

Universidad de las Ciencias Informáticas

Facultad 3



Herramienta informática para la formación de equipos de trabajo en ambiente académico- profesional (SIFE_AP)

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

Autores:

Leidy Rosa Bacerio Gómez

Jessica Mercedes León Sánchez

Tutores:

MsC. Elizabeth Rodríguez Stiven

MsC. Yalice Gámez Batista

DrC. Yoan Martínez Márquez

Ciudad de La Habana, 23 de junio de 2017.

Agradecimientos

Primero agradecer a la persona cuyo nombre he mencionado más en este último curso: mi compañera de tesis, estoy muy contenta de que hayas sido tú la persona que me acompañó en esta travesía. Gracias por la paciencia y la tolerancia en los momentos de tensión, gracias por tu amistad transparente y sincera, pero sobre todo gracias por aquella vez que dijiste: "Lo tuyo es mío", y aún más gracias por cumplirlo.

Quiero agradecer:

A mi madre, por mostrarme que el amor puede llegar hasta el infinito...y mucho más. Por enseñarme que el amor te vuelve paciente, aunque no sueles serlo y guerrero sin tener armadura. Gracias mamita por ser realmente incondicional conmigo.

A mi padre, por todas las conversaciones, por hablarme de la libertad, por "presentarme a Harry Potter", por forjarme mujer sincera.

A mis tutores por su tiempo y dedicación, gracias por su formación como profesionales que ha permitido llevar a cabo esta investigación. Cuentan con mi admiración.

A Jonathan, por ser mi contención, por ser paciente, por ser mi amigo, mi compañero, mi hermano, por ser todo lo que quiero de un novio. Gracias por "el amor en los tiempos de tesis"

A mi loquita favorita: Aídil, gracias por ser mi payaso cuando estoy triste, por tus regaños cuando me equivoco, por saber leer en mí lo que oculto y por entenderme cuando nadie más lo hace.

A mis tías Sonia, Rosy, Olguita y Carí, y a mis primos Jose y Yaillet por su preocupación constante, su ayuda y sus buenos deseos hacia mí.

A mis primos Javier y Jasiel, por cuidarme desde pequeña, por dejarme jugar pelota, bolas, trompos y todo lo que hacían junto a sus amigos.

A mi tía Baby, a mi padrino y a mi madrina, por todo su esfuerzo para ayudarnos a mí y a mamá, porque a pesar de la distancia han sido parte de este proceso.

A mis amigos de toda la vida: Fabio, China y Abdiel, sé que en estos últimos 5 años no hemos pasado mucho tiempo juntos, pero son tan míos que no hay momento de mi vida que me olvide de ustedes.

A mis amigos del barrio, en especial a Claudia, Frank, Franki y Clarita por su apoyo desde el día 1 que comencé la UCI.

A Samu, mi hermanito, por tu ayuda incondicional en cualquier situación, por ser mi cocinero, mi mecánico, mi compañero de trabajo y mi amigo.

A mis compañeros de años, a todos los quiero y recordaré con cariño, en especial a Luis Javier por ayudarme a descubrir lo que era ser universitaria, a Adolfo por los largos debates llenos de consejos e historias, a Jose Carlos por no rendirse conmigo con la programación, a Felix por su elegancia, su diplomacia y el arte que tiene para dialogar, A Julio por el bueno equipo de estudio que fuimos y por tener paciencia para todas mis locuras, a Yoe por su disposición a ayudar siempre, a Asy por su cariño a Lissy por los buenos momentos.

A Nicolau por estar siempre disponible para ayudarme y por sus buenos deseos para mí.

A Flavio por su ayuda constante y el tiempo dedicado a nosotras.

A personas que, aunque ya no estén en la universidad fueron un gran apoyo para mí como Juniel, Ebrik, Eiger y Yassel.

Gracias a todas las personas que de alguna forma u otra contribuyeron a mi formación como profesional en estos cinco años en la universidad.

Agradecimientos

Primero agradecer a mi compañera de tesis, por haberme aguantado en todo este curso y tener paciencia conmigo. Por esa bonita amistad que se ha formado entre las dos, que espero mantener.

Quiero agradecer a mi familia, principalmente a mi mamá que siempre me ha apoyado en todo y me ha dado fuerzas cuando me he sentido insegura. A mi abuelita del alma que siempre cuida de mí y nunca se queja de nada. A mis tutores, por haberme apoyado durante todo el trabajo y ser comprensivos y nunca decirnos que no. A todos los profesores que me han dado clases. A mis compañeros de grupo con los que he compartido estos 5 años.

Jessica León Sánchez.

Dedicatoria

Quiero dedicar este trabajo a mis padres que siempre me han apoyado en todo lo que necesito. A todas mis amistades y mis compañeros de aula.

Jessica León Sánchez.

A mi abuelo, por cada beso en la frente que me dio cuando me observaba con un libre.

Leidy R Bacerio Gómez

Declaración de Autoría

Declaro ser el autor de este trabajo y autorizo a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Leidy Rosa Bacerio Gómez
Autor

Jessica Mercedes León Sánchez
Autor

MsC. Elizabeth Rodríguez Stiven
Tutor

Dr. C Yoan Martínez Márquez
Tutor

MsC. Yalice Gámez Batista
Tutor

Resumen

La Universidad de la Ciencias Informáticas tiene como característica distintiva la vinculación entre la producción de software y el ámbito académico, dos esferas donde el trabajo en equipo propicia una mejora en los resultados de los futuros profesionales. Sin embargo, la formación de dichos equipos de trabajo carece de una fundamentación teórica que respalde la manera en que se forman y de una herramienta que facilite la composición de los equipos teniendo en cuenta más de un solo indicador. La presente investigación propone una herramienta que forme equipos de trabajo teniendo en cuenta cuatro factores: el tamaño de equipo, el aspecto social, la expectativa de éxito en la producción y el desempeño docente.

Para llevar a cabo el trabajo se realiza un diseño que cumple con los parámetros establecidos por la arquitectura y se emplean las herramientas y tecnologías seleccionadas para la implementación. Además, se realiza un período de pruebas que validan el funcionamiento del módulo asegurándose que el mismo cumpla con las normas vigentes para el desarrollo y lograr el objetivo propuesto.

La implementación de esta herramienta brinda la posibilidad a los profesores de la universidad de contar con un sistema útil que confeccione equipos de trabajo con habilidades a desempeñar en un ámbito académico y profesional.

Palabras claves: ambiente académico, ambiente profesional, equipos de trabajo.

Abstract

The integration between the academic and the software development spheres is one of the most distinguishing characteristics of the University of Information Sciences. The work in teams is crucial in these two spheres to promote the improvement in the results of the professionals to be. However, forming these teams lacks of theoretical foundations to support the way students are gathered and there isn't a tool in order to consider more than one criterion at the time to form teams. This research aims at developing a tool to form teams of work considering four criteria: team's size, social approach, successful expectation towards software development and academic performance.

A design is made to carry out the solution considering the established standards of architecture. Selected tools and technologies were used for the development process. Besides, a test strategy is followed that helps to value the correct functioning of the tool in order to fulfill current standards in the software development and achieve the objective assumed in this research.

A tool is offered to professors in order to help them forming teams of work in an academic and professional environment.

Keywords: Academic environment, professional environment, work teams.

Índice

INTRODUCCIÓN.....	1
Capítulo 1: Marco teórico	6
1.1 Conceptos Básicos	6
1.2 Importancia de la formación de equipos de trabajo en ambientes académico-profesional	8
1.3 Elementos a considerar en la formación de los equipos en ambiente académico-profesional.....	9
1.4 Análisis de las herramientas y modelos existentes relacionadas con la formación de equipos	17
1.5 Herramientas y tecnologías para la implementación del sistema	21
1.6 Conclusiones parciales	24
Capítulo 2. Propuesta de solución	26
2.1 Modelo conceptual para la formación de equipos de trabajo en ambiente académico-profesional.	26
2.2 Requisitos de la herramienta	38
2.2.1 Requisitos funcionales	38
2.2.2 Requisitos no funcionales	39
2.3 Historias de usuario	40
2.4 Plan de iteraciones	40
2.5 Plan de entregas	41
2.6 Arquitectura de solución	41
2.7 Modelo de diseño.....	43
2.8 Patrones de diseño.....	44
2.8.1 Patrones GRASP	45
2.8.2 Patrones GoF	46
2.9 Modelo de datos.....	47
2.10 Estándar de codificación.....	48
2.11 Conclusiones parciales	50
Capítulo 3. Verificación de la viabilidad de la investigación:	51
3.1 Prueba de aceptación	52
3.2 Pruebas de unidad.....	54
3.2.1 Pruebas de caja blanca	54
3.3 Valoración del aporte práctico de la investigación.....	57
3.5 Conclusiones parciales	62
Conclusiones Generales	63
Recomendaciones.....	64
Referencias Bibliográficas	65

Índice de Tabla

Índice de Tabla.....	9
Tabla 1. Perfil ideal para tamaño de equipo. Elaboración propia	10
Tabla 2. Comparación de técnicas sociométricas. Elaboración propia	13
Tabla 3. Comparación de modelos y metodología. Elaboración propia.	21
Tabla 3. Historia de usuario: Formar equipo	40
Tabla 4 Plan de iteraciones. Elaboración propia	41
Tabla 5 Plan de entrega. Elaboración propia.....	41
Tabla 7. Caso de prueba: Formar equipos.....	53
Tabla 8. Juego de datos de prueba: Formar equipo (V: válido, I: inválido).....	53
Tabla 9. Rango de valores para el cálculo del coeficiente de concordancia	59
Tabla 10. Cálculo de la suma de rangos	59
Tabla 11. Cálculo de factor de corrección	60

Índice de figuras

Índice de figuras.....	10
Figura 1. Modelo conceptual de formación de equipos.....	27
Figura 2. Etapa caracterización del grupo.....	27
Figura 3. Recopilación de información.....	28
Figura 4. Fase 2: Caracterización del grupo.....	30
Figura 6. Procesamiento de datos.....	31
Figura 7. Formación de equipos.....	31
Figura 8. Pseudocódigo del algoritmo.....	32
Figura 9. Estudiante estrella.....	34
Figura 10. Segundo estudiante.....	34
Figura 11. Tercer estudiante.....	35
Figura 12. Cuarto estudiante.....	36
Figura 13. Quinto estudiante.....	37
Figura 14. Equipos formados.....	38
Figura 16. Modelo de datos.....	48
Figura 17. Estándar de codificación.....	49
Figura 18. Estándar de codificación.....	49
Figura 19. Estándar de codificación.....	49
Figura 20. Estándar de codificación.....	50
Figura 21. Estándar de codificación.....	50
Figura 22. Estrategia para verificar la viabilidad de la solución.....	52
Figura 23. Resultados de las pruebas de aceptación.....	54
Figura 24. Ejemplo caja blanca.....	56
Figura 25. Grafo.....	56
Figura 26. Suma de rango de valores asociados a los criterios de los expertos. Elaboración propia...61	61

INTRODUCCIÓN

La informática es uno de los más grandes logros del hombre. Desde su surgimiento, la sociedad ha experimentado numerosos cambios. A raíz del avance tecnológico hoy en día existen instituciones dedicadas a la producción de software, las cuales han utilizado la informática para el perfeccionamiento de la producción y su desarrollo, convirtiéndose la producción de software en uno de los principales renglones de la economía de muchos países (Acuña & Juristo, 2004).

El desarrollo de software exige que los ingenieros de software trabajen en equipos, aunque realizando tareas interdependientes con relaciones complejas. Por este motivo, los equipos deben planificar su proyecto, hacer un seguimiento de su progreso y coordinar su trabajo. Además, deben ponerse de acuerdo en sus objetivos, tener un método de trabajo común, comunicarse libre y frecuentemente creando un clima de trabajo adecuado para llevar a cabo su actividad.

Cada día crece la tendencia hacia el trabajo en forma colaborativa para alcanzar metas comunes. Cuando se integran aspectos colaborativos a un proceso determinado se busca una mejora en la comunicación, una mayor participación y un fuerte compromiso entre los integrantes de un grupo permitiendo una mayor productividad y una mejor calidad del producto final. Esto cobra vital importancia en la industria del software en la que deben insertarse los estudiantes una vez egresan de la especialidad de Ciencias Informáticas, debido a la necesidad de un equipo multidisciplinario capaz de llevar a feliz término cada una de las fases del desarrollo del software.

La Universidad de las Ciencias Informáticas (UCI) es una universidad insertada en la Industria Cubana del Software, que basa su formación en un modelo de integración formación-producción-investigación, a partir de la vinculación estudio-trabajo. Es por ello considerada una “universidad productiva”, concepto definido por Castro (2004) para señalar las características de la UCI y diferenciarla de las clásicas relación universidad-empresa en este campo, donde los parques tecnológicos han tenido una marcada relevancia (Castro, 2004).

La UCI cuenta con un modelo de formación el cual se conforma de dos ciclos: básico y profesional. Esto significa que los estudiantes al llegar al 3er año de la carrera ingresan a los proyectos de producción de software, donde es esencial el trabajo colaborativo para obtener resultados. Sin embargo, trabajar en equipo no solo es útil en proyectos productivos, sino también en cualquier ámbito donde los individuos tengan intereses en común, y así sucede en la docencia donde los estudiantes tienen que vencer las barreras de la comunicación de las relaciones sociales, el rigor de estudio que se exige y las distintas habilidades técnicas a desempeñar en la carrera.

Para esto, los profesores forman equipos de trabajo de estudiantes que realizan las distintas tareas correspondientes a las asignaturas que imparten. Se basan en información recogida mediante

diagnósticos iniciales de las materias, test sociométricos, notas académicas del curso anterior, reconocimiento de las características psicopedagógicas del estudiante por parte del profesor después de las cuatro primeras semanas del curso y, otras veces, de manera aleatoria o a preferencia de los estudiantes, puesto que resulta un trabajo complejo la formación de equipos de trabajo de forma manual y teniendo en cuenta más de un indicador .

Por otra parte, se analizaron informes semestrales de la facultad 3, los cuales reflejan un insuficiente aprovechamiento de los diagnósticos aplicados a los estudiantes. Esto se debe a la variedad y los diferentes niveles de dificultad que estos presentan. En una entrevista abierta realizada a los profesores de la facultad 3, con el objetivo de conocer acerca del proceso de formación de equipos, se constató que el mismo resulta difícil de realizar a partir de los datos que expresan los diagnósticos, teniendo en cuenta el mayor número de variables que influyen para la formación de dichos equipos. Los entrevistados coinciden en que cada uno prioriza la variable sobre la que más información logran obtener o simplemente solicitan a los estudiantes que formen equipos de trabajo por afinidad.

Surgiendo como:

Situación problemática:

- Carencia de un análisis multivariable de los datos extraídos de los diagnósticos psicopedagógicos aplicados a los estudiantes.
- La composición del equipo no contribuye a explotar las potencialidades favorablemente de los estudiantes.
- Insuficiente aprovechamiento de la predisposición positiva de los estudiantes en su desempeño profesional.

Esto impone el reto de combinar las diferentes técnicas sociométricas profesionales de formación de equipos de desarrollo de software y las variables de rendimiento docente, para formar equipos de trabajo desde la docencia, que permitan dotarlos de herramientas de colaboración con el fin de superar las barreras culturales y de comunicación haciendo crecer la confianza y la comodidad de los miembros.

La aplicación de cada una de estas técnicas resulta de una elevada complejidad teniendo en cuenta el número de variables que se deben tener en cuenta, y las posibles combinaciones de estudiantes para la formación de los equipos. Es por ello que se identifica como problema a resolver:

Problema a resolver:

¿Cómo contribuir a la formación de equipos de trabajo en ambiente académico-profesional a partir del procesamiento de los datos del diagnóstico aplicado a los estudiantes?

Objeto de estudio:

Formación de equipos de trabajos en ambiente académico-profesional.

Campo de acción:

Herramientas informáticas para la formación de equipo de trabajos en ambiente académico-profesional.

Objetivo general:

Desarrollar una herramienta informática que contribuya a la formación de equipos de trabajo en ambiente académico-profesional a partir de los resultados del diagnóstico psicopedagógico realizado a los estudiantes.

Objetivos específicos:

1. Elaborar el marco teórico de la investigación, mediante el estudio de los referentes teóricos asociados a la formación de equipos.
2. Elaborar un modelo teórico para la formación de equipos de trabajo en ambiente académico-profesional basado en indicadores psicopedagógicos.
3. Desarrollar una herramienta informática que soporte el modelo teórico para la formación de equipos de trabajo en ambiente académico-profesional.
4. Valorar la viabilidad de la solución mediante pruebas de funcionalidad y validación del valor teórico de la investigación.

Posibles resultados:

Una herramienta informática para la formación de equipos de trabajo en ambiente académico-profesional, basada en indicadores psicopedagógicos.

Idea a defender

Desarrollando una herramienta informática para el procesamiento de los datos del diagnóstico, se contribuirá a formar equipos de trabajo en ambiente académico-profesional ajustados a las características de los estudiantes.

Para el cumplimiento de la presente investigación se combinan los siguientes métodos y técnicas en la búsqueda y procesamiento de la información:

Métodos teóricos:

Método analítico-sintético: el empleo de este método permitió realizar un estudio y análisis de toda la bibliografía consultada determinando la importancia de la formación de equipos de trabajo en ambiente académico-profesional.

Método histórico-lógico: se utilizó este método para conocer los antecedentes y las tendencias actuales referidas a la formación de equipos de trabajo en ambiente académico-profesional.

Modelación: se utilizó este método para representar el objeto a investigar, permitiendo obtener como resultado un modelo que media entre el sujeto y el objeto real que ha sido modelado.

Métodos Empíricos:

La observación: a través de la misma se recopilaron datos sobre los mecanismos de formación de equipos en la facultad 3.

La encuesta: se emplea como una técnica de adquisición de información a través de los cuestionarios que se realizan a la hora de validar el nivel de aceptación de la herramienta, donde se podrá conocer la opinión o valoración de los sujetos seleccionados a realizar dicha encuesta.

La entrevista: Se utilizó para llevar a cabo el levantamiento de requisitos de la aplicación y para validar la efectividad de la misma mediante las pruebas de aceptación.

