

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

UNIVERSIDAD TECNOLÓGICA DE LA HABANA

JOSÉ ANTONIO ECHEVERRÍA

CENTRO DE REFERENCIA PARA LA EDUCACIÓN DE AVANZADA (CREA)



**MODELO DE INTEGRACIÓN DE SOFTWARE LIBRE EDUCATIVO AL PROCESO
DE ENSEÑANZA-APRENDIZAJE DE LA INTELIGENCIA ARTIFICIAL**

Tesis en Opción al Grado Científico de Doctor en Ciencias de la Educación

YUNIESKY COCA BERGOLLA

La Habana

2020

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

UNIVERSIDAD TECNOLÓGICA DE LA HABANA

JOSÉ ANTONIO ECHEVERRÍA

CENTRO DE REFERENCIA PARA LA EDUCACIÓN DE AVANZADA (CREA)



**MODELO DE INTEGRACIÓN DE SOFTWARE LIBRE EDUCATIVO AL PROCESO
DE ENSEÑANZA-APRENDIZAJE DE LA INTELIGENCIA ARTIFICIAL**

Tesis en Opción al Grado Científico de Doctor en Ciencias de la Educación

Autor: Prof. Aux. Lic. Yuniesky Coca Bergolla, M. Sc.

Tutor: Prof. Tit. Lic. María Teresa Pérez Pino, Dr. C.

Cotutor: Prof. Tit. Lic. Natalia Martínez Sánchez, Dr. C.

La Habana

2020

AGRADECIMIENTOS

A Mayra Durán, por la idea.

A María Teresa, por su guía.

A la familia, por su constante preocupación y apoyo.

Al programa doctoral y al grupo CREA-CICE, por la formación.

A la Universidad de las Ciencias Informáticas, por ser nicho de ideas y problemas por resolver.

A mis jefes, amigos, compañeros y conocidos, que contribuyeron a consolidar mi espíritu.

DEDICATORIA

A la luz, que es:

El conocimiento cuando llega.

Lucía desde que llegó.

Mónica desde que decidimos comenzar.

La Revolución, como proceso socialista de cambio.

SÍNTESIS

La presente investigación aborda la integración de software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial en la carrera Ingeniería en Ciencias Informáticas. Como resultado científico se obtiene un modelo con tres componentes generales, uno conceptual, uno estructural y uno instrumental. Este último expresado en una estrategia que constituye, junto al software libre educativo desarrollado, el aporte práctico de la investigación. El modelo fue elaborado a partir de la sistematización de los principales referentes teórico-metodológicos relativos al proceso de desarrollo de software educativo, a la integración de las tecnologías de la información y la comunicación, y a las particularidades de la enseñanza de la Inteligencia Artificial. Se obtuvieron valoraciones positivas sobre la pertinencia, consistencia teórica, aplicabilidad y aceptación del modelo, a partir de las técnicas de grupos focales, la de ladov, el método Delphi y una sistematización de experiencias. La novedad científica de la presente investigación radica en abordar la integración de un software educativo que aproveche las libertades del software libre en el proceso de enseñanza-aprendizaje de la Inteligencia Artificial, desde su diseño didáctico hasta su incorporación armoniosa al proceso de enseñanza-aprendizaje.

ÍNDICE

	Pág.
INTRODUCCIÓN	1
1. EL SOFTWARE LIBRE EDUCATIVO EN SU INTEGRACIÓN AL PROCESO DE ENSEÑANZA-APRENDIZAJE DE LA INTELIGENCIA ARTIFICIAL	12
1.1 Diseño didáctico del software libre educativo.....	12
1.2 Enseñanza-aprendizaje de la Inteligencia Artificial.	23
1.3 Integración de software libre educativo al proceso de enseñanza-aprendizaje ...	36
1.4 Operacionalización de la variable	47
1.5 Estado inicial de la variable objeto de estudio.....	50
Conclusiones del capítulo	66
2. MODELO DE INTEGRACIÓN DE SOFTWARE LIBRE EDUCATIVO AL PROCESO DE ENSEÑANZA-APRENDIZAJE DE LA INTELIGENCIA ARTIFICIAL	68
2.1 Caracterización del modelo.....	68
2.2.1 Componente conceptual.....	70
2.2.2 Componente estructural	78
2.2.3 Componente instrumental.....	87
2.2 Valoraciones sobre el modelo	94
2.2.1 Valoración por criterio de expertos	95

2.2.2 Sistematización de experiencias	98
2.2.3 Técnica de satisfacción de ladov.....	111
2.2.4 Valoración por grupos focales	113
2.2.4 Triangulación metodológica.....	114
Conclusiones del capítulo	115
CONCLUSIONES FINALES.....	117
RECOMENDACIONES	119
REFERENCIAS BIBLIOGRÁFICAS.....	i
ANEXOS	i

Índice de tablas

Tabla 1. Tipología del software utilizado en Inteligencia Artificial.....	35
Tabla 2. Escalas valorativas para la evaluación de los indicadores.....	50
Tabla 3. Unidades de estudio para el diagnóstico de la variable.	51
Tabla 4. Valores cualitativos de cada subdimensión por instrumento.....	59
Tabla 5. Influencia de las fuentes de argumentación	96
Tabla 6. Criterios valorativos por cada método o técnica aplicada	115

Índice de figuras

Figura 1. Valor de las subdimensiones de la variable de investigación	65
Figura 2. Representación gráfica del modelo.....	71
Figura 3. Componente estructural.....	79
Figura 4. Componente instrumental del modelo.....	87
Figura 5. Valores de cada pregunta en la primera ronda.	97
Figura 6. Valores de cada pregunta en la segunda ronda.	98
Figura 7. Valores de cada subdimensión en el diagnóstico y la sistematización de experiencias.....	109

INTRODUCCIÓN

La Inteligencia Artificial (IA) es una de las Ciencias de la Computación. Se encarga del estudio de los agentes que reciben percepciones de un entorno y llevan a cabo acciones sobre él, con el objetivo de desarrollar sistemas con comportamiento racional (Russell & Norvig, 2010). Resuelve problemas para los cuales no existe un algoritmo, o este es computacionalmente intratable (Bello, García, García, & Reynoso, 2002), por lo que aborda elementos avanzados de programación para computadoras.

El proceso de enseñanza-aprendizaje de la Inteligencia Artificial en la educación superior ha sido abordado tanto en el ámbito internacional (Urretavizcaya & Onaindía, 2002), (Wollowski et al., 2016), (Alvarez-Dionisi, Mitra, & Balza, 2019) como nacional (Lezcano, 2002). En todos los casos se refieren a la necesidad de utilizar medios informáticos como vía para que los estudiantes realicen actividades prácticas. Las experiencias en el mundo van dirigidas a utilizar medios donde los estudiantes programan, como parte de sus actividades de trabajo independiente, los algoritmos y técnicas abordados en las clases. Este tipo de software es diseñado para ser visualmente atractivo y con elementos metodológicos que orientan a los profesores y estudiantes para programar los aspectos que el profesor planifica.

Varias de estas investigaciones están dirigidas a la creación de videojuegos (Karpov, Sheblak, & Miikkulainen, 2008) (Wong, Zink, & Koenig, 2010), (Sosnowski, Ernsberger, Cao, & Ray, 2013), a la enseñanza de elementos básicos o introductorios de la IA (DeNero & Klein, 2010), (Renz, 2015), (Veliz, Gutierrez, & Kiekintveld, 2016), o a la enseñanza de temas específicos (Abu, 2008) (Taylor,

2010). Estas aplicaciones comparten la característica de permitir el acceso total o parcial a su código fuente. Están estrechamente relacionadas al movimiento de software libre (Free Software Foundation, 2019), donde se refieren al término software libre educativo para denominar al software que utiliza las libertades de este movimiento con algún fin educativo. Estas investigaciones muestran resultados favorables en diversos aspectos del proceso de enseñanza-aprendizaje, sin embargo, se aprecia lo siguiente:

- Estas aplicaciones informáticas tratan temáticas específicas, lo cual limita su utilización en asignaturas con distintos objetivos y contenidos, dentro de la Inteligencia Artificial.
- Insuficiente información para el diseño del software con estas características.
- Limitado aprovechamiento en el proceso de enseñanza-aprendizaje, de las libertades de estudiar, modificar y reutilizar el código fuente, aunque varios están desarrollados bajo licencias de software libre.

En Cuba el desarrollo de los procesos de enseñanza-aprendizaje y sobre todo la correcta utilización de los recursos informáticos y tecnológicos disponibles, se inserta en los documentos del Séptimo Congreso del Partido Comunista de Cuba (PCC, 2017). El lineamiento 102 se propone sostener y desarrollar los resultados alcanzados en varios campos, incluidas las tecnologías educativas y el 108 se propone avanzar gradualmente en el proceso de informatización de la sociedad de forma que se proteja la soberanía tecnológica. Directamente relacionados con la educación, el lineamiento 119 se refiere a avanzar en la informatización del sistema

de educación, desarrollar los servicios en el uso de la red telemática y la tecnología educativa de forma racional, así como la generación de contenidos digitales y audiovisuales. En las Bases del Plan Nacional de Desarrollo Económico y Social hasta el 2030 se define en el eje estratégico “Potencial humano, ciencia, tecnología e innovación” como objetivo 11 “Elevar y fortalecer la soberanía tecnológica en el desarrollo de la informática y las telecomunicaciones, así como fomentar el desarrollo de nuevas plataformas tecnológicas”. A este objetivo se tributa desde la formación de profesionales que utilicen y defiendan las ventajas de las tecnologías libres y la incorporen en su actuar diario.

Las investigaciones en Cuba relacionadas con la enseñanza de la Inteligencia Artificial se han dirigido a temas específicos. Para la enseñanza de la programación lógica se proponen mapas conceptuales (Ríos, Lezcano, & Aljadis, 2008) y un sistema de apoyo a su aprendizaje (Lezcano & Valdés, 1998a). Para el aprendizaje de sistemas expertos se presentó el ambiente integrado SESE (Lezcano & Valdés, 1998b) y el software TechShell (García & Lezcano, 2011). Específicamente en la Universidad de las Ciencias Informáticas se desarrolló un sistema en Prolog para el tema de sistemas basados en reglas (Cruz, 2015). Estas aplicaciones informáticas no están diseñadas para que los estudiantes interactúen con su código fuente y comparten con los estudiados a nivel internacional, que van dirigidos a temáticas específicas.

En la carrera Ingeniería en Ciencias Informáticas (ICI) de la Universidad de las Ciencias Informáticas (UCI) se imparte la Inteligencia Artificial desde el año 2007, a partir del curso 2011-2012 se reestructura el plan de estudio y se incorpora una

segunda asignatura para garantizar una mayor profundización en los contenidos impartidos de esta temática (Colectivo Carrera ICI, 2015).

En el curso 2015-2016 la carrera ICI se desarrolla en cinco facultades. Las asignaturas Inteligencia Artificial I e Inteligencia Artificial II las imparten 24 profesores con diversas categorías y experiencia profesional (Colectivo Disciplina IA, 2015). Los estudiantes se vinculan a centros de desarrollo donde llevan a cabo su disciplina principal integradora Práctica Profesional (PP) durante todo el año. Uno de los principios fundacionales de la UCI fue la vinculación entre el proceso docente y el proceso de desarrollo de software. Estas características de la carrera refuerzan la necesidad de un proceso de enseñanza-aprendizaje flexible, donde se tomen en cuenta las particularidades de cada grupo, pero a la vez se mantenga homogeneidad en el proceso.

El modelo del profesional de Ingeniería en Ciencias Informáticas para el Plan E (Colectivo Carrera ICI, 2019), tiene como objeto de la profesión el proceso de informatización de las organizaciones y define tres objetivos. El primero relacionado con gestionar procesos asociados al tratamiento computacional de la información, para tomar decisiones basadas en datos y generar conocimiento. El segundo dirigido a desarrollar, adoptar y mantener sistemas y servicios informáticos. El tercero enfocado en gestionar, desde una perspectiva de soberanía tecnológica y ciberseguridad, la infraestructura computacional que soporta los sistemas y servicios informáticos.

La Inteligencia Artificial contribuye a los tres objetivos. Al primero de ellos tributa a partir de abordar métodos para la toma de decisiones y el tratamiento computacional

de la información. Al segundo mediante el desarrollo, utilización y mantenimiento de sistemas y servicios informáticos, aportando métodos que contribuyan a la racionalización y optimización de procesos y recursos. En cuanto al tercero, se deben dirigir los esfuerzos a crear consciencia de la importancia de la soberanía tecnológica.

Para lograr una contribución efectiva a estos objetivos es necesario una transformación del proceso de enseñanza-aprendizaje de la Inteligencia Artificial. En esta transformación, los medios de enseñanza-aprendizaje deben jugar un papel fundamental. Se requiere desarrollar en los estudiantes habilidades para el análisis, diseño, implementación, utilización y mantenimiento de sistemas y servicios informáticos, con enfoque de soberanía tecnológica. Para el desarrollo de habilidades como la implementación y mantenimiento del software, es necesaria la interacción del estudiante con el código fuente, tanto para estudiarlo y modificarlo como para crear nuevo.

A partir del análisis de informes finales de las asignaturas y la disciplina y de actas de actividades metodológicas en el curso 2015-2016 se evidencia:

- Ausencia de un software educativo donde los estudiantes puedan interactuar con el código fuente para realizar tareas del PEA.
- Insatisfacciones de los estudiantes en cuanto a la relación entre las tareas realizadas con el software y sus intereses profesionales.
- Escasa vinculación entre las tareas a realizar por los estudiantes con el software y las tareas que realizan en la disciplina principal integradora PP.

Relacionado con las tecnologías educativas se han llevado a cabo varias investigaciones encaminadas a crear modelos (Cabrera, 2008), (Pola, 2014), (Pérez, 2015), (Area-Moreira, Hernández-Rivero, & Sosa-Alonso, 2016), concepciones (Torres, 2008), (Rodríguez, 2010), (García, 2014), estrategias (Gil, 2010), (Álvarez, 2014), (Lombillo, Nambalo, Torres, & Pérez, 2018) o metodologías (Fernández, 2012), (Ramos, 2016). Los principales aspectos abordados son la creación y gestión de recursos educativos, la superación de los profesores y el diseño de software educativo. A partir del análisis de estas investigaciones se ha podido apreciar:

- Limitado tratamiento teórico y metodológico al aprovechamiento de las libertades del software libre en el proceso de enseñanza-aprendizaje.
- Insuficiente tratamiento teórico y metodológico a las relaciones entre el diseño didáctico del software educativo y su incorporación a un proceso de enseñanza-aprendizaje.
- Carencias teóricas en cuanto a las características de un software educativo que aproveche las libertades del software libre, para ser incorporado a un proceso de enseñanza-aprendizaje de la Inteligencia Artificial.

Todo lo analizado hasta el momento evidencia la importancia que tiene la integración de un software educativo que aproveche las libertades del software libre en el proceso de enseñanza-aprendizaje de la Inteligencia Artificial. Sin embargo, existen carencias teóricas y metodológicas para la integración del software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial en la carrera Ingeniería en Ciencias Informáticas. Esta contradicción fundamental conduce al siguiente

problema científico: ¿Cómo contribuir a la integración de software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial en la carrera Ingeniería en Ciencias Informáticas? Para lo cual se propone como **objeto de estudio:** La integración de software educativo al proceso de enseñanza-aprendizaje, quedando enmarcado el **campo de acción** en la integración de software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial en la carrera Ingeniería en Ciencias Informáticas.

Para dar cumplimiento al problema se define como **objetivo de la investigación:** Elaborar un modelo de integración de software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial en la carrera Ingeniería en Ciencias Informáticas que aborde desde el diseño didáctico del software, hasta su incorporación armoniosa al proceso de enseñanza-aprendizaje.

Dirigirán la investigación las siguientes **Preguntas científicas:**

1. ¿Cuáles son los principales referentes, antecedentes y fundamentos teóricos y metodológicos para la integración del software educativo, basado en software libre, al proceso de enseñanza-aprendizaje de la Inteligencia Artificial?
2. ¿Cuál es el estado inicial de la integración del software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial en la carrera Ingeniería en Ciencias Informáticas?
3. ¿Cuáles son los elementos que caracterizan el modelo de integración de software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial en la carrera Ingeniería en Ciencias Informáticas?

4. ¿Cómo se valora el modelo elaborado para la integración del software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial en la carrera Ingeniería en Ciencias Informáticas?

Para dar respuesta a las preguntas se definen como **tareas de investigación**:

1. Sistematización de los principales referentes, antecedentes y fundamentos teóricos y metodológicos para la integración del software educativo, basado en software libre, al proceso de enseñanza-aprendizaje de la Inteligencia Artificial.
2. Caracterización del estado inicial de la integración del software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial en la carrera Ingeniería en Ciencias Informáticas.
3. Determinación de los componentes y relaciones que se establecen en la integración de software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial.
4. Valoración de la contribución del modelo a la integración del software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial en la carrera Ingeniería en Ciencias Informáticas.

Esta investigación tiene como método general el dialéctico-materialista, a partir del cual se realizaron indagaciones teóricas y empíricas. Para las teóricas, se utilizaron los métodos siguientes:

Histórico-lógico: Se utilizó para el estudio de la integración del software educativo al proceso de enseñanza-aprendizaje en su desarrollo y contradicciones. El proceso de enseñanza-aprendizaje de la Inteligencia Artificial, con énfasis en los medios

utilizados. Como resultado principal se obtuvo una periodización de la enseñanza de la Inteligencia Artificial en currículos afines a la informática.

Analítico-sintético: Se utilizó en el estudio de los diferentes puntos de vista en la integración del software educativo al proceso de enseñanza-aprendizaje, más específicamente el software libre al proceso de enseñanza-aprendizaje de la Inteligencia Artificial.

Enfoque sistémico-estructural-funcional: Permitió diseñar el modelo tomando en cuenta la estructura, funciones y exigencias de cada uno de los elementos que lo integran y su relación con el objeto.

Modelación: Se utilizó para estudiar mediante abstracciones teóricas el software libre educativo y su integración al proceso de enseñanza aprendizaje de la Inteligencia Artificial. Se materializa tomando en cuenta sus particularidades y explicando la relación entre sus componentes.

En cuanto a los empíricos:

Observación: Se utilizó en todo el proceso, desde el diagnóstico del problema inicial hasta la aplicación del modelo.

Análisis documental: Para fundamentar el problema en el análisis empírico llevado a cabo, se analizaron informes y documentos rectores del proceso de enseñanza-aprendizaje en la carrera Ingeniería en Ciencias Informáticas.

Encuesta: Se utilizó la encuesta para obtener información de los estudiantes acerca de la utilización de software en las asignaturas de Inteligencia Artificial.

Entrevista: Se utilizaron entrevistas a profesores y directivos para diagnosticar el estado inicial del problema.

Sistematización de experiencias: Para extraer enseñanzas de una experiencia de diseño didáctico e incorporación de un software libre educativo a una asignatura de Inteligencia Artificial. También permitió comparar los resultados alcanzados en la experiencia con los resultados al inicio de la investigación.

Criterio de expertos: Se utilizó el método de base subjetiva Delphi para obtener criterios de expertos sobre el modelo presentado.

Test de satisfacción de ladov: Con el cual se obtuvo criterios de satisfacción de los estudiantes participantes en la experiencia con el software libre educativo.

Como métodos **estadístico–matemáticos** se utilizó la estadística descriptiva, para el procesamiento de los datos obtenidos en la aplicación de los métodos y técnicas empíricos.

Se utilizó la **triangulación metodológica** de los diferentes métodos utilizados, para diagnosticar el problema y para obtener criterios valorativos sobre la pertinencia, consistencia teórica, aceptación y aplicabilidad del modelo.

La **novedad científica** de la presente investigación radica en abordar la integración de un software educativo que aproveche las libertades del software libre en el proceso de enseñanza-aprendizaje de la Inteligencia Artificial, desde su diseño didáctico hasta su incorporación armoniosa al proceso de enseñanza-aprendizaje.

La **contribución de la presente investigación** se enmarca en las Ciencias de la Educación, específicamente a la Tecnología Educativa. En relación a lo teórico la contribución se concreta en la identificación de las relaciones esenciales que se establecen entre el diseño didáctico de un software libre educativo y su incorporación armoniosa al proceso de enseñanza-aprendizaje de la Inteligencia Artificial. Se

reconoce como resultado teórico el sistema de principios que guían el aprovechamiento didáctico de las libertades del software libre de forma contextualizada y hacia la transformación del proceso de enseñanza-aprendizaje.

La contribución práctica se concreta en:

- Estrategia didáctica para la aplicación del modelo de integración de software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial en la carrera Ingeniería en Ciencias Informáticas.
- Desarrollo de un software libre educativo basado en el paradigma de programación para agentes, para su integración al proceso de enseñanza-aprendizaje de la asignatura Inteligencia Artificial II de la carrera Ingeniería en Ciencias Informáticas.

El informe de investigación está estructurado en una introducción, dos capítulos, conclusiones, recomendaciones, referencias bibliográficas y los anexos que complementan la investigación. En el primer capítulo se presentan los fundamentos teóricos y metodológicos del objeto de estudio y el campo de acción y se expone el análisis empírico llevado a cabo para corroborar el problema de la investigación. En el segundo capítulo se describe el modelo con sus componentes y relaciones y se analizan los resultados obtenidos por varios métodos, para obtener valoraciones sobre el modelo.

1. EL SOFTWARE LIBRE EDUCATIVO EN SU INTEGRACIÓN AL PROCESO DE ENSEÑANZA-APRENDIZAJE DE LA INTELIGENCIA ARTIFICIAL

En el presente capítulo se presenta la sistematización de los fundamentos del software educativo y su proceso de desarrollo, incluyendo el diseño didáctico. Se realiza una periodización del proceso de enseñanza-aprendizaje de la Inteligencia Artificial y se aborda la integración de tecnologías de la información y la comunicación al proceso de enseñanza-aprendizaje. A partir de la definición operacional de la variable objeto de estudio, se lleva a cabo el proceso de operacionalización y la evaluación de su estado inicial en la carrera Ingeniería en Ciencias Informáticas.

1.1 Diseño didáctico del software libre educativo

El software educativo (SE) forma parte del conjunto de tecnologías de la información y la comunicación (TIC) que han irrumpido en los procesos de enseñanza-aprendizaje (PEA) con extraordinaria fuerza. El diseño, producción, evaluación, uso y efectos del SE en el PEA, constituyen tareas de la Informática educativa (García, 2014). En ella confluyen diferentes disciplinas que se enmarcan dentro de las Ciencias de la Educación.

Las primeras ideas del uso del software y las computadoras en la educación, suponían la futura sustitución del maestro (Mevarech & Light, 1992). Sin embargo,

con el aumento de su presencia en las escuelas y el desarrollo de sistemas inteligentes, se han ido perfeccionando los métodos de enseñanza, con lo cual se ha reforzado la participación del profesor dentro del proceso, no como centro del mismo sino como mediador.

Los orígenes del SE datan de la enseñanza programada de los años 60, pero cobran una mayor connotación con el desarrollo tecnológico de los años 80. El término software del inglés y ya aceptado en la Real Academia Española (RAE, 2020), se refiere al conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora. En la presente investigación se utiliza, en algunas ocasiones, el término aplicación informática, con el objetivo de esclarecer la redacción en el idioma español.

En cuanto a lo educativo existen diversas posiciones teóricas, pero coinciden en su utilidad dentro del PEA. Se pueden apreciar dos tendencias generales del SE (Ramos, 2016): Una amplia que abarca las aplicaciones informáticas que se diseñan para alcanzar diversos propósitos en el ámbito educativo (Galvis, 1988); y otra, que se ajusta solo al contexto del PEA. Acerca de esta última también se encuentran posiciones divergentes, algunos autores tienen en cuenta su función en el PEA (Gros, 1997) (Coloma, 2008), (Ramos, 2016) por lo que consideran que un SE es cualquier programa informático utilizado en dicho proceso. Mientras, otros plantean que una aplicación informática se considera SE, si ha sido creada con la expresa finalidad o intencionalidad didáctica de apoyar el PEA (Cataldi, 2000) (Rodríguez, 2010), (Área, 2010) (Aguilar, De la Vega, Lugo y Zarco, 2014). Esta posición es asumida en la presente investigación.

Se reconocen, en la bibliografía consultada, varias tipologías de SE. Para clasificarlos siguen diferentes criterios, entre los que destacan: los contenidos que trata, los destinatarios o los objetivos que pretende facilitar. Una de las clasificaciones reconocidas que toma en cuenta su relación directa con el tipo de aprendizaje los identifica como: tutoriales, simuladores, entornos de programación y herramientas de autor (Cataldi, 2000).

Para Cataldi (2000) los tutoriales dirigen el aprendizaje de los alumnos mediante una teoría subyacente conductista de la enseñanza. Los simuladores facilitan el aprendizaje por descubrimiento, mientras las herramientas de autor permiten a los profesores crear programas del tipo tutoriales. Por su parte, los entornos de programación facilitan la adquisición del conocimiento y el aprendizaje. Estos últimos están dirigidos, en lo fundamental, a desarrollar habilidades en la programación, aunque no siempre utilizando directamente el código fuente del mismo.

Rodríguez (2010) presenta una concepción didáctica del SE, como instrumento mediador del aprendizaje. Tiene sus fundamentos en el materialismo dialéctico, en el enfoque histórico-cultural y en los principios y dimensiones de la concepción del aprendizaje desarrollador. Establece un marco conceptual, los factores críticos para su desarrollo, una propuesta metodológica general y principios que deben guiar la mediación didáctica del SE en el PEA. Este autor incluye como factores de mayor valor estratégico para el desarrollo del SE la pertinencia didáctica, la socialización de experiencias, el mantenimiento y actualización del SE, la estrategia didáctica de utilización del SE y su contribución al aprendizaje.

Los principios del SE como instrumento mediador para un aprendizaje desarrollador presentados por Rodríguez (2010) son:

- Integración al proceso de enseñanza-aprendizaje bajo la dirección del profesor.
- Diseño didáctico orientado hacia una mediación desarrolladora.
- Solución intencional de los problemas de aprendizaje.
- Perfeccionamiento didáctico sistemático y participativo.
- Actividades didácticas centradas en unidades de enseñanza-aprendizaje.

Estos principios guían en el desarrollo y utilización de SE desde el enfoque histórico-cultural en el PEA, por lo que se asumen como referentes en la presente investigación.

Por su parte Ramos (2016) plantea algunas exigencias que debe cumplir el SE para que sea eficaz en el contexto donde se pretende incluir:

1. Poseer independencia funcional.
2. Estar avalado por una entidad capaz de autorizar su empleo en el proceso de enseñanza-aprendizaje.
3. Desarrollado por un equipo multidisciplinario que garantice la calidad requerida desde el diseño hasta la evaluación.
4. Poseer una estrategia didáctica que se corresponda con el currículo de la enseñanza al que va dirigido.

Los puntos uno y cuatro se reconocen como importantes, en tanto definen la esencia de un software independiente la primera y su finalidad didáctica la segunda. Sin embargo los puntos dos y tres, que igualmente se defienden por Rodríguez (2010),

no son compartidos en esta investigación. Existen aplicaciones informáticas que, en determinados contextos, pueden considerarse SE y no ser desarrolladas por equipos multidisciplinarios, ni ser avaladas por entidades más allá del propio profesor que la utiliza.

En la presente investigación se defiende la posibilidad de que los profesores desarrollen SE para sus asignaturas. En el caso de profesores con formación en programación de computadoras, como ocurre con los de Inteligencia Artificial, pueden construir sus propios medios de enseñanza, incluido SE. Implicar a los profesores en el desarrollo de sus propios medios ha sido investigado y defendido como positivo en otras investigaciones (Álvarez, 2014).

Para su desarrollo pueden apoyarse, incluso, en los estudiantes, como parte del proceso de enseñanza-aprendizaje. La complejidad del software no tiene que ser alta para considerarse educativo, siempre que esté diseñado para cumplir determinados objetivos en el proceso de enseñanza-aprendizaje.

Los autores consultados coinciden en la importancia del diseño didáctico del SE. Diseñar un software para ser utilizado como medio de enseñanza-aprendizaje y aprovechar sus potencialidades peculiares, constituye uno de los retos actuales de las ciencias pedagógicas (García, 2014). Este se ha concebido de diversas formas, pero siempre desde el propio proceso de desarrollo computacional del software.

Para llevar a cabo el proceso de desarrollo del SE, se han propuesto modificaciones a las metodologías tradicionales de la ingeniería de software (Galvis, 1992) (Marqués, 1995), (Cataldi, Lage, Pessacq, & García, 1997) (Gómez, Gálvis, & Mariño, 1998), (Cataldi, 2000), (Ramos, 2016). En todos los casos se hace énfasis

en el diseño de las situaciones de enseñanza-aprendizaje que se pretenden llevar a cabo con el software. Se considera en la presente investigación que las metodologías tradicionales aportan los elementos generales para el desarrollo de cualquier software, incluido el SE. Pero sí se deben especificar bien los requisitos didácticos que incluirá el SE.

En sentido general, el proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software (Jacobson, Booch, & Rumbaugh, 2000). En el caso particular del SE, su desarrollo requiere precisar las tareas que permiten convertir los requerimientos didácticos, en requerimientos propios de la ingeniería de software convencional.

Las metodologías de desarrollo de software incluyen una etapa de diseño computacional, la cual no debe confundirse con el diseño didáctico del software. El diseño en el ámbito educativo, se asocia a la proyección anticipada del proceso pedagógico (García, 2014). Se reconoce en la bibliografía consultada el papel protagónico del diseño didáctico y del guion de contenidos como documento que lo materializa (Cabero, Morales, Barroso, Roman, & Romero, 2004), (Santos, Cenich, & Miranda, 2004), (Área, 2009).

El diseño didáctico es un proceso sistémico de modelación de las situaciones de enseñanza–aprendizaje (Del Toro, 2006), (García, 2014) que de una forma u otra se plasman en el medio que se va a desarrollar. En el caso del SE, García (2014) propone generar un documento donde se formaliza, materializa, consolida y concreta el diseño didáctico del SE. Este documento, reconocido como guion de contenidos,

forma parte del conjunto de documentos utilizados en el proceso de desarrollo computacional del SE.

En la presente investigación se considera al guion de contenido como un documento dinámico, importante no solo a la hora de desarrollar el software. El guion de contenido reunirá elementos de interés a incorporar al SE y orientaciones generales de cómo utilizarlo en el PEA. De esta forma se considerará como una herramienta de trabajo del profesor, tanto en el proceso de desarrollo del SE como en su incorporación al PEA. Cuando el software esté desarrollado, el guion de contenido se ajustará a posibles nuevas características de la asignatura y a modificaciones futuras del software educativo.

El diseño didáctico de los medios de enseñanza y específicamente del SE, tradicionalmente se ha abordado desde tres dimensiones: semántica, sintáctica y práctica (Del Toro, 2006), (Área, 2010) (García, 2014).

La dimensión semántica se refiere a los contenidos, informaciones y mensajes del mismo. La sintáctica incluye el modo en que se estructura, organiza y simboliza la información. Mientras la práctica hace referencia al cómo y para qué será empleado el medio (Área, 2010).

En Ciudad (2012) se analizan estas dimensiones y se exponen criterios para proponer un nuevo conjunto de dimensiones para evaluar los medios sustentados en las TIC. Este autor consideró necesario sustituir la dimensión sintáctica por las dimensiones: espacial y tecnológica y ampliar el sistema con dos nuevas dimensiones: la de gestión y la personal. De esta forma propone un nuevo sistema de dimensiones: semántica, tecnológica, espacial, personal, de gestión y práctica.

La dimensión semántica establece las relaciones entre los componentes del PEA y resultados posibles con el medio sustentado en las TIC. La dimensión práctica refiere a la estrategia didáctica que guiará el PEA al utilizar el medio sustentado en las TIC. La dimensión tecnológica, establece el sistema de TIC y las relaciones que estas tendrán. La dimensión espacial se refiere a los contextos sustentados en las TIC a través de los cuales se ejecutará el PEA. La dimensión de gestión aborda las formas de administrar tecnológicamente el sistema de TIC que sustentan el medio. Por último la dimensión personal establece qué participantes del PEA utilizarán el medio sustentado en las TIC, los roles que asumirán y sus funciones; así como las relaciones entre estos.

Para la presente investigación se consideró pertinente cambiar el nombre de la dimensión semántica por didáctica, en tanto se analizan en ella específicamente los componentes del PEA. Además, se elimina la dimensión personal, no por quitarle importancia, sino porque se abordada como parte de las demás dimensiones en el diseño didáctico y se profundiza en las relaciones entre los sujetos dentro del PEA.

Las investigaciones analizadas sobre el SE y su proceso de desarrollo, aportan elementos de interés para la presente investigación, mas carecen de un análisis acerca de las ventajas del software libre en el proceso de desarrollo y sobre todo el aprovechamiento didáctico de sus libertades. Por otro lado, no se ha encontrado un tratamiento a las particularidades del SE para ser utilizado en un PEA de la Inteligencia Artificial o asignaturas de programación, donde no solo la interfaz del software tenga la responsabilidad de la interacción con el estudiante. Para el SE basado en software libre y para estudiantes relacionados con carreras de informática,

el propio código fuente asume una fuerte responsabilidad en la interacción con el estudiante.

El movimiento de “conocimiento abierto” tiene gran alcance e impacto en diversas esferas de la sociedad, su presencia en la educación se manifiesta con más claridad en la tecnología educativa, donde organizaciones internacionales como la UNESCO han contribuido de manera significativa a su expansión, con la introducción del concepto de Recursos Educativos Abiertos (REA).

La declaración de París (UNESCO, 2012) fue adoptada oficialmente en el Congreso Mundial de Recursos Educativos Abiertos de 2012; expresamente recomienda a los estados, en la medida de sus posibilidades y competencias, acciones a llevar adelante con respecto a los REA, entre las que están:

- Fomentar el conocimiento y el uso de los recursos educativos abiertos.
- Promover el conocimiento y la utilización de licencias abiertas.
- Alentar la investigación sobre los recursos educativos abiertos.

Varias son las investigaciones llevadas a cabo con el objetivo de potenciar la creación de REA en las instituciones (Cabrera, 2008), (Trillo, 2012), (Torres, Zangla, & Chiarani, 2016). La visión de crear los REA, se ha limitado, en su gran mayoría, a la construcción de materiales que componen los cursos en línea o a objetos de aprendizaje como elemento más específico que incluye un conjunto de criterios que enriquecen un objeto de contenido. Sin embargo, la creación de SE, visto como recurso educativo abierto, no ha sido lo suficientemente tratado desde el punto de vista teórico, ni se ha fomentado desde el punto de vista práctico.

Derivado del movimiento “conocimiento abierto” se encuentra el concepto de “código abierto”, que se refiere a los programas informáticos que permiten el acceso al código con que fueron desarrollados.

El término ‘código abierto’ alcanzó notoriedad en el mundo del desarrollo de software a partir del año 1983 bajo la denominación de Movimiento del Software Libre y establece cuatro libertades:

- La libertad para ejecutar el programa, con cualquier objetivo.
- La libertad para estudiar cómo funciona el programa y adaptarlo a las propias necesidades.
- La libertad para redistribuir copias a fin de ayudar al prójimo.
- La libertad para mejorar el programa y divulgar las mejoras ante el público, para que toda la comunidad se beneficie.

La mayor potencialidad del software libre no es su distribución, ni su seguridad, sino la posibilidad de modificarlo, mejorarlo de forma colaborativa y seguir distribuyéndolo bajo los mismos términos para la comunidad. En el mundo de la educación esta cualidad reviste una importancia adicional, y es la posibilidad de utilizarla en la formación de los estudiantes en asignaturas relacionadas con la programación, es decir, con la construcción de software.

El software libre contribuye al progreso de los estudiantes relacionados con la programación, a partir de leer y comprender el código de programas reales. Aprenden a escribir código para programas grandes, escribiendo modificaciones para

programas que ya existen. Únicamente el software libre ofrece esta posibilidad (Stallman, 2020).

La mayor importancia del SE basado en software libre radica en la posibilidad de adaptarse a las condiciones específicas de una clase. Para carreras afines a la informática, se pueden utilizar como medios donde los estudiantes estudian, modifican, adicionan o reutilizan el código fuente como parte de sus tareas prácticas. A partir de los elementos presentados acerca de las características del SE y la importancia del aprovechamiento de las libertades del movimiento de software libre en un PEA, se define el término **software libre educativo** como toda aplicación informática diseñada con la intencionalidad de aprovechar las libertades del movimiento de software libre, en su función de medio para un proceso de enseñanza-aprendizaje específico.

Existen aplicaciones, basadas en software libre, que son utilizadas en procesos de enseñanza-aprendizaje, incluso pueden ser consideradas SE, sin embargo, no fueron diseñadas con el fin de utilizar las libertades de dicho movimiento en la educación, por tanto salen del alcance de la definición presentada.

A partir del análisis realizado, el **diseño didáctico** del software libre educativo se define como un proceso sistémico de modelación de las situaciones de enseñanza-aprendizaje que se desarrollarán con el software libre educativo, a partir de las interrelaciones entre las dimensiones semántica, práctica, tecnológica, espacial y de gestión. Este se materializará en un guion de contenido, útil tanto para el desarrollo computacional del software, como para su incorporación al PEA para el cual fue diseñado, en esta investigación, el PEA de la Inteligencia Artificial.

1.2 Enseñanza-aprendizaje de la Inteligencia Artificial.

La enseñanza-aprendizaje de la Inteligencia Artificial (IA) en forma coherente y productiva no constituye una tarea simple, ni perfectamente definida. Su complejidad necesita de recursos que ayuden a hacerla más amena y productiva (Lezcano, 2002). El estudio de la IA prepara al estudiante para determinar cuándo es apropiado utilizarla en un problema específico, identificar la representación apropiada y el mecanismo de razonamiento, implementarlo y evaluarlo (ACM-IEEE, 2013).

