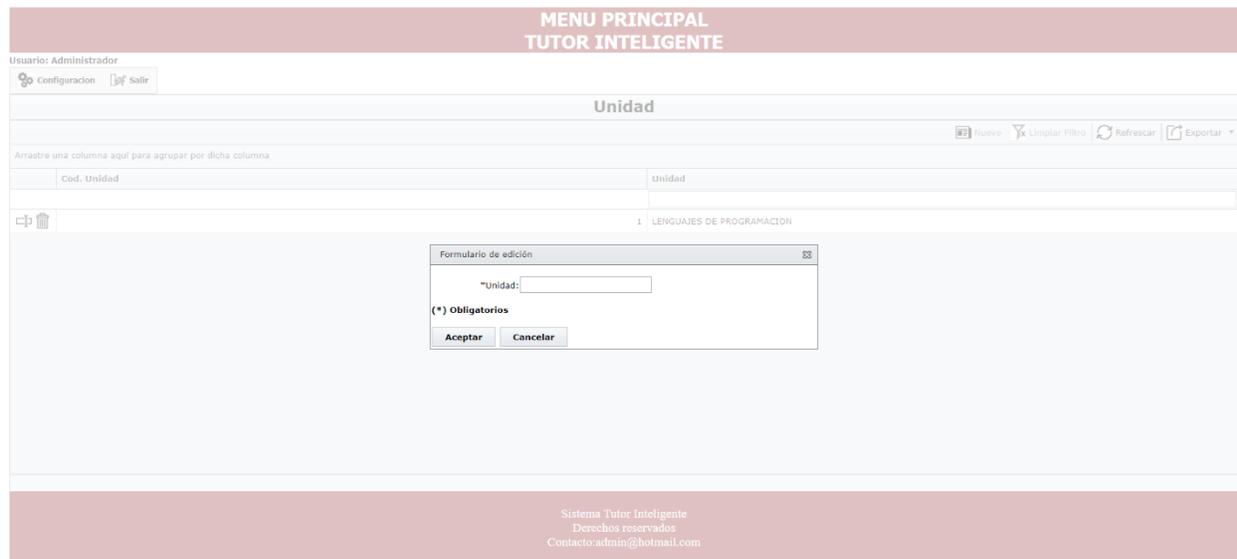


Figura 22: Interfaz de Ingreso de Unidad

4.10.2.5 TEMA

Opción mediante el cual el Administrador del sistema puede dar mantenimiento a los temas que conforman la base de conocimientos.

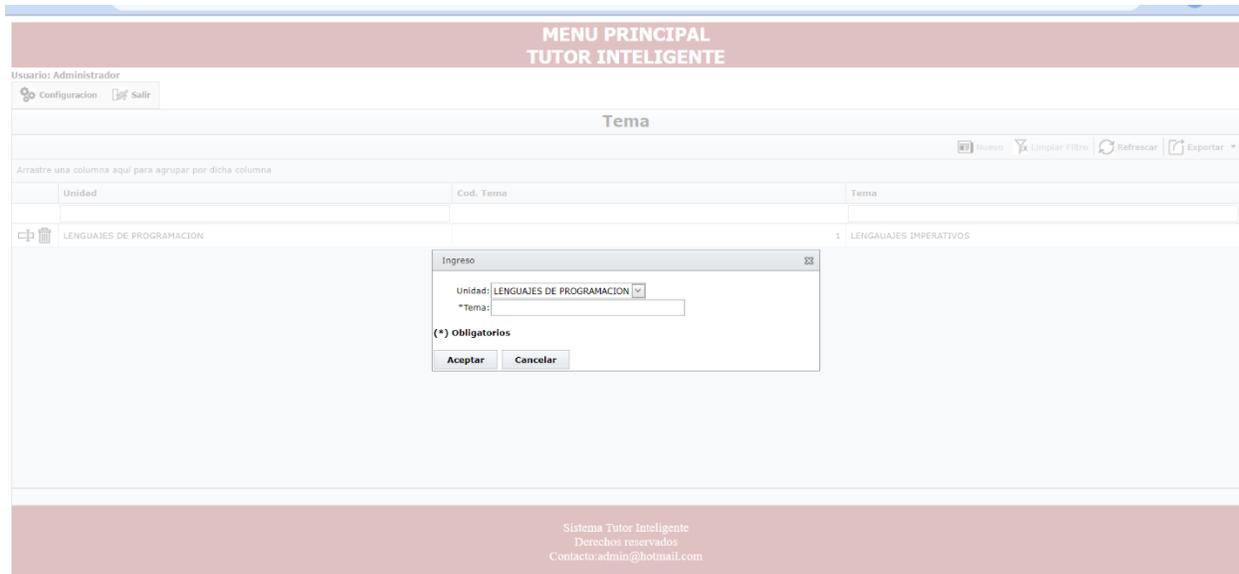


Figura 23: Interfaz de Ingreso de Tema

4.10.2.6 CARGA MASIVA DE USUARIOS

Para la operación del sistema deben encontrarse registrados los usuarios, esta información es realizada por parte del Administrador del Sistema. La información se encuentra con un formato establecido y en Excel.

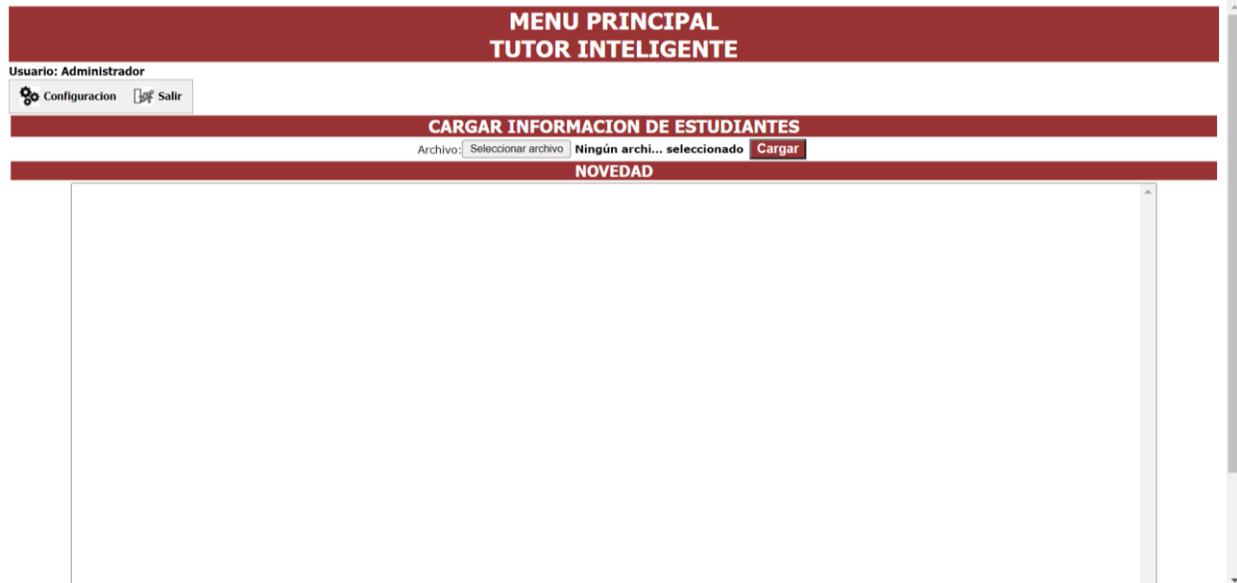


Figura 24: Carga Masiva de usuarios

4.10.3 ROL ESTUDIANTE

4.10.3.1 PANEL DE APRENDIZAJE DEL ESTUDIANTE

Esta pantalla presenta el avance del aprendizaje del estudiante. El avance del aprendizaje corresponde a los contenidos aprendidos y los recursos revisados. El flujo del aprendizaje está dividido en tres niveles: Básico – Intermedio y Avanzado. El avance del aprendizaje depende los contenidos y los cuestionarios que el estudiante revise. Adicional a esto se integró la revisión y la respectiva descarga de los recursos, por medio de esta opción el estudiante descarga los recursos de acuerdo a el contenido. Estos recursos pueden ser imágenes, documentos en PDF, documentos de Word, videos y demás.

Luego de realizar las evaluaciones el sistema tutor asignará los recursos que debe revisar para poder revisar y luego de su revisión podrá realizar nuevamente las evaluaciones y de esta forma poder avanzar con su nivel de aprendizaje.