La medición: Para obtener información numérica del objeto, a partir de la cual se harán estimaciones y comparaciones cuantitativas; además del análisis y las consideraciones de las características prevalecientes y relevantes del fenómeno en estudio.

El presente trabajo se encuentra estructurado en tres capítulos, como se detalla a continuación:

Capítulo 1: Fundamentación teórica:

Este primer capítulo tiene como objetivo describir el marco teórico de la investigación, realizando un análisis de cuáles son las herramientas, modelos existentes para formar equipos, también se analizará la necesidad existente en Universidad de las Ciencias Informáticas de la formación de equipos de trabajo para el trabajo en conjunto. Además, se lleva a cabo una caracterización de las diferentes dimensiones a tener en cuenta para la formación de equipos y se hace alusión a las diferentes herramientas, técnicas y modelos existentes a utilizar en el desarrollo del sistema a implementar.

Capítulo 2: Propuesta de solución:

En este capítulo se presenta la propuesta de solución al problema de investigación planteado, la cual está dividida en dos partes: la creación de un modelo teórico para la formación de equipos de trabajo y el desarrollo de una herramienta que soporte al modelo. Se explica el alcance de la herramienta que se va a desarrollar. Establece las principales características y funcionalidades que tendrá la solución, describiendo los principales conceptos del entorno que serán objeto de análisis para la realización de la herramienta. Posteriormente se hace el levantamiento de requisitos funcionales y no funcionales para darle cumplimiento al objetivo general de la investigación.

Capítulo 3: En el capítulo se aborda sobre las actividades que se llevan a cabo durante la fase de implementación y pruebas. Además, se diseñan y aplican las pruebas para comprobar el correcto funcionamiento de la herramienta y valorar el aporte práctico de la investigación.

Capítulo 1: Marco teórico

Este primer capítulo tiene como objetivo describir el marco teórico de la investigación, abordando los principales conceptos utilizados y resaltando la importancia del trabajo de equipo en un ambiente académico-profesional. También se realiza un análisis de los elementos a tener en cuenta en la formación de equipos de trabajo en este tipo de entorno. Además, se hace un estudio de las herramientas y modelos existentes que se relacionan con la formación de equipos de trabajo. Finalmente se hará alusión a las herramientas y técnicas existentes a utilizar en el desarrollo del sistema a implementar.

1.1 Conceptos Básicos

A continuación, se explicarán los conceptos fundamentales de la investigación.

Equipos de trabajo

En la amplia literatura relacionada con el tema, cada autor ofrece su propia definición de equipo de trabajo. Según Katzenbach un equipo es un número pequeño de personas, con habilidades complementarias, que están comprometidas con un propósito, objetivos de rendimiento y enfoque común, en el que todos son responsables de los resultados obtenidos (Katzenbach, 2000). Humphrey agrega que cada uno tiene asignado roles o funciones específicas que debe desempeñar, donde completar la misión requiere alguna forma de dependencia entre los miembros (Humphrey, 2000).

Sobre la misma idea, un artículo opina acerca del trabajo en equipo que el mismo es una modalidad de articular las actividades laborales de un grupo humano en torno a un conjunto de fines, de metas y de resultados a alcanzar. El trabajo en equipo implica una interdependencia activa entre los integrantes de un grupo que comparten y asumen una misión de trabajo (Covey, 2000).

Por otra parte aunque se suele denominar “equipos” a muchas variedades de esfuerzos grupales, grandes o pequeños, pocas de estas iniciativas merecen ese nombre si se mide al verdadero equipo por el desarrollo superior obtenido (Katzenbach, 2000).

Los equipos de trabajo en las actividades del software pueden entrar en juego en cualquier número de tareas específicas: desarrollo y revisión de requisitos del proyecto, desarrollo de la arquitectura y las directrices del diseño del proyecto, definición de aspectos del entorno técnico, desarrollo de los estándares de codificación del proyecto, coordinar el trabajo en las partes relacionadas del proyecto, diseño de las partes complicadas del sistema, revisión del diseño y código de los desarrolladores individuales, mantenimiento del software una vez construido entre otras (Pressman, 2006).

Para ámbitos dedicados al desarrollo del software Stiven define que un proyecto informático es un conjunto pequeño de personas, enfocado en tareas específicas, con habilidades, actitudes y conocimientos complementarios, comprometidos con un propósito y objetivo común; consciente que la única forma de completar la misión es trabajar en conjunto (Stiven E. , 2012) .Aunque este concepto de equipo está dirigido a proyectos informáticos, se adapta a la presente investigación, puesto que la misma tiene como escenario la Universidad de las Ciencias Informáticas donde se forjan futuros ingenieros informáticos que desarrollen proyectos informáticos desde su etapa de estudiante hasta que lleguen a ser profesionales del desarrollo del software. Por esta razón las autoras asumen el concepto de equipo definido por Stiven.

Ambiente académico

El ambiente académico remite al escenario donde existen y se desarrollan condiciones favorables de aprendizaje. Un espacio y un tiempo en movimiento, donde los participantes desarrollan capacidades, competencias, habilidades y valores. (Duarte, 2003).

Se define el ambiente académico como el “conjunto de factores objetivos y subjetivos que interactúan e influyen sobre el organismo de una persona en el desarrollo del proceso educativo y que contribuyen de forma decisiva a la conservación y fortalecimiento del estado de salud y a su formación general integral” (Sord, 2005).

Ambiente profesional

El ambiente profesional es la percepción que tienen los trabajadores de las propiedades del ambiente del centro en el cual laboran, que influye en el comportamiento colectivo e individual (Furnham, 2002).En este mismo sentido Peiró plantea que el clima laboral está fundado en un cierto nivel de acuerdo en la forma de percibir el ambiente, que coincide con la visión sociocognitiva de las organizaciones”, (Peiró, 1998).

Ambiente académico-profesional

Al analizar los conceptos de ambiente académico y ambiente profesional, se concluye que para la siguiente investigación un ambiente académico-profesional es el escenario donde desarrollarán las habilidades necesarias que van a favorecer al enriquecimiento de valores, competencias, que van a ser explotadas en un ambiente productivo. Con el objetivo de lograr profesionales mejor preparados para desempeñar cualquier tarea asignada del perfil del profesional del ingeniero en ciencias informáticas.

1.2 Importancia de la formación de equipos de trabajo en ambientes académico-profesional

Trabajar en equipo representa una ventaja para cualquier organización. Las empresas modernas se centran cada vez más en la idea del equipo. Las consultoras proponen reestructurar la organización para facilitar el trabajo en equipos, los directores elaboran políticas que destacan la importancia de los equipos dentro de la organización y los empresarios más veteranos recomiendan a su personal que promuevan el equipo en sus departamentos. El equipo se ve como la unidad básica de la vida laboral (Hayes, 2003).

Según los 4 aprendizajes fundamentales de la UNESCO (Delors, 1995) que son aprender a ser, aprender a conocer, aprender a vivir juntos y aprender a hacer, el trabajo en equipo va ayudar a desarrollar una serie de habilidades, donde cada integrante de un equipo aprenderá a desarrollar su propia personalidad, adquirir instrumentos de comprensión, podrá influir sobre su propio entorno para participar y cooperar con los demás. Todo esto con el objetivo de lograr que cada persona tenga capacidad de juicio, de autonomía y responsabilidad personal, aprender a conocer el entorno, la comprensión mutua y lo más importante a trabajar en equipo.

En un artículo que trata sobre el diseño de una estrategia didáctica de trabajo en equipo en función del aprendizaje, la autora plantea que en el ámbito educativo la adquisición de conocimientos se asimila mejor si se hace mediante técnicas de trabajo en equipos planificadas y aplicadas en el aula con los estudiantes (Flores, 2015). Esta misma autora opina que el trabajo en equipo representa una estrategia didáctica que tiene como propósito la mejora de la enseñanza en las aulas. En este mismo ámbito la investigación de (Alonso, Vázquez, Blanco, & Ortiz, 2015) manifiesta que los estudiantes muestran una actitud positiva al trabajar en equipo como herramienta de enseñanza-aprendizaje. Resulta beneficioso el aprendizaje colaborativo para que los estudiantes comprendan mejor los conceptos estudiados, desarrollen la habilidad para exponer su trabajo al resto de los compañeros y refuercen estrategias de negociación, adaptación y capacidad de solucionar problemas.

Otro dato de interés es lo expresado en un artículo sobre el desarrollo de las competencias de trabajo en equipo dentro del proceso de enseñanza-aprendizaje de la programación en el nivel superior. El mismo plantea que cuando se emplea el trabajo colaborativo en el ámbito de la enseñanza -aprendizaje de programación, en combinación con las metodologías empleadas en la industria de desarrollo de software, obtenemos los beneficios pedagógicos derivados del desarrollo de habilidades, se incentiva el pensamiento de alto nivel, se incrementa la retención de los alumnos (menor deserción), se desarrollan habilidades de comunicación oral y de interacción social. No alcanza con asignar solamente tareas para realizar en grupo de estudiantes, sino que el proceso de enseñar y aprender ocurre dentro de verdaderos equipos de desarrollo, donde los roles están bien definidos (Aubin, y otros, 2015).

1.3 Elementos a considerar en la formación de los equipos en ambiente académico-profesional

La formación de equipos de trabajo es un tema que cada día cobra más importancia, puesto que el éxito de cualquier entidad está dado por los recursos humanos y por el capital intelectual que estos poseen (Stiven E. R., 2012). A continuación, se realiza un estudio de los factores a tener en cuenta para la formación de equipos de trabajo en ambiente académico-profesional.

Análisis de investigaciones relacionadas con el tamaño de los equipos en ambiente académico-profesional

No existe una definición consistente de qué constituye un tamaño ideal para un equipo efectivo en la literatura estudiada. El tamaño del equipo siempre ha sido un asunto polémico, pues cada investigador tiene sus propios criterios al respecto (Stiven E. R., 2012). En este epígrafe se realiza un análisis de los trabajos que describen de manera general el rango en que se debe enmarcar el tamaño de los equipos para lograr un mejor desempeño de los mismos

En trabajos relacionados con el tamaño de equipo en ambiente académico, se coincide que el tamaño es una variable que depende del contexto, pero que afecta de manera significativa tanto al proceso como a la efectividad del grupo (Diez, Zárraga-Rodríguez, & García, 2013). Prácticamente todos los autores revisados coinciden en que los equipos tienen que tener un tamaño pequeño, para poder optimizar todo su potencial, cuyo número fluctúa entre 5 y 7 miembros (Duarte, 2003).

William G. Dyer en su libro *Team Building* propone factores claves para lograr equipos exitosos, en este estudio Dyer plantea que un equipo pequeño involucra más porque las acciones de cada uno de sus miembros son visibles y tienen impacto todo el tiempo (Dyer, 2007). Expresa además que el tamaño del grupo está inversamente relacionado con el desempeño, esto se debe a que la coordinación, supervisión y planificación del trabajo es mucho más difícil de lograr en equipos grandes. Con este mismo enfoque Millhiser y Solow en sus estudios sobre como varía el rendimiento en los equipos con replazo, investigan sobre cuándo es necesario dividir un equipo de trabajo. En este sentido afirman que el rendimiento de un equipo disminuye a medida que el tamaño de este aumenta (Millhiser & Solow, D, 2003). En concordancia, Hare en su investigación sobre la teoría general de grupos muestra particular interés en el tamaño de los equipos, considerando que los equipos pequeños fluctúan entre tres y seis miembros mientras que los grandes oscilan entre ocho y doce miembros (Hare, 2003). Este criterio es compartido por Belbin en sus investigaciones, quien considera que el equipo ideal consta de seis miembros (Gonzalez, 2006).

En trabajos relacionados con la actividad del software, los autores coinciden en que los equipos productivos son precisamente los que ostentan un tamaño pequeño. En el estudio realizado en (Jiang,

2007), sobre los factores que influyen en la productividad de los equipos de desarrollo de software, se analizaron 4106 proyectos de software llevados a cabo de 1995 al 2005 para probar la hipótesis de que “los equipos muy grandes en el desarrollo de software tienden a ser ineficientes” por lo que consideran que el tamaño del equipo ideal debe oscilar en el rango de tres a cinco miembros. En concordancia Prieto en su modelo de comparación de equipo considera que el tamaño debe encontrarse en el mismo rango (Prieto M. , 2010).

En general, las definiciones revisadas coinciden en que los equipos tienen que tener un tamaño pequeño, para poder optimizar todo su potencial. En el desarrollo de software es necesario mantener vigilado el tamaño del equipo, como han demostrado (Jiang, 2007), si el equipo es muy grande puede afectar la productividad del grupo. Algunos estudiosos opinan que el tamaño óptimo de un equipo es supervisado por la relación entre la información, la complejidad y el dominio de los conocimientos necesarios para el buen éxito de las tareas (Sim & Heng, 2006) .

En concreto la investigación sobre el tamaño de equipo en ambiente docente conduce a dos criterios, el primero es que son preferibles los grupos con número impar de miembros y que los compuestos por 2-5 miembros desarrollan un buen trabajo al ejercitar los mejores elementos de los grupos pequeños y grandes. Pues son lo suficientemente grandes como para formar una mayoría.

Tabla 1. Perfil ideal para tamaño de equipo. Elaboración propia

Autores	Duarte 2003	Martínez 2010	Hare 2003	González 2006	Prieto M 2010	Jiang 2007
Ambiente	Docente		Genérico		Software	
Tamaño	5-7	2-5	3-6 (pequeños) 8-12 (grande)	6	3-5	3-5

Como se puede observar pese a que definir un tamaño ideal para un equipo es una tarea compleja donde intervienen diferentes variables la mayoría de los autores dentro y fuera del ámbito docente y productivo coinciden en que el tamaño del equipo debe estar entre 2-5 personas. El incremento del tamaño provoca una disminución de la productividad, problemas de coordinación, supervisión y planificación del trabajo.

Análisis de investigaciones relacionadas con las relaciones interpersonales entre los miembros de los equipos en ambiente académico-profesional.

Dado que las fuerzas motrices de cada entidad están constituidas por los recursos humanos de la misma, se hace necesario que entre los mismos exista un ambiente de trabajo agradable y socialmente estable. A continuación, se realiza un análisis bibliográfico donde varios autores aportan su criterio sobre la influencia de las buenas relaciones interpersonales de los miembros de un equipo y el desempeño del mismo.

En un estudio realizado sobre la formación de equipos de trabajo, García plantea que un factor clave en el desempeño de un equipo en un entorno académico está dado por el ambiente social que se genere en el aula, puesto que esto mejora las actividades docentes, la satisfacción personal y el desempeño de los estudiantes (García M. , 2000). Siguiendo esta misma idea Barcelona resalta que la empatía que pueda existir entre los candidatos es esencial para el funcionamiento del equipo, dado que de nada vale que estén cualificados para las tareas, si a la hora de iniciar los trabajos, no hay entre ellos un mínimo de armonía y buena relación (Barcelona, 2016).

Por otra parte, dos de los saberes de la UNESCO (Delors, 1995) están comprometidos a la idea del trabajo de forma colaborativa, lo cual se refleja en el saber denominado "Aprender a vivir juntos", que significa aprender a vivir juntos implicando comprensión mutua entre los individuos en un entorno caracterizado por la paz, y en el saber llamado "Aprender a hacer", que significa influir sobre el propio entorno implicando saber trabajar en equipo. En este mismo sentido el artículo sobre el trabajo en equipo plantea que el mismo se caracteriza por la comunicación fluida entre las personas, basada en relaciones de confianza y de apoyo mutuo.

Autores como (Quintero, 2010), (Stiven E. R., 2012), (Arza, 2013), y (Ampuero, 2015), realizaron investigaciones en la UCI, relacionadas con los equipos de proyectos informáticos donde se destaca la importancia de las relaciones interpersonales de los miembros de un equipo a partir del desarrollo de competencias genéricas como son la responsabilidad social y compromiso ciudadano, la capacidad de comunicación y la capacidad de trabajar en equipo.

Según se puede observar, el comportamiento humano dentro de las entidades es imprescindible debido a que se origina en necesidades y sistemas de valores muy arraigadas en las personas. Por este motivo varios autores han utilizado el status social basado en las relaciones interpersonales de un grupo para la obtención de mejores resultados (Aubin, y otros, 2015). Mediante técnicas referentes a la sociometría es posible conocer y caracterizar el comportamiento social entre individuos, teniendo en cuenta la apatía y la preferencia entre los mismos. Muchas investigaciones se han concentrado en la utilización de estas técnicas en el ámbito docente con el objetivo de manipular el carácter social de un grupo de individuos para potenciar de manera positiva los resultados del mismo.

En un estudio sobre el uso del sociograma para determinar las relaciones existentes en un aula, realizado por (Moreno, 2004), en el cual plantea que la aplicación de un sociograma en un grupo escolar determinado permite al docente conocer la manera en que se relaciona socialmente los miembros del mismo, a la vez que puede tomar conciencia de los beneficios e inconvenientes que generan éstas interacciones entre cada uno de los alumnos de manera individual. En este mismo sentido un artículo sobre la utilidad del sociograma como herramienta para el análisis de las interacciones grupales da a conocer que el test sociométrico y el sociograma muestran las posibles diferencias que se pueden encontrar en una estructura grupal (Pineda & Renero, 2009).

Otra opinión interesante para la investigación, es la que se manifiesta en el artículo "Herramientas ", el cual expone que el test sociométrico es útil para el profesor dado que a través de la misma se puede generar un clima idóneo para el aprendizaje y bienestar general en el aula (Kuz & Falco, 2013). Además esta técnica sociométrica mejora las relaciones sociales dentro de la clase y favorece la integración de los alumnos aislados y rechazados (Kuz & Falco, 2013).

En la investigación sobre la aplicación del test sociométrico como herramienta para el diagnóstico de grupo en apoyo a la tutoría académica, utilizan el psicograma para obtener información sobre la aceptación o rechazo de cada estudiante (Vega, 2013).