Los fundamentos de la IA datan de los silogismos de Aristóteles, presentados en el siglo III antes de nuestra era. Sin embargo, se considera el primer trabajo de Inteligencia Artificial la propuesta de redes neuronales artificiales diseñadas por McCulloch y Pitts en 1943 (Russell & Norvig, 2010). Su consolidación como campo de estudio parte de un trabajo de Turing (1950), donde plantea los elementos a tomar en cuenta para determinar si un programa de computadoras es inteligente. En él se introduce el aprendizaje automático, los algoritmos genéticos y el aprendizaje por refuerzo.

El término Inteligencia Artificial se propone por primera vez en una conferencia en Dartmouth en 1956, donde se analizan las características que la convierten en campo independiente, con problemas específicos que atender dentro de la rama de la informática. Los siguientes años son de un gran auge y se realizan previsiones sobre lo que sería esta nueva ciencia en los siguientes 15 años. A partir de ahí comienzan a aparecer investigaciones relevantes sobre diversos campos dentro de la IA (Russell & Norvig, 2010).

En este contexto la Association for Computing Machinery (ACM), reconocida como la primera sociedad científica y educativa para educar acerca de la Computación (ACM, 2020), incluye en su primera propuesta para currículos de Ciencias de la Computación (Atchison et al., 1968) un curso electivo de IA. Las temáticas que aborda se reconocen como avanzadas para ese entonces e incluyen elementos teóricos básicos y de programación heurística.

En el año 1968 se reconocen los inicios de las investigaciones en IA en Cuba y en 1969 salen a la luz las primeras publicaciones para difundir conceptos de las ciencias de la computación y específicamente de la IA (MES-UH, 2008).

Desde finales de la década del 60 y toda la década del 70 fue de crisis en la IA. Las previsiones no fueron cumplidas, por lo que el financiamiento de los gobiernos para las investigaciones disminuyó significativamente. Muchas investigaciones se dirigieron a demostrar la incapacidad de los algoritmos y métodos, empleados hasta ese momento, para resolver problemas verdaderamente complejos. También por esa fecha aparecen los primeros sistemas basados en el conocimiento, los que requirieron nuevas formas de representar el conocimiento y surgen lenguajes declarativos como el Lisp y el Prolog.

A raíz de esta situación de crisis en la IA, en la próxima propuesta de la ACM (Austing, Barnes, Bonnette, Engel, & Stokes, 1979), la IA sigue siendo un curso electivo con elementos básicos para la profundización de los estudiantes. Se incorporan como elementos de mayor importancia, la representación del conocimiento y se abordan las estrategias de búsqueda como parte de esta

representación y como parte de estructuras de control. Además, se incorporan las temáticas de comunicación, percepción y ejemplos de aplicaciones específicas.

En Cuba se inicia, en el año 1977 con el Plan A, la elaboración de planes de estudio para las carreras universitarias. En 1981 se propone una primera modificación denominada Plan B, en la carrera Cibernética-Matemática se incluye el estudio del Prolog y conceptos y métodos generales de la IA. En la carrera Ingeniería Informática se introduce la lógica matemática y el Prolog.

En la década del 80 la IA se convierte en una industria (Russell & Norvig, 2010) gracias al auge de los sistemas expertos. Se inician proyectos ambiciosos en Japón para obtener la quinta generación de computadoras basadas en la IA. En Estados Unidos y el Reino Unido se retoman proyectos y se crean empresas con investigaciones dirigidas a mejorar la relación hombre-máquina con un fuerte componente de la IA.

Enmarcado en este auge, en 1986 se elabora un nuevo conjunto de planes de estudio en Cuba (Plan C). En la carrera Ciencias de la Computación se profundiza en los elementos generales de la IA y se incluye la simulación y cursos optativos con temáticas específicas como las redes neuronales y el reconocimiento de patrones. En Ingeniería Informática, se enriquecen los contenidos con la asignatura Programación Descriptiva y Sistemas de Bases de Conocimientos I y II.

También condicionado por el auge de la IA, en la propuesta de currículo de la ACM en 1991 (ACM-IEEE-CS, 1991) se incluye por primera vez la IA como parte del currículo básico y se incluyen elementos de historia, aplicaciones de la IA, problemas de estados y estrategias de búsqueda. A raíz de esta propuesta comienzan a

aparecer en el mundo experiencias y propuestas de organización de los cursos de IA (Dillon, 1993).

Los proyectos de inicios de los 80 fracasaron, varias empresas quiebran, lo cual condujo a una nueva crisis de la IA a inicios de los 90. Esta crisis no demoró mucho, en esa misma década resurgen nuevas investigaciones y resultados constantes en varios campos de la IA. Las investigaciones se concentraron en un núcleo más definido y en resolver problemas más específicos del mundo real (Russell & Norvig, 2010).

En 1998 se realiza una revisión de los planes de estudio en Cuba, dando paso a los planes C mejorados. En la carrera Ingeniería Informática se reduce el tiempo a la programación lógica y se renombran las asignaturas a Sistemas Informáticos Inteligentes I y Sistemas Informáticos Inteligentes II.

En 2001 aparece una nueva propuesta de la ACM para currículos de Ciencias de la Computación (ACM-IEEE, 2001). Las principales diferencias con respecto a la propuesta anterior de 1991 fueron (Urretavizcaya y Onaindía, 2002):

- El incremento de la importancia en aspectos como la satisfacción de restricciones y la representación del conocimiento.
- Desaparece de la propuesta del 2001 la utilización de los lenguajes LISP y Prolog como posibles lenguajes a utilizar en tareas prácticas.
- En los cursos complementarios y avanzados proponen los temas de razonamiento temporal, espacial, incertidumbre y conjuntos borrosos; desechando la representación basada en reglas.

- Aparece el tema de agentes, el cual ha venido a desempeñar un papel preponderante en la enseñanza de esta ciencia.

Urretavizcaya y Onaindía (2002) refieren a un vertiginoso desarrollo del currículo en esta etapa y realizan un recuento sobre las ponencias presentadas en los eventos JENUI'97 y CAEPIA-TTIA '2001 (Bahamonde & Otero, 2001), en las cuales se evidencia la gran cantidad de temas tratados y las diversas formas de enseñanza utilizadas.

Algunos de los problemas planteados en la enseñanza de la IA en ese período fueron (Fasli, 2005):

1. Gran diversidad en los temas impartidos como básicos, por la falta de consenso entre los especialistas.
2. Fuerte influencia de las experiencias prácticas en investigaciones y la especialización para decidir los contenidos de los cursos.
3. Falta de metodologías y herramientas que puedan ser utilizadas para la enseñanza de estos temas.

También en el ámbito cubano se plantean elementos que han incidido en la enseñanza de esta ciencia (Lezcano, 2002):

- Imprecisión en los objetivos de enseñanza: Contradicción entre el deseo de que los estudiante aprendan un conjunto básico de conceptos o desarrollarles determinadas habilidades.
- La abstracción de los conceptos: Lo cual implica que sean difíciles de comprender por el estudiante.

- Imprecisión de las herramientas de medición del aprendizaje: No se logra saber qué saben los estudiantes, debido a que las herramientas tradicionales (exámenes, proyectos, etc.) no resultan suficientes para indicarlo.

Estos elementos se mantienen o incluso se agravan con el desarrollo de la propia ciencia. Cada día aparecen nuevos algoritmos que incorporan más niveles de abstracción, se hace más complejo evaluar el aprendizaje de los estudiantes y se mantiene el dilema de conceptos contra habilidades, teoría contra práctica.

Otro de los problemas que ha enfrentado la enseñanza de la IA es la contradicción entre el reducido tiempo que se le asigna en los planes de estudio y la diversidad y complejidad de los temas que trata (Lezcano, 2002).

En el 2002 surge la Universidad de las Ciencias Informáticas (UCI) y con ella la carrera Ingeniería en Ciencias Informáticas (ICI). Su plan de estudio incluye una asignatura de IA que trata elementos básicos de búsquedas, representación del conocimiento, tratamiento de incertidumbre y fundamentos de los sistemas expertos.

En este período aparecen por primera vez en el mundo, experiencias en la utilización de medios donde los estudiantes acceden al código fuente del software para incorporar los métodos y técnicas de IA abordados en las clases.

Kummeneje y Verhagen (2005) presentan las experiencias de 10 años de impartir programación basada en agentes. Realizan una comparación entre diferentes aplicaciones informáticas disponibles. Proponen a los profesores de esta rama, desarrollar su propia aplicación para llevar a cabo las prácticas de laboratorio.

McCluskey y Simpson (2005) critican la utilización de más de una aplicación en un mismo curso y proponen la creación de uno que integre todos los contenidos.

Varios autores han defendido la importancia de utilizar juegos como medios en la enseñanza de la IA (El-Nasr & Smith, 2006), (Wang y Wu, 2009). Sin embargo, autores como Bellas y Alonso (2007) defienden la importancia de abordar problemas reales o basados en la realidad, minimizando la utilización de códigos complejos.

Un análisis de algunos de los programas de asignaturas de IA de diversas carreras de Ciencia de la Computación en el mundo (ACM-IEEE, 2013) arrojó coincidencias en cuanto a utilizar medios con los que el estudiante realiza tareas de programación de los algoritmos estudiados en clases. Este tipo de tareas también es común en la carrera Ciencia de la Computación en Cuba. Otro elemento de interés fue la diversidad en los tipos de software utilizados y en un solo caso es diseñado por el profesor y sus estudiantes con el fin de ser utilizado en clases.

En la presente investigación se defiende la utilización de los medios desde un punto de vista más ingenieril, donde los estudiantes resuelvan problemas computacionales utilizando algoritmos ya programados. Esta programación puede realizarse por el propio profesor en el SLE o mediante la inclusión en él de bibliotecas de algoritmos reconocidas internacionalmente.

Desde finales de la década de los 90 en Cuba se trabaja en utilizar medios para la enseñanza del Prolog. Lezcano y Valdés (1998) presentan un sistema experto para el aprendizaje del lenguaje y Ríos et al. (2008) presentan cómo enseñar el lenguaje con la utilización de mapas conceptuales. Posteriormente, García y Lezcano (2011) presentan un software para el aprendizaje de sistemas expertos basados en Prolog, donde se muestra paso a paso el proceso de inferencia que realiza el sistema.

A pesar del avance de la IA, se mantiene la necesidad de diseñar medios tomando

en cuenta las particularidades de la asignatura donde se vayan a utilizar. Que permitan abordar problemas reales y que se dirijan a lograr un aprendizaje desarrollador en los estudiantes.

En el 2008 se elaboran los Planes D en Cuba, donde se retoma el nombre de IA en las asignaturas de la carrera Ingeniería Informática. Se abordan los fundamentos de la IA en general, de la teoría de decisión bajo incertidumbre y de los procesos de inferencia. Además, modelos de representación y procesamiento del conocimiento, aprendizaje y minería de datos. En Ciencia de la Computación se abordan las formas de representación del conocimiento, estrategias de búsqueda y heurísticas asociadas en espacios de soluciones de problemas. Además el aprendizaje automático y aplicar lenguajes declarativos para la solución de problemas. En la carrera Ingeniería en Ciencias Informáticas (ICI) se lleva a cabo un perfeccionamiento de su plan de estudio, incrementándose a dos las asignaturas de IA. Como elementos más significativos se incorporan elementos de Prolog y aprendizaje automático.

Un hecho de trascendencia histórica fue el curso de Introducción a la IA ofertado por Norving y Thrun en 2011. Este curso fue montado en una plataforma en red y se ofertó de manera abierta para todo el mundo. Matricularon alrededor de 160 000 personas, siendo reconocido como el primer curso masivo y abierto en línea (MOOC por sus siglas en inglés) con éxito. A partir de ese momento se han ofertado varios cursos introductorios de IA en plataformas abiertas (Aznar et al., 2015), lo cual ha permitido ampliar el conocimiento de esta ciencia fuera del marco de las universidades.

En la propuesta más reciente de currículos (ACM-IEEE, 2013) no se aprecian

diferencias importantes. Los elementos de mayor interés son la profundización en el estudio de las estrategias básicas de búsqueda, la eliminación de los elementos de satisfacción de restricciones y la reducción del tiempo para la representación del conocimiento y formas de razonamiento. Se introduce por primera vez como parte del currículo básico el contenido de aprendizaje automático.

Un estudio reciente encuestó a profesores y a investigadores de la IA para contrastar lo que proponían incluir en clases a los estudiantes (Wollowski et al., 2016). Como datos más significativos se destacan el deseo de los investigadores por lograr en los estudiantes habilidades en la ingeniería de sistemas, aspecto que no apareció en las respuestas de los educadores. Los educadores consideran importante tratar los temas desde una visión de juegos, mientras los investigadores desearían exponer a los estudiantes a sistemas con fuerte componente ingenieril. Ya existen ejemplos de cursos que intentan llevar a clases problemas del mundo real y de alta complejidad (Wollowski, Neller, & Boerkoel, 2017), (Sintov et al., 2017).

Los investigadores asumen como importante desarrollar habilidades en los estudiantes para utilizar técnicas y herramientas específicas de IA, mientras los docentes centran sus clases en elementos teóricos y generales.

Este estudio reafirma las contradicciones entre teoría y práctica, lo cual puede atenuarse a partir de la utilización de medios donde los elementos teóricos se puedan llevar a la práctica. El software libre educativo permitirá enfrentar al estudiante a métodos y técnicas ya implementadas que podrá estudiar, modificar y reutilizar. Los problemas a resolver con un software libre educativo de propósito general, permitirá resolver problemas reales y más relacionados con su profesión.

Los más recientes planes de estudio en Cuba (Plan E), mantienen similares sistemas de conocimientos (MES-CUJAE, 2017) (MES-UH, 2018) (MES-UCI, 2019). Es significativo la integración lograda en Ingeniería Informática, donde los elementos de IA se integran con elementos de las matemáticas aplicadas, nombrando a la disciplina Inteligencia Computacional. Esta misma concepción se presentó en la carrera ICI. En las tres carreras quedan asignaturas con el nombre histórico de IA para abordar los elementos de representación del conocimiento, estrategias de búsqueda y aprendizaje.

El informe Horizon del 2019 pronostica que la IA en los próximos dos a tres años será importante para la enseñanza, el aprendizaje y la investigación creativa (Alexander et al., 2019). Dentro de sus retos en el campo de la educación se destacan la comunicación en lenguaje natural, explicar sus razonamientos de alguna manera para ser más confiable para profesores y estudiantes y resolver problemas cada vez más parecidos a los que resuelve el hombre (Bello, 2019). La enseñanza de esta ciencia requiere estar a tono con estos retos. A la vez, exige seguir trabajando en encontrar las invariantes en el sistema de conocimientos y acercar las asignaturas a los problemas profesionales que enfrentará el futuro egresado (MES, 2016).

El análisis llevado a cabo permitió identificar cuatro etapas en la enseñanza de la IA. Para llegar a ellas se tomó como criterios de corte el sistema de conocimientos, los medios utilizados en su PEA y los paradigmas de programación asumidos:

Desde 1956 hasta 1991: Se considera como inicio de la periodización la aparición del término Inteligencia Artificial en 1956. En esta etapa se incluye la IA

paulatinamente en algunos currículos como una temática avanzada. Se incluyen conceptos teóricos generales y técnicas heurísticas como vía para dar solución a problemas intratables computacionalmente. Se propone la utilización de lenguajes de programación para el tratamiento de cadenas.

De 1991 hasta el año 2001: Se inicia con la introducción de la IA como curso básico para los currículos de Ciencias de la Computación en la propuesta de la ACM. Esta etapa se caracteriza por una gran dispersión en los objetivos y habilidades a lograr en los estudiantes y una amplia gama de contenidos tratados. Los medios utilizados son similares a los utilizados en otras asignaturas y disciplinas del currículo. Se generaliza la utilización de lenguajes de programación declarativa.

Desde 2001 hasta 2011: Inicia con la propuesta de la ACM-IEEE que consolida un núcleo de conocimiento básico para estas asignaturas. Se incluyen objetivos para representar el conocimiento, desarrollar métodos de razonamiento, tratar la incertidumbre presente en el conocimiento y desarrollar métodos de aprendizaje automático, aunque este último como temas avanzados, no básicos. Gana en importancia el paradigma de programación para agentes, el cual se mantiene como tendencia. Comienzan a aparecer y se generalizan las experiencias en la utilización de software de código abierto, donde los estudiantes implementan los métodos y técnicas de IA estudiadas en clases.

Del 2011 hasta la actualidad: Inicia con el curso de Introducción a la IA ofertado por Russell y Thrun que devino en motor impulsor de plataformas abiertas en la red. La enseñanza de la IA sale de los entornos de las universidades y se consolida la utilización de software de código abierto como medio a utilizar en dichos cursos. Se

consolida un núcleo reconocido de conocimientos a tratar en los currículos afines a la informática. Se generaliza la visión de la IA desde el paradigma de programación para agentes.

La periodización permitió identificar elementos de interés para la presente investigación:

1. Se determinan como núcleos de conocimientos básicos de la IA: la representación del conocimiento, el razonamiento (incluye las búsquedas y la utilización de heurísticas), el tratamiento de la incertidumbre y el aprendizaje automático.
2. Se identifican tres paradigmas de programación tratados como contenido y utilizados en el desarrollo computacional de los medios: La programación lógica (Prolog), la programación funcional y la programación para agentes, siendo este último la tendencia en los últimos años.
3. Se reconoce la utilización de software de código abierto como particularidad de los medios utilizados en estas asignaturas, donde los estudiantes incorporan parte del código.
4. Se identifican cuatro tipos de software para la enseñanza de la IA, las cuales se resumen en la Tabla 1.

Se reconoce en los medios estudiados la importancia de utilizar el código fuente como vía de interacción con el estudiante. De igual forma, se reconoce la importancia de su utilización para llevar a la práctica los elementos teóricos abordados en las clases.

Tabla 1. Tipología del software utilizado en Inteligencia Artificial

Software	Referencias	Características distintivas
Para visualización de algoritmos	(Lezcano & Valdés, 1998b) (McCluskey & Simpson, 2005) (Abu, 2008), (Resnick et al., 2009), (Hatzilygeroudis, Grivokostopoulou, & Perikos, 2012) (Asensio et al., 2014), (Osella, De Vito, Russo, & Ramón, 2016)	Se visualizan los algoritmos para facilitar la comprensión de los mismos por parte de los estudiantes. Algunos permiten interactuar con la visualización, modificándola, pero en pocos casos existe interacción con el código fuente.
Para la implementación de sistemas multiagentes en general	(Pantic, Zwitterloot, & Grootjans, 2005) (Cliffe, De Vos, Padget, & Casasola, 2005), (Vidal, Buhle, & Goradia, 2005)	Sistemas multiagentes donde se orienta a los estudiantes el desarrollo de sus agentes. Se tiene acceso al código fuente para el nuevo agente pero no necesariamente para todo el sistema.
Para desarrollar jugadores o agentes virtuales	(López, Montaner, & de la Rosa, 2001), (Denzinger & Kidney, 2005), (Taylor, 2010)	Se basan en juegos desarrollados donde se implementa la inteligencia de jugadores virtuales. Permite la interacción mediante la competición entre los jugadores virtuales desarrollados. Existen de código abierto y de código cerrado.
Para la implementación de juegos o partes de ellos	(Kim, 2007), (Wang & Wu, 2009), (DeNero & Klein, 2010), (McGovern, Tidwell, & Rushing, 2011), (Sosnowski et al., 2013), (Riedl, 2015) (Grivokostopoulou, Perikos and Hatzilygeroudis, 2017)	Plataformas de código abierto para el desarrollo de videojuegos o juegos parcialmente implementados, donde los estudiantes implementan los algoritmos y técnicas para darle inteligencia al mismo.

Sin embargo se identifican algunas insuficiencias:

- Aunque varios están desarrollados bajo licencias de software libre, no se aprovechan las libertades de estudiar, modificar y reutilizar el código fuente en el PEA.
- Son, en su gran mayoría, sistemas de visualización gráfica, lo cual limita la posibilidad de resolver problemas reales y relacionados con otros campos de aplicación.
- No se incluye documentación de su diseño didáctico o guías para su utilización en clases, lo cual limita su utilización en otras asignaturas similares o el desarrollo de medios con características similares.

Las dificultades encontradas refuerzan la necesidad de profundizar en las características de este tipo de software, específicamente en el aprovechamiento de las libertades del software libre en el PEA a través de su diseño didáctico. Así mismo, se requiere profundizar en el estudio de las características del PEA y la integración del software libre educativo al PEA de la IA, sus componentes y relaciones.

1.3 Integración de software libre educativo al proceso de enseñanza-aprendizaje

El PEA de forma general ha sido abordado por varios autores. González, Recarey y Addine (2002) realizan un análisis de los términos proceso pedagógico, proceso docente-educativo y proceso de enseñanza-aprendizaje. Abordan cómo estos términos han sido tratados indistintamente por algunos autores y plantean los elementos distintivos de cada uno. Describen al PEA como un proceso pedagógico

escolar que posee las características esenciales de éste, pero se distingue por ser sistemático, planificado, dirigido y específico, por cuanto la interrelación maestro-alumno, deviene en un accionar didáctico mucho más directo, cuyo único fin es el desarrollo integral de la personalidad de los educandos.

El PEA constituye la vía mediatizadora esencial para la apropiación de conocimientos, habilidades, hábitos, normas de relación, de comportamiento y valores, legados por la humanidad, que se expresan en el contenido de enseñanza, en estrecho vínculo con el resto de las actividades docentes y extra docentes que realizan los estudiantes (Zilberstein & Silvestre, 2004).

La enseñanza, el aprendizaje, el desarrollo y la educación son categorías estrechamente vinculadas entre sí. La educación es asumida desde dos acepciones, una amplia que incluye el conjunto de actividades y prácticas sociales donde los grupos humanos promueven el desarrollo personal y la socialización, garantizando el funcionamiento de uno de los mecanismos esenciales de la evolución de la especie: la herencia cultural (Zilberstein y Silvestre, 2004). La otra con un sentido más estrecho es el proceso que se organiza, desarrolla y se sistematiza en la institución docente, en función de transmitir los conocimientos y la experiencia histórico-social de la humanidad y conduce si se estructura adecuadamente, a la instrucción, el aprendizaje, el desarrollo y la formación de los estudiantes.

Addine (2002) asume que enseñar constituye una actividad que, entre otras cosas, facilita y potencia el aprendizaje desarrollador. En cuanto al aprendizaje se considera un proceso en el que participa activamente el estudiante y es dirigido por el profesor.

El estudiante se apropia de conocimientos, habilidades y capacidades, en

comunicación con los otros, en un proceso de socialización que favorece la formación de valores.

La enseñanza y el aprendizaje en conjunto constituyen un proceso que deviene objeto de estudio de la Didáctica. Este se concreta en una situación creada para que el estudiante aprenda a aprender. De esta manera el software libre educativo proporcionará las situaciones que deberá enfrentar el estudiante para operar con la realidad, para apropiarse de las herramientas que le permitan enfrentar al mundo con una actitud científica, personalizada y creadora dentro del PEA.

La utilización del software libre educativo como medio debe contribuir a un PEA esencialmente interactivo y comunicativo, de intercambio de información, compartiendo experiencias, conocimientos y vivencias, que contribuyan a una influencia mutua en las relaciones interpersonales.

El desarrollo es todo cambio esencial y a la vez necesario en el tiempo. Es un tipo de movimiento con tendencia ascendente, que implica la aparición de lo nuevo con una cualidad superior. La fuente del desarrollo radica en la lucha permanente de contrarios dialécticos que interactúan y a la vez mantienen unidad relativa, de forma tal que ambas (unidad y lucha), se tienen que dar para que ocurra el desarrollo.

Varios autores (Colectivo, 2002) (Zilberstein & Silvestre, 2004) defienden el término de enseñanza desarrolladora, que promueva un continuo ascenso en la calidad de lo que el alumno realiza, vinculado inexorablemente al desarrollo integral de su personalidad. Esta enseñanza llega a establecer realmente una unidad entre la instrucción, la educación y el desarrollo.

La Didáctica se sustenta en un sistema teórico que incluye: leyes, principios, categorías, conceptos. Páez (2013) realiza una reconceptualización de la primera ley de la didáctica en el contexto de las actuales transformaciones en el modelo pedagógico de la educación superior. De esa forma la primera ley se formula como la unidad entre el PEA y el contexto social. Mientras, la segunda ley se reconoce como: La relación dialéctica entre los componentes del PEA.

Junto a las leyes se reconocen los principios de la Didáctica. Un principio según el Diccionario de Filosofía (Frolov, 1984), es una idea que guía, una regla fundamental de conducta. Según Labarrere y Valdivia (1988), los principios se presentan como lineamientos prácticos que permiten transformar la realidad, son el punto de partida del profesor y tienen una función transformadora.

Los principios didácticos constituyen regularidades que responden a una u otra posición pedagógica (Zilberstein & Silvestre, 2004). Silvestre (2004) propuso un conjunto de principios para una enseñanza que instruya, desarrolle y eduque, basados en la más rica experiencia de la pedagogía cubana y en el enfoque histórico-cultural. Dentro de los aspectos más relevantes está la estructuración del PEA hacia la búsqueda activa del conocimiento por el estudiante desde posiciones reflexivas, que estimule y propicie el desarrollo de su pensamiento y su independencia. Desarrollar formas de actividad y de comunicación colectivas, atendiendo las diferencias individuales en el desarrollo de los estudiantes y vinculando el contenido de aprendizaje con la práctica social.

Páez (2013) realiza un análisis histórico y crítico sobre varias propuestas de principios didácticos. Retoma lo más valioso de los anteriores, para presentar un

conjunto de principios caracterizados por su nivel de integración, racionalidad y esencialidad propias del modelo pedagógico de la educación superior:

- El carácter cultural, con énfasis en sus componentes científico-técnico e ideológico en el PEA.
- EL carácter instructivo, desarrollador y educativo en el PEA.
- El principio de la atención a la diversidad desde lo grupal e individual en el PEA.
- El carácter intencional e integrador de la instrucción y la educación en el PEA en todos los contextos de formación.

Se reconocen como categorías rectoras de la didáctica: el objetivo, el contenido, los métodos, los medios, la evaluación y las formas de organización (Zilberstein & Silvestre, 2004). Varios autores reconocen en estas categorías los componentes del PEA (González et al., 2002), (Chávez, 2007).

Al objetivo se le considera la categoría rectora y representa el elemento orientador de todo el proceso. El contenido expresa lo que se debe apropiarse el estudiante, incluye la cultura y la experiencia social que debe ser adquirida y se encuentra en dependencia de los objetivos propuestos; en relación a los principios declarados, cumple funciones instructivas, educativas y desarrolladoras. El método constituye el sistema de acciones que regula la actividad del profesor y los estudiantes, en función del logro de los objetivos. Los medios están constituidos por objetos reales, sus representaciones e instrumentos que sirven de apoyo material para la apropiación del contenido, complementando al método, para la consecución de los objetivos. La

evaluación es el proceso para comprobar y valorar el cumplimiento de los objetivos propuestos y la dirección didáctica de la enseñanza y el aprendizaje en sus momentos de orientación y ejecución. Las formas de organización son el soporte en el cual se desarrolla el proceso de enseñanza aprendizaje. (González et al., 2002)

Con el vertiginoso avance de la tecnología, los medios ocupan un lugar cada día más relevante en el PEA. Estos adquieren su verdadero papel cuando se utilizan en sistema y en relación con las restantes categorías didácticas. Están en constante renovación y son de diferente naturaleza, tomando en cuenta el desarrollo tecnológico históricamente alcanzado y de la labor innovadora de los profesores.

Los componentes del PEA se relacionan de manera sistémica entre ellos y con los sujetos participantes en el PEA. El estudiante, el profesor y el grupo son participantes activos, considerados protagonistas en un PEA desarrollador (González et al., 2002), (Castellanos et al., 2002). Todo PEA desarrollador se centra en torno a la persona que aprende. En función de potenciar su aprendizaje se organiza la actividad individual y la comunicación con el profesor y con el grupo.

Para el trabajo con el software libre educativo, cada estudiante aporta sus saberes y experiencias previas, sus intereses y motivaciones. Constituye un sujeto activo, que construye y reconstruye sus aprendizajes, autorregula su actividad de estudio en el contexto particular del aula y la vida del grupo docente, mediante la utilización del software libre educativo. El estudiante como centro del proceso particularizará las tareas generales a realizar con el software. La posibilidad de trabajar directamente con el código fuente, permitirá al estudiante modificar y adicionar funcionalidades según sus intereses y su propio desarrollo.

El grupo como protagonista ha planteado más retos a la didáctica, partiendo de la contradicción dialéctica de la relación entre lo individual y lo colectivo, entre el carácter individual del aprendizaje y el carácter social de la enseñanza (Castellanos et al., 2002). El grupo constituye un órgano vivo, que se conforma en las interacciones y la comunicación, aportando ideas para adicionar funcionalidades al software libre educativo, definir metas y objetivos comunes. Sustentado en la propia filosofía del software libre, se comparten soluciones entre los estudiantes, lo cual condicionará los cambios en el proceso y las modificaciones finales al software libre educativo.

El profesor tiene el encargo social de establecer la mediación indispensable entre la cultura y los estudiantes. Define las tareas generales a realizar por los estudiantes y brinda las ayudas necesarias para que el estudiante las resuelva de manera creativa y avanzando según su propio desarrollo. Dirige todo el proceso, desde el diseño didáctico del software libre educativo hasta su incorporación al PEA.

El uso de los medios de enseñanza es un tema de actualidad en el mundo. Las investigaciones van dirigidas a la incorporación de las TIC al ámbito educativo, pero también, a lograr un enfoque integrador, contextualizado y desarrollador (Lombillo et al., 2018). Desde hace años se hace referencia a las Tecnologías del Aprendizaje y el Conocimiento (TAC), para denominar a las TIC que se dirigen específicamente a mejorar los procesos de enseñanza-aprendizaje (Moya, 2013).

Para la incorporación de las TIC, de forma general, al PEA se identifican tendencias que van más allá del mero uso de las mismas. Algunas de las posiciones teóricas que se han asumido son (Cabrera, 2008):

- La adopción de las TIC (Ibarra, Torres, & Cervantes, 2013), (Aguilar, Luzardo, & Jaimes, 2015). En ella se han desarrollado modelos y teorías que explican la intensidad e intencionalidad del sujeto al utilizarla.
- La asimilación de las TIC (Belloso & Perozo, 2009) (Osorno, 2013). Se hace énfasis en la necesidad del desarrollo de habilidades en los profesores y en los aspectos culturales relacionados con el uso de las TIC en general.
- Integración de las TIC (Fluck, 2003) (Cabrera, 2008) (Torres, 2011), (Rombys-Estévez, 2013), (Álvarez, 2014), (Morales, Trujillo, & Raso, 2015) (Area-Moreira et al., 2016). El elemento distintivo es la transformación del PEA para incluir las TIC. Es el más tratado y aceptado en la actualidad en las investigaciones de corte pedagógico.

Varias investigaciones se relacionan con la integración de las TIC al PEA. Para Fluck (2003) el término “integración” se refiere al cambio reflexivo en los modelos pedagógicos destinados a hacer que las TIC satisfagan las exigencias didácticas de un contexto específico, a la combinación de componentes en un todo complejo y armonioso, donde la tecnología se desvanece en el PEA. Esta idea plantea efectivamente el cambio y la combinación de forma compleja y armoniosa, sin embargo la tecnología no debe desvanecerse dentro del proceso, sino que debe desempeñar un papel esencial, debe transformar el proceso junto al profesor para lograr un proceso centrado en el estudiante, dirigido por el profesor y mediado por la tecnología.

Según Álvarez (2014) en la evolución de la integración de las TIC como fenómeno cultural, han incidido factores como la infraestructura, los medios en sí y las metodologías. Estos factores coexisten en estrecha y dinámica interrelación. La integración de las TIC requiere del dominio de los artefactos tecnológicos y la asimilación de nuevos códigos que sustenten la mediación para transformar la realidad. Diferentes autores y estándares internacionales coinciden, de forma general, en que las dimensiones fundamentales para el estudio de la integración de las TIC en el PEA son la tecnológica, la pedagógica y la política institucional (Álvarez, 2014).

La dimensión tecnológica se asocia a la infraestructura y el soporte técnico a profesores y estudiantes. La dimensión política institucional contempla las estrategias de tecnología educativa adoptadas. La dimensión pedagógica cubre la planificación y desarrollo del PEA con las TIC y la preparación y disposición del profesor para su integración. Mientras la tecnología y la política institucional condicionan y sustentan la integración, la dimensión pedagógica la determina y en ello el profesor resulta esencial (Álvarez, Hernández, & Cabrera, 2013).

Con el propio desarrollo de la educación superior cubana se han creado las condiciones tanto tecnológicas como de política institucional para, más que garantizar, potenciar la integración de tecnologías. Sin embargo, los elementos pedagógicos deben particularizarse para cada tipo de tecnología, por lo que la dimensión que determina es la pedagógica. En la presente investigación se trabaja específicamente la integración del software libre educativo a un PEA específico, las asignaturas de IA. De esta forma se asumen las investigaciones sobre la integración

de TIC como referente fundamental, pero se particularizan en la integración del software libre educativo.

Para Cabrera (2008) “La Integración de las TIC es un proceso contextualizado, sistémico, continuo y reflexivo, orientado a la transformación de la práctica pedagógica y en el seno de los procesos que conforman el PEA, con la finalidad de incorporar armónicamente las tecnologías de información y comunicaciones apropiadas para satisfacer los objetivos educativos.” Esta definición es actualizada posteriormente y se le adiciona la cualidad “planificado” (Álvarez et al., 2013).

Es de destacar en esta definición las cualidades que se atribuyen al proceso de integración y reafirma como fin la transformación del PEA. Aunque este autor y otros que han asumido esta definición se refieren a la integración de forma general de las TIC y no a tecnologías específicas. Esta integración asume la incorporación de tecnologías ya existentes, no se refiere al desarrollo o diseño de nuevas tecnologías.

Para Cabrera (2008) la cualidad contextualizado se expresa a partir de que la integración de las TIC está asociada al espacio en el que se lleva a cabo el PEA. Son de vital importancia las características culturales, la motivación y preferencias tecnológicas de los profesores, por lo que debe abordarse especialmente en cada asignatura, evitando la generalización deliberada de soluciones.

La cualidad sistémica se expresa a partir del conjunto de fases o etapas que le dan carácter de proceso. Debe tomar en cuenta como componentes fundamentales a la tecnología en sí, en este caso el software libre educativo y el PEA de la asignatura específica. Abordar sus relaciones y aportes del software al PEA.

De igual forma el continuo avance de las tecnologías y las nuevas exigencias del PEA exige abordar la integración como un proceso cíclico, continuo. Cabrera (2008) y Álvarez (2014) asumen el esquema del proyecto Apple Classroom of Tomorrow (ACOT) donde los profesores transitan a través de cinco etapas, entrada de nuevas tecnologías, adopción, adaptación, apropiación e invención. Sin embargo, cuando se hace referencia a una misma tecnología, como es el caso del software libre educativo, no es necesario analizar estas etapas de manera cíclica. La continuidad se expresa a través del perfeccionamiento continuo del medio, adaptándolo a las nuevas condiciones del desarrollo tecnológico, a partir de la propia utilización del mismo en cada curso académico, o incluso, de un tema a otro dentro de la misma asignatura.

Cabrera (2008) propone la creación de estructuras que permitan a los docentes acceder a la tecnología, reflexionar y discutir sus ideas con especialistas, con otros profesores o simplemente colaborar con otros mientras trabajan con la tecnología. En la presente investigación se le brinda un valor adicional a la cualidad reflexiva. Se reconoce como importante el ejercicio metacognitivo tanto de los profesores como de los estudiantes. Donde la reflexión no solo sea entre colegas, sino reflexionar acerca del proceso y los aportes que este hace en la superación personal.

La planificación del proceso (Álvarez et al., 2013) se expresa a partir de acciones concretas en el tiempo para lograr un trabajo organizado y sistemático.

Cada una de estas cualidades se expresa en el PEA, a partir de la incorporación del software libre educativo como medio y con el fin de su transformación. De ahí que para la integración del SLE al PEA de la IA, se identifican dos dimensiones: el

diseño didáctico del software libre educativo y la **incorporación armoniosa** del mismo al PEA. Esta última recoge el análisis de las cualidades antes presentadas y se define como el proceso contextualizado, sistémico, continuo, reflexivo y planificado, dirigido a utilizar un software libre educativo en el PEA para el cual fue diseñado didácticamente.

De esta manera se define la **integración de software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial** como el proceso de diseño didáctico de un software libre educativo y su incorporación armoniosa al proceso de enseñanza-aprendizaje de la Inteligencia Artificial con el fin de su transformación.

Del análisis anterior se aprecia la subordinación de la incorporación armoniosa del SLE a su diseño didáctico. De la misma forma, la fuerte relación de ambos al proceso de enseñanza-aprendizaje específico donde se va a llevar a cabo la integración.

1.4 Operacionalización de la variable

Como variable objeto de estudio se declara la **Integración de software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial**. Para abordar sus particularidades se toma como base la definición presentada en el epígrafe 1.3. A partir de esta definición, se asumen dos dimensiones: el diseño didáctico del software libre educativo y su incorporación armoniosa al proceso de enseñanza-aprendizaje. Los rasgos esenciales se muestran a continuación.

- **Diseño didáctico del software libre educativo**

El diseño didáctico del software libre educativo, como fue definido en el epígrafe 1.1 es el proceso sistémico de modelación de las situaciones de enseñanza–aprendizaje

que se desarrollarán con el software libre educativo, a partir de las interrelaciones entre las dimensiones semántica, práctica, tecnológica, espacial y de gestión.

Las dimensiones del diseño didáctico se constituyen en la presente investigación como subdimensiones:

Didáctica: Determina las relaciones del software con los componentes del PEA: objetivos, contenidos, métodos, medios, evaluación y formas de organización. Los medios vistos como un sistema que soporte la utilización del software libre educativo en el PEA.

