



Universidad de las Ciencias Informáticas
Centro de Consultoría y Desarrollo de Arquitecturas
Empresariales

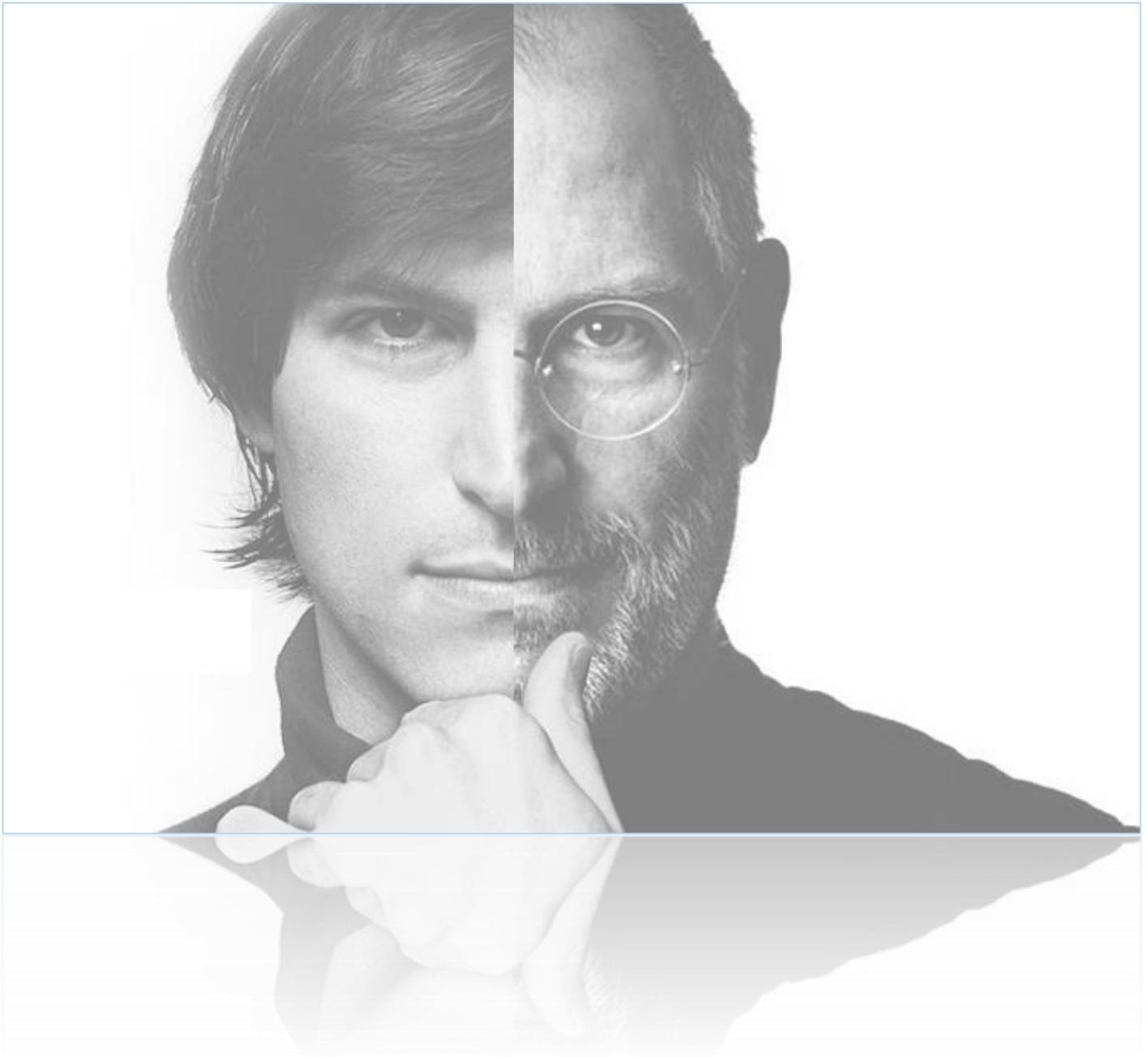
“Método para la evaluación multicriterio de recursos humanos basado en el modelo 2-tuplas para XEDRO-GESPRO 18.05”

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

Autora: Shirley Núñez Mullings

Tutores: Dra.C Surayne Torres López
Ing. Arisney Figueredo Ramos

2 de julio de 2018
“Año 60 de la Revolución”



“El único modo de hacer un gran trabajo es amar lo que haces”

Steve Jobs

*Método para la evaluación multicriterio de recursos humanos
basado en el modelo 2-tuplas para XEDRO-GESPRO 18.05.*



Declaración de autor

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2017.

Shirley Núñez Mullings

Firma de Autor

Dra.C Surayne Torres López

Firma de Tutora

Ing. Arisney Figueredo Ramos

Firma de Tutor

*Método para la evaluación multicriterio de recursos humanos
basado en el modelo 2-tuplas para XEDRO-GESPRO 18.05.*



Dedicatoria:

A mi familia.

*Método para la evaluación multicriterio de recursos humanos
basado en el modelo 2-tuplas para XEDRO-GESPRO 18.05.*



Agradecimientos:

Agradezco a mi familia por apoyarme siempre.

Resumen

En las organizaciones el éxito en el mercado depende principalmente de los conocimientos y competencias de sus empleados. Por tanto, le corresponde a la organización conocer y gestionar las competencias, con todo lo relacionado a evaluar, validar y desarrollarlas. Los procesos de evaluación de las competencias laborales de los miembros de las empresas suelen verse afectados por varios factores que atentan contra el éxito de ejecución y resultados. Actualmente son pocas las empresas que realmente recurren y completan las evaluaciones de las competencias de sus recursos humanos para los distintos fines en los cuales es de vital apoyo. De forma general, la investigación presenta un método que permite la participación de múltiples expertos en la evaluación de las competencias laborales en organizaciones orientadas a proyectos. Se propone para el tratamiento de la evaluación que proveen los expertos, emplear el enfoque basado en la fusión lingüística utilizando el Modelo de Representación Lingüística 2-tupla, teniendo en cuenta que es fácil de comprender y que permite la obtención de resultados lingüísticos interpretables. La propuesta permite la integración de la evaluación de las competencias laborales de los recursos humanos con los distintos procesos de la gestión de recursos humanos, contribuye a la toma de decisiones y favorece los procesos de la gestión por competencias dando tratamiento a la incertidumbre a través de técnicas de computación con palabras.

Palabras claves

Competencias, GESPRO, Gestión de proyectos, Gestión de recursos humanos, Modelo de Representación Lingüística, Resultados lingüísticos.

Abstract

In organizations, success in the market depends mainly on the knowledge and skills of their employees. Therefore, it is up to the organization to know and manage the competences, with everything related to evaluate, validate and develop them. The processes of evaluation of the labor competencies of the members of the companies are usually affected by several factors that threaten the success of execution and results. Currently, there are few companies that actually resort and complete the evaluations of the competencies of their human resources for the different purposes in which it is of vital support. In general, the research presents a method that allows the participation of multiple experts in the evaluation of labor competencies in project-oriented organizations. It is proposed for the treatment of the evaluation that the experts provide, to use the approach based on linguistic fusion using the 2-tuple Linguistic Representation Model, taking into account that it is easy to understand and that it allows obtaining interpretable linguistic results. The proposal allows the integration of the evaluation of the labor competencies of human resources with the different processes of human resources management, contributes to the decision making and favors the processes of the management by competences giving treatment to the uncertainty through computer techniques with words.

Keywords

Competencies, GESPRO, Human Resources Management, Linguistic Representation Model, Linguistic Results, Project Management.

Índice general

Introducción.....	1
Capítulo 1: Fundamentación teórica del sistema.....	6
1.1. Introducción.....	6
1.2. Conceptos relacionados al dominio del problema.....	6
1.2.2. Evaluación por competencias.	7
1.2.3. Problemas de toma de decisiones durante la evaluación de competencias.....	8
1.2.4. Técnicas y modelos para la evaluación de competencias.....	8
1.2.5. Método de evaluación de competencias de los recursos humanos basado en el modelo de representación lingüística con 2-tuplas.	10
1.3. Metodología, tecnologías y herramientas a emplear en la solución.....	11
1.3.1. Metodología de desarrollo de software.....	11
1.3.2. Lenguaje unificado de modelado.	12
1.3.3. Herramientas CASE.....	13
1.3.4. Lenguaje de programación.	13
1.3.5. Marco de trabajo.	14
1.3.6. Entorno de desarrollo integrado.....	15
1.3.7. Sistema gestor de bases de datos (SGBD).....	15
1.3.8. Servidor web.	16
1.4. Conclusiones parciales.....	17
Capítulo 2: Análisis y diseño del sistema.	18
2.1. Introducción.....	18
2.2. Modelo del dominio.	18
2.3. Propuesta del sistema.....	19
2.3.1. Gema whenever	19
2.4. Descripción de la solución propuesta.....	19

2.5. Especificación de los requisitos del software.	26
2.6. Historia de usuario.	29
2.7. Pila de Productos.	30
2.8. Pila de Sprint.	30
2.9. Arquitectura del sistema.	32
2.10. Diagrama de clases del diseño	35
2.11. Modelo de datos.	36
2.12. Conclusiones parciales.	37
Capítulo 3: Implementación y pruebas a la solución propuesta.	39
3.1. Introducción.	39
3.2. Diagrama de despliegue.	39
3.3. Implementación.	40
3.4. Estándares de codificación.	40
3.4.1. Convenciones de nomenclatura.	40
3.5. Pruebas al sistema.	41
3.5.1. Pruebas de caja negra.	41
3.5.2. Pruebas de aceptación.	43
3.6. Resultados de las pruebas.	43
3.7. Técnicas de ladov.	44
3.8. Conclusiones parciales.	45
Conclusiones Generales	47
Recomendaciones	48
Bibliografía	49
Anexos	54