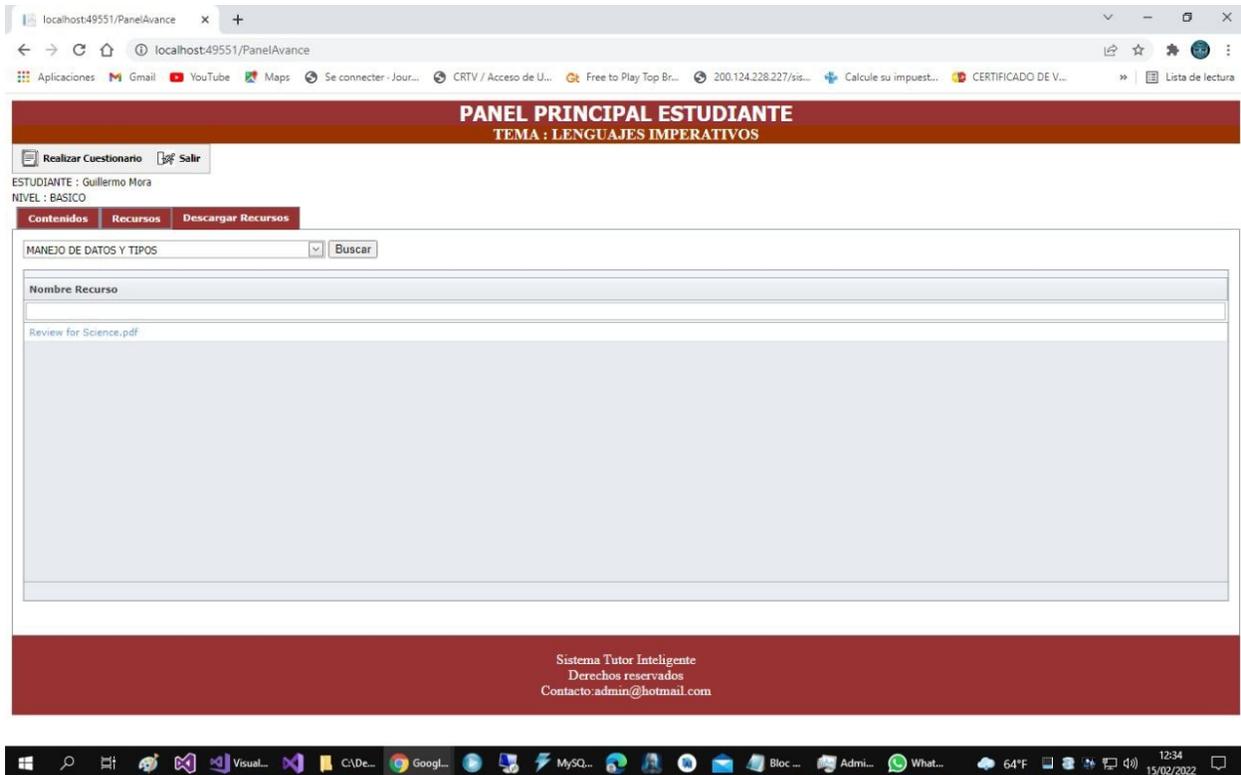


Figura 25: Panel de aprendizaje del estudiante sin recursos

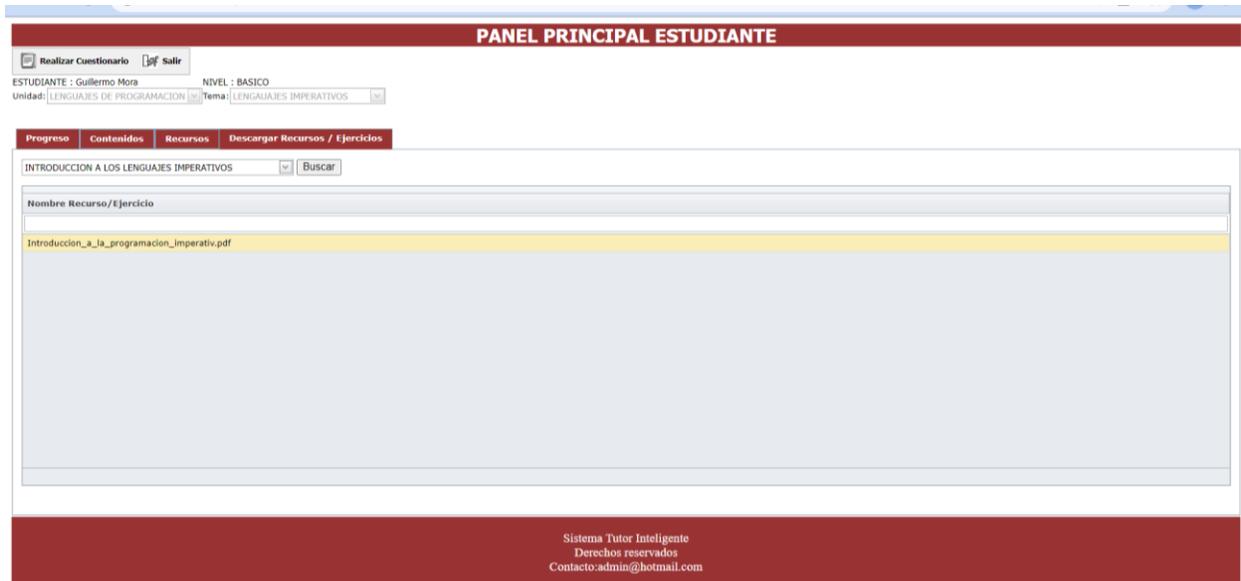


Figura 26: Panel de aprendizaje del estudiante con recursos

4.10.3.2 RESUMEN DE RESPUESTAS

Esta pantalla presenta el resumen de las respuestas realizadas por parte de los estudiantes, El sistema le indicara los contenidos a mejorar y los recursos a revisar en aquellos temas en los que el estudiante tenga respuestas incorrectas.

The screenshot displays a web browser window with the URL localhost:49551/ResumenRespuestas. The page title is 'TUTOR INTELIGENTE - EVALUACION' and the subject is 'TEMA : LENGUAJES IMPERATIVOS'. A 'Salir' button is located at the top left. The main content is a table titled 'RESUMEN RESPUESTA(S)' with a dropdown menu for 'Intentos' set to 1. The table has four columns: 'Pregunta No.', 'Descripción', 'Correcta', and 'Retroalimentacion'. The data for the first attempt is as follows:

Pregunta No.	Descripción	Correcta	Retroalimentacion
2	P1	N	F1
3	P2	N	F2
7	P6	N	F65
8	P7	N	F75
9	P8	N	F85

Below the table, there are expandable sections for 'Intentos: 2' through 'Intentos: 8'. The footer contains the text: 'Sistema Tutor Inteligente', 'Derechos reservados', and 'Contacto: admin@hotmail.com'.

Figura 27: Resumen de las respuestas de la prueba

4.10.3.3 EVALUACION DEL ESTUDIANTE

En esta acción el estudiante realizará la evaluación para probar los conocimientos adquiridos a través de preguntas y respuestas objetivas las cuales de ser contestadas para aprobar los respectivos niveles ya sean básico, intermedio y avanzado, en caso de responder incorrectamente se mostrará la pantalla del feedback con cada una de las retroalimentaciones respectivas.

TUTOR INTELIGENTE - EVALUACION
Tema

PREGUNTA

Que es un lenguaje Imperativo

- Es un lenguaje declarativo donde el programador especifica lo que quiere hacer, en lugar de lidiar con el estado de los objetos.
- Son las instrucciones se ejecutan unas tras otras, de manera secuencial, salvo cuando se encuentran estructuras de control condicionales o bucles.
- Es un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos.
- Es una secuencia es un dato abstracto que representa una secuencia ordenada de valores, donde el mismo valor puede ocurrir más de una vez.

[Siguiente](#)

Sistema Tutor Inteligente
Derechos reservados
Contacto:admin@hotmail.com

Figura 28: Evaluación del estudiante

4.10.3.4 REVISION Y DESCARGA DE RECURSOS

Por medio de esta opción el estudiante descarga los recursos de acuerdo a el contenido. Estos recursos pueden ser imágenes, documentos en PDF, documentos de Word, Videos y demás.

Luego de realizar las evaluaciones el sistema tutor asignará los recursos que debe revisar para poder revisar y luego de su revisión podrá realizar nuevamente las evaluaciones y de esta forma poder avanzar con su nivel de aprendizaje.



Figura 29: Interfaz de revisión y descarga de los recursos pedagógicos

4.10.4 ROL DEL TUTOR

4.10.4.1 BASE DE CONOCIMIENTOS

Mediante esta opción el tutor puede ingresar los contenidos que está conformado la materia de Lenguajes de Programación en Lenguajes Imperativos.



Figura 30: Panel de Ingreso de la Base de Conocimientos

4.10.4.2 AGREGAR RECURSOS

Mediante esta opción el tutor puede agregar los recursos correspondientes a cada uno de los contenidos que posee los Lenguajes Imperativos.

The screenshot shows a web interface with a dark red header containing the text 'MENU PRINCIPAL TUTOR INTELIGENTE'. Below the header, the user is identified as 'Usuario: Galo Cornejo'. There are two buttons: 'Dominio' and 'Salir'. The main content area is titled 'ASIGNACION RECURSOS PEDAGOGICOS (INGRESO)'. It contains several dropdown menus: 'Unidad: LENGUAJES DE PROGRAMACION', 'Tema: LENGUAJES IMPERATIVOS', 'Contenidos: INTRODUCCION A LOS LENGUAJES', and 'Recursos: imagen'. Below these is an 'Archivo:' field with a 'Seleccionar archivo' button and the text 'Ningún archivo seleccionado'. There is an 'Observaciones:' text area. At the bottom of the form are two buttons: 'Regresar' and 'Guardar'. A footer bar at the bottom contains the text: 'Sistema Tutor Inteligente', 'Derechos reservados', and 'Contacto:admin@hotmail.com'.

