

Universidad de las Ciencias Informáticas

Facultad 3

Departamento de Ingeniería de Software



Sistema para crear equipos integrados por estudiantes y profesionales partiendo de las líneas de investigación.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS
INFORMÁTICAS

Autor:

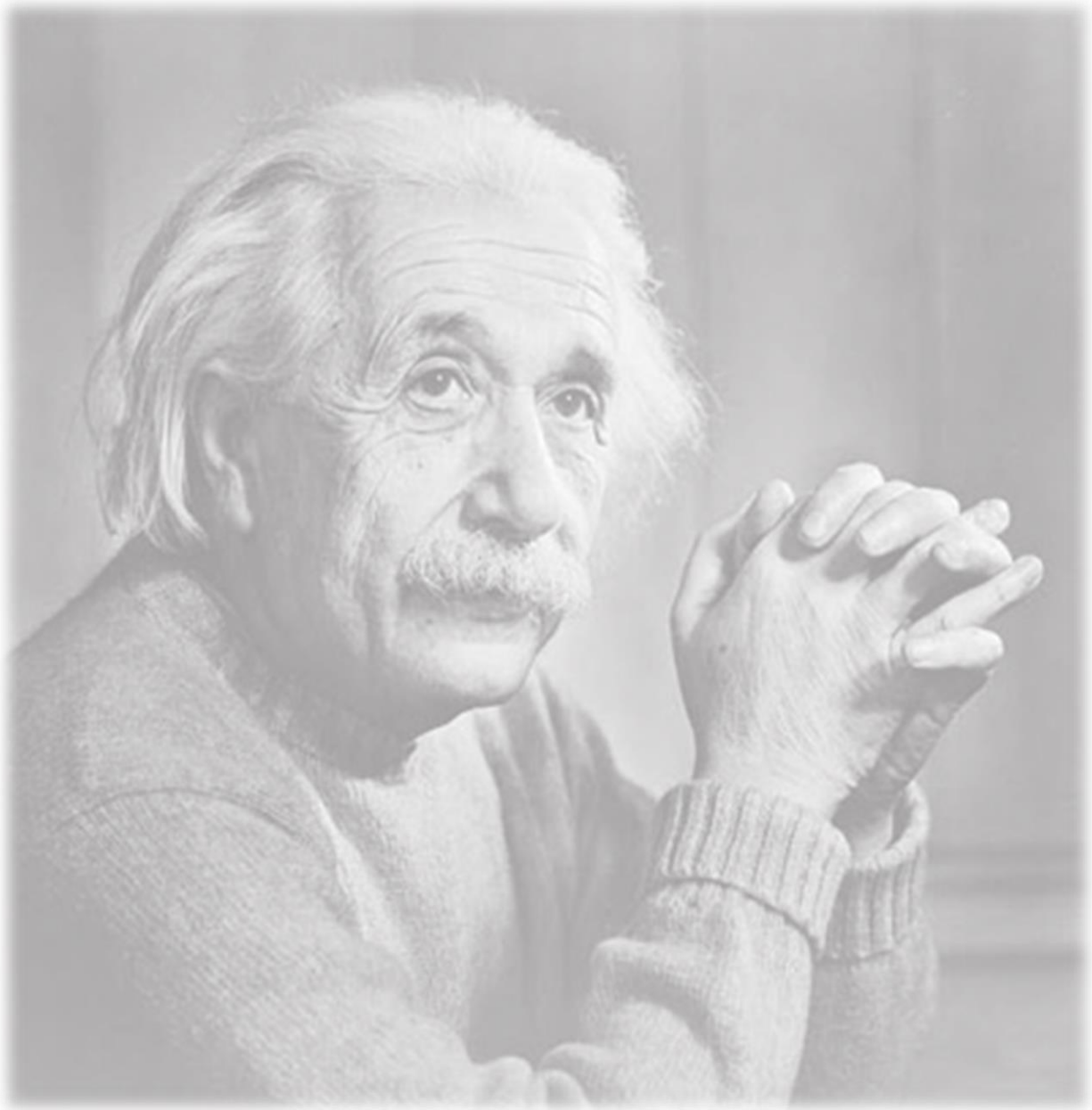
Ernesto Miró Peraza

Tutora:

Ing. Yadira Beatriz Reyes García

La Habana, 2016

“Año 58 de la Revolución”



Siento una enorme gratitud por todos los que me dijeron “No”, gracias a ellos, lo hice yo mismo.

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaramos ser autores del presente trabajo de diploma y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmamos la presente a los ___ días del mes de ___ del año ___.

Autor: Ernesto Miró Peraza

Tutor: ing. Yadira Beatriz Reyes García

RESUMEN

La Universidad de Ciencias Informáticas (UCI) se plantea como misión estratégica formar profesionales comprometidos y altamente calificados en la rama de la informática y producir softwares y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación. Precisamente en esta vinculación es donde entran los equipo o grupos de investigación los cuales se necesita que estén los mejor conformado posible para mejorar los resultados obtenidos en sus proyectos. Actualmente en la UCI no se cuenta con ninguna herramienta para tener en cuenta la relación existente entre estudiantes y las líneas de investigaciones para conformar los distintos equipos de desarrollo, atentando esto contra la calidad de los resultados.

El presente trabajo propone una solución informática para la informatización del proceso de conformación de equipos entre estudiantes y profesionales partiendo de las líneas de investigación. Esta herramienta posibilita tener en tablas la clasificación de los estudiantes y profesionales por las líneas de investigación y realizar búsquedas sobre las mismas, permite seleccionar a decisión del usuario los integrantes para conformar equipos a partir de esas relaciones, contiene gráficas para consultar los resultados de los procesamientos y presenta la opción de exportar los equipos conformados a un archivo con extensión xlsx. Con este sistema se resuelven ciertos problemas como el de tiempo empleado y calidad en la selección de personal para los grupos de investigación.

Índice

Introducción	1
Capítulo 1. Fundamentación Teórica	8
1.1 Estudio del estado del arte.....	8
1.2 Líneas de investigación, áreas del conocimiento y asignaturas	10
1.3 Sistemas basados en conocimientos	12
1.3.1 Rasgos que caracterizan a un sistema basado en conocimiento.	13
1.3.2 Desarrollo de un sistema basado en el conocimiento. Etapas.	14
1.3.3 Mecanismos de aprendizaje.	15
1.3.4 Tipos de sistemas basados en conocimiento.	17
1.3.5 Sistemas basados en reglas	18
1.4 Métodos para el Procesamiento de la Información.....	20
1.5 Metodología de desarrollo de software	20
1.6 Herramientas y tecnologías	23
1.6.1 Lenguaje de programación	23
1.6.2 Entorno de desarrollo integrado.....	24
1.6.3 Gestor de base de datos.....	24
1.6.4 Herramienta de modelado.....	24
1.7 Conclusiones del capítulo	25
Capítulo 2. Características y Diseño del sistema	27
2.1 Propuesta del sistema	27
2.2 Características del sistema	34
2.2.1 Requisitos no funcionales del sistema	34
2.2.2 Funcionalidades del sistema.....	35
2.3 Fase de exploración.....	36
2.3.1 Personal relacionado con el sistema.....	37

2.3.2	Historias de Usuarios.....	37
2.4	Fase de planificación	38
2.4.1	Estimación de esfuerzo por historia de usuario.....	38
2.4.2	Plan de iteraciones	39
2.4.3	Plan de entrega	41
2.5	Fase de diseño	42
2.5.1	Arquitectura	42
2.5.2	Patrones de diseño.....	43
2.5.3	Tarjetas Clase - Responsabilidad – Colaborador	45
2.5.4	Diseño de la base de datos.....	47
2.5.5	Tareas de la ingeniería	47
2.6	Conclusiones del capítulo	49
Capítulo 3. Implementación y Prueba		50
3.1	Fase de implementación.....	50
3.1.1	Estándares de codificación	50
3.1.2	Interfaces del sistema	51
3.2	Fase de prueba.....	57
3.2.1	Pruebas unitarias.....	57
3.2.2	Pruebas de Aceptación.....	58
3.2.3	Pruebas de algoritmos.....	61
3.3	Beneficios tangibles e intangibles	64
3.4	Conclusiones del capítulo	65
Conclusiones		66
Recomendaciones		67
Referencias Bibliográficas.....		68

Índice de Tablas

Tabla 1. Comparación entre herramientas.	9
Tabla 2. Líneas de investigación-Áreas del conocimiento.	10
Tabla 3. Áreas del conocimiento-Asignaturas.	11
Tabla 4. Asignaturas.	11
Tabla 5. Comparativa entre sistemas basados en el conocimiento.	17
Tabla 6. Temáticas. (Mizrahi, 2016).	29
Tabla 7. Personal relacionado con el sistema.	37
Tabla 8. Historia de Usuario 9.	37
Tabla 9. Estimación de esfuerzo por historia de usuario.	38
Tabla 10. Plan de duración de las iteraciones.	40
Tabla 11. Plan de Entrega.	41
Tabla 12. Tarjeta CRC MainWindow.	46
Tabla 13. Tarjeta CRC Ui_MainWindow.	46
Tabla 14. Tarea de Ingeniería 1.	48
Tabla 15. Tarea de Ingeniería 2.	48
Tabla 16. Caso de prueba de aceptación HU10_P1.	59
Tabla 17. Caso de prueba de aceptación HU22_P1.	59
Tabla 18. Caso de prueba de aceptación HU25_P1.	59
Tabla 19. Caso de estudio. Notas de estudiantes.	62
Tabla 20. Caso de estudio. Estudiantes. Resultados manuales.	62
Tabla 21. Caso de estudio. Estudiantes. Resultados del sistema.	62
Tabla 22. Caso de estudio. Profesionales.	63
Tabla 23. Caso de estudio. Profesionales. Resultados manuales.	63
Tabla 24. Caso de estudio. Profesionales. Resultados del sistema.	64
Tabla 25. Ahorro en tiempo.	65

Índice de Figuras

Figura 1. Ciclo para la actualización de conocimiento. Fuente (Ayala, 2006).	16
Figura 2. Modelo conceptual del sistema.	34
Figura 3. Arquitectura en capas para la solución.....	43
Figura 4. Arquitectura en capas para la solución.....	44
Figura 5. Ejemplo del patrón creador.	45
Figura 6. Diagrama Entidad Relación.....	47
Figura 7. Importar archivo. Paso 1.	51
Figura 8. Localizar archivos.	52
Figura 9. Seleccionar tipo de búsqueda.	52
Figura 10. Seleccionar una línea de investigación.	53
Figura 11. Seleccionar botón Mostrar.	53
Figura 12. Listas de personas.	54
Figura 13. Construir un equipo.....	54
Figura 14. Equipo en construcción.	55
Figura 15. Agregar equipo.....	55
Figura 16. Lista de equipos creados.	56
Figura 17. Exportar equipos.	56
Figura 18. Hoja de cálculo con los equipos.	57
Figura 19. Pruebas de iteración 1.	58
Figura 20. Pruebas de iteración 2.	58
Figura 21. Pruebas de iteración 3.	58
Figura 22. Aval de aceptación del cliente.....	61

Introducción

A través de la historia, el desarrollo de tareas con determinado grado de complejidad ha requerido para su ejecución, personas con conocimientos y cualidades específicas. El surgimiento de la Revolución Industrial acrecentó estos requerimientos, potenciados en la presente era por las nuevas tecnologías de la información y las comunicaciones.

Lo que pretende cualquier organización, y por ende cualquier proyecto, es conseguir una mayor eficiencia en su actividad y el objetivo que se plantea es, una vez conseguida, mejorarla. Esta intención de mejora se sustenta en dos elementos: el individuo y el equipo de trabajo. (Bustínduy, y otros, 2004)

En referencia al individuo, y sabiendo que no es sencillo obtener ni transmitir el conocimiento implícito en las personas, se manifiesta la necesidad de seleccionar a los mejores sujetos. En alusión al equipo, el propósito primordial que pretende la gestión es el alinear los intereses de las personas que componen el equipo con los objetivos del proyecto, con el fin de conseguir los mejores resultados y el mayor grado de satisfacción (Bustínduy, y otros, 2004).

La historia ha demostrado que el activo económico de más valor en una organización es el capital humano; son las personas las que determinan, en última instancia, su sostenimiento y éxito. Esto justifica la búsqueda de métodos y procedimientos para la adecuada selección de personal (SP), y lo convierte en uno de los principales procesos de cualquier organización, cuyo objetivo básico es escoger y clasificar los candidatos más adecuados a sus necesidades (Cárdenas, y otros, 2016).

Según (Cárdenas, y otros, 2016), la importancia de la SP se determina por:

- Personas adecuadas exigen menor capacitación.
- Menor tiempo de adaptación a la organización.
- Mayor productividad y eficiencia.
- Personas más satisfechas con su trabajo.
- Mayor permanencia en la organización.

La selección de personal es uno de los procesos más importantes para las organizaciones dado el entorno cambiante y competitivo de las mismas, demanda que cada cargo sea ocupado por

individuos con características especiales, indispensables para ejecutar adecuadamente y sobresalir en un cargo específico. (Vargas, 2011). Sin embargo, la SP no es una tarea trivial, más bien es compleja, por la cantidad de variables que deben ser evaluadas e interpretadas no exentas de errores humanos, para ello se han empleado innumerables métodos, desde los enfoques clásicos hasta los sistemas basados en Inteligencia Artificial. En este último campo han sido implementados sistemas para la minería de datos de personal para trabajar en empresas de alta tecnología, así como sistemas que filtran hojas de vida para seleccionar los candidatos más aptos para un cargo (Giarratano, y otros, 2001) (Kendall, 2001) (Rodríguez, y otros, 2006).

La SP es un problema que tiene varios campos de acción, donde actúan factores psicológicos, técnicos y de experiencia del candidato, su desempeño individual y en grupo. Es muy importante encontrar mecanismos que ayuden a las organizaciones a realizar dicho proceso con muy alta calidad, pero la complejidad de las organizaciones y los cargos hace que esto no sea fácil. El proceso se desarrolla en un alto nivel de incertidumbre y requiere de la experiencia de expertos para tomar decisiones en contextos muy variados, lo cual trae consigo lo que apunta (Gelabert, 2010), “La selección de personal es un campo aún no totalmente determinado, ni se tienen conocimientos sobre las verdades que se puedan aplicar sin temor al error”.

Por otra parte, hoy en día la capacidad de adquisición, almacenamiento y procesamiento de datos crece de forma exponencial, con tecnologías cada vez más sofisticadas. En esta abundancia de recursos informáticos hay grandes posibilidades de extraer información valiosa, pero para ello se requiere conocer los procesos y razonamientos humanos para dar “inteligencia” a los métodos computacionales que analizan los datos y la información para extraer conocimiento que apoye la toma de decisiones a nivel operativo y estratégico, de ahí el papel cada vez más relevante de la Inteligencia Artificial en la vida de los seres humanos (Vargas, 2011).

En Cuba, el tema de investigación abordado tiene una actualidad permanente, ya que las personas son la base fundamental en que se sustenta su sociedad pluralista e incluyente; y su recurso humano, altamente instruido, constituye el pilar de su desarrollo social y económico.

Por tanto, existe necesidad de buscar métodos más eficientes para la SP que guíen y den respuesta exitosa a sus retos presentes y futuros en todos los ámbitos de la economía, la investigación y la docencia. Dentro de ese espectro, los Centros de Educación Superior (CES) juegan un papel

importante, pues la universidad cubana ha dejado de ser un espacio puramente educativo-formativo para convertirse, además, en un centro de investigación y producción.

El VI Congreso del Partido Comunista de Cuba¹ movió la sociedad cubana hacia un reordenamiento e institucionalidad del país. El Ministerio de Educación Superior (MES) haciendo uso de las indicaciones establecidas en los Lineamientos de la Política Económica y Social del Partido y la Revolución, ha retomado con mayor fuerza su misión social y se esperan aportes al desarrollo de la sociedad.

La Universidad de Ciencias Informáticas (UCI) no ha estado al margen de todas estas transformaciones y para cumplir con su misión estratégica de formar profesionales comprometidos, altamente calificados en la rama de la informática y de producir software y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación, requiere de cambios profundos en su modelo actual, transitando hacia la integración de los procesos de formación, producción e investigación. En este último aspecto la dirección de investigaciones de la UCI ha estado trabajando en perfeccionar sus procesos reorganizando las líneas de investigación, los grupos asociados a ellas, la gestión de las publicaciones mediante el Sistema de Gestión Universitaria, dirigiendo sus esfuerzos a organizar las investigaciones en la universidad estimulando la participación de profesores y estudiantes.

Los profesionales de la UCI publican sus artículos en diferentes revistas, en memorias de eventos, asesoran las tesis de grado, tutoran estudiantes en su Práctica Profesional, y sin embargo cuando se va a caracterizar al profesional es difícil decidir en qué línea de investigación se encuentra trabajando o en qué áreas del conocimiento ha estudiado. Esta situación dificulta el proceso de asignación de

¹ El **Sexto Congreso del Partido Comunista de Cuba** se celebró en Ciudad de La Habana, entre los días 16 y 19 de abril de 2011. Su labor se concentró en la solución de los problemas de la economía y en las decisiones fundamentales de la actualización del modelo económico cubano y adoptó los Lineamientos de la Política Económica y Social del Partido y la Revolución.

temas de investigación a estudiantes que estén en el ciclo profesional, tanto para su Proyecto de Investigación y Desarrollo como para su trabajo de culminación de estudios, por parte de la selección del tutor o del tema para el estudiante. En este sentido la selección de un tema de investigación conociendo el desenvolvimiento del estudiante es engorroso, pues hay que consultar su recorrido para saber a qué tipo de investigación se puede asignar.

Este proceso transita por dificultades que entorpecen y produce demora en la investigación y por ende atenta contra el éxito del resultado científico.

Entre estas dificultades se han identificados las siguientes:

1. Los tutores escogen a los estudiantes de forma empírica o por afinidad.
2. Los estudiantes son asignados a temas de tesis según la ubicación física donde se encuentren.
3. No se aplica en muchos casos el desglose de las pirámides de investigación² para conformar los equipos.
4. No se tienen en cuenta las notas de las asignaturas de los estudiantes para asociarlos a una línea de investigación.

Por estas razones, la formación de equipos de investigación ha sido una de las constantes preocupaciones de la dirección, profesores y estudiantes de la UCI, debido a la dificultad que existe en formalizar adecuadamente las relaciones humanas. En todos estos procesos la subjetividad está presente como una característica permanente, lo que llevó para su solución, la aplicación de técnicas de Inteligencia Artificial, en especial la Minería de Datos, para el desarrollo de un modelo computacional, el cual confirmó las grandes posibilidades que brinda este campo.

² Las pirámides de investigación hacen alusión a la relación, mediante las tesis, de los distintos niveles intelectuales que pueda poseer un conjunto de personas. Un ejemplo de ello es la dependencia de la tesis de un doctor con las distintas tesis de másteres aspirantes al doctorado.

El análisis anterior conduce al **problema**: La no implementación del conocimiento sobre la relación existente entre las líneas de investigación y el desempeño académico de estudiantes, está afectando la formación de equipos de investigación.

Se identifica como **objeto de estudio**: sistemas basados en conocimiento.

La investigación se enmarca en el **campo de acción**: Sistemas basados en reglas para la selección de personal.

Se plantea como **objetivo general**: Desarrollar un sistema basado en reglas para la asignación de líneas de investigación que facilite la toma de decisiones en la creación de equipos conformados por estudiantes y profesionales.

Se plantean como **Objetivos Específicos**:

1. Establecer el marco teórico-conceptual para fundamentar el uso de sistemas basados en conocimiento para la selección de personal.
2. Realizar el modelo de la solución para la comprensión del proceso de conformación de equipos partiendo de las líneas de investigación.
3. Diseñar la solución de software para facilitar la implementación del sistema.
4. Implementar el diseño de la solución de software.
5. Validación de la solución de software mediante pruebas de Caja Blanca, Caja Negra³ y un caso de estudio.

Partiendo de lo antes expuesto, surge la siguiente **idea a defender**:

³ Las pruebas de **Caja Blanca** (también conocidas como pruebas de caja de cristal) es uno de los tipos de pruebas más importante que se aplican al software. Estas pruebas se realizan sobre las funciones internas de un módulo, en contraposición de las pruebas de **Caja Negra** que se aplican a los requisitos funcionales desde el exterior del módulo, en la integración de los subsistemas. Ambas pruebas se complementan y traen como resultado que disminuya en un gran por ciento el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad.

Si se tiene un sistema que utilice el conocimiento de la relación que existe entre las líneas de investigación, los estudiantes y los profesionales, se mejorará el proceso de formación de los equipos de investigación en cuanto al tiempo y calidad de los mismos.