Índice de tablas

Tabla 1 Preferencias lingüísticas de los múltiples expertos.....	22
Tabla 2 Valores colectivos de los criterios para cada relación usuario-competencia.....	24
Tabla 3 Valores colectivos de las competencias.....	26
Tabla 4 Especificación de requisitos funciones del sistema.....	27
Tabla 5 Historia de usuario “Listar usuarios a evaluar”.....	29
Tabla 6 Pila de Productos.....	30
Tabla 7 Pila de Sprint para la tarea Listar usuarios a evaluar.....	31
Tabla 8 Pila de Sprint para la tarea Evaluar usuario.....	31
Tabla 9 Pila de Sprint para la tarea Eliminar Evaluación.....	31
Tabla 10 Pila de Sprint para la tarea Ver detalles.....	31
Tabla 11 Pila de Sprint para la tarea exportar a pdf.....	31
Tabla 12 Pila de Sprint para la tarea exporta a csv.....	31
Tabla 13 Pila de Sprint para la tarea asignar pesos a los expertos.....	32
Tabla 14 Caso de prueba del RF: Listar usuarios a evaluar.....	42
Tabla 15 Prueba de aceptación sobre la Historia de usuario Listar usuarios a evaluar.....	43
Tabla 16 Historia de usuario “Evaluar usuario”.....	54
Tabla 17 Historia de usuario “Eliminar Evaluación”.....	54
Tabla 18 Historia de usuario “Ver detalles”.....	55
Tabla 19 Historia de usuario “Exportar a pdf”.....	56
Tabla 20 Historia de usuario “Exportar a csv”.....	56
Tabla 21 Historia de usuario “Asignar pesos a los expertos”.....	57
Tabla 22 Caso de prueba del RF: Evaluar a usuarios.....	57
Tabla 23 Caso de prueba del RF: Eliminar evaluación.....	58
Tabla 24 Caso de prueba del RF: Ver detalles.....	59
Tabla 25 Caso de prueba del RF: Exportar a pdf.....	59
Tabla 26 Caso de prueba del RF: Exportar a csv.....	60
Tabla 27 Caso de prueba del RF: Asignar pesos a los expertos.....	60
Tabla 28 Prueba de aceptación sobre la Historia de usuario Evaluar usuario.....	60
Tabla 29 Prueba de aceptación sobre la Historia de usuario Eliminar evaluación.....	61
Tabla 30 Prueba de aceptación sobre la Historia de usuario Ver detalles.....	62
Tabla 31 Prueba de aceptación sobre la Historia de usuario Exportar a pdf.....	63
Tabla 32 Prueba de aceptación sobre la Historia de usuario Exportar a csv.....	63

Tabla 33 Prueba de aceptación sobre la Historia de usuario Asignar pesos a los expertos.64

Índice de figuras

Fig. 1 Fases del proceso de AD en las que se basa el método propuesto (Clemen, 1995).....	10
Fig. 2 Modelo de dominio.....	18
Fig. 3 Función de pertenencia triangular.....	21
Fig. 4 Variable lingüística para la evaluación de competencias.....	21
Fig. 5 Funcionamiento del MVC.....	33
Fig. 6 Diagrama de clases.....	35
Fig. 7 Modelo de datos.....	37
Fig. 8 Diagrama de despliegue.....	39
Fig. 9 Ejemplo de uso de estándares de codificación.....	40
Fig. 10 Relación de no conformidades detectadas en las pruebas.....	44

Introducción.

En la actualidad la competencia global afecta a la mayoría de las empresas y organizaciones y, por tanto, mantenerse competitivas en ese entorno es su objetivo principal. Tanto es así que la mayoría de las empresas son conscientes de que conseguir este objetivo depende de un desarrollo continuo de sus recursos humanos (Andres, 2009).

En las empresas orientadas a proyectos donde la mayoría de los recursos humanos están concentrados en proyectos, el éxito en el mercado depende principalmente de los conocimientos y desempeño laboral de sus empleados. Ante la necesidad de encontrarle soluciones a los problemas de gestión y desempeño de sus empleados, estas entidades optan por la aplicación de sistemas de recursos humanos basados en las competencias laborales. Este método de gestión de los recursos humanos ha mostrado que las competencias laborales son fundamentales para el desarrollo y sostenibilidad de las entidades (Tejada, 2005).

La evaluación basada en competencias es un proceso que permite obtener evidencias sobre el desempeño de una competencia, siendo el resultado un juicio sobre si la persona que aprende ha conseguido dominar y poner en práctica la idoneidad requerida. Este proceso toma como referente principal, el desempeño, en el cual los conocimientos, habilidades y las actitudes son valorados en relación a los requerimientos del puesto.

El enfoque de competencia laboral surge en un marco de transformación de la producción y del trabajo, y de nuevas exigencias respecto a la forma de desempeño del individuo en el sitio de trabajo. Ser más competitivo no sólo requiere atender mecánicamente las necesidades del mercado de trabajo, sino también las propias del individuo, su formación integral. Poseer los conocimientos fundamentales, las habilidades sociales y las actitudes que permitan al individuo resolver problemas y enfrentar situaciones de contingencia, así como transferir su saber, su saber-hacer y su saber-ser a distintos contextos. Este es el sentido de la competencia laboral.

Contar con un método de evaluación ayuda a las empresas a descubrir necesidades de formación, comprobar si las técnicas de selección han sido eficientes, ofrecer retroalimentación al empleado y crear nuevas formas de motivación. Hay que tener claro que cualquier sistema de medición del rendimiento del personal debe servir para que el trabajador mejore porque si esto ocurre, la organización también mejorará.

En la actualidad, la mejor manera de medir el desempeño de un colaborador es por medio de una evaluación de desempeño, herramienta que realiza un análisis profundo y afinado acerca del nivel de rendimiento de un trabajador. La evaluación de desempeño no es 100% exacta porque parte de la información, se basa en la percepción del evaluador que por lo general la expresa en términos cuantitativos y no cualitativos, dando valores matemáticos al comportamiento del individuo en la organización. Por ello, para tener una herramienta mucho más efectiva es necesario considerar otros aspectos que contribuyan a cerrar las brechas de la percepción: los objetivos y las funciones de las personas puestas a prueba.

(Gil, 2007) Opina que la evaluación de competencias intenta conocer entre otros aspectos el rendimiento de sus miembros o su potencial de desarrollo, donde desde el punto de vista administrativo la evaluación sirve a la selección y asignación del personal adecuado para los puestos de trabajo dentro de la empresa, a la toma de decisiones salariales o las estrategias de desarrollo o acciones formativas, entre otros.

Se reportan en la literatura investigaciones que se apoyan en las ciencias informáticas para la aplicación de técnicas de inteligencia artificial en la evaluación de competencias o del desempeño de los recursos humanos. Tales son los casos de las utilizadas en la minería de datos, árboles de decisión, lógica borrosa, redes neuronales y computación con palabras (Xiaofan, 2010) (Jiang, y otros, 2010) (Jing, 2009) (Wang, y otros, 2010) y (Herrera, y otros, 2001). Estas propuestas dan soluciones novedosas, pero comparten la característica de requerir de la evaluación por expertos de determinados índices o rasgos definidos o de contar con los resultados de otras técnicas de evaluación invasivas.

(Andrés, 2009) Propone un modelo lingüístico de evaluación del desempeño de 360-grados, enfocado en el tratamiento de los problemas de toma de decisiones multi-experto con información lingüística multi-granular. Sin embargo, aunque no sigue la filosofía de utilización de los indicadores de la gestión de proyectos y permite que diferentes escalas lingüísticas pertenezcan al marco de evaluación, sirve como fundamento para la elaboración de un método de evaluación de las competencias laborales en entidades orientadas a proyectos. Se trata de un método de valoración de las habilidades y competencias del empleado mediante el que se le informa de las competencias en las que destaca y en las que deben mejorar. La evaluación 360-grados requiere de toda la información general que pueden aportar los jefes, los colaboradores, los subordinados y los elementos externos sobre la idoneidad del evaluado. Con ello se logra conocer el desempeño del evaluado a través de los criterios ofrecidos por todos los relacionados con el trabajador en cuestión, y así detectar áreas de oportunidad del individuo, del equipo y de la organización y llevar a cabo acciones precisas para mejorar el desempeño del personal. Medir y evaluar el rendimiento de los trabajadores es imprescindible para comprobar el grado de cumplimiento de los objetivos propuestos a nivel

individual y para poder potenciar las habilidades de cada uno de ellos y corregir lo que se esté haciendo mal.

En el año 2010 la Universidad de las Ciencias Informáticas (UCI), centro de estudio que tiene entre sus misiones producir aplicaciones y servicios informáticos; desarrolló la herramienta de Gestión de Proyectos XEDRO-GESPRO como plataforma extensible, la cual posee un conjunto de áreas de procesos que donde incluye módulos para la gestión documental, de alcance, tiempo, riesgos, calidad, costo, recursos humanos, entre otros. Esta aplicación actualmente navega por su versión 17.05, usada en la red de centros de desarrollo de la universidad. En lo relacionado con la Gestión de recursos humanos tiene subsistemas relacionados con planificación, la selección, desarrollo y administración del personal. Este último subsistema está dirigido a la evaluación del desempeño de los trabajadores, de la demostración de sus competencias laborales.

En el tema de Evaluación de Competencias XEDRO-GESPRO ofrece un método evaluativo donde se recoge la apreciación de una sola persona, que por lo general está basado en una medida cuantitativa con valores matemáticos. Esta evaluación no ofrece una medida real que dé la medida de apreciación verídica del cumplimiento de la competencia por el evaluado, porque ningún sujeto es 5 bien o 2 mal. Se corre el riesgo de la subjetividad y errores de apreciación, que ocurren cuando se solicita la valoración de un solo sujeto evaluador, por permitir que la misma esté sesgada por el criterio particular del mismo, abarcando para la evaluación una sola arista; la que percibe ese experto, lo que disminuye la fiabilidad de esa apreciación.

Debido a la necesidad de dar solución a la situación planteada se identifica como **problema de la investigación** que la Suite de Gestión de Proyectos XEDRO-GESPRO 17.05 no permite la total fiabilidad de los resultados de la evaluación de competencia de los recursos humanos por reconocer el criterio de un solo evaluador o experto.