Tecnológica: Analiza las características del software, específicamente en las relacionadas con el movimiento de software libre que deben ser tomadas en cuenta en su desarrollo. Elementos de interés son la accesibilidad al código, la organización arquitectónica del software que permita su adecuado estudio y modificación y la disponibilidad de documentación de ayuda para utilizar el software en el PEA.

Espacial: Se determinan las tareas para los distintos espacios en los cuales se utiliza el software dentro del PEA: individual, por equipo y grupal.

Gestión: Incluye características de la licencia del software y las vías que permiten su correcta distribución. Así como las características tecnológicas que garanticen su utilización en el PEA, tomando en cuenta las condiciones existentes.

Práctica: Incluye, como estrategia específica, el diseño de tareas donde los estudiantes tengan que estudiar, modificar y reutilizar el código fuente del software.

- **Incorporación armoniosa del SLE al PEA**

Esta dimensión está conformada por cinco subdimensiones que se explicitan en la definición presentada en el epígrafe 1.3. Los rasgos distintivos de cada una de las subdimensiones se ofrecen a continuación:

Contextualizada: Se refiere a la relación entre las tareas definidas para el estudiante y el contexto. Se incluye el trabajo de los estudiantes en su PP, así como sus propios intereses enriquecidos con sus conocimientos culturales y profesionales.

Sistémica: Esta subdimensión se evalúa integralmente, las demás dimensiones con sus indicadores tributan a evaluar el enfoque sistémico de todo el proceso. Específicamente se analiza el componente software libre educativo y su relación con la asignatura, así como su impacto en el PEA.

Continua: Asume la incorporación armoniosa del SLE como un proceso que no termina con la evaluación de los resultados, sino que los mismos sirven de base para próximos cursos. Se analiza si se toman en cuenta las soluciones de los estudiantes para mejorar el software. De igual forma la continuidad dentro de la propia asignatura es de interés. Se analiza si existe relación desde las primeras tareas del estudiante con el software hasta las más complejas al final del curso.

Reflexiva: Se refiere a la percepción que tiene el profesor y los estudiantes acerca de la importancia de la utilización del software libre educativo. En el caso de los profesores se valora tanto su percepción sobre la importancia del SLE para su autosuperación, como para los estudiantes como futuros profesionales.

Planificada: Vinculada a la organización y la creación de las condiciones para llevar a cabo la incorporación armoniosa del SLE al PEA. Incluye el análisis del tiempo utilizado por el profesor para el diseño didáctico del software, así como para su

autosuperación. Se considera importante la cantidad de clases en que se planifican tareas con el software, así como el tiempo planificado y asignado para el trabajo independiente del estudiante con el SLE. Por último las vías previstas y utilizadas para la retroalimentación de todo el proceso de integración del SLE al PEA de la IA.

La operacionalización de la variable aparece en el Anexo 1.

Para la evaluación de las dimensiones, subdimensiones y sus indicadores, se establecieron escalas valorativas cualitativas para cada indicador (Anexo 2), las que se relacionan con valores cuantitativos entre cinco (máximo) y cero (mínimo), estableciendo un convenio entre lo cuantitativo y lo cualitativo. A partir del promedio de los valores que cada instrumento aplicado asigna para evaluar los indicadores (Anexo 3), se determina el índice (I) de evaluación del indicador. Este permite determinar un criterio cualitativo del indicador, fijando los criterios de Alto, Medio y Bajo como se muestra en la Tabla 2. Este proceder facilita el procesamiento estadístico y su valoración.

Tabla 2. Escalas valorativas para la evaluación de los indicadores

Escala	Rango	Nivel
5	Si $4 \leq I \leq 5$	ALTO
4		
3	Si $2 \leq I < 4$	MEDIO
2		
1	Si $0 \leq I < 2$	BAJO
0		

1.5 Estado inicial de la variable objeto de estudio

Para caracterizar el estado inicial de la integración de SLE al PEA de la IA en la carrera ICI en el curso escolar 2015-2016, se emplearon varios métodos de

investigación. Se determinaron como unidades de estudio los directivos, los profesores, los estudiantes, las asignaturas y el software utilizado en cada asignatura. La selección de la muestra de directivos, profesores y estudiantes fue intencional, se analizaron las dos asignaturas lectivas, así como el software que se utiliza en cada una de ellas.

Para el caso de los directivos se consideró como población los jefes de departamento docente, los jefes de colectivo de asignatura de cada facultad y el jefe de la disciplina. Los profesores de interés para el estudio fueron los que imparten las asignaturas de la disciplina, tanto las lectivas como optativas. Para la selección de la muestra se seleccionaron de forma intencional, de manera que estuvieran representadas todas las facultades y los cargos en el caso de los directivos.

Para los estudiantes se optó por hacer un estudio exploratorio con los dos grupos docentes del investigador. Se considera útil en tanto el PEA se imparte de forma similar en todas las facultades de la universidad y los grupos son conformados con estudiantes de varias provincias del país y diversas vías de ingreso a la educación superior.

La población y muestra por cada unidad de estudio se muestra en la Tabla 3.

Tabla 3. Unidades de estudio para el diagnóstico de la variable.

	Directivos	Profesores	Estudiantes	Asignaturas	Software
Población	13	25	524	2	2
Muestra	6	11	53	2	2
Por ciento	46%	44%	10%	100%	100%

A continuación se presentan los resultados de los instrumentos aplicados:

Entrevista a directivos:

Se aplicó una entrevista con preguntas valorativas sobre las subdimensiones de cada dimensión (Anexo 4), adicionalmente se aplicó un cuestionario para recoger los valores cuantitativos otorgados a cada indicador (Anexo 5). Los resultados de cada indicador, expresados por los directivos, se recogen en el Anexo 6.

Se entrevistaron seis directivos, todos son graduados de perfil informático y con categoría docente de Profesor Asistente. Tienen entre cinco y 10 años de experiencia en la disciplina y cuatro son Máster.

En la subdimensión didáctica, el 50% de los directivos reconocen la existencia de orientaciones precisas para utilizar el software en las clases de laboratorio, pero no en otras formas organizativas del PEA, el otro 50% aunque no cree que hayan orientaciones precisas, sí reconocen la posibilidad de utilizarlo. El 66% plantea que hay indicaciones para la utilización del software con los medios disponibles en los laboratorios y en la plataforma utilizada en la carrera. De igual forma, todos afirman que están definidas las formas de evaluación parcial con el software.

La dimensión de mejores resultados fue la de gestión. Sobre las garantías para la distribución y uso del software, todos afirman que existen buenas condiciones para su utilización en el PEA.

Sin embargo, sobre la subdimensión contextualizada todos plantean que a los estudiantes no se les asignan tareas relacionadas con sus intereses ni con su PP. El 83% afirma que no es posible asignar tareas de modificación del código fuente y el 66% que no son posibles las tareas de reutilización. De igual forma, la mayoría (83%)

considera que el trabajo con el software no contribuye a la superación de los profesores.

Entrevista a profesores:

Para la entrevista a los profesores se utilizó la misma guía que para los directivos (Anexo 4) y el mismo cuestionario para recoger los valores cuantitativos otorgados a cada indicador (Anexo 5). Los resultados cuantitativos se aprecian en el Anexo 7.

Los 11 profesores seleccionados son graduados de perfil informático, dos Instructores (18%), un Profesor Titular (9%) y ocho Profesores Asistentes (73%). Seis son Máster (55%) y un Doctor en Ciencias Técnicas (9%).

De forma general los resultados fueron similares a los resultados de los directivos. Es significativo que todos los profesores reconocen que el software Weka (Waikato, 2019) no se aprovecha todo lo que se pudiera en el PEA. Sólo se utiliza su interfaz visual para configurar y ejecutar los diversos algoritmos que trae implementados. No se utiliza en el aula por lo que no se vinculan a él los medios disponibles, como la pizarra y la computadora.

El 45% de los profesores afirman que algunas veces toman en cuenta los intereses de algunos estudiantes para asignarle las tareas. Igualmente el 36% afirma reutilizar algunos resultados, de los estudiantes con mayor aprovechamiento académico, de un tema a otro en la asignatura. El 64% alegan que realizan cortes del trabajo en las clases de laboratorio o en encuentros en otros horarios.

La mayoría de profesores (73%) afirman que aunque el trabajo con el software no tributa a su investigación, tiene importancia para su autosuperación.

Encuesta a estudiantes:

A los estudiantes se les aplicó una encuesta (Anexo 8) con la cual se recogieron los resultados que se muestran en el Anexo 9.

Los grupos de estudiantes encuestados fueron el 5501, conformado por 28 estudiantes, ocho mujeres y 20 hombres y el 5503, conformado por 25 estudiantes, siete mujeres y 18 hombres.

Como elementos de mayor interés el 21% refiere que el trabajo con el software aporta conocimientos específicos de la asignatura, y aporta elementos para su futuro trabajo profesional; el 43% dice que solo aporta elementos de interés para la asignatura y más de un tercio (36%) dice que no tiene ningún aporte.

Es de destacar como, en promedio, asignan casi un punto menos (0.84) al valor de la subdimensión tecnológica con respecto a los valores obtenidos por directivos (1.67) y profesores (1.70). Los demás indicadores presentan valores similares a los obtenidos en las entrevistas a directivos y profesores.

Análisis documental:

Para el análisis documental se siguió la guía que aparece en el Anexo 10. Esta permitió asignar un valor a cada indicador evaluado (Anexo 11).

Las asignaturas de interés para el estudio son las dos asignaturas lectivas Inteligencia Artificial I (IA-I) e Inteligencia Artificial II (IA-II). Se tomaron en cuenta para el análisis documental:

- Programa analítico de la disciplina Inteligencia Artificial. Plan D del Ingeniero en Ciencias Informáticas.

- Programa analítico de la asignatura IA-I. Plan D del Ingeniero en Ciencias Informáticas.
- Programa analítico de la asignatura IA-II. Plan D del Ingeniero en Ciencias Informáticas.
- Plan calendario de la asignatura IA-I. Curso 2015-2016.
- Plan Calendario de la asignatura IA-II. Curso 2015-2016.
- Acta de reunión del concentrado metodológico de la disciplina. Julio de 2016.
- Informe final de la disciplina Inteligencia Artificial en el curso 2015-2016.

De la subdimensión didáctica se pudo apreciar que aunque no existen indicaciones precisas en los programas de las asignaturas, se pueden trabajar varios objetivos y contenidos con el SLE en diversas formas organizativas. Adicionalmente están las indicaciones de cómo tomarlo en cuenta como evaluación parcial del estudiante.

De la subdimensión espacial se evalúan sus tres indicadores. A partir de las indicaciones de los programas analíticos y de la especificación de tareas en los planes calendario, se observa que pueden llevarse a cabo tareas para el trabajo con el software tanto a nivel de grupo como de forma individual y en equipo. De igual forma se pueden inferir varias tareas posibles para el estudio, modificación y reutilización del código fuente, indicadores de la subdimensión práctica. Sin embargo, no se especifica este tipo de tareas en ninguno de los documentos analizados.

Relacionado con la subdimensión contextualizada de la dimensión incorporación armoniosa, en el informe final de la disciplina no se evidencia la asignación a los estudiantes de tareas relacionadas con sus intereses o la PP.

Aunque ningún software está diseñado específicamente para las asignaturas, pueden ser utilizados en parte de la misma. En el análisis de los programas de las asignaturas y el informe final de la disciplina se aprecia la importancia del software para el PEA. Estos indicadores responden a la subdimensión sistémica.

Para evaluar la subdimensión continua se analizó el informe final de la disciplina. En este documento no se evidencia la reutilización de resultados de cursos anteriores. Tampoco se evidencia haber realizado algún tipo de acción con vistas a reutilizar los resultados obtenidos por los estudiantes para reutilizarlos en el siguiente curso.

En cuanto a la subdimensión reflexiva, ni en el informe final de la asignatura, ni en el acta de reunión metodológica se expresan beneficios del software para la superación de los profesores, aunque tampoco insatisfacciones con ese objetivo. Sí se expresa la importancia del software Weka para el desarrollo de habilidades en los estudiantes, relacionadas con la minería de datos.

Para la subdimensión planificada se analizaron todos los documentos disponibles. En el plan del proceso docente de IA-I se planifican tareas con el software en el 12,5% de las horas lectivas totales. En IA-II se planifican en el 16% de las horas lectivas totales. El trabajo independiente con el software se les orienta a los estudiantes, pero se controla solo a partir de las tareas planificadas en las clases de laboratorio. En los documentos se evidencia que los profesores opinan sobre el proceso a partir de sus experiencias en el proceso, pero no se planifican y aplican vías de retroalimentación.

Análisis del software:

Para el análisis de cada software se siguió la guía que aparece en el Anexo 12, la cual permitió asignar un valor a cada indicador evaluado por cada una de las dos aplicaciones informáticas, como se muestra en el Anexo 13.

En la asignatura IA-I se analizó el software JPrologEditor, utilizado como interfaz de desarrollo para la programación lógica. Se evidencia la posibilidad de trabajar varios objetivos y contenidos, utilizar diversos métodos y apoyarse en diversos medios dentro del PEA. Su documentación de ayuda está disponible, pero no incorpora indicaciones para su utilización en el PEA.

Por ser software libre, su código fuente está disponible pero, al ser una aplicación compleja, desarrollada con el propósito de ser utilizada en la construcción de otras aplicaciones, es poco legible y su arquitectura no permite que sea extensible, por lo que se hace complejo el estudio o modificación del código fuente. Está amparado en una licencia libre, es multiplataforma y se encuentra disponible en la red al menos en horario laboral.

Se evidencia la posibilidad de asignar tareas para el trabajo en grupo, individual y por equipo, así como de estudio del código fuente, pero no es posible asignar tareas para la modificación y reutilización del código, esta característica influye en la no reutilización de los resultados de los estudiantes de un curso a otro.

En IA-II se analiza el software Weka (Waikato, 2019), utilizado para el análisis de algoritmos de minería de datos. Este software puede ser utilizado en parte de la asignatura, específicamente en el tema relacionado con la minería de datos. Se puede trabajar con él varios objetivos y contenidos de la asignatura. Su propio diseño

de interfaz facilita la utilización de varios métodos, al igual que permite apoyarse en varios medios para su trabajo en el PEA.

Su diseño permite que partes del código sean perfectamente legibles por los estudiantes, existe abundante documentación para su uso, incluso elaborada por el propio colectivo de la asignatura, pero no un guion que permita de forma organizada su utilización en el PEA.

Su arquitectura brinda facilidades para modificar o adicionar nuevas funcionalidades, aunque no se utiliza un paradigma específico de la IA. También se pueden inferir tareas para el trabajo en grupo, en equipo o de forma individual, así como tareas de estudio, modificación e incluso con facilidades para la reutilización de los métodos que trae implementados. Está amparado en una licencia libre, es multiplataforma y se encuentra disponible en la red. A pesar de sus facilidades, no se aprecian modificaciones en el software a partir de la reutilización de resultados de los estudiantes.

Con el objetivo de profundizar en los elementos cualitativos y la caracterización del objeto de estudio se realiza una valoración de cada subdimensión, las dimensiones y la variable de estudio, tomando en cuenta los resultados de los instrumentos aplicados y la experiencia del investigador en más de 10 años de trabajo en la disciplina. El Anexo 14 recoge los resultados de cada indicador por instrumento y la derivación de la evaluación para las subdimensiones, dimensiones y la variable objeto de estudio. En la Tabla 4 se muestra un resumen de los valores cualitativos de cada subdimensión, dimensión y la variable objeto de estudio, a partir de los resultados de cada instrumento.

Tabla 4. Valores cualitativos de cada subdimensión por instrumento

Dimensión	Subdimensión	Entrevista directivos	Entrevista profesores	Encuesta estudiantes	Análisis documental	Análisis del software	Subdimensión	Dimensión	VOE
1	1.1	MEDIO	MEDIO	BAJO	MEDIO	MEDIO	MEDIO	BAJO	BAJO
	1.2	BAJO	BAJO	BAJO		MEDIO	BAJO		
	1.3	BAJO	BAJO		BAJO	BAJO	BAJO		
	1.4	ALTO	ALTO	MEDIO		ALTO	ALTO		
	1.5	BAJO	BAJO		MEDIO	BAJO	BAJO		
2	2.1	BAJO	BAJO	BAJO	BAJO		BAJO	BAJO	
	2.2	MEDIO	MEDIO	BAJO	BAJO	BAJO	BAJO		
	2.3	BAJO	BAJO	BAJO	BAJO	BAJO	BAJO		
	2.4	BAJO	MEDIO	MEDIO	BAJO		BAJO		
	2.5	BAJO	BAJO	MEDIO	BAJO		BAJO		

Subdimensión didáctica (1.1):

De forma general se aprecia coincidencia entre los resultados de los distintos instrumentos, solo los estudiantes otorgan nivel BAJO, los demás la evalúan de nivel MEDIO. El análisis de los documentos y el software evidencian relación de nivel MEDIO de los contenidos y los métodos con el software, pero los entrevistados y encuestados consideran la relación de nivel BAJO.

Dentro de los elementos más significativos se identifica que para trabajar con el software Weka sólo se utiliza su interfaz visual para configurar y ejecutar los diversos métodos que trae implementados. Se pudo comprobar la no utilización del software en el aula para exponer o explicar sus características.

Es determinante en la evaluación de esta subdimensión que no existen indicaciones precisas para el trabajo con cada software en el PEA. Sin embargo, se aprecian potencialidades en ellos, lo que permite evaluar la subdimensión con nivel MEDIO.

Subdimensión tecnológica (1.2).

Los profesores coinciden con el análisis del software en que parte del código, sobre todo del Weka, es legible. En tanto los estudiantes evalúan de nivel BAJO la disponibilidad de ayuda. Existe coincidencia en señalar de nivel BAJO la organización de la arquitectura, aunque, en el caso de Weka, permite modificar o adicionar funcionalidades.

En esta subdimensión resalta que a pesar de ser software libre, ningún software presenta facilidades para utilizar su código en el PEA, sobre todo el JPrologEditor. Se le suma a esto la no existencia de un diseño didáctico expresado en algún documento que facilite la utilización del software en el PEA. Estos elementos determinan la evaluación de la subdimensión con nivel BAJO.

Subdimensión Espacial (1.3).

Todos los instrumentos evidenciaron que no están bien definidas ni elaboradas las tareas a realizar en los distintos espacios, por lo que los tres indicadores son evaluados de nivel BAJO. Tanto los profesores como los directivos coinciden con el análisis del software al inferir varias tareas posibles para el trabajo grupal, por equipos o individuales con el software. La evaluación de esta subdimensión es de nivel BAJO.

Subdimensión de gestión (1.4).

Tanto la posibilidad de distribuir el software, como las condiciones tecnológicas para su utilización están garantizadas. Ambas aplicaciones informáticas están desarrolladas bajo licencias libres, son multiplataforma y se encuentran disponibles en la red. Existe el equipamiento necesario en los laboratorios, los cuales pueden ser

accedidos por el estudiante no solo en horario de clases, también en horario de estudio independiente. Aunque en las aulas existen computadoras y televisores para proyectar, los profesores y estudiantes plantean que no siempre están en buenas condiciones para su utilización, por lo que evalúan los indicadores con nivel MEDIO. En conclusión se evalúa la subdimensión con nivel ALTO.

Subdimensión práctica (1.5).

Las tareas de estudio del código fuente se pueden inferir. En el análisis documental se aprecia la posibilidad de asignar tareas de modificación y reutilización del código, por lo que esos indicadores son evaluados de nivel MEDIO por ese método. Sin embargo, tanto el análisis del software, como las entrevistas a los profesores y a los directivos, reflejan la imposibilidad de hacerlo. De forma general se evalúa la subdimensión como de nivel BAJO.

Dimensión diseño didáctico del software

De forma general las condiciones tecnológicas tanto para la gestión del software como para su utilización están garantizadas. Las características del software, aunque con limitaciones respecto a su arquitectura, pueden contribuir a cada uno de los componentes del PEA. Las principales dificultades están relacionadas con las limitaciones para el estudio y modificación del código fuente, así como en la falta de un documento donde se definan las posibles tareas a realizar por los estudiantes.

Se evidencia un nivel MEDIO en la subdimensión didáctica y nivel ALTO en la subdimensión de gestión. Sin embargo, es determinante para evaluar la dimensión, el nivel BAJO de la subdimensión tecnológica y la subdimensión espacial. A partir de

la caracterización de cada una de las subdimensiones se evalúa la dimensión con nivel BAJO.

Contextualizado (2.1)

Todos los instrumentos aplicados evidenciaron la inexistencia de tareas relacionadas con la PP de los estudiantes. Los directivos y los documentos analizados evidencian que no se toman en cuenta tampoco los intereses de los estudiantes, con lo que se evalúan sus indicadores con nivel BAJO. Sin embargo, el 45% de los profesores afirman que aunque no existen vías bien definidas para conocer sus intereses, se toman en cuenta a la hora de asignar las tareas a algunos estudiantes, lo cual confirma el 53% de los estudiantes. Esta subdimensión es evaluada con nivel BAJO.

Sistémico (2.2)

Tanto las entrevistas como el análisis del software y los documentos reflejan que existe un nivel BAJO de relación entre el software y la asignatura, en ambos casos se pueden utilizar en parte de la asignatura, pero no fueron diseñados específicamente para ello. Se reconoce que la no utilización del software afectaría el tema de la asignatura que lo utiliza, lo que refleja un nivel MEDIO del indicador. Aunque es importante señalar que un 58% de estudiantes solo lo catalogan como útil en el PEA, pero no utilizarlo no afectaría el PEA. La subdimensión se evalúa con nivel BAJO.

Continuo (2.3)

Se hace evidente en todos los instrumentos el nivel BAJO de reutilización de resultados con el software de un curso a otro, o de un tema a otro en la asignatura.

Algunos profesores (36%) y algunos estudiantes (21%) expresan que en algunos casos excepcionales se reutilizan resultados. Esta subdimensión es evaluada con nivel BAJO.

Reflexivo (2.4)

Se comparte por los profesores (82%) y directivos (83%) el aporte del trabajo con el software a objetivos o habilidades específicas en los estudiantes, con lo cual evalúan de nivel MEDIO ese indicador. Sin embargo, evalúan de nivel BAJO el aporte del trabajo con el software a la autosuperación de los profesores. Por su parte los estudiantes evalúan de nivel MEDIO el indicador relacionado con el aporte del software, pero el 79% no reconoce un aporte significativo a su futura vida profesional, aspecto que corroboran las entrevistas a los profesores. La subdimensión es evaluada con nivel BAJO.

Planificado (2.5)

El análisis de los instrumentos arrojó que es de nivel BAJO la dedicación de los profesores al diseño didáctico del software, así como a su autosuperación para trabajar con el software. Se comparte como de nivel MEDIO la proporción de clases en las que se utiliza el software. Por otro lado, aunque los profesores, directivos y estudiantes evalúan de nivel MEDIO la orientación y control del trabajo independiente a los estudiantes, el análisis documental refleja que es de nivel BAJO. De forma general se evalúa la subdimensión con nivel BAJO.

Dimensión incorporación armoniosa

De forma general, aunque hay indicadores con nivel MEDIO, todas las subdimensiones se evalúan de nivel BAJO. Se reconoce la importancia de la

utilización del software, pero no existe un diseño de las tareas contextualizadas ni reutilización de los resultados de los estudiantes. De igual forma se evidencian problemas en la planificación para su utilización en el PEA. Se puede apreciar un nivel BAJO de la dimensión.

Integración de software libre educativo al proceso de enseñanza-aprendizaje

En cuanto a la variable se evalúa de nivel BAJO, ya que tanto la dimensión diseño didáctico del software como la dimensión incorporación armoniosa del SLE al PEA, se evalúan de nivel BAJO.

Las subdimensiones reflejan que de forma general la subdimensión de gestión tiene nivel ALTO y la didáctica nivel MEDIO, las demás tienen nivel BAJO (Figura 1).

Del análisis llevado a cabo se aprecian potencialidades para llevar a cabo la integración de software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial. La caracterización de los profesores evidencia que todos son graduados de especialidades afines a la informática y con al menos cinco años de experiencia. Las condiciones tecnológicas en los laboratorios y en las aulas favorecen la utilización del software en el proceso de enseñanza-aprendizaje.

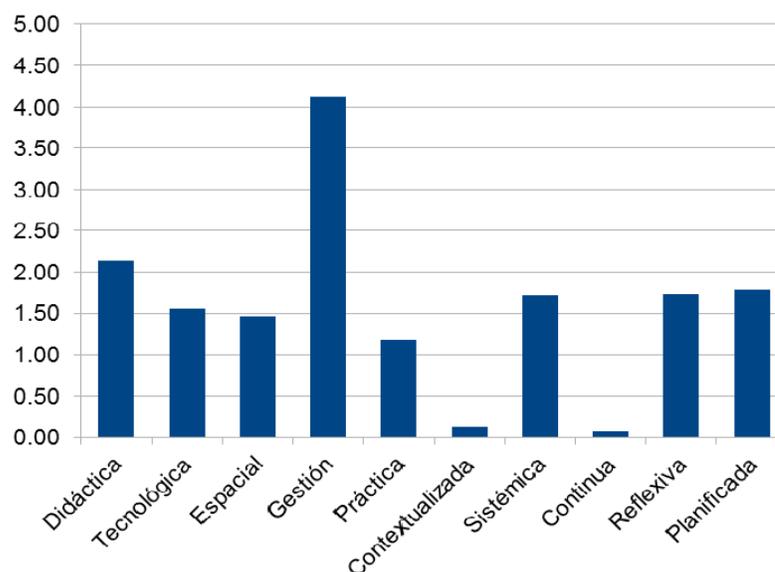


Figura 1. Valor de las subdimensiones de la variable de investigación

Sin embargo se corroboran como problemas:

- Limitaciones en la arquitectura del software, tales como la no utilización de un paradigma de programación específico de la Inteligencia Artificial y no está estructurado de manera que facilite el estudio y modificación de su código fuente.
- Ausencia de un diseño didáctico del software, que incluya las tareas generales a realizar por los estudiantes en el proceso de enseñanza-aprendizaje.
- Escaso aprovechamiento de las libertades del software libre en el proceso de enseñanza-aprendizaje.
- Deficiente contextualización de las tareas a realizar por los estudiantes con el software en el proceso de enseñanza-aprendizaje.

- Baja percepción del aporte del software utilizado a la autosuperación de los profesores y el no reconocimiento de su aporte al futuro trabajo profesional de los estudiantes.

Conclusiones del capítulo

El proceso de desarrollo de software educativo tiene como particularidad esencial su diseño didáctico a través de un documento que lo materialice. El guion de contenido será útil no solo en el desarrollo computacional del software, también en su incorporación al proceso de enseñanza-aprendizaje para el cual fue diseñado.

La periodización de la enseñanza de la Inteligencia Artificial permitió identificar como núcleos de conocimiento la representación del conocimiento, el tratamiento de la incertidumbre, el razonamiento y el aprendizaje automático, con el paradigma de programación para agente como tendencia a ser utilizado. Se evidenció como particularidad de los medios utilizados la inclusión de su código fuente como vía de interacción con el estudiante, pero con un limitado aprovechamiento de las libertades de estudiar, modificar y reutilizar el código fuente en el proceso de enseñanza-aprendizaje.

La integración del software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial se plantea como el proceso que incluye tanto el diseño didáctico del software libre educativo, llevado a cabo por el propio profesor; como su incorporación armoniosa al proceso de enseñanza-aprendizaje para el cual fue diseñado.

El proceso de operacionalización de la variable objeto de estudio permitió realizar una caracterización de la integración de software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial en la carrera Ingeniería en Ciencias Informáticas. Se reflejan potencialidades para llevar a cabo la integración, pero se aprecian insuficiencias que reafirman la necesidad de determinar los componentes y las relaciones esenciales que permitan la integración, desde un marco teórico hasta su materialización, con vistas a la transformación del proceso de enseñanza-aprendizaje de la Inteligencia Artificial teniendo al software libre educativo como medio.

2. MODELO DE INTEGRACIÓN DE SOFTWARE LIBRE EDUCATIVO AL PROCESO DE ENSEÑANZA-APRENDIZAJE DE LA INTELIGENCIA ARTIFICIAL

En el capítulo se presenta el modelo de integración de software libre educativo al proceso de enseñanza-aprendizaje. El componente conceptual con sus fundamentos y principios, el componente estructural con los elementos esenciales para la integración y el componente instrumental a partir de la estrategia didáctica que permite llevar a la práctica el modelo.

Se presentan las valoraciones sobre el modelo, el método Delphi para obtener criterio de expertos, una sistematización de experiencias del diseño didáctico de un software libre educativo y su incorporación armoniosa a una de las asignaturas de Inteligencia Artificial de la carrera Ingeniería en Ciencias Informáticas. Además, las técnicas de satisfacción de ladov y de grupos focales. Por último se presenta la triangulación metodológica.

2.1 Caracterización del modelo

Varios autores reconocen a los modelos como una representación de la realidad, que supone un alejamiento o distanciamiento de la misma. Dicha representación conceptual es simbólica, parcial y selectiva de aspectos de esa realidad (Gimeno, 1981) (Iriarte, 2007), (Pérez, 2015) y (Rodríguez-Travieso, 2017).

Valle (2012) especifica que un modelo es una representación de aquellas características esenciales del objeto y debe incluir cómo puede ser cambiado e implementado con vistas a la transformación de la realidad. Este autor propone los componentes generales que pueden asumir los modelos: Fin y objetivos, principios, caracterización del objeto de investigación, estrategia o metodología, formas de implementación y formas de evaluación.

Otros autores como Iriarte (2007) y Rodríguez-Travieso (2017) proponen características generales que deben tener los modelos y afirman que los componentes de un modelo pueden variar en dependencia de las necesidades del investigador. De igual forma Frigg y Hartmann (2018) asumen que la estructura del modelo está en estrecha relación con la naturaleza del objeto a representar.

Estos criterios son tomados en cuenta para presentar el modelo de integración de software libre educativo al proceso de enseñanza-aprendizaje de la IA a partir de los siguientes elementos:

Fin del modelo:

- Integrar software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial.

Objetivos:

- Establecer los fundamentos y principios para desarrollar el PEA de las asignaturas de IA mediante la integración de software libre educativo.
- Describir los principales elementos que intervienen en la integración, así como las relaciones esenciales que se establecen entre ellos.

- Delimitar las etapas y acciones que permitan desarrollar el proceso de integración bajo los principios declarados.

Caracterización del objeto:

Se determinaron tres componentes generales, los cuales responden a cada uno de los objetivos definidos en el modelo:

- **Componente conceptual:** Representa los principios que rigen el modelo y sus fundamentos teóricos y metodológicos.
- **Componente estructural:** Representa los elementos principales que intervienen en la integración del software libre educativo al PEA de la IA y sus relaciones fundamentales. Tiene una relación de subordinación al componente conceptual.
- **Componente instrumental:** Constituye la materialización de la integración y se expresa a través de una estrategia didáctica para la implementación y evaluación del modelo. Tiene una relación de subordinación al componente estructural.

Como se puede apreciar en la representación gráfica (Figura 2) cada uno de los componentes generales contribuyen a lograr el fin del modelo.

2.2.1 Componente conceptual

El componente conceptual garantiza la consistencia teórica del modelo. Aborda los fundamentos filosóficos, psicológicos, sociológicos, pedagógicos, didácticos, tecnológicos y jurídicos que sustentan al modelo. Estos fundamentos y las exigencias del proceso de integración determinan los tres principios declarados y que guían las relaciones entre los elementos del componente estructural y el componente instrumental.

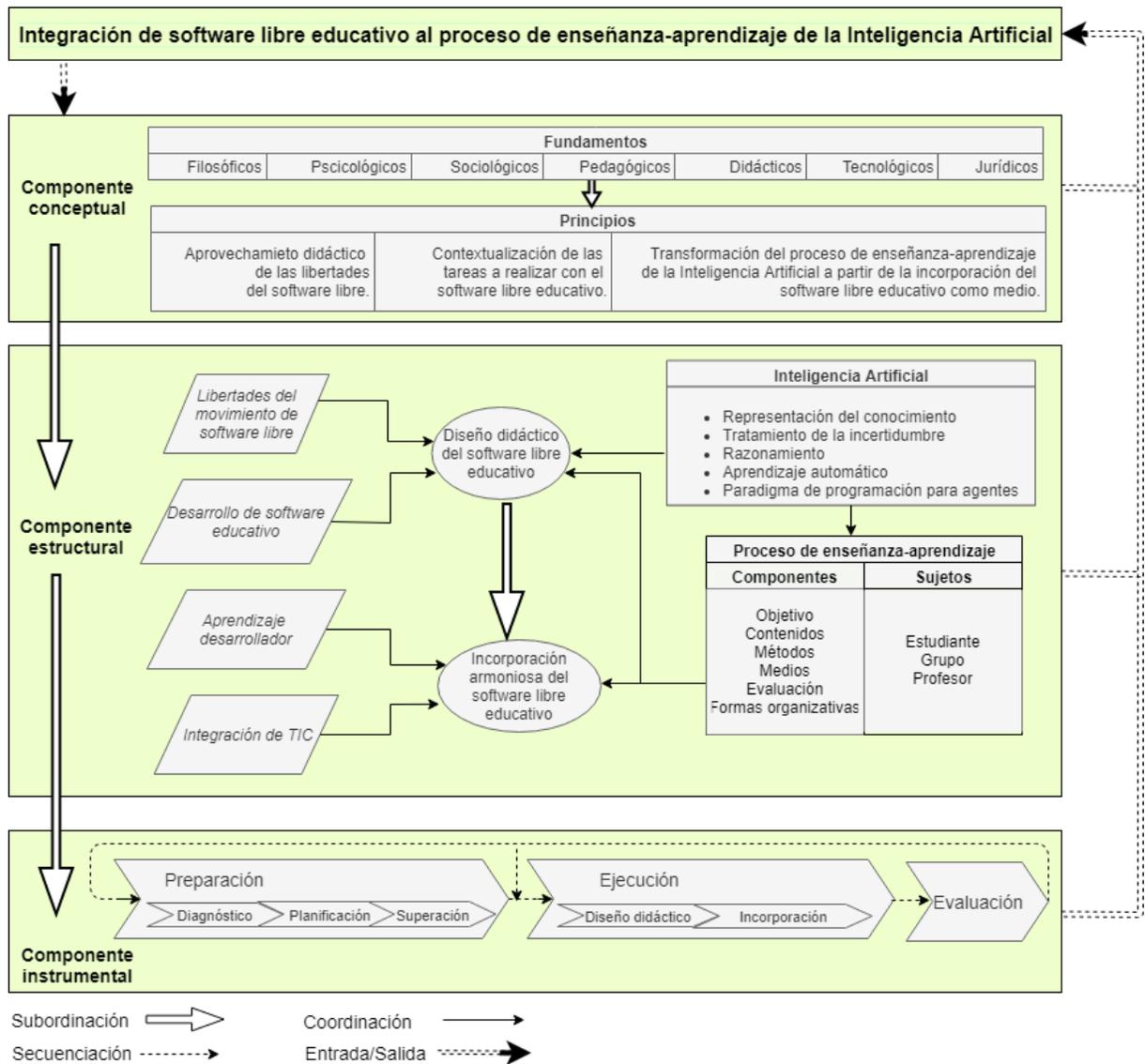


Figura 2. Representación gráfica del modelo

Fundamentos filosóficos

En base al enfoque dialéctico-materialista, toda la investigación y por tanto el modelo, parte de asumir una realidad objetiva que es llevada al plano mental y de ahí se materializa con una consecuente transformación de la realidad. Se consideran de

manera real y objetiva las relaciones existentes entre teoría y práctica y se toman en consideración las interacciones internas y la interdependencia causal entre los distintos componentes del objeto de investigación. Se analizan las contradicciones internas de cada componente y entre ellos.

De la filosofía de la educación se asume la relación íntima entre hombre y educación, así como la unidad entre teoría y práctica educativa en correspondencia con el sistema social (Chávez, Fundora, & Pérez, 2010).

Sociológicos.

La educación debe responder a las necesidades e intereses del entorno y los estudiantes. Las influencias entre la sociedad y la educación sólo pueden entenderse como una interacción recíproca, una interdependencia que se manifiesta compleja y diversa (Blanco, 1997). La escuela constituye una institución social organizada capaz de perpetuar, reproducir y transmitir la ideología y la cultura (Marx, 1875) y tiene el propósito de lograr individuos autónomos, capaces de pensar y actuar por sí mismos, lo cual se potencia con la solución de problemas de su entorno a través de las tareas con el software libre educativo.

Atendiendo a la proyección social del PEA, como proceso de socio-individualización en unidad dialéctica (Chávez, Suárez, & Permuy, 2003) y la necesidad de conjugar armónicamente las relaciones interpersonales en este proceso, se presta especial atención a la mediación tecnológica o instrumental del software libre educativo como elemento dinamizador de las mediaciones sociales y pedagógicas, y a las relaciones de interacción que propicia entre estudiante-software-profesor, a la vez que se abordan formas de organizar tareas colectivas y colaborativas.

Psicológicos.

Del enfoque histórico-cultural se reconoce que las funciones psíquicas superiores son consecuencia de un proceso de mediación a través de instrumentos, en condiciones de interacción social y dependiente de leyes histórico-sociales (Colectivo, 2016). La mediación según Vigotsky puede ser instrumental o social. La instrumental es la relación dialéctica entre sujeto y objeto (actividad). La social es la relación dialéctica entre sujeto y las demás personas (comunicación).

La integración del software libre educativo se hace efectiva mediante la relación entre el estudiante (sujeto) y el software libre educativo (objeto). Las tareas son definidas desde el diseño didáctico del software libre educativo y llevadas a la práctica con la incorporación armoniosa de este al PEA. En este proceso se trabaja la comunicación en las relaciones del estudiante con los demás estudiantes y con el profesor.