Para el desarrollo satisfactorio del presente trabajo de diploma se emplearon métodos científicos tanto teóricos como empíricos buscando mantener un balance entre lo cuantitativo y lo cualitativo de la información obtenida a través de la investigación.

Métodos Teóricos

Para apoyar el desarrollo de la investigación se emplean los siguientes **métodos científicos**:

Análisis – Síntesis, permite, luego de la revisión de documentos y bibliografías y la aplicación de instrumentos de recolección de datos, realizar un análisis descomponiendo sus partes e integrando en una toda la información, y estructurar de forma sintética el contenido de la investigación. Resultará de especial interés el análisis de un grupo de investigaciones desarrolladas en los últimos años, que de alguna manera tratan aspectos relacionados con el objeto de estudio, para tener un mejor entendimiento acerca del tema.

Histórico – Lógico, es clave para identificar los antecedentes y comportamiento del proceso de conformación de equipos de investigación; revelando la esencia de la estructura del objeto de investigación propuesto.

Inductivo – Deductivo, permite hacer una propuesta de solución para las insuficiencias detectadas en el proceso de conformación de equipos de investigación y realizar una proyección del estado futuro deseado, definiendo indicadores concretos para su medición.

Modelación, la utilización de este método posibilitó la obtención de los principales artefactos de cada una de las fases de desarrollo del sistema.

Métodos Empíricos

Entrevista: Instrumento para recolectar opiniones sobre la relación de las asignaturas y las áreas del conocimiento identificadas en la universidad. Se realizaron un total de 20 entrevistas con las siguientes características: Los profesionales entrevistados fueron profesores de más de cinco años de experiencia en las asignaturas seleccionadas para el análisis, y especialistas de la producción del Centro de Informatización de Entidades (CEIGE), el Centro de Gobierno Electrónico (CEGEL) y el Centro de Soluciones Libres (CESOL), Especialista de Seguridad Informática que trabajan las áreas

temáticas de las facultades 1 y 3. Se realizó la triangulación de los datos y la triangulación de participantes a partir de los resultados de la entrevista con el objetivo de confrontar sus opiniones para ver las diferencias y semejanzas.

Medición: Se realiza con el objetivo de obtener información numérica acerca del proceso de asignación de tesis en la facultad 3. Además, se utiliza para evaluar el comportamiento del software diseñado y su pertinencia con los requisitos establecidos.

Este documento de tesis está estructurado en tres capítulos que se ajustan a las tareas de las etapas del proyecto y donde se describe todo el proceso de investigación. El **Capítulo 1** aborda el marco conceptual enmarcado en la tesis, donde se precisan los conceptos fundamentales que se trabajan en la investigación y que involucran la inteligencia artificial, los sistemas basados en conocimiento, la selección de personal, y las herramientas usadas para el desarrollo del proyecto. En el **Capítulo 2** se realiza una descripción general de la solución propuesta, se especifican los requisitos funcionales y no funcionales que se tendrán en cuenta para la implementación de la misma. Se detallan aspectos relacionados con su arquitectura y especifican los patrones del diseño aplicados y los artefactos derivados de la metodología de desarrollo de software seleccionada. Luego, en el **Capítulo 3** se describe la etapa de implementación, se elaboran y documentan las pruebas realizadas a la solución propuesta para demostrar el correcto funcionamiento de la misma, y por último se analizan los resultados obtenidos tras la aplicación de la herramienta en un caso de estudio. Se finaliza con las conclusiones, recomendaciones y referencias señaladas a lo largo del proyecto.

Capítulo 1. Fundamentación Teórica

El objetivo de este capítulo es estudiar e identificar los conceptos y elementos teóricos que permitan entender las características principales de un sistema basado en reglas para conformar equipos de investigación, siendo necesarios además para la realización y descripción del mismo. También se realiza una descripción de los parámetros a tener en cuenta para los estudiantes y profesionales. Por último, se describen las principales metodologías y herramientas utilizadas en la solución propuesta.

1.1 Estudio del estado del arte

Como parte de la investigación se realizó un estudio de los sistemas informáticos existentes para la conformación de equipos de investigación, tanto en el país como en el resto del mundo, que pudieran ser la solución al problema planteado. A continuación, se abordan los más significativos.

TalentForJobs

TalentForJobs es una empresa de desarrollo de Software avanzado. Su objetivo es proveer a los profesionales de RRHH de las tecnologías necesarias para que afronten los nuevos retos de reclutamiento impulsados por las nuevas tendencias de mercado y centralizar todos los procesos de selección de los clientes ofreciéndoles la posibilidad de aplicar varias y nuevas estrategias de reclutamiento y llevándoles así hacia un mayor control de sus campañas de selección y de su base de datos, por un lado, y por el otro, a un ahorro considerable en tiempo y costes. Se basa en modelos muy dinámicos lo que lo hace muy flexible frente a las cambiantes demandas de los clientes.

Ofrece una herramienta de trabajo no gratuita para todos los técnicos y responsables de selección y un espacio propio para cualquier integrante de cualquier departamento o cliente de una empresa, así como un espacio propio para cada candidato y además un servicio gratuito de asesoramiento tecnológico. (Talent4jobs, 2017)

Altamira Recruiting

Altamira Recruiting, de carácter privado, posibilita la búsqueda de texto completo y la generación de los campos en la base de datos que su organización requiere basándose en que es fundamental que un software soporte los procesos de la organización. Con este programa de reclutamiento y selección de personal es posible una personalización compleja: adicionar nuevos campos, definir listas adicionales, elegir el formato de los reportes, construir plantillas específicas de correo, diseñar el

portal de empleo perfecto, implementar los reportes y mucho más. La recepción de miles solicitudes por correo es una tarea agotadora, requiere tiempo y tiende a generar errores. El software de empleo Altamira Recruiting libera las bandejas de correo y permite organizar la tarea de reclutamiento de manera eficiente. Cuando se reciben cientos de candidaturas para una oferta de empleo es imprescindible lograr examinarlas rápidamente. Además, clasifica los candidatos a partir de un conjunto de criterios permitiendo que su organización se concentre en los mejores (Altamira , 2017).

Teamssoft

Teamssoft es una aplicación web que da soporte a la gestión de recursos humanos basado en un enfoque por competencias. Entre sus principales funcionalidades se encuentran: gestionar competencias, roles, proyectos y trabajadores teniendo en cuenta sus características psicológicas. El sistema incluye la asignación de personal a equipos de proyectos de software y la evaluación de las competencias técnicas y genéricas de los trabajadores. Además, brinda soporte a los procesos de capacitación y plan de carrera. (Ampuero, 2009)

Este sistema tiene incorporado un módulo de tratamiento estudiantil, el cual permite la gestión de estudiantes y la selección de estos para un proyecto. Además, da soporte a la toma de decisiones permitiendo seleccionar estudiantes según los criterios que determine el especialista de recursos humanos o jefe de proyecto. Permite dar seguimiento a los que pertenecen a un proyecto, teniendo en cuenta el fondo de tiempo planificado y real a dedicar al proyecto (Navarro, 2010).

Comparativa de sistemas

Como se muestra en la tabla 1 solo un software tiene en cuenta el tratamiento con los estudiantes y aun así el mismo no procesa su recorrido académico. Por otra parte, Altamira y TalentForJobs son softwares no gratuitos y además tienen una dependencia de internet para su uso. En general ninguno está asociado directamente a líneas investigativas por lo que no pueden ser la solución al problema planteado, pero algunas de sus características técnicas servirán como ejemplo para poder realizar un software que ejecute el proceso de selección de personal.

Tabla 1. Comparación entre herramientas.

Herramientas	Gestión de personal	Selección de personal	Evaluación de personal	Tratamiento de estudiantes	Libre	Internet
--------------	---------------------	-----------------------	------------------------	----------------------------	-------	----------

TalentForJobs	x	x	x			x
Altamira Recruiting	x	x	x			x
Teamsoft	x	x	x	x	x	

1.2 Líneas de investigación, áreas del conocimiento y asignaturas

Es importante resaltar que se pretende conformar equipos de investigación entre estudiantes y profesionales tomando como parámetro de similitud las líneas de investigación, las cuales forman parte de la estructura organizativa del sistema de ciencia, tecnología e innovación (CTI) de la UCI. Las líneas de investigación o líneas científicas son las direcciones del trabajo científico que la Universidad decide priorizar al máximo nivel posible, para obtener resultados científicos y la innovación tecnológicas (Dirección de Investigaciones de la UCI, 2017).

De los estudiantes, se conoce su recorrido académico del cual se deriva la relación con las áreas del conocimiento a las que pertenece el mismo y por tanto a la línea de investigación a la que está asociado. En cuanto a los profesionales, se conocen características como años de experiencia y tesis tuteladas siendo ellas las premisas para decidir a qué línea de investigación pertenece el individuo.

A continuación, se muestran las relaciones líneas de investigación-áreas del conocimiento y asignaturas-áreas del conocimiento heredadas ambas del desarrollo en curso de la tesis de maestría de la ingeniera Yadira Beatriz Reyes García y de (Dirección de Investigaciones de la UCI, 2017).

Tabla 2. Líneas de investigación-Áreas del conocimiento.

	Líneas de Investigación	Áreas del conocimiento
1	Visión por computadora (señales, realidad virtual)	1,2
2	Computación científica (computación de alto rendimiento, matemática física, bioinformática)	3,4,5
3	Inteligencia artificial y Reconocimiento de patrones	2,6,7
4	Redes y seguridad	2,8,9
5	Ingeniería de sistemas y gestión	7,10,11,12
6	Software libre y Código abierto (tecnologías abiertas)	13
7	Informática aplicada en la sociedad	7,15
8	Ciencias pedagógicas y humanísticas	14

Tabla 3. Áreas del conocimiento-Asignaturas.

	Área del conocimiento	Asignaturas
1	Realidad virtual y simulaciones (RVS)	1,2,5,6,7,9,16,18,19,24,26,28,30,31,36
2	Procesamiento de imágenes y señales (PDIS)	1,2,6,13,16,17, 24,31,36
3	Bioinformática (BIO)	1,2,6,7,16,19,24,30,31,34
4	Matemática y Física aplicada a la computación (MF)	1,2,5,6,7,8,10,16,18,30,31
5	Programación de avanzada o computación de alto rendimiento (CAR)	1,2,5,6,7,18,24,25,26,28,30,31
6	Inteligencia Artificial (IA)	1,2,5,6,7,16,18,19,24,25,29,30,31,32
7	Gestión de Proyectos Informáticos (GPI)	1,2,3,4,5,6,7,12,14,15,20,29,30,31,37
8	Redes de telecomunicaciones	1,2,5,6,13,17,23,30,35
9	Seguridad Informática (SI)	1,2,6,7,10,13,17,21,26
10	Inteligencia organizacional (ITO)	1,2,5,6,7,12,14,18,20,24,27,29,30,31,37
11	Tecnologías de base de datos(BD)	1,2,3,4,6,11,22,24,25,29
12	Ingeniería de Software(ISW)	1,2,3,4,5,6,7,15,30,
13	Software libre (SL)	1,2,6,9,10,33,34,36
14	Informática educativa (IED)	1,2,5,6
15	Informática Aplicada en la Sociedad	1,2,3,4,5,6,9,34

Tabla 4. Asignaturas.

Muestra de asignaturas para la investigación			
No	Lectivas	No	Optativas
1	Idioma Extranjero III	20	Comercio Electrónico
2	Idioma Extranjero IV	21	Informática Forense y Hacking Ético
3	Ingeniería de Software I	22	Optimización de Bases de Datos
4	Ingeniería de Software II	23	Protocolo TCP/IP y sus Aplicaciones
5	Investigación de Operaciones	24	Reconocimiento de Patrones
6	Metodología de la Investigación Científica	25	Almacenes de Datos
7	Probabilidades y Estadística	26	Hardware de Computadoras
8	Programación IV	27	Inteligencia Organizacional
9	Programación V	28	Lenguajes de Programación
10	Sistemas Operativos	29	Minería de Procesos(Optativa)

11	Sistemas de Bases de Datos II	30	Optimización
12	Subsistemas de Organizaciones	31	Redes Neuronales Artificiales
13	Teleinformática	32	Web Semántica
14	Componente Profesional de Ingeniería y Gestión de Software	33	Análisis y Diseño de Algoritmos
15	Gestión de Software	34	Arquitectura Orientada a Servicios
16	Inteligencia Artificial I	35	Configuración de Equipos de Interconexión de Redes
17	Redes y Seguridad Informática	36	Gráficos por Computadora
18	Simulación	37	Factibilidad de Proyectos de Inversión
19	Inteligencia Artificial II		

1.3 Sistemas basados en conocimientos

En la década del 70 se reconoció que los métodos de solución de problemas generales eran insuficientes para resolver los problemas orientados a aplicaciones. Se determinó que era necesario conocimiento específico sobre el problema, limitado a los dominios de aplicación de interés, en lugar de conocimiento general aplicable a muchos dominios lo que condujo al desarrollo de sistemas basados en el conocimiento (SBC) (Lio, 1998).

El conocimiento representado en los SBC es el de los expertos en el dominio. Una parte de un conocimiento experto consiste de relaciones de causa y efecto. Estas relaciones o reglas se originan a partir de la experiencia pasada y son llamadas heurísticas. Ellas representan conocimiento informal o atajos, que permiten a un experto encontrar rápidamente una solución a un problema sin tener que realizar un análisis detallado de una situación particular debido a un análisis de un problema similar resuelto de forma exitosa anteriormente o a relaciones que han sido aprendidas como resultado de un intento fallido pasado para resolver un problema similar. El experto puede no recordar todos los detalles del análisis del problema original, pero puede reconocer el enfoque aplicado a un problema similar (Lio, 1998).

El uso de heurísticas en los SBC es marcadamente diferente de las funciones heurísticas usadas en la búsqueda heurística. Las funciones heurísticas representan conocimiento general usado para guiar la búsqueda, mientras que la heurística en los SBC consiste en el conocimiento heurístico usado por un experto en la solución de un problema, esta es más específica del dominio y más profunda en naturaleza (Lio, 1998).

Un experto es una persona que posee habilidades que le permiten sacar conclusiones de experiencias pasadas y rápidamente focalizar sobre el centro de un problema dado. La mayor

posibilidad del éxito de un experto en la solución de problemas se debe a que ha adquirido un conjunto de relaciones de causa y efecto poderosas basadas en la experiencia. Un experto es capaz de utilizar este conocimiento básico para reconocer rápidamente rasgos sobresalientes del problema, clasificar este de acuerdo a estas características y buscar una solución. Se reconoce que en lugar de relaciones de causas y efecto los expertos recuerdan casos similares al problema a resolver y toman de ellos las soluciones dadas como punto de partida para encontrar la solución al nuevo problema (Lio, 1998).

Al experto lo caracterizan: una mayor amplitud en la variedad de problemas que puede resolver que los otros profesionales del dominio, más rapidez en encontrar una solución y más calidad en la solución encontrada. Vale resaltar que los expertos humanos generalmente poseen conocimiento privado (Lio, 1998).

En términos generales, un SBC puede ser definido como un sistema computarizado que usa conocimiento sobre un dominio para arribar a una solución de un problema de ese dominio. Esta solución es esencialmente la misma que la obtenida por una persona experimentada en el dominio del problema cuando se enfrenta al mismo problema (Lio, 1998).

1.3.1 Rasgos que caracterizan a un sistema basado en conocimiento.

Realmente, a los SBC lo caracterizan más rasgos que simplemente el hecho de duplicar el conocimiento y experiencia de un experto humano para un dominio específico. Tres conceptos fundamentales relativos a los SBC lo distinguen de los programas basados en búsqueda general (Lio, 1998):

1. La separación del conocimiento de cómo este es usado (distinción entre conocimiento y estrategia de control).
2. El uso de conocimiento muy específico del dominio.
3. Naturaleza heurística, en lugar de algorítmica, del conocimiento empleado.

Otros rasgos que lo distinguen de la programación convencional son que no requieren analizar completitud y pueden dar diversas soluciones (Lio, 1998).

Generalmente los términos SBC y Sistemas Expertos se usan indistintamente, aunque algunos autores limitan el uso del término SE a aplicaciones donde el conocimiento al nivel de experto es requerido (Lio, 1998).

Ventajas:

1. Amplia distribución de experticidad escasa.
2. Fácil modificación (conocimiento explícito y accesible).
3. Consistencia en las respuestas (los expertos humanos pueden diferir en sus explicaciones, incluso un mismo experto puede responder de forma diferente en momentos diferentes).
4. Gran accesibilidad (los SBC trabajan las 24 horas todos los días).
5. Preservación de la experticidad (constituye una memoria institucional y poseen la capacidad para adquirir nuevo conocimiento y perfeccionar el que poseen).
6. Solución de problemas que incluyen datos incompletos.
7. Explicación de soluciones (justifica sus conclusiones y explica por qué hace una pregunta).
8. Permite evaluar el efecto de nuevas estrategias añadiendo o modificando conocimiento.
9. Constituye un entrenador en el dominio de aplicación.

Desventajas:

1. Las respuestas no siempre son correctas.
2. Conocimiento limitado al dominio de experticidad.
3. Ausencia de sentido común.
4. No reconocen el límite de su conocimiento.

1.3.2 Desarrollo de un sistema basado en el conocimiento. Etapas.

En general, según (Ayala, 2006), el proceso de desarrollo de un sistema experto puede dividirse en las siguientes 5 etapas:

Identificación: Determinar las características del problema. El experto describe distintos casos y el ingeniero del conocimiento formula una descripción preliminar del problema. El experto en el dominio normalmente sugerirá cambios en la misma, suministrando ejemplos adicionales que aclaren los aspectos más significativos del problema. El ingeniero del conocimiento modifica su descripción preliminar y la presenta al experto en el dominio para que sugiera nuevos cambios, hasta que ambos llegan al convencimiento de que la descripción es la adecuada.

Conceptualización: Diseñar estructuras para organizar el conocimiento. En esta etapa el ingeniero del conocimiento crea un diagrama del problema en el que se representa gráficamente las relaciones existentes entre los procesos. Al igual que en la etapa de identificación, se produce un proceso iterativo entre el experto y el ingeniero del conocimiento hasta que ambos estén de acuerdo en los conceptos claves, y las relaciones existentes entre ellos están adecuadamente conceptualizados.

Formalización: En esta etapa el problema se conecta con las técnicas de desarrollo apropiadas. En este caso, se generan árboles de decisión detallados que representen adecuadamente el conocimiento del experto.

Implementación: Formulación de las reglas que incorporen el conocimiento. Se utilizan las herramientas y técnicas especificadas a fin de obtener un primer prototipo del sistema.

Prueba: Implementación de las reglas. El proceso de prueba se considera superado cuando las soluciones propuestas por el sistema sean tan válidas como las del experto humano.

1.3.3 Mecanismos de aprendizaje.

El mecanismo de aprendizaje es el sistema que va a ser capaz de adquirir nuevos conocimientos por medio de un proceso que le permita evaluar su propio aprendizaje. Integra diversas técnicas y algoritmos de aprendizaje que interactúan con el experto para obtener nuevos elementos de conocimiento, modificaciones al existente y sobre todo revisar y corregir la consistencia. (Ayala, 2006) Este módulo debe observar una conducta inteligente capaz de advertir elementos relevantes de conocimiento y desechar aquellos “inútiles” o poco importantes, además de resolver los conflictos generados por las contradicciones. (Ayala, 2006)

Una técnica muy efectiva para la adquisición del conocimiento, después de que la implementación ha comenzado, según (Ayala, 2006) consiste en:

Actualizar conocimiento:

Actualiza el conocimiento existente utilizando la técnica de simulación manual o los resultados de prueba.

Integración del conocimiento:

Mantiene la consistencia con la base de conocimientos actualmente disponible.

Implementación del conocimiento:

Incorpora el nuevo conocimiento con los ajustes correspondientes de la base.

Prueba:

Explora el nuevo conocimiento en conjunto con el adquirido anteriormente sometiéndolo a los casos de prueba que verifiquen su consistencia.

Comparación:

Comparar los resultados generados por el experto contra las conclusiones obtenidas por el sistema. El programa puede producir conclusiones erróneas o puede no correr hasta el final.

Cualquiera de estos casos indica una incompleta o incorrecta heurística o fallas en la implementación.

Donde los resultados difieran, encontrar las reglas o procedimientos que producen la discrepancia. Los expertos y el equipo de adquisición del conocimiento pueden examinar en el conocimiento documentado el conjunto de reglas y procedimientos relacionados a esas discrepancias y crear una simulación manual en ciertas partes del análisis para ver si las reglas y procedimientos están escritas en base al conocimiento experto. Para añadir este proceso, la implementación en computadora puede ser usada para producir una ruta del razonamiento del sistema o proveer de resultados intermedios.