Figura 31: Panel de Ingreso de los Recursos

4.10.4.3 SEGUIMIENTO

Mediante esta opción el tutor puede visualizar los avances del aprendizaje que están realizando los estudiantes en el aprendizaje de los lenguajes imperativos.

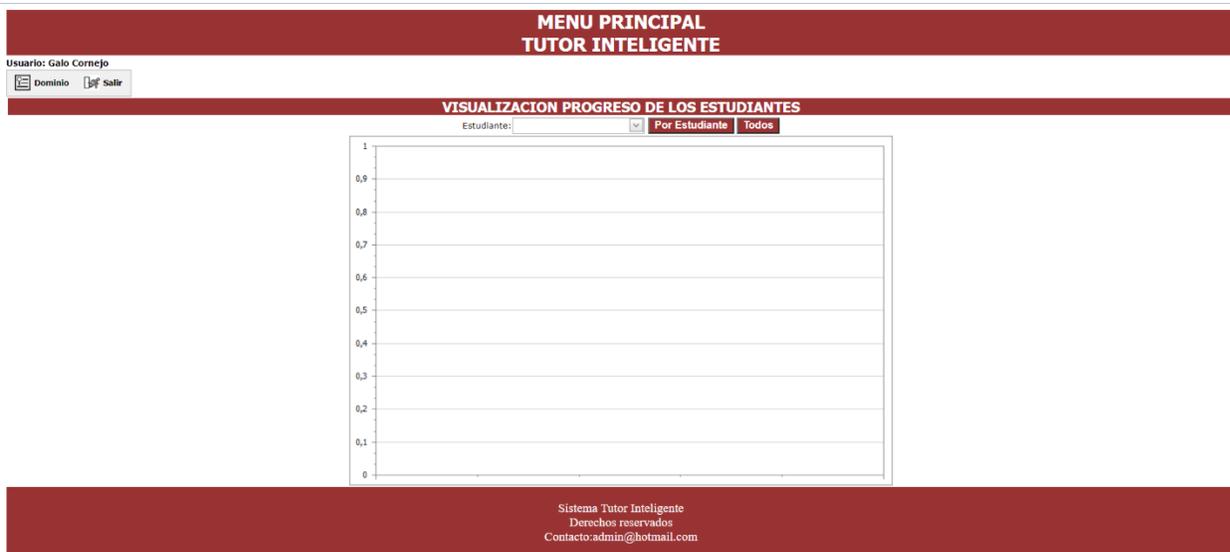


Figura 32: Panel de Seguimiento del Estudiante

4.10.4.4 PARAMETROS DE AVANCE

En esta opción se puede ver cómo se pueden ingresar los datos parametrizables para darle un valor al avance que el estudiante pueda tener.



Figura 33: Panel de Parametrización de los avances

4.11 DESCRIPCION FUNCIONAL DEL PROTOTIPO

El prototipo del sistema tutor inteligente tiene como fin complementar el aprendizaje de la materia de lenguaje de programación.

El aprendizaje de la materia se basa en la revisión de los recursos que deben ser revisados por parte del estudiante y además las distintas evaluaciones para medir el nivel de conocimientos que va adquiriendo el estudiante.

El sistema consta de tres roles Estudiante – Tutor – Administrador, mediante el rol el sistema habilita las opciones y funcionalidades para su operación y flujo de los datos.

La funcionalidad principal del Sistema Tutor Inteligente es la interacción del sistema con el Estudiante. El aprendizaje de la materia se basa en la revisión de los recursos y desarrollo de ejercicios desde el nivel Básico al Nivel Avanzado. El avance del aprendizaje por parte del estudiante es mediante la ejecución de las distintas evaluaciones que se le van presentando para medir el nivel de conocimientos en los diferentes niveles.

El Administrador del sistema previo a la interacción del estudiante con el sistema debe cargar las preguntas que servirán para evaluar el conocimiento adquirido. Así mismo configurar los parámetros del sistema que sirvan para la operación del sistema.

Posteriormente el tutor deberá ingresar los contenidos de la materia y asociar los recursos correspondientes con el fin de que los estudiantes puedan revisarlos.

El estudiante deberá revisar los recursos compartidos por el tutor y realizar los ejercicios propuestos. Posterior a la revisión podrá realizar las evaluaciones y poder avanzar en los niveles de aprendizaje.

Para habilitar las evaluaciones el sistema valida los porcentajes de revisión y desarrollo de ejercicios.

Luego de realizar las evaluaciones el sistema presenta un resumen de las respuestas con su respectiva retroalimentación.

A continuación, el detalle de las opciones que consta el Sistema con su funcionalidad resumida en una tabla.

Tabla 51: Descripción Funcional de software con cada una de sus características

Perfil	Opción	Descripción funcional
Tutor	Base de Conocimiento	Está organizado por Contenidos estos deben ser revisados y aprendidos por parte del estudiante. Estos contenidos están basados al Syllabus de la materia de Lenguaje de Programación. Los contenidos deben ser registrados de manera secuencial para seguir el flujo del aprendizaje y deben ser ingresados por nivel (Básico - Intermedio y Avanzado)
	Agregar Recursos	El tutor debe subir los recursos que los estudiantes deban revisar para obtener los conocimientos necesarios para rendir las evaluaciones correspondientes. Estos recursos deben estar relacionados por el correspondiente

		contenido. Los recursos pueden ser documentos de Word, Presentaciones, documentos PDF y demás
	Seguimiento	El sistema presentara los avances del aprendizaje del estudiante. El avance del aprendizaje depende de los contenidos y recursos que serán revisados y evaluados por medio de los cuestionarios que debe aprobar. Se presenta de manera gráfica. El tutor podrá escoger el estudiante específico o consultar de todos los estudiantes que estén registrados en el sistema
Administrador	Carga Masiva de Usuarios	La carga de los usuarios es mediante el proceso de subir la información de un archivo en Excel. Los usuarios luego que estén cargados en el sistema podrán obtener los conocimientos del contenido de la materia de lenguaje de programación
	Registro de usuario	El registro se lo realizara a estudiantes específicos que no se encontraban en el archivo Excel. Los estudiantes iniciaran en el nivel básico y a medida que revisen y aprueben los contenidos avanzaran de nivel
	Parámetros del Sistema	Información necesaria para el funcionamiento del sistema. La operación del sistema dependerá de los valores registrados en los parámetros iniciales
	Cargar Banco de Preguntas	Opción donde se cargarán las preguntas que se presentaran para las evaluaciones que deben realizar los estudiantes.
Estudiante	Visualizar progreso	Mediante esta opción el estudiante podrá visualizar los contenidos aprendidos y cuáles deben ser aprobados para completar el conocimiento de la materia
	Visualizar Resumen Preguntas	El estudiante podrá visualizar el resultado de las respuestas realizadas en los cuestionarios y los recursos que deben ser revisados para mejorar sus conocimientos

	Diagnosticar	En esta opción el sistema le presenta las preguntas que deben ser contestadas y poder verificar si obtuvo el puntaje necesario para aprobar el nivel
	Descargar Recursos	Mediante esta opción el estudiante podrá descargar los recursos que corresponden a los contenidos y de esta forma revisarlos para rendir las evaluaciones y avanzar en los niveles
	Panel de Avance	Opción mediante el cual el estudiante podrá visualizar el avance en su aprendizaje. Así mismo le permitirá realizar las evaluaciones de los contenidos que contiene la materia y poder descargar los recursos para su revisión
Interface	Cambiar Clave	Esta opción permite la modificación de la clave para el acceso al sistema. El cambio se debe realizar para mejorar la seguridad del sistema
	Menú Dinámico	El menú presentara las opciones correspondientes al perfil de los usuarios. Los perfiles contemplados en el sistema son : Estudiante - Tutor y Administrador .
	Algoritmo Matemático Actualiza Perfil	El algoritmo servirá para evaluar si el estudiante posee los conocimientos necesarios para avanzar de nivel. El modelo matemático está basado en los recursos revisados y en las evaluaciones realizados por parte del estudiante. El conocimiento total de la materia está basado en completar los niveles. VER ANEXO 3.
	Iniciar Sesión	Para el ingreso al sistema el usuario deberá ingresar sus credenciales entregados por parte del administrador.

CONCLUSIONES

La presente tesis fue desarrollada con el fin de servir de herramienta complementaria para el proceso de enseñanza de la materia de Lenguaje de Programación enfocándose en los lenguajes imperativos.

En el desarrollo de la tesis se identificó como funciona el proceso de tutoría realizado por los docentes y el mismo fue replicado para la implementación del Sistema Tutor Inteligente STI.