Pedagógicos

Se parte de considerar la pedagogía como ciencia (Chávez et al., 2003) teniendo como características esenciales la proyección social, la orientación humanística y el carácter transformador. Las relaciones entre educación e instrucción, formando una unidad dialéctica, que se debe materializar en todo proceso pedagógico de forma organizada y con una adecuada dirección, basada en programas y metodologías que propicien un desarrollo pleno de los participantes en dicho proceso. Esta unidad se expresa mediante la vinculación de la enseñanza con la vida, con tareas y problemas de la sociedad y del entorno.

Didácticos

Se parte de la didáctica como una de las ciencias de la educación. Se asumen las leyes, principios y categorías presentadas en el epígrafe 1.3. Se defiende la contextualización como un proceso lógico de desarrollo, que permite ubicarse en situaciones concretas de relevancia y actualidad en la asignatura, disciplina, en la sociedad, y que es usada como marco motivacional y conductor temático para la presentación, desarrollo y evaluación de los contenidos, con fines de aprendizaje (Addine, 2002).

Tecnológicos

La tecnología se asume como una práctica social que involucra conocimientos, destrezas, problemas organizacionales, valores e ideologías (Núñez, 1999). La ciencia y la tecnología son procesos sociales que deben aportar al desarrollo de la sociedad en su conjunto.

El software libre educativo es tratado como medio (González-Castro, 1986) dentro de un PEA cada vez más influenciado por las nuevas tecnologías. El movimiento de software libre se basa en una concepción liberadora y social de la tecnología. Defiende la socialización del conocimiento con beneficios compartidos, lo cual en el ámbito educativo desempeña un papel, no solo instructivo, sino educativo y de educación en valores.

Se toman como referentes fundamentales las investigaciones sobre la integración de las TIC al PEA, para llevar a cabo la incorporación armoniosa del SLE al PEA de la IA y que fueron presentadas en el epígrafe 1.3. De igual forma se asumen como referentes las investigaciones presentadas en el epígrafe 1.1 sobre el proceso de

desarrollo de SE, las que sirven de base para el diseño didáctico del SLE. Se reconocen los paradigmas de programación asociados a la IA y se asume el paradigma de programación para agentes como base para el desarrollo del software libre educativo. De igual forma se asumen las características de la IA como ciencia y su enseñanza, presentados en el epígrafe 1.2.

Jurídicos

Desde el punto de vista jurídico, se sustenta el modelo en las licencias creadas para el desarrollo de software libre. La familia de licencias GPL (FSF, 2019) brindan la posibilidad de estudiar, modificar y reutilizar el software desarrollado.

Por otro lado se toman en cuenta el conjunto de reglamentaciones del Ministerio de Educación Superior (MES) de Cuba y las indicaciones para la elaboración de los planes E (MES, 2016).

Principios del modelo de integración de SLE al PEA de la IA

Para determinar los principios del modelo se asume la posición de Valle (2012) cuando los define como las regularidades más generales y esenciales que caracterizan el proceso o fenómeno en estudio y que guían la dirección de la transformación de éste. Este autor afirma que los principios deben constituir un sistema donde no se derive uno de otro, su cantidad debe ser mínima pero en su totalidad pueden explicar la teoría.

La integración de SLE al PEA de la IA se regirá por tres principios. Los cuales están estrechamente relacionados, donde cada uno tiene su función específica que contribuye al proceso de integración como un todo.

1. Aprovechamiento didáctico de las libertades del software libre.

Se parte de la necesidad del diseño didáctico de una aplicación informática desarrollada bajo los términos del software libre. Este principio traza las pautas para la integración del SLE al PEA de la IA. Se basa en el aprovechamiento de las libertades desde el diseño didáctico del software hasta su incorporación armoniosa al PEA de la IA.

El aprovechamiento se expresa a través de la elaboración, asignación y control de tareas a los estudiantes, relacionadas con el estudio, modificación y reutilización del código fuente del software libre educativo utilizado en el PEA de la IA.

Dentro del PEA la libertad de estudiar el código permitirá utilizar partes del mismo en conferencias y realizar tareas escritas en clases prácticas. La libertad de modificar el código permitirá asignar tareas de familiarización a los estudiantes. La reutilización se expresa, en lo fundamental, en las clases de laboratorio integradoras y en las tareas extraclase orientadas a los estudiantes. Estas tareas son de mayor complejidad y requieren mayor esfuerzo del estudiante. Por su parte la libertad de distribución de las mejoras y modificaciones, será utilizada en el intercambio de resultados entre los estudiantes. Así se contribuye al desarrollo de valores de solidaridad, en tanto el estudiante brinda los resultados de su trabajo para el beneficio de sus compañeros.

2. Contextualización de las tareas a realizar con el software libre educativo.

Este principio se vincula directamente con el anterior ya que precisa aún más las características de las tareas a realizar por el estudiante con el software libre educativo. Estas estarán relacionadas en primer lugar con la propia asignatura y

disciplina, además, con el centro de desarrollo o línea de investigación donde se vincula el estudiante para desarrollar su PP. Otro elemento a tomar en cuenta son los intereses del propio estudiante.

La asignatura enmarcará el contenido y las habilidades específicas, sin embargo el alcance de las tareas estarán en el contexto que tributará los problemas profesionales necesarios para trabajar en el proceso de integración del SLE. La utilización de un diagnóstico sobre los intereses del estudiante propiciará criterios útiles para que el profesor asigne de manera personalizada las tareas a realizar con el software por cada estudiante.

3. Transformación del proceso de enseñanza-aprendizaje a partir de la incorporación del software libre educativo como medio.

Este principio, relacionado directamente con los anteriores, especifica el carácter transformador de la integración de SLE al PEA de la IA. El PEA se enriquece con la integración de SLE, logrando una transformación dialéctica. Mantiene sus características esenciales, pero incorpora al software como medio, para enriquecer cada uno de los componentes del PEA.

Este principio asume la posibilidad de un proceso de enseñanza-aprendizaje diseñado para un entorno real o virtual. Mediado por la utilización del software con la búsqueda y análisis de información, llevando a cabo la comunicación presencial o mediada por cualquier forma de la tecnología.

Este principio se materializará asumiendo un objetivo general derivado de los objetivos del modelo del profesional y de la disciplina. A partir de este se derivarán

los objetivos específicos y habilidades a lograr mediante la utilización del software libre educativo en la asignatura específica.

El contenido deberá estar sujeto al sistema de conocimientos de la asignatura, pero tendrá sus particularidades a partir de las tareas a realizar por cada estudiante.

Los métodos problémicos (Font & Paez, 2017) desempeñan un importante papel al trabajar con el software. Pero deben incluirse otros métodos orales e intuitivos (Labarrere & Valdivia, 1988) para trabajar con el software en cada una de las tipologías de clases y otras formas de organización del PEA.

El sistema de medios debe tener como centro el software libre educativo. Para su correcta utilización se deben utilizar, además de medios que soporten el mismo, como computadoras de escritorio, laptop o teléfonos, medios tradicionales como la pizarra y los libros de texto. El uso de los medios de los propios estudiantes brindará la posibilidad de un mayor acercamiento entre el SLE y el actuar diario del estudiante.

Por último la evaluación debe tomar en cuenta el desarrollo del estudiante, de la misma forma, se debe tomar en cuenta el diagnóstico y el trabajo sistemático llevado a cabo por este. Todo el trabajo debe verse reflejado en la nota final del estudiante.

2.2.2 Componente estructural

El componente estructural está formado por ocho elementos agrupados en tres grupos. El grupo elementos centrales está conformado por el diseño didáctico del software libre educativo, al cual se subordina la incorporación armoniosa de este al PEA de la IA.

El grupo de los referentes fundamentales incluye las libertades del movimiento de software libre y el desarrollo de SE en relación de coordinación con el diseño didáctico del software libre educativo. El aprendizaje desarrollador y la integración de TIC, en relación de coordinación con la incorporación armoniosa del SLE.

El grupo elementos básicos específicos se constituye de los componentes y sujetos del PEA en relación de coordinación tanto con el diseño didáctico como con la incorporación armoniosa. Las características a tomar en cuenta de la IA, sus núcleos de conocimiento y el paradigma de programación específico asumido. Este elemento en relación de coordinación con el diseño didáctico y con el proceso de enseñanza-aprendizaje (Figura 3).

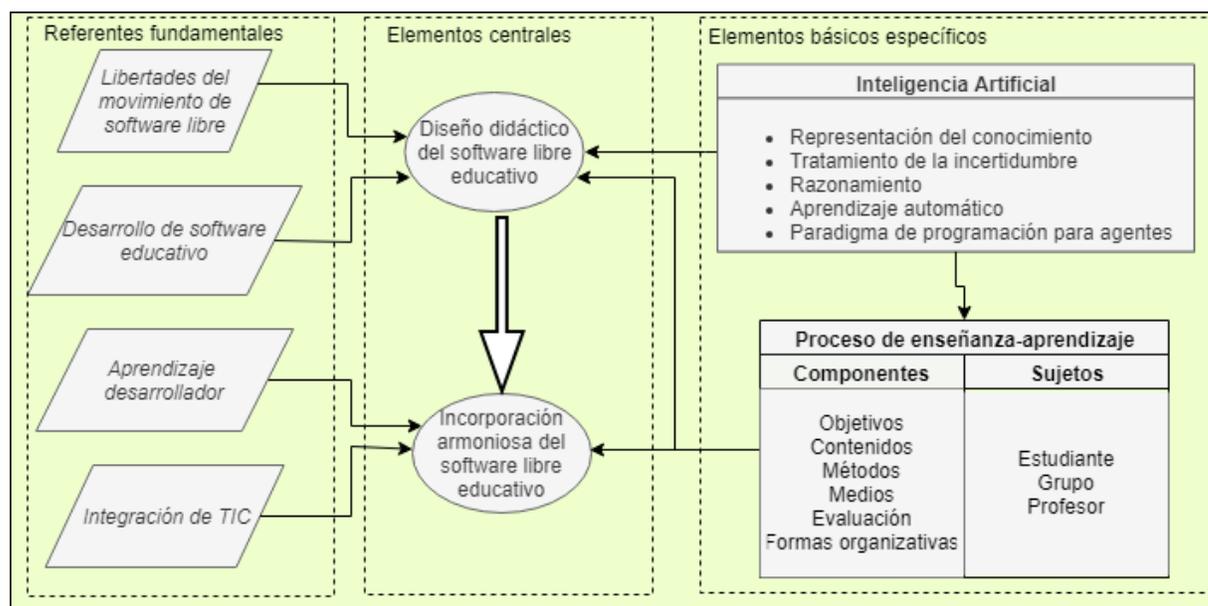


Figura 3. Componente estructural

El **diseño didáctico del software libre educativo** abordado en el epígrafe 1.1 es el primero de los elementos centrales del modelo. En él se ponen de manifiesto los tres principios del modelo. El principio de aprovechamiento didáctico de las libertades del

software libre se manifiesta en el diseño de las tareas de estudio, modificación y reutilización del código fuente. Este principio se garantiza a partir de la utilización de licencias asociadas al movimiento de software libre y con una arquitectura del software que brinde facilidades para estudiar, modificar y reutilizar el código fuente.

La contextualización de las tareas a realizar con el software libre educativo se manifiesta a la hora de tomar en cuenta la asignatura en la cual se va a incorporar para especificar las relaciones del software con cada uno de los componentes del PEA. Se garantiza a partir de una arquitectura del software que permita la resolución de tareas genéricas, que permitan la contextualización de las mismas a problemas diversos.

El principio de transformación del proceso de enseñanza-aprendizaje de la Inteligencia Artificial a partir de la incorporación del software libre educativo como medio, se manifiesta al especificar la relación del software con cada uno de los componentes del PEA. Se garantiza con el guion de contenido, donde se plasma todo el diseño del software, necesario para su desarrollo computacional y su incorporación al PEA para el cual fue diseñado.

El diseño didáctico asume como referentes fundamentales las investigaciones sobre el desarrollo de software educativo, específicamente sobre su diseño didáctico y el guion de contenido como documento que lo materializa. Incorpora como particularidad el diseño dirigido a aprovechar las libertades del software libre, lo cual constituye otro referente fundamental.

El movimiento de software libre es abordado en el epígrafe 1.1. La defensa de las libertades del software libre contribuye al diseño didáctico del software libre

educativo. Se trabajan desde el punto de vista pedagógico y se hace énfasis en las libertades para lograr tareas en los estudiantes de estudiar, modificar y reutilizar el código fuente del software.

Estudiar el código fuente se sustenta íntegramente en la libertad uno. Richard Stallman, el creador del movimiento de software libre, defiende la utilidad de estudiar el código fuente para formar buenos desarrolladores de software, un objetivo esencial de la carrera Ingeniería en Ciencias Informáticas.

Modificar el código fuente también se sustenta en la libertad uno y adicionalmente la tres, la cual defiende la posibilidad de mejorar el software. Con estas libertades se ha garantizado el desarrollo de este movimiento y lograr resultados de un alto valor, comparables con el software privativo. Se defiende como una vía para relacionarse con previos y futuros desarrolladores del mismo software, haciendo sentirse parte de un resultado común.

Por otro lado, deben abordarse las cuestiones éticas de reconocer el trabajo previo realizado y siguiendo la libertad tres distribuir los nuevos aportes para que los demás se beneficien. Este aspecto contribuye a la formación en valores no solo como buenos profesionales, sino como profesionales integrales, a tono con los principios de la sociedad socialista.

Este tipo de tarea en los estudiantes desarrolla habilidades en cuanto a la programación y suelen tener un nivel de complejidad mayor a las de estudiar el código fuente.

Reutilizar el código fuente se relaciona con las cuatro libertades. La libertad cero permite ejecutar el programa con el objetivo de analizarlo, modificarlo y reutilizarlo.

La libertad uno permite estudiar y adaptarlo. La libertad dos para distribuir copias para el trabajo en equipo, como casi siempre es el desarrollo de un software. Por último la libertad tres permite mejorar el programa, en este caso desarrollar uno nuevo basado en el anterior o reutilizando parte del mismo y distribuirlo para que toda la comunidad se beneficie.

La reutilización de código es la tarea de mayor complejidad para los estudiantes. Desarrolla habilidades como desarrolladores de software, específicamente para reutilizar algoritmos ya implementados en bibliotecas de IA.

En el diseño didáctico se toman como elementos básicos específicos las particularidades de la IA, abordadas en el epígrafe 1.2. Se asume la misma como la ciencia de la computación encargada de aplicar métodos de representación, procesamiento y extracción del conocimiento, mediante programación multiparadigma, en el desarrollo de sistemas informáticos con comportamiento racional. En los currículos se presenta, generalmente, como una asignatura, aunque se ha trabajado como disciplina o conjunto de asignaturas dentro de una disciplina que incluye elementos de matemáticas aplicadas y discretas. En todos los casos analizados se imparte una asignatura con los elementos esenciales de la IA y que justifican la propuesta que se presenta.

La representación del conocimiento incluye las formas clásicas (reglas, redes semánticas, lógica, basada en casos entre otras), formas más recientes y avanzadas como las ontologías y formas no convencionales que logren estructurar el conocimiento de manera que sea interpretado por la computadora y fácilmente accesible por los usuarios.

El tratamiento de la incertidumbre incluye en lo fundamental el tratamiento probabilista y la lógica difusa. El razonamiento incluye métodos de búsqueda a ciegas y heurísticos, así como otras formas de razonamiento lógico e inferencias. En tanto el aprendizaje automático puede verse desde el punto de vista supervisado, no supervisado y reforzado; para la construcción de modelos, el entrenamiento o la extracción de conocimiento.

Estos núcleos deben estar representados de manera total o parcial en el diseño didáctico del software y relacionados directamente con las características de la asignatura de IA en la cual se va a incorporar.

De igual forma la asignatura responderá a estos núcleos desde su diseño curricular, lo cual no es objetivo de la presente investigación, pero se reconoce como necesario para que se corresponda con una asignatura de IA.

El paradigma de programación para agentes guiará el desarrollo del software libre educativo. Puede llevarse a cabo el desarrollo de un software asumido como un agente o como un sistema multiagente. El primer caso asume las características básicas de una aplicación informática, desde la perspectiva de un agente racional con percepciones, razonamiento y acciones sobre un entorno específico (Russell & Norvig, 2010). En el segundo caso se incluirá la utilización de alguna metodología de desarrollo de sistemas multiagentes y alguna plataforma como JADE (Telecom Italia, 2019) para llevar a cabo el sistema.

Los componentes del PEA tributan al diseño didáctico, desde el programa de la asignatura para la cual se va a diseñar el SLE. Se enriquece con la especificación de cada una de las características del software, dirigidas a su incorporación al propio

PEA, con el fin de su transformación. Esta incorporación no solo contribuye a sus componentes, también a los sujetos que intervienen en el proceso a partir de la especificación de tareas dirigidas a los estudiantes y al grupo de forma general.

El profesor desempeña un papel determinante, al diseñar el software para ser incorporado al PEA, y dirige en dicha incorporación, su utilización por el estudiante y el grupo.

La **incorporación armoniosa del SLE al PEA de la IA** es el segundo de los elementos centrales y se subordina al diseño didáctico del SLE. Esa subordinación se expresa a partir de la utilización del guion de contenidos, donde están plasmadas las características del software libre educativo y las orientaciones específicas, incluidas las tareas generales a asignar a los estudiantes.

En este elemento se ponen de manifiesto los tres principios del modelo. El principio de aprovechamiento didáctico de las libertades del software libre se manifiesta en la reutilización de resultados de un tema a otro de la asignatura o de un curso a otro mediante el perfeccionamiento del SLE. Se garantiza este principio, a partir de la asignación, seguimiento y control de tareas contextualizadas donde el estudiante necesite estudiar, modificar y reutilizar el código fuente.

La contextualización de las tareas a realizar con el software libre educativo se manifiesta en las tareas a realizar por los estudiantes relacionadas con sus intereses y con su PP. Se garantiza desde la planificación de las tareas.

El principio de transformación del proceso de enseñanza-aprendizaje de la Inteligencia Artificial a partir de la incorporación del software libre educativo como medio, se manifiesta en varios sentidos. La contextualización de las tareas a realizar

por los estudiantes va dirigida a la transformación del estudiante como sujeto activo, donde alcanza su protagonismo al identificar problemas de su entorno y contribuir a su solución desde el trabajo con el SLE.

La incorporación del SLE al PEA, desde un enfoque sistémico, aumenta su relación general con la asignatura y su impacto en el PEA. El SLE, como componente del sistema, abarca todo el PEA en su relación con los componentes y sujetos.

La reutilización de resultados de los estudiantes permite perfeccionar el SLE y con él el PEA. Los estudiantes vuelven a ser protagonistas al contribuir al medio que será utilizado, en próximos cursos, por otros estudiantes. Este protagonismo, dirigido por el profesor tanto en la elaboración de tareas generales en el diseño didáctico del SLE, hasta su incorporación armoniosa al PEA, contribuirá a incrementar la percepción de profesores y estudiantes sobre la importancia del software y su utilización en el PEA.

La correcta planificación de los elementos más relacionados con la incorporación armoniosa garantiza la transformación del PEA. Se alcanzarán con ella los objetivos propuestos, tanto en el PEA, como en la integración del SLE al PEA de la IA de forma general.

La incorporación armoniosa del SLE al PEA asume como referentes fundamentales los fundamentos para un aprendizaje desarrollador presentados en el epígrafe 1.3.

El trabajo con el software se estructura de manera que se oriente al estudiante la búsqueda de problemas relacionados con su PP y que puedan ser resueltos con el software. Se planifican actividades de seguimiento a las tareas para aclarar dudas y controlar el avance de los resultados. Para llevar a cabo las tareas de estudio y

modificación del software, se diseñan tareas que permitan que el estudiante identifique las características del software que más se relacionan con su trabajo en la PP y profundice en los aspectos que sean de su interés y necesidad, para dar cumplimiento a las tareas más complejas de reutilización del código fuente.

La sistematicidad en el trabajo con el software libre educativo y mantener tareas con el mismo durante todo el PEA, contribuirá a su motivación y a la necesidad de aprender y buscar vías para hacerlo.

El software libre educativo para la IA, incluirá en su desarrollo los núcleos definidos de representación del conocimiento, tratamiento de la incertidumbre, razonamiento y aprendizaje automático. Estos son conceptos generales del nivel teórico que el estudiante debe ir particularizando en el propio estudio con el software, identificando las particularidades de cada uno y encontrando la relación de los mismos con sus necesidades cognitivas, para darle solución a las tareas. De esta manera se trabaja el desarrollo de los procesos lógicos del pensamiento.

El software libre educativo se considera un medio activo, un espacio de creación virtual (EDUCAUSE, 2018) donde el estudiante puede comprobar mediante la actividad sus conocimientos teóricos. Esto convierte al proceso en eminentemente activo y desarrollador. El estudiante resolverá tareas de forma individual, otras en equipos pequeños y otras de manera grupal. Las tareas individuales y por equipo serán debatidas, potenciando la comunicación en el grupo. Se aportará de esta manera a soluciones de otros equipos.

Desde la propia asignación de tareas personalizadas, se atienden las diferencias individuales de los estudiantes. En las actividades definidas por el profesor para dar

seguimiento a las tareas asignadas, se brindará atención a las dificultades que va encontrando cada estudiante. De igual manera con el desarrollo de las tareas en equipo y grupales, todos aportan al tránsito efectivo del nivel inicial del estudiante hacia el que se aspira.

Las tareas más complejas están relacionadas con la reutilización del código del software libre educativo, el cual tiene como objetivo dar respuesta a un problema relacionado con su PP. Este tipo de tarea estimula la valoración que realiza el estudiante del propio proceso y su aporte a la práctica social.

De la integración de tecnologías de la información y la comunicación se asumen sus cualidades, las cuales se constituyen en subdimensiones para el análisis de la incorporación armoniosa y que fueron abordadas en el epígrafe 1.3 y epígrafe 1.4.

Del PEA tributan a la incorporación armoniosa sus componentes y sujetos, los cuales fueron abordados en el epígrafe 1.3. A cada uno de ellos tributan las características de la IA, como asignatura específica.

2.2.3 Componente instrumental

El componente instrumental permite materializar la integración de SLE al PEA de la IA y se expresa en una estrategia con tres etapas generales (Figura 4).

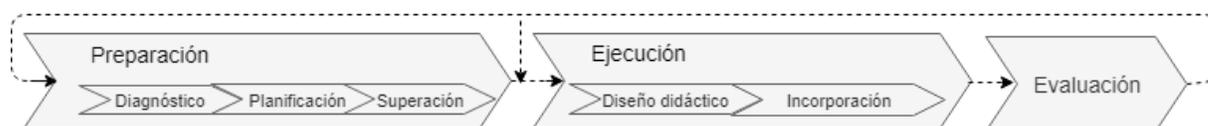


Figura 4. Componente instrumental del modelo

Etapa I. Preparación

Fase I.I: Diagnóstico.

Esta etapa permitirá diagnosticar el estado de la integración del software libre educativo al PEA de la IA. Para ello se utilizarán los resultados del proceso de operacionalización de la variable de estudio presentado en el epígrafe 1.4. Las acciones específicas que se proponen son:

Adecuación de los instrumentos: Permitirá adecuar los instrumentos según las características y necesidades del profesor.

Aplicar los instrumentos: Se seleccionará la muestra por cada unidad de estudio y se aplicarán los instrumentos.

Procesar la información: Esta acción permitirá al profesor tomar decisiones importantes en cuanto a los aspectos con mayores dificultades para planificar el diseño didáctico del software a incorporar al PEA de la IA.

Fase I.II: Planificación

La planificación juega un papel importante para lograr la organización del trabajo a realizar. Las acciones específicas son:

Preparación de un diagnóstico a los estudiantes. En el diagnóstico se indagará sobre el centro o línea de investigación donde el estudiante desarrolla su PP, sus intereses relacionados con la disciplina y sus gustos o aspiraciones relacionadas con su profesión y la asignatura. Además, su nivel de interés por desarrollar software y las habilidades básicas en la programación.

Adecuación del plan calendario. Aunque la adecuación no sea significativa, se planificará en el plan calendario de la asignatura las tipologías de clases y los

objetivos de las que incluirán el trabajo con el SLE. Se incluirá la planificación de horas de trabajo con el software en las clases y las horas inducidas.

Planificación del diseño didáctico del software. Esta acción es de suma importancia. Primeramente se planificará la búsqueda de algún software libre que responda a las características de la asignatura. De existir, se procederá a planificar su diseño didáctico y posible adecuación computacional.

En el caso que no exista un software libre con las características deseadas, se procederá a planificar el diseño didáctico, seleccionar una metodología de desarrollo de software y llevar a cabo su análisis y desarrollo. Se recomiendan metodologías ágiles en tanto una característica esencial de este tipo de software es su complejidad baja o media, de manera que sirva de ejemplo y estudio para los estudiantes.

Planificación de vías de retroalimentación del proceso. Se planificarán vías de retroalimentación como pueden ser: encuestas, entrevistas, guías de observación, técnicas de satisfacción.

Fase I.III: Superación

A partir del diagnóstico y planificación, será necesario una fase de superación del claustro, para la generalización de la integración. En esta etapa se brindan las herramientas para que los profesores se incluyan en el proceso con los conocimientos y habilidades básicas necesarias. Las acciones concretas son:

Organización de las actividades de superación necesarias. En base al diagnóstico realizado y la planificación del proceso se organizarán actividades que van desde el sistema de trabajo metodológico de la asignatura, hasta posibles cursos de postgrado. Se garantizará la preparación del claustro para enfrentar el proceso de

integración. Esta superación estará encaminada tanto a elementos teórico-metodológicos como de contenido de la ciencia en cuestión.

Ejecución de las acciones de superación. Se ejecutarán en el tiempo que sea necesario, en concentrados metodológicos, cursos de postgrado o a través de planes de trabajo metodológicos de la disciplina o asignatura. Se organizarán las actividades centrales a nivel de carrera y universidad y a nivel de facultad.

Etapas II. Ejecución

Fase II.I. Diseño didáctico del software libre educativo.

Esta fase define sus acciones concretas a partir de las subdimensiones definidas para el estudio de la integración de software libre educativo al PEA de la IA. Va dirigida a transformar la realidad, contribuyendo a cada uno de los indicadores declarados en la presente investigación. Los resultados de esta fase se plasmarán en un guion de contenido que guiará el proceso de análisis, diseño y desarrollo computacional del software y será herramienta de trabajo metodológico para el profesor en la incorporación del SLE al PEA. Las acciones específicas son:

Descripción de las relaciones entre el software libre educativo y cada uno de los componentes de la didáctica (Asociada a la subdimensión didáctica). Se declararán los objetivos, relacionados con la asignatura, que serán tratados con el software libre educativo, los contenidos a incluir, indicaciones de los métodos para trabajar el software en cada una de las formas de organización del PEA. Orientaciones para utilizar el SLE con los medios disponibles, así como la forma de evaluar al estudiante con el SLE.

Elaboración de las tareas a realizar en los diversos espacios disponibles (Asociada a la subdimensión espacial). Se elaborarán las tareas generales a desarrollar por los estudiantes de forma individual, por equipos o en el grupo. Estas pueden estar dirigidas para el trabajo tanto en el aula, como en el laboratorio, en plataformas de enseñanza-aprendizaje o incluso en los propios dispositivos de los estudiantes.

Especificación de las características del software para garantizar su distribución y utilización en el proceso de enseñanza-aprendizaje (Asociada a la subdimensión de gestión). Se determinará la licencia que tendrá el software, se especifican, a partir de las condiciones reales, las características que debe poseer el software. Además, se especificará la forma de distribuirlo a través de plataformas disponibles y cómo crear las condiciones para hacerlo.

Elaboración de tareas que aprovechen las libertades del software libre (Asociada a la subdimensión práctica). Se elaborarán las tareas a realizar por los estudiantes para estudiar, modificar y reutilizar el código fuente del software libre educativo.

Análisis, diseño y desarrollo computacional del software libre educativo (Asociada a la subdimensión tecnológica). Primeramente se determinarán las características generales del software que garanticen la legibilidad del código fuente por el estudiante, como puede ser el lenguaje de programación a utilizar, la definición de la arquitectura del software y elaborar una adecuada documentación de ayuda. A partir de los elementos ya plasmados en el guion de contenido, se procederá al

análisis, diseño e implementación siguiendo alguna metodología de desarrollo de software.

Fase II.II. Incorporación del software libre educativo al proceso de enseñanza-aprendizaje de la asignatura de IA. En esta fase se tomará en cuenta lo especificado en el guion de contenido.

Diagnóstico inicial a los estudiantes. Se aplicará el diagnóstico ya planificado por el profesor. Sus resultados serán tomados en cuenta en la asignación de las tareas a cada estudiante.

Presentación del software libre educativo. En una de las clases iniciales se presentará a los estudiantes el software con que se trabajará. Como parte de la presentación se pueden orientar tareas para el estudio de su arquitectura y principales componentes.

Familiarización de los estudiantes con el software. Se orientará en clases prácticas o clases de laboratorio, tareas de modificación del código fuente a los estudiantes. Estas ayudarán al estudiante a familiarizarse con el código fuente del SLE.

Asignación de tareas personalizadas a realizar con el software. Luego de estar familiarizados con el software y conocer los elementos básicos sobre la asignatura, ya los estudiantes estarán en condiciones de identificar un problema en su entorno para resolver con el SLE. El problema puede estar relacionado con su PP, con otras asignaturas del año o con sus propios intereses.

Durante un tiempo, que puede oscilar entre una semana o dos, el profesor analiza todas las propuestas de problemas, los adecua a las características de la asignatura

y los asigna a cada estudiante. El profesor tomará en cuenta los resultados del diagnóstico aplicado, para la especificación de las tareas a todos los estudiantes, incluidos los que no encontraron problemas o los que presentaron algún problema que no se corresponde con los objetivos de la asignatura.

Seguimiento a las tareas personalizadas a realizar con el software. El profesor dará seguimiento a las tareas a partir de las clases de laboratorio, aprovechará los encuentros para aclarar dudas y brindar las ayudas que necesite el estudiante para llevar a cabo su tarea. Adicionalmente se crearán espacios de aclaración de dudas, estos serán reales o virtuales, en plataformas de enseñanza-aprendizaje, redes sociales u otros similares.

Evaluación final al estudiante. La evaluación estará en función del desarrollo del estudiante. Incluirá criterios para la evaluación sistemática, parcial y final, a partir del trabajo con el SLE.

Etapas III. Evaluación

Ya concluida la asignatura el profesor realizará un conjunto de acciones que permitan evaluar y dar continuidad al proceso.

Análisis de resultados de los estudiantes. Se evaluarán cada una de las soluciones llevadas a cabo por el estudiante. Las que cumplan con los requerimientos básicos necesarios serán entregadas a los responsables de la PP o de los centros de desarrollo para valorar su inclusión como resultados del centro.

Evaluar el posible rediseño del software. Las soluciones de los estudiantes aportarán elementos para perfeccionar el SLE, ya sea un nuevo algoritmo o la

modificación de alguno de los ya implementados. El profesor decidirá los elementos a incluir en el software con vistas a su perfeccionamiento para los cursos siguientes.

Análisis de resultados del proceso. Se analizará todo el proceso desde la etapa de preparación y cada una de sus acciones hasta los resultados finales. Este análisis incluirá el informe final de la asignatura, la promoción del grupo y la calidad de las calificaciones. Se utilizarán formas de obtener información como pueden ser encuestas a los estudiantes o debate directo con ellos.

Retroalimentación del modelo. Después de analizar cada uno de estos elementos se tomarán decisiones sobre cada uno de los elementos dinámicos del modelo, dígase el SLE, el programa de la asignatura, cada uno de los componentes del PEA, o cualquier otro elemento de interés que se derive de la aplicación del modelo.

2.2 Valoraciones sobre el modelo

En este epígrafe se presenta el proceso llevado a cabo para obtener valoraciones sobre el modelo. Primeramente se aplicó el método Delphi para valorar el modelo por criterios de expertos. Posteriormente, se llevó a cabo una sistematización de experiencias para el diseño didáctico y la incorporación armoniosa de un software libre educativo al PEA de la asignatura IA-II de la carrera Ingeniería en Ciencias Informáticas. Se aplicó la técnica de satisfacción de Ladov a los estudiantes involucrados en la sistematización de experiencias. Los resultados alcanzados hasta ese momento fueron presentados a grupos focales que aportaron criterios con vistas a la generalización del modelo. Finalmente se llevó a cabo una triangulación

metodológica para llegar a conclusiones sobre la consistencia teórica, pertinencia, aceptación y aplicabilidad del modelo.

2.2.1 Valoración por criterio de expertos

La aplicación del método Delphi en las ciencias pedagógicas se ha generalizado, por las posibilidades que ofrece de obtener opiniones sobre la validez de resultados teóricos. Este método permitió obtener de un grupo de expertos, opiniones que se fundamentan científicamente, sobre los fundamentos, principios, componentes y sus relaciones, etapas de la estrategia y otros elementos del modelo.

Se plantearon como requisitos para la selección de los expertos que contaran con más de 10 años de experiencia en la enseñanza en la educación superior y alguna experiencia en utilización de tecnologías para la educación. Se valoró como importante contar con una categoría docente de Profesor Asistente o superior, y que fueran Máster o Doctor. Como requisito principal se tomó en cuenta que estuvieran dispuestos a participar.

El criterio de expertos es valorado como una fuente confiable, basada en la evaluación de la competencia de los candidatos a partir de la determinación del coeficiente de desempeño (k) que tiene en consideración su autovaloración. La expresión matemática que lo representa refiere que:

$$k = 1/2 (k_c + k_a) \text{ donde:}$$

k_c : coeficiente de conocimientos, basado en la valoración particular que tienen los candidatos sobre sus conocimientos del tema.

k_a : coeficiente de argumentación, basado en la valoración sobre las fuentes de argumentación, teniendo en consideración los niveles de influencia.

En la bibliografía se pueden apreciar diversidad de criterios para determinar las fuentes de argumentación. En la presente investigación se proponen las que se muestran en la Tabla 5.

Tabla 5. Influencia de las fuentes de argumentación

Fuente de argumentación		Grado de influencia		
		Alto	Medio	Bajo
1	Investigaciones teóricas o experimentales realizadas por usted	0.25	0.19	0.125
2	Experiencia obtenida en la actividad profesional (Impartición de docencia utilizando software)	0.4	0.33	0.2
3	Experiencia obtenida en la actividad profesional (Docencia de postgrado recibida sobre utilización de TIC)	0.2	0.16	0.1
4	Análisis de literatura o publicaciones de autores nacionales	0.05	0.04	0.025
5	Análisis de literatura o publicaciones de autores extranjeros	0.05	0.04	0.025
6	Intuición	0.05	0.04	0.025
	Suma de valores	1	0.8	0.5

Se considera que los valores del coeficiente de competencia oscilan entre $0 \leq k \leq 1$.

Valorando los niveles de experticia de la siguiente forma:

- Alto ($0,8 < K \leq 1$)
- Medio ($0,5 < K \leq 0,8$)
- Bajo ($K \leq 0,5$)

Para el proceso de evaluación se envió un cuestionario de evaluación (Anexo 15. a) a un total de 30 posibles expertos, de los cuales respondieron 18. Se tuvo como

resultados la selección final de 17 expertos ya que uno obtuvo nivel bajo de competencia. Los seleccionados fueron 10 con nivel alto y siete con nivel medio (Anexo 15. b).

Una vez seleccionados los expertos se envió para su consideración un resumen del modelo y un cuestionario donde debieron evaluar, en una escala de Likert de cinco, ocho aspectos relacionados con el modelo (Anexo 15. c).

Luego de la primera ronda (Anexo 15. d) se puede apreciar que la media de las preguntas cuatro y siete fueron las más altas. Siendo la pregunta cinco la de mayor desviación. En la Figura 5 se muestran los valores emitidos por cada experto a cada pregunta.

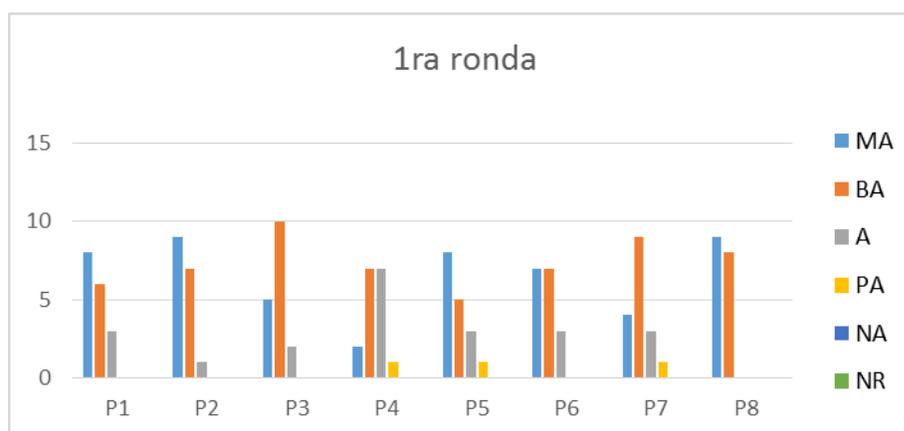


Figura 5. Valores de cada pregunta en la primera ronda.

Las sugerencias hechas en respuesta al cuestionario en la primera ronda se dieron a conocer en una segunda ronda a los expertos anónimamente, junto a los arreglos realizados al modelo. Los resultados alcanzados en esta segunda ronda se muestran en el Anexo 15. e). Como se muestra en la Figura 6, el 100% de los expertos considera muy adecuado o bastante adecuado cada aspecto del cuestionario.

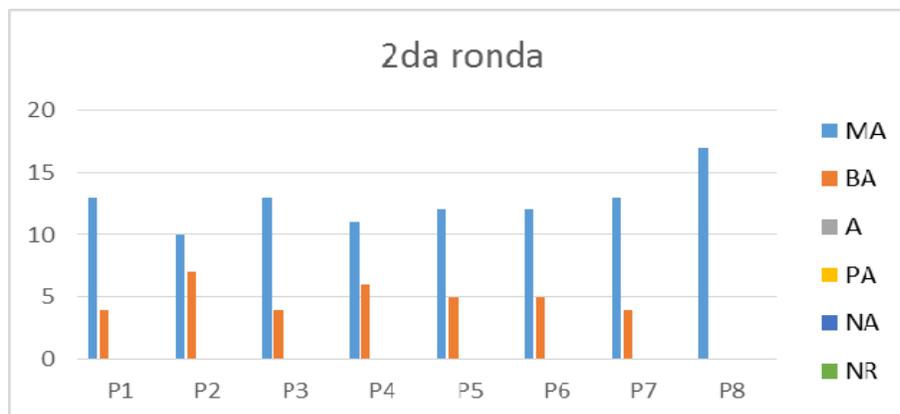


Figura 6. Valores de cada pregunta en la segunda ronda.