El **objeto de estudio** está dirigido a la evaluación de competencias, enmarcado en el **campo de acción** la evaluación de las competencias laborales en la plataforma XEDRO-GESPRO.

Para dar solución al problema planteado se propone como **objetivo general** presentar un método para aplicar e implementar en la Suite de Gestión de Proyectos XEDRO-GESPRO que permita la participación de múltiples expertos en la evaluación de las competencias laborales de los trabajadores en organizaciones orientadas a proyectos.

En este método se propone para el tratamiento de la evaluación que proveen los expertos, emplear el enfoque basado en la fusión lingüística utilizando el Modelo de Representación Lingüística 2-tupla, teniendo en cuenta que es fácil de comprender y que permite la obtención de resultados lingüísticos interpretables.

Tareas de la investigación:

1. Realización del marco conceptual para precisar los principales conceptos que se emplean en la investigación.
2. Estudio del estado del arte de las herramientas existentes para sintetizar los resultados alcanzados en la revisión bibliográfica e indagaciones realizadas relacionadas con la evaluación de competencia.
3. Selección de la metodología para definir los métodos y técnicas necesarias que guiarán el desarrollo de las funcionalidades para la evaluación de las competencias laborales utilizando el método 2-tuplas.
4. Descripción de las herramientas, tecnologías y lenguajes a utilizar para desarrollar el módulo para la evaluación de las competencias laborales utilizando el método 2-tuplas.
5. Definición de los requisitos funcionales y no funcionales para identificar las capacidades que tienen que ser alcanzadas por el sistema para cumplir los objetivos trazados.
6. Descripción de la arquitectura que soporta la implementación de las funcionalidades.
7. Realización del análisis y diseño del mecanismo para describir los artefactos.
8. Implementación de las funcionalidades que dan cumplimiento a los requisitos identificados.
9. Realización de pruebas al módulo en un entorno controlado para valorar la calidad del producto implementado.
10. Valoración de los resultados obtenidos.

Para el desarrollo de la investigación se utilizan los siguientes métodos de investigación científica:

Métodos teóricos:

- **Analítico - Sintético:** para la realización de la investigación se analizan y confrontan los criterios de disímiles autores sobre los conceptos obtenidos acerca de los hechos que se relacionan con la evaluación de las competencias, partiendo de la descomposición del objeto de estudio en cada una de sus partes para estudiarlas en forma individual. Con los conocimientos obtenidos sobre el tema se procedió a sintetizar los resultados de la investigación para desarrollar la solución propuesta.
- **Histórico - Lógico:** la utilización de este método permite obtener una breve reseña sobre la evolución, desarrollo y funcionamiento de los sistemas de evaluación de competencias laborales, investigar las leyes generales que los norman y establecer la relación actual con el desarrollo de las tecnologías informáticas, así como confrontar los elementos extraídos de las fuentes consultadas para establecer los fundamentos teóricos de la investigación.

- **Modelación:** se utiliza para modelar y representar la evaluación de las competencias, supone una aproximación intuitiva a la realidad y que tiene por función básica la de ayudar a comprender las teorías y las leyes.

Métodos empíricos:

- **Entrevista:** con la utilización de este método se establece una conversación con algunos responsables del módulo de los recursos humanos del Departamento de Gestión de Proyectos y la jefa del centro recopilando toda la información necesaria para la gestión de recursos humanos y la evaluación de las competencias en la plataforma XEDRO-GESPRO.

El trabajo de diploma se estructura de la siguiente manera: introducción, tres capítulos que serán descritos a continuación, conclusiones, recomendaciones, bibliografía y anexos.

Capítulo 1: Fundamentación teórica del sistema.

En este capítulo se presenta la definición del marco teórico de la investigación. En el mismo se abordan conceptos que son esenciales para la comprensión del problema de la investigación. Contiene además un análisis de la metodología, herramientas, tecnologías y lenguajes a utilizar en el desarrollo del sistema.

Capítulo 2: Análisis y diseño del sistema.

En el capítulo se describe la propuesta de solución al problema de la investigación. Abarca lo referente al modelo de dominio. Se realiza el levantamiento de los requisitos funcionales y no funcionales, los diagramas de clases del diseño por medio de los cuales se representa la estructura estática del sistema y los patrones de arquitectura y diseño utilizados.

Capítulo 3: Implementación y pruebas a la solución propuesta.

En el capítulo se aborda lo referente a la implementación de la solución propuesta. Se diseñan y aplican las pruebas para comprobar el correcto funcionamiento del método.

Capítulo 1: Fundamentación teórica del sistema.

1.1. Introducción.

En este capítulo se exponen diferentes conceptos importantes para la comprensión del dominio del problema objeto de investigación. Se profundiza en lo relacionado con las dificultades en la toma de decisiones durante la evaluación de competencias, técnicas y modelos utilizados para ello, así como el método de evaluación de competencias de los recursos humanos basado en el modelo de representación lingüística con 2-tuplas. Se aborda además un estudio de la metodología, herramientas, tecnologías y lenguajes a emplear en el desarrollo de la solución del problema.

1.2. Conceptos relacionados al dominio del problema.

En esta sección se ofrecen algunos conceptos para el mejor entendimiento y comprensión de la investigación.

1.2.1. Gestión por competencias laborales.

En (Gil, 2007) el autor opina que la consideración de las competencias parte de las características y comportamiento de los recursos humanos que realizan con eficacia las tareas de un puesto de trabajo. Estas involucran tanto los rasgos psicológicos, como conductas observables que son el resultado de actitudes, valores, conocimientos y habilidades.

Por otra parte (Tejada, 2005) opinan, que la competencia es un conjunto de elementos integrados que deben ser evaluados para desarrollar su utilidad y la evolución de las competencias profesionales. La competencia profesional se proyecta en un contexto cambiante, en constante evolución haciendo necesaria su evaluación y adaptación al entorno de trabajo. Estas pueden ser obtenidas durante la vida activa, formando un componente capital de flexibilidad y de adaptación a la evolución de los puestos de trabajo. Dicha adaptación se presenta en el desempeño eficaz, efectivo y exitoso, lograr la colaboración y resolución de problemas.

Las competencias como base de la gestión de recursos humanos en las organizaciones permiten el dominio de un lenguaje común para todos sus miembros, centran los esfuerzos los trabajadores hacia la obtención de resultados, favorece la predicción de la conducta futura de los empleados sobre la base de su comportamiento pasado y propicia la colación entre el perfil de exigencias del puesto y el perfil competencial de las personas (Pereda, 2004).

Cruz y Vega definen la gestión por competencias como un “modelo de gestión que permite evaluar las competencias específicas que requiere un puesto de trabajo de la persona que lo ejecuta, además, es una herramienta que permite flexibilizar la organización, ya que logra separar la organización del trabajo de la gestión de personas, introduciendo a estas como actores principales en los procesos de cambio de las empresas y finalmente, contribuir a crear ventajas competitivas de la organización” (Cruz, y otros, 2001).

Entre las ventajas aportadas por la gestión por competencias laborales en (Tejada, 2005) los autores destacan entre otras:

- Flexibilidad para que la organización se adapte ante cambios de clientes y del mercado, a través del despliegue de los empleados basado en el modelo de competencias.
- Proporciona a los empleados oportunidades para adquirir y aplicar nuevos conocimientos y habilidades.
- Los empleados conocen lo que se espera de ellos en el puesto actual y que competencias necesitan para desarrollarse.

De esta manera se puede decir que la gestión por competencias profesionales de los recursos humanos en una organización se basa principalmente en el desarrollo, evaluación, seguimiento y control de las competencias profesionales que tienen sus empleados. Proporciona una fuente de conocimientos sobre sus capacidades y viabiliza sus procesos de toma de decisiones.

1.2.2. Evaluación por competencias.

Sobre la evaluación de competencias Gil entiende como el proceso de recolección de la información de las competencias desarrolladas por un trabajador, las cuales son comparadas con el perfil de competencias definido para el puesto de trabajo que desempeña. Esta se encuentra presente durante toda la trayectoria profesional del recursos humanos como una forma de evaluar su desempeño laboral en el puesto que desarrolla (Gil, 2007).

El desempeño laboral de los recursos humanos en la empresa demuestra las competencias profesionales con las que cuenta para ejercer cabalmente en el puesto de trabajo, y de esta forma cumplir eficaz y eficientemente con las tareas que le corresponden. Por tanto, su evaluación basada en las evaluaciones de las competencias profesionales demuestra mediante el análisis de los resultados productivos y comportamientos, la medida en que el trabajador es competente e idóneo para el desempeño adecuado en el rol jugado.

A través de la evaluación de las competencias se logra un mejor conocimiento y comprensión de la conducta laboral de los recursos humanos, basándose en el análisis de sus habilidades, deficiencias y posibilidades de desarrollo. Su ejecución permite la selección de personal competente para un puesto de trabajo y el seguimiento y control del desempeño de este. También propicia información que fomenta decisiones tales como planes estratégicos para la mejora continua y reconocimiento salariales de los empleados en la organización.

1.2.3. Problemas de toma de decisiones durante la evaluación de competencias.

De acuerdo con lo planteado (Andrés, 2009) la evaluación de competencias laborales se presenta como un problema de toma de decisiones.

La evaluación presenta como conflicto la emisión de valoraciones por parte de los evaluadores que concuerden con el grado de cumplimiento de las competencias reales del evaluado. Se ve afectada por la capacitación o las competencias que poseen los evaluadores para la correcta ejecución de su rol como evaluador. Específicamente cuando los encargados del proceso son miembros de la misma organización la evaluación puede sufrir alteraciones dependiendo de las relaciones interpersonales entre compañeros de trabajo.

El rol de evaluador requiere de conocimiento y experiencia sobre el puesto de trabajo al que evalúa. La conducción de la evaluación de las competencias por un evaluador que no se considere un experto en el área que evalúa hace que el proceso sea vulnerable ante varios errores, provocando situaciones engorrosas y comprometer los resultados de la evaluación.