Si la simulación manual del conocimiento documentado concuerda con el experto, buscar un error de implementación en el programa. Si no, regresa al primer paso y determina como el nuevo conocimiento modificó las reglas y procedimientos que causaron la discrepancia. Se pueden analizar los resultados obtenidos por el experto y el sistema a través de los pasos de la Ilustración 1.

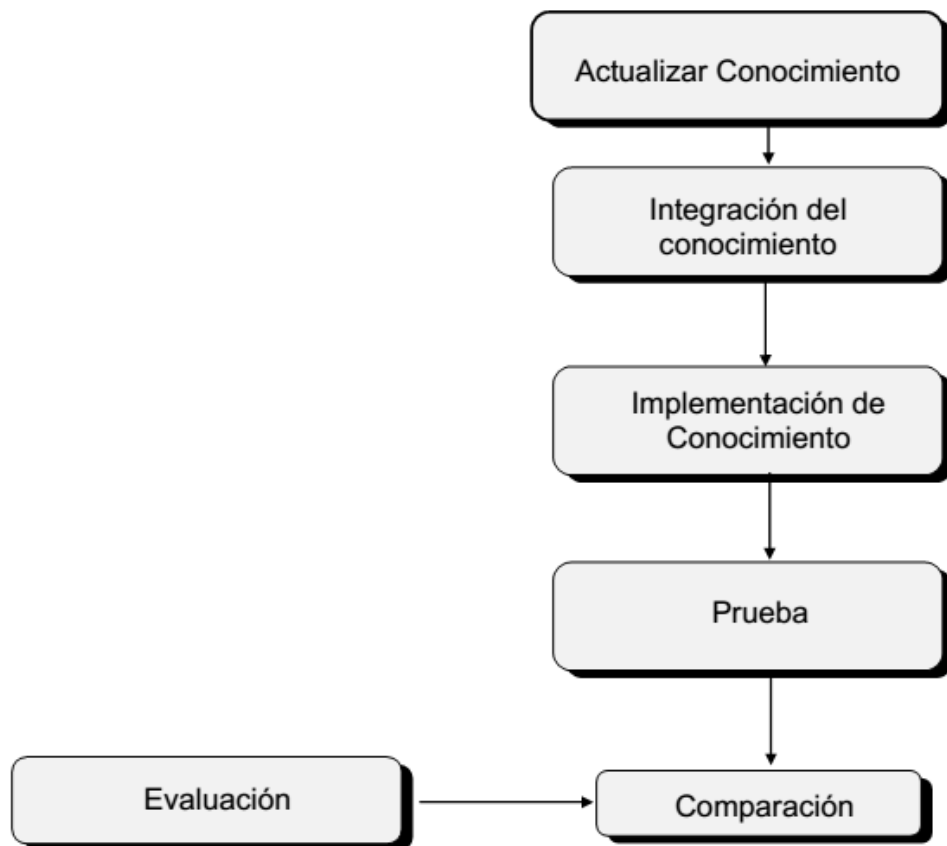


Figura 1. Ciclo para la actualización de conocimiento. Fuente (Ayala, 2006).

Evaluación:

Los expertos revisan los resultados arrojados por las pruebas y correcciones aplicadas al sistema, determinan la confiabilidad de sus respuestas y la eficiencia en su operación. Al final emiten un dictamen de aprobación o rechaza el contenido y manejo de la base de conocimientos.

Este proceso continúa en un ciclo para que el Sistema Experto y el conocimiento documentado estén en continua revisión y expansión.

1.3.4 Tipos de sistemas basados en conocimiento.

Los sistemas basados en conocimiento se pueden diferenciar por las formas representativas del conocimiento, el tipo de elementos que se usa para su procesamiento y su simplicidad.

En la siguiente tabla se analizaron seis tipos de sistemas basados en la bibliografía (Ayala, 2006).

Tabla 5. Comparativa entre sistemas basados en el conocimiento.

Sistemas	Formas representativas del conocimiento	Elementos que se usan para su procesamiento
Sistemas basados en reglas (SBR)	Reglas de producción.	Un caso y conocimiento en forma de reglas.
Sistemas basados en frames (SBF)	Frames	Un caso y frames u objetos y clases.
Sistemas basados en casos (SBCasos)	Casos	Un caso y conocimientos en una base de datos con casos del dominio.
Sistemas basados en probabilidades (SBP)	Probabilidades o frecuencias .	Un caso, Historia de los casos y probabilidades de ocurrencia.
Redes expertas (RE)	Pesos y alguna otra FRC.	Un caso con características muy complejas de analizar o existencia de ruido en ellas.
Sistemas basados en modelos (SBM)	Modelo del artefacto.	Un caso y datos para la modelación matemática.

Como se menciona más adelante en el epígrafe Métodos para el Procesamiento de la Información para clasificar a los estudiantes y profesionales, se cuenta con un conjunto de reglas programables heredadas de otras investigaciones, siendo estas los elementos para procesar la información.

Se llega a la conclusión de que el sistema basado en reglas será la mejor solución a implementar porque es el que emplea las reglas con las que se cuenta para procesar la información, es simple, es el que se acopla al tipo de problema dado y a la naturaleza de sus datos y, además, no se cuentan con los elementos necesarios como para utilizar otro tipo de sistema.

1.3.5 Sistemas basados en reglas

En nuestra vida diaria encontramos muchas situaciones complejas gobernadas por reglas deterministas: sistemas de control de tráfico, sistemas de seguridad, transacciones bancarias, etc. Los sistemas basados en reglas son una herramienta eficiente para tratar estos problemas. Las reglas deterministas constituyen la más sencilla de las metodologías utilizadas en sistemas expertos. La base de conocimiento contiene las variables y el conjunto de reglas que den en el problema, y el motor de inferencia obtiene las conclusiones aplicando la lógica clásica a estas reglas. Por regla se entiende una proposición lógica que relaciona dos o más objetos e incluye dos partes, la premisa y la conclusión. Cada una de estas partes consiste en una expresión lógica con una o más afirmaciones objeto-valor conectadas mediante los operadores lógicos y, o, o no. Una regla se escribe normalmente como "Si premisa, entonces conclusión". (GIL, 2015)

Como ejemplo de problema determinista que puede ser formulado usando un conjunto de reglas, considérese una situación en la que un usuario (por ejemplo, un cliente) desea sacar dinero de su cuenta corriente mediante un cajero automático (CA). En cuanto el usuario introduce la tarjeta en el CA, la máquina la lee y la verifica. Si la tarjeta no es verificada con éxito (por ejemplo, porque no es legible), el CA devuelve la tarjeta al usuario con el mensaje de error correspondiente. En otro caso, el CA pide al usuario su número de identificación personal (NIP). Si el número fuese incorrecto, se dan tres oportunidades de corregirlo. Si el NIP es correcto, el CA pregunta al usuario cuánto dinero desea sacar. Para que el pago se autorice, la cantidad solicitada no debe exceder de una cierta cantidad límite diaria, además de haber suficiente dinero en su cuenta (GIL, 2015).

Un Sistema Basado en Reglas (SBR) es un sistema basado en el conocimiento, en el cual se realiza una representación simbólica declarativa de un dominio mediante reglas de producción o reglas condicionales (GIL, 2015).

Componentes fundamentales de un SBR

1. Base de conocimiento:

Conformada por la base de datos y la base de reglas, que permiten la representación del conocimiento sobre el dominio de aplicación del sistema. En la base de hechos se representa el conocimiento en las variables de entrada y salida del sistema, que forman parte de las reglas semánticas almacenadas en la base de reglas. (Césari, 2012)

2. Base de hechos:

Representa el conocimiento del estado del sistema en un cierto instante. Usualmente se representa en una base de datos, y su información está directamente enlazada con la base de conocimientos. Un hecho puede provocar la conformación de una regla. (Césari, 2012)

3. Motor de inferencia:

Por cada entrada al sistema, el motor de inferencia utiliza el conocimiento almacenado en forma de hechos y reglas para generar una salida. Para la obtención de conclusiones, se utilizan diferentes mecanismos y estrategias de inferencia y control. Estas pueden ser simples o compuestas, estas últimas permiten estructurar el conocimiento en diferentes niveles y en este caso se da un encadenamiento de reglas que produce finalmente una conclusión. (Césari, 2012)

4. Interfaz de usuario:

Permite la comunicación entre el usuario y el Sistema Experto. El usuario puede consultar con el sistema a través de menús y gráficos, y este le responde con resultados. También es interesante mostrar la forma en que extraen las conclusiones a partir de los hechos. (Césari, 2012)

5. Módulo de adquisición del conocimiento

A través de este componente, se construye el sistema o se actualiza el conocimiento de la base de conocimientos en general. Adicionalmente, por medio de este módulo se pueden realizar actividades relacionadas con la configuración del sistema, específicamente del motor de inferencia, de acuerdo con las necesidades del usuario. (Césari, 2012)

Mecanismo de inferencia en SBR

Encadenamiento de reglas: es considerada una de las estrategias más usadas en la obtención de conclusiones compuestas que son las que resultan de la aplicación de más de una regla (Césari, 2012). Este tipo de inferencia parte de la observación de hechos en las variables de entrada para,

mediante el encadenamiento de reglas, alcanzar un hecho de salida deseado. Atendiendo al sentido del encadenamiento, se diferencian dos tipos de inferencia: hacia delante y hacia atrás. (Césari, 2012)

1. Encadenamiento hacia delante

Este tipo de inferencia parte de la observación de hechos en las variables de entrada mediante el encadenamiento de reglas (Font, 2008).

2. Encadenamiento hacia atrás:

El encadenamiento hacia atrás no tiene como objetivo inferir el valor de una variable, sino la demostración de una hipótesis. (Font, 2008)

1.4 Métodos para el Procesamiento de la Información

El sistema que se pretende implementar necesita recolectar información sobre los estudiantes y los profesionales de la UCI, procesarla, y así poder clasificar a estos dos por una línea de investigación para entonces conformar los equipos de investigación.

Datos de Estudiantes

Para el procesamiento de los datos de estudiantes se utilizará una función la cual es producto de la investigación de la tesis de maestría de la ingeniera Yadira Beatriz Reyes García. Este resultado está basado en reglas obtenidas de la aplicación de técnicas de inteligencia artificial en las cuales se usaron datos pertenecientes a los estudiantes de las facultades uno (1) y tres (3).

Datos de Profesionales

Para el tratamiento de los profesionales se utiliza un conjunto de reglas en forma de árbol de decisión ponderado extraído de (Mizrahi, 2016) donde se realizaron las fases del descubrimiento del conocimiento de los datos. La generación de este árbol fue a partir de la aplicación de técnicas de minería de datos e inteligencia artificial sobre una base de entrenamiento de profesionales. Las variables que se tuvieron en cuenta para el entrenamiento son Departamento Docente o Área, Años de Experiencia, Temática 1, Temática 2 y Línea de Investigación.

1.5 Metodología de desarrollo de software

El desarrollo de un software no es una tarea fácil, se debe contar con un proceso bien detallado, para esto se necesita aplicar una metodología que sea capaz de llevar a cabo el control total del producto. Las metodologías de desarrollo de software surgen ante la necesidad de utilizar una serie de

procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto de software. Dichas metodologías pretenden guiar a los desarrolladores, sin embargo, los requisitos de un software son muy variados y cambiantes, y se ha dado lugar a que exista una gran variedad de ellas (Letelier, 2006).

Las metodologías se dividen en dos grupos, tradicionales (pesadas) y ágiles (ligeras). Las tradicionales, se centran en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar, y requiere una extensa documentación, pretendiendo prever todo de antemano, además dependen de un equipo de desarrollo bastante grande. En las ágiles es más importante lograr que un producto de software se desarrolle con la calidad requerida en menor tiempo, que realizar una buena documentación en más tiempo. En este tipo de metodología el cliente está presente en todo momento y colabora con el proyecto, que posee un equipo de desarrollo pequeño (Letelier, 2006).

Programación extrema

Programación extrema (XP, por sus siglas en inglés) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. Además, se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (Joskowicz, 2008).

Características de la metodología XP (Beck, 2000):

- XP es una metodología “liviana” que no tiene en cuenta la utilización de elaborados casos de uso y la generación de una extensa documentación.
- XP tiene asociado un ciclo de vida y es considerado a su vez un proceso.
- La tendencia de entregar software en espacios de tiempo cada vez más pequeños con exigencias de costos reducidos y altos estándares de calidad.
- XP define Historias de Usuario (HU) como base del software a desarrollar, estas historias las escribe el cliente y describen escenarios sobre el funcionamiento del programa. A partir de las HU y de la arquitectura perseguida se crea un plan de liberaciones entre el equipo de desarrollo y el cliente.

Fases de la metodología XP:

- **Exploración:** Es la fase en la que se define el alcance general del proyecto. En esta fase, el cliente define lo que necesita mediante la redacción de sencillas “historias de usuarios”. Los programadores estiman los tiempos de desarrollo en base a esta información. Debe quedar claro que las estimaciones realizadas en esta fase son primarias (ya que estarán basadas en datos de muy alto nivel), y podrían variar cuando se analicen más en detalle en cada iteración. (Joskowicz, 2008)
- **Planificación:** La planificación es una fase corta, en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario, y, asociadas a éstas, las entregas. Típicamente esta fase consiste en una o varias reuniones grupales de planificación. El resultado de esta fase es un Plan de Entregas, como se detallará en la sección “Reglas y Practicas”. (Joskowicz, 2008)
- **Diseño:** Durante esta etapa se crea un diseño evolutivo que va mejorando incrementalmente y que permite hacer entregas pequeñas y frecuentes de valor para el cliente, basado principalmente en el desarrollo de las tarjetas Clase-Responsabilidad- Colaboración (CRC).
- **Desarrollo:** En esta fase se realiza la implementación de las HU que fueron seleccionadas por cada iteración. Al inicio se lleva a cabo un chequeo del plan de iteraciones por si es necesario realizar modificaciones. Como parte de este plan se crean tareas de ingeniería para ayudar a organizar la implementación exitosa de las HU.
- **Pruebas:** Esta fase permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones. XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñadas por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñada por el cliente final.

El ciclo de desarrollo consiste en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.

5. Vuelve al paso 1.

A partir del estudio de XP, se concluye que responde a las necesidades principales de tiempo, entorno y cantidad programadores, e incluye al cliente como parte fundamental del equipo de desarrollo. Además, se preocupa más en el avance exitoso del producto que en generar una documentación detallada del mismo, siendo capaz de adaptarse a los cambios de requisitos en cualquier punto del ciclo de vida del proyecto. Estos elementos demuestran que es una metodología factible para guiar el proceso de desarrollo de la solución, por lo que se decide incluir en la propuesta.

1.6 Herramientas y tecnologías

En todo proceso investigativo es necesario la utilización de sistemas de soporte que permitan organizar, facilitar, agilizar y automatizar las tareas generadas durante el transcurso de la investigación. Las herramientas, lenguajes y tecnologías empleadas que se describen a continuación son de vital importancia para una correcta realización de la misma.

1.6.1 Lenguaje de programación

Python: se trata de un lenguaje interpretado o de script, con tipado dinámico, multiplataforma y orientado a objetos, que permite la programación imperativa, funcional y orientada a aspectos (Duque, 2011).

Se seleccionó Python en su versión 3.5.0 porque es muy bueno para problemas docentes y educativos y además, su sintaxis es simple, clara y sencilla logrando de esta manera que los programas elaborados en este lenguaje parezcan pseudocódigo. Además, el tipado dinámico, el gestor de memoria, la gran cantidad de librerías disponibles y la potencia del lenguaje, entre otros, hacen que desarrollar una aplicación en Python sea sencillo y rápido.

PyQt: es un conjunto de enlaces Python para la biblioteca gráfica Qt. El módulo está desarrollado por la firma británica Riverbank Computing y se encuentra disponible para Windows, GNU/Linux y Mac OS bajo diferentes licencias. PyQt se distingue por su sencillez, por poseer un número importantes de herramientas que gestionen su manipulación y por su posibilidad de adecuarse a las distintas plataformas de software.

Utilizando PyQt en su versión 5.6 en el desarrollo de la herramienta informática, se puede crear una interfaz visual sencilla y sin muchos contratiempos, ya que PyQt posee los componentes visuales necesarios para su desarrollo, así como una abundante documentación y ejemplos.

1.6.1.1 Librerías adicionales

Para poder satisfacer todas las necesidades funcionales del sistema es necesario utilizar librerías como **Xlrd** la cual permite leer información de archivos Excel, **Xlsxwriter** que aporta las funcionalidades para crear y modificar archivos Excel, **Psycopg2** que al utilizarla es posible realizar conexiones con un servidor de base de datos postgresql y **Unittest** que es la herramienta para realizar pruebas de caja blanca durante las iteraciones de la metodología seleccionada.

1.6.2 Entorno de desarrollo integrado

Se selecciona **PyCharm** como entorno de desarrollo integrado, pues se utiliza para desarrollar en el lenguaje de programación Python. Proporciona un buen análisis del código y apoya el desarrollo web con Django. Es un IDE multi-plataforma. Entre las principales características que denotan esta herramienta una de ellas es su excelente codificación de asistencia y análisis, con la finalización de código, resaltado de sintaxis y de errores, así como el desarrollo de aplicaciones rápidas. Presenta una excelente navegación por el código, así como diferentes vistas especializadas del proyecto. Contiene una poderosa herramienta de refactorización del lenguaje Python, también integra herramientas para el desarrollo web, así como para pruebas unitarias entre otras herramientas. (Jetbrains, 2017)

1.6.3 Gestor de base de datos

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos (Momjian, 2001).

Para el desarrollo de este sistema se utilizó como gestor para la base de datos PostgreSQL v9.4 ya que:

- Soporta distintos tipos de datos.
- Incorpora una estructura de datos.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

1.6.4 Herramienta de modelado

Visual Paradigm: es una herramienta de diseño UML y herramienta CASE UML diseñada para ayudar al desarrollo de software. Ofrece un paquete completo necesario para la captura de requisitos, la planificación del software, la planificación de pruebas, el modelado de clases y el modelado de datos (Started, 2010).

Las principales características de la herramienta son:

- Soporta las últimas versiones del UML.
- Posee un poderoso generador de documentación y reportes en formato PDF, HTML y MS Word.
- Proporciona soporte para varios lenguajes en la generación de código e ingeniería inversa como: Java, C++, CORBA IDL, PHP, Ada y Python.
- Disponibilidad en múltiples plataformas (Windows, Linux)
- Capacidades de ingeniería directa e inversa.

Se selecciona Visual Paradigm for UML en su versión 8.0 como herramienta para el modelado UML, esta permite trabajar de forma colaborativa, hacer un trabajo organizado y ágil. Posibilita la realización de los diagramas necesarios para el desarrollo y mejor entendimiento de la aplicación. Permite realizar ingeniería inversa a partir del código fuente. Al ser seleccionado el lenguaje de modelado UML, es conveniente tener en cuenta su vinculación con Visual Paradigm, resaltando que este último presenta abundantes tutoriales de UML y demostraciones interactivas.

1.7 Conclusiones del capítulo

Con el desarrollo de este capítulo se obtuvo un mejor dimensionamiento del problema a partir del análisis de los principales conceptos asociados a la solución. Realizando un estudio del estado del arte, se encontraron soluciones con intensiones similares teniendo en cuenta otros factores totalmente diferentes a lo necesario, ratificando esto la ausencia de un software que dé solución al problema planteado. Dado que el sistema debe simular la tarea de un experto a la hora de asignar líneas de investigación a los individuos, se investigó todo lo referente a los sistemas expertos basados en conocimiento obteniendo como resultado que el sistema basado en reglas es el que mejor se acopla al escenario en cuestión. Para la implementación del software fue seleccionado un conjunto de herramientas y tecnologías basadas en licencias de software libre y por el mismo problema en cuestión, encaminado a obtener un producto de alta independencia tecnológica y utilizable en diferentes plataformas. Finalmente se escogió la metodología XP para guiar el proceso de desarrollo de la solución.

Capítulo 2. Características y Diseño del sistema

En este capítulo se presenta una propuesta de solución para realizar el proceso de asignación de líneas de investigación posibilitando la conformación de equipos mediante las mismas. Se especifican los requisitos funcionales y no funcionales del software y se obtienen los artefactos correspondientes a las fases de exploración, planificación y diseño de la metodología seleccionada. Además, se define la arquitectura y los principales patrones de diseño utilizados en el desarrollo de la solución.

2.1 Propuesta del sistema

Teniendo en cuenta los objetivos que persigue el presente trabajo de diploma, se propone desarrollar como solución un sistema para la toma de decisiones el cual permitirá escoger, entre estudiantes y profesionales, los miembros para un equipo de una línea de investigación determinada. Para este resultado es necesario obtener dos archivos preparados por el usuario final, uno para los estudiantes y uno para los profesionales.