Luego del proceso de enseñanza realizado por el docente el estudiante podría complementar el aprendizaje utilizando el STI mediante la revisión de los contenidos y la elaboración de evaluaciones. En el prototipo se clasifican los contenidos por niveles esta división permite el avance del aprendizaje de manera eficiente garantizando un mejor entendimiento de la materia.

Los contenidos están basados en el Syllabus de la materia y los recursos compartidos están relacionados a los contenidos y a los niveles de aprendizaje.

El proceso de aprendizaje depende de los recursos cargados en el sistema, estos servirán para que el estudiante pueda descargarlos para su revisión y preparación para demostrar en las evaluaciones disponibles en el sistema.

El prototipo construido está orientado a una variante de sistema tutor inteligente, siendo como tal un sistema adaptativo de evaluación de conocimiento contribuye al componente trabajo autónomo del estudiante, este podrá mejorar sus conocimientos a medida que progresa con cada elemento implementado.

El prototipo está desarrollado en un ambiente WEB donde el estudiante podrá utilizarlo desde cualquier lugar y lo más importante en cualquier momento dependiendo del tiempo que tenga disponible.

El prototipo permite visualizar el progreso del aprendizaje de los estudiantes donde muestra los contenidos aprendidos. Este proceso lo puede visualizar tanto el estudiante como el docente. En el caso del docente podrá analizar donde los estudiantes tienen

complicaciones y para solucionar debe compartir más recursos que apoyen a mejorar sus conocimientos.

En las evaluaciones realizadas por los estudiantes el sistema le permitirá mostrar el resultado de las respuestas y mediante la utilización de un modelo matemático el sistema evaluará si el estudiante puede avanzar al siguiente nivel.

Luego de su implementación y utilización del sistema se deben analizar los datos registrados para la toma de decisiones con el fin del mejoramiento del aprendizaje por parte de docentes y autoridades.

RECOMENDACIONES

El prototipo de sistema de tutor inteligente se contextualiza en evaluaciones adaptativas, dejando abierta la implementación de agentes inteligentes o inteligencia artificial en próximos trabajos.

Es necesario que se defina una prueba de diagnóstico para evaluar el nivel del estudiante, categorizarlo por el nivel al grupo de estudiantes que se los desee evaluar. Se recomienda que se definan y asignen los roles para la parte administrativa para que se encargue de administrar y controlar el ingreso de los usuarios, contraseñas y demás. Se ha priorizado que el centro del programa sea la parte del estudiante, que es donde se llevará a cabo el análisis de los conocimientos adquiridos por el estudiante a través de un modelo matemático que tendrá como reglas ponderar con asignaciones respectivas los recursos revisados y las preguntas contestadas. Con una calificación mínima de 7 para poder pasar al siguiente nivel.

La sección de aprendizaje y las reglas es muy importante es por ello que se haría necesario implementar nuevas características y funcionalidades en cada uno de los módulos del tutor como nuevas reglas del tuto, asignaciones del tutor y nuevas interacciones del tutor.

BIBLIOGRAFÍAS

- Active, B. (29 de Octubre de 2015). *Why you should use a Virtual Learning EnvironmenT*. Obtenido de Why you should use a Virtual Learning EnvironmenT: <http://goo.gl/d8YjDP>
- Acuña, P. (2019, February 26). *Aprendizaje Adaptativo: Conocimiento personalizado*. EVirtualplus. <https://www.evvirtualplus.com/aprendizaje-adaptativo/>
- Alberto. (2015). *Una introducción al diseño adaptable*. Code Envato Tuts+. <https://code.tutsplus.com/es/tutorials/an-introduction-to-adaptive-design--cms-22888>
- Alea, M., Miqueo, J., & Aguilar, A. (2005). Consideraciones sobre las habilidades fundamentales en la enseñanza de la informática. *V Conferencia Internacional Científico Pedagógica de Educación Física y Deportes*. Universidad del Deporte Nancy Uraña Romagosa.
- Aretio, G. (15 de September de 2001). Obtenido de <https://www.redalyc.org/pdf/780/78011231005.pdf>
- Arweb Agencia Digital Costa Rica. (2014, September 26). ¿Qué es Bootstrap y cómo funciona en el diseño web? *Blog, ARWEB Agencia Digital Costa Rica*. <https://www.arweb.com/blog/%c2%bfque-es-bootstrap-y-como-funciona-en-el-diseno-web/>
- Bastar, S. G. (2012). Metodología de la Investigación. En R. T. S.C, *Metodología de la Investigación*. Ciudad de Mexico.
- Bruno Pelaez. (2018). *¿Qué es el desarrollo rápido de aplicaciones (RAD)?* Capterra. <https://www.capterra.es/blog/1218/que-es-el-desarrollo-rapido-de-aplicaciones-rad>
- Canales, M. (2006). *Metodologías de la investigación social*. LOM Ediciones. <https://www.digitaliapublishing.com/a/13069/metodologias-de-la-investigacion-social>
- Carrillo, M. d. (2014). *Estilos de Aprendizaje de docentes y alumnos y su relacion con el rendimiento academico*.
- Cataldi. (20 de March de 2004). *Revista Electronica Educativa*. Obtenido de <http://www.redicces.org.sv/jspui/bitstream/10972/485/1/Tutor%20inteligente.pdf>
- Cuadros comparativos entre entrevista y encuesta | Cuadro Comparativo*. (2018, June 10). <https://cuadrocomparativo.org/cuadros-comparativos-entre-entrevista-y-encuesta/>
- CUAED, U. (2016). *Opción múltiple V2.1*. Opción múltiple V2.1. https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod_resource/content/1/contenido/evaluacion/opcion_multiple/index.html
- Davila, G. G., & Davila, M. C. G. (2000). *Metodología de la Investigación*. Grupo Editorial Patria.
- Díaz-Bravo, L., Torruco-García, U., Martínez-Hernández, M., & Varela-Ruiz, M. (2013). La entrevista, recurso flexible y dinámico. *Investigación en educación médica*, 2(7), 162–167.

- Díaz, K. I., & Crespo, T. (2010). La conceptualización de las habilidades informáticas. *Revista electrónica IPLAC*, 5.
- Díaz Tejera, K. I., Fierro Martín, E., & Muñoz Pentón, M. A. (2018). La enseñanza de la programación: Una experiencia en la formación de profesores de informática. *Educación*, 27(53), 73–91. <https://doi.org/10.18800/educacion.201802.005>
- Economipedia. (2018). *Investigación descriptiva*. Economipedia. <https://economipedia.com/definiciones/investigacion-descriptiva.html>
- Edacom. (2015). *Edacom*. Obtenido de Edacom: <https://blog.edacom.mx/importancia-ensenar-programacion-ninos>
- Educacao, P. (2020). *Portal Educação—Artigo*. <https://siteantigo.portaleducacao.com.br/conteudo/artigos/informatica/principais-caracteristicas-do-mysql/66677>
- Elige Educar. (2021). *6 metodologías de enseñanza que todo profesor innovador debería conocer*. Elige Educar. <https://eligeeducar.cl/ideas-para-el-aula/6-metodologias-ensenanza-profesor-innovador-deberia-conocer/>
- FastTest. (2013). *FastTest*. FastTest: Adaptive Testing and Online Assessment. <https://www.fasttestweb.com/fasttest-2/>
- Felder, R. (1990). How engineering students learn, how engineering professor teach, and what goes wrong in the process. En R. Felder, *How engineering students learn, how engineering professor teach, and what goes wrong in the process*. (págs. 82-84). Minesota.
- Feng, M., Heffernan, N., & Koedinger, K. (1991). *Student modeling in an Intelligent Tutoring System*. 21.
- Ferreira, A., Salcedo, P., Kotz, G., & Barrientos, F. (2012). La Arquitectura de ELE-TUTOR: Un Sistema Tutorial Inteligente para el Español como Lengua Extranjera. *Revista Signos*, 45(79), 102–131. <https://doi.org/10.4067/S0718-09342012000200001>
- Galvis. (12 de June de 1994). Obtenido de https://www.researchgate.net/publication/242587527_Ingenieria_de_software_educativo_con_modelaje_orientado_a_objetos_Un_medio_para_desarrollar_micro-mundos_interactivos
- Galvis. (12 de June de 1994). *ReaserchGate*. Obtenido de ReaserchGate: https://www.researchgate.net/publication/242587527_Ingenieria_de_software_educativo_con_modelaje_orientado_a_objetos_Un_medio_para_desarrollar_micro-mundos_interactivos