Por tanto, concordando con (Gil, 2007) mitigar estos problemas es posible a través de la sistematización de la evaluación de competencias, automatizar y establecer modelos y técnicas que incluyan la recolección de las evidencias del trabajo realizado por los recursos humanos durante el desempeño de un puesto de trabajo.

1.2.4. Técnicas y modelos para la evaluación de competencias.

Los procesos de evaluación del desempeño laboral basados en las competencias profesionales de los recursos humanos suelen combinar técnicas psicológicas tradicionales con las experiencias laborales y resultados obtenidos durante el desempeño en el puesto de trabajo.

En (Gil, 2007) son presentados varios métodos agrupados en distintas clasificaciones que suelen emplearse para guiar los procesos de evaluación de competencias profesionales. Tales son las técnicas: basadas en el análisis de la experiencia práctica; basadas en características y experiencias de los sujetos; y las técnicas

basadas en valoraciones. Empleando y combinando el enfoque que proponen estas técnicas varios autores han propuesto técnicas y modelos de evaluación de desempeño basado en las competencias profesionales.

En (Cuesta, 2011) el autor propone realizar desde la planeación estratégica hasta el control de la gestión de las competencias, orientando los pasos a seguir en cada caso. Plantea la inclusión de dimensiones para las competencias que aterricen o las hagan observables, elemento que se considera relevante para su aplicación. Establece tres niveles para la gestión de las competencias: competencias organizaciones o distintivas, competencias de procesos de trabajo y competencias laborales a nivel de los puestos de trabajo. Es una investigación que está orientada a las empresas cubanas en general y no está soportada por alguna herramienta que automatice su utilización.

También el modelo para la asignación de recursos humanos a equipos de proyectos de software (Andres, 2009) propone un modelo formal para la asignación de recursos humanos a equipos de proyectos de software que considera factores que contribuyan a la asignación individual a los roles del proyecto y a la formación del equipo (competencias, carga de trabajo, costo por lejanía y características psicológicas para el trabajo en equipo). Tiene como característica relevante la identificación de 16 roles invariantes del proceso de desarrollo de software, clasificados en Roles técnicos y Roles de gestión (Ampuero, 2007). Se determina además un conjunto de competencias asociadas a los roles definidos y clasificadas en genéricas y estratégicas, identificando el nivel de importante de cada competencia en los roles.

Por otra parte, se han diseñado e implementado algoritmos basados en la minería de datos para la evaluación de competencias de los recursos humanos. En (Xiaofan, 2010) se define la gestión del desempeño como un subsistema de la gestión empresarial que constituye un mecanismo importante para alcanzar los objetivos en los diferentes niveles de la organización y establecer su plan de ejecución. Este algoritmo si bien incluye la aplicación de los árboles de decisión como técnica de minería de datos en la gestión del desempeño de los recursos humanos, no está enfocado a la evaluación del desempeño ni de las competencias.

(Wang, y otros, 2010) Propone 14 índices para caracterizar y evaluar a los recursos humanos. El modelo trabaja con valores numéricos para caracterizar los rasgos de las personas, elemento desventajoso para los expertos que evalúan cada uno de los atributos del desempeño ya que estos utilizan normalmente etiquetas lingüísticas. No tiene en cuenta la incertidumbre y la vaguedad de la información aspecto propio a los datos relacionados con el desempeño de los recursos humanos. Esto constituye un inconveniente en el momento de discutir las evaluaciones con las personas ya que complejiza la posibilidad de explicar y demostrar una decisión.

(Andrés, 2009) Define un modelo básico de evaluación del desempeño de 360-grados fundamentado en la Teoría de la Toma de Decisiones, que permita realizar este proceso de manera formal y estructurada. En la investigación se realizan dos propuestas: un modelo de evaluación del desempeño basado en técnicas de Programación por Metas y un modelo lingüístico de evaluación del desempeño de 360-grado, en el cual las opiniones de los evaluadores son expresadas mediante diversas escalas cualitativas definidas a través de términos lingüísticos y utiliza el modelo de representación de la información lingüística basado en 2-tuplas. En el caso del modelo de evaluación del desempeño basado en técnicas de Programación por Metas, las valoraciones emitidas por los diversos evaluadores serán numéricas, introduciendo un proceso de agregación basado en técnicas propias de la Programación por Metas. Ofreciendo mayor flexibilidad en la agregación de las evaluaciones, permitiéndoles clasificar y ordenar a los empleados. La segunda propuesta, el modelo de evaluación del desempeño de 360-grados con múltiples escalas lingüísticas, basado en el enfoque clásico de los procesos de análisis de decisión. En ambos casos se requiere involucrar múltiples evaluadores para lograr la evaluación.

1.2.5. Método de evaluación de competencias de los recursos humanos basado en el modelo de representación lingüística con 2-tuplas.

El método tiene 2 precondiciones indispensables para que los resultados de su aplicación sean satisfactorios.

- Las competencias laborales en la organización deben estar definidas, ser observables y medibles y ser reflejo verdadero del desempeño requerido en la organización y proyecto para alcanzar sus objetivos (ONN, 2007).
- Deben existir evidencias de las actividades asignadas a los recursos humanos de la organización. De ello depende el cálculo de los indicadores que constituyen la base para la caracterización y evaluación de las competencias laborales.

Como se muestra en la figura 1, la estructuración del método propuesto se basa en 3 de las fases que propone el Análisis de Decisión (AD), a continuación se muestran adaptadas a la evaluación de competencias de los recursos humanos:

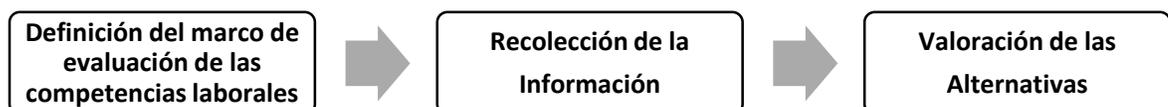


Fig. 1 Fases del proceso de AD en las que se basa el método propuesto (Clemen, 1995).

1.3. Metodología, tecnologías y herramientas a emplear en la solución.

Para el desarrollo de un software se deben tener en cuenta ciertos elementos, como son la metodología, tecnologías y las herramientas básicas para su construcción. A continuación, se describen las principales características de las utilizadas.

1.3.1. Metodología de desarrollo de software.

La metodología para el desarrollo de software en un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Una metodología para el desarrollo de software comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto software desde que surge la necesidad del producto hasta que se cumple el objetivo por el cual fue creado. Una definición estándar de metodología puede ser el conjunto de métodos que se utilizan en una determinada actividad con el fin de formalizarla y optimizarla. Determina los pasos a seguir y cómo realizarlos para finalizar una tarea (Inteco, 2010).

SCRUM.

Es una metodología nos permite encontrar prácticas emergentes en dominios complejos, como la gestión de proyectos de innovación. En lugar de proporcionar una descripción completa y detallada de cómo deben realizarse las tareas de un proyecto, genera un contexto relacional e iterativo, de inspección y adaptación constante para que los involucrados vayan creando su propio proceso. Esto ocurre debido a que no existen ni mejores ni buenas prácticas en un contexto complejo. Es el equipo de involucrados quien encontrará la mejor manera de resolver sus problemáticas. Este tipo de soluciones serán emergentes. (Alaimo, 2013). El departamento de Gestión de Proyectos utiliza esta metodología debido a que el personal está organizado en pequeños equipos de trabajo maximizando así la comunicación, el intercambio de conocimientos e información, reduciendo así, al mínimo los gastos generales. Este departamento desarrolla proyectos en entornos complejos, en los cuales necesitan obtener resultados favorables y rápidos, y donde los requisitos son cambiantes o poco definidos.

Pila de productos.

La Pila de Producto es el instrumento metodológico del marco de trabajo Scrum, que se usa para listar las características o funcionalidades del software a desarrollar, para priorizarlas de acuerdo a las necesidades del área de negocio. Su contenido se desarrolla a partir de las historias de usuario identificadas por el dueño de producto. La pila de producto permitirá tener visualización de las funcionalidades a desarrollar, priorizar

las características del software según las necesidades del negocio, dejar registrado el esfuerzo necesario para desarrollar la historia y asignarla a una iteración (PMOinformatica.com, 2016).

Pila de sprint.

La metodología Scrum divide el proceso de desarrollo de software en una serie de iteraciones, denominadas Sprints. Al comienzo de la iteración se realiza una reunión de planificación (Sprint Planning), donde se define la lista de tareas de la iteración (Sprint Backlog). En Scrum la plantilla de lista de tareas de la iteración (Sprint Backlog), es una lista de los elementos de la pila de producto (Product Backlog) y sus tareas componentes, seleccionados por el equipo Scrum para la iteración. Por ende, para elaborar la lista de tareas de la iteración, es necesario tener definidas previamente la pila de producto (Product Backlog) y los elementos del product backlog, los cuales pueden estar documentados en la forma de historias de usuario (Existen otras formas, pues Scrum no prescribe ninguna) (PMOinformatica.com, 2016).

1.3.2. Lenguaje unificado de modelado.

Lenguaje unificado de modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group).

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados (Disca, 2012).

Es importante remarcar que UML es un lenguaje de modelado para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir.

UML 2.1

Se está convirtiendo rápidamente en el estándar aceptado para especificar, documentación y visualización de sistemas de software. También se utiliza para el modelado de sistemas no de software, y se implementa ampliamente en la mayoría de los sectores de la industria, incluyendo las finanzas, militar y de ingeniería. (uml-diagrams, 2010).

Se utiliza este lenguaje de modelado por proporcionar orientación en cuanto al orden de las actividades de un equipo, dirigir las tareas de los desarrolladores individuales y del equipo en su conjunto, por ofrecer criterios para el seguimiento, la medición de los productos y las actividades de un proyecto.

1.3.3. Herramientas CASE.

Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas son de gran ayuda en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras (Martínez, 2016).

Visual Paradigm 8.0.