Estudiantes

El archivo de los estudiantes debe contener las treinta y siete calificaciones de las asignaturas mostradas en el epígrafe 1.2 y presentar la siguiente estructura:

- El archivo debe llamarse Est_Asig.
- En cada tupla estará almacenada la información de las notas, el nombre con los apellidos y el carnet de identidad del estudiante correspondiente.
- Cada nota estará comprendida en el rango de 2-5 puntos o “-“ en caso de no haber cursado esa asignatura.
- Debe contener 37 asignaturas, específicamente las mostradas en la **¡Error! No se encuentra el origen de la referencia.**

Se muestra a continuación el pseudocódigo para clasificar a los estudiantes:

Clasificar por líneas al estudiante.

ENTRADA: notas, asignaturas_por_area, áreas_por_linea, líneas, areas.

SALIDA: líneas_a_las_que_pertenece.

VARIABLES: i,j,w,z,r,u,t: enteras; suma_notas: entra; promedio: real; cantiad_asignaturas: entero;

INICIO

Para i=1 hasta longitud de líneas:

 cantidad_asignaturas=0

 suma_notas=0

```
Para j=1 hasta longitud de áreas_por_linea
  Si líneas en i es la línea de áreas_por_linea en j
    Para w=1 hasta longitud de áreas_por_linea en j
      Para z=1 hasta longitud de areas
        Si areas en z es áreas_por_linea en (j,w)
          Para r=1 hasta longitud de asignaturas_por_area
            Si áreas en z es el área de asignaturas_por_area en r
              Para u=1 hasta longitud de asignaturas_por_area en r
                Para t=1 hasta longitud de notas
                  Si la asignatura de notas en t es la asignatura de asignaturas _por_area en
                    (r,u)
                      suma_notas= Suma_notas + nota de notas en t
                      cantidad_asignaturas= cantidad_asignaturas + 1
                Fin Para
              Fin Para
            Fin Para
          Fin Para
        Fin Para
      Fin Para
    Fin Para
  Fin Para
  promedio = suma_notas/cantidad_asignaturas
  Si promedio >=3.5
    Agregar línea a líneas_a_las_que_pertenece
  Fin Para
  Si líneas_a_las_que_pertenece esta vacia
    Agregar a líneas_a_las_que_pertenece la línea 7
FIN
```

Profesionales

El archivo de los profesionales debe llamarse Prof_Info para identificarlo y contener por cada tupla la información correspondiente referente a nombre y apellidos, departamento docente o área (Dpto Doc o Área), si es máster o doctor (MSc o Dr), años de experiencia(AE), temática 1(T1), temática 2 (T2) y el número de Solapín.

Información permitida por cada variable:

- Nombre y apellidos: nombre y apellidos del profesional.
- Dpto Doc, o Área: el nombre del lugar al que pertenezca el individuo.
- (MSc o Dr): máster o doctor.
- (AE): la cantidad de años que ha empleado en ejercer su profesión.
- Solapín: el número de solapín del individuo.
- Para los parámetros T1, T2: nombre de solo dos temáticas abordadas en el transcurso de su profesión, uno por cada T. Según (Dirección de Investigaciones de la UCI, 2017) las temáticas pueden ser:

Tabla 6. Temáticas. (Mizrahi, 2016).

Temáticas			
1	Informática Educativa	13	Minería de Proceso
2	Informática Jurídica	14	Seguridad Informática
3	Explotación de Minas, Metalúrgica	15	Impacto Social de las TIC
4	Matemática y Física Aplicada a la Computación	16	Inteligencia Organizacional
5	Informática Médica	17	Tecnologías de Base de Datos
6	Ciencias Empresariales	18	Software Libre
7	Gestión de Proyectos	19	Formación del Ingeniero Informático
8	Web Semántica	20	Bioinformática
9	Inteligencia Artificial	21	Sistemas de Código Abierto
10	Procesamiento de Imágenes y Señales	22	Programación Avanzada
11	Ingeniería de Software	23	Realidad Virtual y Simulaciones
12	Redes de Telecomunicaciones	24	Reconocimiento de Patrones

A continuación, se muestra el árbol de decisión ponderado mencionado en el epígrafe Métodos para el Procesamiento de la Información para clasificar a los profesionales y las instrucciones para interpretarlo:

Instrucciones

“Temática 1”: nodo raíz. Por ejemplo:

-Si “Temática 1” = Realidad Virtual y Simulaciones entonces el individuo pertenece a la línea de investigación 1.

-Si “Temática 1” = Informática Educativa entonces el individuo pertenece a la línea de investigación 8.

“Temática 2”, “AE”, “Dpto Doc o Área” y “MSc o Dr” son subnodos. Los “|” en el árbol representan el nivel donde se está posicionado. Por ejemplo:

-Si “Temática 1” = Ciencias Empresariales y “AE” > 9 entonces el individuo pertenece a la línea de investigación 5.

-Si “Temática 1” = Ciencias Empresariales y “AE” <= 9 entonces el individuo pertenece a la línea de investigación 7.

Árbol

Temática 1 = Realidad Virtual y Simulaciones: 1 (4.0)

Temática 1 = Informática Educativa: 8 (4.0)

Temática 1 = Informática Jurídica: 7 (4.0)

Temática 1 = Explotación de minas, metalúrgica: 2 (1.0)

Temática 1 = Matemática y Física aplicada a la computación: 2 (7.0/1.0)

Temática 1 = Informática Médica: 7 (1.0)

Temática 1 = Ciencias Empresariales

| AE <= 9: 7 (4.0/1.0)

| AE > 9: 5 (2.0)

Temática 1 = Gestión de Proyectos

| Temática 2 = Matemática y Física aplicada a la computación: 5 (0.0)

| Temática 2 = Web Semántica: 5 (0.0)

| Temática 2 = Gestión de Proyectos: 5 (0.0)

| Temática 2 = Informática Educativa: 8 (6.0/2.0)

- | Temática 2 = Inteligencia Artificial
- | | Dpto Doc o Área = CB: 3 (0.0)
- | | Dpto Doc o Área = CEIGE: 7 (1.0)
- | | Dpto Doc o Área = TP: 5 (2.0/1.0)
- | | Dpto Doc o Área = CEGEL: 3 (0.0)
- | | Dpto Doc o Área = CS: 3 (0.0)
- | | Dpto Doc o Área = ISW: 3 (2.0)
- | Temática 2 = Procesamiento de imágenes y señales: 5 (0.0)
- | Temática 2 = Ingeniería de Software: 5 (9.0)
- | Temática 2 = Redes de telecomunicaciones: 5 (0.0)
- | Temática 2 = Reconocimiento de patrones: 5 (0.0)
- | Temática 2 = Minería de Proceso: 5 (0.0)
- | Temática 2 = Impacto social de las TIC: 5 (2.0/1.0)
- | Temática 2 = Inteligencia organizacional: 5 (1.0)
- | Temática 2 = Tecnologías de base de datos: 5 (0.0)
- | Temática 2 = Informática Jurídica: 5 (1.0)
- | Temática 2 = Ciencias Empresariales: 7 (1.0)
- | Temática 2 = Software libre: 5 (1.0)
- | Temática 2 = Formación del ingeniero informático: 5 (0.0)
- | Temática 2 = Seguridad Informática: 5 (1.0)
- Temática 1 = Inteligencia Artificial
- | Temática 2 = Matemática y Física aplicada a la computación: 3 (0.0)
- | Temática 2 = Web Semántica: 3 (0.0)
- | Temática 2 = Gestión de Proyectos: 3 (6.43/1.0)
- | Temática 2 = Informática Educativa: 3 (2.14)

- | Temática 2 = Inteligencia Artificial: 3 (1.07)
- | Temática 2 = Procesamiento de imágenes y señales: 3 (1.07)
- | Temática 2 = Ingeniería de Software: 5 (1.07/0.07)
- | Temática 2 = Redes de telecomunicaciones: 3 (0.0)
- | Temática 2 = Reconocimiento de patrones: 3 (0.0)
- | Temática 2 = Minería de Proceso: 3 (0.0)
- | Temática 2 = Impacto social de las TIC: 3 (0.0)
- | Temática 2 = Inteligencia organizacional: 3 (0.0)
- | Temática 2 = Tecnologías de base de datos: 3 (0.0)
- | Temática 2 = Informática Jurídica: 3 (0.0)
- | Temática 2 = Ciencias Empresariales: 3 (0.0)
- | Temática 2 = Software libre: 6 (2.14/0.14)
- | Temática 2 = Formación del ingeniero informático: 3 (0.0)
- | Temática 2 = Seguridad Informática: 3 (0.0)
- Temática 1 = Web semántica: 3 (3.0)
- Temática 1 = Procesamiento de imágenes y señales: 3 (5.0/1.0)
- Temática 1 = Reconocimiento de patrones: 3 (2.0)
- Temática 1 = Bioinformática: 3 (1.0)
- Temática 1 = Informática jurídica: 7 (12.0)
- Temática 1 = Formación del ingeniero informático: 8 (31.0)
- Temática 1 = Ingeniería de Software: 5 (4.0/2.0)
- Temática 1 = Tecnologías de base de datos
- | MSc o Dr = Sin_Categoría: 3 (3.0)
- | MSc o Dr= Doctor: 3 (0.0)
- | MSc o Dr = Máster: 5 (2.0)

Temática 1 = Seguridad Informática: 7 (1.0)

Temática 1 = Sistemas de código abierto: 6 (2.0)

Temática 1 = Programación Avanzada: 5 (1.0)

Temática 1 = Redes de telecomunicaciones: 4 (3.0/1.0)

Temática 1 = Software libre: 6 (1.0)

Descripción del sistema propuesto

El software, para poder procesar información, deberá importar los archivos de estudiantes y de profesionales a petición del usuario. El sistema debe ofrecer dos tipos de búsquedas, la primera consiste en seleccionar una línea de investigación para mostrar una lista de estudiantes y una de profesionales, ambas asociadas a ella, la segunda, es para partir de un estudiante y una de sus líneas de investigación y consecuentemente mostrar las listas de los estudiantes y profesionales asociados a la línea. El usuario podrá seleccionar uno a uno los individuos de las listas y moverlos al área del equipo en construcción hasta que esté listo para ser agregado a la lista de equipos conformados. El sistema posibilitará al usuario modificar o eliminar de la lista de equipos y el poder exportarla a una hoja de cálculo la cual, será de utilidad para el usuario. Será posible visualizar las estadísticas de las relaciones estudiantes-líneas, profesionales-líneas y equipos-líneas. Se contará con una base de datos donde se almacena los resultados del procesamiento y la lista de equipos generada por el usuario permitiendo esto poder guardar los avances del proceso de selección de equipos al cerrar la aplicación para un uso posterior.

Para un mejor entendimiento se conformó el siguiente modelo conceptual:

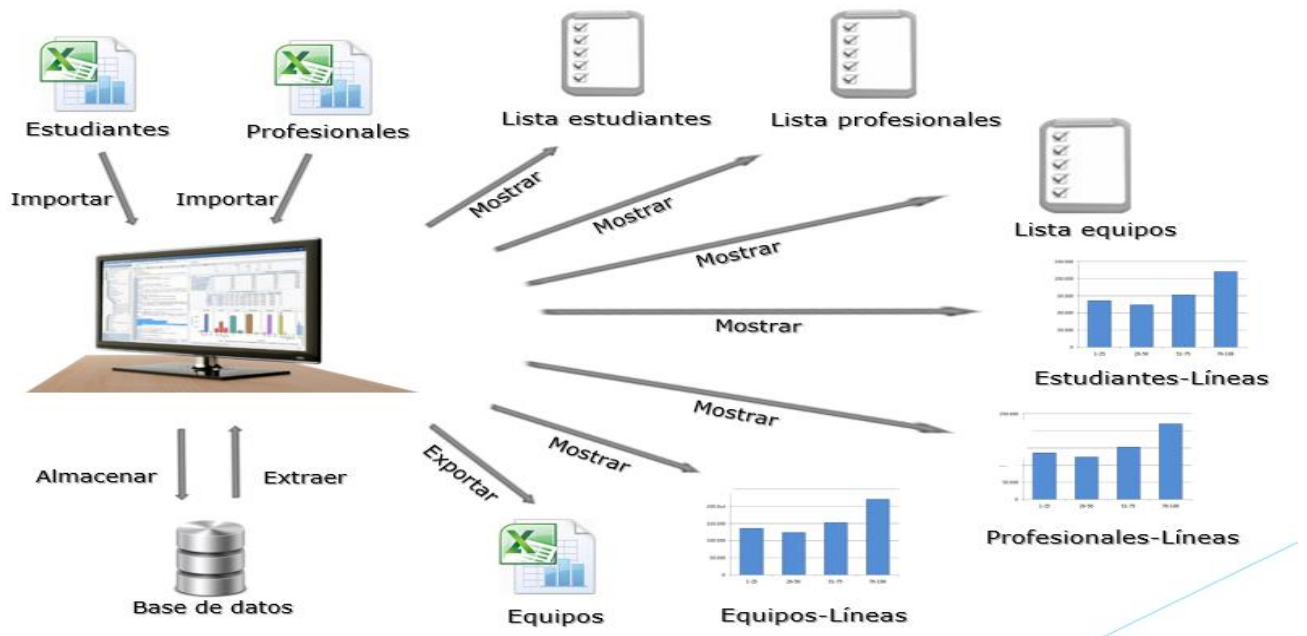


Figura 2. Modelo conceptual del sistema.

2.2 Características del sistema

Una característica del sistema no es más que una condición o necesidad que presenta un usuario para resolver un problema o alcanzar un objetivo. También se puede decir que es una capacidad que debe estar presente en un sistema para satisfacer un contrato. Las características del sistema pueden dividirse en funcionales y no funcionales (Sommerville, 2015).

2.2.1 Requisitos no funcionales del sistema

Las características no funcionales expresan exigencias de cualidades que deben tener los sistemas. Éstos, señalan restricciones, y hacen al producto atractivo, usable, rápido o confiable. En la mayoría de los casos, influyen en el éxito de todo producto informático (Sommerville, 2015).

Para lograr una buena calidad en el sistema se listaron las siguientes propiedades y cualidades:

Usabilidad

- Debe tener una interfaz gráfica, visualmente atractiva para el usuario, la aplicación podrá ser usada por cualquier usuario con conocimientos básicos sobre informática y debe mostrar mensajes al usuario que le ayuden a llevar a cabo la tarea a realizar.

Seguridad

- Integridad: El sistema debe impedir la corrupción en sus resultados. Para el cumplimiento de este requerimiento, la aplicación no contará con la opción de editar los resultados referentes a las líneas de investigación lo que evitará una modificación en la relación Profesional-Línea.
- Disponibilidad: La aplicación deberá estar disponible en todo momento para el personal de toma de decisiones.

Software

Se debe tener instalado el gestor de base de datos PostgreSQL 9.0 o superior.

Hardware

- La estación de trabajo debe contar con al menos 1,0 GB de *Random Access Memory* (RAM, por sus siglas en inglés).
- La unidad de almacenamiento o disco duro deberá poseer al menos 1,0 GB.
- El procesador deberá contar con una velocidad de 1.0 GHz o superior.

2.2.2 Funcionalidades del sistema

Un requisito funcional (RF) define una función del sistema de software o sus componentes. Una función es descrita como un conjunto de entradas, comportamientos y salidas. Los requerimientos funcionales pueden ser: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que se supone que un sistema debe cumplir. Estos son complementados por los requisitos no funcionales, que se enfocan en cambio, en el diseño o la implementación (Sommerville, 2015).

A continuación, se muestran los RF identificados:

RF1: Introducir Nombre de la Base de Datos.

RF2: Introducir Contraseña del Servidor.

RF3: Mostrar Ayuda.

RF4: Importar Archivos de profesionales y estudiantes.

RF4.1: Localizar Archivos

RF4.2: Validar Archivos.

RF4.3: Procesar Archivos.

RF5: Seleccionar Tipo de Búsqueda.

RF6: Buscar Estudiante.

RF7: Mostrar Líneas de Investigación.

RF8: Mostrar Listas de Personas.

RF8.1: Mostrar Lista de los Estudiantes por Línea de Investigación.

RF8.2: Mostrar Lista de los Profesionales por Línea de Investigación.

RF9: Seleccionar Una Persona de las Listas de Personas.

RF10: Agregar Persona al Equipo en Construcción.

RF11: Mostrar Equipo en Construcción.

RF12: Seleccionar Persona del Equipo en Construcción.

RF13: Eliminar Persona del Equipo en Construcción.

RF14: Mostrar Lista de Equipos Conformados.

RF15: Gestionar Lista de Equipos Conformados.

RF15.1: Eliminar Equipo.

RF15.2: Modificar Equipo.

RF15.3: Seleccionar Equipo.

RF15.4: Agregar Equipo.

RF16: Mostrar Relación Estudiantes-Líneas.

RF17: Mostrar Relación Profesionales-Líneas.

RF18: Mostrar Relación Equipos-Líneas.

RF19: Exportar Lista de Equipos Hacia una Hoja de Cálculo.

RF20: Salir

2.3 Fase de exploración

La fase de exploración es la primera fase definida por la metodología XP, en esta se define el alcance real del sistema permitiendo una familiarización del equipo de desarrollo con las herramientas, tecnologías y procesos. Esta fase comienza por la creación de una serie de historias de usuarios

(HU) las cuales definen mediante su redacción qué es lo que verdaderamente necesita el cliente y es aquí donde los programadores estiman el tiempo de desarrollo. (Joskowicz, 2008)

2.3.1 Personal relacionado con el sistema

Se define como personal relacionado con el sistema a todas aquellas personas que realizan una función o interactúan con el sistema de una forma u otra.

Tabla 7. Personal relacionado con el sistema.

Personal relacionado con el sistema	Justificación
Usuario	Personal designado a la toma de decisiones para la conformación de equipos de investigación.

2.3.2 Historias de Usuarios

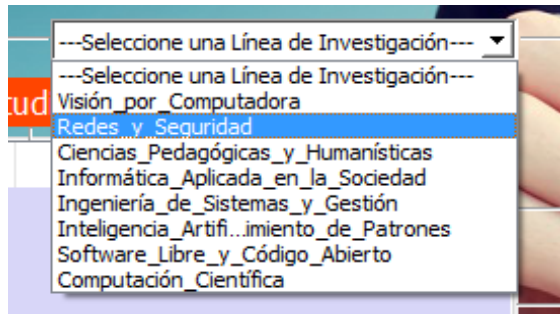
Las HU son la técnica utilizada en XP para especificar las funcionalidades del software, brindan detalles sobre la estimación del riesgo y cuánto tiempo será empleado en su implementación. El cliente es el encargado de asignarle una prioridad a cada HU y es el equipo de desarrollo el encargado de asignarle un costo, este se traduce en las semanas que llevará el desarrollo de las mismas. Si las HU según lo planificado demoran en desarrollarse, se sugiere dividirla en HU más pequeñas. También, es importante destacar que las HU nuevas pueden describirse en cualquier momento, con esto se comprueba la flexibilidad de la metodología (Beck, 2000).

A continuación, se muestra una de las 26 HU definidas por el cliente y el equipo de desarrollo, las restantes se encuentran en el **¡Error! No se encuentra el origen de la referencia.:**

Tabla 8. Historia de Usuario 9.

Historia de Usuario	
Número: 9	Usuario: Sistema
Nombre de historia: Mostrar Líneas de Investigación.	
Prioridad: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1/2	Iteración asignada: 1
Programador responsable: Ernesto Miró Peraza.	
Descripción: Mostrar la lista de los estudiantes y sus líneas de investigación como producto	

Interfaz:



Observaciones:

- Se mostrarán las líneas en dependencia de cual tipo de búsqueda este seleccionada.

2.4 Fase de planificación

Durante esta fase se realiza una estimación del esfuerzo que costará implementar cada historia de usuario. Esta estimación se expresa utilizando como medida el punto. Un punto se considera como una semana ideal de trabajo donde los miembros de los equipos de desarrollo trabajan el tiempo planeado sin interrupción alguna. Esta fase permite enmarcar el alcance del proyecto, que tiene como fin el desarrollo y la entrega del sistema requerido. (Joskowicz, 2008)

2.4.1 Estimación de esfuerzo por historia de usuario

Para el desarrollo de la aplicación propuesta se ha realizado una estimación de esfuerzo por cada una de las HU identificadas, a continuación, se muestran estos resultados:

Tabla 9. Estimación de esfuerzo por historia de usuario.