El 100% considera muy adecuado el aspecto ocho, referido a la relevancia y pertinencia del modelo. Mientras más del 70% considera muy adecuado los aspectos uno, tres, cinco, seis y siete. Los aspectos dos y cuatro alcanzan más de un 50% de criterios de muy adecuado.

A partir de la evaluación que otorgaron los expertos a cada aspecto y los criterios emitidos, se llevaron a cabo un conjunto de mejoras al modelo, siendo las más significativas:

1. Inclusión de la representación de los referentes fundamentales en el componente estructural.
2. Hacer más explícito en el gráfico las dimensiones de la integración de software libre educativo al PEA de la IA.
3. Incluir las particularidades de la IA y los componentes del PEA.

2.2.2 Sistematización de experiencias

La sistematización de experiencias en educación, examina la reconstrucción de experiencias, la interpretación crítica como proceso de reflexión que ordena y

relaciona datos, hechos, modos de actuar para construir aprendizajes que conllevan a mejores tomas de decisiones en el ámbito educativo (Pavón, 2017).

La sistematización refiere a aquellas acciones o experiencias que se derivan de intervenciones intencionadas y con objetivos claros, en la que los actores involucrados participan del resultado esperado. En el contexto universitario se reconocen dentro de las fuentes de experiencias las estrategias de enseñanza y la producción de material didáctico (Pérez de Maza, 2016), las cuales se corresponden con la presente investigación.

Varios autores han propuesto pasos o momentos para llevar a cabo la sistematización de experiencias (Berrutti, Cabo, & Dabezies, n.d.), (Pérez de Maza, 2016), (Jara, 2018). A partir de los puntos coincidentes de estos autores y las características de la investigación, se organizó el proceso de sistematización de experiencias de la siguiente manera:

1. **Planificación de la sistematización:** Incluyó la especificación de los participantes, el objetivo, el objeto, los ejes de sistematización y los pasos a seguir.
2. **Recuperación del proceso vivido:** Se realizó una reconstrucción histórica y se ordenaron fuentes de información.
3. **Reflexión crítica:** Se realizó una interpretación crítica y se compararon los resultados con los obtenidos en el análisis empírico al inicio de la investigación.
4. **Cierre:** Se especificaron conclusiones a partir de las experiencias y se elaboraron productos de comunicación de resultados.

1. Planificación de la sistematización

Los participantes de la experiencia fueron un recién graduado en adiestramiento (RGA), dos estudiantes de 5to año de la carrera Ingeniería en Ciencias Informáticas y el investigador. La sistematización fue llevada a cabo por el investigador con la colaboración de los demás participantes. El RGA participó como ayudante en el proceso de diseño didáctico y en las clases de la asignatura. Los estudiantes llevaron a cabo el desarrollo computacional del SLE, tutorados por el investigador, quien, además, fue el profesor de la asignatura IA-II, donde fue incorporado el SLE.

El objetivo de la sistematización fue analizar una práctica concreta de integración de SLE al PEA de una de las asignaturas de IA de la carrera Ingeniería en Ciencias Informáticas. El investigador previamente ha desarrollado otras aplicaciones informáticas para asignaturas de IA (Coca, Rosell, & Velazquez, 2017), (Coca & Pérez, 2018) y lleva más de ocho años incorporando software donde los estudiantes estudian y modifican el código fuente como parte de las tareas en asignaturas relacionadas con la IA (Coca, 2011), (Coca, 2015) (Coca, 2016). La experiencia sistematizada se corresponde con el diseño didáctico del software Sistema Multiagente para la Extracción de Conocimiento en Bases de Datos (SMEKDB) y su incorporación armoniosa al PEA de la asignatura IA-II en el curso 2017-2018.

Como eje de la sistematización se tomaron las acciones definidas en el componente instrumental del modelo. Se analizaron cada uno de los artefactos generados de la metodología de desarrollo de software utilizada. Se analizó el guion de contenido elaborado para el software, los documentos presentados en los talleres de trabajo de diploma de los estudiantes, así como los planes de clases del profesor.

2. Recuperación del proceso vivido:

Para llevar a cabo el diseño didáctico del software libre educativo, se elaboró un perfil de trabajo de diploma donde se especificaron los objetivos generales del software a desarrollar. Se seleccionaron dos estudiantes de 5to año de la Facultad 4, con buenos resultados en las asignaturas de IA y con interés de trabajar en la investigación.

En los primeros encuentros se explicó el trabajo, se aclararon las dudas y se elaboró un plan de formación con el cronograma general de las tareas a realizar. El investigador, con la ayuda de los estudiantes, elaboró el guion de contenido (Anexo 16), a partir de las subdimensiones definidas para el diseño didáctico del software.

Para la subdimensión didáctica se analizó el programa analítico y el plan calendario de la asignatura IA-II. Se determinaron los objetivos y contenidos a trabajar con el SLE, se especificaron métodos específicos para incorporar el software en cada forma organizativa del PEA. Se especificaron indicaciones para la utilización de los medios disponibles en el aula y en los laboratorios, la forma de trabajar en la plataforma de enseñanza-aprendizaje y cómo aprovechar los medios de los propios estudiantes. Por último se especificaron las vías para la evaluación sistemática, parcial y final a los estudiantes, tomando en cuenta el trabajo con el software.

Como parte de la subdimensión espacial se elaboraron tareas generales para trabajar con el software en el grupo, por equipos y de forma individual por los estudiantes. De igual forma, según la subdimensión práctica, se elaboraron tareas generales para estudiar el código fuente, para modificarlo y reutilizarlo.

Para garantizar la utilización y distribución del software como expresa la subdimensión de gestión se especificaron los requerimientos tecnológicos del software para ser utilizado según las condiciones de aulas y laboratorios. Se llevó a cabo la adecuación del curso en la plataforma de enseñanza-aprendizaje Zera (Manso, Cañizares, & Febles, 2016), para incluir el software, los documentos de ayuda y otras orientaciones necesarias.

Las acciones anteriores tienen su concreción en la subdimensión tecnológica. Aquí se definen los elementos computacionales para el software, de manera que tome en cuenta las características ya diseñadas, en su desarrollo computacional (Gutiérrez, Cedeño, & Coca, 2017). Para llevar a cabo el sistema se siguió la metodología Ingenias (Botía, González, Gómez, & Pavón, 2008), en la cual se definen metamodelos que guían y ayudan en la estructuración arquitectónica del sistema multiagente. Durante la fase de análisis, los metamodelos desarrollados fueron el de agentes, el de organización, el de objetivos y tareas y el de entorno, posteriormente en la fase de diseño se refinaron dichos metamodelos y se agregó la especificación de las tareas mediante el metamodelo de interacción.

El sistema quedó conformado por los agentes Smek, Connector, Arff, Classifier y Clusterer. Smek es el agente principal, encargado de la administración del SMA y de la interacción con el usuario. El agente Connector se responsabiliza de la primera etapa del proceso de transformación de bases de datos de PostgreSQL (PostgreSQL, 2020) en archivos con extensión arff con que trabaja el Weka. El agente Arff inicia la segunda etapa de dicho proceso con la información suministrada por Connector, y realiza tareas de edición sobre archivos arff. Por su parte, los

agentes Classifier y Clusterer ejecutan las operaciones de clasificación y agrupamiento respectivamente.

La especificación de los requisitos no funcionales, funcionales y demás artefactos de la metodología se recogen en el documento del trabajo de diploma disponible en el repositorio institucional de la UCI (Gutiérrez et al., 2017). Los elementos de mayor interés en el desarrollo fueron la utilización de la biblioteca libre JADE para el desarrollo de los agentes y la utilización de Weka como biblioteca de algoritmos de minería de datos. La inclusión de estas bibliotecas exigió especificar la forma de vincularlas al entorno de desarrollo NetBeans (NetBeans, 2020), utilizado para crear el proyecto computacional. La utilización de bases de datos en el gestor libre PostgreSQL requirió hacer una transformación de las tablas en formato SQL, al formato arff.

El cronograma de desarrollo fue cumplido sin dificultades, se logra un software con las características deseadas (Anexo 17). Se presentaron resultados en los eventos científicos estudiantiles: Copa de ingeniería de software y la Jornada Científica del Ingeniero en Ciencias Informáticas. Se obtuvo premio en el evento de ingeniería de software. La defensa del trabajo de diploma fue satisfactoria, en tiempo y con evaluación de excelente.

Luego del diseño didáctico del software se llevó a cabo la planificación de las tareas a desarrollar en el PEA. Se elaboraron los planes de clases siguiendo las indicaciones del diseño didáctico del software. El grupo asignado al profesor fue el 4505, integrado por 18 estudiantes, tres hembras y 15 varones.

En la primera conferencia de la asignatura se aplicó el diagnóstico (Anexo 18) a los estudiantes. La elaboración del mismo se realizó para complementar la información que dispone el profesor en la entrega pedagógica al inicio del curso y los resultados del diagnóstico integral realizado a los estudiantes al ingresar a la carrera.

Ya los estudiantes han recibido una asignatura de IA, por lo que tienen conocimientos generales de la misma. No obstante, se realizó un recordatorio general de los núcleos de conocimientos y problemas generales que resuelve. Con este punto de partida, se orientó la búsqueda de un problema profesional a ser resuelto con tareas de minería de datos. Este problema podía estar relacionado con el centro de desarrollo donde realizan su PP u otra inquietud o interés personal. Junto al problema debían entregar una base de datos en PostgreSQL a la cual se le aplicarían los algoritmos de minería de datos.

En la primera frecuencia de la semana seis, se recogieron los problemas redactados por los estudiantes. Dos estudiantes no entregaron problemas, alegando que no encontraron, ni supieron cómo redactar uno de sus propios intereses. A todos los demás problemas entregados, se les hicieron adecuaciones para ajustarlos a los objetivos y contenidos de la asignatura. Para los dos estudiantes que no entregaron problemas se elaboraron tareas, a partir de la información recogida en el diagnóstico. En todos los casos se tuvo en cuenta las indicaciones del guion de contenido para la asignación de tareas de reutilizar el código fuente.

La primera conferencia del tema aprendizaje automático se impartió en la semana ocho. En ella se presentó el SLE, su interfaz y las funcionalidades generales. Como trabajo independiente se orientaron las tareas personalizadas a cada estudiante.

En la siguiente clase práctica se realizaron tareas por equipo para el estudio del código fuente, según las indicaciones del guion de contenido. Estas tareas fueron presentadas y debatidas en el grupo. En la clase de laboratorio posterior, se asignaron las primeras tareas de modificación del código fuente, igualmente por equipos y tomando en cuenta las indicaciones del guion de contenido.

En las semanas 11 y 12, en las clases de laboratorio correspondientes, se realizaron tareas relacionadas con el contenido específico y se aclararon dudas a los estudiantes sobre el trabajo independiente asignado al inicio para reutilizar el código fuente. Siempre que fue posible se realizaron las preguntas y se aclararon las dudas de forma grupal, donde los propios compañeros de aula podían exponer sus experiencias y las formas de dar solución a los problemas enfrentados.

En la clase de laboratorio de la semana 14 se efectuó la presentación de los resultados por los estudiantes. Como elementos más significativos, tres estudiantes reutilizaron los agentes en JADE del SMA, cuando el resto solo reutilizó los métodos de acceso a la biblioteca Weka. Todos reutilizaron el agente de transformación de PostgreSQL a formato arff.

Para la nota final por el trabajo desarrollado se consideró que cuatro estudiantes vencieron los objetivos, por lo que fueron eximidos de la prueba final. A otros tres estudiantes se les subió la nota alcanzada en la prueba final, tomando como criterio fundamental su trabajo con el SLE.

Todas las soluciones de los estudiantes se valoraron para posibles modificaciones al software. Fueron incorporadas finalmente dos modificaciones al software, un nuevo algoritmo que incorporó la versión actualizada del Weka y un cambio en la interfaz

que permitió un acceso más cómodo a la funcionalidad de comparación de algoritmos.

3. Reflexión crítica:

Para el análisis crítico del proceso, se analizaron cada uno de los indicadores de la variable de investigación. La dimensión diseño didáctico se analiza a través del guion de contenido.

Subdimensión didáctica:

En el guion de contenido se recogen indicaciones precisas para abordar varios objetivos y contenidos de la asignatura. Se especifican diversos métodos a utilizar según la tipología de clases y se relacionan un conjunto de medios a utilizar de manera que soporten el trabajo con el SLE. Adicionalmente se realizan indicaciones para evaluar a los estudiantes de forma sistemática, parcial y para definir su evaluación final a partir de su trabajo con el software. Precisamente la evaluación dejó algunas insatisfacciones, ya que al estar definida una evaluación final en la asignatura, el trabajo con el software no siempre influyó en la nota final del estudiante.

Subdimensión espacial:

Se plasman en el guion de contenido tres tareas generales para el trabajo en grupo, tres para el trabajo en equipo y tres para el trabajo individual.

Subdimensión práctica:

Se muestran en el guion de contenido cuatro tareas para el estudio del código fuente, cuatro para la modificación del código fuente y dos para la reutilización del código fuente.

Subdimensión de gestión:

Está amparado en una licencia libre, es multiplataforma y se encuentra disponible en la plataforma de enseñanza-aprendizaje Zera. Se definieron las características del software para su correcto funcionamiento en los laboratorios y aulas. Sin embargo, en las aulas no siempre se logró que ejecutara correctamente.

Subdimensión tecnológica:

El código fuente es legible, se utiliza el paradigma de programación para agentes con una arquitectura organizada con facilidades para modificar y adicionar nuevas funcionalidades. Se cuenta con el guion de contenido y ayudas en el mismo software, todos disponibles en la plataforma de enseñanza-aprendizaje Zera.

De forma general se aprecia un diseño didáctico del software libre educativo que contribuye al cumplimiento de los indicadores definidos en la investigación. En cuanto a la dimensión incorporación armoniosa, el análisis por subdimensiones arrojó lo siguiente:

Subdimensión contextualizada:

Se utilizó un diagnóstico para conocer el trabajo en la PP de los estudiantes y otros intereses profesionales y personales. Se asignan las tareas de búsqueda de problemas a los propios estudiantes, lo cual contribuye a su implicación en el proceso. Todas las tareas personalizadas asignadas a los estudiantes, toman en cuenta o su trabajo en la PP o sus intereses.

Subdimensión sistémica:

El software fue diseñado para utilizarse en uno de los dos temas generales de la asignatura, pero su alcance permite asignar tareas de investigación a los

estudiantes, antes de comenzar el tema para el cual fue diseñado. Se considera que la no utilización del software afectaría el cumplimiento de los objetivos relacionados con el tema de minería de datos.

La reutilización dentro de la misma asignatura no fue posible con el software porque fue dirigido su diseño a uno de los dos temas. Sí se hizo efectiva la reutilización de resultados para los siguientes cursos, en tanto se analizaron todas las soluciones de los estudiantes y se seleccionaron dos de ellas para mejorar el software.

Subdimensión reflexiva:

El trabajo con el software tributa al tema de investigación del profesor y del RGA que formó parte del colectivo de la asignatura. Se reconoce que a los estudiantes les aportará no solo al vencimiento de los objetivos y al desarrollo de las habilidades específicas de la asignatura, también contribuye al desarrollo de habilidades generales para su futura vida profesional. Los propios estudiantes reconocen la utilidad del software, incluso para estudiar y reutilizar el código en su futura vida profesional.

Subdimensión planificada:

El profesor dedicó más de 20 horas totales para el diseño didáctico del SLE y para su autosuperación. Se planificaron y cumplieron las tareas con el software en un 31% de clases y se asignan y controlan dos trabajos independientes que implican más de seis horas para el estudiante.

Para la retroalimentación del proceso se aplicó la técnica de satisfacción de ladov a los estudiantes y se desarrolló la sistematización de experiencias.

A partir de este análisis, se asignó un valor a cada indicador, como fue definido en el proceso de operacionalización de la variable objeto de estudio (Anexo 2). Los resultados se muestran en el Anexo 19. Un análisis comparativo de las subdimensiones al inicio de la investigación con los resultados obtenidos en la sistematización se muestra en la Figura 7.

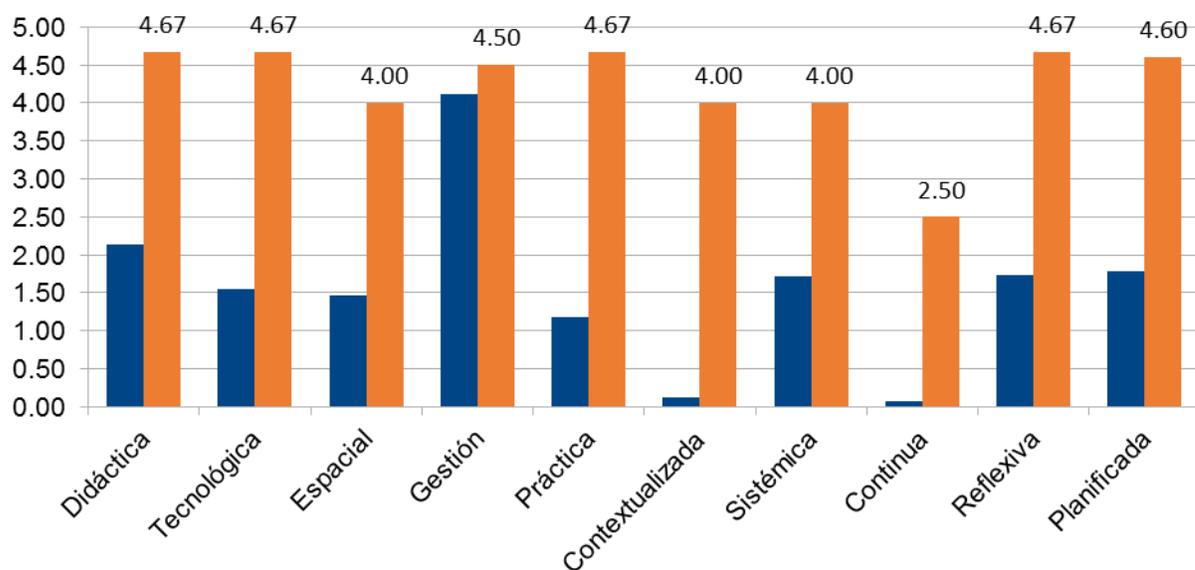


Figura 7. Valores de cada subdimensión en el diagnóstico y la sistematización de experiencias

4. Cierre:

Concluido el análisis crítico de la sistematización de experiencias se puede afirmar lo siguiente:

1. Todas las subdimensiones mejoran sus indicadores con respecto a los valores obtenidos en el análisis empírico de la investigación. La subdimensión continua es la única que obtiene nivel Medio.

2. El trabajo de los estudiantes con el software no siempre influyó en la nota final, dadas las características de la asignatura. Se sugiere modificar el programa analítico de las asignaturas de manera que tomen en cuenta la integración del software libre educativo.
3. La reutilización de resultados dentro de la misma asignatura no fue posible, ya que el diseño didáctico del software tomó en cuenta solo uno de los dos temas generales de la asignatura. Se sugiere realizar el diseño didáctico del software tomando en cuenta toda la asignatura.
4. Se aprecia que el software, a diferencia de los analizados previamente, brinda facilidades para su incorporación armoniosa al PEA de la IA. Se cuenta con un diseño didáctico que incluye tareas generales a realizar por los estudiantes en el PEA aprovechando las libertades de estudio, modificación y reutilización del código fuente. En la asignación de tareas a los estudiantes se toman en cuenta sus intereses y su trabajo en la PP, logrando una mayor contextualización del trabajo con el software libre educativo.
5. Se muestra a través de la experiencia sistematizada, la posibilidad de aplicar la integración de SLE al PEA de la IA desde el diseño didáctico del software por el propio profesor, hasta su incorporación armoniosa al PEA para el cual fue diseñado.

Los principales resultados de la sistematización fueron presentados en dos talleres metodológicos. El primero al colectivo de profesores de la disciplina IA en la carrera Ingeniería en Ciencias Informáticas. El segundo en el Departamento de Técnicas de Programación de la Facultad 4, integrado por profesores de las disciplinas

Programación, Sistemas Digitales e Inteligencia Artificial. Estos colectivos de profesores fueron utilizados como grupos focales para obtener criterios sobre la pertinencia, consistencia teórica, aceptación de la propuesta y aplicabilidad del modelo.

Se diseñó un curso de postgrado con las principales experiencias, tanto teóricas como metodológicas. Este curso se impartió como parte de la escuela de verano 2018 de la UCI (Anexo 20).

2.2.3 Técnica de satisfacción de ladov

La técnica de satisfacción de ladov, surge para establecer la satisfacción por la profesión en carreras pedagógicas (Kusmina, 1970). Se ha generalizado para evaluar la satisfacción de personas en diversos ámbitos, incluido el educativo. A partir de esta técnica se determina la satisfacción individual y grupal.

Para la presente investigación se consideró importante obtener criterios de satisfacción de los estudiantes que participaron de la experiencia de incorporar el SLE al PEA de la asignatura Inteligencia Artificial II. Para ello se elaboró un cuestionario (Anexo 21. a) con tres preguntas cerradas e intercaladas. Se toman como preguntas para el cuadro lógico la 10, la ocho y la tres (Anexo 21. b). La respuesta a estas tres preguntas permitió ubicar al estudiante, según el cuadro lógico, en una escala de satisfacción:

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

5- Clara insatisfacción

6- Contradictoria

De los 18 estudiantes 10 tienen máxima satisfacción, cinco expresan estar más satisfechos que insatisfechos, mientras tres tienen respuestas no definidas o contradictorias. A partir de estas cantidades se calcula el Índice de Satisfacción Grupal (ISG) según la fórmula:

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

A (Máxima satisfacción): Cantidad de estudiantes con índice de satisfacción uno.

B (Más satisfecho que insatisfecho): Cantidad de estudiantes con índice de satisfacción dos.

C (No definido o contradictorio): Cantidad de estudiantes con índice de satisfacción tres o seis.

D (Más insatisfecho que satisfecho): Cantidad de estudiantes con índice de satisfacción cuatro.

E (Máxima insatisfacción): Cantidad de estudiantes con índice de satisfacción cinco.

El valor obtenido del ISG fue de 0.69, lo cual confirma buena satisfacción grupal.

Las preguntas abiertas del cuestionario fueron tomadas en cuenta para enriquecer el modelo. El señalamiento más importante realizado fue el poco tiempo que alegan tener para resolver la tarea de reutilización del código. Este aspecto puede ser atenuado con un rediseño de la asignatura, de manera que tome en cuenta la

integración del SLE para brindar más tiempo para el seguimiento y orientación de las tareas con el software libre educativo.

2.2.4 Valoración por grupos focales

La técnica de grupos focales se reconoce como la discusión espontánea sobre un tema seleccionado con anterioridad, guiados por un moderador o facilitador (Martínez, 2012), (Ciudad, 2016). Debe desarrollarse en un grupo pequeño, de entre seis y 15 miembros y debe tener un carácter de taller.

Para la presente investigación se consideraron los siguientes aspectos para conformar los grupos focales: número total de grupos, cantidad de participantes por grupos y nivel de relación técnico-pedagógica con el PEA de la Inteligencia Artificial.

Se conformaron dos grupos focales con un total de 19 personas. Sus principales características se muestran en el Anexo 22. a):

G1. Metodológico: Departamento de Técnicas de Programación Facultad 4.

G2. Metodológico específico: Colectivo universitario de la disciplina IA.

Para un mejor desarrollo de los talleres se utilizó una guía de temas, que permitiera conocer los criterios en relación a la variable estudiada (Anexo 22. b). Se desarrolló un taller con cada uno de los grupos focales y se utilizaron un conjunto de criterios operacionales (Anexo 22. c).

Los resultados cualitativos de los talleres efectuados (Anexo 22. d) fueron sometidos a los criterios operacionales para obtener los elementos tanto positivos como negativos con unanimidad, mayoría o minoría de criterios (Anexo 22. e).

Posteriormente se llevó a cabo un proceso de perfeccionamiento del modelo tomando en cuenta los criterios operacionales. Se trabajó en todos los planteamientos negativos o sugerencias con unanimidad y mayoría de criterios y se incluyó el análisis de los que tuvieron minoría de criterios. Las principales propuestas que se tomaron en cuenta fueron en el componente instrumental. Específicamente en especificar mejor las acciones para la asignación y seguimiento de las tareas de reutilización del código fuente a los estudiantes.

2.2.4 Triangulación metodológica

Se procedió a una triangulación de los resultados obtenidos en cada una de las técnicas y métodos empleados. La triangulación metodológica es descrita como la confrontación de los resultados obtenidos por diferentes métodos en la búsqueda de las semejanzas y diferencias en los criterios valorativos y de esta forma incrementar la credibilidad de la investigación realizada (Hernández, Fernández, & Baptista, 2014).

Se asumieron como criterios valorativos la pertinencia del modelo, su consistencia teórica, su aplicabilidad y la aceptación de la propuesta por los principales implicados en el proceso, estudiantes y profesores. En la Tabla 6 se muestra la relación de criterios valorativos por método aplicado.

Se constató a partir de los criterios expresados por los expertos, la pertinencia y la consistencia teórica del modelo, sus componentes y relaciones, así como la posibilidad de implementarlo. La técnica de grupos focales aportó elementos tanto teóricos como metodológicos y de aseguramiento para llevar a la práctica el modelo.

Se ofrecieron criterios sobre la pertinencia y consistencia teórica del modelo.

Tabla 6. Criterios valorativos por cada método o técnica aplicada

Criterio valorativo/método o técnica	Criterio de expertos	Sistematización de experiencias	Técnica de satisfacción de ladov	Grupos focales
Pertinencia	X	X		X
Consistencia teórica del modelo	X	X		X
Aplicabilidad	X	X		X
Aceptación de la propuesta		X	X	X

La sistematización de experiencias mostró la aplicabilidad de la propuesta y evidenció la mejora de los indicadores definidos para la integración de software libre educativo al PEA de la IA. Las técnicas de grupos focales y de satisfacción de ladov mostraron la aceptación de la propuesta tanto por los profesores, como por los estudiantes, respectivamente.

Conclusiones del capítulo

El modelo elaborado contiene en su esencia características de los modelos para el diseño didáctico del software educativo y de los modelos para la integración de tecnologías al proceso de enseñanza-aprendizaje. Lo diferencia sus tres componentes generales, un componente conceptual, al cual se subordina el componente estructural y a este el componente instrumental. Siendo fundamentales las relaciones esenciales que se establecen entre el diseño didáctico del software libre educativo y su incorporación armoniosa al proceso de enseñanza-aprendizaje de la Inteligencia Artificial.

La triangulación de los resultados obtenidos por el método Delphi, la sistematización de experiencias, la técnica de satisfacción de ladov y de grupos focales permitieron obtener criterios valorativos positivos sobre la pertinencia, la consistencia teórica del modelo, su aplicabilidad y la aceptación de la propuesta tanto por profesores como por estudiantes.

CONCLUSIONES FINALES

La valoración de las tendencias teóricas, históricas y metodológicas de la integración del software educativo al proceso de enseñanza aprendizaje de la Inteligencia Artificial corrobora la actualidad y pertinencia del problema abordado. Se demuestra la necesidad del aprovechamiento de las libertades del software libre en el diseño didáctico de un software educativo y su incorporación armoniosa al proceso de enseñanza aprendizaje de la Inteligencia Artificial.

El diagnóstico realizado demostró la existencia de deficiencias en la integración de software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial en la carrera Ingeniería en Ciencias Informáticas. En el diseño didáctico, se constatan limitaciones en la arquitectura del software, ausencia de un diseño didáctico del software, escaso aprovechamiento de las libertades del software libre en el proceso de enseñanza-aprendizaje, deficiente contextualización de las tareas a realizar por los estudiantes con el software en el proceso de enseñanza-aprendizaje y baja percepción del aporte del software utilizado a la autosuperación de los profesores y su aporte al futuro trabajo profesional de los estudiantes. Se constató la necesidad de abordar los componentes y las relaciones esenciales de la integración con vistas a la transformación del proceso de enseñanza-aprendizaje, teniendo al software libre educativo como medio.

El modelo presentado se sustenta en tres principios y en fundamentos teóricos y metodológicos. Aborda como elementos esenciales el diseño didáctico del software libre educativo y su incorporación armoniosa al proceso de enseñanza-aprendizaje de la Inteligencia Artificial. La aplicación del modelo se realiza a través de una

estrategia didáctica con tres etapas: preparación, ejecución y evaluación. Se expresan las relaciones esenciales que se establecen entre sus componentes.

La valoración de la contribución del modelo se realizó a partir de los resultados de la aplicación de la técnica de grupos focales, el criterio de expertos, la técnica de satisfacción de ladov y la sistematización de experiencias. A partir de la triangulación metodológica se constató la consistencia teórica del modelo, su aplicabilidad y la aceptación de la propuesta por profesores y estudiantes.

RECOMENDACIONES

Tomar en cuenta, desde el diseño curricular de las asignaturas de Inteligencia Artificial del Plan E del Ingeniero en Ciencias Informáticas, la integración de software libre educativo, de manera que se le otorgue mayor tiempo al trabajo independiente del estudiante y mayor peso en la evaluación al trabajo con el software libre educativo.

Profundizar en las relaciones que se establecen entre el software libre educativo y el estudiante en los diversos espacios de interacción, ya sea personal, en equipo o el grupo, utilizando entornos virtuales de enseñanza-aprendizaje y los medios personales de los propios estudiantes.

REFERENCIAS BIBLIOGRÁFICAS

1. Abu, S. (2008). Developing visualization tool for teaching AI searching algorithms. *Information Technology Journal*, 7(2), 350–355.
2. ACM-IEEE-CS. (1991). COMPUTING CURRICULA 1991. *Communications of the ACM*, 34(6), 69–84.
3. ACM-IEEE. (2001). *Computing Curricula 2001. Computer Science*.
4. ACM-IEEE. (2013). CS2013: Computer Science Curricula 2013. In *Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. <https://doi.org/10.1145/2534860>
5. ACM. (2020). Association for Computing Machinery. Advancing Computing as a Science & Profession. Sitio oficial: <https://www.acm.org/>
6. Addine, F. (2002). !Didáctica! ¿Qué Didáctica? En *Didáctica: Teoría y práctica* (pp. 8–18). La Habana.
7. Aguilar, A., Luzardo, M., & Jaimes, L. (2015). Adopción de TIC en la Universidad Pontificia Bolivariana Bucaramanga, Colombia. *Educere*, 19(62), 157–168.
8. Aguilar, I., De la Vega, J. A., Lugo, O., & Zarco, A. (2014). Análisis de criterios de evaluación para la calidad de los materiales didácticos digitales. *Revista Iberoamericana de Ciencia, Tecnología y Sociedad - CTS*, 9(25), 73–89.
9. Alexander, B., Ashford-Rowe, K., Barajas-Murphy, N., Dobbin, G., Knott, J., McCormack, M., ... Weber, N. (2019). *EDUCAUSE Horizon Report: 2019 Higher Education Edition*. Disponible en: <https://library.educause.edu/-/media/files/library/2019/4/2019horizonreport.pdf?la=en&hash=C8E8D444AF372E705FA1BF9D4FF0DD4CC6F0FDD1>.
10. Alvarez-Dionisi, L.E., Mittra, M., & Balza, R. (2019). Teaching Artificial Intelligence and Robotics to Undergraduate Systems Engineering Students. *I.J.Modern Education and Computer Science*, (7), 54-63.
11. Álvarez, A. (2014). *Estrategia pedagógico-tecnológica para la integración de las tecnologías de la información y las comunicaciones en el proceso de enseñanza-aprendizaje desde la producción de materiales educativos digitales en el Instituto Superior Politécnico José Antonio*. Tesis en opción al grado científico de Doctor en Ciencias de la Educación. Instituto Superior Politécnico José Antonio Echeverría.
12. Álvarez, A., Hernández, L., & Cabrera, J. F. (2013). Estudio de las dimensiones de la integración de las TIC en una universidad tecnológica cubana. *Revista Cubana de Ingeniería*, IV(3), 5–14.

13. Area-Moreira, M., Hernández-Rivero, V., & Sosa-Alonso, J. J. (2016). Modelos de integración didáctica de las TIC en el aula. *Comunicar*, XXIV(47), 79–87. <https://doi.org/http://dx.doi.org/10.3916/C47-2016-08>
14. Área, M. (2009). *Introducción a la tecnología educativa: Manual electrónico*. Universidad de La Laguna (España).
15. Área, M. (2010). *Los medios de enseñanza: Conceptualización y tipología*. Universidad La Laguna. Disponible en: http://www.uclm.es/PROFESORADO/RICARDO/Clasificaciones_medios/doc_ConcepMed.html
16. Asensio, J., Mora, A., Fernández, A., García-Sánchez, P., Merelo, J., & Castillo, P. (2014). Progamer: aprendiendo a programar usando videojuegos como metáfora para visualización de código. *ReVisión*, 7(2), 93–101.
17. Atchison, W. F., Schweppe, E. J., Viavant, W., Young, D. M., Conte, S. D., & Hamblen, J. W. Rheinboldt, W. C. (1968). Curriculum 68: Recommendations for academic programs in computer science: a report of the ACM curriculum committee on computer science. *Communications of the ACM*, 11(3), 151–197. <https://doi.org/doi:10.1145/362929.362976>
18. Austing, R. H., Barnes, B. H., Bonnette, D. T., Engel, G. L., & Stokes, G. (1979). Curriculum '78: recommendations for the undergraduate program in computer science. *Communications of the ACM*, 22(3), 147–166.
19. Aznar, F., Compañ, P., Pujol, M., Rizo, R., Rizo, C., Sempere, M., ... Viejo, D. (2015). Consideraciones iniciales del cambio de metodología presencial a MOOC. El caso de Sistemas Inteligentes. *Jornadas de Redes de Investigación En Docencia Universitaria*, 1465–1475. Universidad de Alicante.
20. Bahamonde, A., & Otero, R. (2001). ESPECIAL CAEPIA-TTIA ' 2001. IX Conferencia de La Asociación Española Para La Inteligencia Artificial. Gijón, España.
21. Bellas, F., & Alonso, A. (2007). Metodología de trabajo y experiencias de aprendizaje colaborativo y evaluación continua en la disciplina de Sistemas Multiagente. *XIII Jornadas de Enseñanza Universitaria de La Informática*, 65–73.
22. Bello, R. (2019). Trends of Artificial Intelligence: challenge in Education. *International Conference on Interactive Collaborative and Blended Learning*, 51. Santiago de Cuba.
23. Bello, R. E., García, Z. Z., García, M. M., & Reynoso, A. (2002). *Aplicaciones de la Inteligencia Artificial* (Primera ed). Guadalajara.

24. Beloso, N., & Perozo, M. (2009). Asimilación de tecnología de información y comunicación en las alcaldías de Venezuela. *Revista de Ciencias Sociales*, XV(1), 139–147.
25. Berrutti, L., Cabo, M., & Dabezies, M. J. (n.d.). *Sistematización de experiencias de extensión*. Montevideo. Uruguay: Comisión Sectorial de Extensión y Actividades en el Medio.
26. Blanco, A. (1997). *Introducción a la sociología de la educación*. Ciudad de La Habana: Instituto Superior Pedagógico “Enrique José Varona.”
27. Botía, J. A., González, J. C., Gómez, J., & Pavón, J. (2008). IWPAAMS2007-07: The Ingenias Project: Methods And Tool For Developing Multiagent Systems. *IEEE LATIN AMERICA TRANSACTIONS*, 6(6), 529–534. <https://doi.org/10.1109/TLA.2008.4908186>
28. Cabero, J., Morales, J. A., Barroso, J., Roman, P., & Romero, R. (2004). La red como instrumento de formación. Bases para el diseño de materiales didácticos. *Revista de Medios y Educación*, (22), 5–23.
29. Cabrera, J. F. (2008). *Modelo de Centro Virtual de Recursos para contribuir a la integración de las TIC en el Proceso de Enseñanza Aprendizaje en el Instituto Superior Politécnico José Antonio Echeverría*. Tesis en opción al grado científico de Doctor en Ciencias de la Educación. CUJAE.
30. Castellanos, D., Castellanos, B., Llivina, M., Silverio, M., Reinoso, C., & García, C. (2002). *Aprender y Enseñar en la Escuela: Una Concepción Desarrolladora*. La Habana: Editorial Pueblo y Educación.
31. Cataldi, Z, Lage, F., Pessacq, R., & García, R. (1997). Ingeniería del software educativo. *Informática Industrial*, 16. <https://doi.org/10.1016/j.coal.2012.01.012>
32. Cataldi, Zulma. (2000). *Metodología de diseño, desarrollo y evaluación de software educativo*. Facultad de Informática. UNLP.
33. Chávez, J. (2007). A propósito de la Didáctica. *Actividad Desarrollada En La Asociación de Pedagogos de Cuba*, 29. La Habana.
34. Chávez, J., Fundora, R., & Pérez, L. (2010). *Filosofía de la educación*. Ciudad de La Habana: ICCP-MINED.
35. Chávez, J., Suárez, A., & Permuy, L. (2003). *Un acercamiento necesario a la pedagogía general*. La Habana: ICCP.
36. Ciudad, F. A. (2012). *Diseño didáctico de un entorno virtual para la integración academia-industria en la disciplina ingeniería y gestión de software en la Universidad de las Ciencias Informáticas*. Tesis en opción al grado científico de Doctor en Ciencias de la Educación. Universidad de las Ciencias Informáticas.