Visual Paradigm (anteriormente VP-UML) es una herramienta UML. La herramienta está diseñada para una amplia gama de usuarios, incluyendo ingenieros de software, analistas de sistemas, analistas de negocios y arquitectos de sistemas, o para cualquier persona que esté interesada en la construcción de forma fiable los sistemas de software a gran escala con un enfoque orientado a objetos (Paradigm, 2010).

Se escoge la herramienta Visual Paradigm por soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas, sirviendo como guía y facilitando el proceso de desarrollo de todas las fases de creación del módulo (Stewart, 2016).

1.3.4. Lenguaje de programación.

Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos de hardware y software existentes. (Kaneiwa, y otros, 2015).

Ruby 2.3.1.

Ruby es un lenguaje sencillo y flexible que atrae a programadores de todos los sectores y que promete una grata experiencia en el trabajo habitual, además es un lenguaje multipropósito que permite desarrollos en áreas como; aplicaciones comerciales, acceso a base de datos, proceso y transformación de XML, aplicaciones distribuidas y aplicaciones web. Como lenguaje es moderno y orientado a objetos que combina una importante flexibilidad con alta productividad. Incorpora algunas de las mejores características de otros

lenguajes como Smalltalk, Java y Perl. Es multiplataforma ya que se integra perfectamente a grandes arquitecturas, funcionando incluso en dispositivos móviles. Tiene un alcance ilimitado y hoy se encuentra presente en aplicaciones que van desde el desarrollo web hasta la simulación de ambientes complejos sin perder usabilidad. Permite utilizar la más simple expresión para un programa o algoritmo, esto sumado a las actuales prácticas ágiles permite desarrollarlo en forma amigable (Ruby, 2018).

1.3.5. Marco de trabajo.

Un marco de trabajo o *framework* es un diseño abstracto orientado a objetos para un determinado tipo de aplicación, es un patrón arquitectónico que proporciona una plantilla extensible para un tipo específico de aplicaciones. Es un conjunto cohesivo de interfaces y clases que colaboran para proporcionar los servicios de la parte central e invariable de un subsistema lógico. El marco de trabajo diseñado posee una arquitectura orientada a componentes y está compuesto por cinco paquetes principales: núcleo, servicios, persistencia, componentes y aplicación de Usuario. Cada paquete está compuesto por componentes y estos a su vez definen una interfaz de comportamiento con el objetivo de extender sus funcionalidades y adaptarse a posibles escenarios (Anglada, 2013).

Ruby on Rails 4.2.7.

Es un *framework* de desarrollo de aplicaciones web escrito en el lenguaje de programación Ruby. Está diseñado para hacer que la programación de aplicaciones web sea más fácil, haciendo supuestos sobre lo que cada desarrollador necesita para comenzar. Te permite escribir menos código realizando más que muchos otros lenguajes y *frameworks*.

La filosofía de Rails se basa en estos dos principios:

- DRY (del inglés, "*Don't Repeat Yourself*") - sugiere que escribir el mismo código una y otra vez es una mala práctica.
- "Convención sobre Configuración" - significa que Rails hace algunas suposiciones sobre lo que quieres hacer y cómo vas a hacerlo, en lugar de requerir que especifiques cada pequeña cosa a través de un sin fin de archivos de configuración. (Rails, 2008).

Entre las características de Ruby on Rails podemos destacar que está basado en el patrón de diseño MVC (Modelo-Vista-Controlador) con el que se separa la lógica de la presentación, que nos permite dejar de pensar siempre en SQL para pasar a pensar en objetos, que dispone de utilidades para generar rápidamente interfaces de administración, es independiente de la base de datos, podemos realizar tests continuos para garantizar que nuestra aplicación funciona como esperamos, se puede extender mediante el uso de plugins,

se integra muy fácilmente con librerías de efectos como script.aculo.us; y todo ello escribiendo muy pocas líneas de código, muy claras y fáciles de entender (Rails, 2008).

1.3.6. Entorno de desarrollo integrado.

Un Entorno de Desarrollo Integrado(IDE), es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI) (IBM, 2009).

RubyMine 2017.3

RubyMine incluye un completo editor de código Ruby consciente de los detalles específicos del lenguaje dinámico. Proporciona asistencia de codificación inteligente, refactorización de código y capacidades de análisis de código profundo. Con una fácil configuración de proyectos, gestión automática de *Ruby Gems* y consolas incorporadas, tiene todo lo que necesita un desarrollador de Ruby en un entorno de desarrollo (RubyMine, 2000-2018).

Este IDE es el más recomendable a utilizar en la implementación del método de evaluación de competencia de los recursos humanos ya que contiene características de Ruby on Rails dedicadas y permite el completamiento de código.

1.3.7. Sistema gestor de bases de datos (SGBD).

Un sistema gestor de base de datos (SGBD) se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Ayuda a realizar las siguientes acciones:

- Definición y manipulación de los datos.
- Mantenimiento de la integridad de los datos dentro de la base de datos.
- Control de la seguridad y privacidad de los datos.

PostgreSQL 9.5.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código. 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 (OBE, 2015).

1.3.8. Servidor web.

Un servidor web es, un software instalado en el equipo con todas las condiciones necesarias para servir o entregar páginas web que le sean solicitadas por un navegador, asegurando que se muestren y representen todos los elementos necesarios para su correcto funcionamiento y visualización.

Apache 2.2.

Es un software de código abierto, libre de uso y totalmente configurable, es en este momento el más utilizado en la red, ya sea en plataformas Linux o Windows (Moro, 2013).

El servidor está dirigido a servir a una gran cantidad de plataformas web que trabajan sobre sistemas operativos como Unix, Windows, Linux, Solaris, Novell NetWare, FreeBSD, Mac OS X, Microsoft Windows, OS / 2 (Payer, y otros, 2015).

Entre sus principales ventajas se encuentran:

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Apache es una tecnología gratuita de código fuente abierto. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto. Esto le da una transparencia a este software de manera que si queremos ver que es lo que estamos instalando como servidor lo podemos saber, sin ningún secreto, sin ninguna puerta trasera.
- Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este, y están ahí para que los instalemos cuando los necesitemos. Otra cosa importante es que cualquiera que posea una experiencia decente en la programación de C o Perl puede escribir un módulo para realizar una función determinada.
- Apache trabaja con gran cantidad de Perl, PHP y otros lenguajes de script. Perl destaca en el mundo del script y Apache utiliza su parte del pastel de Perl tanto con soporte CGI como con soporte mod perl. También trabaja con Java y páginas jsp. Teniendo todo el soporte que se necesita para tener páginas dinámicas.
- Apache te permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en tu servidor (Apache, 2016).

1.4. Conclusiones parciales.

En el capítulo que finaliza se concluye que:

- La gestión por competencias es la base de la gestión de los recursos humanos en una organización, flexibiliza la adaptación de ésta ante los cambios del mercado, establece la correspondencia entre las exigencias del puesto de trabajo y el perfil de competencias de los individuos.
- Es necesario evaluar las competencias de los individuos como base para los procesos de selección de personal, el seguimiento y control de su desempeño, así como las necesidades de capacitación aspectos esenciales para la toma de decisiones en función del cumplimiento de los planes estratégicos y la mejora continua de la organización.
- El modelo de representación de la información lingüística basado en 2-tuplas realiza la evaluación de competencias involucrando múltiples evaluadores con el fin de combinar sus diferentes criterios y lograr una evaluación de mayor precisión y libre de apreciaciones, tomando en cuenta las opiniones de los evaluadores expresadas mediante diversas escalas cualitativas definidas a través de términos lingüísticos
- Se utiliza la metodología SCRUM para la gestión de proyectos, así como el lenguaje de programación Ruby 2.3.1 con el framework Rails 4.2.7, RubyMine 2017.3 es la herramienta para llevar a cabo el entorno de desarrollo de la aplicación.
- Para la realización del diseño de diagramas, el modelado de las clases y la captura de los requisitos se emplea el Visual Paradigm 8.0 para UML 2.1 y el gestor de base de datos el PostgreSQL 9.5.

Capítulo 2: Análisis y diseño del sistema.

2.1. Introducción.

En el presente se describe la propuesta de solución para el problema de la investigación exponiendo todo lo referente al análisis del modelo de dominio, el levantamiento de los requisitos funcionales y no funcionales del software, el diagrama de clase del diseño con el cual se representa la estructura estática del sistema y los patrones de arquitectura y diseño utilizados.

2.2. Modelo del dominio.

Un modelo de dominio es un artefacto de la disciplina de análisis, construido con las reglas de UML durante la fase de concepción, en la tarea construcción del modelo de dominio, presentado como uno o más diagramas de clases y que contiene, no conceptos propios de un sistema de software sino de la propia realidad física. Los modelos de dominio pueden utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de software o de otro tipo. Similares a los mapas mentales utilizados en el aprendizaje, el modelo de dominio es utilizado por el analista como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir (MD, 2008).