Historia de Usuario	Puntos de Estimación
1. Introducir Nombre de la Base de Datos.	1/2
2. Introducir Contraseña del Servidor.	1/2
3. Mostrar Ayuda.	1/4
4. Localizar Archivos.	1
5. Validar Archivos.	1
6. Procesar Archivos.	2
7. Seleccionar Tipo de Búsqueda.	1/4
8. Buscar Estudiante.	1/2

9. Mostrar Líneas de Investigación.	1/2
10. Mostrar Lista de los Estudiantes por Línea de Investigación.	1/2
11. Mostrar Lista de los Profesionales por Línea de Investigación.	1/2
12. Seleccionar Una Persona de las Listas de Personas.	1/2
13. Agregar Persona al Equipo en Construcción.	1
14. Mostrar Equipo en Construcción.	1
15. Seleccionar Persona del Equipo en Construcción.	1/4
16. Eliminar Persona del Equipo en Construcción.	1/2
17. Mostrar Lista de Equipos Conformados.	1
18. Eliminar Equipo.	1/2
19. Modificar Equipo.	1/2
20. Seleccionar Equipo.	1/2
21. Agregar Equipo.	1/2
22. Mostrar Relación Estudiantes-Líneas.	1/2
23. Mostrar Relación Profesionales-Líneas.	1/2
24. Mostrar Relación Equipos-Líneas.	1/2
25. Exportar Lista de Equipos Hacia una Hoja de Cálculo.	1/2
26. Salir	1/4

2.4.2 Plan de iteraciones

Una vez identificadas las HU del sistema y estimado el esfuerzo dedicado a la realización de cada una de estas, se procede a definir qué Historia de Usuario será implementada para cada iteración, que no es más que la planificación de la etapa de implementación del sistema estableciendo una sola iteración (Beck, 2000).

- Iteración 1: Se desarrollarán las Historias de Usuario 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 y 11, siendo estas la base fundamental del sistema, pues permiten importar y procesar los archivos Excel y mostrar los resultados del procesamiento de los mismos lo que da paso a las demás funcionalidades.
- Iteración 2: Se llevará a cabo el desarrollo de las Historias de Usuario 12, 13, 14, 15, 16, 17, 18, 19, 20 y 21 las cuales poseen gran prioridad para el cliente porque permiten conformar los equipos con el personal que posee la relación por línea de investigación.
- Iteración 3: Se encargará de la implementación de las Historias de Usuario 22, 23, 24, 25 y 26. En general esta iteración permite obtener el resultado de lo implementado en las

iteraciones anteriores según la petición del cliente, dando la posibilidad de exportar la lista resultante a una hoja de cálculo. Una vez concluida se obtendrá la versión 1.0 del producto.

Plan de duración de las iteraciones

Como parte del ciclo de vida de un proyecto utilizando la metodología XP se crea el plan de duración de cada una de las iteraciones. Este plan se encarga de mostrar las HU que serán complementadas en cada una de las iteraciones, así como la duración estimada de cada una y el orden en que se implementarán (Beck, 2000).

Tabla 10. Plan de duración de las iteraciones.

Iteración	Orden de la Historia de Usuario a Implementar	Duración Total
1	<ol style="list-style-type: none"> 1. Introducir Nombre de la Base de Datos. 2. Introducir Contraseña del Servidor. 3. Mostrar Ayuda. 4. Localizar Archivos. 5. Validar Archivos. 6. Procesar Archivos. 7. Seleccionar Tipo de Búsqueda. 8. Buscar Estudiante. 9. Mostrar Líneas de Investigación. 10. Mostrar Lista de los Estudiantes por Línea de Investigación. 11. Mostrar Lista de los Profesionales por Línea de Investigación. 	7 y ½ semanas
2	<ol style="list-style-type: none"> 12. Seleccionar Una Persona de las Listas de Personas. 13. Agregar Persona al Equipo en Construcción. 14. Mostrar Equipo en Construcción. 15. Seleccionar Persona del Equipo en Construcción. 	6 y 1/4 semanas

	<p>16. Eliminar Persona del Equipo en Construcción.</p> <p>17. Agregar Equipo.</p> <p>18. Mostrar Lista de Equipos Conformados.</p> <p>19. Seleccionar Equipo.</p> <p>20. Eliminar equipo.</p> <p>21. Modificar Equipo.</p>	
3	<p>22. Mostrar Relación Estudiantes-Líneas.</p> <p>23. Mostrar Relación Profesionales-Líneas.</p> <p>24. Mostrar Relación Equipos-Líneas.</p> <p>25. Exportar Lista de Equipos Hacia una Hoja de Cálculo.</p> <p>26. Salir.</p>	2 y 1/4 semanas

2.4.3 Plan de entrega

En el plan de entrega se detalla la fecha inicio de la implementación del sistema y la fecha final de cada iteración, los productos obtenidos divididos por subsistemas, así como el módulo sobre el cual se está implementando (Beck, 2000).

Tabla 11. Plan de Entrega.

Sistema	Final iteración 1	Final iteración 2	Final iteración 3
Fecha de inicio 1/febrero/2017)	(25/marzo/2017)	(10/mayo/2017)	(27/mayo/2017)
Sistema para conformar equipos entre estudiantes y profesionales partiendo de las líneas de investigación.	v0.1	v0.2	v1.0

2.5 Fase de diseño

La metodología de desarrollo XP plantea prácticas especializadas que accionan directamente en la realización del diseño para lograr un sistema robusto y reutilizable. Se trata en todo momento de conservar su simplicidad, es decir, crear un diseño evolutivo que va mejorando incrementalmente y que permite hacer entregas pequeñas y frecuentes de valor para el cliente, basado principalmente en el desarrollo de las tarjetas Clase-Responsabilidad-Colaboración (CRC). (Joskowicz, 2008)

2.5.1 Arquitectura

La arquitectura de software de un sistema informático, se basa en la estructura y organización del mismo, proporcionando una forma coherente de establecer los patrones y abstracciones para que los analistas y desarrolladores trabajen en una misma línea hacia la implementación del sistema (Sommerville, 2015).

Estilo arquitectónico a utilizar

Un estilo es un concepto descriptivo que define una forma de articulación u organización arquitectónica. El conjunto de los estilos cataloga las formas básicas posibles de estructuras de software. Estos permiten expresar un esquema de organización estructural esencial para un sistema de software (Pressman 2005). En este trabajo se hace uso del estilo arquitectónico en capas, logrando que el sistema quede organizado y así tener un orden lógico en la programación del mismo.

Arquitectura en capas

La arquitectura en capas se define como una organización jerárquica donde cada capa proporciona servicios a la inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. Con esto se logra abstraer las funcionalidades de una capa de manera tal que pueda ser totalmente remplazada sin afectar a las otras, solamente cambiar las referencias de las implicadas en el cambio (PELÁEZ, 2009). En la siguiente ilustración se presenta la arquitectura de la solución.

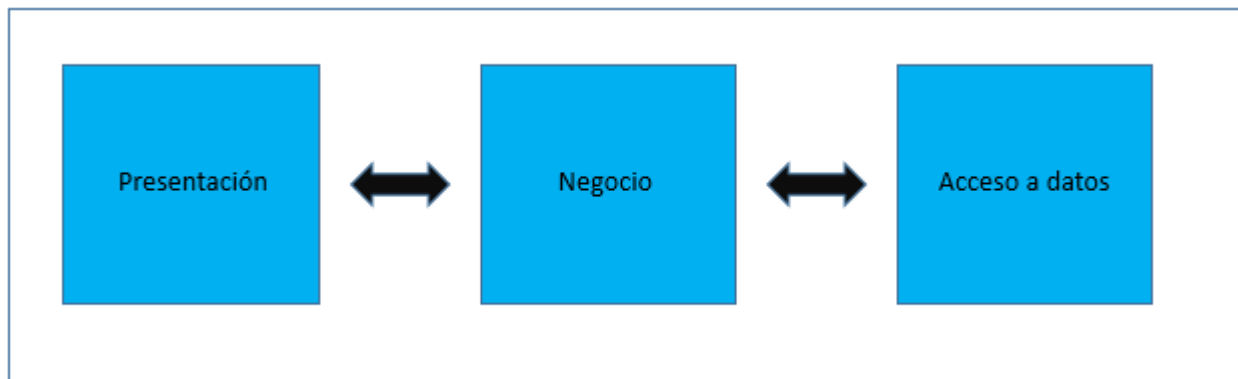


Figura 3. Arquitectura en capas para la solución.

Capa de presentación: es la parte de la aplicación con que el usuario interactúa, por lo que deberá cumplir muchos requisitos. Estos requisitos abarcan factores generales como la facilidad de uso, rendimiento, diseño e interactividad. Es importante que la aplicación tenga un buen diseño para apoyar una experiencia de usuario intuitiva, desde el principio, ya que la experiencia del usuario es influenciada por muchos aspectos diferentes de la arquitectura de la aplicación.

Capa de negocio: es donde residen las clases gestoras de la información, se reciben las peticiones del usuario y se envían las respuestas a la capa de presentación. Se nombra capa de negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes del usuario y presentar los resultados obtenidos, y con la capa de acceso a datos, para enviar datos que necesitan persistirse en la base de datos o recibirlos de la misma.

Capa de acceso a datos: está constituida por las clases gestoras del acceso a datos, encargadas de acceder a los mismos y realizan todo el almacenamiento de la información. Esta capa recibe solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

2.5.2 Patrones de diseño

Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software. Un patrón de diseño es una solución a un problema de diseño, una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular, el cual es el encargado de identificar Clases, Instancias, Roles, Colaboraciones y la distribución de responsabilidades. Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software (Larman, 2005).

2.5.2.1 Patrones para Asignar Responsabilidades

Los Patrones GRASP (General Responsibility Assignment Software Patterns, Patrones Generales de Software para Asignación de Responsabilidades) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. A continuación, se explican los patrones de asignación de responsabilidades que se implementan en el software a realizar (Larman, 2005).

Experto: Es el principio básico de asignación de responsabilidades, el cual indica que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo se obtendrá un diseño con mayor cohesión, la información se mantiene encapsulada y permite contar con un sistema robusto y fácil de mantener (Larman, 2005). En la siguiente ilustración se muestra una clase como ejemplo del patrón:

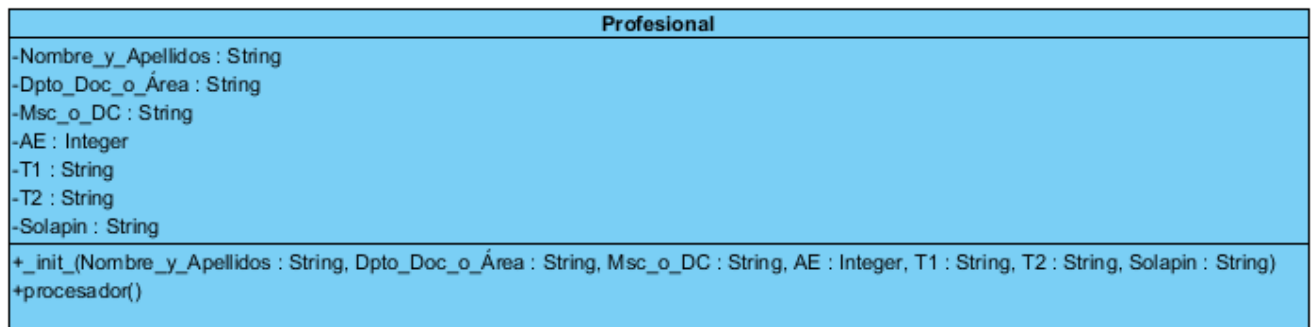


Figura 4. Arquitectura en capas para la solución.

Creador: El patrón creador ayuda a identificar quién debe ser el responsable de la creación o instanciación de nuevos objetos o clases, de forma tal que una instancia de un objeto sólo pueda ser creada por el objeto que contiene la información necesaria para ello (Larman, 2005).

```

from charts.Ui_grafica import Ui_Form
from PyQt5.QtWidgets import QMainWindow

from dd.BD import DB
from PyQt5 import QtCore

class grafica(QMainWindow, Ui_Form):
    def __init__(self, graf, nombre_db, passwd):
        super(grafica, self)
        self.nombre_db=nombre_db
        self.passw=passwd
        self.setupUi(self)
        if graf==1:
            self.calculator_equipos()
        elif graf==2:
            self.calculator_estudiantes()

        else:
            self.calculator_profesionales()

        self.show()

    def calculator_equipos(self):

        db=DB(self.nombre_db,self.passw)
        maxi=302
        totalequipos=db.seleccionar_todos('equipo_prof')
        totalequipos=len(totalequipos)
        if totalequipos==0:
            totalequipos=1

```

Figura 5. Ejemplo del patrón creador.

Alta cohesión: Se aplica para realizar un diseño que evite contener clases con un alto grado de abstracción, que asuman responsabilidades que podían haber delegado a otros objetos o que tengan responsabilidades muy complejas y este plantea que la información que almacena una clase debe de ser coherente y debe estar (en la medida de lo posible) relacionada con la clase (Larman, 2005). Este patrón se puede evidenciar en todas las clases.

Bajo Acoplamiento: Es la idea de tener las clases lo menos ligadas entre sí como sea posible. De tal forma que, en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases (Larman, 2005).

2.5.3 Tarjetas Clase - Responsabilidad – Colaborador

Las tarjetas CRC constituyen una forma simple de organizar las clases más notables para las funcionalidades del sistema, con el objetivo de desarrollar una representación organizada de las clases.

Un modelo CRC es una colección de tarjetas índices estándar que representan clases. Las tarjetas se dividen en tres secciones. A lo largo del borde superior de la tarjeta se escribe el nombre de la clase. En el cuerpo de la tarjeta a la izquierda se listan las responsabilidades de la clase que no es más que lo que la clase sabe o hace y a la derecha los colaboradores que son aquellas clases que se requieren para que una clase reciba la información necesaria para completar una responsabilidad (Pressman, 2005).

A continuación, se muestran dos tarjetas CRC, el resto se encuentra en el **¡Error! No se encuentra el origen de la referencia.:**

Tabla 12. Tarjeta CRC MainWindow.

Clase: MainWindow	
Responsabilidad	Colaboración
Es la clase controladora principal.	Ayuda Grafica Estudiante Profesional DB Ui_MainWindow

Tabla 13. Tarjeta CRC Ui_MainWindow.

Clase: Ui_MainWindow	
Responsabilidad	Colaboración

Es la clase que define los elementos de la ventana principal.	MainWindow
---	------------

2.5.4 Diseño de la base de datos

La construcción de la base de datos es una de las tareas principales en el diseño de una aplicación, en esta se ponen de manifiesto la estructura necesaria para el correcto funcionamiento de la misma.

A continuación, se muestra el DER (Diagrama Entidad Relación) diseñado para el sistema:

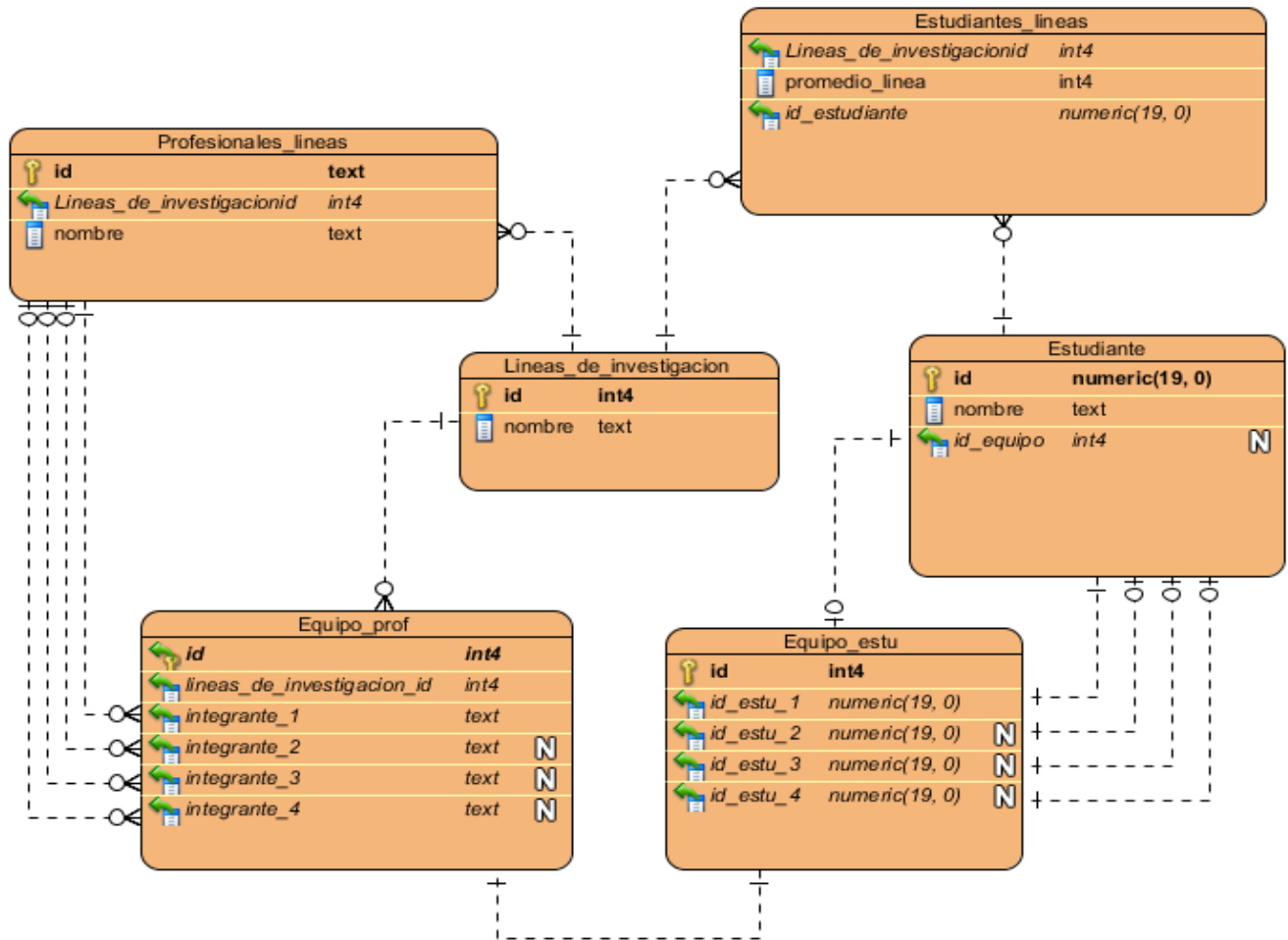


Figura 6. Diagrama Entidad Relación.

2.5.5 Tareas de la ingeniería

Las tareas de ingeniería (TI) son utilizadas como apoyo a las historias de usuarios brindando un detalle más profundo sobre la implementación de estas. Además, realizan una estimación del tiempo que demora cada una de ellas. Son el punto fundamental para el desarrollo del software (Beck, 2000).

A continuación, se muestran dos tareas de ingeniería correspondiente a cada HU, las restantes se encuentran en el **¡Error! No se encuentra el origen de la referencia.:**

Tabla 14. Tarea de Ingeniería 1.

Tarea de Ingeniería	
Número de Tarea: 1	Número Historia de Usuario: 1
Nombre de Tarea: Introducir Nombre de la Base de Datos.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1/2
Programador Responsable: Ernesto Miró Peraza	
Descripción: El usuario accede al sistema y debe introducir el nombre de la base de datos. En caso que desee abortar el proceso puede accionar los botones de salida o cancelar.	

Tabla 15. Tarea de Ingeniería 2.

Tarea de Ingeniería	
Número de Tarea: 2	Número de Tarea: 2
Nombre de Tarea: Introducir Contraseña del Servidor.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 1/2
Programador Responsable: Ernesto Miró Peraza	

Descripción: En sucesión de la entrada del nombre de la base de datos, el usuario debe introducir la contraseña del servidor donde se encuentra la base de datos. En caso que desee abortar el proceso puede accionar los botones de salida o cancelar.

2.6 Conclusiones del capítulo

La propuesta de solución definida facilitó la comprensión del sistema para realizar el proceso de conformación de equipos de investigación entre estudiantes y profesionales. La identificación de los requisitos permitió un mayor entendimiento de las necesidades del cliente. Mediante la descripción de las HU divididas por iteraciones y la planificación del esfuerzo dedicado al desarrollo de cada una de ellas, se logró una mejor organización del trabajo y el establecimiento de fechas de culminación para cada iteración. La utilización del estilo arquitectónico en capas y del patrón de diseño GRASP permitió una mejor estructuración de la aplicación fomentando el diseño de la misma.