- Galvis. (12 de March de 2000). *ReaserchGate*. Obtenido de ReaserchGate: https://www.researchgate.net/publication/242587527_Ingenieria_de_software_educativo_con_modelaje_orientado_a_objetos_Un_medio_para_desarrollar_micro-mundos_interactivos
- Garcia Aretio. (1994). Obtenido de <https://www.redalyc.org/pdf/780/78011231005.pdf>
- Hernández, J. A. D., & Rengifo, Y. S. P. (2015). Los sistemas tutores inteligentes y su aplicabilidad en la educación. *Horizontes Pedagógicos*, 17(2), 104–116.
- Herrera, J. (2017). *La investigación cualitativa*. <http://148.202.167.116:8080/xmlui/handle/123456789/1167>
- Incredibuild. (2021). *What is Visual Studio?* Incredibuild. <https://www.incredibuild.com/integrations/visual-studio>
- Jacobs, P. F., & Reid, D. T. (1992). *Rapid prototyping & manufacturing: Fundamentals of stereolithography* (1st ed). Society of Manufacturing Engineers.
- Jaimez-González R. Carlos; (2016). *Portal web con recursos didácticos digitales para el aprendizaje de HTML y CSS*. http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S2007-74672017000200833
- Laedu.Digital. (2020, April 15). Aprendizaje y evaluación adaptativos. *Estudia en línea*. <https://laedu.digital/2020/04/15/aprendizaje-y-evaluacion-adaptativos/>
- Laura M. (2017). *Aprender programación desde cero—Guía completa de desarrollo web*. BitDegree.org Online Learning Platforms. <https://es.bitdegree.org/tutoriales/aprender-programacion-desde-cero/>
- L. Edith (4 de June de 1993). Las configuraciones didacticas. En *Las configuraciones didacticas* (pág. 109). Paidós. Obtenido de <https://media.utp.edu.co/referencias-bibliograficas/uploads/referencias/libro/689-las-configuraciones-didacticas-una-nueva-agenda-para-la-ensenanza-superiorpdf-lyQWd-articulo.pdf>
- Mark Otto, Jacob Thornton, and Bootstrap. (2020). *Bootstrap*. Bootstrap. <https://getbootstrap.com/>
- Mathieu Mihaela Juganaru. (2016). *Introducción a la programación*. 59.
- Microsoft. (2017). *What is .NET? An open-source developer platform*. Microsoft. <https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet>
- Mora, S. L. (2002). *Programación de aplicaciones web: Historia, principios básicos y clientes web*. Editorial Club Universitario. <https://dialnet.unirioja.es/servlet/libro?codigo=204176>

Observatorio Digital. (2020). *10 Herramientas para aprender a programar con niños – O.Digital Observatorio.Digital*. <https://observatorio.digital/blog/10-herramientas-para-aprender-a-programar-con-ninos/>

Oracle. (2022). *Mysql*. <https://www.mysql.com/>

Oracle Mexico. (2020). *¿Qué es la inteligencia artificial (IA)?* Oracle. <https://www.oracle.com/mx/artificial-intelligence/what-is-ai/>

Ortiz de Zevallos, C. M. V. (2014). Estrategias didácticas para el desarrollo de la identidad cultural en educación primaria. *Educación*, 23(45), 25–50.

Ortiz-García, J. M. (2006). *Guía descriptiva para la elaboración de protocolos de investigación*. 12(3), 12.

Perales. (27 de July de 2000). Obtenido de https://issuu.com/carlitosanchez0/docs/ense__ar_y_aprender

Pinion, D. (2020, July 22). Ventajas de aprender a programar a temprana edad. *Pinion Education*. <https://pinion.education/es/blog/ensenanza-y-futuro-de-la-programacion/>

QuestionPro. (2018, September 27). *¿Qué es la investigación experimental?* *QuestionPro*. <https://www.questionpro.com/blog/es/investigacion-experimental/>

Rock Content. (2019, April 20). *¿Qué es un lenguaje de programación y qué tipos existen?* *Rock Content - ES*. <https://rockcontent.com/es/blog/que-es-un-lenguaje-de-programacion/>

Rodríguez Chávez, M. H., & Rodríguez Chávez, M. H. (2021). Sistemas de tutoría inteligente y su aplicación en la educación superior. *RIDE. Revista Iberoamericana para la Investigación y el Desarrollo Educativo*, 11(22). <https://doi.org/10.23913/ride.v11i22.848>

Rodriguez Puerta Alejandro (2019, November 17). Tipos de aprendizaje y sus características (con ejemplos). *Lifeder*. <https://www.lifeder.com/tipos-aprendizaje-psicologia/>

Rouhiainen, L. (2018). *Inteligencia artificial: 101 cosas que debes saber hoy sobre nuestro futuro*.

Sacristan, J. G. (1990). Los materiales y la enseñanza. En *Los materiales y la enseñanza*.

Salgueiro, F. (7 de February de 2001). Obtenido de <https://core.ac.uk/download/pdf/301043695.pdf#:~:text=El%20subm%C3%B3dulo%20de%20Protocolos%20Pedag%C3%B3gicos%20es%20el%20encargado,perfil%2C%20encargado%20de%20evaluar%20las%20caracter%C3%ADsticas%20del%20estudiante>

Santiago. (17 de July de 2017). *Blogspot*. Obtenido de <http://jhonnovoa69.blogspot.com/2017/07/rol-del-tutor-virtual.html>

Sistema Tutores Inteligentes—EcuRed. (2018). https://www.ecured.cu/Sistema_Tutores_Inteligentes

Subitus. (2019, July 31). *¿Qué son las evaluaciones adaptativas?* *SUBITUS - Expertos en e-learning*. <https://www.subitus.com/evaluaciones-adaptativas-y-actualizadas-para-e-learning/>

- UCSG. (2021). *Repositorio Digital UCSG*. http://repositorio.ucsg.edu.ec/simple-search?location=%2F&query=tutor&rpp=10&sort_by=score&order=desc
- Urretavizcaya, M. (2001). Sistemas Inteligentes en el ambito de la Educacion. *INTELIGENCIA ARTIFICIAL*, 5(12), 292. <https://doi.org/10.4114/ia.v5i12.703>
- Verywell Mind. (2018). *Why Alfred Binet Developed IQ Testing for Students*. Verywell Mind. <https://www.verywellmind.com/history-of-intelligence-testing-2795581>

ANEXOS

ANEXO #1

Anexo 1: Syllabus Nuevo Rediseño de Lenguajes de Programación del período B-2021 otorgado por el docente de la materia. Ing. José Miguel Erazo



SYLLABUS NUEVO REDISEÑO

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL
INGENIERIA - COMPUTACIÓN (R)

Fecha: 10-12-2021 06:22:28

Malla:	REDISEÑO 2020		
Ciclo:	III	Periodo:	PERIODO B-2021
Asignatura:	LENGUAJES DE PROGRAMACIÓN	Paralelo:	PARALELO A

1. DATOS GENERALES:

No. DE HORAS TOTALES	No. DE HORAS COMPONENTE DE DOCENCIA		No. DE HORAS COMPONENTE DE PRÁCTICAS DE APLICACIÓN Y EXPERIMENTACIÓN	No. DE HORAS COMPONENTE DE APRENDIZAJE AUTÓNOMO	No. DE HORAS SEMANALES
192:00	80:00		64:00	48:00	05:00
	ACTIVIDADES DE APRENDIZAJE ASISTIDAS POR EL PROFESOR	ACTIVIDADES DE APRENDIZAJE COLABORATIVO			No. DE HORAS SEMESTRALES PROGRAMADAS
	16:00	64:00			144:00
UNIDAD DE ORGANIZACIÓN CURRICULAR (UOC)	BÁSICA	CAMPO DE FORMACIÓN	PRAXIS PROFESIONAL		
PRÁCTICAS PREPROFESIONALES	NO	No. DE HORAS:			
ASIGNATURA INTEGRADORA	NO				
ASIGNATURA DE ITINERARIO	NO				
PROFESOR(A):	ERAZO AYON JOSE MIGUEL				
PERFIL DEL PROFESOR(A):	- INGENIERO EN SISTEMAS COMPUTACIONALES - MAGISTER EN ADMINISTRACION DE EMPRESAS				

2. OBJETIVO DE LA ASIGNATURA:

DESARROLLAR E IMPLEMENTAR SOLUCIONES INFORMÁTICAS UTILIZANDO LENGUAJES DE PROGRAMACIÓN.