Otra investigación revisada utiliza el psicodrama para alcanzar una visión más amplia del contexto existencial, puesto que en el momento dramático se pierde el dualismo entre fantasía y realidad que en ese mismo instante coexisten, representando mediante el drama los problemas sociales que posee el grupo, donde cada miembro interpreta un rol real y/o fantaseado. La investigación se enmarcó en ambiente educativo con el objetivo de que los estudiantes tomaran conciencia de los problemas sociales que se representaban mediante el psicodrama, de esta manera, los mismo proponían soluciones positivas para el problema social optado por nuevas actitudes sociales (García, González, Morales, & Magán, 2000).

En la Tabla 2, que aparece a continuación se realiza un análisis sobre las técnicas sociométricas estudiadas teniendo en cuenta aspectos tales como la evaluación de las relaciones interpersonales dentro de un grupo, el ambiente donde son aplicadas, la usabilidad y con qué objetivo fue creado.

Tabla 2. Comparación de técnicas sociométricas. Elaboración propia

Técnicas sociométricas	Evaluación de las relaciones interpersonales	Ambiente en que se manifiestan	Aplicación	Usabilidad
Sociograma	Si	Docente	Microcosmos individual	Si
Test sociométrico	Si	Docente	Relaciones entre grupos	Si
Psicograma	Si	Docente	Medir sentimientos de aceptación y rechazo en un grupo	Si
Psicodrama	Si	Docente	Cociente cultural	Si

Valorando las características y los factores que tienen en cuenta las técnicas sociométricas analizadas se evidenció que el test sociométrico es el más factible para la investigación, pues es un test que además de cumplir con las características planteadas, ha sido utilizado en ambientes docentes. La información que proporciona su aplicación resulta valiosa para la formación de los equipos. El hecho de que cada persona y el resto del equipo trabajen con sus preferidos, ayuda a incrementar el entendimiento y permite que logren cumplir con los objetivos propuestos.

Conjuntamente, resulta muy favorable que las personas del equipo puedan trabajar en armonía para que puedan desempeñar y desarrollar las funciones y habilidades donde con mayor probabilidad pueden alcanzar el éxito, ya que esto contribuye a elevar la motivación de los miembros y los resultados del proyecto.

Análisis de investigaciones relacionadas con la docencia en la formación de equipos en ambiente académico-profesional.

Cuando se quiere alcanzar una meta, es fundamental la posesión de conocimientos que permitan el desarrollo de actividades dirigidas a lograr el objetivo deseado.

Para un mejor entendimiento de por qué en la siguiente investigación la docencia es otro factor importante en la formación de equipos se hace referencia a la tesis de Maestría (García M. , 2000).

Los miembros del equipo deben estar bien cualificados para realizar su trabajo y tener deseo de cooperar. Además, solo pueden trabajar juntos como equipo, después que todos los integrantes conocen los papeles de los otros con quienes interactúan. Cuando existe este entendimiento de

inmediato comienzan a actuar como un equipo basado en las exigencias de la situación, sin esperar que alguien dé una orden. Es decir, los miembros del equipo responden voluntariamente a los requisitos del trabajo y llevan a cabo cuanto se necesita para alcanzar las metas del equipo y sobre todo las necesidades que dé la organización (García M. , 2000).

Como parte de la investigación se consultaron investigaciones realizadas en la UCI por los autores (Stiven E. R., 2012), (Ampuero, 2015), (Quintero, 2010) y (Arza, 2013), en las cuales se definieron una serie de competencias que abogaban al desarrollo de habilidades y del conocimiento por parte de los estudiantes como son la capacidad de abstracción, análisis y síntesis, la capacidad de aprender y actualizarse permanentemente, la capacidad de aplicar los conocimientos en la práctica, capacidad para identificar, plantear y resolver problemas, habilidades para buscar, procesar y analizar información procedente de fuentes diversas, la capacidad de investigación y la capacidad para tomar decisiones. La misma idea se manifiesta en un artículo sobre el trabajo en equipo el cual caracteriza al mismo como un medio para coordinar las habilidades humanas y generar con acuerdo respuestas rápidas a problemas cambiantes y específicos (Delors, 1995).

En relación con lo anterior Belbin en su modelo de la personalidad para el trabajo en equipo, define una categoría compuesta por tres roles mentales, resaltando la importancia de que en un equipo existan miembros con un alto desarrollo de habilidades y especialidades que aporten pensamientos inteligentes para la elaboración de determinadas actividades que realicen los equipos de trabajo (Belbin M. , 2010). La misma idea se ve reflejada en los cuatro saberes de la UNESCO, específicamente en el denominado "Aprender a conocer", el cual hace referencia a la adquisición de instrumentos para lo comprensión. Este saber implica placer al comprender y descubrir nuevos conocimientos aprendiendo a conocer el entorno (Delors, 1995).

Contribuyendo con la idea de la necesidad de tener en cuenta aspectos docente para la asignación de los miembros al equipo, se consultó el artículo titulado " Metodología para ubicar estudiantes en roles del proceso del desarrollo de software". El mismo revela que es importante analizar la información de los resultados académicos del estudiante en función de las competencias de cada rol a desempeñar dentro de un equipo para definir su ubicación y su plan de formación una vez ubicados (Arza-Pérez, Verdecia-Martínez, & Lavandero-García, 2013).

Por lo tanto, en la presente investigación otro indicador importante a tener en cuenta como rasgo distintivo del estudiante para su integración en un equipo de trabajo, es la categoría docente del mismo. Dicho indicador responde a una evaluación del comportamiento del estudiante en el ámbito docente que es realizada y/u otorgada por el profesor, y puede tomar valor de:

- M (3 a 3,5): estudiante en pésimas condiciones (aptitudes y actitudes).

- R (3,5 a 4): estudiante con condiciones regulares (que se defienden solo en algunas esferas).
- B (4 a 4,5): estudiante con condiciones aceptables (que se defienden en varias esferas y presentan buenas aptitudes y actitudes).
- E (4,5 a 5): estudiante con condiciones excelentes (en todas las esferas).

El modo o los aspectos que cada profesor utilice o tengan presente para evaluar a sus estudiantes es propio de ellos y el sistema trabajará con los resultados de dicha evaluación.

La escala presentada para medir el desempeño docente es la utilizada en la UCI, en evaluaciones como los cortes evaluativos de las asignaturas, los diagnósticos de conocimientos y en asambleas propias de los estudiantes universitarios.

Modelo de personalidad

A continuación, se hace referencia a la tesis de maestría de la Msc. Elizabeth Rodríguez Stiven. En la misma, la autora realiza un análisis de los modelos de personalidad existentes, que son aplicados en ambientes productivos, llegando a concluir que el más factible para la investigación es el Modelo de Belbin, pues es un modelo libre, fácil de utilizar y ha sido ampliamente utilizado en el ámbito del software (Stiven E. , 2012). A continuación, se explica en qué consiste el modelo de Belbin.

Modelo de Belbin

El modelo de Belbin permite identificar los roles de equipo preferidos y evitados de cada persona. Meredith Belbin define rol de equipo como "nuestra particular tendencia a comportarnos, contribuir y relacionarnos socialmente" e identifica nueve roles clasificados en tres categorías (Belbin, 2010):

- Roles Mentales: Cerebro (CE), Monitor-Evaluador (ME) y Especialista (ES)
- Roles de Acción: Impulsor (IS), Implementador (ID) y Finalizador (FI)
- Roles Sociales: Coordinador (CO), Investigador de Recursos (IR) y Cohesionador (CH)

Las categorías representan el grado de orientación de las personas hacia el desempeño de tareas (roles de acción), hacia el mundo de las ideas (roles mentales) o hacia las relaciones con las personas (roles sociales). Acorde a la teoría de Belbin (Belbin, 2010) en un equipo deben estar representados todos los roles, deben predominar los roles de acción y se debe lograr un balance entre los roles mentales y sociales (Basri, 2008). Sin embargo, no siempre los equipos llegan a tener nueve miembros es por ello que en equipos pequeños los miembros pueden y lo hacen asumir más de un rol. Tras el análisis de equipos existentes y su rendimiento o comportamiento el uso de algunos conceptos relacionados con la teoría de Belbin pueden conducir a algunas mejoras (Chapman, 2011). Por ejemplo:

- Un bajo rendimiento del equipo demanda de un buen coordinador o finalizador.
- La existencia de conflictos requiere de un fuerte coordinador.
- Rendimiento mediocre requiere un investigador de recursos o cerebro.
- Equipos propensos al error necesitan de un monitor–evaluador.

Diferentes roles son importantes en diferentes circunstancias, por ejemplo:

- Un nuevo equipo necesita un fuerte cohesionador
- Situaciones competitivas demandan un Cerebro con buenas ideas
- En áreas de riesgo un buen monitor evaluador puede ser muy necesario.

Este modelo ha sido utilizado en investigaciones vinculadas al software para evaluar el impacto de diferentes roles de Belbin en el equipo de proyecto (Thomas, 1999). En (Prieto M. , 2010) se propone un modelo en donde se utilizan los perfiles propuestos por la autora para determinar la diversidad del equipo, evaluar la personalidad de sus miembros y probar que equipos con perfiles similares pueden realizar el mismo trabajo de forma eficaz con tecnología diferente. Mientras (Rajendran, 2005) publicó un estudio realizado sobre equipos de trabajo de desarrollo de software en el que concluye que los perfiles de Belbin bien balanceados contribuyen a la efectividad del equipo, que es la hipótesis fundamental de Belbin. En (Quintero, 2010) los perfiles de Belbin son utilizados como evidencias en la evaluación de competencias genéricas.

En (André, 2009) se utilizan los roles para definir patrones que contribuyen a la formación de equipos de proyectos de software, esta autora plantea que un equipo bien formado debe contar con los siguiente tres patrones: se requiere la presencia de todas las categorías de roles de Belbin, con predominio en la preferencia por los roles de acción sobre los roles mentales y de los mentales sobre los sociales; se requiere al menos un miembro del equipo con preferencia por el rol Cerebro (CE); al asignar el Jefe de proyecto se debe verificar que esta persona tenga preferencia por uno de los siguientes roles de Belbin: Impulsor (IS) y Coordinador (CO).

La herramienta diseñada por el propio Belbin para la validación de su teoría es un test que se encuentra a disposición de todos: el Cuestionario de Autopercepción de Belbin (SPI) (nombrado así por sus siglas en inglés), el cual ha sido ampliamente utilizado por disímiles autores en diversos campos. El derecho de autor del test pertenece al autor Meredith Belbin definido en (Belbin, 2010) donde se establece que cualquier persona puede realizar el cuestionario publicado en su libro.

El uso de estos cuatro factores va dirigido a que existe una necesidad de conseguir un adecuado equilibrio en los equipos de trabajo, nos permite centrarnos en las investigaciones que han realizado los teóricos acerca del equilibrio de roles, para lograr que los grupos trabajen en equipo. Algunos investigadores que han testado la denominada hipótesis del "equilibrio de los roles" (Gonzalez, 2006), la cual afirma que los equipos que contengan mayor cantidad de roles, desempeñaran mejor la tarea (serán más efectivos) que aquellos que contengan menor cantidad de roles, han constatado esta efectividad a través de sus trabajos empíricos. Las investigaciones demostraron que la efectividad de los equipos estaba relacionada con la existencia de una variedad alta de roles, es decir, con la representación de todos los roles de equipo.

Es indispensable que cada persona ocupe un rol dentro del equipo, que viene dado por su experiencia y capacidades personales. Cada uno de estos roles, tiene definido objetivos, actividades, interacción con otros roles, herramientas a utilizar y un plan de trabajo, de acuerdo a la metodología empleada. Es posible que no se requieran todos los roles durante el desarrollo de un software. Por ejemplo, el desarrollo de un sistema de información de gran tamaño requerirá más roles que uno de menor tamaño (HUMPHREY, 2004).

1.4 Análisis de las herramientas y modelos existentes relacionadas con la formación de equipos

A continuación, se realiza un análisis de las herramientas y modelos existen relacionadas con la formación de equipos.

Algoritmo para formar equipos (Dickey & Olaosebikan, 2016).

Una empresa llamada Gigster crea un servicio que emplea una amplia red de desarrolladores, diseñadores y gerentes de productos independientes, utiliza su propio algoritmo personalizado (llamado "Karma") para reunir a estos completos extraños en cuestión de minutos para lanzar proyectos complejos. Aquí está un rápido desglose de cómo funciona Gigster (y, por extensión, Karma):

Digamos que tienes una idea para una nueva aplicación, pero necesitas ayuda para convertir tu esbozo de servilleta en un prototipo funcional. Introduce su idea en el sitio web de Gigster y en ningún momento, tiene una propuesta de proyecto completo y cotización generada por su motor de precios personalizados. Una vez que un cliente firma, Karma identifica a los profesionales independientes (o Gigsters, como les gusta llamarse a sí mismos) - fuera de la red de más de mil freelancers remotos - para hacer el trabajo. La pregunta sigue siendo: ¿Cómo un algoritmo sostiene a la gente "correcta" para reunir para un proyecto? Aquí radica la magia (o la ciencia, en realidad). Karma clasifica a cada

freelancer en la comunidad basándose en tres criterios principales: el rendimiento (cuántos proyectos ha completado con éxito), la logística (si los Gigsters están en la misma zona horaria, lo que podría afectar la velocidad de entrega del proyecto) y si (han trabajado juntos en proyectos anteriores). Después de explorar la red buscando personas con el nivel de habilidad apropiado, basado en el tamaño y el alcance del proyecto, Karma busca a personas con clasificaciones similares, ya que estas clasificaciones son un indicador de que ciertas personas podrían trabajar mejor juntas sobre otras.

Después de analizar las características se concluye que el algoritmo (Karma) no aporta a la investigación puesto que ha sido usado en ambiente profesional, además de formar los equipos sin tener en cuenta ninguna de las dimensiones propuestas anteriormente.

Herramienta para formar equipos de trabajo

La herramienta consultada en el artículo (Team Building spanish, 2007), es una herramienta para formar equipos de trabajo, la cual ha sido diseñada para dar apoyo a la publicación sobre control del cáncer: "Aplicación de los conocimientos, Guía de la OMS para desarrollar programas eficaces". Un paso crucial para la puesta en práctica eficaz de planes y programas oncológicos es la creación de su equipo, así como el equipo clínico multidisciplinar para la gestión del diagnóstico, tratamiento y cuidados paliativos de los pacientes.

Esta herramienta es genérica y, por lo tanto, puede ser adaptada, no sólo para el cáncer, sino también para otras situaciones y programas en los que el equipo juegue un papel fundamental.

La herramienta creada por la OMS (Organización Mundial de la Salud) no aporta a la investigación puesto de que ha sido usada en la Salud, aunque con la característica de ser genérica y poder adaptarse a cualquier entorno, dicha herramienta solo forma equipos teniendo en cuenta un solo indicador. En la investigación actual, los equipos serán formados teniendo en cuenta 3 dimensiones (social, psicopedagógica y expectativa de éxito).

Herramientas sociométricas

Existen varias herramientas de software para el análisis de redes sociales, que son aplicaciones de plataformas independientes, fácilmente accesibles, usadas para realizar investigaciones en redes sociales, aplicándose a diferentes fuentes de datos, para obtener diferentes niveles de análisis y trabajar con los test sociométricos.

NodeXL: Es una herramienta que es un complemento para Excel 2007 que permite visualizar y analizar redes de grafos (nodos), y fue creada por el equipo del sociólogo Marc Smith en el Microsoft Research

y colaboradores de las universidades de Maryland y Washington. Para su usabilidad deberá instalarse el templete opensource para Microsoft Excel 2007 para obtener y visualizar el grafo correspondiente en un ambiente familiar como es una ventana de Excel.

Classroom Sociometrics: es una herramienta especializada de software, aplicada a la escuela, que permite darnos a través de sus gráficas un sentido de status social de los alumnos en la escuela, así como en forma el clima global del aula. Es propiedad de Walsh's Classroom Sociometrics y fue creado en 2009 en Illinois. A través de este software se puede realizar la nominación sociométrica. Permite también juntar información y mostrarla gráficamente a través de grafos, además de crear y modificar sociogramas. Las gráficas que modela esta herramienta presentan diferencias significativas: distingue entre varón (mediante un cuadrado) y mujer (mediante un círculo) y presenta el porcentaje de relaciones bidireccionales.

Agent SocialMetric: es un sistema integrador online que devela las percepciones de los estudiantes acerca de las relaciones interpersonales en un aula de clases, permite el análisis de la interacción existente entre los estudiantes y el docente dentro del marco temático, posibilitándole al docente la toma de decisiones pedagógicas precisas.

A pesar de que las herramientas sociométricas estudiadas no formen equipos, las mismas brindan información que resulta útil para la investigación, debido a que se logra entender la personalidad y los modos de comportamiento preferidos de los individuos.

Modelo para la formación de equipos de proyectos de software (Ampuero, 2015).

El modelo responde a un problema de optimización combinatorio multiobjetivo planteando cuatro funciones objetivas y doce tipos de restricciones. Para dar solución al problema se implementaron algoritmos metaheurísticos y para tratar las restricciones que incluye el modelo se implementó la estrategia de rechazo.

El modelo (Ampuero) estudiado a pesar de que ha sido utilizado ambiente profesional no aporta a la investigación actual, debido a que lo que busca la autora es optimizar una función.

Modelo para la evaluación de la composición de equipos de proyectos informáticos (Stiven E. , 2012).

El modelo propuesto por la autora para la evaluación de la composición puede ser utilizado en el proceso de desarrollo de los equipos de proyectos informáticos con el objetivo de incluir las

experiencias obtenidas de sus resultados en los futuros planes y estrategias de desarrollo a fin de aumentar las posibilidades de éxito de los equipos.

El modelo (Stiven) estudiado a pesar de no ser usado para la formación de equipos ya que es aplicado para mejorar el por ciento de evaluaciones correctas y el esfuerzo requerido para realizar la evaluación de equipo, el mismo ayuda a definir un tamaño ideal de equipo.

Modelo de trabajo en equipo para el ámbito docente (IMO, 2013).