37. Ciudad, F. A. (2016). *Diseño de Entornos Virtuales para la integración academia – industria. Implementación en la Disciplina Ingeniería y Gestión de Software*. La Habana: Publicia.
38. Cliffe, O., De Vos, M., Padget, J., & Casasola, E. (2005). Teaching Multi-Agent Systems in the UK and in Latin America. *Innovation in Teaching and Learning in Information and Computer Sciences*, 4(3), 1–29. <https://doi.org/10.11120/ital.2005.04030004>
39. Coca, Y. (2011). Utilización de herramientas de visualización para el aprendizaje de la Inteligencia Artificial. En: *XIV Congreso de Informática en la Educación. Informática 2011*. La Habana.
40. Coca, Y. (2015). Propuesta didáctica para la enseñanza de la inteligencia artificial con herramientas abiertas de visualización interactiva. En: *Compumat 2015*. La Habana.
41. Coca, Y. (2016). Herramientas abiertas de visualización interactiva como medio de enseñanza para las ciencias de la computación. En: *XVI Congreso Internacional de Informática En La Educación. Informática 2016*. La Habana.
42. Coca, Y., & Pérez, M. T. (2018). Diseño didáctico de software para la enseñanza-aprendizaje de la Inteligencia Artificial. En: *I Taller Internacional Tecnología y Educación*. La Habana.
43. Coca, Y., Rosell, L. B., & Velazquez, A. (2017). Modelo de agente lógico con inferencia basada en hechos. *Revista Cubana de Ciencias Informáticas*, 11(2), 29–45.
44. Colectivo. (2002). *Didáctica: teoría y práctica* (F. Addine & S. Recarey, Eds.). La Habana.
45. Colectivo. (2016). *El pensamiento de Lev Semiónovich Vigotsky. Su vigencia en la educación*. Santo Domingo. República Dominicana: Biblioteca del pensamiento crítico.
46. Colectivo Carrera ICI. (2015). *Programa de la disciplina Inteligencia Artificial. Plan D*. Universidad de las Ciencias Informáticas. La Habana.
47. Colectivo Carrera ICI. (2019). *Defensa territorial del Plan E*. Universidad de las Ciencias Informáticas. La Habana.
48. Colectivo Disciplina IA. (2015). *Plan de Trabajo Metodológico. Curso 2015-2016*. Universidad de las Ciencias Informáticas. La Habana.
49. Coloma, O. (2008). *Concepción didáctica para la utilización del software educativo en el proceso de enseñanza aprendizaje*. Tesis en opción al grado científico de Doctor en Ciencias Pedagógicas. Instituto Superior Pedagógico “José de la Luz y Caballero.”

50. Cruz, R. (2015). Un sistema experto para el desarrollo del proceso docente educativo en la disciplina Inteligencia Artificial. *Revista Ciencias Pedagógicas*, (2), 1–10.
51. Del Toro, M. (2006). *Modelo de diseño didáctico de hiperentornos de enseñanza - aprendizaje desde una concepción desarrolladora*. Tesis en opción al grado científico de Doctor en Ciencias Pedagógicas. Instituto Superior Pedagógico “Enrique José Varona.”
52. DeNero, J., & Klein, D. (2010). Teaching Introductory Artificial Intelligence with Pac-Man. En: *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 1885–1889.
53. Denzinger, J., & Kidney, J. (2005). Teaching Multi-Agent Systems using the ARES Simulator. *Innovation in Teaching and Learning in Information and Computer Sciences*, 4(3), 1–23.
54. Dillon, R. (1993). Introducing artificial intelligence into a high school’s computer curriculum. *Technological Horizons In Education*, 20(8), 9–11.
55. EDUCAUSE. (2018). *NMC Horizon Report Preview 2018 Higher Education Edition*.
56. El-Nasr, M., & Smith, B. (2006). Learning through Game Modding. *ACM Computers in Entertainment*, 4(1).
57. Fasli, M. (2005). Editorial. *Innovation in Teaching and Learning in Information and Computer Sciences*, 4(3), 1–2. <https://doi.org/10.11120/ital.2005.04030001>
58. Fernández, D. (2012). *Una metodología para la evaluación del software educativo*. Tesis en opción al grado científico de Doctor. Universidad de Cienfuegos.
59. Fluck, A. (2003). *Integration or transformation: a cross-national study of information and communication technology in school education*. University of Tasmania.
60. Font, O., & Paez, V. (2017). *Los métodos del proceso de enseñanza-aprendizaje desde la didáctica general*. La Habana: Universidad de Ciencias Pedagógicas “Enrique José Varona.”
61. Free Software Foundation. (2019). gnu.org. Recuperado el 1ro de octubre de 2019 de <https://www.gnu.org/education/edu-software.en.html>
62. Frigg, R., & Hartmann, S. (2018). Models in Science. En E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Summer 201). Disponible en: <https://plato.stanford.edu/archives/sum2018/entries/models-science/>
63. Frolov, I. (1984). *Diccionario de Filosofía*. <https://doi.org/10.5840/schoolman195229367>

64. FSF. (2019). El sistema operativo GNU. Licencias website. Recuperado de: <https://www.gnu.org/licenses/licenses.es.html>
65. Galvis, Á. (1992). *Ingeniería de Software Educativo*. Santafé de Bogotá: Ediciones Uniandes.
66. Galvis, A. H. (1988). Ambientes de enseñanza-aprendizaje enriquecidos con computador. *Boletín de Informática Educativa*, 1(2), 117–139.
67. García, S., Rico, P (2015). *Concepción desarrolladora para el diseño didáctico del software educativo de la escuela primaria*. La Habana: Editorial Universitaria
68. García, Y., & Lezcano, M. G. (2011). Teachshell versión 2.0 herramienta de apoyo a la enseñanza de sistemas expertos. En: *Compumat*.
69. Gil, J. E. (2010). *Estrategia de gestión de recursos educativos abiertos en forma de objetos de aprendizaje en la Universidad de La Habana*. Tesis en opción al grado científico de Doctor en Ciencias de la Educación. Universidad de La Habana.
70. Gimeno, J. (1981). *Teoría de la enseñanza y desarrollo del currículo*. Madrid: Anaya.
71. Gómez, R., Gálvis, Á., & Mariño, O. (1998). Ingeniería de software educativo con modelaje orientado a objetos. Un medio para desarrollar micromundos interactivos. *Informática Educativa*, 11(1), 9–30.
72. González-Castro, V. (1986). *Teoría y práctica de los medios de enseñanza*. La Habana: Pueblo y Educación.
73. González, A. M., Recarey, S., & Addine, F. (2002). El proceso de enseñanza-aprendizaje : Un reto para el cambio educativo. En *Didáctica: Teoría y práctica* (pp. 47–69). La Habana.
74. Grivokostopoulou, F.; Perikos, I. and Hatzilygeroudis, I. (2017). A Collaborative Game for Learning Algorithms. En *Proceedings of the 9th International Conference on Computer Supported Education - Volume 1: CSEDU*, 543-549. DOI: 10.5220/0006377405430549.
75. Gros, B. (1997). *Diseños y programas educativos: Pautas pedagógicas para la elaboración de software*. Ariel.
76. Gutiérrez, J. A., Cedeño, Y., & Coca, Y. (2017). *Sistema multiagente para la extracción de conocimiento en bases de datos*. Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas. Universidad de las Ciencias Informáticas.
77. Hatzilygeroudis, I., Grivokostopoulou, F., & Perikos, I. (2012). Teaching Aspects of Constraint Satisfaction Algorithms Via a Game. En: *Third AAAI Symposium on Educational Advances in Artificial Intelligence*, 2371–2372.

78. Hernández, R., Fernández, C., & Baptista, P. (2014). *Metodología de la investigación* (Sexta edic). México: McGraw-Hill.
79. Ibarra, M. A., Torres, L. A., & Cervantes, K. E. (2013). La adopción de las TIC en las PyMEs del sector manufacturero de Baja California. En: *XVIII Congreso Internacional de Contaduría, Administración e Informática*. Recuperado de : <http://premio.investiga.fca.unam.mx/docs/ponencias/2013/10.1.pdf>
80. Iriarte, L. (2007). *Modelo para la gestión de información en la producción de contenidos didácticos en la nueva universidad cubana*. Tesis en opción al grado científico de Doctor en Ciencias de la Educación. Universidad Agraria de La Habana.
81. Jacobson, I., Booch, G., & Rumbaugh, J. (2000). *El proceso unificado de desarrollo de software* (A. Otero, Ed.). Madrid: Pearson Educación.
82. Jara, O. (2018). *La sistematización de experiencias: práctica y teoría para otros mundos posibles* (Primera ed). Bogotá, Colombia: Fundación Centro Internacional de Educación y Desarrollo Humano.
83. Karpov, I. V, Sheblak, J., & Miikkulainen, R. (2008). OpenNERO : a Game Platform for AI Research and Education. En: *Fourth Artificial Intelligence and Interactive Digital Entertainment Conference OpenNERO.*, 220–221.
84. Kim, I. (2007). *UTBot : A Virtual Agent Platform for Teaching Agent System Design*. 2(1), 48–53.
85. Kummeneje, J., & Verhagen, H. (2005). Teaching Agent Programming to a Hybrid Student Population. *Innovation in Teaching and Learning in Information and Computer Sciences*, 4(3), 1–9.
86. Kusmina, N. (1970). *Metódicas investigativas de la actividad pedagógica*. Editorial Leningrado.
87. Labarrere, G., & Valdivia, G. (1988). *Pedagogía*. La Habana: Editorial Pueblo y Educación.
88. Lezcano, M. (2002). La enseñanza asistida por computadora y la enseñanza de la Inteligencia Artificial. In *Aplicaciones de la Inteligencia Artificial* (Primera Ed, pp. 372–386). Guadalajara.
89. Lezcano, M., & Valdés, G. (1998a). Logsim. Sistema de apoyo al aprendizaje de la programación lógica. *Revista Iberoamericana de Informática Educativa*, (9).
90. Lezcano, M., & Valdés, V. (1998b). Un ambiente integrado para la enseñanza de sistemas expertos en forma participativa. *Informática Educativa*, 11(1), 69–81.

91. Lombillo, I., Nambalo, J., Torres, A., & Pérez, B. (2018). La innovación educativa en el uso de los medios de enseñanza: una propuesta de solución que incluye las TIC. *Revista Cubana de Educación Superior*, 37(3).
92. López, B., Montaner, M., & de la Rosa, J. (2001). Utilización de un simulador de fútbol para enseñar inteligencia artificial a ingenieros. *Actas de Las Jornadas de Enseñanza Universitaria de La Informática*.
93. Manso, Y., Cañizares, R., & Febles, J. P. (2016). Diseño web adaptativo para la plataforma educativa ZERA. *Revista Cubana de Ciencias Informáticas*, 10(2), 100–115.
94. Marqués, P. (1995). *Metodología para la elaboración de software educativo* (E. Estel, Ed.). Barcelona (España).
95. Martínez, N. R. (2012). Reseña metodológica sobre los grupos focales. *Diálogos*, 6(9), 47–53.
96. Marx, C. (1875). *Crítica al programa de Gotha*. La Habana: Ciencias Sociales.
97. McCluskey, T., & Simpson, R. (2005). The Use of an Integrated Tool to Support Teaching and Learning in Artificial Intelligence. *Innovation in Teaching and Learning in Information and Computer Sciences*, 4(3). <https://doi.org/10.11120/ital.2005.04030007>
98. MCGovern, A., Tidwell, Z., & Rushing, D. (2011). Teaching Introductory Artificial Intelligence through Java-based Games. En: *Second Symposium on Educational Advances in Artificial Intelligence*, 1729–1737.
99. MES-CUJAE. (2017). *Plan de estudio E. Ingeniería Informática*. Universidad Tecnológica de La Habana. La Habana.
100. MES-UCI. (2019). *Plan de estudio E. Ingeniería en Ciencias Informáticas*. Universidad de las Ciencias Informáticas. La Habana.
101. MES-UH. (2008). *Programa de la Disciplina Inteligencia Artificial, Plan de Estudios D, Ciencia de la Computación*. Universidad de La Habana. La Habana.
102. MES-UH. (2018). *Plan de estudio E. Ciencia de la Computación*. Universidad de La Habana. La Habana.
103. MES. (2016). *Documento base para el diseño de los planes de estudio “E.”* Ministerio de Educación Superior. La Habana.
104. Mevarech, Z. ., & Light, P. . (1992). Cooperative Learning with Computers. *Learning and Instruction*, 2(3), 155–285.

105. Morales, M., Trujillo, J. M., & Raso, F. (2015). Percepciones acerca de la integración de las TIC en el proceso de enseñanza-aprendizaje de la universidad. *Pixel-Bit. Revista de Medios y Educación*, (46), 103–117.
106. Moya, M. (2013). De las TICs a las TACs: la importancia de crear contenidos educativos digitales. *Didáctica, Innovación y Multimedia (DIM)*, (27). Disponible en: <http://www.pangea.org/dim/revista.htm>
107. NetBeans. (2020). Apache NetBeans. Fits the Pieces Together. Disponible en: <https://netbeans.org/index.html>
108. Núñez, J. (1999). *La Ciencia y la Tecnología como procesos sociales. Lo que la educación científica no debería olvidar*. La Habana: Félix Varela.
109. Osella, G., De Vito, C., Russo, C., & Ramón, H. (2016). AutoPython: Una herramienta para la automatización de sesiones interactivas de Python. En: *XI Congreso de Educación en Tecnología y Tecnología en Educación*, 394–402.
110. Osorno, R. (2013). *Gestión de servicios como medio de asimilación de las TIC en las organizaciones*. Universitat Ramon Llull.
111. Páez, V. (2013). Propuesta de reconceptualización de la primera ley y principios didácticos. En *Proyecto de investigación: Sistematización teórica de la Pedagogía y la Didáctica en el contexto actual, para el perfeccionamiento de la formación profesional pedagógica* (pp. 12–24). La Habana: Universidad de Ciencias Pedagógicas “Enrique José Varona.”
112. Pantic, M., Zwitserloot, R., & Grootjans, R. (2005). Teaching Introductory Artificial Intelligence Using a Simple Agent Framework. *IEEE Transactions on Education*, 48(3), 382–390.
113. Pavón, R. (2017). *Metodología con aprovechamiento de las TIC para el desarrollo de la competencia comunicativa en inglés académico-profesional en la carrera de Ingeniería en Ciencias Informáticas*. Tesis en opción al grado científico de Doctor en Ciencias de la Educación. Universidad de La Habana.
114. PCC. (2017). *Documentos del 7mo. Congreso del Partido*. Disponible en: <http://www.granma.cu/file/pdf/gaceta/último PDF 32.pdf>
115. Pérez, B. (2015). *Modelo para la producción de hipervideos adaptativos basados en objetos de aprendizaje en la Universidad Agraria de La Habana*. Tesis en opción al grado científico de Doctor en Ciencias de la Educación. Universidad Agraria de La Habana.
116. Pérez de Maza, T. (2016). *Guía didáctica para la sistematización de experiencias en contextos universitarios* (Primera ed; U. N. Abierta, Ed.). Caracas: Ediciones del Vicerrectorado Académico.

117. Pola, J. S. (2014). *Modelo de virtualización de la formación en el Instituto Superior de Ciencias de la Educación de Luanda*. Tesis en opción al grado científico de Doctor en Ciencias de la Educación. Universidad de Ciencias Pedagógicas “Enrique José Varona”.
118. PostgreSQL. (2020). PostgreSQL: The World’s Most Advanced Open Source Relational Database. Recuperado el 8 de marzo de 2020, del sitio The PostgreSQL Global Development Group. Disponible en: <https://www.postgresql.org/>
119. RAE. (2020). Diccionario de la lengua española. <https://www.rae.es/>
120. Ramos, N. (2016). *Una metodología para el proceso pedagógico de desarrollo de software educativo de Química en la educación general cubana*. Tesis en opción al grado científico de Doctor en Ciencias de la Educación. Instituto Central de Ciencias Pedagógicas.
121. Renz, J. (2015). AIBIRDS: The Angry Birds Artificial Intelligence Competition Angry Birds as a Challenge for AI. En: *Twenty-Ninth AAAI Conference on Artificial Intelligence AIBIRDS*., 4326–4327.
122. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... Kafai, Y. (2009). Scratch: Programming for all. *Communications of the Acm*, 52(11), 60–67. <https://doi.org/10.1145/1592761.1592779>
123. Riedl, M. O. (2015). *A Python Engine for Teaching Artificial Intelligence in Games*. arXiv:1511.07714 [cs.CY]
124. Ríos, L. R., Lezcano, M., & Aljadis, A. (2008). Enseñando Prolog con mapas conceptuales. *Revista Cubana de Ciencias Informáticas*, 2(3–4), 65–70. Disponible en <https://www.redalyc.org/articulo.oa?id=378344354007>
125. Rodríguez-Travieso, R. (2017). *Modelo para el mejoramiento del desempeño profesional pedagógico de los directivos de la educación de jóvenes y adultos en la provincia Mayabeque*. Tesis en opción al grado científico de Doctor en Ciencias Pedagógicas. Universidad de Ciencias Pedagógicas “Enrique José Varona.”
126. Rodríguez, L. A. (2010). *Concepción didáctica del software educativo como instrumento mediador para un aprendizaje desarrollador*. Tesis en opción al grado científico de Doctor en Ciencias Pedagógicas. Universidad de Ciencias Pedagógicas “Félix Varela y Morales.”
127. Rombys-Estévez, D. (2013). Integración de las TIC para una “buena enseñanza”: opiniones, actitudes y creencias de los docentes en un instituto de formación de formadores. *Cuadernos de Investigación Educativa*, 4(19), 69–86.
128. Russell, S., & Norvig, P. (2010). *Artificial Intelligence. A modern approach* (Third Edit; M. Hirsch, Ed.). New Jersey: Prentice Hall.

129. Santos, G., Cenich, G., & Miranda, A. (2004). Entorno hipermedia para aprender: una alternativa para los ejemplos elaborados. *Pixel Bit. Revista de Medios y Educación*, (22), 89–98. Disponible en: <http://recyt.fecyt.es/index.php/pixel/index>
130. Silvestre, M. (2004). Aprendizaje: Problemas, retos y soluciones. En *Didáctica desarrolladora desde el enfoque histórico-cultural* (pp. 24–60). México: Ediciones CEIDE.
131. Sintov, Nicole. Kar, Debarun. Nguyen, Thanh. Fang, Fei. Hoffman, Kevin. Lyet, Arnaud. Tambe, Milind. (2017). Keeping it real: Using real-world problems to teach AI to diverse audiences. *AI MAGAZINE*, SUMMER, 35-48.
132. Sosnowski, S., Ernsberger, T., Cao, F., & Ray, S. (2013). SEPIA: A Scalable Game Environment for Artificial Intelligence Teaching. En: *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 1592–1597.
133. Stallman, R. (2020). Por qué las escuelas deben usar exclusivamente software libre. Recuperado el 27 de enero de 2020, desde el sitio El sistema operativo GNU. Disponible en: <https://www.gnu.org/education/edu-schools.es.html>
134. Taylor, M. E. (2010). Teaching Reinforcement Learning with Mario: An Argument and Case Study. En: *Second Symposium on Educational Advances in Artificial Intelligence Teaching*, 1737–1742.
135. Telecom Italia. (2019). Jade Site. Recuperado el 22 de octubre de 2019, desde el sitio JAVA Agent Development Framework. Disponible en: <https://jade.tilab.com/>
136. Torres, A. (2008). *Concepción teórico-metodológica para la producción de teleclases en formato digital destinadas a la nueva universidad cubana*. Tesis en opción al grado científico de Doctor en Ciencias de la Educación. Universidad Agraria de La Habana.
137. Torres, C. (2011). Técnicas de enseñanza y TIC en la Universidad. *Horizontes Educativos*, 16(2), 31–42.
138. Torres, S. V., Zangla, M. S., & Chiarani, M. C. (2016). Evaluación del impacto Repositorio CIE. En: *XI Congreso de Educación En Tecnología y Tecnología En Educación*, 557–564.
139. Trillo, M. P. (2012). Recursos Educativos en Abierto: evolución y modelos. *Foro de Educación*, (14), 191–205.
140. Turing, A. (1950). Computing machinery and intelligence. *Mind*, (49), 433–460.
141. UNESCO. (2012). *Declaración de París de 2012 sobre los REA*. París.

142. Urretavizcaya, M., & Onaindía, E. (2002). Docencia Universitaria de Inteligencia Artificial. *Revista Iberoamericana de Inteligencia Artificial*, (17), 23–30.
143. Valle, Al. (2012). *La investigación pedagógica. Otra mirada*. Ciudad de La Habana: Instituto Central de Ciencias Pedagógicas.
144. Veliz, O., Gutierrez, M., & Kiekintveld, C. (2016). Teaching Automated Strategic Reasoning Using Capstone Tournaments. En: *Sixth Symposium on Educational Advances in Artificial Intelligence*, 4134–4135.
145. Vidal, J., Buhle, P., & Goradia, H. (2005). Tools and Lessons from a Multiagent Systems'Class. *Innovation in Teaching and Learning in Information AndComputer Sciences*, 4(3), 1–15.
146. Waikato, U. of. (2019). Weka 3: Machine Learning Software in Java. Recuperado el 9 de julio de 2019. Disponible en: <https://www.cs.waikato.ac.nz/ml/weka/>
147. Wang, A. I., & Wu, B. (2009). *An Application of a Game Development Framework in Higher Education*. 2009. <https://doi.org/10.1155/2009/693267>
148. Wollowski, M., Neller, T., & Boerkoel, J. (2017). Artificial Intelligence Education. *AI MAGAZINE*, (SUMMER 2017).
149. Wollowski, M., Selkowitz, R., Brown, L. E., Goel, A., Luger, G., Marshall, J., ... Norvig, P. (2016). A survey of current practice and teaching of AI. En: *Sixth Symposium on Educational Advances in Artificial Intelligence (EAAI-16)*, 4119–4124.
150. Wong, D., Zink, R., & Koenig, S. (2010). *Teaching Artificial Intelligence and Robotics via Games [Abstract] **.
151. Zilberstein, J., & Silvestre, M. (2004). *Didáctica desarrolladora desde el enfoque histórico-cultural*. México: Ediciones CEIDE.

ANEXOS

Anexo 1. Operacionalización de la variable	ii
Anexo 2. Relación cualitativa-cuantitativa de los indicadores según la escala asumida.	iv
Anexo 3. Relación indicador-instrumento aplicado.....	xi
Anexo 4. Guía de entrevista a directivos y profesores	xii
Anexo 5. Cuestionario para evaluación cuantitativa de indicadores de la guía de entrevista.	xiii
Anexo 6. Resultados de la entrevista a directivos	xvi
Anexo 7. Resultados de la entrevista a profesores	xvii
Anexo 8. Encuesta aplicada a estudiantes.....	xviii
Anexo 9. Resultados de la encuesta a estudiantes.....	xx
Anexo 10. Guía para el análisis documental	xxi
Anexo 11. Resultados del análisis documental	xxii
Anexo 12. Guía de análisis del software	xxiii
Anexo 13. Resultados del análisis al software	xxiv
Anexo 14. Triangulación de métodos en el diagnóstico	xxv
Anexo 15. Criterio de expertos por método Delphi.....	xxvii
Anexo 16. Guion de contenido del software libre educativo SMEKDB.....	xxxii
Anexo 17. Imágenes del software SMEKDB	xxxv
Anexo 18. Diagnóstico aplicado a los estudiantes	xxxvi
Anexo 19. Resultados según la sistematización de experiencias	xxxvii
Anexo 20. Programa analítico del curso de postgrado IA-VIP	xxxviii
Anexo 21. Técnica de satisfacción de ladov	xli
Anexo 22. Técnica de grupos focales	xliii

Anexo 1. Operacionalización de la variable

Dimensión: Diseño didáctico del SLE		
Subdimensión	Índice	Indicador
Didáctica	1.1.1	Relación entre el software y los objetivos.
	1.1.2	Relación entre el software y los contenidos.
	1.1.3	Relación entre el software y las diferentes formas organizativas del PEA donde puede ser incorporado.
	1.1.4	Relación entre el software y los diversos métodos que pueden utilizarse para trabajar en el PEA.
	1.1.5	Relación entre el software y los diversos medios disponibles para trabajar en el PEA.
	1.1.6	Relación entre el software y la evaluación del estudiante.
Tecnológica	1.2.1	Accesibilidad al código fuente.
	1.2.2	Disponibilidad de documentación de ayuda.
	1.2.3	Organización de la arquitectura.
Espacial	1.3.1	Tareas diseñadas para el trabajo con el grupo.
	1.3.2	Tareas diseñadas para el trabajo en equipos.
	1.3.3	Tareas diseñadas para el trabajo individual.
Gestión	1.4.1	Características del software que garantizan su distribución.
	1.4.2	Características del software que garanticen su correcta utilización según las condiciones tecnológicas existentes.
Práctica	1.5.1	Tareas diseñadas para el estudio del código fuente del software.
	1.5.2	Tareas diseñadas para la modificación del código fuente del software.
	1.5.3	Tareas diseñadas para la reutilización del código fuente del software.

Dimensión: Incorporación armoniosa		
Subdimensión	Índice	Nombre
Contextualizada	2.1.1	Tareas a realizar con el software, relacionadas con la Práctica Profesional de los estudiantes.
	2.1.2	Tareas a realizar con el software, relacionadas con los intereses de los estudiantes.
Sistémica	2.2.1	Relación general entre el software y la asignatura.
	2.2.2	Impacto de la utilización del software en el proceso de enseñanza-aprendizaje.
Continua	2.3.1	Reutilización de los resultados de un tema a otro.
	2.3.2	Reutilización de los resultados de un curso a otro de la misma asignatura.
Reflexiva	2.4.1	Percepción de los profesores acerca del aporte de la utilización del software a su autosuperación.
	2.4.2	Percepción de los profesores acerca del aporte de la utilización del software al aprendizaje de sus estudiantes.
	2.4.3	Percepción de los estudiantes acerca del aporte del software a su futura vida profesional.
Planificada	2.5.1	Tiempo dedicado por el profesor al diseño didáctico del software.
	2.5.2	Clases en que se realizan tareas relacionadas con el software.
	2.5.3	Trabajo independiente dedicado por el estudiante para trabajar con el software.
	2.5.4	Tiempo dedicado por el profesor a su autopreparación.
	2.5.5	Retroalimentación del profesor sobre el proceso.

Anexo 2. Relación cualitativa-cuantitativa de los indicadores según la escala asumida.

Subdimensión	Indicador	5	4	3	2	1	0
Didáctica	1.1.1	Existen indicaciones precisas de cómo trabajar todos los objetivos de la asignatura con el software.	Existen indicaciones precisas de cómo trabajar varios objetivos de la asignatura con el software.	Existen indicaciones precisas de cómo trabajar algún objetivo de la asignatura con el software.	Aunque no existen indicaciones precisas se pueden trabajar varios objetivos de la asignatura con el software.	Aunque no existen indicaciones precisas se puede trabajar un objetivo con el software.	El software no permite trabajar ningún objetivo de la asignatura.
	1.1.2	Existen indicaciones precisas de cómo trabajar todos los contenidos de la asignatura con el software.	Existen indicaciones precisas de cómo trabajar varios contenidos de la asignatura con el software.	Existen indicaciones precisas de cómo trabajar algún contenido de la asignatura con el software.	Aunque no existen indicaciones precisas se pueden trabajar varios contenidos de la asignatura con el software.	Aunque no existen indicaciones precisas se puede trabajar un contenido con el software.	El software no permite trabajar ningún contenido de la asignatura.
	1.1.3	Existen indicaciones precisas de cómo trabajar con el software en todas las formas organizativas del PEA.	Existen indicaciones precisas de cómo trabajar con el software en varias formas organizativas del PEA.	Existen indicaciones precisas de cómo trabajar con el software en alguna forma organizativa del PEA.	Aunque no existen indicaciones precisas se puede trabajar con el software en varias formas organizativas del PEA.	Aunque no existen indicaciones precisas se puede trabajar con el software en alguna forma organizativa del PEA.	Con el software no se puede trabajar en ninguna forma organizativa del PEA.
	1.1.4	Existen indicaciones precisas de cómo utilizar más de tres métodos a la hora de trabajar con el software.	Existen indicaciones precisas de cómo utilizar dos o tres métodos a la hora de trabajar con el software.	Existen indicaciones precisas de cómo utilizar algún método a la hora de trabajar con el software.	Aunque no existen indicaciones precisas se pueden utilizar varios métodos a la hora de trabajar con el software.	Aunque no existen indicaciones precisas se puede utilizar algún método a la hora de trabajar con el software.	El software no es posible utilizarlo en el PEA.
	1.1.5	Existen indicaciones precisas para la utilización de un sistema de medios que	Existen indicaciones precisas para la utilización de varios medios como apoyo al	Existen indicaciones precisas para la utilización de medios como apoyo al trabajo	Aunque no existen indicaciones precisas, se pueden utilizar varios medios	Aunque no existen indicaciones precisas, se pueden utilizar	No hay condiciones para apoyar con otros medios el trabajo con

Subdimensión	Indicador	5	4	3	2	1	0
	Tecnológica		apoyen el trabajo con el software, incluido el uso de los propios medios de los estudiantes.	trabajo con el software, tanto en plataformas, laboratorios y aulas.	con el software, ya sea en plataformas, laboratorios o aulas.	como apoyo al trabajo con el software, tanto en plataformas, laboratorios y aulas.	medios como apoyo al trabajo con el software, ya sea en plataformas, laboratorios o aulas.
1.1.6		Existen indicaciones precisas para evaluar de forma integral el desarrollo del estudiante en la asignatura, a partir de su trabajo con el software.	Existen indicaciones precisas para tomar en cuenta el trabajo con el software en la nota final del estudiante.	Existen indicaciones precisas para tomar en cuenta el trabajo con el software como evaluación parcial del estudiante.	Existen indicaciones precisas para tomar en cuenta el trabajo con el software como evaluación frecuente del estudiante.	Aunque no existen indicaciones precisas se puede tomar en cuenta el trabajo con el software para evaluar al estudiante.	El software no es adecuado para evaluar al estudiante.
1.2.1		El código fuente del software es perfectamente legible en su totalidad.	El código fuente del software es legible en más de un 70%.	Ciertas partes del código fuente del software son perfectamente legibles.	Ciertas partes del código fuente del software son legibles aunque con alguna dificultad.	El código fuente del software está disponible pero es poco legible.	El código fuente del software no está disponible.
1.2.2		Existe un guion de contenido o guía de utilización en el PEA. Además existe documentación de ayuda en el software, en medios digitales y en plataformas en línea.	Existe un guion de contenido o guía de utilización en el PEA. Aunque no existan otras formas de ayuda.	Aunque no existe guion de contenido o guía de utilización, existe documentación en el mismo software, en medios digitales y en plataformas en línea.	Existe documentación de ayuda en el software y algún otro formato.	La ayuda solo está disponible o bien en el mismo software o en otro formato.	No existe documentación de ayuda.
1.2.3		Se utiliza un paradigma de programación adecuado para la enseñanza de la IA. La arquitectura está organizada con facilidades para modificar y adicionar nuevas	Se utiliza un paradigma de programación de IA. La arquitectura brinda facilidades para modificar y adicionar nuevas funcionalidades.	Se utiliza un paradigma de programación de IA, pero la arquitectura no brinda facilidades para modificar o adicionar nuevas funcionalidades.	Aunque no se utiliza un paradigma de programación de IA, la arquitectura brinda facilidades para modificar o adicionar nuevas	Aunque no se utiliza un paradigma de programación de IA, la arquitectura permite modificar o adicionar nuevas funcionalida-	La arquitectura del software no permite que sea extensible. Se hace complejo el estudio o modificación del

Subdimensión	Indicador	5	4	3	2	1	0
		funcionalidades.			funcionalidades.	des.	código fuente.
Espacial	1.3.1	Están bien definidas y elaboradas más de tres tareas para el trabajo en grupo.	Están bien definidas y elaboradas dos o tres tareas para el trabajo en grupo.	Está bien definida y elaborada una tarea para el trabajo en grupo.	Se pueden inferir más de tres tareas posibles para el trabajo en grupo.	Se pueden inferir hasta tres tareas posibles para el trabajo en grupo.	No es posible asignar tareas para el trabajo con el grupo.
	1.3.2	Están bien definidas y elaboradas más de tres tareas para el trabajo en equipo.	Están bien definidas y elaboradas dos o tres tareas para el trabajo en equipo.	Está bien definida y elaborada una tarea para el trabajo en equipo.	Se pueden inferir más de tres tareas posibles para el trabajo en equipo.	Se pueden inferir hasta tres tareas posibles para el trabajo en equipo.	No es posible asignar tareas para el trabajo en equipo.
	1.3.3	Están bien definidas y elaboradas más de tres tareas para el trabajo individual.	Están bien definidas y elaboradas dos o tres tareas para el trabajo individual.	Está bien definida y elaborada una tarea para el trabajo individual.	Se pueden inferir más de tres tareas posibles para el trabajo individual.	Se pueden inferir hasta tres tareas posibles para el trabajo individual.	No es posible asignar tareas para el trabajo individual.
Gestión	1.4.1	Está amparado en una licencia libre, es multiplataforma y se encuentra disponible en alguna plataforma de acceso libre y permanente.	Está amparado en una licencia libre, es multiplataforma y se encuentra disponible en la red al menos en horario laboral.	Está amparado en una licencia libre, no es multiplataforma y se encuentra disponible en la red al menos en horario laboral.	Está amparado en una licencia libre y hay posibilidades de acceso y distribución.	Licencia restringida pero disponible por alguna vía.	Licencia restringida y no disponible en la red.
	1.4.2	Sus características están acorde a las condiciones existentes en aulas y laboratorios.	Sus características están acordes al equipamiento en laboratorios, pero es muy limitada la posibilidad de usarlo en las aulas.	Sus características permiten utilizarlo con ciertas limitaciones en los laboratorios, pero está acorde a las condiciones existentes en las aulas.	Sus características permiten utilizarlo con ciertas limitaciones en los laboratorios, pero no es posible utilizarlo en las aulas.	Sus características permiten utilizarlo con ciertas limitaciones en las aulas, pero en los laboratorios no es posible el trabajo personalizado.	Sus características no permiten utilizarlo en los laboratorios ni en las aulas.
Práctica	1.5.1	Están bien definidas y elaboradas más de tres tareas para el estudio del código fuente.	Están bien definidas y elaboradas dos o tres tareas para el estudio del código	Está bien definida y elaborada una tarea de estudio del código fuente.	Se pueden inferir más de tres tareas posibles para el estudio del código fuente.	Se pueden inferir hasta tres tareas posibles para el estudio del código	No es posible asignar tareas para el estudio del código

Subdimensión	Indicador	5	4	3	2	1	0
				fuelle.			
	1.5.2	Están bien definidas y elaboradas más de tres tareas para la modificación del código fuente.	Están bien definidas y elaboradas dos o tres tareas para la modificación del código fuente.	Está bien definida y elaborada una tarea de modificación del código fuente.	Se pueden inferir más de tres tareas posibles para la modificación del código fuente.	Se pueden inferir hasta tres tareas posibles para la modificación del código fuente.	No es posible asignar tareas para la modificación del código fuente
	1.5.3	Están bien definidas y elaboradas más de tres tareas para la reutilización del código fuente.	Están bien definidas y elaboradas dos o tres tareas para la reutilización del código fuente.	Está bien definida y elaborada una tarea para la reutilización del código fuente.	Se pueden inferir más de tres tareas posibles para la reutilización del código fuente.	Se pueden inferir hasta tres tareas posibles para la reutilización del código fuente.	No es posible asignar tareas para la reutilización del código fuente.

Dimensión incorporación armoniosa

Subdimensión	Indicador	5	4	3	2	1	0
Contextualizada	2.1.1	Existen vías bien definidas para conocer el trabajo en la PP y se toman en cuenta a la hora de asignar las tareas a todos los estudiantes.	Existen vías bien definidas para conocer el trabajo en la PP y se toman en cuenta a la hora de asignar las tareas a más de un 50% de los estudiantes.	Existen vías bien definidas para conocer el trabajo en la PP y se toman en cuenta a la hora de asignar las tareas a algunos estudiantes aventajados.	Aunque no existen vías bien definidas para conocer el trabajo en la PP, se toman en cuenta a la hora de asignar las tareas a varios estudiantes.	Aunque no existen vías bien definidas para conocer el trabajo en la PP, se toman en cuenta a la hora de asignar las tareas a algunos estudiantes.	No se toma en cuenta el trabajo en la PP de los estudiantes.
	2.1.2	Existen vías bien definidas para conocer sus intereses y se toman en cuenta a la hora de	Existen vías bien definidas para conocer sus intereses y se toman en cuenta a la hora de asignar las tareas a más de un 50%	Existen vías bien definidas para conocer sus intereses y se toman en cuenta a la hora de asignar las tareas a algunos	Aunque no existen vías bien definidas para conocer sus intereses, se toman en cuenta a la hora de asignar las tareas a	Aunque no existen vías bien definidas para conocer sus intereses, se toman en cuenta a la hora de asignar las	No se toman en cuenta los intereses de los estudiantes.