3. RESULTADOS DEL APRENDIZAJE DE LA ASIGNATURA:

IDENTIFICA LOS PRINCIPIOS DE LA PROGRAMACIÓN IMPERATIVA.
DISEÑA E IMPLEMENTA UN PROGRAMA EN UN LENGUAJE DE PROGRAMACIÓN IMPERATIVO.
RECONOCE LOS PRINCIPIOS FUNDAMENTALES DEL PARADIGMA FUNCIONAL.
DISEÑA E IMPLEMENTA PROGRAMAS EN ALGÚN TIPO DE LENGUAJE FUNCIONAL.
DISEÑA E IMPLEMENTA PROGRAMAS EN UN LENGUAJE DE PROGRAMACIÓN BASADO EN LOS PARADIGMAS ESTUDIADOS.
DESARROLLA APLICACIONES WEB UTILIZANDO INTERFACES DE USUARIO DE ALTO NIVEL
CONSTRUYE APLICACIONES QUE CONSUMEN UNA BASE DE DATOS
DISEÑA E IMPLEMENTA APLICACIONES WEB Y DE ESCRITORIO CON COMUNICACIÓN A BASES DE DATOS

4. ESTRUCTURA DE LA ASIGNATURA POR UNIDADES:

UNIDAD	No. DE HORAS TOTALES PARA CADA UNIDAD	OBJETIVOS	RESULTADOS DEL APRENDIZAJE	No. DE HORAS DE APRENDIZAJE AUTÓNOMO (AU) PARA CADA UNIDAD
LENGUAJES IMPERATIVOS	20:00	IDENTIFICAR LOS PRINCIPIOS DE LA PROGRAMACIÓN IMPERATIVA. DISEÑAR E IMPLEMENTAR UN PROGRAMA EN UN LENGUAJE DE PROGRAMACIÓN IMPERATIVO.	IDENTIFICA LOS PRINCIPIOS DE LA PROGRAMACIÓN IMPERATIVA. DISEÑA E IMPLEMENTA UN PROGRAMA EN UN LENGUAJE DE PROGRAMACIÓN IMPERATIVO.	12:00
LENGUAJES FUNCIONALES	20:00	RECONOCER LOS PRINCIPIOS FUNDAMENTALES DEL PARADIGMA FUNCIONAL. DISEÑAR E IMPLEMENTAR PROGRAMAS EN ALGÚN TIPO DE LENGUAJE FUNCIONAL.	RECONOCE LOS PRINCIPIOS FUNDAMENTALES DEL PARADIGMA FUNCIONAL. DISEÑA E IMPLEMENTA PROGRAMAS EN ALGÚN TIPO DE LENGUAJE FUNCIONAL.	12:00
INTERFACES WEB	20:00	DESARROLLAR INTERFACES WEB UTILIZANDO BIBLIOTECAS MULTIPLATAFORMAS PARA SITIOS WEBS. DISEÑAR Y MAQUETAR SITIOS WEBS DE ALTO NIVEL	DESARROLLAA INTERFACES WEB UTILIZANDO BIBLIOTECAS MULTIPLATAFORMAS PARA SITIOS WEBS. DISEÑA Y MAQUETAA SITIOS WEBS DE ALTO NIVEL	12:00
CONEXIONES A BASES DE DATOS	20:00	COMPRENDER LOS MÉTODOS DE CONEXIÓN A UNA BASE DE DATOS PARA CREAR, MODIFICAR, ELIMINAR Y ELIMINAR REGISTROS. DESARROLLAR INTERFACES DE USUARIO QUE PERMITAN COMUNICARSE CON INFORMACIÓN ALMACENADA EN UNA BASE DE DATOS.	COMPRENDE LOS MÉTODOS DE CONEXIÓN A UNA BASE DE DATOS PARA CREAR, MODIFICAR, ELIMINAR Y ELIMINAR REGISTROS. DESARROLLAR INTERFACES DE USUARIO QUE PERMITAN COMUNICARSE CON INFORMACIÓN ALMACENADA EN UNA BASE DE DATOS.	12:00

LENGUAJES IMPERATIVOS

TEMAS	CONTENIDOS	No. DE HORAS PARA CADA TEMA
LENGUAJES IMPERATIVOS	INTRODUCCIÓN	02:00
LENGUAJES IMPERATIVOS	MANEJO DE DATOS Y TIPOS	02:00
LENGUAJES IMPERATIVOS	ASIGNACIONES Y EXPRESIONES	02:00
LENGUAJES IMPERATIVOS	FLUJOS DE CONTROL	02:00
LENGUAJES IMPERATIVOS	COMPONENTES DE UN PROGRAMA IMPERATIVO	02:00
LENGUAJES IMPERATIVOS	EJEMPLOS DE PROGRAMAS IMPERATIVOS	02:00
LENGUAJES IMPERATIVOS	USO DE LENGUAJES DE PROGRAMACIÓN PREPROCESADOR DE HYPER TEXTO	04:00
LENGUAJES IMPERATIVOS	MANEJO DE LENGUAJE DE PROGRAMACIÓN MULTIPARADIGMA	04:00

LENGUAJES FUNCIONALES

TEMAS	CONTENIDOS	No. DE HORAS PARA CADA TEMA
LENGUAJES FUNCIONALES	INTRODUCCIÓN A LOS LENGUAJES FUNCIONALES	02:00
LENGUAJES FUNCIONALES	DEFINICIÓN DE FUNCIÓN	02:00
LENGUAJES FUNCIONALES	LISTAS	04:00
LENGUAJES FUNCIONALES	TIPOS Y POLIMORFISMOS	02:00
LENGUAJES FUNCIONALES	FUNCIONES DE ORDEN SUPERIOR	04:00
LENGUAJES FUNCIONALES	LAZY EVALUATION	02:00
LENGUAJES FUNCIONALES	EJEMPLOS DE PROGRAMAS FUNCIONALES	02:00
LENGUAJES FUNCIONALES	USO DE LENGUAJE DE PROGRAMACIÓN JAVASCRIPT	02:00

INTERFACES WEB

TEMAS	CONTENIDOS	No. DE HORAS PARA CADA TEMA
INTERFACES WEB	CONCEPTOS	02:00
INTERFACES WEB	LENGUAJE DE ETIQUETAS HTML5	04:00
INTERFACES WEB	LENGUAJE DE ETIQUETAS CSS	04:00
INTERFACES WEB	BOOTSTRAP: BOTONES, ETIQUETAS, TABLAS GRID	04:00
INTERFACES WEB	INTEGRACIÓN DE BOOTSTRAP CON LOS CONTROLES WEB	02:00
INTERFACES WEB	VALIDACIÓN DE DATOS	04:00

CONEXIONES A BASES DE DATOS

TEMAS	CONTENIDOS	No. DE HORAS PARA CADA TEMA
CONEXIONES A BASES DE DATOS	INTRODUCCIÓN A LAS BASES DE DATOS	02:00
CONEXIONES A BASES DE DATOS	CADENAS DE CONEXIÓN	01:00
CONEXIONES A BASES DE DATOS	ELEMENTOS DEL CRUD	02:00
CRUD	INGRESO DE INFORMACIÓN	03:00
CRUD	LECTURA DE INFORMACIÓN	03:00
CRUD	MODIFICACIÓN DE INFORMACIÓN	03:00
CRUD	ELIMINACIÓN DE INFORMACIÓN	02:00
USO DE CONTROLES	CARGA DE DATOS EN CONTROLES: TEXTBOX, DROPDOWNLIST, RADIOBUTTONLIST, LIST, CHECKBOXLIST, FILEUPLOAD	02:00
MODELAMIENTO DE BASE DE DATOS	MODELAMIENTO DE BASE DE DATOS: ASIGNADOR RELACIONAL DE OBJETOS (ORM)	02:00

6. DETALLE DE LA PRÁCTICA EXPERIMENTAL

La práctica experimental EN CLASES es de 16:00 horas; que se distribuirán de la siguiente forma:		
PRIMER PARCIAL	08:00	DESCRIPCIÓN: - DESARROLLO DE SITIO WEB UTILIZANDO LENGUAJES DE PROGRAMACIÓN IMPERATIVOS Y FUNCIONAL, ORIENTADO A LA SOLUCIÓN DE UN PROBLEMA EMPRESARIAL. - DESARROLLO DE APLICACIÓN WEB UTILIZANDO UNA CONEXIÓN DE BASE DE DATOS, LA BASE DE DATOS MANTENDRÁ ENTIDADES RELACIONADAS PARA PODER OBTENER LOS DATOS DE DIFERENTES ENTIDADES Y DESARROLLAR EL CRUD DE CADA UNA DE ELLAS.
SEGUNDO PARCIAL	08:00	
La práctica experimental GUIADA es de 64:00 horas; que se distribuirán de la siguiente forma:		
PRIMER PARCIAL	32:00	DESCRIPCIÓN: - DESARROLLO DE EJERCICIOS UTILIZANDO LOS CONTENIDOS APRENDIDOS EN CLASE, TANTO EN TALLER CON EL DOCENTE COMO EN CASA PRÁCTICAS Y APRENDIZAJE EN LA APLICACIÓN SOLOLEARN. LA CUAL CONTIENE CURSOS CON COMPILADORES DESDE EL CELULAR, PARA PRÁCTICAS DE LOS LENGUAJES DE PROGRAMACIÓN ESTUDIADOS. - DESARROLLO DE EJERCICIOS UTILIZANDO LOS CONTENIDOS APRENDIDOS EN CLASE, TANTO EN TALLER CON EL DOCENTE COMO EN CASA INVESTIGACIÓN SOBRE COMPONENTES TECNOLÓGICOS INVESTIGACIÓN SOBRE CONEXIONES CON OTRAS BASES DE DATOS.
SEGUNDO PARCIAL	32:00	