El modelo siguiente se basa en el modelo integrado de efectividad del trabajo en equipo, IMO, desarrollado para entornos organizacionales no académicos por (MATHIEU, MAYNARD, RAPP, & GILSON, 2008). El modelo sugerido por estos autores representa el trabajo en equipo como un conjunto de procesos (*Mediators*) condicionados y/o influidos por unos factores previos o de entrada (*Inputs*) y que dan lugar a unos resultados (*Outcomes*). Además, incorpora el carácter cíclico y dinámico de estos procesos de forma que los resultados (*outcomes*) se convierten en nuevos *inputs* para el proceso. Este modelo ha servido de base en los últimos años para diferentes revisiones del trabajo en equipo, siendo aceptado en el ámbito académico como marco conceptual válido y de alto valor (JACA, 2011).

Con el estudio del modelo (IMO) se llega a la conclusión, que además de ser adaptado para poder utilizarse en un ambiente docente, no es utilizado para formar equipo, puesto a que solo se usa para evaluar el desempeño de los equipos de trabajo.

Metodología para ubicar estudiantes en roles del proceso de desarrollo de software (Arza, 2013).

La autora propone una metodología que, basada en las tendencias actuales de soluciones a procesos de selección de personal, brinda un esquema de fases y actividades a realizar para la ubicación de un estudiante en un rol del proceso de desarrollo de software, sustentado en un modelo que integra técnicas de inteligencia artificial para obtener los criterios para la toma de decisiones a partir de los elementos que del estudiante y los roles se conocen, teniendo en cuenta la imprecisión y la incertidumbre que está presente en estos procesos.

La metodología (Arza) estudiada no aporta a la investigación, a pesar de utilizar una dimensión que es usada en nuestra investigación, la finalidad de la misma no es formar equipos. La metodología es usada para la ubicación de estudiantes en roles profesionales, la cual define un conjunto de fases que están en correspondencia con las entradas, salidas y componentes del modelo.

En la Tabla 3 se realiza un análisis comparativo de los modelos y metodologías estudiados teniendo en cuenta aspectos tales como su finalidad: si son usados para formar equipos, el ambiente al que va

dirigido: si es docente o profesional, y si cumplen con el objetivo de la investigación que es formar equipos de trabajo.

Tabla 3. Comparación de modelos y metodología. Elaboración propia.

Herramientas y modelos	Finalidad	Ambiente	Proponen algoritmo	Objetivo de la investigación
Algoritmo (Karma)	Formar equipos	Productivo	Si	No
Herramienta (OMS)	Formar equipos	Salud	No	No
NodeXL	Conocer status social de un aula	Docente	No	No
Classroom Sociometrics	Conocer status social de un aula	Docente	No	No
Agent SocialMetric	Conocer status social de un aula	Docente	No	No
Modelo (Ampuero)	Optimización de competencias	Docente	Si	No
Modelo (Stivens)	Evaluar al equipo	Productivo	No	No
Modelo (IMO)	Evaluar desempeño de equipo	Docente	No	No
Metodología (Arza)	Ubicar estudiantes a roles	Docente	No	No

Después de analizar las características de dichos modelos se concluye que no aportan a la investigación, a pesar de existir dos modelos que proponen un algoritmo para formar equipos, los mismos no tienen en cuenta las dimensiones propuestas en la investigación actual para formarlos.

1.5 Herramientas y tecnologías para la implementación del sistema

A continuación se describen las metodologías, herramientas y tecnologías utilizadas para realizar el análisis, diseño e implementación del software.

Metodología de Desarrollo de Software para la solución del sistema

Para guiar el proceso de desarrollo de un software es imprescindible el uso de una metodología, que sea capaz de guiar al equipo de trabajo con el objetivo de aumentar la calidad y eficiencia en el proceso de creación de software.

En la presente investigación se decidió utilizar una metodología ágil, debido a que la solución propuesta representa un software pequeño, de corta duración y con un reducido número de miembros. Atendiendo al corto período de tiempo, y a la cantidad de integrantes del proyecto (conformado solo por dos programadores y un cliente), se determina utilizar la metodología Extreme Programming (XP). Ya que la misma permite estructurar, planear y controlar el proceso de desarrollo del producto a implementar (Perissé, 2006).

La metodología XP responde a las necesidades de la investigación y a las condiciones de trabajo. Esta metodología permite incrementar considerablemente la productividad del equipo de desarrollo, conseguir con más facilidad la satisfacción del cliente, mantener los estándares de calidad y cumplir los plazos con mayor exactitud y consistencia.

Lenguaje de modelado UML

En el presente trabajo se hace uso del lenguaje UML, en su versión 2.0. Se utiliza este lenguaje de modelado ya que podemos tener una orientación en cuanto al orden de las actividades de un equipo, ayudando esto a dirigir las tareas de los desarrolladores individuales y del equipo en conjunto, además de ofrecer criterios para el seguimiento y la medición de los productos y las actividades de un proyecto.

Herramienta CASE

Visual Paradigm es una herramienta CASE de modelado visual que facilita la construcción de artefactos en un proceso de desarrollo de software mediante el Lenguaje de Modelado Unificado (UML). Soporta el ciclo completo de desarrollo de software y permite su documentación en diferentes formatos, empleando UML como lenguaje de modelado. Teniendo en cuenta la existencia de una Licencia UCI para el uso de Visual Paradigm y las características de esta herramienta, para el desarrollo del presente trabajo se selecciona la misma en su versión 8.0, debido a que está diseñada para una amplia gama de usuarios, incluyendo los ingenieros de software, analistas de sistemas, analistas de negocios y arquitectos de sistemas, o para cualquier persona que esté interesada en la construcción de sistemas de software que utilizan un enfoque orientado a objetos. Además, es compatible con los últimos estándares de la notación UML (PARADIGM, 2005).

Lenguajes de programación

PHP 5: PHP, acrónimo de "PHP: *Hypertext Preprocessor*", es un lenguaje de 'scripting' de propósito general y de código abierto que está especialmente pensado para el desarrollo web y que puede ser embebido en páginas HTML. Su sintaxis recurre a C, Java y Perl, siendo así sencillo de aprender. La meta principal de este lenguaje es permitir a los desarrolladores web escribir dinámica y rápidamente páginas web generadas (Achour, 2011).

JavaScript: JavaScript es un lenguaje de script multiplataforma orientado a objetos. Es un lenguaje pequeño y ligero; no es útil como un lenguaje independiente, más bien está diseñado para una fácil incrustación en otros productos y aplicaciones, tales como los navegadores Web. Dentro de un entorno anfitrión, JavaScript puede ser conectado a los objetos de su entorno para proveer un control programable sobre éstos (Network, 2014).

Para la implementación del componente se escoge como el lenguaje PHP5 es simple para el desarrollo de aplicaciones web, pero que a su vez ofrece características avanzadas para programadores expertos. Por otra parte, el lenguaje JavaScript por su adaptabilidad y uso tanto desde el lado del cliente como del lado del servidor.

Framework

Symfony 2.8: *Symfony* es un marco de trabajo de desarrollo web para PHP. Permite a los desarrolladores construir y mantener más fáciles sitios web con PHP (Github, 2014). Está basado en la arquitectura Modelo-Vista-Controlador (MVC), surgió como un proyecto de software libre el cual se ha convertido en uno de los *framework* más populares de PHP que existe en la actualidad.

Para la siguiente investigación se hace uso el framework de desarrollo Symfony, ya que garantiza rapidez y confiabilidad a la hora de la creación de productos usando PHP. Es compatible con la mayoría de los gestores de bases de datos, como: PostgreSQL, Oracle, SQL Server. Además, debido a su destacado uso del patrón arquitectónico Modelo-Vista-Controlador que facilita la separación de estos elementos.

Bootstrap: es un framework CSS que permite dar forma a un sitio web mediante librerías CSS que incluyen tipografías, botones, cuadros, menús y otros elementos que pueden ser utilizados en cualquier sitio web. En la investigación actual las autoras deciden trabajar con Bootstrap debido a que es una excelente herramienta para crear interfaces de usuario limpias y totalmente adaptables a todo tipo de dispositivos y pantallas, sea cual sea su tamaño. Además, Bootstrap ofrece las herramientas necesarias para crear cualquier tipo de sitio web utilizando los estilos y elementos de sus librerías (Otto & Thornton, 2015).

Entorno de Desarrollo Integrado

NetBeans IDE: es un proyecto de código abierto con una gran base de usuarios y una comunidad en constante crecimiento. NetBeans IDE es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito para Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso (Corporation, 2011).

Este IDE es idóneo para el desarrollo del sistema ya que se usa fundamentalmente por ser multilinguaje, pero además por ser de código abierto y gratuito.

Sistema Gestor de Base de Datos

PostgreSQL 9.4: PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (Martinez, 2010).

PgAdmin III: PgAdmin cuenta con una rica administración en código abierto y desarrollo de plataforma para PostgreSQL. La aplicación puede utilizarse en sistemas operativos como Linux, FreeBSD, Solaris, Mac OSX y las plataformas de Windows para administrar PostgreSQL, así como las versiones comerciales y derivados de PostgreSQL como Postgres Plus Advanced Server y base de datos Greenplum.etc.

Se utiliza como gestor de bases de datos PostgreSQL debido a que admite un gran ilimitado de bases de datos. Mientras que la herramienta PgAdmin permite el trabajo con las bases de datos de forma sencilla, brindando facilidades en la escritura de consultas y el manejo de las bases de datos.

1.6 Conclusiones parciales

Como resultado del estudio del estado del arte de la formación de equipo de trabajo en ambiente académico profesional se concluye que:

- El trabajo en equipo es esencial para los profesionales informáticos, puesto que la producción de software exige el desempeño de diferentes roles que traigan consigo diferentes responsabilidades en un ambiente colaborativo para lograr un producto de calidad.

- Respecto a los factores a tener en cuenta para la formación de equipos de trabajo se determinó que el tamaño de los equipos no debe exceder de cinco miembros, entre los cuales debe existir un equilibrio de roles y un buen clima social para lograr un fin determinado.
- Aunque la bibliografía consultada aportó conocimiento valioso a la investigación ninguno de los trabajos revisados propone una herramienta para la formación de equipos de trabajo en ambiente académico-profesional teniendo en cuenta el tamaño del equipo.
- El estudio de las herramientas y tecnologías permitió seleccionar XP como metodología de desarrollo de software, UML 2.0 como lenguaje de modelado y Visual Paradigma 8.0 como herramienta CASE. Como lenguajes de programación PHP 5 y JavaScript; Symfony 2.8 y Bootstrap 3 como *framework* de desarrollo; NetBeans 8.0 como IDE de desarrollo.

Capítulo 2. Propuesta de solución

Para la construcción y desarrollo de un sistema informático, es de suma importancia la utilización de métodos y técnicas, que permitan la solución de problemas presentes a lo largo del ciclo de vida de un software. La correcta captura de requisitos y el modelamiento del sistema permiten que se mitiguen las fallas que puedan aparecer durante su desarrollo, además de establecer un entendimiento común entre lo que se desea y lo que se elabora. La implementación de la solución propuesta se caracteriza por basarse en los principios y reglas de la metodología XP, utilizando las tecnologías y herramientas definidas. La metodología empleada para la descripción del sistema se divide en fases y actividades dentro de estas, generándose un grupo de artefactos que permiten especificar los elementos fundamentales del sistema.

2.1 Modelo conceptual para la formación de equipos de trabajo en ambiente académico-profesional.

Para la formación de equipos de trabajo en ambiente académico-profesional se presenta un modelo que consta de dos partes fundamentales: Caracterización del grupo y formación de equipo. La entrada para la primera etapa está constituida por la lista de los estudiantes de un grupo docente, a los cuales se le asocia los datos referentes a las tres dimensiones que el modelo tiene en cuenta para la formación de equipos de trabajo. Estas dimensiones serán nombradas en la presente investigación como Psicopedagógica, Desempeño docente y Expectativa de éxito en la producción, haciendo referencia a las relaciones interpersonales, la evaluación docente y a la expectativa de éxito según la personalidad. Luego se procesa esta información, teniendo como salida de esta primera etapa la matriz de datos genéricos de los estudiantes. Otra salida serían los vectores de los estudiantes estrellas y de los estudiantes islas. La entrada de la segunda etapa está compuesta por los datos que conforman la salida de la etapa anterior, ya que son los datos necesarios para que se ejecute el algoritmo que conformará los equipos.

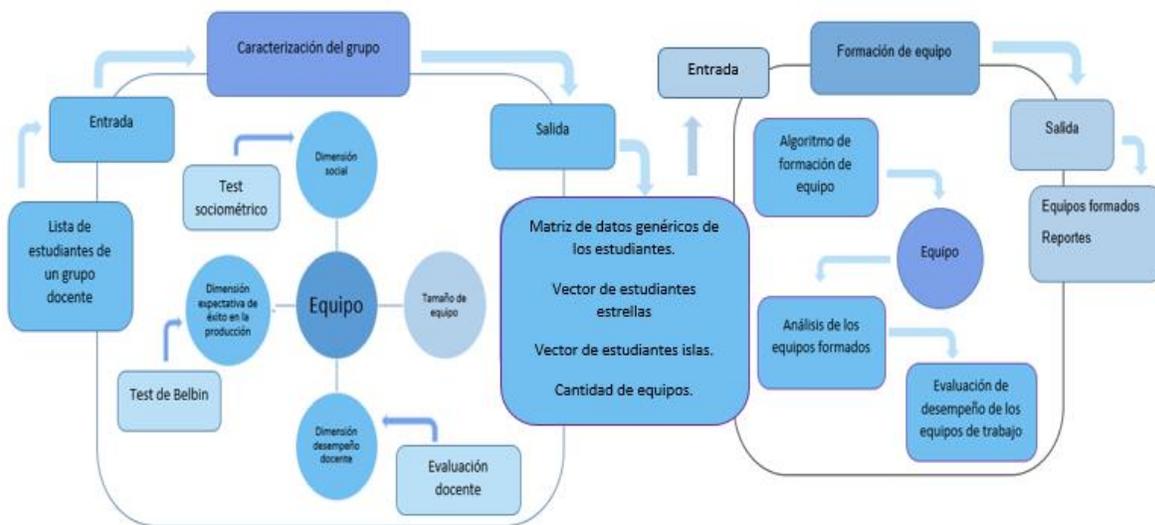


Figura 1. Modelo conceptual de formación de equipos.

Primera etapa: Caracterización del grupo

A continuación se explica en qué consiste la etapa: Caracterización del grupo, y se explicará cada una de sus fases.

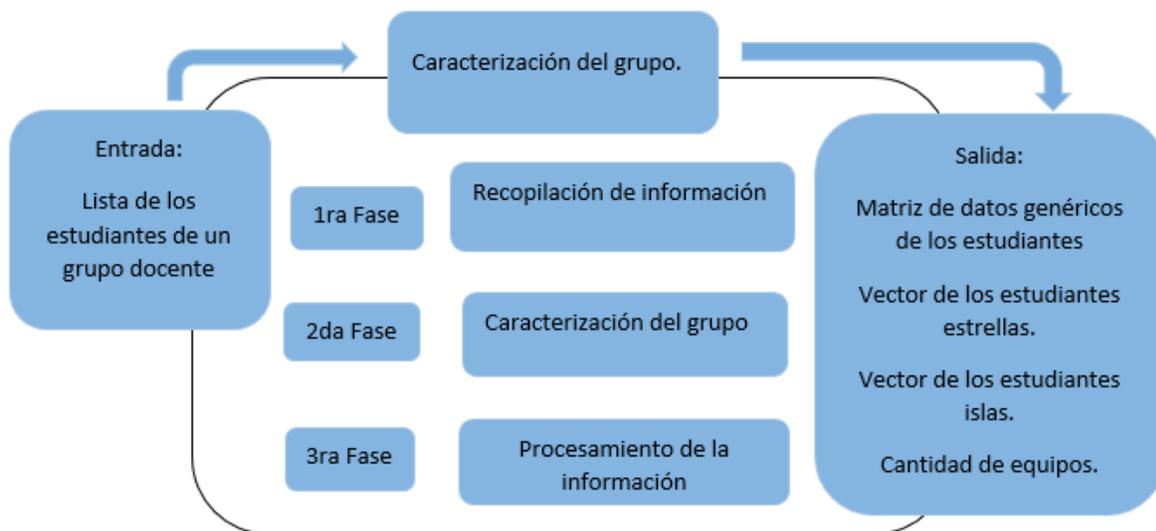


Figura 2. Etapa caracterización del grupo.

Primera fase: Recopilación de información

Esta primera fase consiste en la aplicación del test sociométrico y del test de Belbin, correspondientes a la dimensión social y la dimensión expectativa de éxito en la producción respectivamente, además de la obtención de la evaluación docente de los estudiantes referente

a la dimensión desempeño docente que se tienen en cuenta en el modelo para la formación de equipos de trabajo en ambiente académico-profesional. Junto a esto será definida la cantidad de equipos que se formarán en la etapa posterior.

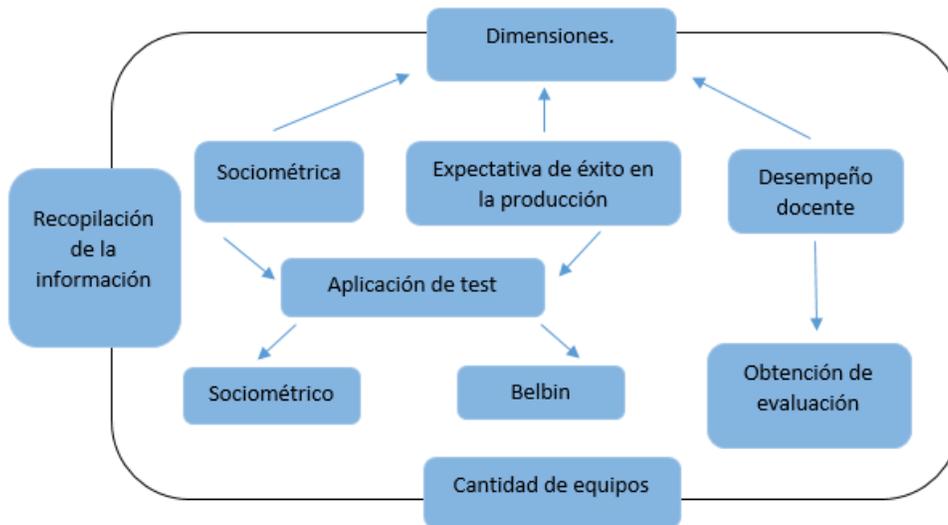


Figura 3. Recopilación de información

Como resultado de esta fase se les aplica a los estudiantes un test sociométrico, en donde responderán tres preguntas realizadas por el profesor. Además, responderán el test de autopercepción de Belbin, el cual arrojará como resultado el rol favorito a desempeñar por un estudiante. Por último, el profesor les asignará a los estudiantes una evaluación que puede ser M, R, B, E.