Capítulo 3. Implementación y Prueba

En el siguiente capítulo se define el estándar de codificación que se estará utilizando en el desarrollo de la solución. Además, se realizan las pruebas definidas por la metodología seleccionada, así como las pruebas unitarias para verificar el código, y las pruebas de aceptación para comprobar si al final de cada iteración se consiguió la funcionalidad requerida. Se realiza un caso de estudio para verificar la validez de los resultados de la solución propuesta.

3.1 Fase de implementación

Luego de haber definido los elementos necesarios en la etapa de planificación y diseño se pasa a la de codificación o implementación de la aplicación, donde se da cumplimiento al plan de iteraciones. En esta fase se realiza la implementación de las HU que fueron seleccionadas por cada iteración. Al finalizar esta fase el cliente estará listo para comenzar a realizar las pruebas pertinentes. (Joskowicz, 2008)

3.1.1 Estándares de codificación

La metodología XP resalta que la comunicación de los programadores es a través del código implementado, por lo que es necesario que sigan ciertos estándares de programación, de manera que cualquier persona del equipo de desarrollo pueda modificar el código. Además, se hace preciso que el código sea entendible para que posteriormente otros programadores puedan apoyarse en el trabajo y desarrollen otras soluciones.

En el caso de la herramienta que se desarrolla, el estándar que se utiliza es:

Importaciones.

- Las importaciones se encuentran en líneas separadas.

Comentarios

- Se utilizan comentarios de una línea para guiar al lector con la información referente al código.

Comentarios de una línea: comentario pequeño que solo abarca una línea y describe el código que le sigue.

```
# Esto es un comentario de una línea
```

Estilo de los nombres

- **Clases e Interfaces:** los nombres de las clases presentan la primera letra en mayúscula, en caso de ser un nombre compuesto, la inicial de cada palabra se representa en mayúscula y van separadas por un guión bajo “_”. Se utilizan nombres simples y de alguna manera que describan el contenido, se usan palabras completas, a no ser que la abreviatura sea muy conocida.
- **Métodos y variables:** los nombres de los métodos se representan en minúscula y en caso de ser un nombre compuesto se separan por un guión bajo “_”. El nombre de las variables debe ser relativamente corto y expresando el significado del contexto en que sea utilizado.

3.1.2 Interfaces del sistema

Una vez concluido el desarrollo del sistema, se obtiene la fusión de los requisitos funcionales, los no funcionales y el diseño como propuesta de solución al problema planteado. En las siguientes figuras se muestran las funcionalidades más importantes del sistema en forma de flujo para crear un equipo y exportarlo.

Primero se selecciona la opción “Importar archivos”

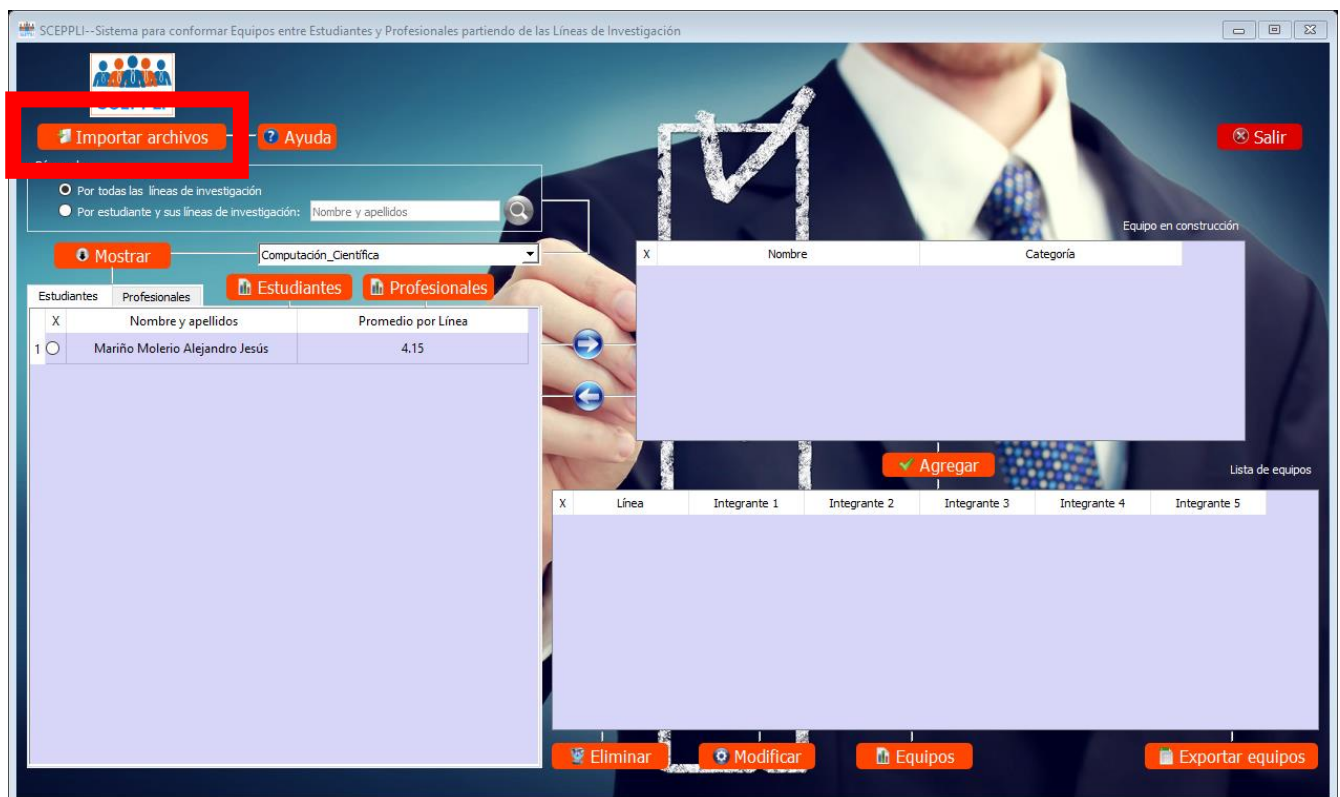


Figura 7. Importar archivo. Paso 1.

Se localizan los archivos en un directorio del sistema operativo.

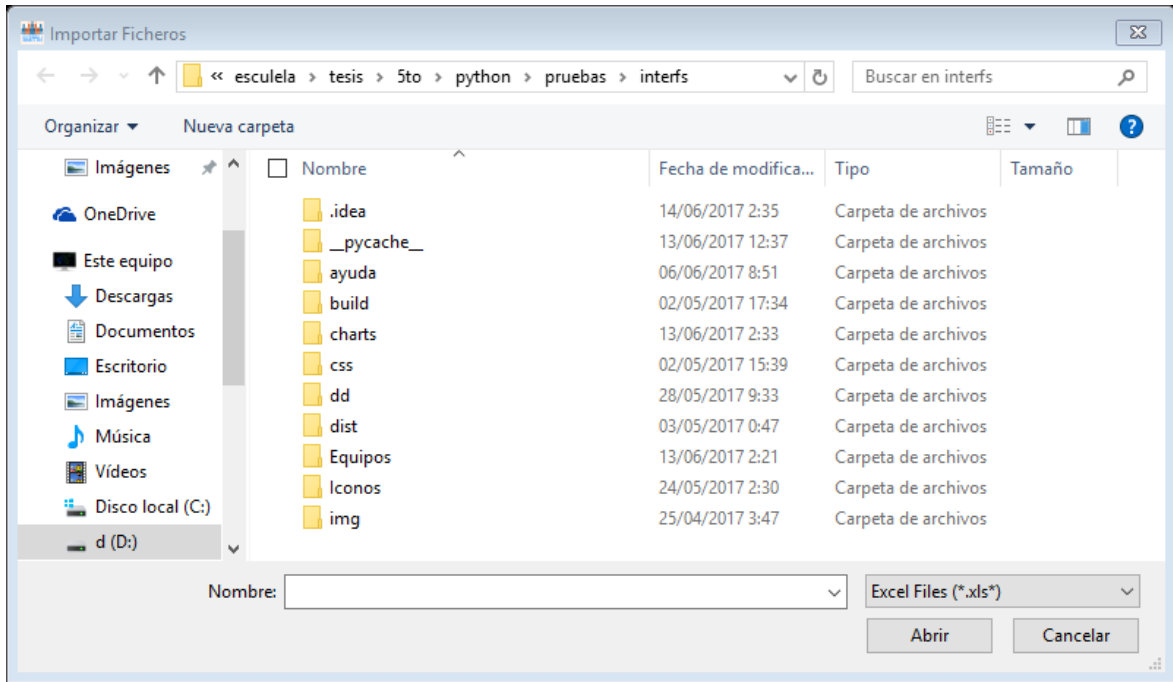


Figura 8. Localizar archivos.

Seleccionar el tipo de búsqueda a realizar.

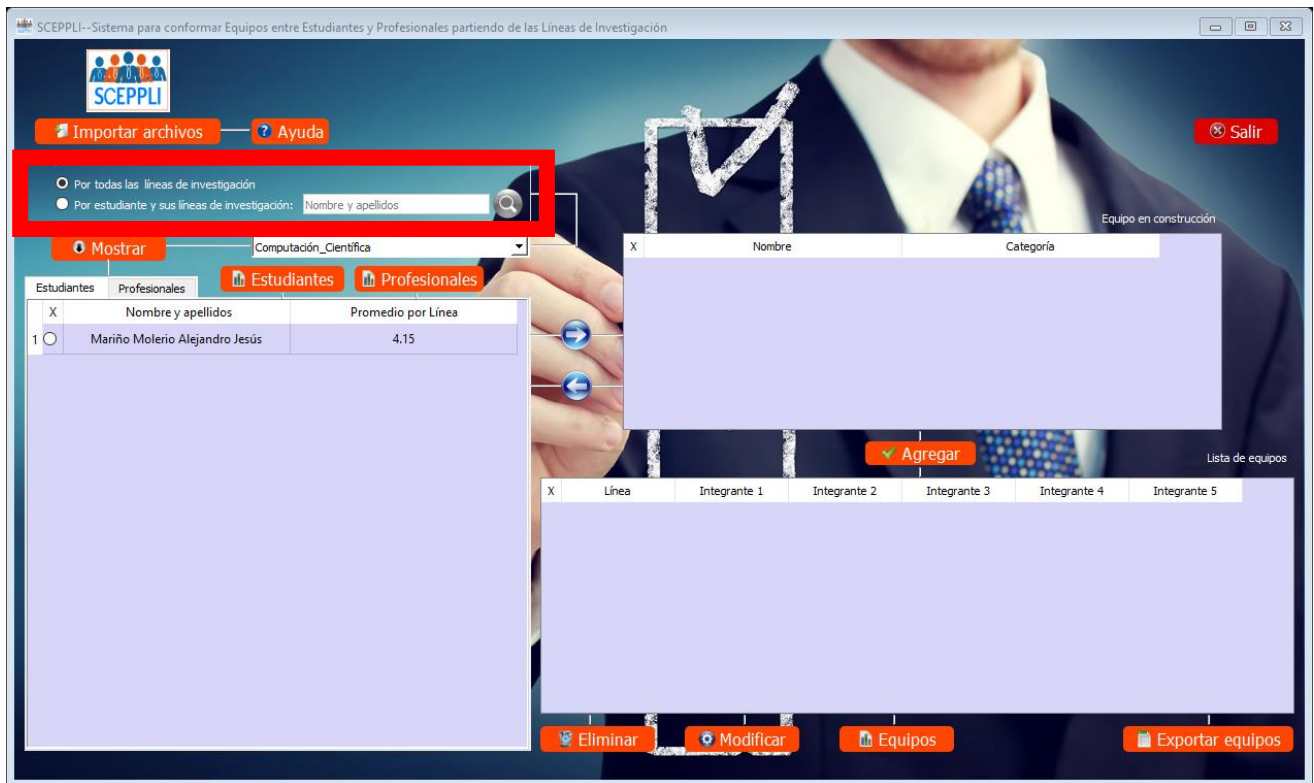


Figura 9. Seleccionar tipo de búsqueda.

Seleccionar una línea de investigación para mostrar sus clasificados.

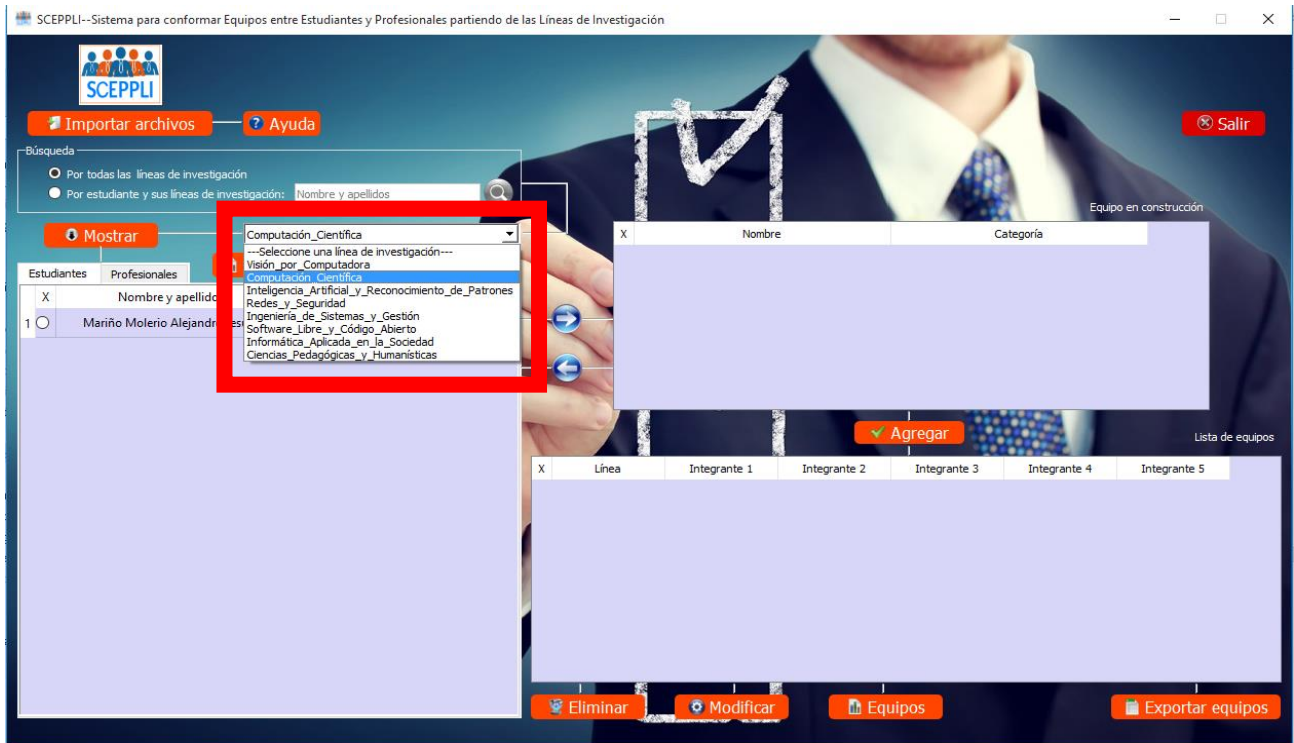


Figura 10. Seleccionar una línea de investigación.

Se selecciona la opción Mostrar para mostrar las listas de personas de la línea seleccionada.

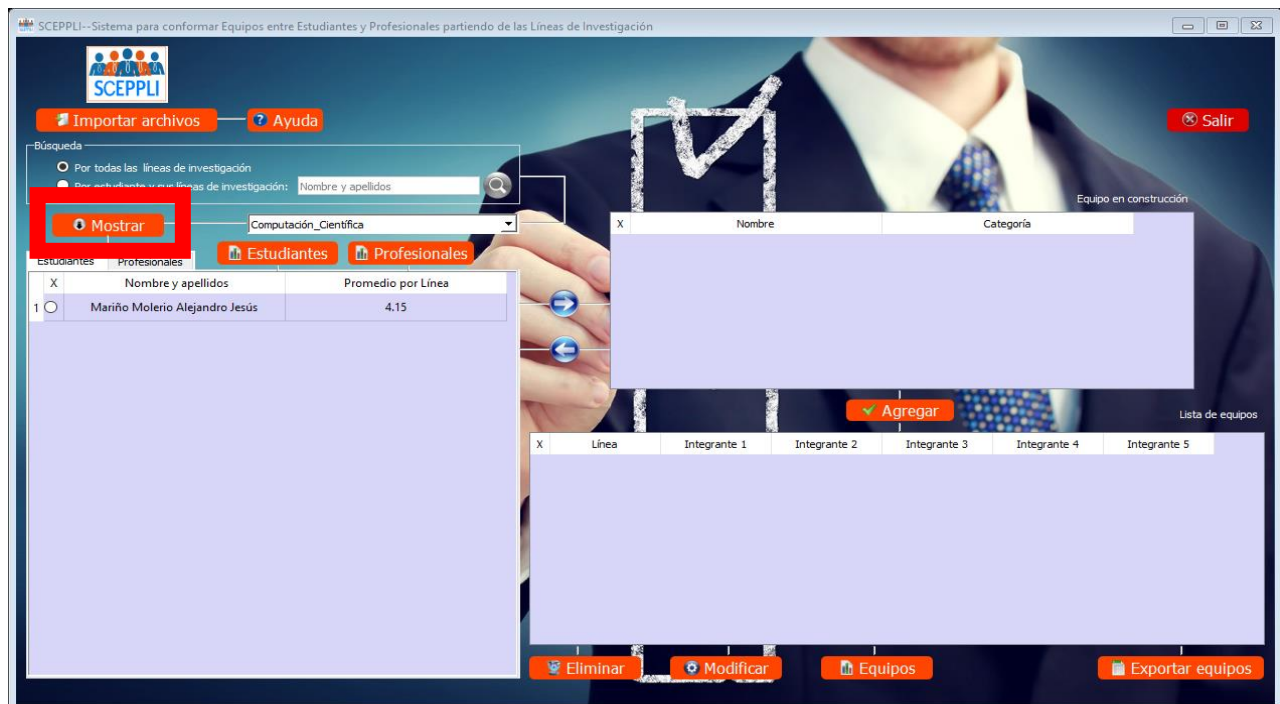


Figura 11. Seleccionar botón Mostrar.

Se muestran las personas clasificadas por la línea seleccionada.

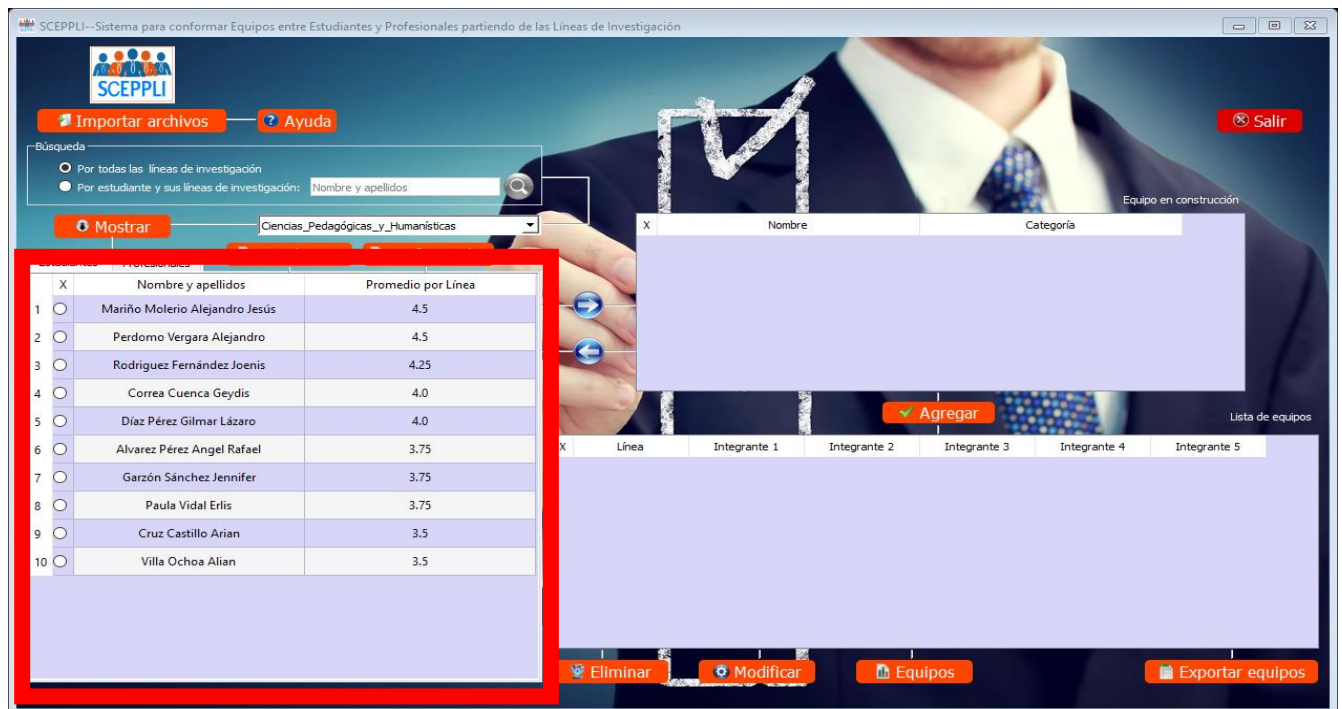


Figura 12. Listas de personas.