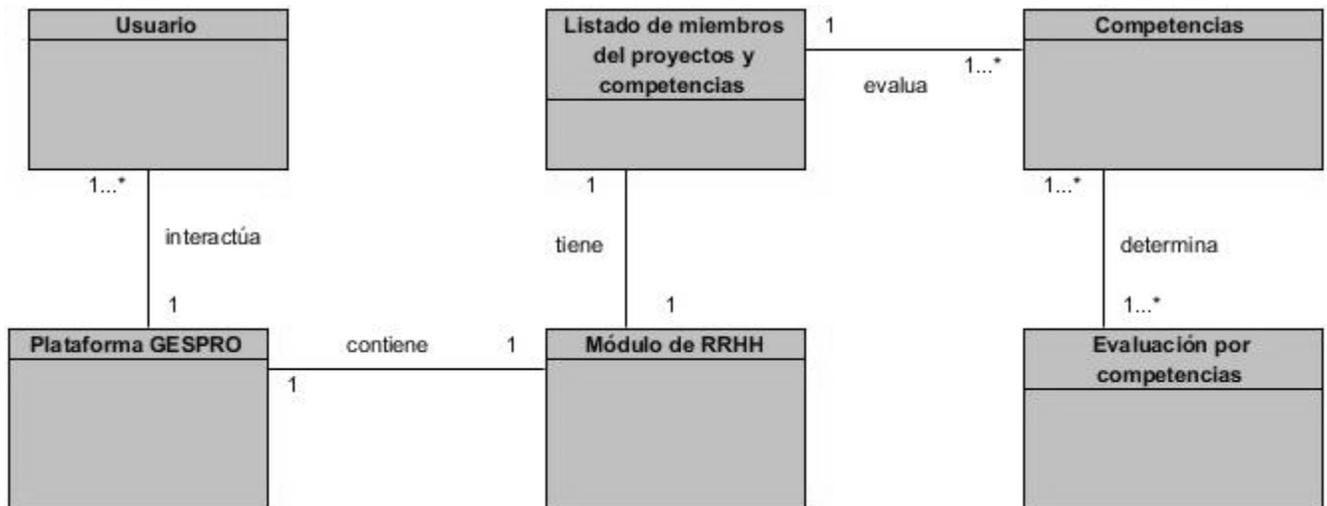


Fig. 2 Modelo de dominio

A continuación, se realiza una breve descripción de los principales conceptos utilizados:

Usuario: múltiples expertos que accederán a la plataforma de Gestión de Proyectos (XEDRO-GESPRO) para emitir las evaluaciones de competencias.

Plataforma XEDRO-GESPRO: herramienta para la gestión de proyecto extensible que actualmente incluye varios módulos como apoyo al área de recursos humanos (RRHH) permitiendo una gran facilidad a la evaluación de competencias de los recursos humanos.

Módulo de recursos humanos: software que permite la evaluación de competencia de los usuarios.

Listado de usuarios y competencias: lista todos los usuarios y las distintas competencias a evaluar.

Competencia: rasgos persistentes para las posibles evaluaciones que puede emitir cada experto.

Evaluación por Competencia: evaluación final por competencia de cada usuario expresada por múltiples expertos.

2.3. Propuesta del sistema.

La presente investigación propone como solución un método que se integre a la plataforma XEDRO-GESPRO para la conciliación de los criterios de varios expertos sobre la evaluación de las competencias en el módulo de recursos humanos, utilizando el modelo de representación lingüística de 2-tuplas facilitando la vía de evaluación ya que se puede dar una opinión tanto cuantitativa como cualitativa, siendo de gran ayuda a la hora de dar una evaluación más justa.

En el método a desarrollar como propuesta de solución el usuario autenticado como administrador define los expertos dándoles permisos de evaluadores. El principal campo a llenar para el administrador sería el peso del experto, en el cual se asigna un peso a cada uno, según la importancia y confiabilidad para emitir las evaluaciones. La solución propuesta es capaz de calcular las evaluaciones, haciendo para esto un corte todos los días 28 de cada mes, ejecutándose automáticamente gracias al empleo de la gema whenever.

2.3.1. Gema whenever

La gema whenever, es una gema de Ruby que permite crear tareas en **Crontab** de manera sencilla. Whenever está muy integrado al *framework* de desarrollo web Ruby on Rails, aunque se puede utilizar en un programa independiente que genere salidas de tareas para **Crontab** ya que la gema actualiza el archivo **Crontab**. Para el desarrollo del método propuesto, se emplea la gema whenever, permitiendo al sistema realizar de manera automática el cálculo de las evaluaciones por competencias de manera mensual, mediante la tarea definida en el módulo propuesto (Calaña, 2016).

2.4. Descripción de la solución propuesta.

El método que se propone tendrá como entrada las relaciones usuario-competencia que serán evaluadas por múltiples expertos de acuerdo a diferentes criterios. Este método tiene como salida el listado de las

evaluaciones de las competencias de los usuarios que permitirá a los especialistas, acelerar el proceso de evaluaciones de los recursos humanos y la asignación de tareas idóneas.

Para definir el marco de evaluación se deben seguir los siguientes pasos:

Paso 1: identificar un conjunto de expertos $E = \{e_i \vee i \in (1, \dots, m)\}$ que analizan la caracterización de las competencias laborales. Teniendo en cuenta que todos los expertos no necesariamente tienen la misma preparación y experticia, se define un vector de pesos $W^e = (w_{e1}, \dots, w_{em})$, que permite indicar la importancia y confiabilidad de los expertos para la evaluación, tal que $w_{e_i} \in [0,1]$, el cual es 0.5 por defecto o asignado por el administrador según su criterio.

Paso 2: identificar los criterios que se tienen en cuenta para la evaluación de las competencias laborales de los recursos humanos $C = \{c_k \vee k \in (1, \dots, p)\}$. Los criterios que se tienen en cuenta para la evaluación son los indicadores: IRHE (Indicador de eficacia), IRHF (Indicador de eficiencia), IRHC (Indicador de efectividad) y IRHCC (Indicador de consistencia del conjunto) para la caracterización de las competencias laborales. Estos criterios pueden tener diferentes pesos asignados de acuerdo a su importancia en la evaluación de competencias los cuales serían 0.27, 0.27, 0.27, 0.19 respectivamente, para los que se define el vector de pesos $w_c = (w_{irhe}, w_{irhf}, w_{irhc}, w_{irhcc})$, tal que $w_{c_i} \in [0,1]$ y $w_{irhe} + w_{irhf} + w_{irhc} + w_{irhcc} = 1$.

Paso 3: para expresar las preferencias se utiliza el vector de utilidad $X = (x_j^{ki}, \dots, x_j^{ki})$, donde x_j^{ki} representa la preferencia del experto e_i sobre la relación usuario-competencia a_j de acuerdo al criterio c_k . Se propone que la evaluación de los expertos sea ofrecida en valores lingüísticos que pertenecen al conjunto de términos lingüísticos $S: x_j^{ki} = s_j^{ki} \in S = \{s_0, \dots, s_g\}$ siendo $g + 1$ la cardinalidad del conjunto S .

Para la utilización de dominios lingüísticos se requiere que se realice una adecuada elección de los descriptores del conjunto de términos y su semántica, se propone la variante de usar una estructura ordenada de etiquetas donde la semántica de los términos lingüísticos se deriva de la propia estructura ordenada. Se propone el siguiente conjunto de etiquetas (ninguno, muy bajo, bajo, medio, alto, muy alto, perfecto) en lo adelante a este conjunto se denomina el conjunto básico de etiquetas (CBTL).

Se construye además la variable lingüística que representa la evaluación de las competencias formada por los conjuntos borrosos, donde cada término lingüístico $s_i \in S$ tiene asociada una función de pertenencia $\mu_{s_i} \in [0,1]$ de tipo triángulo (ver figura 3). La representación gráfica de la variable lingüística de evaluación de competencias se muestra en la figura 4.

$$Triángulo(x,a,b,c) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & c \leq x \end{cases}$$

Fig. 3 Función de pertenencia triangular

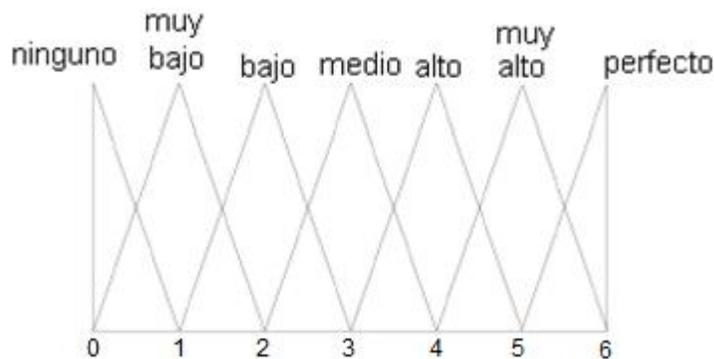


Fig. 4 Variable lingüística para la evaluación de competencias

Condiciones que deben cumplir los términos lingüísticos de acuerdo con lo planteado en (Martínez, 1999) (Espinilla, 2009) (Andrés, 2009) (Herrera, y otros, 2000) (Herrera, y otros, 2005).

- Las etiquetas que constituyen los términos lingüísticos de los conjuntos borrosos que representan la variable lingüística de evaluación de las competencias, se distribuyen uniformemente en el intervalo [0,6], como se muestra en la figura 4.
- La cardinalidad de los términos, expresada como la cantidad de funciones de pertenencia que la forman, debe ser impar y se recomienda el valor de siete. El término medio debe estar representado en el valor 3 y el resto de los términos colocados simétricamente alrededor de este. Los términos sugeridos se muestran en la figura 4.
- Existe un operador de negación: $\neg(s_i) = s_j, j = g - i(g + 1 \text{ es la cardinalidad de } T(H))$.
- Tiene un operador de maximización: $\max(s_i; s_j) = s_i \text{ si } s_i \geq s_j$.
- Tiene un operador de minimización: $\min(s_i; s_j) = s_i \text{ si } s_i \leq s_j$.

Recopilación de las preferencias de los expertos.

Los expertos proveen sus valoraciones a través de vectores de preferencia $X = (x_j^{ki}, \dots, x_j^{ki})$, que se recopilan como se muestra en la Tabla 1.

Tabla 1 Preferencias lingüísticas de los múltiples expertos

Usuario - Competencia	Criterios	Expertos		
		e_1	...	e_m
a_1	C_{IRHE}	x_1^{IRHE1}	...	x_1^{IRHEm}
	C_{IRHF}	x_1^{IRHF1}	...	x_1^{IRHFm}
	C_{IRHC}	x_1^{IRHC1}	...	x_1^{IRHCm}
	C_{IRHCC}	x_1^{IRHCC1}	...	x_1^{IRHCCm}
a_2	C_{IRHE}	x_2^{IRHE1}	...	x_2^{IRHEm}
	C_{IRHF}	x_2^{IRHF1}	...	x_2^{IRHFm}
	C_{IRHC}	x_2^{IRHC1}	...	x_2^{IRHCm}
	C_{IRHCC}	x_2^{IRHCC1}	...	x_2^{IRHCCm}
⋮	C_{IRHE}
	C_{IRHF}
	C_{IRHC}
	C_{IRHCC}
a_n	C_{IRHE}	x_n^{IRHE1}	...	x_n^{IRHEm}
	C_{IRHF}	x_n^{IRHF1}	...	x_n^{IRHFm}

	c_{IRHC}	x_n^{IRHC1}	...	x_n^{IRHCm}
	c_{IRHCC}	x_n^{IRHCC1}	...	x_n^{IRHCCm}

Análisis de la evaluación de competencias laborales.