Además, se deben definir los aspectos siguientes:

Sea:

$G = \{m_1, m_2, \dots, m_n\}$ un grupo docente de n estudiantes

n : cantidad de estudiantes en el grupo. $10 < n \leq 25$.

$E = \{e_1, e_2, \dots, e_k\}$ El conjunto de los equipos de trabajo de j miembros que se forman en G .

Donde:

k : cantidad de equipos de j miembros $2 \leq k \leq 12$

j : cantidad de miembros por equipo $2 \leq j \leq 5$

La conformación del conjunto E de equipos de j miembros en el grupo docente G se realizará teniendo en cuenta una caracterización de los estudiantes del grupo a partir tres dimensiones:

D_1 : Social

Instrumento: test sociométrico

Indicadores:

ES= {es1}: conjunto de estudiantes estrellas

Es: Estudiante estrella va a ser el estudiante más votado en el test sociométrico

EI= {ei1}: conjunto de estudiantes islas

EI: Estudiante isla: Va a ser aquel estudiante que no fue votado en ninguna opción en el test sociométrico.

IPj = [p1, p2, p3] índice de preferencia por estudiantes

D_2 : Desempeño docente

Instrumento a utilizar es externo al sistema, cada profesor decide cómo medir el desempeño académico del estudiante.

Indicador académico: es introducido por el docente y debe cumplir con la escala nominal [E B R M]).

IA: {ev}: conjunto de evaluaciones

D_3 : Expectativa de éxito

Instrumento: Test de autopercepción de Belbin

Indicador: roles preferidos de primer y segundo nivel

eej= [rp1, rp2]

RP1: rol que más puntuación obtenga.

RP2: 2do rol que más puntuación obtenga

Fase 2: Caracterización del grupo

Después de obtener los resultados del test sociométrico, del test de Belbin y de la evaluación docente, correspondientes a las tres dimensiones a tener en cuenta para la formación de equipos de trabajo en ambiente académico-profesional, y de haber definido la cantidad de equipos, se construirá una matriz que contendrá esos resultados por cada estudiante,

caracterizando de esta manera al grupo docente con respecto a la dimensión social, expectativa de éxito en la producción y desempeño docente.



Figura 4. Fase 2: Caracterización del grupo

Después de aplicados los test explicados anteriormente en la primera fase, se obtiene la figura siguiente:

Estudiante	Test Sociométrico			Roles de Belbin		Desempeño docente
Nombre	Preferencia 1	Preferencia 2	Preferencia 3	Nivel 1	Nivel 2	Evaluación
Laura	Carlos	Ana	José	ID	ES	B

Figura 5. Matriz genérica por estudiante

Fase 3: Procesamiento de datos

Esta fase constituye la última de la primera etapa del modelo. En ella se procesan los datos que contiene la matriz con información referente a las características de los estudiantes correspondientes de las tres dimensiones que tiene en cuenta el modelo para la formación de equipos de trabajo en ambiente académico-profesional. Como resultado de la información procesada se obtienen la lista de estudiantes estrellas y de estudiantes islas.

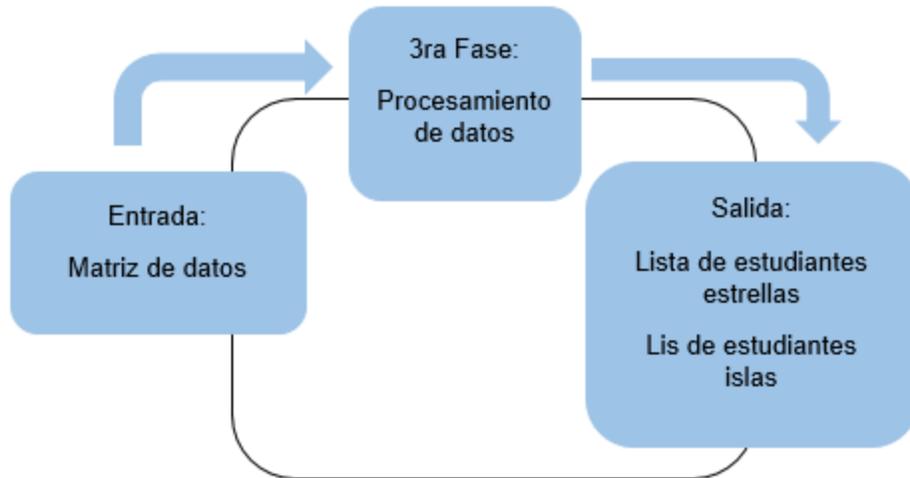


Figura 6. Procesamiento de datos.

Segunda etapa: Formación de equipos de trabajo.

Esta segunda etapa del modelo propuesto para la formación de equipo de trabajo en ambiente académico-profesional será explicada de la misma manera que la primera etapa, es decir se detallará cada una de sus etapas para una mejor comprensión, las mismas son: Ejecución del algoritmo y análisis del resultado.



Figura 7. Formación de equipos

1ra Fase: Ejecución de algoritmo

```

1. Inicio [Algoritmo Formar Equipos]
2. Asignar EstudiantesEstrellas()
   3. Si cantInslas>0
     3.1 Asignar EstudiantesIslas()
4. Mientras cantEstudiantes>0
   4.1 Desde i=1 hasta cantEstudiantes hacer
     4.1.1 ultimo=UltimoEstudianteAsignado()
     4.1.2 Si ultimo.Categoria() distinto ListadoEstudiantes[i].Categoria()
       4.1.2.1 Si ultimo.evaluacion() distinto ListadoEstudiantes[i].Evaluacion()
         4.1.2.1.1 AsignarEstudianteaEquipo(ListaEstudiantes[i])
         4.1.2.1.2 Romper
       4.2 Si NoHayEstudianteConCategoriaDiferente()
         4.2.1 Desde i=1 hasta cantEstudiantes hacer
           4.2.1.1 ultimo=UltimoEstudianteAsignado()
           4.2.2 Si ultimo.Rol() distinto ListadoEstudiantes[i].Rol()
             4.2.2.1 Si ultimo.evaluacion() distinto ListadoEstudiantes[i].Evaluacion()
               4.2.2.1.1 AsignarEstudianteaEquipo(ListaEstudiantes[i])
               4.2.2.1.2 Romper
           4.3 Si NoHayEstudianteConRolDiferente()
             4.3.1 AsignarEstudianteEquipo(EscogeCualquiera())
     5. Fin

```

Figura 8. Pseudocódigo del algoritmo

Modelo computacional para formar los equipos a partir de su caracterización

Procedimiento para la formación de equipos de trabajo de 5 miembros.

1-Determinar G,E,j.

2-Obtener caracterización del grupo.

3-Procesamiento de la información.

4-Obtención de reglas para formar equipos.

Las reglas se definen en función del tamaño del equipo. El profesor va a decidir de cuántos integrantes desea formar los equipos en el aula.

Para equipos de cinco integrantes se definen las siguientes reglas:

Condición de Belbin:

El estudiante que será elegido (sea isla o no), tendrá que poseer una categoría distinta de Belbin con respecto al estudiante que se esté comparando, respetando siempre la prioridad entre categorías de Belbin. Si no es posible cumplir con esto entonces se buscará un estudiante con un rol diferente al del estudiante con el que se está comparando.

Prioridad entre categorías de Belbin:

Se debe priorizar que los roles de acción predominen sobre los mentales y estos sobre los sociales.

Condición académica:

El estudiante que será elegido (sea isla o no), tendrá que poseer un desempeño docente distinta a la del estudiante con que se esté comparando.

Tratamiento de islas:

Se le otorgará prioridad a las islas que existan en el grupo (en caso de que existan), comenzando a ubicarlas en el equipo con el estudiante estrella que mayores votos a su favor tenga. Para eso se buscará en la lista de islas todos aquellos estudiantes que en primer lugar cumpla con la Condición de Belbin.

Luego de obtener la lista de islas que cumple con la condición de Belbin se buscará entre estos los que cumplan con la Condición académica. Si no existe ningún estudiante que cumpla esta condición será elegido como segundo integrante del equipo aquel que solo cumpla con la Condición de Belbin.

Ausencia de islas:

En caso de que no existan islas o las que existan ya estén ubicadas y falten equipos por integrar a su segundo miembro, se comenzará a llenar los equipos en el orden en que el estudiante estrella de cada equipo tenga menor votos a su favor.

4.1-Elegir el primer miembro del equipo:

Los estudiantes con mayor cantidad de votos obtenidos según la preferencia de los estudiantes del grupo serán denominados estudiante estrella o cabeza de equipo.

Se tendrá tanto estudiantes estrellas como cantidad de equipos y estos serán los primeros integrantes de cada equipo.



Figura 9. Estudiante estrella

4.2-Elegir el segundo miembro de los equipos:

-Se comenzará realizando el **Tratamiento de isla**. Si después de esto no falta ningún equipo por integrar su segundo estudiante se pasará a ubicar el próximo miembro.

-Luego se realizará la **Ausencia de islas**.

-Se buscará los estudiantes que pertenezca a la lista de estudiantes que prefieran al estudiante estrella y que cumpla con la **Condición de Belbin**.

Luego de obtener la lista de estudiantes que cumple con la condición de Belbin se buscará entre estos los que cumplan con la **Condición académica**. Si no existe ningún estudiante que cumpla esta condición será elegido como segundo integrante del equipo aquel que solo cumpla con la **Condición de Belbin**.

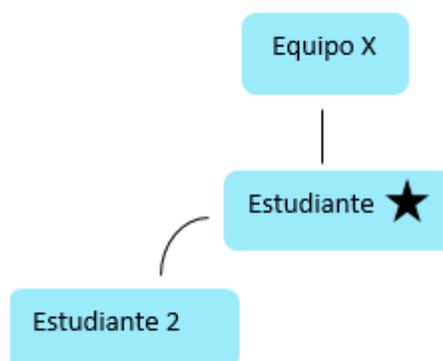


Figura 10. Segundo estudiante

4.3-Elegir el tercer miembro de los equipos:

-Se comenzará realizando el **Tratamiento de isla**. Si después de esto no falta ningún equipo por integrar su tercer estudiante se pasará a ubicar el próximo miembro.

-Luego se realizará la **Ausencia de islas**.

-Para elegir al tercer miembro del equipo que no es una isla, se escogerá a los estudiantes que aparezcan en la lista de estudiantes que sean preferidos por el segundo estudiante del equipo al que se integrarán y que cumplan con la **Condición de Belbin**.

-Luego de obtener la lista de estudiantes que cumple con la condición de Belbin se buscará entre estos los que cumplan con la **Condición académica**. Si no existe ningún estudiante que cumpla esta condición será elegido como segundo integrante del equipo aquel que solo cumpla con la **Condición de Belbin**.

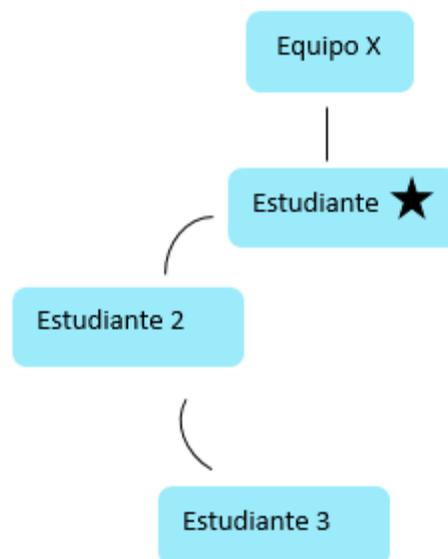


Figura 11. Tercer estudiante

4.3-Elegir el cuarto miembro de los equipos:

-Se comenzará realizando el **Tratamiento de isla**. Si después de esto no falta ningún equipo por integrar su cuarto estudiante se pasará a ubicar el próximo miembro.

-Luego se realizará la **Ausencia de islas**.

-Para elegir al cuarto miembro del equipo que no es una isla, se escogerá a los estudiantes que aparezcan en la lista de estudiantes que sean preferidos por el segundo estudiante del equipo al que se integrarán y que cumplan con la **Condición de Belbin**.

-Luego de obtener la lista de estudiantes que cumple con la condición de Belbin se buscará entre estos los que cumplan con la **Condición académica**. Si no existe ningún estudiante que

cumpla esta condición será elegido como segundo integrante del equipo aquel que solo cumpla con la **Condición de Belbin**.

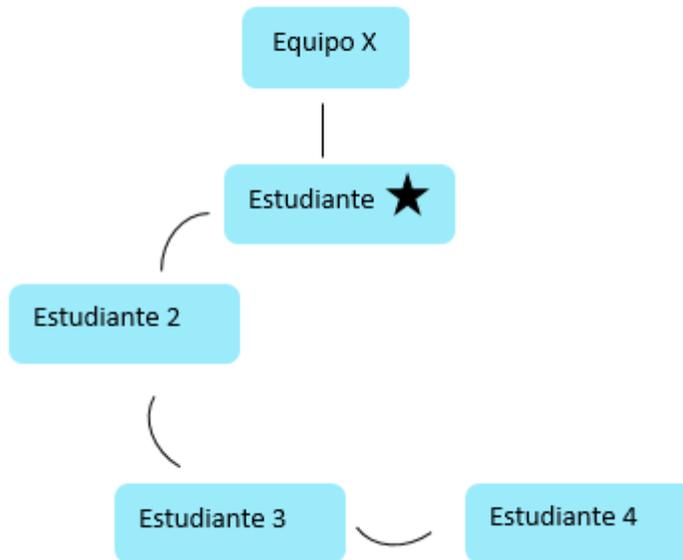


Figura 12. Cuarto estudiante

4.4- Elegir el quinto miembro de los equipos:

-Se comenzará realizando el **Tratamiento de isla**. Si después de esto no falta ningún equipo por integrar su cuarto estudiante se pasará a ubicar el próximo miembro.

-Luego se realizará la **Ausencia de islas**.

-Para elegir al cuarto miembro del equipo que no es una isla, se escogerá a los estudiantes que aparezcan en la lista de estudiantes que sean preferidos por el segundo estudiante del equipo al que se integrarán y que cumplan con la **Condición de Belbin**.

-Luego de obtener la lista de estudiantes que cumple con la condición de Belbin se buscará entre estos los que cumplan con la **Condición académica**. Si no existe ningún estudiante que cumpla esta condición será elegido como segundo integrante del equipo aquel que solo cumpla con la **Condición de Belbin**.

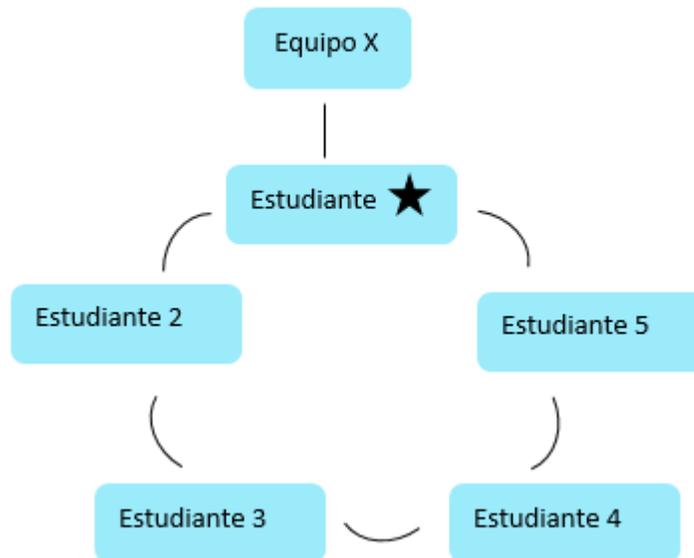


Figura 13. Quinto estudiante

Fase 2: Análisis de resultados

Una vez los equipos ya estén formados los profesores tendrán la última palabra sobre si esos equipos serán los que ellos utilizarán con sus docentes. Es decir, los profesores tendrán la posibilidad de intercambiar estudiantes de los equipos creados, en caso de que por alguna razón ellos estimen que este cambio sea conveniente para un mejor funcionamiento del equipo.

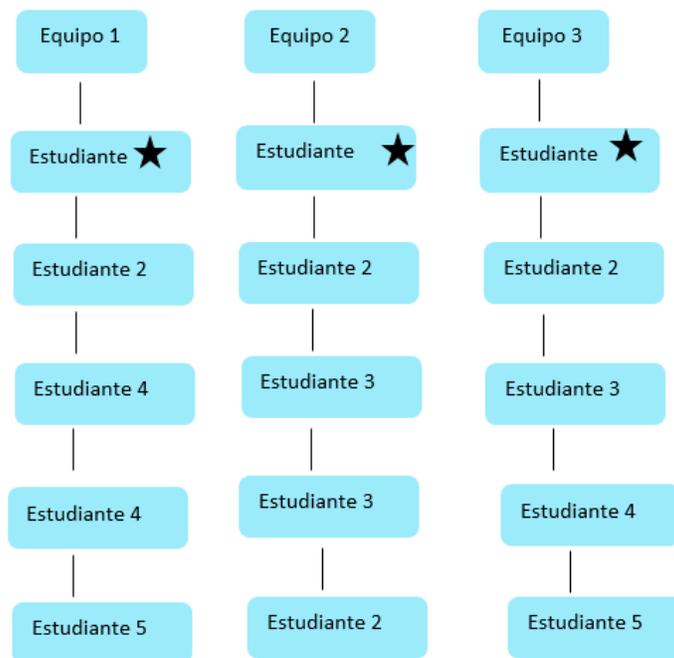


Figura 14. Equipos formados

2.2 Requisitos de la herramienta

Para la obtención de los requisitos del sistema se utilizaron dos técnicas: en un primer momento la observación la cual proporcionó una forma objetiva para conseguir los requerimientos, no es dependiente de otras personas y se estudian los hechos sin intermediarios. Como segunda técnica se empleó la entrevista que sería un intento sistemático de recoger información de otra persona a través de una comunicación interpersonal que se lleva a cabo por medio de una conversación estructurada.