Se selecciona uno a uno los integrantes para construir un equipo.

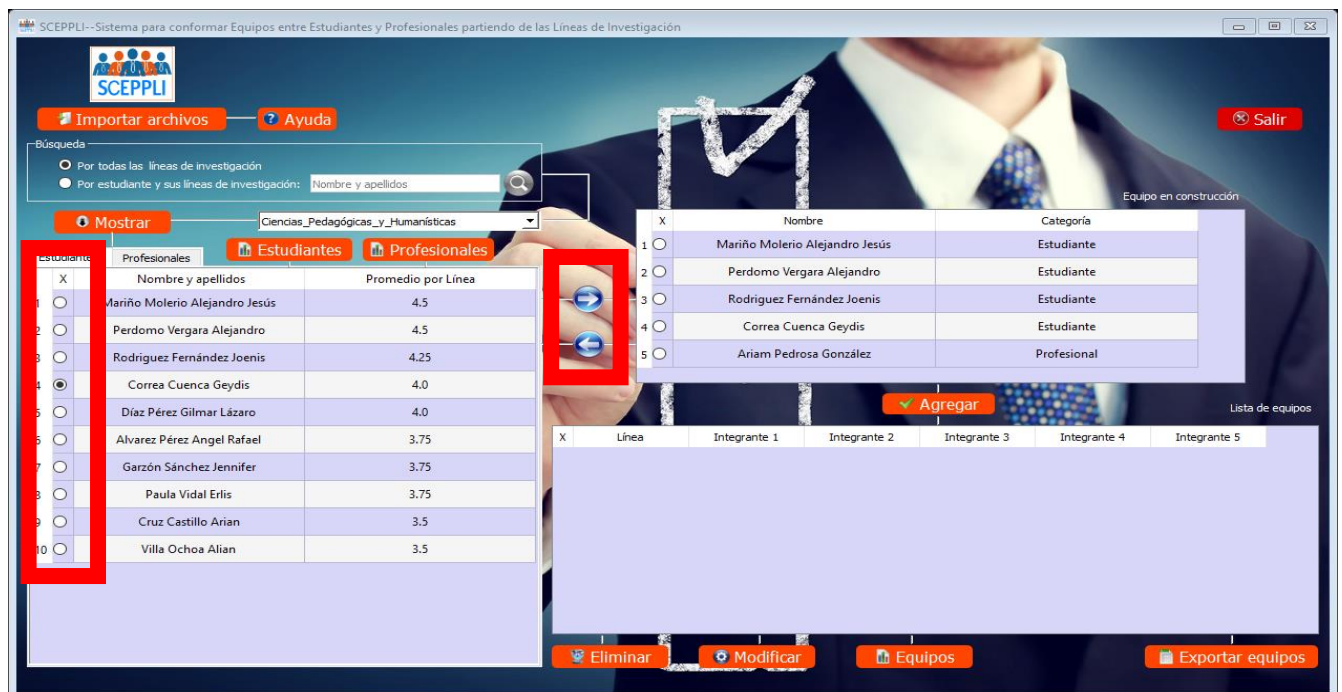


Figura 13. Construir un equipo.

Se muestran los integrantes del equipo en construcción.

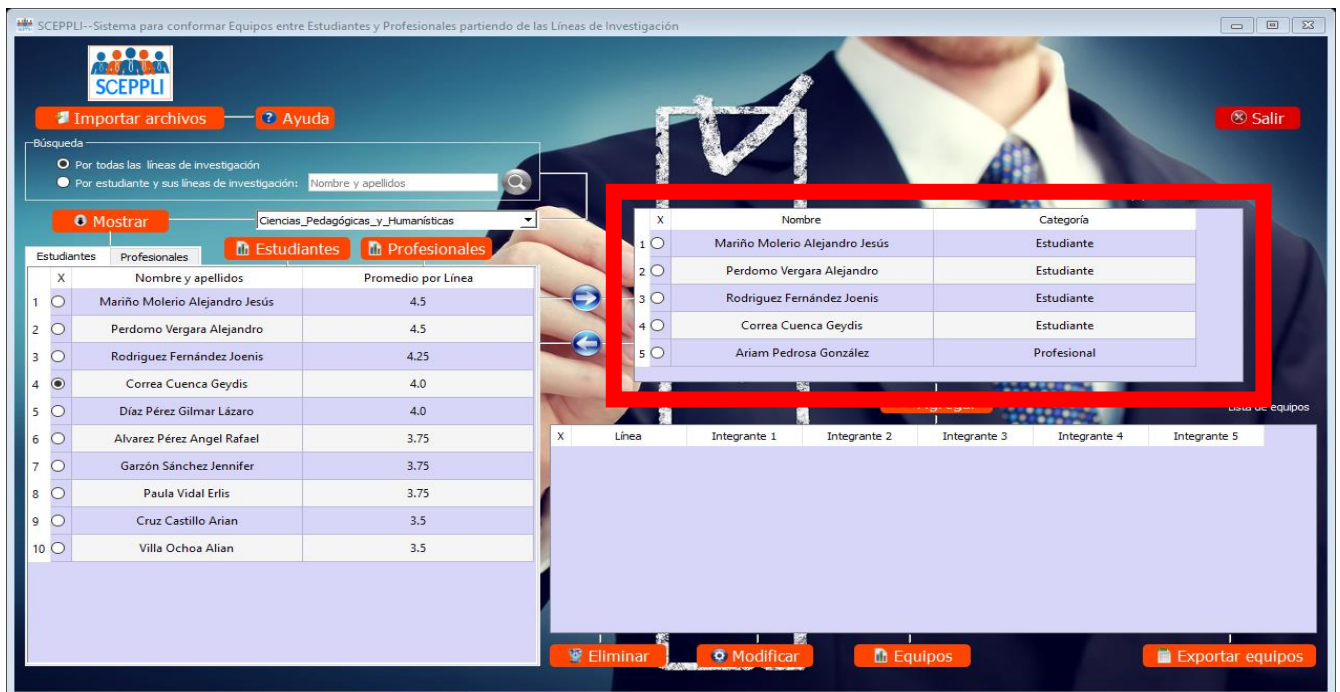


Figura 14. Equipo en construcción.

Agregar el equipo en construcción a la lista de equipos creados.

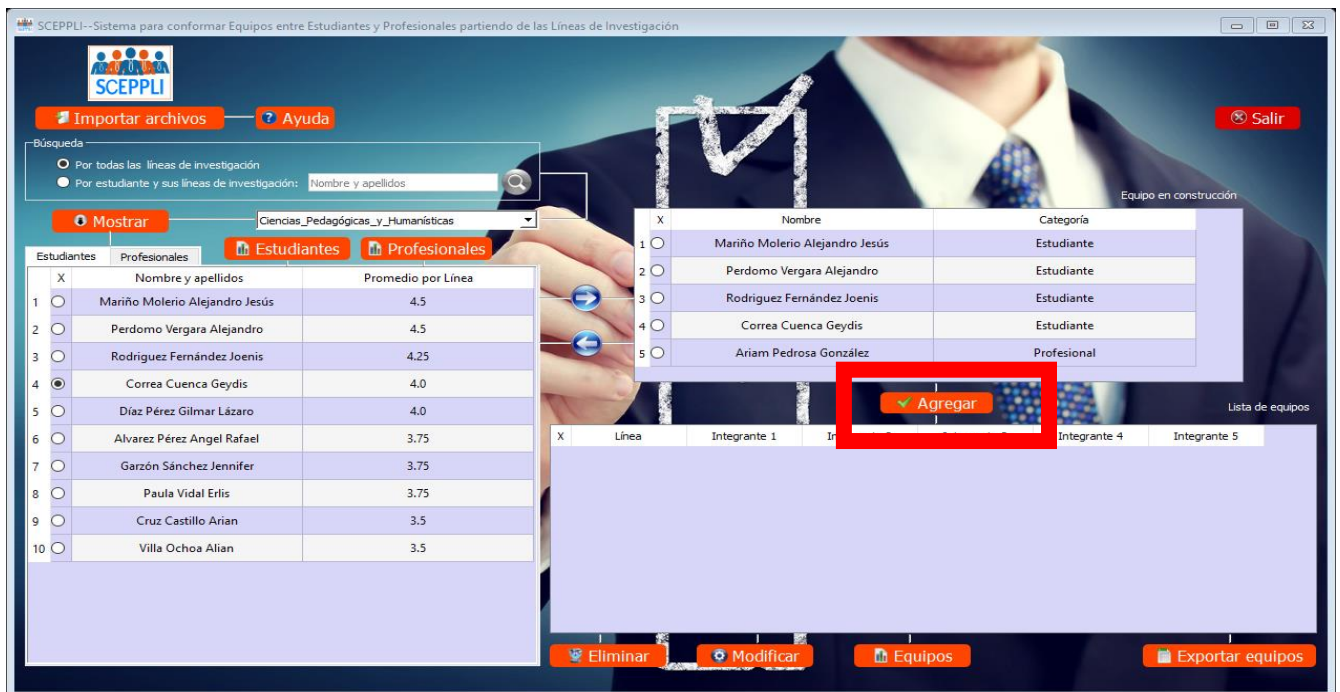


Figura 15. Agregar equipo.

Se muestran los equipos conformados.

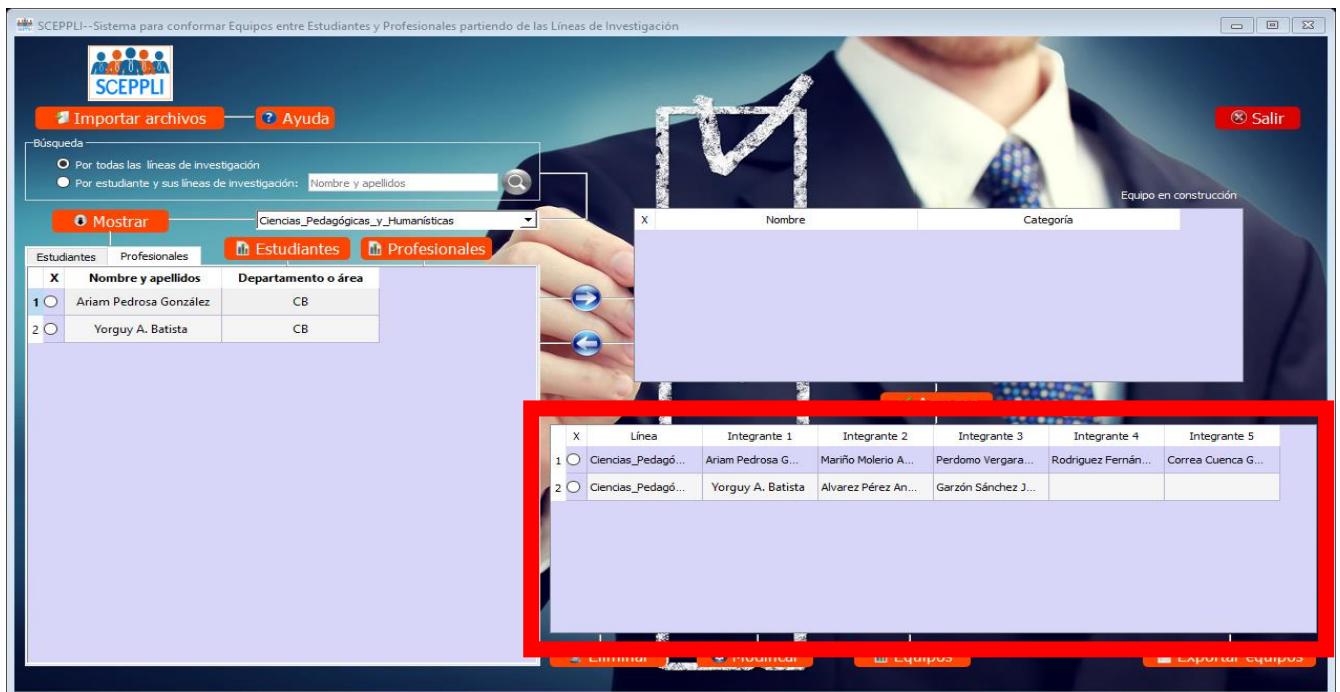


Figura 16. Lista de equipos creados.

Exportar equipos conformados a una hoja de cálculo seleccionando la opción Exportar equipos.

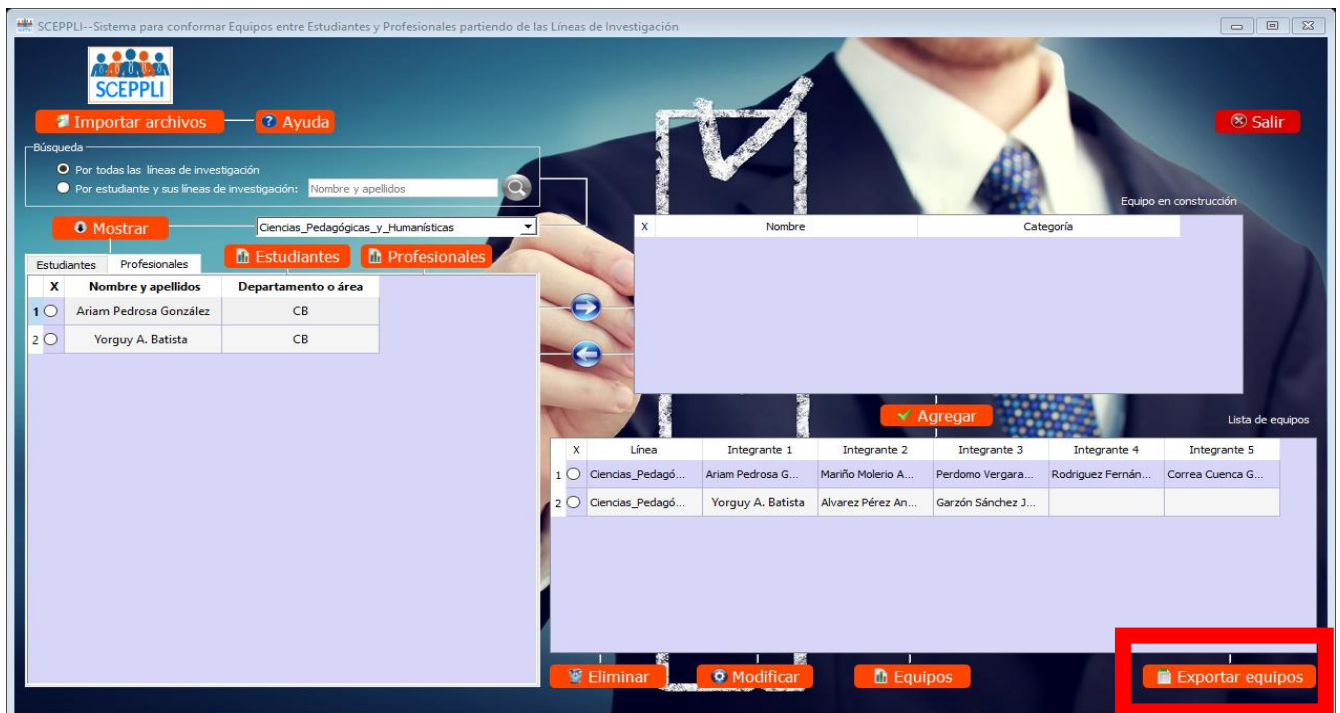


Figura 17. Exportar equipos.

Se exportan los equipos y se muestra la hoja de cálculo resultante.

The screenshot shows the Microsoft Excel interface with the 'Equipos.xlsx' file open. The spreadsheet contains the following data:

	A	B	C	D	E	F
1	Líneas de investigación	Integrante 1	Integrante 2	Integrante 3	Integrante 4	Integrante 5
2	Computación_Científica	Sandy Díaz Ramos	Mariño Molerio Alejandro Jesús			
3	Ciencias_Pedagógicas_y_Hur	Yorguy A. Batista	Garzón Sánchez Jennifer			
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						

Figura 18. Hoja de cálculo con los equipos.

3.2 Fase de prueba

Quando se desarrolla una solución informática se deben realizar pruebas para verificar que el programa funcione correctamente. Estas pruebas normalmente tienen que ser ejecutadas en varias ocasiones y se ven afectadas por los cambios que se introducen conforme se va construyendo la solución. XP divide las pruebas en dos grupos: pruebas de aceptación o pruebas funcionales diseñadas por el cliente final, destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida y pruebas unitarias, encargadas de verificar el código y diseñadas por los programadores. (Joskowicz, 2008)

3.2.1 Pruebas unitarias

Las pruebas unitarias son pruebas de caja blanca, encargadas de verificar el código y son diseñadas por los programadores. Cada uno de los desarrolladores tiene que ir probando constantemente lo que va obteniendo en el transcurso de la implementación del sistema, para garantizar que las funcionalidades exigidas por el cliente se estén implementando correctamente. Estas pruebas se realizaron cada vez que se terminaba de implementar una funcionalidad probándola directamente en el entorno real (Ble, 2010).

A continuación, se detallan los resultados obtenidos durante las 3 iteraciones realizadas. Se seleccionaron 17 fragmentos de código, detectándose 0 errores. Todas estas pruebas fueron realizadas con la herramienta Unittest, la cual está contenida en las librerías del lenguaje de programación Python.

Iteración 1: En la primera iteración se seleccionaron 4 fragmentos de código en los cuales no se detectaron errores.

```
Ran 4 tests in 0.141s
OK
Destroying test database for alias 'default'...
```

Figura 19. Pruebas de iteración 1.

Iteración 2: En la segunda iteración se seleccionaron 5 fragmentos de código en los cuales no se detectaron errores.

```
Ran 5 tests in 0.179s
OK
Destroying test database for alias 'default'...
```

Figura 20. Pruebas de iteración 2.

Iteración 3: En la tercera iteración se seleccionaron 8 fragmentos de código en los cuales no se detectaron errores.

```
Ran 8 tests in 0.337s
OK
Destroying test database for alias 'default'...
```

Figura 21. Pruebas de iteración 3.

3.2.2 Pruebas de Aceptación

Las pruebas de aceptación XP son especificadas por el cliente, y se centran en las características y funcionalidades generales del sistema, que son visibles y revisables por parte del usuario. Estas pruebas derivan de las HU que se han implementado como parte de la liberación del software (Joskowicz, 2008).

Tabla 16. Caso de prueba de aceptación HU10_P1.

Caso de prueba de aceptación	
Código: HU10_P1	Historia de Usuario: 10
Nombre: Mostrar Lista de los Estudiantes por Línea de Investigación.	
Descripción: Prueba para validar la funcionalidad Mostrar Lista de los Estudiantes por Línea de Investigación.	
Condiciones de ejecución: <ul style="list-style-type: none"> • El usuario debe haber seleccionado una de los tipos de búsqueda. • El usuario debe haber seleccionado una línea de investigación. • El usuario debe escoger la opción Mostrar. • El usuario debe seleccionar la pestaña Estudiantes. 	
Resultados esperados: En caso de no estar seleccionado el tipo de búsqueda, el sistema deberá mostrar un mensaje de pidiendo que se seleccione. En caso de estar seleccionado el tipo de búsqueda y no estar seleccionado la línea de investigación, el sistema deberá mostrar un mensaje de pidiendo que se seleccione la línea. En caso de no existir estudiantes en dicha línea se mostrará un mensaje. En cualquier otro caso, se muestran las listas.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 17. Caso de prueba de aceptación HU22_P1

Caso de prueba de aceptación	
Código: HU22_P1	Historia de Usuario: 22
Nombre: Mostrar Relación Estudiantes-Líneas.	
Descripción: Prueba para validar la funcionalidad Mostrar Relación Estudiantes-Líneas.	
Condiciones de ejecución: <ul style="list-style-type: none"> • El usuario debe escoger la opción Estadísticas de Estudiantes. 	
Resultados esperados: El sistema debe mostrar una ventana con una gráfica de barras la cual visualiza la relación estudiantes-líneas de investigación.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 18. Caso de prueba de aceptación HU25_P1

Caso de prueba de aceptación	
Código: HU25_P1	Historia de Usuario: 25
Nombre: Exportar Lista de Equipos Hacia una Hoja de Cálculo.	
Descripción: Prueba para validar la funcionalidad Exportar Lista de Equipos Hacia una Hoja de Cálculo.	
Condiciones de ejecución: <ul style="list-style-type: none"> El usuario debe escoger la opción Exportar Equipos. 	
Resultados esperados: El sistema debe generar una tabla con los equipos conformados en una hoja de cálculo Excel en una carpeta con el nombre Equipos dentro de la carpeta raíz del programa.	
Evaluación de la prueba: Prueba satisfactoria.	