Subdimensión	Indicador	5	4	3	2	1	0
		asignar las tareas a todos los estudiantes.	de los estudiantes.	estudiantes aventajados.	varios estudiantes.	tareas a algunos estudiantes.	
Sistémica	2.2.1	El software fue diseñado para utilizarse en toda la asignatura.	El software fue diseñado para utilizarse en gran parte de la asignatura.	El software fue diseñado para utilizarse en una parte muy específica de la asignatura.	El software puede ser utilizado en toda la asignatura.	El software puede ser utilizado en parte de la asignatura.	El software no tiene relación alguna con la asignatura.
	2.2.2	Sin el software sería imposible llevar a cabo el PEA.	La no utilización del software afectaría el PEA.	La no utilización del software afectaría alguna parte específica del PEA.	Se reconoce como útil la utilización del software para el PEA.	No es necesario utilizar el software para llevar a cabo el PEA de forma satisfactoria.	La utilización del software afecta el PEA.
Continua	2.3.1	Se reutilizan los resultados de cada estudiante en las tareas de los siguientes temas.	Sólo se dejan de reutilizar algunos casos por razones de calidad o interés del profesor.	Se reutilizan algunos resultados de los estudiantes más aventajados en las tareas de los siguientes temas.	Se reutilizan algunos resultados por interés del propio estudiante.	Sólo en casos excepcionales se reutilizan los resultados.	No se reutilizan resultados de un tema a otro.
	2.3.2	Se analizan todas las soluciones de los estudiantes y se evalúan los cambios al software para reutilizar todo lo que sea posible.	Se analizan todas las soluciones de los estudiantes y se evalúan para hacer algún cambio muy específico en el software.	Se analizan algunas soluciones de los estudiantes y se valora cuáles y de qué forma reutilizar.	Sólo en casos excepcionales se analizan posibles cambios al software reutilizando las soluciones.	En casos excepcionales se reutiliza la solución brindada por algún estudiante como nueva versión del software.	No se reutilizan resultados de un curso a otro.
Reflexiva	2.4.1	El trabajo con el software tributa a su tema de investigación y le ha valido para	El trabajo con el software tributa a su tema de investigación y le ha valido para trabajos en eventos o	El trabajo con el software tributa a la línea de investigación y se reconoce como importante para	Aunque el trabajo con el software no tributa a su investigación, se reconoce como importante para	Es indiferente la utilización o no del software.	Solo roba tiempo que pudiera utilizarse en la superación personal.

Subdimensión	Indicador	5	4	3	2	1	0
		vencer objetivos de superación como maestrías o doctorados.	publicaciones.	la autosuperación.	su autopsuperación.		
	2.4.2	El trabajo con el software tributa tanto al vencimiento de los objetivos como al desarrollo de las habilidades concretas de la asignatura. Así como al desarrollo de habilidades generales para su futura vida profesional.	El trabajo con el software tributa al vencimiento de algunos objetivos y al desarrollo de algunas habilidades concretas de la asignatura. Desarrolla habilidades generales para su futura vida profesional.	El trabajo con el software tributa al vencimiento de algunos objetivos y al desarrollo de algunas habilidades concretas de la asignatura.	El trabajo con el software aporta a alguna habilidad específica de la asignatura.	El trabajo con el software no aporta elementos de interés.	El trabajo con el software solo roba tiempo de aprendizaje a los estudiantes.
	2.4.3	El trabajo con el software tributa a su trabajo como informático, podrán estudiar o reutilizar el código en futuros trabajos y seguirán temas de investigación relacionados con el tema.	El trabajo con el software tributa a su trabajo como informático, podrán estudiar o reutilizar el código en futuros trabajos.	El trabajo con el software aporta conocimientos específicos de la asignatura, y aporta elementos para su futuro trabajo profesional.	El trabajo con el software aporta elementos de interés para la asignatura, pero no aporta significativamente para el futuro trabajo profesional.	El trabajo con el software no aporta elementos de interés para la asignatura, ni para el futuro trabajo profesional.	El trabajo con el software solo roba tiempo de aprendizaje.

Subdimensión	Indicador	5	4	3	2	1	0
Planificada	2.5.1	El profesor dedica más de 20 horas totales para el diseño didáctico del software.	El profesor dedica entre 12 y 20 horas totales para el diseño didáctico del software.	El profesor dedica entre seis y 12 horas totales para el diseño didáctico del software.	El profesor dedica entre tres y 6 horas totales para el diseño didáctico del software.	El profesor dedica menos de tres horas totales para el diseño didáctico del software.	El profesor no dedica tiempo para el diseño didáctico del software.
	2.5.2	Se planifican tareas con el software y se realizan en más del 50% de las clases, según plan calendario de la asignatura.	Se planifican tareas con el software y se realizan en al menos el 20% de las clases, según plan calendario de la asignatura.	Aunque se planifican tareas en varias clases según plan calendario de la asignatura, se realizan en menos de un 20%.	No se planifican tareas con el software, pero se realizan tareas en algunas clases.	En las clases solo se orienta la utilización del software por el estudiante.	No se orienta en clases la utilización del software.
	2.5.3	Se planifican, asignan y controlan más de seis horas de trabajo independiente del estudiante.	Se planifican, asignan y controlan entre dos y seis horas de trabajo independiente del estudiante.	Se asigna el trabajo independiente con el software y se controla a través de resultados parciales.	Se asigna el trabajo independiente con el software y se controla a través del resultado final.	Se orienta la utilización del software pero nunca se controla.	No se orienta trabajo independiente a los estudiantes para utilizar el software.
	2.5.4	El profesor dedica más de 20 horas totales para su autopreparación.	El profesor dedica entre 12 y 20 horas totales para su autopreparación.	El profesor dedica entre seis y 12 horas totales para su autopreparación.	El profesor dedica entre tres y seis horas totales para su autopreparación.	El profesor dedica menos de tres horas totales para su autopreparación.	El profesor no dedica tiempo para su autopreparación.
	2.5.5	El profesor planifica y ejecuta más de dos vías de retroalimentación del proceso.	El profesor planifica y ejecuta una o dos vías de retroalimentación del proceso.	El profesor planifica y ejecuta una o dos vías de retroalimentación del proceso.	El profesor utiliza alguna vía de retroalimentación del proceso.	El profesor indaga de forma intuitiva sobre el proceso.	No hay retroalimentación sobre el proceso.

Anexo 3. Relación indicador-instrumento aplicado

Indicador	I1. Entrevista a directivos	I2. Entrevista a profesores	I3. Encuesta a estudiantes	I4. Análisis documental	I5. Análisis del software
1.1.1	X	X		X	X
1.1.2	X	X	X	X	X
1.1.3	X	X	X	X	
1.1.4	X	X			X
1.1.5	X	X	X		X
1.1.6	X	X		X	
1.2.1	X	X	X		X
1.2.2	X	X	X		X
1.2.3	X	X	X		X
1.3.1	X	X		X	X
1.3.2	X	X		X	X
1.3.3	X	X		X	X
1.4.1	X	X	X		X
1.4.2	X	X	X		X
1.5.1	X	X		X	X
1.5.2	X	X		X	X
1.5.3	X	X		X	X
2.1.1	X	X	X	X	
2.1.2	X	X	X	X	
2.2.1	X	X	X	X	X
2.2.2	X	X	X	X	
2.3.1	X	X	X		
2.3.2	X	X		X	X
2.4.1	X	X		X	
2.4.2	X	X		X	
2.4.3		X	X		
2.5.1	X	X			
2.5.2	X	X		X	
2.5.3	X	X	X	X	
2.5.4	X	X			
2.5.5	X	X		X	

Anexo 4. Guía de entrevista a directivos y profesores

Objetivo: Obtener criterios emitidos a viva voz relacionados con la integración de aplicaciones informáticas que permitan el acceso al código fuente en las asignaturas de Inteligencia Artificial.

Nombre del entrevistado: _____

1. Datos generales

Años de experiencia	1 <= X < 5	5 <= X < 10	10 <= X < 15	X >= 15
	___	___	___	___
Categoría docente	Instructor	Asistente	Profesor Auxiliar	Profesor Titular
	___	___	___	___
Formación académica	Universitario		Máster	Doctor en Ciencia
	___		___	___
Perfil graduado de	Ingeniería	Licenciatura	Pedagógico	Otro (especifique)
	___	___	___	___

2. Sobre el diseño didáctico del software

- 2.1 ¿Qué relación observa entre el software utilizado en las asignaturas y cada uno de los componentes de la didáctica?
- 2.2 ¿Qué características del software facilitan y cuáles entorpecen su utilización en el PEA?
- 2.3 Explique si el software brinda facilidades para asignar tareas a los estudiantes de forma individual, por equipo y grupales.
- 2.4 ¿El software tiene las características necesarias para ser distribuido y utilizado en las condiciones tecnológicas actuales?
- 2.5 Explique si el software brinda facilidades para asignar tareas a los estudiantes para estudiar, modificar o reutilizar su código fuente.

3. Sobre su incorporación al PEA

- 3.1 Explique si asigna tareas a los estudiantes tomando en cuenta sus intereses o las actividades que realizan en la Práctica Profesional.
- 3.2 ¿Qué relación general aprecia entre el software y la asignatura?
- 3.3 Explique si le da continuidad a los resultados que se obtienen del trabajo de los estudiantes con el software.
- 3.4 Exponga cuánto aporta el trabajo con el software a la autosuperación personal de los profesores y a los estudiantes como futuros profesionales.
- 3.5 Exponga las actividades que realiza de planificación para utilizar el software en el PEA.

4. Conclusiones de la entrevista

Anexo 5. Cuestionario para evaluación cuantitativa de indicadores de la guía de entrevista.

Objetivo: Se aplica a directivos y profesores entrevistados para que asignen un valor cuantitativo a las respuestas cualitativas que ofrecen a cada pregunta.

1. Sobre el software utilizado en clases

Relación entre el software y los objetivos.

Respuesta: 5 __, 4 __, 3 __, 2 __, 1 __, 0 __

Relación entre el software y los contenidos.

Respuesta: 5 __, 4 __, 3 __, 2 __, 1 __, 0 __

Relación entre el software y las diversas formas organizativas del PEA.

Respuesta: 5 __, 4 __, 3 __, 2 __, 1 __, 0 __

Relación entre el software y los diversos métodos de enseñanza que pueden utilizarse.

Respuesta: 5 __, 4 __, 3 __, 2 __, 1 __, 0 __

Relación entre el software y los diversos medios disponibles para trabajar con él.

Respuesta: 5 __, 4 __, 3 __, 2 __, 1 __, 0 __

Relación entre el software y la evaluación del estudiante.

Respuesta: 5 __, 4 __, 3 __, 2 __, 1 __, 0 __

Accesibilidad al código fuente del software.

Respuesta: 5 __, 4 __, 3 __, 2 __, 1 __, 0 __

Disponibilidad de documentación de ayuda.

Respuesta: 5 __, 4 __, 3 __, 2 __, 1 __, 0 __

Organización de la arquitectura del software.

Respuesta: 5 __, 4 __, 3 __, 2 __, 1 __, 0 __

Tareas diseñadas para el trabajo con el grupo.

Respuesta: 5 __, 4 __, 3 __, 2 __, 1 __, 0 __

Tareas diseñadas para el trabajo en equipos.

Respuesta: 5 __, 4 __, 3 __, 2 __, 1 __, 0 __

Tareas diseñadas para el trabajo individual.

Respuesta: 5 __, 4 __, 3 __, 2 __, 1 __, 0 __

Características del software que garanticen su distribución en el PEA.

Respuesta: 5 __, 4 __, 3 __, 2 __, 1 __, 0 __

Características del software que garanticen su correcta utilización según las condiciones tecnológicas existentes.

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

Tareas diseñadas para el estudio del código fuente del software.

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

Tareas diseñadas para la modificación del código fuente del software.

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

Tareas diseñadas para la reutilización del código fuente del software.

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

2. Sobre la incorporación del software al PEA

Tareas a realizar con el software, relacionadas con la Práctica Profesional de los estudiantes.

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

Tareas a realizar con el software, relacionadas con los intereses de los estudiantes.

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

Relación general entre el software y la asignatura.

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

Impacto de la utilización del software en el proceso de enseñanza-aprendizaje.

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

Reutilización de los resultados de un tema a otro.

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

Reutilización de los resultados de un curso a otro de la misma asignatura.

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

Percepción de los profesores acerca del aporte de la utilización del software a su autosuperación.

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

Percepción de los profesores acerca del aporte de la utilización del software al aprendizaje de sus estudiantes.

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

Percepción de los estudiantes acerca del aporte del software a su futura vida profesional.

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

Tiempo dedicado por el profesor al diseño didáctico del software.

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

Clases en que se realizan tareas relacionadas con el software.

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

Trabajo independiente dedicado por el estudiante para trabajar con el software.

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

Tiempo dedicado por el profesor a su autoperseparación.

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

Retroalimentación del profesor sobre el proceso.

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

Anexo 6. Resultados de la entrevista a directivos

Dimensión	Subdimensión	Indicador	Respuestas						Indicador		subdimensión		dimensión		VOE	
			5	4	3	2	1	0	Índice	Nivel	Índice	Nivel	Índice	Nivel	Índice	Nivel
1	1.1	1.1.1	0	0	0	6	0	0	2.00	MEDIO	2.28	MEDIO	1.98	BAJO	1.64	BAJO
		1.1.2	0	0	0	5	1	0	1.83	BAJO						
		1.1.3	0	0	3	2	1	0	2.33	MEDIO						
		1.1.4	0	0	1	3	2	0	1.83	BAJO						
		1.1.5	0	0	4	2	0	0	2.67	MEDIO						
		1.1.6	0	0	6	0	0	0	3.00	MEDIO						
	1.2	1.2.1	0	0	1	3	2	0	1.83	BAJO	1.67	BAJO				
		1.2.2	0	0	3	3	0	0	2.50	MEDIO						
		1.2.3	0	0	0	0	4	2	0.67	BAJO						
	1.3	1.3.1	0	0	0	0	6	0	1.00	BAJO	1.28	BAJO				
		1.3.2	0	0	0	0	6	0	1.00	BAJO						
		1.3.3	0	0	0	5	1	0	1.83	BAJO						
	1.4	1.4.1	2	4	0	0	0	0	4.33	ALTO	4.50	ALTO				
		1.4.2	4	2	0	0	0	0	4.67	ALTO						
	1.5	1.5.1	0	0	0	4	2	0	1.67	BAJO	0.72	BAJO				
1.5.2		0	0	0	0	1	5	0.17	BAJO							
1.5.3		0	0	0	0	2	4	0.33	BAJO							
2	2.1	2.1.1	0	0	0	0	0	6	0.00	BAJO	0.00	BAJO				
		2.1.2	0	0	0	0	0	6	0.00	BAJO						
	2.2	2.2.1	0	0	0	1	5	0	1.17	BAJO	2.00	MEDIO				
		2.2.2	0	1	3	2	0	0	2.83	MEDIO						
	2.3	2.3.1	0	0	0	0	0	6	0.00	BAJO	0.00	BAJO				
		2.3.2	0	0	0	0	0	6	0.00	BAJO						
	2.4	2.4.1	0	0	0	1	5	0	1.17	BAJO	1.58	BAJO				
		2.4.2	0	0	1	4	1	0	2.00	MEDIO						
	2.5	2.5.1	0	0	0	0	4	2	0.67	BAJO	1.70	BAJO				
		2.5.2	0	0	6	0	0	0	3.00	MEDIO						
		2.5.3	0	0	3	3	0	0	2.50	MEDIO						
2.5.4		0	0	0	2	4	0	1.33	BAJO							
2.5.5		0	0	0	0	6	0	1.00	BAJO							

Anexo 7. Resultados de la entrevista a profesores

Dimensión	Subdimensión	Indicador	Respuestas					Indicador		subdimensión		dimensión		VOE		
			5	4	3	2	1	0	Índice	Nivel	Índice	Nivel	Índice	Nivel	Índice	Nivel
1	1.1	1.1.1	0	0	0	11	0	0	2.00	MEDIO	2.12	MEDIO	1.86	BAJO	1.65	BAJO
		1.1.2	0	0	0	6	5	0	1.55	BAJO						
		1.1.3	0	0	5	4	2	0	2.27	MEDIO						
		1.1.4	0	0	2	5	4	0	1.82	BAJO						
		1.1.5	0	0	4	4	3	0	2.09	MEDIO						
		1.1.6	0	0	11	0	0	0	3.00	MEDIO						
	1.2	1.2.1	0	0	1	9	1	0	2.00	MEDIO	1.70	BAJO				
		1.2.2	0	0	5	5	1	0	2.36	MEDIO						
		1.2.3	0	0	0	0	8	3	0.73	BAJO						
	1.3	1.3.1	0	0	0	2	9	0	1.18	BAJO	1.24	BAJO				
		1.3.2	0	0	0	2	9	0	1.18	BAJO						
		1.3.3	0	0	0	4	7	0	1.36	BAJO						
	1.4	1.4.1	5	5	1	0	0	0	4.36	ALTO	4.05	ALTO				
		1.4.2	2	4	5	0	0	0	3.73	MEDIO						
	1.5	1.5.1	0	0	0	6	5	0	1.55	BAJO	0.67	BAJO				
		1.5.2	0	0	0	0	3	8	0.27	BAJO						
		1.5.3	0	0	0	0	2	9	0.18	BAJO						
	2	2.1	2.1.1	0	0	0	0	0	11	0.00	BAJO	0.23				
2.1.2			0	0	0	0	5	6	0.45	BAJO						
2.2		2.2.1	0	0	0	1	10	0	1.09	BAJO	2.05	MEDIO				
		2.2.2	0	2	7	2	0	0	3.00	MEDIO						
2.3		2.3.1	0	0	0	0	4	7	0.36	BAJO	0.18	BAJO				
		2.3.2	0	0	0	0	0	11	0.00	BAJO						
2.4		2.4.1	0	0	0	8	2	1	1.64	BAJO	2.12	MEDIO				
		2.4.2	0	0	4	5	2	0	2.18	MEDIO						
		2.4.3	0	0	0	10	1	0	1.91	BAJO						
2.5		2.5.1	0	0	0	0	7	4	0.64	BAJO	1.62	BAJO				
		2.5.2	0	0	11	0	0	0	3.00	MEDIO						
		2.5.3	0	0	7	4	0	0	2.64	MEDIO						
		2.5.4	0	0	0	5	6	0	1.45	BAJO						
		2.5.5	0	0	0	0	11	0	1.00	BAJO						

Anexo 8. Encuesta aplicada a estudiantes

Objetivo: Obtener información de los estudiantes sobre la utilización de aplicaciones informáticas en las clases de Inteligencia Artificial.

Estimada (o) estudiante: Esta encuesta forma parte de una investigación de doctorado. Su objetivo es conocer acerca de la utilización de aplicaciones informáticas en las clases de Inteligencia Artificial. Sus resultados serán confidenciales y solo se utilizarán con motivos científicos. Le pedimos de favor expresar con la mayor fidelidad posible sus criterios.

Analice detenidamente los aspectos tratados en cada pregunta y brinde su respuesta marcando según la escala comprendida desde el valor cinco (5) hasta el valor cero (0) en orden descendente.

1. Sobre el diseño didáctico del software

¿Cómo evalúa la relación entre el software y los contenidos de la asignatura?

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

¿Cómo evalúa la cantidad de clases en que se utiliza el software?

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

¿Cómo evalúa la utilización de medios disponibles en el aula y los laboratorios para trabajar con el software?

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

¿Cómo evalúa la legibilidad del código fuente?

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

¿Cómo evalúa la disponibilidad de documentación de ayuda?

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

¿Cómo evalúa la organización de la arquitectura del software?

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

¿Cómo evalúa las características del software que garanticen su distribución en clases?

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

¿Cómo evalúa las características del software que garanticen su correcta utilización en los laboratorios y aulas?

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

2. Sobre la incorporación del software a las clases de Inteligencia Artificial

¿Cómo evalúa las tareas que realiza con el software, relacionadas con su Práctica Profesional?

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

¿Cómo evalúa las tareas que realiza con el software, relacionadas con sus intereses personales o profesionales?

Respuesta: 5__, 4__, 3__, 2__, 1__, 0__

¿Cómo evalúa la relación general entre el software y la asignatura?

Respuesta: 5 __, 4 __, 3 __, 2 __, 1 __, 0 __

¿Cómo evalúa el impacto de la utilización del software a la calidad de las clases?

Respuesta: 5 __, 4 __, 3 __, 2 __, 1 __, 0 __

¿Cómo evalúa la reutilización de sus resultados con el software de un tema a otro en la asignatura?

Respuesta: 5 __, 4 __, 3 __, 2 __, 1 __, 0 __

¿Cómo evalúa el aporte de la utilización del software a su futura vida profesional?

Respuesta: 5 __, 4 __, 3 __, 2 __, 1 __, 0 __

¿Cómo evalúa el trabajo independiente dedicado por ud para trabajar con el software?

Respuesta: 5 __, 4 __, 3 __, 2 __, 1 __, 0 __

Anexo 9. Resultados de la encuesta a estudiantes

Dimensión	Subdimensión	Indicador	Respuestas						índice		subdimensión		Dimensión		variable	
			5	4	3	2	1	0	Índice	Nivel	Índice	Nivel	Índice	Nivel	Índice	Nivel
1	1.1	1.1.2	0	0	0	16	37	0	1.30	BAJO	1.87	BAJO	1.99	BAJO	1.59	BAJO
		1.1.3	0	0	1	51	1	0	2.00	MEDIO						
		1.1.5	0	0	16	37	0	0	2.30	MEDIO						
	1.2	1.2.1	0	0	0	0	53	0	1.00	BAJO	0.83	BAJO				
		1.2.2	0	0	0	11	42	0	1.21	BAJO						
		1.2.3	0	0	0	0	15	38	0.28	BAJO						
	1.4	1.4.1	0	46	7	0	0	0	3.87	MEDIO	3.92	MEDIO				
		1.4.2	0	52	1	0	0	0	3.98	MEDIO						
	2	2.1	2.1.1	0	0	0	0	0	53	0.00	BAJO	0.26				
2.1.2			0	0	0	0	28	25	0.53	BAJO						
2.2		2.2.1	0	0	0	2	49	2	1.00	BAJO	1.67	BAJO				
		2.2.2	0	0	20	31	2	0	2.34	MEDIO						
2.3		2.3.1	0	0	0	0	11	42	0.21	BAJO	0.21	BAJO				
2.4		2.4.3	0	0	0	0	53	0	1.00	BAJO	1.00	BAJO				
2.5		2.5.3	0	4	36	11	2	0	2.79	MEDIO	2.79	MEDIO				

Anexo 10. Guía para el análisis documental

Objetivo: Revisar cómo se refleja la integración de aplicaciones informáticas al proceso de enseñanza-aprendizaje, en los documentos metodológicos e informes de la disciplina y asignaturas de Inteligencia Artificial.

Aspectos a revisar:

- Relación entre el software utilizado y los componentes de la didáctica: Objetivos, contenidos, formas organizativas y evaluación del estudiante.
- Elaboración y asignación de tareas al estudiante para resolver con el software de manera individual, en equipo y en grupo.
- Elaboración y asignación de tareas al estudiante para estudiar, modificar y reutilizar el código fuente del software.
- Asignación de tareas al estudiante, relacionadas con su PP y sus propios intereses personales y profesionales.
- Relación general entre el software y la asignatura.
- Impacto del software en el PEA.
- Reutilización de resultados de los estudiantes para enriquecer el software.
- Percepción de los profesores acerca del aporte de la utilización del software a su autosuperación.
- Percepción de los profesores acerca del aporte de la utilización del software al aprendizaje de sus estudiantes.
- Cantidad de clases en que se planifican y realizan tareas relacionadas con el software.
- Resultados del trabajo independiente, asignado a los estudiantes, relacionado con el software.
- Vías de retroalimentación que utiliza el profesor sobre la utilización del software.

Anexo 11. Resultados del análisis documental

Dimensión	Subdimensión	Indicador	Respuestas						índice		subdimensión		dimensión		VOE	
			5	4	3	2	1	0	Índice	Nivel	Índice	Nivel	Índice	Nivel	Índice	Nivel
1	1.1	1.1.1	0	0	0	1	0	0	2.00	MEDIO	2.25	MEDIO	2.00	MEDIO	1.55	BAJO
		1.1.2	0	0	0	1	0	0	2.00	MEDIO						
		1.1.3	0	0	0	1	0	0	2.00	MEDIO						
		1.1.6	0	0	1	0	0	0	3.00	MEDIO						
	1.3	1.3.1	0	0	0	1	0	0	2.00	MEDIO	1.67	BAJO				
		1.3.2	0	0	0	0	1	0	1.00	BAJO						
		1.3.3	0	0	0	1	0	0	2.00	MEDIO						
	1.5	1.5.1	0	0	0	1	0	0	2.00	MEDIO	2.00	MEDIO				
		1.5.2	0	0	0	1	0	0	2.00	MEDIO						
1.5.3		0	0	0	1	0	0	2.00	MEDIO							
2	2.1	2.1.1	0	0	0	0	0	1	0.00	BAJO	0.00	BAJO				
		2.1.2	0	0	0	0	0	1	0.00	BAJO						
	2.2	2.2.1	0	0	0	0	1	0	1.00	BAJO	1.50	BAJO				
		2.2.2	0	0	0	1	0	0	2.00	MEDIO						
	2.3	2.3.2	0	0	0	0	0	1	0.00	BAJO	0.00	BAJO				
	2.4	2.4.1	0	0	0	0	1	0	1.00	BAJO	1.50	BAJO				
		2.4.2	0	0	0	1	0	0	2.00	MEDIO						
	2.5	2.5.2	0	0	1	0	0	0	3.00	MEDIO	1.67	BAJO				
2.5.3		0	0	0	0	1	0	1.00	BAJO							
2.5.5		0	0	0	0	1	0	1.00	BAJO							

Anexo 12. Guía de análisis del software

Objetivo: Analizar las características del software con vistas a su integración al proceso de enseñanza-aprendizaje de la Inteligencia Artificial.

Software: _____

Aspectos a analizar:

- Funcionalidades del software que tributen a los objetivos del programa de la asignatura.
- Funcionalidades del software que tributen a los contenidos específicos del sistema de conocimiento declarado en el programa de la asignatura.
- Funcionalidades y características del software que permitan utilizar diversos métodos de enseñanza.
- Características del software que permitan integrarse a los diversos medios disponibles para el PEA.
- Accesibilidad al código fuente.
- Disponibilidad de documentación de ayuda para utilizar el software.
- Organización de la arquitectura del software.
- Posibilidades explícitas o implícitas para realizar tareas grupales.
- Posibilidades explícitas o implícitas para realizar tareas en equipo.
- Posibilidades explícitas o implícitas para realizar tareas individuales.
- Características del software que permitan su distribución.
- Características del software que garanticen su correcta utilización, según las condiciones tecnológicas existentes.
- Posibilidades explícitas o implícitas para realizar tareas de estudio del código fuente.
- Posibilidades explícitas o implícitas para realizar tareas de modificación del código fuente.
- Posibilidades explícitas o implícitas para realizar tareas de reutilización del código fuente.
- Relación general entre el software y la asignatura.
- Evidencias de reutilización de resultados de estudiantes que hayan ofrecido mejoras al software.

Anexo 13. Resultados del análisis al software

Dimensión	Subdimensión	Indicador	Respuestas						indicador		subdimensión		dimensión		variable	
			5	4	3	2	1	0	Índice	Nivel	Índice	Nivel	Índice	Nivel	Índice	Nivel
1	1.1	1.1.1	0	0	0	2	0	0	2.00	MEDIO	2.00	MEDIO	2.07	MEDIO	1.88	BAJO
		1.1.2	0	0	0	2	0	0	2.00	MEDIO						
		1.1.4	0	0	0	2	0	0	2.00	MEDIO						
		1.1.5	0	0	0	2	0	0	2.00	MEDIO						
	1.2	1.2.1	0	0	1	0	1	0	2.00	MEDIO	2.00	MEDIO				
		1.2.2	0	0	2	0	0	0	3.00	MEDIO						
		1.2.3	0	0	0	1	0	1	1.00	BAJO						
	1.3	1.3.1	0	0	0	1	1	0	1.50	BAJO	1.67	BAJO				
		1.3.2	0	0	0	1	1	0	1.50	BAJO						
		1.3.3	0	0	0	2	0	0	2.00	MEDIO						
	1.4	1.4.1	0	2	0	0	0	0	4.00	ALTO	4.00	ALTO				
		1.4.2	0	2	0	0	0	0	4.00	ALTO						
	1.5	1.5.1	0	0	0	2	0	0	2.00	MEDIO	1.33	BAJO				
		1.5.2	0	0	0	1	0	1	1.00	BAJO						
		1.5.3	0	0	0	1	0	1	1.00	BAJO						
2	2.2	2.2.1	0	0	0	0	2	0	1.00	BAJO	1.00	BAJO	0.50	BAJO		
	2.3	2.3.2	0	0	0	0	0	2	0.00	BAJO						

Anexo 14. Triangulación de métodos en el diagnóstico

Triangulación cuantitativa-cualitativa

Dimensión	Subdimensión	Indicador	I. 1	I. 2	I. 3	I. 4	I. 5	indicador		subdimensión		dimensión		variable	
								Índice	Nivel	Índice	Nivel	Índice	Nivel	Índice	Nivel
1	1.1	1.1.1	2.00	2.00		2.00	2.00	2.00	MEDIO	2.13	MEDIO	1.97	BAJO	1.66	BAJO
		1.1.2	1.83	1.55	1.30	2.00	2.00	1.74	BAJO						
		1.1.3	2.33	2.27	2.00	2.00		2.15	MEDIO						
		1.1.4	1.83	1.82			2.00	1.88	BAJO						
		1.1.5	2.67	2.09	2.28		2.00	2.26	MEDIO						
		1.1.6	3.00	3.00		3.00		3.00	MEDIO						
	1.2	1.2.1	1.83	2.00	1.04		2.00	1.72	BAJO	1.55	BAJO				
		1.2.2	2.50	2.36	1.21		3.00	2.27	MEDIO						
		1.2.3	0.67	0.73	0.28		1.00	0.67	BAJO						
	1.3	1.3.1	1.00	1.18		2.00	1.50	1.42	BAJO	1.46	BAJO				
		1.3.2	1.00	1.18		1.00	1.50	1.17	BAJO						
		1.3.3	1.83	1.36		2.00	2.00	1.80	BAJO						
	1.4	1.4.1	4.33	4.36	3.83		4.00	4.13	ALTO	4.11	ALTO				
		1.4.2	4.67	3.73	3.92		4.00	4.08	ALTO						
	1.5	1.5.1	1.67	1.55		2.00	2.00	1.80	BAJO	1.18	BAJO				
1.5.2		0.17	0.27		2.00	1.00	0.86	BAJO							
1.5.3		0.33	0.18		2.00	1.00	0.88	BAJO							
2	2.1	2.1.1	0.00	0.00	0.00	0.00		0.00	BAJO	0.12	BAJO				
		2.1.2	0.00	0.45	0.53	0.00		0.25	BAJO						
	2.2	2.2.1	1.17	1.09	1.00	1.00	1.00	1.05	BAJO	1.71	BAJO				
		2.2.2	2.83	3.00	2.34	2.00		2.54	MEDIO						
	2.3	2.3.1	0.00	0.36	0.21			0.19	BAJO	0.08	BAJO				
		2.3.2	0.00	0.00		0.00	0.00	0.00	BAJO						
	2.4	2.4.1	1.17	1.64		1.00		1.27	BAJO	1.74	BAJO				
		2.4.2	2.00	2.18		2.00		2.06	MEDIO						
		2.4.3		1.91	2.02			1.96	BAJO						
	2.5	2.5.1	0.67	0.64				0.65	BAJO	1.79	BAJO				
		2.5.2	3.00	3.00		3.00		3.00	MEDIO						
		2.5.3	2.50	2.64	2.79	1.00		2.23	MEDIO						
2.5.4		1.33	1.45				1.39	BAJO							
2.5.5		1.00	1.00		1.00		1.00	BAJO							

Triangulación cualitativa

Dimensión	Subdimensión	Indicador	I. 1	I. 2	I. 3	I. 4	I. 5	Nivel indicador	Sub-dimensión	dimensión
1	1.1	1.1.1	MEDIO	MEDIO		MEDIO	MEDIO	MEDIO	MEDIO	BAJO
		1.1.2	BAJO	BAJO	BAJO	MEDIO	MEDIO	BAJO		
		1.1.3	MEDIO	MEDIO	MEDIO	MEDIO		MEDIO		
		1.1.4	BAJO	BAJO			MEDIO	BAJO		
		1.1.5	MEDIO	MEDIO	MEDIO		MEDIO	MEDIO		
		1.1.6	MEDIO	MEDIO		MEDIO		MEDIO		
	1.2	1.2.1	BAJO	MEDIO	BAJO		MEDIO	BAJO	BAJO	
		1.2.2	MEDIO	MEDIO	BAJO		MEDIO	MEDIO		
		1.2.3	BAJO	BAJO	BAJO		BAJO	BAJO		
	1.3	1.3.1	BAJO	BAJO		MEDIO	BAJO	BAJO	BAJO	
		1.3.2	BAJO	BAJO		BAJO	BAJO	BAJO		
		1.3.3	BAJO	BAJO		MEDIO	MEDIO	BAJO		
	1.4	1.4.1	ALTO	ALTO	MEDIO		ALTO	ALTO	ALTO	
		1.4.2	ALTO	MEDIO	MEDIO			ALTO		
	1.5	1.5.1	BAJO	BAJO		MEDIO	MEDIO	BAJO	BAJO	
1.5.2		BAJO	BAJO		MEDIO	BAJO	BAJO			
1.5.3		BAJO	BAJO		MEDIO	BAJO	BAJO			
2	2.1	2.1.1	BAJO	BAJO	BAJO	BAJO		BAJO	BAJO	
		2.1.2	BAJO	BAJO	BAJO	BAJO		BAJO		
	2.2	2.2.1	BAJO	BAJO	BAJO	BAJO	BAJO	BAJO	BAJO	
		2.2.2	MEDIO	MEDIO	MEDIO	MEDIO		MEDIO		
	2.3	2.3.1	BAJO	BAJO	BAJO			BAJO	BAJO	
		2.3.2	BAJO	BAJO		BAJO	BAJO	BAJO		
	2.4	2.4.1	BAJO	BAJO		BAJO		BAJO	BAJO	
		2.4.2	MEDIO	MEDIO		MEDIO		MEDIO		
		2.4.3		BAJO	BAJO			BAJO		
	2.5	2.5.1	BAJO	BAJO				BAJO	BAJO	
		2.5.2	MEDIO	MEDIO		MEDIO		MEDIO		
2.5.3		MEDIO	MEDIO	MEDIO	BAJO		MEDIO			
2.5.4		BAJO	BAJO				BAJO			
2.5.5		BAJO	BAJO		BAJO		BAJO			

Anexo 15. Criterio de expertos por método Delphi

a) Cuestionario para evaluación de los expertos

Estimado(a): Como resultado de una investigación en el área de las Ciencias de la Educación se ha elaborado un modelo de integración de software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial. Es de nuestro interés someter a criterio de expertos la propuesta para obtener una valoración de sus principales componentes. Como primer paso requerimos de su autovaloración respecto a los niveles de conocimiento y experticia que posee en el tema. Agradecemos con anticipación su inestimable ayuda en tales propósitos.

1. Datos personales

Años de experiencia	1 <= X < 5	5 <= X < 10	10 <= X < 15	X >= 15
	—	—	—	—
Categoría docente	Instructor	Asistente	Profesor Auxiliar	Profesor Titular
	—	—	—	—
Formación académica	Universitario		Máster	Doctor en Ciencia
	—		—	—
Perfil de graduado	Ingeniería	Licenciatura	Pedagógico	Otro (especifique)
	—	—	—	—

2. Marque con una X, en una escala creciente del 1 al 10 el valor que considere se corresponde con el grado de conocimiento o información que posee sobre la utilización de software educativo u otras tecnologías de la informática y las comunicaciones en el proceso de enseñanza-aprendizaje.

1	2	3	4	5	6	7	8	9	10

3. ¿Qué fuentes ha utilizado para obtener conocimiento sobre el tema?

Fuente de argumentación	Grado de influencia		
	Alto	Medio	Bajo
1. Investigaciones teóricas o experimentales realizadas por usted			
2. Experiencia obtenida en la actividad profesional (Impartición de docencia utilizando software educativo u otra tecnología similar)			
3. Experiencia obtenida en la actividad profesional (Docencia de postgrado recibida sobre utilización de TIC)			
4. Análisis de literatura o publicaciones de autores nacionales			
5. Análisis de literatura o publicaciones de autores extranjeros			
6. Su intuición			

b) Clasificación de los expertos.

Expertos	Kc	Fuente de argumentación						ka	k	Clasificación
		1	2	3	4	5	6			
1	1.00	0.25	0.40	0.20	0.05	0.05	0.05	1.00	1.00	COMPETENCIA ALTA
2	0.90	0.25	0.40	0.20	0.05	0.05	0.05	1.00	0.95	COMPETENCIA ALTA
3	1.00	0.20	0.40	0.16	0.05	0.05	0.04	0.90	0.95	COMPETENCIA ALTA
4	1.00	0.25	0.40	0.20	0.05	0.04	0.05	0.99	1.00	COMPETENCIA ALTA
5	0.70	0.20	0.32	0.20	0.04	0.04	0.04	0.84	0.77	COMPETENCIA MEDIA
6	0.30	0.25	0.32	0.10	0.04	0.04	0.03	0.78	0.54	COMPETENCIA MEDIA
7	0.40	0.13	0.32	0.10	0.03	0.03	0.04	0.64	0.52	COMPETENCIA MEDIA
8	1.00	0.25	0.40	0.16	0.05	0.05	0.05	0.96	0.98	COMPETENCIA ALTA
9	0.80	0.25	0.32	0.16	0.05	0.05	0.04	0.87	0.84	COMPETENCIA ALTA
10	0.70	0.20	0.32	0.16	0.03	0.04	0.04	0.79	0.74	COMPETENCIA MEDIA
11	0.70	0.25	0.40	0.20	0.04	0.04	0.04	0.97	0.84	COMPETENCIA ALTA
12	0.80	0.20	0.40	0.20	0.04	0.04	0.04	0.92	0.86	COMPETENCIA ALTA
13	0.30	0.20	0.20	0.16	0.04	0.03	0.04	0.67	0.48	COMPETENCIA BAJA
14	0.40	0.25	0.32	0.20	0.05	0.04	0.04	0.90	0.65	COMPETENCIA MEDIA
15	0.70	0.20	0.32	0.16	0.05	0.04	0.04	0.81	0.76	COMPETENCIA MEDIA
16	0.80	0.25	0.40	0.20	0.04	0.04	0.05	0.98	0.89	COMPETENCIA ALTA
17	0.60	0.20	0.32	0.20	0.05	0.05	0.05	0.87	0.74	COMPETENCIA MEDIA
18	0.70	0.25	0.40	0.20	0.05	0.05	0.04	0.99	0.85	COMPETENCIA ALTA

c) Cuestionario de evaluación del modelo

Objetivo: Obtener criterio de expertos sobre el modelo de integración de software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial.