A continuación, se exponen los requisitos funcionales y no funcionales, definidos para la realización y desarrollo del sistema que se propone. Los mismos se han especificado a partir de las características y funcionalidades que debe tener el sistema que se va a desarrollar

2.2.1 Requisitos funcionales

- RF 1 Autenticar usuario
- RF 2 Realizar test sociométrico
- RF 3 Realizar test de Belbin
- RF 4 Registrar evaluación docente
- RF 5 Registrar cantidad de equipos

- RF 6 Formar equipo
- RF 7 Visualizar equipos formados
- RF 8 Exportar equipos a PDF

2.2.2 Requisitos no funcionales

Los requisitos no funcionales describen aspectos del sistema que son visibles por el usuario y no incluyen una relación directa con el comportamiento funcional del sistema. Son propiedades o cualidades que el producto debe tener y cumplir.

A continuación, se enuncian los requisitos no funcionales identificados en la propuesta de solución:

Hardware

- Se recomienda para su funcionamiento óptimo máquinas con procesador: 1.6 GHz. o superior y memoria RAM: 512 MB o superior.
- Como requerimiento las máquinas deben poseer de disco duro: 40 GB o superior.

Software

- A nivel del cliente o usuario a la herramienta se puede acceder desde cualquier plataforma o sistema operativo, solo es necesario contar con una conexión a la red apropiada y un navegador web.
- Se recomienda el uso del navegador Mozilla Firefox para su funcionamiento óptimo.

Interfaz

- El sistema debe ofrecer una interfaz intuitiva y sencilla de usar, amigable y donde cualquier usuario podría hacer uso de esta sin dificultad.

Seguridad

- La seguridad estará regida por un mecanismo de control de acceso basado en roles, convenientemente asignados a los usuarios del sistema al momento de su creación.
- Se establecerá un mecanismo de autenticación para el acceso al sistema.

2.3 Historias de usuario

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas (Xp., 2011).

Tabla 3. Historia de usuario: Formar equipo

Historia de usuario	
Numero: HU-06	Usuario: Profesor
Nombre de historia: Formar equipo	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 1
Programadores responsables: Jessica León	
Leidy Bacerio	
Descripción: La funcionalidad permitirá al profesor formar equipos.	
Observaciones:	

2.4 Plan de iteraciones

El plan de iteraciones es una técnica que utiliza la Metodología XP, como parte de su ciclo de vida, donde crea un plan de duración de cada una de las iteraciones, en este caso se hace para el único equipo de desarrollo con el cual se cuenta. Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado, donde se muestra la duración de cada iteración, así como el orden en que serán implementadas las historias de usuario en cada una de las mismas.

Tabla 4 Plan de iteraciones. Elaboración propia

No. Iteración	Historia de usuario	Prioridad	Esfuerzo estimado
1	1. Autenticar	Alta	4 semanas
	2. Formar equipo	Alta	
2	3. Realizar test sociométrico	Media	9 semanas
	4. Realizar test de Belbin	Media	
	5. Registrar evaluación docente	Media	
	6. Visualizar informe con los equipos formados	Baja	
	7. Exportar informe con los equipos	Baja	

2.5 Plan de entregas

El plan de entrega se planifica mediante la realización de reuniones con el equipo de desarrollo y el cliente, donde se definen el tiempo de realización del sistema informático. El objetivo de este plan es producir rápidamente versiones de la aplicación que sean operativas, aunque estas no cuenten con toda la funcionalidad pretendida.

Tabla 5 Plan de entrega. Elaboración propia

Iteraciones	Final de la iteración (7 de marzo de 2017)	Final de la iteración (10 de mayo de 2017)
Iteración 1	x	
Iteración 2		x

2.6 Arquitectura de solución

La Arquitectura de Software es una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y

las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión del detalle inherente a la mayor parte de las abstracciones (Clements, 1996). Para el desarrollo del sistema se ha definido emplear una arquitectura base, para orientar el diseño de las capas lógicas a través de una propuesta de diseño base; organizar la forma de codificar según las propuestas de convenciones o estándares de códigos y recursos; brindar una estructura física para soportar el código, creando así un esqueleto base; y proponer mecanismos de colaboración entre los componentes integrados en ella. La arquitectura a la cual se hace referencia es la arquitectura en 3 niveles, en la cual existe un nivel intermediario. Esto significa que la arquitectura generalmente está compartida por:

1. Un cliente, es decir, el equipo que solicita los recursos, equipado con una interfaz de usuario (generalmente un navegador Web) para la presentación.
2. El servidor de aplicaciones (también denominado software intermedio), cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo.
3. El servidor de datos, que proporciona al servidor de aplicaciones los datos que requiere.

Patrón Modelo-Vista-Controlador (Model-View-Controller, según sus siglas en inglés (MVC)): usado principalmente en aplicaciones que manejan gran cantidad de datos y transacciones complejas donde se requiere una mejor separación de conceptos para que el desarrollo esté estructurado de una mejor manera, facilitando la programación en diferentes capas de manera paralela e independiente. MVC sugiere la separación del software en 3 estratos: Modelo, Vista y Controlador, que serán explicados brevemente: (Arquitectura del Software, s.f.).

Modelo: Es la representación de la información que maneja la aplicación. El modelo en sí son los datos puros que puestos en contexto del sistema proveen de información al usuario a la aplicación misma (Arquitectura del Software, s.f.).

Vista: Es la representación del modelo en forma gráfica disponible para la interacción con el usuario. En el caso de una aplicación web, la “vista” es una página HTML con contenido dinámico sobre el cual el usuario puede realizar operaciones. (Arquitectura del Software, s.f.)

Controlador: Es la capa encargada de manejar y responder las solicitudes del usuario, procesando la información necesaria y modificando el modelo en caso de ser necesario.

En la etapa de diseño se propone el empleo de tarjetas CRC (Clase Responsabilidad Colaboración), en lugar de diagramas de clases para la descripción del sistema en notación UML.

Las tarjetas CRC admiten desprenderse del método de trabajo basado en procedimientos y trabajar con una metodología basada en objetos, además permite que el equipo completo contribuya en la tarea del diseño. La característica más sobresaliente de las tarjetas CRC es su sencillez y adaptabilidad. Las tarjetas CRC son fichas que representan cada una de las clases del sistema, en ellas se describen brevemente las responsabilidades de la clase, mostrando un listado de las clases con las que colabora para ejecutar dichas responsabilidades. Su utilización en el diseño potencia el uso de patrones de asignación de responsabilidades.

A continuación, se muestra la tarjeta CRC correspondiente a la historia de usuario Formar equipo.

Tabla 6. Tarjeta CRC

Tarjeta CRC	
Clase: DefaultController.php	
Responsabilidades:	Colaboraciones:
Esta clase es la encargada de formar los equipos	formarequipo.html.twig preferido.html.twig preferido.php preferidoRepository.php belbin.html.twig belbin.php belbinRepository.php evaluacion.html.twig evaluacion.php evaluacionRepository.php

2.8 Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular, propiciando sencillez y reutilización. Para un mejor diseño de la solución, se hace uso de los siguientes patrones:

2.8.1 Patrones GRASP

Experto: Propone asignar las responsabilidades a las clases de acuerdo a la información que contienen las mismas cumpliendo así un principio básico e intuitivo de la programación orientada a objetos (LARMAN, 2005). En la presente solución se utiliza el mencionado patrón dentro de la capa de acceso a datos, ya que Symfony utiliza el objeto relacional de mapeo (ORM, por sus siglas en inglés) Doctrine en la capa del modelo en la clase “belbinRepository.php”, esto permite que se pueda manejar su información como un objeto de tipo entidad mapeada.

```
class belbinRepository extends EntityRepository{

    public function Belbin($estudiante)

    {

        $query=$this->_em->createQueryBuilder('c')

        ->select('c')

        ->from('AppBundle:belbin','c')

        ->where('c.usuario in (:estudiante)')

        ->setParameter('estudiante',$estudiante)

        ->getQuery()->execute();

        return $query;

    }

}
```

Creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos (LARMAN, 2005). Este patrón tiene como función la de encontrar un creador que necesite conectarse al objeto creado en alguna situación. Se observa en el sistema propuesto en la clase “Defaultcontroller.php”, la cual crea instancias de la clase que provee la información necesaria para su propio manejo. De manera general las clases enmarcadas dentro de la capa controlador distribuyen las responsabilidades de comunicación con las instancias de la capa de presentación.

`$em = $this->getDoctrine()->getEntityManager();` se crea un objeto para acceder a las funcionalidades de la clase entidad.

Bajo acoplamiento: Determina el nivel de dependencia de una clase con respecto a otras. Su uso potencia la reutilización, el mantenimiento y la mitigación de efectos a producirse en una clase al hacer cambios en otra. En la presente solución se pone de ejemplo la existencia de una baja interdependencia entre la clase controladoras (DefaultController.php), el modelo (belbinRepository.php) y la vista (belbin_new.html.twig). Esta característica evidencia la posibilidad de efectuar cambios en estas sin que las otras sufran grandes afectaciones.

Alta cohesión: Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Significa que las clases del sistema tienen asignadas solo las responsabilidades que les corresponden y mantienen una estrecha relación con el resto de las clases. Este patrón se evidencia en la propuesta de solución con las clases “DefaultController.php” y “FormarEquipos.php” pues las mismas muestran responsabilidades relacionadas coherentemente, que se complementan entre sí. Esta situación garantiza que exista un bajo acoplamiento, favoreciendo un equilibrio al interactuar los objetos.

Controlador: El patrón controlador sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, manejando los eventos de entrada de dicha interfaz (LARMAN, 2005). En la aplicación se manifiesta el uso de este patrón con la creación de varios controladores que se encargan de manejar eventos, el controlador “DefaultController.php” ejemplifica claramente el uso de este patrón.

2.8.2 Patrones GoF

Los patrones GoF (por sus siglas en inglés The Gang of Four) se utilizan para diseñar objetos y solucionar problemas de creación de instancias, ya que ayudan a encapsular y abstraer dicha creación. En la propuesta de solución se manifiesta dichos patrones:

Singleton: Puesto en práctica con el fin de garantizar que una clase solo tenga una única instancia, proporcionando un punto de acceso local a la misma (LARMAN, 2005). El uso de este patrón se manifiesta en la clase “EntityManager.php” que tiene la función de acceder a las prestaciones de Doctrine, la cual, al constituir una única instancia, es accedida directamente sin que exista la posibilidad de crear nuevos objetos de su tipo.

```
$this->getDoctrine()->getEntityManager();
```

Decorador: Es un patrón de tipo estructural que permite añadir dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender funcionalidades (LARMAN, 2005). Este patrón está presente en la propuesta de solución a través de las plantillas twig para las vistas. Dichas plantillas heredan de la plantilla “base.html.twig”, lo cual permite declarar bloques editables, dentro de los cuales se realizar modificaciones específicas manteniendo una homogeneidad en la apariencia.

```
{% extends ':default:principal.html.twig' %}
```

```
{% block dashboard %} {% endblock %}
```

Agente Remoto: Este patrón es utilizado cuando se requiere que el sistema en cuestión se comunique con un servicio externo y no se desea o no es posible acceder directamente a este (LARMAN, 2005). La utilización de este patrón en la solución se evidencia en la autenticación mediante el Web Service disponible para la UCI. Este servicio permite la autenticación y obtención de los datos referentes a los usuarios de la universidad utilizando también la clase mediadora “DefaultController.php”.

```
$datos=$this->container-get("besimple.soap.client.autenticacion")->ObtenerPersonaDadoUsuario($usuario);
```

2.9 Modelo de datos

El modelado y diseño de la base de datos es un factor de suma importancia para la creación de software. Tiene como objetivo generar un conjunto de entidades relacionadas entre sí, que permitan el almacenamiento de la información con un mínimo de redundancia, manteniendo su integridad y facilitando la recuperación para su consulta.

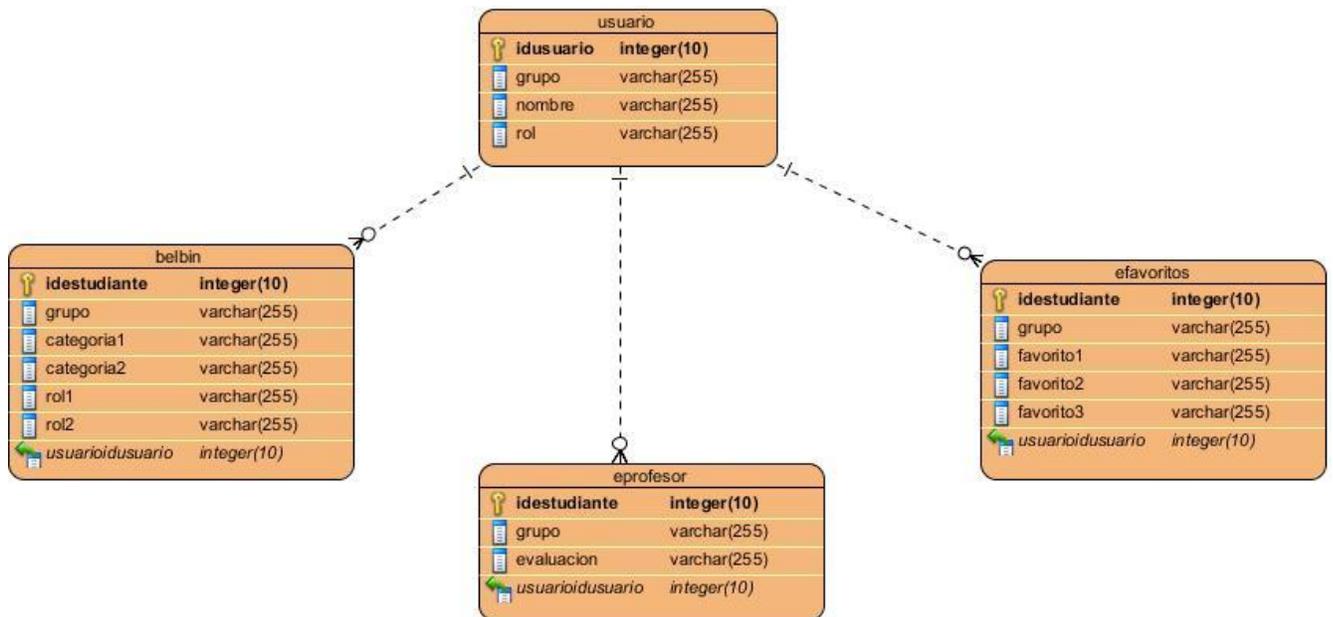


Figura 16. Modelo de datos

2.10 Estándar de codificación

Para asegurar la calidad de un software uno de los instrumentos que se utilizan es la adopción de estándares de codificación. Los mismos presentan ventajas como:

- Asegurar la legibilidad del código entre distintos programadores facilitando el trazo del mismo.
- Proveer una guía para el encargado de mantenimiento/actualización del sistema, con código claro y bien documentado.
- Facilitar la portabilidad entre plataformas y aplicaciones.

Los estándares de codificación permiten entender de manera rápida, fácil y sencilla, el código empleado en el desarrollo de un software. Es de gran importancia el usar técnicas de codificación y realizar buenas prácticas de programación con vistas a generar un código de alta calidad. Si se aplica de forma continua un estándar de codificación bien definido, y posteriormente se efectúan revisiones del código, caben muchas posibilidades de que un proyecto de software se convierta en un sistema fácil de comprender y de mantener, garantizando un mantenimiento óptimo de dicho código por parte del programador. Es por ello que para la implementación de la herramienta se definen los requisitos que se presentan a continuación.

Identación: La unidad de indentado es de 4 espacios. El uso de la tabulación debe ser evitado porque no existe un estándar que determine con precisión el ancho que va a producir la tabulación (CALLEJA, Estándares de codificación).

```

438 public function EscogerCualquiera($listadoest)
439 {
440     for($j=0;$j<count($listadoest);$j++)
441     {
442         if($listadoest[$j]!=0)
443         {
444             return $listadoest[$j];
445         }
446     }
447
448     return -1;
449 }

```

Figura 17. Estándar de codificación

Declaración de variables: Cada variable debe ser declarada en una línea y comentada. También deben aparecer ordenadas alfabéticamente. El nombre de las variables debe de comenzar con letras minúsculas y cada palabra relevante por la que esté compuesta debe ser con letra mayúscula (CALLEJA, Estándares de codificación).

```
$db = $em->getConnection(); // objeto de conexion
```

Figura 18. Estándar de codificación

Declaración de funciones: No debe haber espacio entre el nombre de la función y el paréntesis izquierdo, ni entre este y la lista de parámetros. Debe haber un espacio entre el paréntesis derecho y la llave de comienzo del cuerpo de la función. Las sentencias del cuerpo deben estar en la línea siguiente. La llave que cierra debe estar alineada con la línea de declaración de la función. Los nombres de las funciones se rigen por las mismas características que el de las variables (CALLEJA, Estándares de codificación).

```

348 public function MiEvaluacion($id)
349 {
350     $em = $this->getDoctrine()->getEntityManager();
351     $db = $em->getConnection();

```

Figura 19. Estándar de codificación

Sentencia return: Una sentencia return no debe utilizar paréntesis “()” alrededor del valor que se retorna. La expresión cuyo valor se retorna debe comenzar en la misma línea que la palabra reservada return, terminada con un punto y coma (CALLEJA, Estándares de codificación).

```

554
555
556
557
558
559
560
561
562
563

```

```

    if($cantidad[$i]==0)
    {
        $devolver[$cont]=" ";
        $devolver[$cont]=$ids[$i];
        $cont++;
    }
}

return $devolver;

```

Figura 20. Estándar de codificación

Sentencia if: La sentencia siempre lleva “{}” (CALLEJA, Estándares de codificación).

```

554
555
556
557
558
559
560

```

```

    if($cantidad[$i]==0)
    {
        $devolver[$cont]=" ";
        $devolver[$cont]=$ids[$i];
        $cont++;
    }

```

Figura 21. Estándar de codificación

Espacios en blanco: Las líneas en blanco facilitan la lectura determinando secciones de código lógicamente relacionados. Los espacios en blanco pueden ser (o no deben ser) utilizados en las siguientes circunstancias:

- No se debe utilizar un espacio en blanco entre el identificador de una función y el paréntesis que abre la lista de parámetros. Esto ayuda a distinguir entre palabras reservadas y llamadas a funciones.
- Para cualquier operador binario excepto el punto “.”, el paréntesis que abre “(” y el corchete que abre “[” todos deben ser separados por un espacio entre operandos y operador.
- No se debe utilizar el espacio para separar un operador unario de su operando excepto cuando ese operador es una palabra como `typeof`.
- Cada punto y coma “;” en una sentencia de control debe ser seguido por un espacio.
- Debe dejarse un espacio luego de cada coma “,”. (JURISCO, 2005)

2.11 Conclusiones parciales

Luego del desarrollo del presente capítulo se puede llegar a las siguientes conclusiones:

- Se hizo un empleo de la metodología XP en función de describir el desarrollo de la solución propuesta.