Análisis de los resultados

Para validar que el resultado obtenido por el sistema coincide con el resultado esperado por el cliente se diseñaron un total de 7 casos de prueba de aceptación en conjunto cliente-desarrolladores. De este total, se obtuvieron 3 no conformidades las cuales fueron corregidas en la misma iteración dando como resultado 7 pruebas satisfactorias de 7 casos de prueba aplicados. De todo el proceso se obtiene un aval de aceptación el cual se muestra más adelante. Además de las pruebas con el cliente se realizaron pruebas con el grupo de calidad del centro de informatización de entidades de la UCI, ver **¡Error! No se encuentra el origen de la referencia..**

Aval:

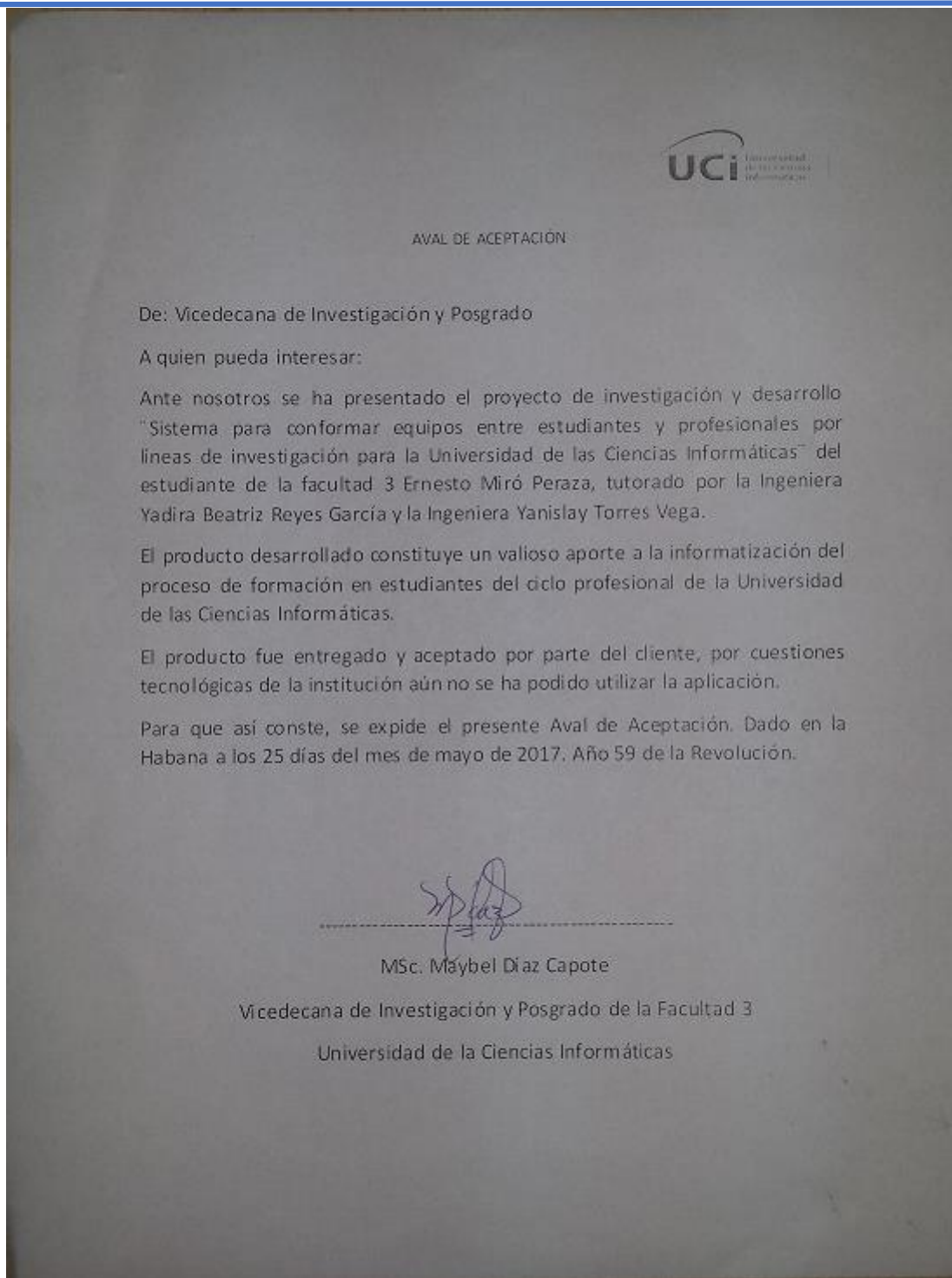


Figura 22. Aval de aceptación del cliente.

3.2.3 Pruebas de algoritmos.

Según lo descrito en el epígrafe 1.3.3 se tiene que aplicar un mecanismo de aprendizaje al sistema y para ello se cuenta con un caso de estudio a procesar y así evaluar sus resultados en comparativa

con los resultados que se obtienen haciendo el proceso manualmente. Para realizar dicha tarea se presentan 10 estudiantes y 10 profesionales donde los dictámenes definidos por el experto para evaluar la comparación de resultados se representan en bien o mal partiendo de que se está bien cuando no se encuentran errores y mal cuando se encuentra al menos 1.

Estudiantes

La siguiente tabla contiene las notas en las 37 asignaturas, en el mismo orden que en (**¡Error! No se encuentra el origen de la referencia.**), de los 10 estudiantes analizados.

Tabla 19. Caso de estudio. Notas de estudiantes.

1	5	4	4	5	5	4	5	5	5	5	4	5	5	5	5	4	5	5	5	-	-	5	-	5	5	5	-	-	-	-	5	5	-	5	-	-	-	
2	5	5	3	5	4	4	5	3	5	3	3	3	3	4	4	3	3	5	3	-	5	5	-	-	-	4	-	3	5	-	-	3	-	5	-	-	-	-
3	3	4	3	5	3	4	4	5	4	3	5	3	4	3	4	3	3	3	4	3	-	5	5	-	5	-	4	-	5	-	4	-	4	-	-	-	-	
4	4	4	3	4	3	4	3	3	3	3	3	2	2	3	3	3	2	3	3	4	-	5	-	-	-	3	-	3	4	-	-	4	3	-	-	-	-	
5	4	3	4	3	3	4	3	3	3	2	3	3	3	3	5	3	3	3	4	4	-	4	3	-	-	-	4	3	-	4	-	4	-	4	-	-	-	
6	3	4	3	4	4	4	4	4	5	3	5	4	3	4	5	4	3	4	4	-	5	-	5	5	-	4	-	3	5	-	4	-	-	-	-	-		
7	5	4	5	5	3	4	3	3	3	3	4	4	3	5	5	3	3	3	4	4	-	5	5	-	5	-	5	-	5	-	-	-	4	5	-	-	-	
8	5	5	3	3	3	3	5	3	2	3	3	2	3	4	4	3	2	3	4	3	-	4	-	-	4	-	4	-	5	4	-	-	5	4	-	-	-	
9	4	4	3	5	3	4	3	3	3	5	3	3	4	4	4	3	3	4	4	4	5	-	-	-	-	4	-	3	-	-	5	5	-	4	-	-	-	
0	4	4	4	5	5	4	5	5	5	3	4	5	5	5	5	4	5	4	5	-	-	5	5	-	-	5	5	-	5	5	-	5	-	-	-	-		

Resultados obtenidos manualmente:

Tabla 20. Caso de estudio. Estudiantes. Resultados manuales.

Estudiante	Líneas y promedios obtenidos
1	1(3.91), 2(4.15), 3(3.84), 4(3.81), 5(3.63), 6(3.5), 7(3.86), 8(4.5)
2	7(3.6), 8(4.5)
3	8(3.5)
4	8(3.75)
5	8(3.5)
6	8(3.75)
7	5(3.53), 6(3.5), 7(3.73), 8(4)
8	8(4)
9	8(3.75)
0	4(3.62), 5(3.73), 7(3.78), 8(4.25)

Resultados obtenidos por el sistema:

Tabla 21. Caso de estudio. Estudiantes. Resultados del sistema.

Estudiante	Líneas y promedios obtenidos
1	1(3.91), 2(4.15), 3(3.84), 4(3.81), 5(3.63), 6(3.5), 7(3.86), 8(4.5)
2	7(3.6), 8(4.5)
3	8(3.5)
4	8(3.75)
5	8(3.5)
6	8(3.75)
7	5(3.53), 6(3.5), 7(3.73), 8(4)
8	8(4)
9	8(3.75)
0	4(3.62), 5(3.73), 7(3.78), 8(4.25)

Análisis de los resultados

Al comparar los resultados no se obtienen diferencias por lo que se define que el sistema relaciona a los estudiantes con las líneas de investigación encontrando un total de 0 errores en el proceso.

Profesionales

A continuación, se muestran los 10 profesionales a analizar:

Tabla 22. Caso de estudio. Profesionales.

#	Dpto Área	Doc o MSc o Dr	AE	Temática 1	Temática 2
1	CEGEL	Máster	9	Ciencias Empresariales	-
2	CEGEL	-	3	Gestión de Proyectos	Inteligencia Artificial
3	CB	-	2	Informática Educativa	Matemática y Física Aplicada a la Computación
4	CB	-	6	Informática Jurídica	Web Semántica
5	CB	Doctor	13	Explotación de Minas, Metalúrgica	Gestión de Proyectos
6	CB	Doctor	12	Matemática y Física Aplicada a la Computación	Informática Educativa
7	CB	-	2	Informática Médica	Informática Educativa
8	Rectorado	Máster	8	Matemática y Física Aplicada a la Computación	Informática Educativa
9	CEIGE	Máster	10	Ciencias Empresariales	Gestión de Proyectos
10	CB	-	2	Informática Educativa	Inteligencia Artificial

Tabla 23. Caso de estudio. Profesionales. Resultados manuales.

Resultados manuales	
Profesional	Líneas
1	7
2	3
3	8
4	7
5	2
6	2
7	7
8	2
9	5
10	8

Tabla 24. Caso de estudio. Profesionales. Resultados del sistema.

Resultados del sistema	
Profesional	Líneas
1	7
2	3
3	8
4	7
5	2
6	2
7	7
8	2
9	5
10	8

Análisis de los resultados

Al comparar los resultados no se obtienen diferencias por lo que se define que el sistema relaciona a los profesionales con las líneas de investigación encontrando un total de 0 errores en el proceso.

3.3 Beneficios tangibles e intangibles

Los beneficios de informatizar un proceso se manifiestan de diferentes formas, pueden ser económicos y de orden social, sin que uno sea más importante que otro. El sistema propuesto está dirigido al sector educacional y además no es un producto comercial, por lo que los beneficios son de orden social.

El sistema propuesto contribuye a facilitar la conformación de equipos de investigación en la universidad de ciencias informáticas y su principal objetivo es resolver los problemas relacionados con dicho proceso. Los beneficios de su uso pueden ser tangibles o intangibles, los primeros son los únicos que se pueden tocar o percibir de manera precisa.

Beneficios tangibles

- Se utilizan tecnologías y herramientas libres y gratuitas, por lo que no se incurre en gastos de licencias.

Beneficios intangibles

- Incremento de la eficiencia en el proceso de confección de los equipos.
- Incremento de tiempo del personal encargado de la planificación docente para realizar otras funciones del Vicedecanato docente.
- El sistema propuesto permite obtener la relación existente entre las líneas de investigación y los individuos, por lo que existe una notable reducción del tiempo destinado a la conformación de los equipos.

En la siguiente tabla se compara la media del tiempo empleado manualmente por 2 personas y el tiempo del sistema para procesar la información de diez estudiantes y diez profesionales. Es importante resaltar que las prestaciones del cómputo donde se realizaron estas pruebas son:

-Procesador: AMD A8-7410 (4 CPUs), ~2.2GHz.

-Memoria: 8192MB RAM.

Tabla 25. Ahorro en tiempo.

Criterios	Clasificación de estudiantes y profesionales		Ahorro
	Manualmente	Con el sistema	
Tiempo	3 h	1 s	2:59:59 h

- Incremento en la satisfacción del personal dedicado al proceso docente educativo al facilitar la ejecución de una tarea que suele ser lenta, compleja y tediosa.

3.4 Conclusiones del capítulo

En el capítulo anteriormente abordado, se crearon los estándares de codificación el cual proporcionó un buen entendimiento del código y una mejor organización del mismo. Las pruebas de aceptación y de caja blanca efectuadas facilitaron detectar, documentar y corregir las no conformidades existentes en el sistema implementado. La realización del caso de estudio y los conocimientos de los beneficios tanto tangibles como intangibles, evidenciaron la efectividad de la solución presentada obteniendo un software capaz de relacionar tanto a los estudiantes como a los profesionales a las distintas líneas de investigación.

Conclusiones

Con la investigación, se obtuvo una herramienta para dar solución al problema que interviene en la creación de equipos conformados por estudiantes y profesionales partiendo de las líneas de investigación. En función de estos resultados se arribó a las siguientes conclusiones:

- La definición del marco teórico referencial de la investigación relacionado con el proceso de selección de personal para conformar equipos de investigación, fundamentaron la necesidad de desarrollar una aplicación que se adapte a los objetivos expuestos y satisfaga las necesidades de la universidad.
- Al utilizar técnicas de la ingeniería de software se realizó el modelo del sistema y se definieron los respectivos artefactos que se tienen en cuenta en la metodología utilizada para una mejor comprensión de la definición de la propuesta de solución.
- Se desarrolló la primera versión de un sistema la cual facilita el proceso de conformación de equipos partiendo de las líneas de investigación, utilizando la programación por satisfacción de restricciones, que no requiere de usuarios con mucha experiencia en esta labor.
- Las pruebas aplicadas para la verificación de la solución informática y la valoración de los resultados a través de un caso de estudio, demostraron que el sistema cumple con los requisitos definidos, garantizando su correcto funcionamiento y sus aportes a la sociedad.

Recomendaciones

Teniendo en cuenta los resultados obtenidos en la realización del presente trabajo de diploma se recomienda:

- ✓ Realizar una mejora de la solución implementada de manera tal que, al ejecutar las operaciones pertinentes, disminuya el tiempo de respuesta.
- ✓ Incluir al sistema las respectivas funcionalidades que permitan clasificar a los estudiantes por sus datos demográficos con el objetivo de incrementar la efectividad de los resultados.
- ✓ Estandarizar el sistema de forma tal que pueda ser empleado por otras organizaciones y puedan modificar el motor de inferencias de acuerdo a lo necesitado.
- ✓ Mostrar otros tipos de gráficas para incrementar el apoyo visual del usuario.

Referencias Bibliográficas

Altamira . 2017. Altamira Recruiting. [En línea] 1 de marzo de 2017. [Citado el: 1 de marzo de 2017.] <http://www.altamirahrm.com/es/software/software-de-reclutamiento>.

Ampuero, Margarita André. 2009. *Un modelo para la asignación de recursos humanos a equipos de proyectos de software*. Ciudad de La Habana, Cuba : s.n., 2009.

Ayala, Alejandro Peña. 2006. *Sistemas basados en conocimiento: una base para su concepción y desarrollo*. 2006.

Beck, Kent. 2000. *Extreme programming explained: embrace change*. [En línea] Addison-Wesley Professional. , 2000. [Citado el: 28 de enero de 2017.] <http://books.google.es/books?hl=es&lr=&id=G8EL4H4vf7UC&oi=fnd&pg=PR13&dq=Extreme+Programming+Explained&ots=j9vFtsgXyl&sig=Xz6T5Ne01wTeLnPskTctYLBSTdo>.

Ble, Carlos. 2010. *Diseño ágil con TDD*. 2010.

Bustínduy, Jesús Basterrechea y Murillo, Joan Gómez. 2004. *Configuración de un equipo de proyecto informático eficiente*. s.l. : Universitat Oberta de Catalunya TFC: Sistemes d'informació. Gestió de projectes- Memoria, 2004.

Cárdenas, Rodrigo Salomón Mendoza y Zárate, Jorge Nicasio Cáceres. 2016. *Sistema experto para la selección de postulantes en puestos de una agencia bancaria usando la metodología CommonKads*. Lima : Facultad de Ingeniería de Sistemas, Pregrado, Universidad Nacional Mayor De San Marcos. Lima- Perú, 2016.

Césari, Matilde. 2012. *Sistema Basado en Reglas*. [En línea] 2012. [Citado el: 2 de febrero de 2017.] <http://es.scribd.com/doc/109492619/SISTEMAS-BASADOS-REGLAS..>

Duque, Raúl González. 2011. *Python para todos*. [En línea] 2011. [Citado el: 2 de marzo de 2017.] <http://dspace.universia.net/handle/2024/919> .

Font, José María. 2008. *Generación de Sistemas Basados en Reglas mediante Programación Genética*. Madrid : Universidad Politécnica de Madrid, 2008.

G, Kendall. 2001. *Introduction to Artificial Intelligence*. s.l. : University of Nottingham, 2001.

Gelabert, M Porret. 2010. *Gestión de personas. Manual para la gestión del capital humano en las organizaciones*. s.l. : Cuarta ed. España: ESIC editorial, 2010.

GIL, Arturo. 2015. Técnicas de Análisis de Datos en WEKA. *ingeniería*. [En línea] 2015. [Citado el: 15 de febrero de 2017.] isa.umh.es/asignaturas/crss/tutorialWEKA.pdf..

J, Giarratano y G., Riley. 2001. *Sistemas Expertos: Principios y Programación*. s.l. : Thomson Editores, 2001.

2016. JetBrains. *PyCharm*. [En línea] 2016. [Citado el: 3 de febrero de 2016.] <https://www.jetbrains.com/pycharm/>.

Joskowicz, José. 2008. *Reglas y prácticas en eXtreme Programming*. s.l. : Universidad de Vigo, 2008.

Krishnamoorthy. 1996. *Artificial Intelligence and Expert Systems for Engineers*. s.l. : Developments in Experts Systems, 1996.

Larman, Craig. 2005. *UML y Patrones. Una introducción al análisis y el diseño orientado a objetos y al proceso unificado*. India : Pearson Education, 2005.

Letelier, Patricio. 2006. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). [En línea] 2006. [Citado el: 27 de enero de 2017.] http://www.cyta.com.ar/ta0502/b_v5n2a1.htm.

Lio, Dr.Daniel Gálvez. 1998. Curso de Sistemas Basados en Conocimiento. 1998.

Mizrahi, Alejandro Iglesias. 2016. *Desarrollo del Sistema de Recomendación de equipos de investigación para tesis de grado*. La Habana, Cuba : s.n., 2016.

Momjian, Bruce. 2001. *PostgreSQL: Introduction and Concepts*. . New York : Addison Wesley, 2001.

Navarro, Yeselky Vigoa. 2010. *TEAMSOF+ : Sistema para la gestión del trabajo en equipo en el desarrollo de proyectos de software. Módulo de tratamiento estudiantil*. Ciudad de la habana, Cuba : s.n., 2010.

PELÁEZ, JUAN. 2009. Arquitectura basada en capas. [En línea] mayo de 2009. [Citado el: 10 de marzo de 2017.] <http://www.juanpelaez.com/geek-stuff/arquitectura/arquitectura-basada-en-capas/>.

Pressman, Roger S. 2005. *Ingeniería de software. Un enfoque práctico. Capítulo 8 Modelado de Análisis*. s.l. : Palgrave Macmillan, 2005.

Rodríguez, A, Hernández, J y Placido, A. 2006. *Ingeniería del Conocimiento: Extracción del Conocimiento*. s.l. : Facultad de Informática. Universidad de las Palmas de Gran Canaria, 2006.

Sommerville, Ian. 2015. *Ingeniería del Software. 9na Edición.* Madrid : Pearson Educación, S.A, 2015.

Started, Getting. 2010. Software Design Tools for Agile Teams, with UML, BPMN and More. . [En línea] 2010. [Citado el: 3 de marzo de 2017.] <http://www.visual-paradigm.com/>.

Talent4jobs. 2017. Talent4jobs. [En línea] 1 de marzo de 2017. [Citado el: 1 de marzo de 2017.] <http://dealer-ti.com/software-reclutamiento/>.

Vargas, A E Pérez. 2011. *Desarrollo de un prototipo de sistema experto para el apoyo en la toma de decisiones del proceso de selección de personal.* Medellín : Trabajo de Investigación, Universidad EAFIT Escuela de Ingeniería. Departamento de Informática y Sistemas, Medellín, Colombia, 2011.