Se realiza la agregación de las preferencias para obtener los valores lingüísticos de evaluación para cada competencia.

Unificación.

1. Transformación de las etiquetas lingüísticas a 2-tuplas lingüísticas.

A continuación, se presentan las funciones que permiten transformar etiquetas lingüísticas, representadas mediante 2-tuplas en valores numéricos y viceversa.

Sea $S = \{s_0, s_1, \dots, s_g\}$ un conjunto de términos lingüísticos y $\beta \in [0, g]$ un valor que representa el resultado de una operación de agregación simbólica. El conjunto de 2-tuplas asociado a la información equivalente a β es obtenido mediante la siguiente función:

$$\Delta S: [0, g] \rightarrow (S) \times$$

$$\Delta S(\beta) = (s_i, \alpha), \text{ con } \begin{cases} s_i = \text{round}(\beta) \\ \alpha = \beta - i, \alpha \in \end{cases}$$

Donde round asigna a β el número entero $i \in \{0, 1, \dots, g\}$ más cercano a β . Así, la 2-tupla lingüística $(s_i, \alpha_i) \in (S)$, que expresa la información equivalente a un valor numérico $\beta \in [0, g]$ en el intervalo de granularidad de S , se obtiene usando la función ΔS .

Nótese que ΔS es biyectiva y $\Delta S^{-1}: (S) \rightarrow [0, g]$ está definida por $\Delta S^{-1}(s_i, \alpha) = i + \alpha$. Así, toda 2-tupla de (S) estará identificada con un valor numérico en el intervalo $[0, g]$.

En este proceso de unificación, a cada valor de preferencia lingüístico suministrado por un experto $s_i \in \text{CBTL}$ tal que $\text{CBTL} = \{\text{ninguno}, \text{muy bajo}, \text{bajo}, \text{medio}, \text{alto}, \text{muy alto}, \text{perfecto}\}$, se le asocia el conjunto borroso S_T identificado por el término lingüístico que coincide con s_i y con traslación simbólica 0.

La transformación de un conjunto de valoraciones lingüísticas definidas en S_i en $F(S_T)$ se realiza como sigue:

Agregación.

De los operadores de agregación para 2-tuplas se utiliza el operador media ponderada teniendo en cuenta que en el marco del análisis se definió un vector de pesos que permite asignar a cada evaluador una determinada importancia, de acuerdo a su experticia y confiabilidad para la evaluación de competencias laborales; y otro vector de pesos para delimitar la importancia que tiene cada criterio para la evaluación.

- **Media ponderada extendida.**

La media ponderada permite que los valores x_i tengan diferente importancia, para lo cual cada valor x_i debe tener un peso asociado w_i que remarca su importancia. De modo que para un conjunto de 2-tuplas $x = \{(s_1, \alpha_1), \dots, (s_n, \alpha_n)\}$ con un vector de pesos asociado a cada 2-tupla, $W = (w_1, \dots, w_n)$, la extensión del operador se obtiene como se muestra:

$$\acute{x}^e(x) = \Delta \left(\frac{\sum_{i=1}^n \Delta^{-1}(s_i, \alpha_i) \cdot w_i}{\sum_{i=1}^n w_i} \right) = \Delta \left(\frac{\sum_{i=1}^n \beta_i \cdot w_i}{\sum_{i=1}^n w_i} \right)$$

1. Calcular el valor colectivo de cada criterio para cada relación.

Partiendo de que se manejan las preferencias de múltiples expertos, se determina el valor colectivo de los criterios para cada relación usuario-competencia, que teniendo en cuenta el peso de cada experto (w^{ei}), se podrá realizar de la siguiente manera Tabla 2.

$$(s_b, \alpha_b)_j^k = \acute{x}^e((s_a, \alpha_a)_j^{k1}, \dots, (s_a, \alpha_a)_j^{km})$$

Tabla 2 Valores colectivos de los criterios para cada relación usuario-competencia

Usuario - Competencia	Criterios	Expertos			Valores colectivos x criterios
		e_1	...	e_m	
a_1	c_{IRHE}	$(s_a, \alpha_a)_1^{IRHE1}$...	$(s_a, \alpha_a)_1^{IRHEm}$	$(s_b, \alpha_b)_1^{IRHE}$
	c_{IRHF}	$(s_a, \alpha_a)_1^{IRHF1}$...	$(s_a, \alpha_a)_1^{IRHFm}$	$(s_b, \alpha_b)_1^{IRHF}$

	c_{IRHC}	$(s_a, \alpha_a)_{1}^{IRHC1}$...	$(s_a, \alpha_a)_{1}^{IRHCm}$	$(s_b, \alpha_b)_{1}^{IRHC}$
	c_{IRHCC}	$(s_a, \alpha_a)_{1}^{IRHCC1}$...	$(s_a, \alpha_a)_{1}^{IRHCCm}$	$(s_b, \alpha_b)_{1}^{IRHCC}$
α_2	c_{IRHE}	$(s_a, \alpha_a)_{2}^{IRHE1}$...	$(s_a, \alpha_a)_{2}^{IRHEm}$	$(s_b, \alpha_b)_{2}^{IRHE}$
	c_{IRHF}	$(s_a, \alpha_a)_{2}^{IRHF1}$...	$(s_a, \alpha_a)_{2}^{IRHFm}$	$(s_b, \alpha_b)_{2}^{IRHF}$
	c_{IRHC}	$(s_a, \alpha_a)_{2}^{IRHC1}$...	$(s_a, \alpha_a)_{2}^{IRHCm}$	$(s_b, \alpha_b)_{2}^{IRHC}$
	c_{IRHCC}	$(s_a, \alpha_a)_{2}^{IRHCC1}$...	$(s_a, \alpha_a)_{2}^{IRHCCm}$	$(s_b, \alpha_b)_{2}^{IRHCC}$
⋮	c_{IRHE}
	c_{IRHF}
	c_{IRHC}
	c_{IRHCC}
α_n	c_{IRHE}	$(s_a, \alpha_a)_{n}^{IRHE1}$...	$(s_a, \alpha_a)_{n}^{IRHEm}$	$(s_b, \alpha_b)_{n}^{IRHE}$
	c_{IRHF}	$(s_a, \alpha_a)_{n}^{IRHF1}$...	$(s_a, \alpha_a)_{n}^{IRHFm}$	$(s_b, \alpha_b)_{n}^{IRHF}$
	c_{IRHC}	$(s_a, \alpha_a)_{n}^{IRHC1}$...	$(s_a, \alpha_a)_{n}^{IRHCm}$	$(s_b, \alpha_b)_{n}^{IRHC}$
	c_{IRHCC}	$(s_a, \alpha_a)_{n}^{IRHCC1}$...	$(s_a, \alpha_a)_{n}^{IRHCCm}$	$(s_b, \alpha_b)_{n}^{IRHCC}$

2. Calcular la evaluación de cada competencia.

Luego de contar con el valor colectivo de los criterios para cada competencia, se procede a determinar la evaluación de la relación usuario-competencia que se está analizando. Teniendo en cuenta el peso de cada criterio $w_c = (w_{IRHE}, w_{IRHF}, w_{IRHC}, w_{IRHCC})$, esta evaluación se puede obtener como se muestra en la Tabla 3:

$$(s_c, \alpha_c)_j = \acute{x}^e((s_b, \alpha_b)_j^1, \dots, (s_b, \alpha_b)_j^n)$$

Tabla 3 Valores colectivos de las competencias

Usuario-Competencias	Valores colectivos de las competencias
a_1	$(s_c, \alpha_c)_1$
a_2	$(s_c, \alpha_c)_2$
\vdots	\vdots
a_n	$(s_c, \alpha_c)_n$

Almacenamiento de los resultados.

Una vez que se obtiene la evaluación de cada relación usuario-competencia, son almacenados en la base de conocimiento de evaluación de competencias. Los resultados lingüísticos pueden ser mostrados a los expertos que intervinieron en el proceso de evaluación. Pueden acceder a esta evaluación otros interesados como jefes de proyectos u otros directivos y los propios recursos humanos evaluados.

Para almacenar en la base de conocimiento las evaluaciones obtenidas, se aplica la función $\Delta S^{-1}(s_i, \alpha) = i + \alpha$ que convierte la 2-tupla a un valor numérico en el intervalo $[0, g]$. Estos resultados serán utilizados como patrón para el entrenamiento del sistema basado en conocimiento que se utilice.

2.5. Especificación de los requisitos del software.

Para el correcto desarrollo de un software es necesario la obtención de los requisitos del sistema, siendo esta una de las etapas importantes para el desarrollo del mismo. Estos requisitos se pueden clasificar en requisitos funcionales y no funcionales, como se ven a continuación.

2.5.1. Requisitos funcionales.

Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requisitos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer y describen lo que debe hacer. Estos requisitos dependen del tipo de software que se

desarrolle, de los posibles usuarios del software y del enfoque general tomado por la organización al redactar requisitos (Escárcega, 2008.).