- Se especificaron y validaron los requisitos funcionales y no funcionales del sistema teniendo en cuenta las características que el mismo debe cumplir, identificándose 7 requisitos funcionales agrupados en 6 historias de usuario.
- Se fundamentó la elección del patrón Modelo Vista Controlador para la arquitectura del sistema.
- Se obtuvieron las tarjetas CRC correspondientes a cada una de las clases y se evidenció como los patrones de diseño pueden contribuir a la obtención de un producto final de mayor calidad.
- Por último, se definen los estándares de codificación lo que hará más sencillo el análisis del código y el mantenimiento del sistema.

Capítulo 3. Verificación de la viabilidad de la investigación:

En el capítulo se describirá la estrategia seguida para verificar la viabilidad de la investigación. Primeramente, se verificará el cumplimiento de la funcionalidad del sistema a través de las pruebas de liberación realizadas por el grupo de calidad del software del Centro de informatización de entidades (CEIGE) y la prueba de aceptación del cliente. Además, adjuntándose a la metodología se realiza también prueba de caja negra usando la técnica de partición de equivalencia y prueba de caja blanca usando la técnica de camino básico. Finalmente se valora el aporte práctico de la investigación a través de su valor científico y su aplicabilidad mediante la consulta a profesores.

Estrategia para la verificación de la viabilidad de la investigación



Figura 22. Estrategia para verificar la viabilidad de la solución

3.1 Prueba de aceptación

El objetivo de estas pruebas es verificar los requisitos, por lo que los propios requisitos del sistema son la principal fuente de información a la hora de construir las pruebas de aceptación. Las pruebas de aceptación son creadas a partir de las historias de usuario. Durante una iteración la historia de usuario seleccionada en la planificación de iteraciones se convertirá en una prueba de aceptación. El cliente o usuario especifica los aspectos a testear cuando una historia de usuario ha sido correctamente implementada. Una historia de usuario puede tener más de una prueba de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento y no se considera completa hasta que no supera sus pruebas de aceptación.

Para la aplicación de estas pruebas a la solución obtenida, se confeccionaron casos de prueba de aceptación por cada HU. Seguidamente la tabla 7 muestra el caso de prueba de aceptación para la HU: Formar equipo.

Tabla 7. Caso de prueba: Formar equipos

Caso de prueba de aceptación			
Historia de usuario	HU-06 Formar equipo.		
Descripción	Permite formar equipo a partir de tres dimensiones.		
Condiciones de ejecución	El usuario se autentica en el sistema. El usuario autenticado sea profesor. Los estudiantes hayan realizado todas las encuestas.		
Escenarios de prueba	Flujo del escenario	Resultados esperados	
EP1	Formar equipo introduciendo datos válidos	Se introducen datos válidos para formar equipos.	Se registran los datos.
EP2	Formar equipo no especificando algunos de los campos	Se introducen datos dejando campos obligatorios vacíos. Se intenta pasar a la siguiente página del protocolo.	El sistema muestra una alerta sobre los campos obligatorios que permanecen vacíos. Se resaltan los campos vacíos. El protocolo no pasa de página.
EP3	Formar equipos dejando campos obligatorios vacíos	Se introducen datos dejando campos obligatorios vacíos	El sistema muestra una alerta sobre los campos obligatorios que permanecen vacíos. Se resaltan los campos vacíos.

Tabla 8. Juego de datos de prueba: Formar equipo (V: válido, I: inválido)

Escenario de prueba	Juego de datos de prueba	
	Grupo	Cantidad de equipos
EC1	G-3204	5
	(V)	(V)

EC2		4
	(I)	(V)
	G3204	
	(V)	(I)
EC3		
	(I)	(I)

A continuación, se muestran los resultados de la prueba de aceptación, donde en una primera iteración se obtuvo un total de 10 no conformidades (NC), 4 críticas y 6 no críticas. En la segunda iteración se obtuvo un total 4 NC, 1 crítica y 3 no críticas, y para una tercera iteración no se detectó ninguna NC, evidenciando de esta forma que el sistema funciona correctamente.

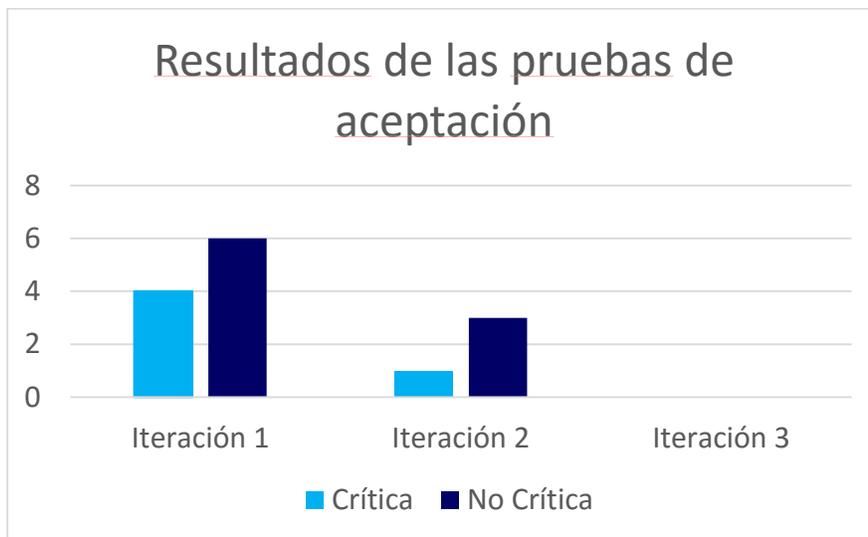


Figura 23. Resultados de las pruebas de aceptación

3.2 Pruebas de unidad

La metodología ágil XP divide las pruebas en dos grupos: pruebas unitarias y pruebas de aceptación. Las pruebas unitarias son desarrolladas por los programadores y se encargan de verificar el código automáticamente y las pruebas de aceptación.

3.2.1 Pruebas de caja blanca

El método de pruebas de caja blanca precisa del acceso al código del programa de modo que se pueda comprobar su lógica interna.

Las pruebas de caja blanca intentan garantizar que:

- Se ejecutan al menos una vez todos los caminos independientes de cada componente.
- Se utilizan las decisiones en su parte verdadera y en su parte falsa
- Se ejecuten todos los bucles en sus límites
- Se utilizan todas las estructuras de datos internas

Prueba del camino básico

Para la aplicación de las pruebas de caja blanca se hizo uso de la técnica camino básico. El método del camino básico permite obtener una medida de la complejidad de un diseño procedimental, y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez. A continuación se evidencian una serie de pasos:

- Obtener el grafo de flujo, a partir del diseño o del código del componente.
- Obtener la complejidad ciclomática del grafo de flujo.
- Definir el conjunto básico de caminos independientes.
- Determinar los casos de prueba que permitan la ejecución de cada uno de los caminos anteriores.
- Ejecutar cada caso de prueba y comprobar que los resultados son los esperados.

A continuación, se muestra un ejemplo basado en un diagrama de flujo que representa la estructura de control del programa en el método PosiblesCategorías.

```

public function PosiblesCategorias($categ1,$categ2)
{
    $categorias=array("accion","mental","social");
    $verdadero=array(true,true,true);
    $devolver=array();
    $i=0;
} 1

for($j=0;$j<3;$j++) 2
{
    if($categorias[$j]==$categ1) 3
    $verdadero[$j]=false; 4
} 5
for($j=0;$j<3;$j++) 6
{
    if($categorias[$j]==$categ2) 7
    $verdadero[$j]=false; 8
} 9

for($j=0;$j<3;$j++) 10
{
    if($verdadero[$j]) 11
    {
        $devolver[$i]=$categorias[$j]; } 12
    $i++;
}

} 13
return $devolver; 14
}
    
```

Figura 24. Ejemplo caja blanca

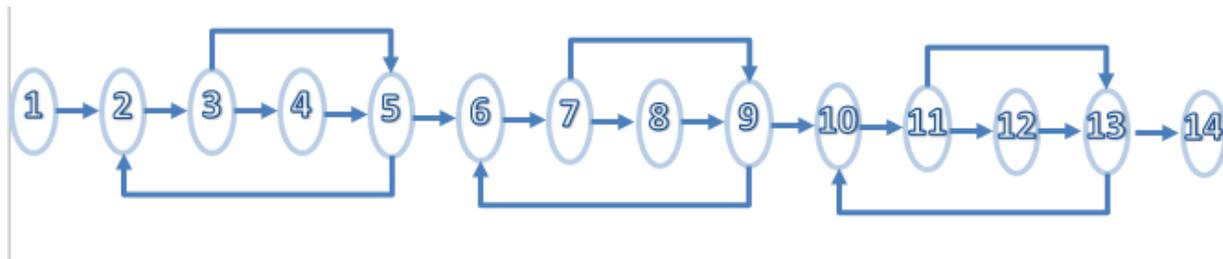


Figura 25. Grafo

Luego se calcula la complejidad ciclomática que no es más que una métrica de software que proporciona una medida cuantitativa de la complejidad lógica del software. El valor calculado mediante la complejidad ciclomática define el número de rutas independientes y proporciona un límite superior para el número de pruebas que debe aplicarse.

G-Grafo, A-número de aristas, N-número de nodos.

$$V(G) = (A - N) + 2 = (19 - 14) + 2 = 7$$

A=19; N=14; P=6; R=7

A partir del valor de la complejidad ciclomática obtenemos el número de caminos independientes, que nos dan un valor límite para el número de pruebas que tenemos que

diseñar. En el ejemplo, el número de caminos independientes es 7, y los caminos independientes son:

Camino 1: 1,2,3,4,5,6,7,8,9,10,11,12,13,14

Camino 2: 1,2,3,4,5,6,7,8,9,10,11,13,14

Camino 3: 1,2,3,4,5,6,7,9,10,11,12,13,14

Camino 4: 1,2,3,4,5,6,7,9,10,11,13,14

Camino 5: 1,2,3,5,6,7,8,9,10,11,12,13,14

Camino 6: 1,2,3,5,6,7,8,9,10,11,13,14

Camino 7: 1,2,3,5,6,7,9,10,11,12,13,1

Luego de tener elaborados los Grafos de Flujos y los caminos a recorrer, se preparan los casos de prueba que forzarán la ejecución de cada uno de esos caminos. Para definir los casos de prueba es necesario tener en cuenta:

Descripción: se describe el caso de prueba y de forma general se tratan los aspectos fundamentales de los datos de entrada.

Condición de ejecución: se especifica cada parámetro para que cumpla una condición deseada y así ver el funcionamiento del procedimiento.

Entrada: se muestran los parámetros que serán la entrada al procedimiento.

Resultados Esperados: se expone el resultado esperado que debe devolver el procedimiento después de efectuado el caso de prueba.

Caso de prueba para el camino básico # 1,2,3,4,5,6,7,8,9,10,11,12,13,14

Entrada: categoria1, categoria2

Resultados esperados: Se espera que se obtenga las categorías a este usuario.

Resultados obtenidos: Satisfactorio. Se muestran las categorías que complementan a los datos de entrada.

3.3 Valoración del aporte práctico de la investigación

Para validar la calidad del modelo conceptual a partir de su valor científico y su aplicabilidad se utiliza el método de evaluación por criterio de expertos, con el objetivo de elaborar pronósticos a largo plazo. Teniendo en cuenta que, la utilización sistemática de las valoraciones intuitivas

de un grupo de expertos para obtener un consenso de opiniones de perfecto acuerdo entre las partes, refuerzan la validez de la propuesta.

Los métodos de consulta a expertos se destacan por su amplia aplicación en disímiles campos a nivel internacional. Estos son utilizados con frecuencia para obtener información en campos científicos nuevos y en continua evolución como el de las TIC (Tecnología de la información y las comunicaciones).

El método de evaluación por criterio de expertos que se utiliza en la investigación permite aprovechar la sinergia del debate en el grupo, preservando el anonimato mediante el uso de flujos de retroalimentación.

Las etapas propuestas para la realización del método de evaluación por criterio de expertos son las siguientes:

1. Selección de expertos. Estudio preliminar de expertos para la toma de decisiones.
2. Elaboración del cuestionario, para la validación del Modelo Conceptual.
3. Determinación de la concordancia de los expertos.
4. Desarrollo práctico y explotación de los resultados. Durante este período, ningún experto conocerá la identidad de los otros miembros del grupo de debate.

1. Selección de los expertos

Se considera experto a los profesores que serán futuros usuarios de la. La selección se realizó teniendo en cuenta los siguientes criterios:

- I. Profesor principal del año.
- II. Jefe de departamento
- III. Jefe de asignatura
- IV. Profesor de una asignatura
- V. Especialista de producción vinculado a la docencia

Definición de los atributos

Para realizar la evaluación del modelo conceptual se define n los indicadores o atributos que serán evaluados por los expertos. A partir de estos se confecciona una encuesta para que los expertos expresen su valoración en relación con el modelo. Los atributos son:

A1: Valor Científico.

P1-Necesidad de la utilización de un modelo teórico que caracterice la composición adecuada de un equipo de trabajo a partir de las características de sus miembros referentes a la psicopedagogía, la docencia y la expectativa de éxito en la producción, teniendo en cuenta el tamaño de equipos para la formación de los mismos en ambiente académico-profesional

A3: Aplicabilidad.

P2, P3-Capacidad de que sea práctico, posible de utilizar y de adaptarse a las características de los grupos docentes.

Elaboración y lanzamiento de los cuestionarios

En este momento se elabora una encuesta (ver Anexo 12) que se adapte a las características de los expertos para validar el modelo conceptual que se propone.

Para validar el modelo conceptual se utilizó el coeficiente de concordancia (ω) de Kendall, el cual mide el grado de asociación entre varios conjuntos (k) de N entidades. Es útil para determinar el grado de acuerdo entre varios jueces, o la asociación entre tres o más variables. Ecuación (Kendall & Babington-Smith, 1938)

$$\omega = 12 \sum \frac{D^2}{m^2 n(n^2 - 1)}$$

$$\sum R = [mn(n + 1)] / 2.$$

Dónde:

ω = Estadística de Kendall.

$$D = \sum R - (\sum R/n).$$

m = Rango de evaluadores= Cantidad de encuestados.

n = Número de objetos= Cantidad de preguntas realizadas.

$\sum R$ = Suma de rangos.

Tabla 9. Rango de valores para el cálculo del coeficiente de concordancia

Valor 4	Muy valiosa	75 y 100%
Valor 3	Valiosa	50 y 75%
Valor 2	Poco valiosa	25 y 50%
Valor 1	Valiosa	0 y 25%

En dependencia de la evaluación que otorgue el experto a cada pregunta será el valor asociado que se asignará. El resultado de las evaluaciones se convierte en valores para obtener los rangos.

Tabla 10. Cálculo de la suma de rangos

Rango	Ex 1	Ex 2	Ex 3	Ex 4	Ex 5	Ex 6	Ex 7	Ex 8	Ex 9	Suma_rangos
--------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	--------------------

Valor 1	3	4	3	3	3	4	4	4	4	32
Valor 2	3	4	3	3	4	3	4	4	4	32
Valor 3	3	3	3	3	4	3	4	4	4	31
Suma Total										$\sum R = 95$

$$\sum R/n = \frac{95}{3} = 31$$

Tabla 11. Cálculo de factor de corrección

Suma de rangos	$ D = \sum R - 31$	D^2
32	1	1
32	1	1
31	0	0
$\sum R = 95$		$\sum = 2$

Calculo de W:

$$\omega = 12 \sum D^2 / m^2 n(n^2 - 1) = 31(4) / 36 * 3(3-1) = 124 / 216 = 0.57$$

W: expresa el grado de concordancia entre los nueve expertos al dar un orden evaluativo a los aspectos valorados. Este coeficiente siempre será positivo y su valor estará comprendido en el rango de 0 a 1.

Cálculo del Chi Cuadrado Real:

Si real $x^2 < x^2(\alpha, N - 1)$ entonces existe concordancia entre los expertos. El Chi cuadrado calculado se compara con los valores de las tablas estadísticas probadas para tales efectos con $\alpha = 0,05$ y $N = 3$, para un nivel de confianza del 95%.

Se busca en la tabla de distribución estadística $x^2(0,05; 2) = 10,5965$

comprobándose que:

$$x^2 \text{ real} < x^2(\alpha, N - 1) = 0,57 < 10,5965$$

La comparación realizada entre el valor de real y el valor tabulado, permite concluir que existe concordancia entre los expertos consultados.

Resultados de la validación del modelo

Mediante la encuesta los expertos emitieron sus criterios a cerca del valor científico y la aplicabilidad de la investigación. Estos criterios fueron cuantificados según indica la figura 26. El valor científico está representado en la encuesta por una pregunta, y la aplicabilidad está representada por dos preguntas. Cada pregunta podía obtener un máximo de treinta y seis puntos como suma de los criterios emitidos por los expertos. En la figura 26 se muestra la puntuación que obtuvo cada indicador.