7. METODOLOGÍA

LAS SESIONES DE APRENDIZAJE COMBINARÁN LA EXPOSICIÓN DEL DOCENTE CON LA PARTICIPACIÓN ACTIVA DE LOS ESTUDIANTES PARA DESARROLLAR LOS CONTENIDOS, LOS TRABAJOS INDIVIDUALES Y GRUPALES.
EL DOCENTE ASUME EL ROL DE MEDIADOR PARA PRESENTAR LOS CONTENIDOS CONCEPTUALES Y, DE ORGANIZADOR DE SITUACIONES, PARA ASEGURAR LA PARTICIPACIÓN DE LOS ALUMNOS EN LOS TALLERES GRUPALES.
LAS TÉCNICAS DIDÁCTICAS SUGERIDAS PARA EL DESARROLLO DE LAS CLASES ASISTIDAS POR EL DOCENTE, EL APRENDIZAJE COLABORATIVO Y LA APLICACIÓN PRÁCTICA:
- EXPOSICIÓN DE LAS TAREAS DE INVESTIGACIÓN BIBLIOGRÁFICA CON PARTICIPACIÓN ACTIVA DE LOS ESTUDIANTES.
- SOLUCIÓN DE PROBLEMAS PROPUESTOS POR EL PROFESOR EN EL AULA VIRTUAL PARA SER DESARROLLADOS EN CLASE O EN EL DOMICILIO EN GRUPOS DE TRABAJO.
- PRESENTACIÓN EN EL AULA DE EXPERIMENTOS DEMOSTRATIVOS, APLICACIONES, VIDEOS Y SIMULACIONES DE FENÓMENOS FÍSICOS QUE REFUERZEN LOS CONCEPTOS TEÓRICOS PROPORCIONADOS EN LA CLASE.
- REALIZACIÓN POR EL ESTUDIANTE DE PRÁCTICAS DE LABORATORIO DE ACUERDO A UNA GUÍA. LOS EXPERIMENTOS TIENEN RELACIÓN DIRECTA CON LOS TEMAS DESARROLLADOS EN CLASE.
- USO DE PROGRAMAS INFORMÁTICOS (PHP, C#, JAVASCRIPT)

PARA LAS CLASES SE HARÁ USO INTENSIVO DE MULTIMEDIA, PRESENTADORES DE DIAPOSITIVAS, PROCESADORES DE TEXTO, VIDEOS, INTERNET Y AULA VIRTUAL. ESTO SE COMPLEMENTA CON EL USO DE NOTAS Y APUNTES DE CLASE DEL PROFESOR COLOCADOS EN EL AULA VIRTUAL. SE DESARROLLARÁN TÉCNICAS DE INVESTIGACIÓN Y DINÁMICAS DE GRUPO, CON UNA SERIE DE EJERCICIOS Y PROBLEMAS QUE PERMITIRÁN FIJAR EN EL ESTUDIANTE LOS CONOCIMIENTOS ADQUIRIDOS.
SE CONSTITUIRÁN EQUIPOS PARA INVESTIGAR E INTERCAMBIAR EXPERIENCIAS DE APRENDIZAJE Y TRABAJO QUE SE EXPRESARÁ EN LA ELABORACIÓN Y DESARROLLO DE TRABAJOS DE INVESTIGACIÓN.
EL DOCENTE DETECTARÁ LOS APRENDIZAJES NO LOGRADOS POR LOS ESTUDIANTES AL FINAL DE CADA EVALUACIÓN Y ORGANIZARÁ LAS ACCIONES PEDAGÓGICAS NECESARIAS PARA OPTIMIZAR LOS APRENDIZAJES EN LOS PUNTOS CRÍTICOS DETECTADOS.

PRECISIONES DEL PROFESOR

DESARROLLO DE EJERCICIOS UTILIZANDO LOS CONTENIDOS APRENDIDOS EN CLASE, TANTO EN TALLER CON EL DOCENTE COMO EN CASA INVESTIGACIÓN SOBRE COMPONENTES TECNOLÓGICOS INVESTIGACIÓN SOBRE CONEXIONES CON OTRAS BASES DE DATOS.

Para el componente D-AP (16 Horas), se recomienda que la asignatura desarrolle en : horas de clases prácticas (TALLER, SEMINARIO, EJERCICIO, LABORATORIO).

8. EVALUACIÓN

APRENDIZAJE EN AULA Y AUTÓNOMO	APRENDIZAJE PRÁCTICO EXPERIMENTAL	EVALUACIÓN PARCIAL
35%	30%	35%

ASISTIDO POR EL PROFESOR: LECCIONES POR CADA UNIDAD FINALIZADA.
COLABORATIVO: TALLER POR CADA UNIDAD FINALIZADA.
PRÁCTICAS DE A Y E: INVESTIGACIÓN Y ENTREGA DE EJERCICIOS DE PRÁCTICA POR CADA TEMA TRATADO EN CADA UNIDAD.
EVALUACION PARCIAL: REACTIVOS DE OPCIÓN MÚLTIPLE (TIPO SELECCIÓN, RELACIÓN, DESARROLLO) EN RELACIÓN A TAXONOMÍA DE BLOOM EN NIVELES DE CONOCIMIENTO, COMPRENSIÓN Y APLICACIÓN)

PRECISIONES DEL PROFESOR

LA EVALUACIÓN SE REALIZARÁ DE MANERA CONTINUA Y FINAL. SE ENVIARÁN TAREAS INDIVIDUALES Y GRUPALES LAS CUALES SERÁN CALIFICADAS DE 0 A 10 DE ACUERDO A CONTENIDO, DESARROLLO Y PRESENTACIÓN. SE CALIFICARÁ DE ACUERDO AL SIGUIENTE FORMATO: 20% DE LA NOTA POR LA IDENTIFICACIÓN DEL PROBLEMA O CASO, 80% DE LA NOTA POR EL DESARROLLO ADECUADO Y EL CUMPLIMIENTO DE LOS RESULTADOS ESPERADOS EN EL PROCESO DE DESARROLLO, ACTUACIONES DENTRO DE CLASE, INVESTIGACIONES, ENSAYOS, LECTURAS DIRIGIDAS, CURSO EN LÍNEA EXPOSICIONES. LOS ENSAYOS E INVESTIGACIONES DEBEN PASAR POR EL SOFTWARE ANTI-PLAGIO Y DEBERÁN ADJUNTAR LAS REFERENCIAS Y BIBLIOGRÁFICAS. TODAS LAS EVALUACIONES SE CALIFICARÁN SOBRE DIEZ (10).

9. BIBLIOGRAFÍA

8.1 Básica

Tipo: LIBROS **Editorial:** PERASON

Título: CONCEPTS OF PROGRAMMING LANGUAGES **Edición:** 10

Autor: ROBERT W. SEBESTA **Año Pub.:** 2016

ISBN:

Link:

8.2 Complementaria

Tipo: LIBROS **Editorial:** O'REILLY
Título: LEARNING PHP, MYSQL & JAVASCRIPT: WITH JQUERY, CSS & HTML5 **Edición:**
Autor: NIXON, ROBIN **Año Pub.:** 2009
ISBN:
Link:

Tipo: LIBROS **Editorial:** RAMA
Título: MICROSOFT C#, CURSO DE PROGRAMACIÓN. 2ª EDICIÓN (PROFESIONAL) **Edición:** 2
Autor: CEBALLOS, FRANCISCO JAVIER **Año Pub.:** 2011
ISBN:
Link:

Tipo: LIBROS **Editorial:** AMAZON KINDLE
Título: APRENDIENDO JAVASCRIPT: DESDE CERO HASTA ECMASCRIPT 6 **Edición:**
Autor: CARLOS AZAUSTRE **Año Pub.:** 2015
ISBN:
Link:

ANEXO #2

Anexo 2: Entrevista al docente de la materia lenguajes de Programación - Ing. José Erazo Ayón con sus respectivas preguntas y respuestas

Entrevista al Ing. José Erazo Ayón docente de la materia Lenguajes de Programación	
Pregunta a realizar	<i>¿Qué tipos de aprendizaje cree usted que se emplean en sus clases?</i>
Respuesta del Entrevistado	Según lo planteado en clase, a mi parecer el aprendizaje asociativo es el que se emplea, basado con los problemas que se plantean en clase y las herramientas que se utilizan (presentaciones Power Point, soluciones de código, ejercicios a realizar en clase, entre otros), se busca que el cerebro del estudiante reaccione y pueda hacer un “match” con los conocimientos impartidos y la problemática planteada haciendo comparativas con escenarios de la vida real en cada uno de los temas de los lenguajes de programación
Pregunta a realizar	<i>Según sus experiencias impartiendo clases, ¿qué habilidades desarrollan sus estudiantes cuando enseña su materia?</i>
Respuesta del Entrevistado	Resolución de problemas, aprender a modelar situaciones reales en sistema que se va a codificar, no solo en la materia de lenguajes de programación sino para el ámbito laboral, en otras