Estimado experto (a): _____

Con el objetivo de determinar la pertinencia del modelo de integración de software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial en la carrera Ingeniería en Ciencias Informáticas, se le solicita su consideración y evaluación de los criterios que se exponen. Coloque una x en las casillas de la derecha, que correspondan a la evaluación otorgada por usted. Para ello debe tener en cuenta la escala siguiente:

- 1- (MA) – Muy adecuada.
- 2- (BA) – Bastante adecuada.
- 3- (A) – Adecuada.
- 4- (PA) – Poco adecuada.
- 5- (I) – Inadecuada.

No.	Aspectos a valorar	MA	BA	A	PA	I
1	Pertinencia de los objetivos del modelo					
2	Pertinencia y adecuación de los fundamentos del modelo					
3	Pertinencia y adecuación de los principios del modelo					
4	Adecuación de los elementos del componente estructural y sus relaciones					
5	Coherencia entre los elementos del componente estructural y el componente conceptual					
6	Pertinencia y adecuación del componente instrumental para la implementación del modelo					
7	Coherencia en la representación gráfica del modelo					
8	Relevancia y pertinencia del modelo presentado					
Observaciones:						

d) Resultados de la primera ronda

Experto	Evaluación de cada aspecto							
	1	2	3	4	5	6	7	8
1	2	2	2	3	2	2	2	2
2	1	1	2	1	1	1	1	2
3	2	3	2	2	2	2	3	1
4	1	1	2	3	1	1	1	2
5	1	2	2	1	1	1	2	2
6	2	1	1	2	1	2	2	2
7	2	1	1	2	1	1	2	2
8	3	1	3	3	2	1	2	1
9	1	2	2	2	1	3	4	1
10	1	1	1	3	2	2	2	1
11	2	2	2	3	1	2	2	1
12	3	2	2	3	3	3	1	1
13	1	2	1	2	3	3	2	1
14	1	1	2	3	1	2	3	2
15	3	2	2	4	4	2	2	2
16	2	1	1	2	2	1	3	1
17	1	1	3	2	3	1	1	1
Suma rangos	29	26	31	41	31	30	35	25
Media elementos	1.71	1.53	1.82	2.41	1.82	1.76	2.06	1.47
Desviación elemento	0.77	0.62	0.64	0.80	0.95	0.75	0.83	0.51

e) Resultados de la segunda ronda

Experto	Evaluación de cada aspecto							
	1	2	3	4	5	6	7	8
1	1	2	1	2	2	1	2	1
2	1	1	1	1	1	1	1	1
3	1	1	1	1	2	2	1	1
4	1	2	1	1	1	1	1	1
5	1	1	1	1	1	1	2	1
6	1	2	1	1	1	1	1	1
7	1	2	1	1	1	1	1	1
8	2	1	2	2	1	1	1	1
9	1	1	2	2	1	2	1	1
10	1	1	1	1	1	2	1	1
11	2	2	1	1	1	1	2	1
12	2	2	1	1	2	1	1	1
13	1	1	1	1	1	2	2	1
14	1	1	1	2	1	1	1	1
15	2	2	2	2	2	2	1	1
16	1	1	1	1	2	1	1	1
17	1	1	2	2	1	1	1	1
Suma rangos	21	24	21	23	22	22	21	17
Media elementos	1.24	1.41	1.24	1.35	1.29	1.29	1.24	1.00
Desviación elemento	0.44	0.51	0.44	0.49	0.47	0.47	0.44	0.00

Anexo 16. Guion de contenido del software libre educativo SMEKDB

Software: Sistema multiagente para la extracción de conocimiento en bases de datos (SMEK-DB)

Asignatura: *Inteligencia Artificial II*

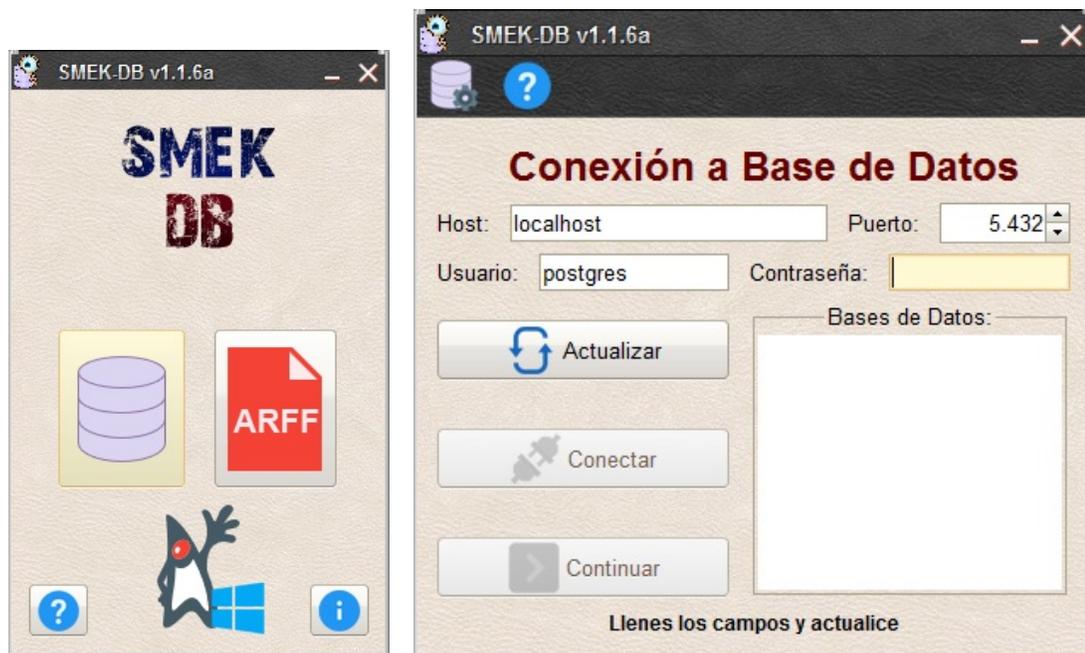
Dimensión	Aspecto a diseñar	Resultados
Didáctica	Objetivos	<ul style="list-style-type: none"> - Aplicar algoritmos de aprendizaje supervisado para resolver problemas de clasificación en minería de datos. - Aplicar algoritmos de aprendizaje no supervisado para resolver problemas de agrupamiento en minería de datos. - Utilizar el Weka como biblioteca de algoritmos para resolver problemas de minería de datos.
	Contenidos	<p>Algoritmos de clasificación:</p> <ul style="list-style-type: none"> - ID3 para la construcción de árboles de decisión. - K vecinos más cercanos. - Bayes ingenuo. - Optimización mínima secuencial (SMO) para el entrenamiento de máquinas de soporte vectorial. <p>Algoritmos de agrupamiento:</p> <ul style="list-style-type: none"> - Agrupamiento conceptual COBWEB. - Agrupamiento probabilístico por maximización de la expectativa EM. - Agrupamiento por media K-Means.
	Métodos y formas organizativas	<p>En la primera conferencia de la asignatura:</p> <ul style="list-style-type: none"> - Se realiza un recordatorio sobre los núcleos de conocimiento de la IA y los problemas que resuelve. - Se orienta el trabajo independiente de buscar un problema relacionado con el aprendizaje automático, específicamente con minería de datos y una base de datos relacionada con el tema a la cual se le aplicarán los algoritmos con el software. Los problemas serán recogidos al menos con una semana de antelación al inicio del tema en la asignatura. <p>En la primera conferencia del tema:</p> <ul style="list-style-type: none"> - Se expondrán y explicarán las características generales del SLE. Su interfaz, los algoritmos que incluye con una breve explicación de cada uno. Su estructura arquitectónica, con los agentes que lo forman y la función de cada uno. - Se asigna el trabajo independiente personalizado de reutilización del código fuente. - Se orienta el estudio de las indicaciones para la clase de laboratorio. <p>En la clase práctica siguiente:</p> <ul style="list-style-type: none"> - Se realizan actividades conjuntas de estudio del código fuente. Los parámetros que requiere alguno de los algoritmos de clasificación. <p>En la primera clase de laboratorio:</p> <ul style="list-style-type: none"> - Se resuelve por el profesor un ejemplo de utilización del software para aplicar un algoritmo a una base de datos. Puede utilizarse una de las reconocidas internacionalmente, como la Iris. - Se realiza un ejercicio de resolución conjunta de modificación del código fuente. Puede ser la configuración de uno de los algoritmos en el SLE. - Se orienta a partir de algún método para el trabajo en equipo una tarea de modificación del código fuente. Modificar la configuración del mismo algoritmo y configurar otros.

Dimensión	Aspecto a diseñar	Resultados
		<p>En las siguientes clases prácticas:</p> <ul style="list-style-type: none"> - Utilizar métodos de resolución conjunta para realizar tareas de modificación del código fuente mediante la configuración de los algoritmos vistos en las conferencias. <p>En los siguientes laboratorios:</p> <ul style="list-style-type: none"> - Utilizar métodos para el trabajo independiente y grupal en la resolución de tareas de modificación del código fuente. - Realizar aclaraciones de dudas mediante métodos de intercambio en el grupo sobre las tareas de reutilización del código asignada a cada estudiante.
	Medios	<p>En las aulas:</p> <ul style="list-style-type: none"> - Se utilizará la computadora y el televisor para mostrar el software. Su ejecución y partes del código fuente. - Se utilizará la pizarra y plumones para exponer y explicar las características del SLE. - Se utilizarán los medios disponibles de los estudiantes, laptop, tabletas o teléfonos para visualizar el código del software y resolver en las libretas las tareas que se le asignan. <p>En los laboratorios:</p> <ul style="list-style-type: none"> - Se utilizarán las computadoras para ejecutar y modificar el software. - Se utilizará la pizarra y plumones siempre que sea necesario para explicar tareas o aclarar dudas. - Se accederá al curso diseñado en la plataforma de enseñanza-aprendizaje para obtener los materiales necesarios y realizar tareas interactivas. <p>En el trabajo independiente:</p> <ul style="list-style-type: none"> - Se utilizarán los medios de los estudiantes para resolver las tareas. - Se accederá al curso diseñado en la plataforma de enseñanza-aprendizaje para acceder a los materiales necesarios y para interactuar con compañeros y el profesor mediante los recursos disponibles de chat, correo, wiki.
	Evaluación	<p>Evaluación frecuente:</p> <ul style="list-style-type: none"> - Se realizará en cada una de las tareas en conferencias, clases prácticas, clases de laboratorio y en el control del trabajo independiente. <p>Evaluación parcial:</p> <ul style="list-style-type: none"> - La exposición del resultado de la tarea de reutilización del código en el laboratorio se tomará como evaluación parcial de la asignatura para el tema de aprendizaje automático. <p>Evaluación final:</p> <ul style="list-style-type: none"> - Los estudiantes que por sus resultados en el trabajo con el software demuestren haber vencido los objetivos de la asignatura podrán ser eximidos del examen final. En el caso de los que obtengan tres o cuatro puntos se tomará con peso el trabajo con el software para subir su nota.
Espacial	Tareas para el trabajo en grupo	<ul style="list-style-type: none"> - Analizar las funcionalidades del SLE. - Analizar la estructura arquitectónica del SLE. - Aclaraciones de dudas generales sobre el SLE.
	Tareas para el trabajo en equipo	<ul style="list-style-type: none"> - Analizar las funciones de cada agente en el SLE. - Modificar los parámetros de los distintos algoritmos de clasificación. - Modificar los parámetros de los distintos algoritmos de agrupamiento.

Dimensión	Aspecto a diseñar	Resultados
	Tareas para el trabajo individual	<ul style="list-style-type: none"> - Aplicar algún algoritmo de clasificación a una base de datos. - Aplicar algún algoritmo de agrupamiento a una base de datos. - Comparar los resultados de aplicar varios algoritmos a una base de datos.
Práctica	Tareas para el estudio del código fuente	<ul style="list-style-type: none"> - Estudiar cómo el software libre educativo realiza la llamada a los métodos implementados en Weka. - Estudiar los parámetros de cada algoritmo de clasificación en Weka. - Estudiar los parámetros de cada algoritmo de agrupamiento en Weka. - Estudiar los criterios de comparación de los algoritmos de clasificación. - Estudiar los criterios de comparación de los algoritmos de agrupamiento.
	Tareas para la modificación del código fuente	<ul style="list-style-type: none"> - Modificar los parámetros de los algoritmos de clasificación y visualizar los resultados a través de la interfaz. - Modificar los parámetros de los algoritmos de agrupamiento y visualizar los resultados a través de la interfaz. - Aplicar un nuevo método de clasificación de los que incorpora Weka, con sus respectivas configuraciones, desde el código fuente. - Aplicar un nuevo método de agrupamiento de los que incorpora Weka, con sus respectivas configuraciones, desde el código fuente.
	Tareas para la reutilización del código fuente	<ul style="list-style-type: none"> - Personalizar la interfaz del SLE reutilizando los agentes para resolver una tarea específica de minería de datos. - Crear una aplicación nueva que reutilice métodos o agentes del SLE para resolver una tarea específica de minería de datos.
Gestión	Orientaciones para lograr la distribución del software	<ul style="list-style-type: none"> - La licencia del software será GPL que garantiza todas las libertades del software libre. - Se utilizará como lenguaje de programación Java, una tecnología multiplataforma que permite su ejecución en Linux y en Windows, los dos sistemas operativos utilizados institucionalmente. - Se incorporará al curso de la asignatura en la plataforma de enseñanza-aprendizaje Zera.
	Requerimientos para una correcta utilización del SLE	<ul style="list-style-type: none"> - El software deberá ejecutarse con al menos 1GB de memoria RAM, en microprocesador Intel Dual Core de 2.6 GHz o superior.
Tecnológica	Requisitos para la accesibilidad al código	<ul style="list-style-type: none"> - Se utilizará el lenguaje de programación Java, ya conocido por los estudiantes. - Se utilizará un estándar de codificación tradicional. - Se incorporarán comentarios al código.
	Disponibilidad de ayudas	<ul style="list-style-type: none"> - Se elabora la ayuda del software y se ofrecen los artefactos de la metodología utilizada. - Se brindará el presente guion de contenido como guía metodológica para la incorporación del software al PEA. - Se mantendrá disponible en la plataforma de enseñanza-aprendizaje Zera, la información necesaria para la incorporación del SLE.

Dimensión	Aspecto a diseñar	Resultados
	Organización arquitectónica	<p>- Se utilizará el paradigma de programación para agentes, específico de la IA.</p> <p>- Se diseñará un agente para cada funcionalidad general, lo cual facilita el acceso al código para su modificación y reutilización.</p> <p>Smek: Agente principal, encargado de la administración del SMA y de la interacción con el usuario.</p> <p>Connector: Se responsabiliza de la transformación de bases de datos de PostgreSQL en archivos ARFF.</p> <p>Arff: Realiza tareas de edición sobre archivos ARFF.</p> <p>Classifier: Ejecuta las operaciones de clasificación a través de su relación con la biblioteca Weka.</p> <p>Clusterer: Ejecuta las operaciones de agrupamiento a través de su relación con la biblioteca Weka.</p>

Anexo 17. Imágenes del software SMEKDB



Anexo 18. Diagnóstico aplicado a los estudiantes

Objetivo: Complementar la información del diagnóstico integral del estudiante con informaciones sobre sus intereses y conocimientos básicos sobre la asignatura.

Diagnóstico inicial a los estudiantes

0. Sobre usted

Nombre y apellidos: _____

Centro o grupo de investigación al que está vinculado para la Práctica Profesional:
_____.

Proyecto en el que trabaja: _____

1. Sobre sus gustos y conocimientos básicos relacionados con la asignatura.

- a) Diga las tres actividades que más le gusta realizar en su tiempo libre.

- b) Mencione tres palabras que tengan relación con la Inteligencia Artificial. Diga qué relación tienen.

- c) Diga las tres asignaturas que más le han gustado en la carrera ¿Por qué?

- d) ¿Le gustaría realizar tareas en la asignatura, relacionadas con su Práctica Profesional?

- e) En una escala ascendente del 1 al 10 ¿Qué nota se otorga en conocimientos de programación?

- f) En una escala ascendente del 1 al 10 ¿Qué nota le gustaría tener en conocimientos de programación?

- g) ¿Qué espera de la asignatura?

Anexo 19. Resultados según la sistematización de experiencias

Dimensión	Subdimensión	Indicador	indicador		subdimensión		dimensión		variable	
			Índice	Nivel	Índice	Nivel	Índice	Nivel	Índice	Nivel
1	1.1	1.1.1	4	ALTO	4.67	ALTO	4.53	ALTO	4.35	ALTO
		1.1.2	4	ALTO						
		1.1.3	5	ALTO						
		1.1.4	5	ALTO						
		1.1.5	5	ALTO						
		1.1.6	5	ALTO						
	1.2	1.2.1	4	ALTO	4.67	ALTO				
		1.2.2	5	ALTO						
		1.2.3	5	ALTO						
	1.3	1.3.1	4	ALTO	4.00	ALTO				
		1.3.2	4	ALTO						
		1.3.3	4	ALTO						
	1.4	1.4.1	5	ALTO	4.50	ALTO				
		1.4.2	4	ALTO						
	1.5	1.5.1	5	ALTO	4.67	ALTO				
1.5.2		5	ALTO							
1.5.3		4	ALTO							
2	2.1	2.1.1	4	ALTO	4.00	ALTO	4.14	ALTO		
		2.1.2	4	ALTO						
	2.2	2.2.1	4	ALTO	4.00	ALTO				
		2.2.2	4	ALTO						
	2.3	2.3.1	0	BAJO	2.50	MEDIO				
		2.3.2	5	ALTO						
	2.4	2.4.1	5	ALTO	4.67	ALTO				
		2.4.2	5	ALTO						
		2.4.3	4	ALTO						
	2.5	2.5.1	5	ALTO	4.60	ALTO				
		2.5.2	4	ALTO						
2.5.3		5	ALTO							
2.5.4		5	ALTO							
2.5.5		4	ALTO							

Anexo 20. Programa analítico del curso de postgrado IA-VIP

Programa Analítico de un Curso de Postgrado

Datos Generales	
Nombre:	Inteligencia Artificial con una visión práctica (IA-VIP)
Créditos:	2
Total de horas	96
Modalidad	Presencial

Fundamentación

La Inteligencia Artificial (IA) ha sido tratada desde varias posiciones teóricas. Algunos la defienden desde el punto de vista de desarrollar sistemas similares a los humanos, mientras que otros la asumen como sistemas con inteligencia o racionalidad. Algunos se han centrado en simular el pensamiento y otros en el comportamiento (Figura 1).

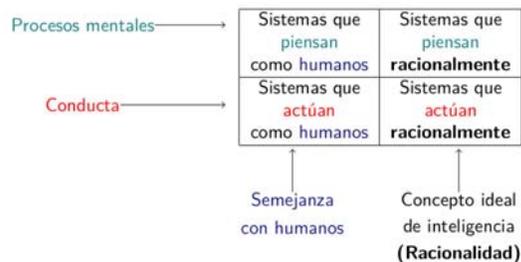


Figura 1. Posiciones

teóricas sobre la IA. Adaptado de (Russell y Norvig, 2010)

El desarrollo de la Inteligencia Artificial es innegable en el mundo. Para muchos es una disciplina compleja y que requiere de muchos conocimientos y tecnologías para desarrollarla. Para otros es solo teoría o ciencia ficción. Para Russell y Norving (2010) es la disciplina encargada de desarrollar sistemas que actúan racionalmente, de donde defienden la visión de la IA como el desarrollo de agentes. Esta forma de ver el desarrollo computacional derivó en la aceptación del paradigma de programación para agentes.

Estas ideas se defienden en el presente curso y se toma como base el libro "Artificial Intelligence. A Modern Approach" de estos autores. Además se ha enriquecido esta visión con la experiencia del profesor en el desarrollo de sistemas basados en el paradigma de agentes para desarrollar software libre educativo. Estos sistemas utilizan métodos, algoritmos y herramientas de IA para resolver problemas genéricos que puedan adaptarse a problemas reales.

Se asume en el curso a la Inteligencia Artificial como la disciplina de las Ciencias de la Computación encargada de aplicar métodos de representación, procesamiento y extracción del conocimiento, mediante programación multiparadigma, en la concepción de sistemas informáticos con comportamiento racional. De esta forma se pretende acercar la Inteligencia Artificial a problemas concretos. Se presentan sus tecnologías más recientes, núcleos de conocimiento y

aplicaciones más representativas, todo ello desde la solución de problemas prácticos que propongan los propios estudiantes del curso, o que se deriven de sus propios intereses personales o profesionales.

Objetivo General

Caracterizar la Inteligencia Artificial desde una visión práctica, utilizando software libre educativo para el desarrollo de sistemas con comportamiento racional.

Descripción de los Temas

Tema 1.

Objetivo: Caracterizar la Inteligencia Artificial como la disciplina encargada del desarrollo de sistemas con comportamiento racional.

Sistema de conocimientos:

- IA con una visión de agentes para la construcción de sistemas con comportamiento racional. Problemas que resuelve la IA.
- La enseñanza de la IA con software libre educativo.
- Paradigmas, metodologías, arquitecturas y herramientas actuales para el desarrollo de sistemas con comportamiento racional.

Tema 2.

Objetivo: Caracterizar los núcleos de conocimiento de la Inteligencia Artificial con una visión práctica.

Sistema de conocimiento:

- Representación del conocimiento. Reglas, casos, redes semánticas, ontologías.
- Razonamiento. Búsquedas a ciegas, heurísticas y metaheurísticas. Máquinas de inferencia.
- Tratamiento de la incertidumbre. Valores de certeza, lógica difusa, Bayes ingenuo.
- Aprendizaje automático. Supervisado, no supervisado y reforzado. Aprendizaje de árboles de decisión. Aprendizaje con redes neuronales, redes bayesianas y máquinas de soporte vectorial.

Tema 3.

Objetivo: Utilizar métodos de IA para resolver problemas profesionales, mediante la utilización de software libre educativo.

Sistema de conocimiento:

- Aplicaciones de la IA. Minería de datos, procesamiento de lenguaje natural, visión por computadoras y robótica.

Indicaciones Metodológicas y de Organización del curso

El curso se organiza en conferencias donde se exponen los elementos teóricos básicos y se ejemplifican. Se debe dejar la bibliografía y las orientaciones en cada clase para que los estudiantes cumplan las horas inducidas previstas.

Se propone una clase de laboratorio donde se presentan varios de los ejemplos prácticos de soluciones a problemas que requieren IA.

En la 2da actividad se realiza una clase teórico-práctica donde los estudiantes proponen problemas profesionales que pueden ser tratados con la IA y seleccionan uno para darle solución en el transcurso de la asignatura.

Las dos últimas actividades son para dar las propuestas de solución de cada uno a los problemas profesionales escogidos.

En cada clase se deben ir tratando las posibles soluciones a los problemas escogidos por los estudiantes.

Sistema de Evaluación del curso

La evaluación principal será la exposición de la solución al problema profesional escogido. El mismo puede incluir solución teórica fundamentada, modelación del problema mediante un método de IA específico o diseño e implementación de la solución o parte de ella utilizando software libre educativo.

La nota final debe tomar en cuenta, además de la tarea final, la asistencia y participación en clases.

Bibliografía

- Russell, Stuart and Norvig, Peter. Artificial Intelligence. A Modern Approach. 3th edition. Prentice Hall. 2010.
- Mitchell, T. M. Machine Learning, McGraw-Hill Science/Engineering/Math, 1997.
- Materiales elaborados por el profesor.

Elaborado por:	
Nombre:	Yuniesky Coca Bergolla
Cargo:	Metodólogo. Jefe de disciplina de Inteligencia Artificial
Fecha:	Febrero
Año de confección:	2018

Anexo 21. Técnica de satisfacción de ladov

a) Cuestionario aplicado a los estudiantes

Estimada(o) estudiante: Agradecemos que responda el siguiente cuestionario con toda franqueza y honestidad, el mismo responde al perfeccionamiento de la disciplina Inteligencia Artificial. Lea cuidadosamente cada pregunta antes de responder. El cuestionario es anónimo. Muchas gracias.

1. ¿Te gusta la Inteligencia Artificial?

Sí ___ No ___ No sé ___

2. ¿Cuáles son los tres (3) temas que más te gustan de IA?

(a) _____

(b) _____

(c) _____

3. ¿Estás satisfecho con el software y la forma en que se utilizó en la asignatura?

Sí ___ No ___ No sé ___

4. ¿Qué es lo que más te gusta de cómo se te enseña hoy la asignatura?

5. ¿Qué es lo que más te disgusta de cómo se te enseña hoy la asignatura?

6. ¿Tienes alguna preocupación con respecto a la asignatura?

Sí ___ No ___

(a) En caso afirmativo, lista tus preocupaciones:

7. ¿Cuáles son las tres (3) asignaturas que más te gustan de tu carrera?

(a) _____

(b) _____

(c) _____

8. Si en tu vida profesional te encuentras con una situación relacionada con la Inteligencia Artificial. ¿Recurrirías a estudiar o reutilizar el código del software utilizado en clases u otro similar?

Sí ___ No ___ No sé ___

Explique su respuesta:

9. ¿Qué aspectos le gustaron más del trabajo con el software en la asignatura?

(a) _____

(b) _____

(c) _____

10. ¿Te satisface la forma en que se te orientaron las tareas a realizar con el software, relacionadas con estudiar, modificar y reutilizar el código fuente?

(a) ___ Me gusta mucho

(b) ___ Me gusta más de lo que me disgusta

(c) ___ Me da lo mismo

(d) ___ Me disgusta más de lo que me gusta

(e) ___ No me gusta nada

(f) ___ No sé decir

b) Cuadro lógico de la técnica de satisfacción de ladov.

Pregunta 10. ¿Te satisface la forma en que se te orientaron las tareas a realizar con el software, relacionadas con estudiar, modificar y reutilizar el código fuente?	Pregunta 3. ¿Estás satisfecho con el software y la forma en que se utilizó en la asignatura?								
	Sí			No sé			No		
	Pregunta 8. Si en tu vida profesional te encuentras con una situación relacionada con la Inteligencia Artificial. ¿Recurrirías a estudiar o reutilizar el código del software utilizado en clases u otro similar?								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta	6	6	6	6	4	4	6	4	5
No puedo decir	2	3	6	3	3	3	6	3	4

Anexo 22. Técnica de grupos focales

a) Composición de los grupos focales.

Grupo	Nivel de dirección	Integrantes	Observación
<p>Metodológico. Departamento de Técnicas de Programación de la Facultad 4.</p>	Facultad	<p>Nueve profesores de las asignaturas de programación, sistemas digitales e inteligencia artificial.</p> <p>Tres son Máster, tres Profesores Auxiliares, tres Profesores Asistentes, dos Instructores y un Recién Graduado en Adiestramiento.</p>	Promedio de años de experiencia en la docencia de asignaturas de la especialidad: cinco
<p>Metodológico-específico. Colectivo de la disciplina Inteligencia Artificial</p>	Universidad	<p>10 profesores de la disciplina Inteligencia Artificial en la Universidad de las Ciencias Informáticas. Seis imparten docencia de las asignaturas lectivas de Inteligencia Artificial. Dos imparten la asignatura optativa Reconocimiento de Patrones y dos la asignatura optativa Redes Semánticas.</p> <p>Un Doctor en Ciencias de la Educación, dos son Máster. Cinco Profesores Asistentes, dos Profesores Auxiliares y tres Instructores.</p>	Promedio de años de experiencia en la docencia en IA: seis

b) Guía de temas utilizada en los talleres.

Tema 1: Valoración de los principios, componentes y relaciones del modelo de integración de SLE al PEA de la IA.

- 1.1 ¿Qué opina de los principios del modelo?
- 1.2 ¿Cómo cree que son y deben ser las relaciones de los componentes del modelo?
- 1.3 ¿Qué criterio le merece el vínculo entre el PEA de la IA y su entorno?
- 1.4 ¿Qué criterio le merece aprovechar las libertades del software libre como parte del proceso de enseñanza-aprendizaje?

Tema 2: Valoración de la dimensión instrumental para llevar a cabo la aplicación del modelo.

- 2.1 ¿Qué opina de la dimensión instrumental del modelo?
- 2.2 ¿Qué criterio le merece cada una de las etapas previstas?
- 2.3 ¿Qué añadiría o suprimiría de las acciones propuestas en cada etapa para favorecer la integración del software libre educativo al PEA de la IA?

Tema 3: Valoración del software libre educativo desarrollado para la asignatura IA-II.

- 3.1 ¿Considera usted que el software desarrollado cumple los requerimiento mínimos necesarios para integrarse al PEA de la IA?
- 3.2 ¿Qué añadiría o suprimiría del software para mejorar su integración al PEA de la IA?
- 3.3 ¿Cuánto cree que aportará la integración del software a la superación de los estudiantes y autosuperación de los profesores?

Tema 4: Coherencia entre el software libre, el modelo propuesto y las condiciones reales en la UCI, para lograr la integración.

- 4.1 ¿Considera que el modelo propuesto y el software desarrollado contienen los aspectos fundamentales a considerarse en la UCI?
- 4.2 ¿Cuáles son los elementos, a su consideración, que pudieran limitar la implementación generalizada del modelo de integración de software libre educativo al PEA de la IA en la carrera Ingeniería en Ciencias Informáticas?

c) Criterios operacionales para grupos focales.

Criterio operacional	Escala de valoración
Unanimidad de criterio	Se consideró cuando los dos grupos focales coincidieron en la respuesta, y las opiniones que se dieron dentro de cada grupo tuvieron consenso.
Mayoría de criterios	Fue considerada cuando más de la mitad de los miembros de los dos grupos coincidió en la respuesta.
Minoría de criterios	Cuando un solo grupo y menos de la mitad de los miembros por cada grupo coincidió con el mismo tipo de respuesta.

d) Resumen del informe final de los talleres con los grupos focales

Tema 1: Valoración de los principios, componentes y relaciones del modelo de integración de SLE al PEA de la IA.

1.1 ¿Qué opina de los principios del modelo?: Los grupos focales coincidieron en considerar como pertinentes los principios del modelo. Sustentados en los fundamentos presentados, abarcan los criterios básicos a tomar en cuenta para la integración de software libre educativo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial.

Criterio operacional: Unanimidad de criterios.

Propuestas: No hubo.

1.2 ¿Cómo cree que son y deben ser las relaciones de los componentes del modelo?: En los dos grupos varios de los integrantes abordaron el tema de las relaciones y la mayoría coincidió en profundizar en las relaciones entre los componentes del proceso de enseñanza-aprendizaje y la integración del software libre educativo.

Criterio operacional: Mayoría de criterio.

Propuestas: Redactar con mayor profundidad las relaciones entre los componentes del modelo.

1.3 ¿Qué criterio le merece el vínculo entre el PEA de la IA y su entorno?: Fue unánime el reconocimiento de la importancia de este vínculo y cómo se aborda en el modelo.

Criterio operacional: Unanimidad de criterios.

Propuestas: No hubo.

1.4 ¿Qué criterio le merece aprovechar las libertades del software libre como parte del proceso de enseñanza-aprendizaje?: Fue unánime el reconocimiento de la importancia de trabajar no solo la implementación, también el estudio y modificación del código como libertades fundamentales del movimiento de software libre. Se ofrecieron varias ideas de cómo llevarlo a la práctica. Dentro de las principales ideas se propuso incluir bibliotecas de algoritmos de minería de datos en otros lenguajes para que los estudiantes estudien la manera de utilizarlas, estudien y modifiquen los parámetros de los distintos algoritmos.

Criterio operacional: Unanimidad de criterios.

Propuestas: Desarrollar aplicaciones informáticas con estas características en otros lenguajes de programación y con otras bibliotecas diferentes.

Tema 2: Valoración de la dimensión instrumental para llevar a cabo la aplicación del modelo.

2.1 ¿Qué opina de la dimensión instrumental del modelo?: De forma general los dos grupos pidieron abordar mejor el proceso de diseño didáctico y desarrollo computacional del software libre educativo.

Criterio operacional: Mayoría de criterios.

Propuestas: Redactar con mayor profundidad el diseño didáctico y dar elementos sobre el proceso de desarrollo computacional del software libre educativo.

2.2 ¿Qué criterio le merece cada una de las etapas previstas?: Coincidieron en que las etapas son generales y se ajustan a las necesidades.

Criterio operacional: Unanimidad de criterios.

Propuestas: No hubo.

2.3 ¿Qué añadiría o suprimiría de las acciones propuestas en cada etapa para favorecer la integración del software libre educativo al PEA de la IA?: En los dos grupos la mayoría propuso incluir una acción específica para el seguimiento a las tareas, con el objetivo de profundizar en ese importante aspecto. El grupo dos plantea que el proceso de investigación para llevar a cabo el modelo es importante y por ende pudiera incluirse como una acción específica. Este planteamiento a pesar de ser minoría se valoró como importante por lo que se tomó en cuenta en el perfeccionamiento del modelo. En el grupo uno la mayoría propone tareas en las facultades en la acción de superación.

Criterio operacional: Mayoría de criterios.

Propuestas: Redactar de forma separada el seguimiento a las tareas a realizar con el software. Incluir como parte de las acciones de superación la investigación. Incluir en esta misma acción actividades en las facultades.

Tema 3: Valoración del software libre educativo desarrollado para la asignatura IA-II.

3.1 ¿Considera usted que el software desarrollado cumple con los requerimientos mínimos necesarios para integrarse al PEA de la IA?: Hubo total consenso en las ventajas de este software y sus potencialidades para ser utilizado en el PEA de la IA.

Criterio operacional: Unanimidad de criterios.

Propuestas: No hubo.

3.2 ¿Qué añadiría o suprimiría del software para mejorar su integración al PEA de la IA?: En los grupos se aportaron elementos de interés para perfeccionar el software. Hubo consenso en que cumple con los requerimiento básicos y que puede ser perfeccionado en el mismo proceso de integración y reutilización de curso a curso. Algunos de los aspectos sugeridos fue en la accesibilidad a las funcionalidades, de manera que sea más intuitiva y en la inclusión de otros gestores de bases de datos.

Criterio operacional: Mayoría de criterios.

Propuestas: Tomar en cuenta las propuestas realizadas para el perfeccionamiento del software.

3.3 ¿Cuánto cree que aportará la integración del software a la superación de los estudiantes y autosuperación de los profesores?: En todos los grupos se reconoce el aporte de la propuesta, tanto a la mejor preparación de los estudiantes, como a la superación técnica y pedagógica de los profesores.

Criterio operacional: Unanimidad de criterios.

Propuestas: No hubo.

Tema 4: Coherencia entre el software libre, el modelo propuesto y las condiciones reales en la UCI, para lograr la integración.

4.1 ¿Considera que el modelo propuesto y el software desarrollado contienen los aspectos fundamentales a considerarse en la UCI?: En los dos grupos se realizaron planteamientos sobre el vínculo del software con los centros de desarrollo, sobre todo los centros que abordan sistemas web, para los cuales el software no está desarrollado. Se valoró el criterio

sin embargo por ser la minoría de centros y planteamientos, se consideró oportuno mantener el software como producto de escritorio que puede ser utilizado para resolver problemas genéricos, incluso comunes a aplicaciones web.

Criterio operacional: Minoría de criterios.

Propuestas: Sugerir en próximas iteraciones del modelo tomar en cuenta el desarrollo de aplicaciones web.

4.2 ¿Cuáles son los elementos a su consideración que pudieran limitar la implementación generalizada del modelo de integración de software libre educativo al PEA de la IA?:

Fue unánime las preocupaciones con las condiciones tecnológicas. Se reconoció que hoy hay condiciones básicas pero pudiera afectarse en algún momento. Se abordó la posibilidad creciente de que los estudiantes utilicen sus propios medios en las clases. Se mostró preocupación por el decreciente interés y motivación de los estudiantes por la programación. Así como el limitado tiempo de los profesores para prepararse para este tipo de actividades.

Criterio operacional: Unanimidad de criterios.

Propuestas: Incluir indicaciones en los programas de las asignaturas para mitigar los riesgos presentados.

d) Resultado según los criterios operacionales

Tema	Pregunta	Criterio operacional
1	1.1 ¿Qué opina de los principios del modelo?	Unanimidad de criterios
	1.2 ¿Cómo cree que son y deben ser las relaciones de los componentes del modelo?	Mayoría de criterio
	1.3 ¿Qué criterio le merece el vínculo entre el PEA de la IA y su entorno?	Unanimidad de criterios
	1.4 ¿Qué criterio le merece aprovechar las libertades del software libre como parte del proceso de enseñanza-aprendizaje?	Unanimidad de criterios
2	2.1 ¿Qué opina de la dimensión instrumental del modelo?	Mayoría de criterios
	2.2 ¿Qué criterio le merece cada una de las etapas previstas?	Unanimidad de criterios
	2.3 ¿Qué añadiría o suprimiría de las acciones propuestas en cada etapa para favorecer la integración del software libre educativo al PEA de la IA?	Mayoría de criterios
3	3.1 ¿Considera usted que el software desarrollado cumple los requerimiento mínimos necesarios para integrarse al PEA de la IA?	Unanimidad de criterios
	3.2 ¿Qué añadiría o suprimiría del software para mejorar su integración al PEA de la IA?	Mayoría de criterios
	3.3 ¿Cuánto cree que aportará la integración del software a la superación de los estudiantes y autosuperación de los profesores?	Unanimidad de criterios
4	4.1 ¿Considera que el modelo propuesto y el software desarrollado contienen los aspectos fundamentales a considerarse en la UCI?	Minoría de criterios
	4.2 ¿Cuáles son los elementos a su consideración que pudieran limitar la implementación generalizada del modelo de integración de software libre educativo al PEA de la IA?	Unanimidad de criterios