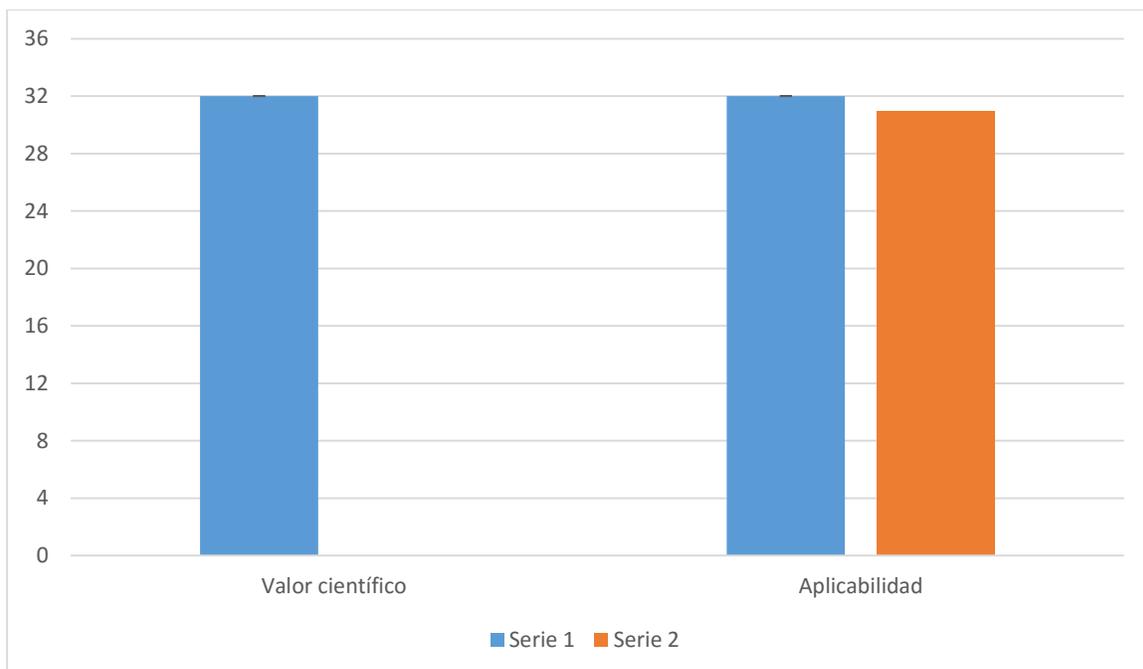


Figura 26. Suma de rango de valores asociados a los criterios de los expertos. Elaboración propia

Como se muestra en la Figura 26 todos los indicadores recibieron puntuaciones por encima del 80% de la máxima puntuación permisible. Lo que indica que de forma general el valor práctico de la aplicación ha sido aceptado por los expertos. La aplicabilidad fue uno de los indicadores que mayor puntuación recibió, lo cual demuestra que los expertos consideran que los resultados de la investigación son realmente prácticos y útiles.

3.5 Conclusiones parciales

Luego del desarrollo del presente capítulo se puede arribar a las conclusiones siguientes:

- Quedaron definidos los estándares de codificación con los que cumple la herramienta SIFE_AP.
- Se validó la aplicación realizada mediante las pruebas unitarias y de aceptación, así como por encuesta a los profesores de la facultad.
- Mediante las pruebas de funcionamiento realizadas se evaluaron los diferentes elementos de la aplicación y se comprobó que las funcionalidades implementadas mostraron un correcto funcionamiento y una respuesta a los requisitos funcionales garantizando las necesidades de los clientes.

Conclusiones Generales

Como resultado del estudio del estado del arte se concluye que la elaboración del marco teórico demuestra la necesidad de definir un modelo para la formación de equipos de trabajos en ambiente académico-profesional basado en el tamaño del equipo, las relaciones personales, el desempeño docente y la expectativa de éxito de los estudiantes para los roles a desempeñar dentro de un proyecto informático; que además esté soportado en una herramienta que facilita la formación de equipos de trabajo.

Del desarrollo de la solución se concluye lo siguiente:

Sobre la propuesta de solución:

- Se confeccionó un modelo teórico para la formación de equipos de trabajo a partir de cuatros elementos. El mismo puede ser utilizado dentro de los grupos docente con el objetivo de lograr mejores resultados en las tareas a realizar y que cada uno de los miembros del equipo sean capaces de desarrollar habilidades de trabajo en equipo desde su etapa de estudiante para desempeñarlas en su vida profesional.
- Se diseñó un algoritmo para la formación de equipos de trabajo de carácter homogéneo internamente y heterogéneo externamente.
- Se desarrolló una herramienta que soporta el modelo teórico propuesto en la investigación, la cual cumple con todos los requisitos definidos en la investigación.

Sobre los resultados obtenidos:

- Se validó la herramienta mediante la aplicación de pruebas unitarias y pruebas de aceptación demostrando que la aplicación SIFE_AP cumple con las funcionalidades identificadas.
- El valor práctico del modelo fue evaluado de aceptable por los expertos en relación a su valor científico y su aplicabilidad.

Recomendaciones

Realizar acciones futuras de desarrollo encaminadas a elevar los niveles de decisión del profesor sobre la composición final de los equipos.

Continuar enriqueciendo el modelo para la formación de equipos, considerando la riqueza que brinda las dimensiones definidas en la presente investigación.

Desarrollar futuras aplicaciones con el objetivo de automatizar el consumo del dato primario proveniente al test de Belbin.

Referencias Bibliográficas

- Achour, M. (2011). *Manual de PHP*. Obtenido de <http://php.net/manual/es/preface.php>
- Acuña, S. T., & Juristo, N. (2004). Assigning people to roles in software projects. *Software: Practice and Experience*.
- Alonso, M. Y., Vázquez, M. S., Blanco, V. G., & Ortiz, J. H. (2015). *Aplicación del aprendizaje cooperativo en terapia ocupacional*. Obtenido de <https://web.ua.es/es/ice/jornadas-redes-2015/documentos/tema-2/410751.pdf>
- Ampuero, M. A. (2015). Solución al problema de conformación de equipos de proyectos de software utilizando la biblioteca de clases BICIAM. *Revista cubana de Ciencias Informáticas*.
- André, M. (2009). Un modelo para la asignación de recursos humanos a equipos de proyectos de software. *Arquitectura del Software*. (s.f.). Obtenido de http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rivera_1_a/capitulo2.pdf
- Arza, L. (2013). Metodología para ubicar estudiantes en roles del proceso de desarrollo de software. La Habana.
- Arza-Pérez, L., Verdecia-Martínez, E. Y., & Lavandero-García, J. (2013). *Metodología para ubicar estudiantes en roles del proceso de desarrollo de software*. Obtenido de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-59362013000300008
- Aubin, V., Blautzik, L., Guatelli, R., Pafundi, F., Nisi, Z., & Zanga, M. (2015). *Desarrollo de las competencias de trabajo en equipo, dentro del proceso de enseñanza-aprendizaje de la programación en la educación superior*. Obtenido de <http://docplayer.es/14927786-Desarrollo-de-las-competencias-de-trabajo-en-equipo-dentro-del-proceso-de-ensenanza-aprendizaje-de-la-programacion-en-el-nivel-superior.html>
- Barcelona, U. d. (2016). *OBS Business School*. Obtenido de Criterios para seleccionar tu equipo de trabajo: <http://www.obs-edu.com/int/blog-project-management/factor-humano/criterios-para-seleccionar-a-tu-equipo-de-trabajo-en-un-proyecto>
- Basri, H. (2008). An Evaluation on Belbin's Team Roles Theory. *World Applied Sciences Journal*.
- Belbin. (25 de abril de 2010). *Home to belbin® team roles*. Obtenido de from <http://www.belbin.com/rte.asp?id=7>
- Belbin, M. (2010). *Management Teams: Why They Succeed or Fail*. Obtenido de <http://www.belbin.com/rte.asp?id=28>
- Briggs, i. l. (2004). Introducción al Type (MBTI).
- CALLEJA, A. M. (s.f.). Estándares de codificación.
- Capretz, L. F. (2003). *Personality types in software engineering*. Obtenido de [http://dx.doi.org/10.1016/S1071-5819\(02\)00137-4](http://dx.doi.org/10.1016/S1071-5819(02)00137-4).
- Castro, F. (2004). *Hacia un desarrollo sostenible en la Era de la Globalización*. La Habana: Científico-Técnica.
- Cattell, H. E., & Mead, A. D. (2008). The Sixteen Personality Factor Questionnaire. En *The Sage Handbook of Personality Theory and Assessment: Personality measurement and testing* (págs. 135-178).

- Chapman, A. (2011). *Personality theories, types and tests*. Obtenido de <http://www.businessballs.com/personalitystylesmodels.htm>
- Clements, P. (1996). A Survey of Architecture Description Languages. *Proceedings of the International Workshop on Software Specification and Design*.
- Cockburn, A. (2000). *Characterizing People as Non-Linear, First-Order Components in Software Development*. Obtenido de <http://alistair.cockburn.us/Characterizing+people+as+non-linear,+first>
- Corporation, O. (2011). *NetBeans Policies and Terms of Use*. Obtenido de https://netbeans.org/index_es.html
- Covey, S. R. (2000). *Trabajo en equipo*. Obtenido de <http://www.buenosaires.iipe.unesco.org/sites/default/files/modulo09.pdf>
- Delors, J. (1995). Los cuatro pilares de la educación. En *La educación encierra un tesoro* (págs. 91-103). Ediciones UNESCO.
- Devito, A., & Greathead, D. (2004). Code review and personality: is performance linked to MBTI type? UK: Newcastle: School of Computing Science.
- Dickey, R., & Olaosebikan, D. (mayo de 2016). *Several People Are Typing—The Official Slack Blog*. Obtenido de The science of teams: <https://slackhq.com/the-science-of-teams-e195af24c5d7>
- Diez, E. V., Zárraga-Rodríguez, M., & García, C. J. (2013). Obtenido de <http://dx.doi.org/10.3926/ic.399>
- Duarte, J. (2003). AMBIENTES DE APRENDIZAJE. UNA APROXIMACION CONCEPTUAL. En *Estudios pedagógicos* (págs. 97-113). Obtenido de http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-07052003000100007&nrm=iso
- Dyer, W. J. (2007). *Team Building*. San Francisco: JossyBass.
- Eguiluz, J. (2006). *Introducción a JavaScript*.
- Flores, B. S. (2015). *DISEÑO DE UNA ESTRATEGIA DIDÁCTICA DE TRABAJO EN EQUIPO*. Obtenido de <http://repositorio.pucesa.edu.ec/bitstream/123456789/1356/1/75743.pdf>
- Furnham, A. (2002). *Psicología organizacional*.
- García, L. M., González, D. A., Morales, M. d., & Magán, M. L. (2000). *El sociograma como técnica de intervención socioeducativa*. Obtenido de <http://revistas.ucm.es/index.php/CUTS/article/view/CUTS9898110165A>
- García, M. (2000). *Formacion de equipos de trabajo*. Obtenido de <http://eprints.uanl.mx/7631/1/1020130091.PDF>
- Github. (2014). *What is Symfony?* Obtenido de <https://github.com/symfony/symfony>
- Gonzalez, H. (2006). *La importancia del trabajo en equipo como factor de éxito en proyectos*. Obtenido de <http://www.belbin.com/rte.asp?id=28>
- GONZÁLEZ, H. (2006). *La importancia del trabajo en equipo como factor de éxito en proyectos*.
- Gorla, N., & Lam, Y. W. (2004). Who should work with whom?: building effective software project teams. 79-82.
- Hare, A. P. (2003). Roles, relationships, and groups in organizations: some conclusions and recommendations. Small group research.
- Hayes, N. (2003). *Dirección de equipos de trabajo. Una estrategia para el éxito*. Madrid.
- HUMPHREY, W. (2004). *Introduction to the Team of Software*.

- Humphrey, W. S. (2000). *The Personal Software Process*.
- JACA, C. (2011). Modelo de evaluación para la sostenibilidad de los equipos de mejora. En *Tesis Doctoral*. Navarra.
- Jiang, Z. (2007). The Variation of Software Development. En *Proceeding of World Academy of Science, Engineering and Technology* (págs. 355-359).
- JURISCO, N. M. (2005). *Técnicas de evaluación de software*. Obtenido de http://eva.uci.cu/file.php/257/Documentos/Recursos_bibliograficos/Libros_y_articulos_UD_3/Comun/Tecni
- Katzenbach, J. R. (2000). Obtenido de https://books.google.com.cu/books?id=phByqWOFpWEC&printsec=frontcover&hl=es&source=gbs_atb#v=twopage&q&f=false
- Kendall, K., & Babington-Smith. (1938). Obtenido de [https://www.google.com.cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwjL6IeLvMHUAhXDfRoKHfI0CYQQFggIMAA&url=http%3A%2F%2Fwww.spentamexico.org%2Fv9-n2%2FA5.9\(2\)31-40.pdf&usq=AFQjCNfFRU6OPw-1y1BeyL_m7cB_3maPeA&sig2=sBj7BCqO7w6qHbAYziW](https://www.google.com.cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwjL6IeLvMHUAhXDfRoKHfI0CYQQFggIMAA&url=http%3A%2F%2Fwww.spentamexico.org%2Fv9-n2%2FA5.9(2)31-40.pdf&usq=AFQjCNfFRU6OPw-1y1BeyL_m7cB_3maPeA&sig2=sBj7BCqO7w6qHbAYziW)
- Kuz, A., & Falco, M. (2013). *Herramientas sociométricas aplicadas al ambiente áulico*. Obtenido de <http://conaiisi.unsl.edu.ar/2013/93-456-1-DR.pdf>
- LARMAN, C. (2005). *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Mexico.
- Martinez, R. (2 de octubre de 2010). *obre PostgreSQL*. Obtenido de http://www.postgresql.org.es/sobre_postgresql
- Martín-Moreno, Q. (2004). Aprendizaje colaborativo y redes de conocimiento. En *Actas de las IX Jornadas Andaluzas de Organización y Dirección de Instituciones Educativas* (págs. 55-70).
- MATHIEU, J., MAYNARD, M., RAPP, T., & GILSON, L. (2008). Team effectiveness. *Journal of Management*,.
- McDonald, S., & Edwards, H. M. (2007). Who Should Test Whom? Examining the use and abuse of personality tests in software engineering. *Communications of the ACM*.
- Millhiser, W., & Solow, D. (2003). *Mathematical Models for Studying When to Divide a Team*. Cleveland: Department of Operations Weatherhead School of Management.
- Moreno. (2004). *Herramientas sociométricas aplicadas al ambiente áulico*.
- Network, M. D. (2014). *JavaScript*. Obtenido de <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Introducci%C3%B3n>
- Otto, M., & Thornton, J. (2015). *Bootstrap 3, el manual oficial*. Obtenido de http://librosweb.es/libro/bootstrap_3/
- PARADIGM, V. (2005). *Visual Paradigm Suite User's Guide*.
- Peiró, J. &. (1998). *Tratado de Psicología del Trabajo* Vol. 1.
- Pérez, E. M. (2004). Adaptación del Inventario de Personalidad 16PF-IPIP a un Contexto de Orientación.
- Perissé, M. (2006). *Una Metodología Simplificada*.
- Pineda, I., & Renero, L. (2009). Utilidad del sociograma como herramienta para el análisis de las interacciones grupales. *Psicología para América Latina*.
- Pleva, J. (2013). *PHP: Hypertext Preprocessor*. Wisconsin, Estados Unidos de América: Universidad de Winsconsin.

FORMACIÓN DE EQUIPOS DE TRABAJO EN AMBIENTE ACADÉMICO PROFESIONAL

- Pressman. (2006). *Ingeniería del software, un enfoque práctico*.
- Pressman, R. (2006). *Ingeniería del software, un enfoque práctico*.
- Pressman, R. (2006). Ingeniería del Software: Un enfoque práctico. *Revista de métodos cuantitativos para la economía y la empresa*.
- Prieto, M. (2010). Obtenido de <http://www.manuelprieto.com/publicaciones/gruposDeTrabajoDeInstalacionYDesa>
- Prieto, M. (2010). *Método Cuantitativo para Integración y Comparación de Grupos de Trabajo de Instalación y Desarrollo de Software*. Obtenido de <http://www.manuelprieto.com/publicaciones/gruposDeTrabajoDeInstalacionYDesa>
- Quintero, L. P. (2010). Modelo para la evaluación por competencias en proyectos informáticos de la Universidad de las Ciencias Informáticas.
- Rajendran, M. (2005). En *Analysis of team effectiveness in software development teams working on hardware and software environments using Belbin Self-perception Inventory* (págs. 738-753).
- Rodríguez, A. (2001). *La estructura interna de los grupos*.
- Rutherford, R. H. (2001). Using personality inventories to help form teams for software engineering class projects. 76-76.
- Sim, S., & Heng, A. (2006). *The relationship between team characteristics with team performance in Malaysian teams*. Sidney: Unpublished Master of Business.
- Sord, O. E. (2005). Algunas consideraciones acerca del ambiente académico.
- Stiven, E. (2012). *MODELO PARA LA EVALUACIÓN DE LA COMPOSICIÓN DE EQUIPO DE PROYECTOS*.
- Stiven, E. R. (2012). *MODELO PARA LA EVALUACIÓN DE LA COMPOSICIÓN DE EQUIPO DE PROYECTOS INFORMÁTICOS*.
- Team Building spanish*. (2007).
- Team, S. T. (2011). *Características de PHP*.
- Thomas, R. (1999). Group Dynamics and Software Engineering. *Paper presented at the Object Oriented Systems, Languages and Applications Conference*.
- Vega, O. Z. (2013). *Aplicación del test sociométrico como herramienta para el diagnóstico de grupo en apoyo a la tutoría académica*. Obtenido de <http://congresos.uaslp.mx/Documents/Tutorias/APLICACION%20DEL%20TEST.pdf>
- Xp., A. (2011). Obtenido de <https://Artefactos%20de%20XP/programacion-extrema2.shtml>
- Zaninetti, J. (2008). *Symfony: la guía definitiva*.