	<p>palabras los estudiantes en la materia Lenguajes de Programación, adquieren la habilidad de interpretar problemas para así poder plantear distintas soluciones.</p>
<p>Pregunta a realizar</p>	<p><i>¿Qué herramientas usted utiliza para facilitar el aprendizaje de los estudiantes en la materia lenguajes de programación?</i></p>
<p>Respuesta del Entrevistado</p>	<p>En cuanto a software, serían los distintos programas y bases de datos para poder compilar (Visual Studio, MySql, Visual Code, Apache, entre otros), así mismo planeamiento de ejercicios de programación desde 0, con su respectiva proyección en clase ya sea presencial o virtual, y dando explicaciones, pautas y retroalimentaciones respectivas para que sea un aprendizaje donde el estudiante interactúe con el docente y asocie cada concepto que se imparta en las clases.</p>
<p>Pregunta a realizar</p>	<p><i>¿Qué protocolo pedagógico generalmente maneja al impartir sus clases? ¿Su protocolo varía en base al tema impartido?</i></p>
<p>Respuesta del Entrevistado</p>	<p>A nivel virtual creo que es un protocolo general, en este caso se ingresa a la sesión de clases, conversamos un poco, realizamos retroalimentaciones de las</p>

	<p>clases anteriores, despejando cualquier duda que exista. Se comenta lo que se va a trabajar el día de la clase, se empieza a trabajar con normalidad y mientras avance la clase, elaboro preguntas con el fin de identificar si el estudiante está inmerso en el tema, cuando algo no está claro. Doy paso a ejemplos asociativos de la clase que imparto con la vida real para que así se despejen dudas de lo que doy a explicar a los estudiantes de mi materia.</p> <p>Mi protocolo a nivel virtual varía muy poco, ocasionalmente formo grupos para que realicen presentaciones en Power Point (exposiciones) y el proyecto de tutoría que sería como un “Aprendizaje basado en proyectos” con el fin de que los estudiantes demuestren lo aprendido realizando un pequeño proyecto con una fecha de fin determinada</p>
<p>Pregunta a realizar</p>	<p><i>¿Qué estilo de aprendizaje cree que tienen los estudiantes de la materia Lenguajes de Programación?</i></p>
<p>Respuesta del Entrevistado</p>	<p>A mi parecer creo que los estudiantes son pragmáticos debido a que a mi parecer mis estudiantes a mi criterio son experimentales, eficaces, objetivos, y sobre todo son muy concretos a la hora de realizar sus tareas y actividades.</p>

	<p>En lo personal trato de siempre dar las herramientas en clases, para que los estudiantes prueben las soluciones bajo diferentes problemas o situaciones y que por el mismo ingenio del estudiante, ellos encuentren una solución</p>
<p>Pregunta a realizar</p>	<p><i>A nivel presencial, los ayudantes de cátedra, ¿Eran suficientes para complementar y mejorar el nivel de aprendizaje del estudiante?</i></p>
<p>Respuesta del Entrevistado</p>	<p>La funcionalidad del ayudante de cátedra como su nombre lo indica es de ayudar, cuando se trata de ayudar lo hacen, pero depende del estudiante y la voluntad que este tenga para poder aprender o mejorar sus conocimientos. Esto no solo debe ser en la materia de Lenguajes de Programación, sino también en todas las materias que el estudiante ve a lo largo de la carrera.</p>
<p>Pregunta a realizar</p>	<p><i>¿En que se basa el programa de estudio de su materia? ¿Está sujeto a cambios?</i></p>
<p>Respuesta del Entrevistado</p>	<p>Se basa como tal en la necesidad que tienen los estudiantes, es decir, aprender a programar en ciertos lenguajes de programación y hacer un “match” con las prácticas pre-profesionales y el resto de las asignaturas de la carrera.</p>

	El programa de estudios si está sujeto a cambios porque el ambiente laboral cambia, las tecnologías cambian, pero se realizaría el cambio siempre y cuando se justifique la mejora para la formación académica-profesional de los estudiantes.
Pregunta a realizar	<i>Menciona 2 veces la palabra “match” ¿qué es para usted esta palabra en el ámbito académico?</i>
Respuesta del Entrevistado	Para mi quiere decir que el estudiante al momento de asociar o familiarizar un término, concepto o tema en concreto tenga una especie de “clic” o entendimiento de lo que se imparte en las clases.
Pregunta a realizar	<i>¿Alguna vez ha considerado usted que su syllabus o plan de estudio debió cambiar totalmente para mejorar el aprendizaje del estudiante? ¿Porque?</i>
Respuesta del Entrevistado	Si, si se lo ha cambiado. Es más, no ha pasado un semestre que he dado igual una materia, siempre es cambiante. En otras palabras se ajusta el syllabus a las necesidades de los estudiantes y sobre todo el mercado.
Pregunta a realizar	<i>¿En qué aspectos cree usted que beneficiará el uso del tutor inteligente para la enseñanza y aprendizaje de los estudiantes?</i>

<p>Respuesta del Entrevistado</p>	<p>El tema de la programación es muy autodidacta, a mi criterio basta que enseñen lo del inicio y el resto depende del estudiante. Yo creo que ayudará mucho y permitirá centralizar el tema de la auto-ayuda que se tiene para el proceso de aprendizaje del estudiante de la materia Lenguajes de Programación</p>
<p>Pregunta a realizar</p>	<p><i>¿Cómo considera usted que se puede impartir el aprendizaje de los lenguajes de programación a los estudiantes, a través del tutor inteligente?</i></p>
<p>Respuesta del Entrevistado</p>	<p>Podría ser de manera complementaria a la formación que se lleva en la carrera, o como una complementación de las ayudantías de cátedra. Yo creo que podría servir como ayuda para fortalecer temas, tópicos y vacíos que tenga el estudiante en su aprendizaje de los lenguajes de programación como tal.</p> <p>En mi materia, el tiempo queda corto, yo mando sistemas de autoeducación cuando vemos temas de soloLearn, en el cual los estudiantes se conectan a esta plataforma, los estudiantes siguen los cursos y a su vez se van autoevaluando y este certificado para mí como docente, es necesario para la evaluación de los diferentes parciales.</p>

<p>Pregunta a realizar</p>	<p><i>¿Cuál es la calificación que por lo general (Promedio) que sacan los estudiantes en sus evaluaciones? y ¿Qué conclusión saca a partir de estas? ¿Es necesaria una retroalimentación?</i></p>
<p>Respuesta del Entrevistado</p>	<p>Está un poco más arriba de la media, entre 6-7/10, habiendo también los estudiantes que despuntan y van 8/10 para arriba.</p> <p>Las retroalimentaciones son necesarias, pero tampoco abusar de ellas. Depende del estudiante si pone de su parte, auto-aprendizaje y autocrítica en cuanto a que vacíos tiene y en que puede mejorar.</p>

DECLARACIÓN Y AUTORIZACIÓN

Yo, Guillermo Enrique Mora Carrión, con C.C: # **0923654396** autor/a del trabajo de integración curricular: **Diseño de un protipo de tutor inteligente Para aprendizaje en lenguajes de programación**, previo a la obtención del título de **Ingeniero en Ciencias de la Computación** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, **8 de marzo de 2022**



f. _____

Nombre: **Guillermo Enrique Mora Carrión**

C.C: **0923654396**

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TEMA Y SUBTEMA:	Diseño de un prototipo de tutor inteligente Para aprendizaje en lenguajes de programación		
AUTOR(ES)	Mora Carrión Guillermo Enrique		
REVISOR(ES)/TUTOR(ES)	Ing. Cornejo Gómez Galo Enrique		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Ingeniería		
CARRERA:	Ingeniería en Ciencias de la Computación		
TITULO OBTENIDO:	Ingeniero en Ciencias de la Computación		
FECHA DE PUBLICACIÓN:	8 de marzo de 2022	No. DE PÁGINAS:	118
ÁREAS TEMÁTICAS:	Desarrollo web, Tutor Inteligente, Diseño del Tutor		
PALABRAS CLAVES/ KEYWORDS:	Aprendizaje, Tutor inteligente, Métodos de enseñanza, Aplicación web, sistema adaptativo de evaluación de conocimiento, aprendizaje pragmático, estudiantes, modelo matemático, lenguaje de programación.		
RESUMEN/ABSTRACT:	<p>El presente trabajo tiene como objetivo es diseñar un prototipo de sistema de tutor inteligente basado en WEB para el aprendizaje de la materia Lenguajes de programación. El prototipo ha sido diseñado como una variante de sistema tutor inteligente, dando como resultado un sistema adaptativo de evaluación de conocimiento. Se utilizó la metodología de prototipo rápido considerando cada uno de los módulos respectivos siendo estos el estudiante, dominio, tutor e interfaz.</p> <p>Se realizó una entrevista al Ing. José Erazo, con el objetivo de abordar temas relacionados con la enseñanza y aprendizaje para la materia Lenguajes de Programación. Las conclusiones de la entrevista indican que la implementación del prototipo, servirá de gran manera para los estudiantes con el fin mejorar el aprendizaje pragmático de los mismos. Cabe recalcar que el estudiante es quien decide si mejorar su aprendizaje o no.</p>		
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTOR/ES:	Teléfono: +593992200004 +15166695403	E-mail: guillermomoracarrion@gmail.com	
CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE):	Nombre: Ing. Edison José, Toala Quimi, Mgs.		
	Teléfono: +593990976776		
	E-mail: edison.toala@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			