Tabla 4 Especificación de requisitos funciones del sistema

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF-1	Listar usuarios a evaluar	Lista los usuarios existentes que van a ser evaluados por los expertos.	Media	Media
RF-2	Ver detalles	Muestra un listado de las competencias correspondientes a cada usuario, los indicadores de eficacia, eficiencia y consistencia del conjunto, la evaluación del sistema, así como la evaluación multi-experto, la opción evaluar y eliminar.	Alta	Alta
RF-3	Evaluar usuario	Muestra un formulario con las posibles evaluaciones de los indicadores.	Alta	Alta
RF-4	Eliminar Evaluación	Permite eliminar del sistema la evaluación dada por el usuario autenticado.	Media	Media
RF-5	Exportar a pdf	Permite exportar la información del sistema a un documento en pdf.	Media	Media
RF-6	Exportar a csv	Permite exportar la información del sistema a un documento en csv.	Media	Media
RF-7	Asignar pesos a los expertos	Permite al usuario correspondiente la opción de asignarle un peso a cada experto según su nivel de experiencia.	Alta	Alta

2.5.2. Requisitos no funcionales.

Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requisitos no funcionales a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema. Los requisitos no funcionales, como su nombre sugiere, son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema. Los requisitos no funcionales rara vez se asocian con características particulares del sistema (Sosa, 2008).

Software.

- Se puede acceder al método a través de XEDRO-GESPRO desde cualquier sistema operativo.
- Se necesita conexión a la red y un navegador web instalado.

Hardware.

- El sistema debe contar, como requerimiento mínimo, con 256MB de memoria RAM.
- Se necesita que el sistema tenga al menos un microprocesador Intel Pentium IV, con una frecuencia de 2.0 GHz.

Accesibilidad.

- Los usuarios de la plataforma XEDRO-GESPRO tienen acceso a un conjunto limitado de funcionalidades, según el rol establecido en la plataforma.

Interfaz de usuario.

- La interfaz deberá ser agradable y sencilla de usar, que permita a cualquier usuario hacer uso de la aplicación sin dificultad.

Usabilidad.

- El método deberá poseer una interfaz intuitiva, de tal forma que pueda ser usado por cualquier persona con conocimientos básicos en el manejo de una computadora.

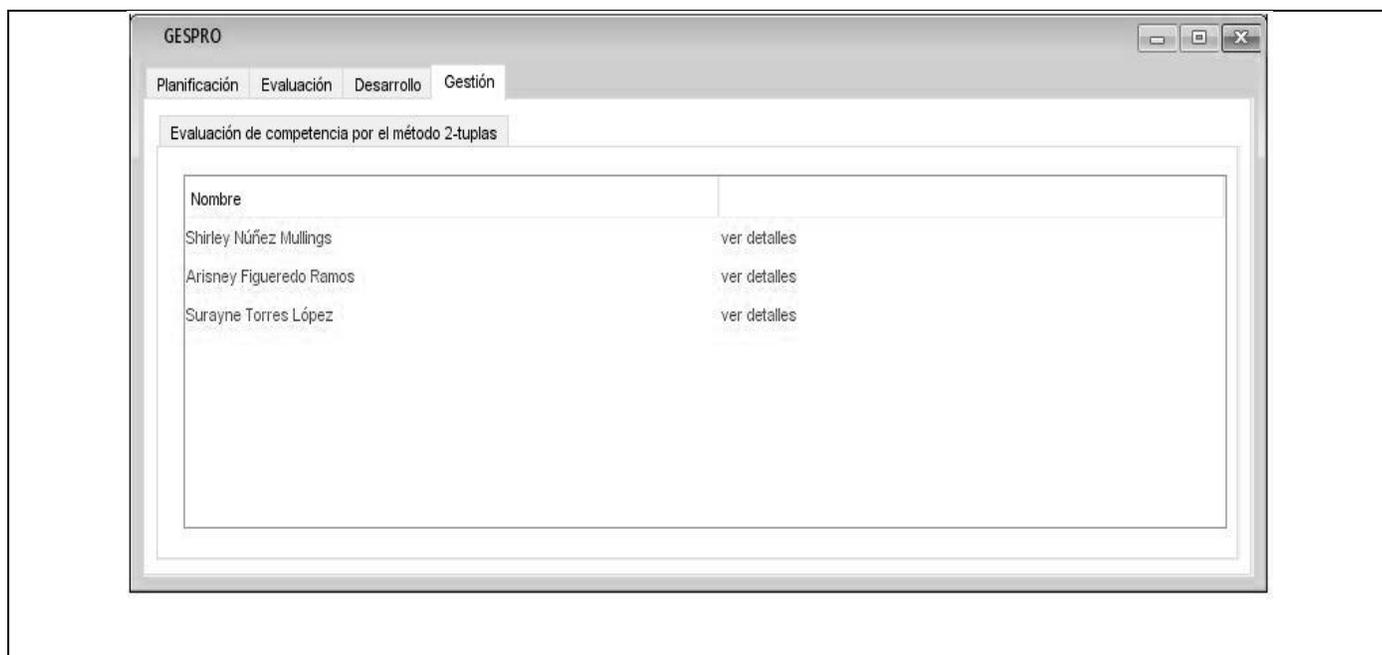
2.6. Historia de usuario.

Las historias de usuario son utilizadas en las metodologías de desarrollo ágiles para la especificación de requisitos (acompañadas de las discusiones con los usuarios y las pruebas de validación). Las historias de usuario son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos. Las historias de usuario permiten responder rápidamente a los requisitos cambiantes. Estas definen lo que se debe construir en el proyecto de software, tienen una prioridad asociada definida por el cliente de manera de indicar cuales son las más importantes para el resultado final, serán divididas en tareas y su tiempo será estimado por los desarrolladores (Historias de usuarios, 2016).

Durante el diseño de la propuesta de solución se identificaron 7 historias de usuarios que responden a las diferentes funcionalidades solicitadas por el cliente y presentan una descripción para que el desarrollador conozca su posterior implementación. A continuación, se describe la Historia de usuario referente al requisito funcional "Listar usuarios a evaluar", el resto de las Historias de usuarios se encuentran en los anexos del 1 al 6.

Tabla 5 Historia de usuario "Listar usuarios a evaluar"

Historia de usuario	
Número: 1	Nombre: Listar usuarios a evaluar
Cantidad de modificaciones: Ninguna	
Usuario: Shirley Núñez Mullings	Iteración asignada: 1
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Puntos estimados: 1.5	
Descripción: Lista los usuarios existentes que van a ser evaluados por los expertos.	
Observaciones: NA	
Prototipo de interfaz:	



2.7. Pila de Productos.

Siguiendo el modelo propuesto por la metodología SCRUM y teniendo en cuenta la importancia para el cliente de cada una de las tareas, se utiliza la Pila de Productos para listar los requisitos funcionales antes descritos.

Tabla 6 Pila de Productos

ID	Descripción	Prioridad	Duración (semanas)
1	Listar usuarios a evaluar	Media	1
2	Evaluar usuario	Alta	4
3	Eliminar Evaluación	Media	1
4	Ver detalles	Alta	4
5	Exportar a pdf	Media	1
6	Exporta a csv	Media	1
7	Asignar pesos a los expertos	Alta	4

2.8. Pila de Sprint.

Definida la Pila de Productos, se realiza la Pila de Sprint, que es una lista de los elementos de la Pila de Producto y sus tareas componentes, para posteriormente llevar a cabo el proceso de desarrollo.

Tabla 7 Pila de Sprint para la tarea Listar usuarios a evaluar

Tarea 1. Listar usuarios a evaluar	Estado	Estimado (días)	Real
Buscar información en la base de datos	100%	1	2
Mostrar usuarios	100%	3	5

Tabla 8 Pila de Sprint para la tarea Evaluar usuario

Tarea 2. Evaluar usuario	Estado	Estimado (días)	Real
Hacer tratamiento de privilegios	100%	2	2
Listar las competencias a evaluar	100%	6	7
Crear espacio para la evaluación	100%	3	3
Crear botón Evaluar	100%	2	3
Crear base datos	100%	3	3
Crear tablas	100%	3	4
Crear relaciones entre tablas	100%	2	2

Tabla 9 Pila de Sprint para la tarea Eliminar Evaluación

Tarea 3. Eliminar Evaluación	Estado	Estimado (días)	Real
Crear el botón Eliminar	100%	1	1
Eliminar información de la base de datos	100%	1	1
Mostrar mensaje de confirmación	100%	2	1

Tabla 10 Pila de Sprint para la tarea Ver detalles

Tarea 4. Ver detalles	Estado	Estimado (días)	Real
Crear botón Visualizar detalles	100%	1	1
Acceder a la base de datos	100%	2	3
Mostar detalles	100%	5	7

Tabla 11 Pila de Sprint para la tarea exportar a pdf

Tarea 5. exportar a pdf	Estado	Estimado (días)	Real
Crear botón de exportar a pdf	100%	2	1

Tabla 12 Pila de Sprint para la tarea exporta a csv

Tarea 6. exporta a csv	Estado	Estimado (días)	Real
Crear botón de exportar a csv	100%	2	1

Tabla 13 Pila de Sprint para la tarea *asignar pesos a los expertos*

Tarea 6. asignar pesos a los expertos	Estado	Estimado (días)	Real
Hacer tratamiento de privilegios	100%	2	2
Asignar los permisos de evaluador a los usuarios	100%	6	7
Crear tablas para los pesos	100%	3	3
Crear formulario para asignar pesos	100%	2	3
Crear base datos	100%	3	3
Crear tablas	100%	3	4
Crear relaciones entre tablas	100%	2	2
Asignar pesos a los evaluadores	100%	5	6

2.9. Arquitectura del sistema.

La arquitectura es la parte de la Ingeniería de *Software* que se ocupa de la descripción y el tratamiento de la estructura de un sistema como una serie de componentes, con el fin de organizar adecuadamente los distintos subsistemas, y permitir la integración de diferentes grupos de desarrollo en el mismo proyecto. Habitualmente se vincula con la fase de ejecución, aunque este detalle no es estrictamente necesario. Su principal objetivo es hacer énfasis en la importancia de la descripción estructural de los sistemas *software*, un aspecto bien conocido, pero poco tratado (Quintero, 2002).

2.9.1. Modelo-Vista-Controlador.

El MVC o Modelo-Vista-Controlador es un patrón de arquitectura de software que, utilizando 3 componentes (Vistas, Models y Controladores) separa la lógica de la aplicación de la lógica de la vista en una aplicación. Se utiliza tanto en componentes gráficos básicos hasta sistemas empresariales. Permite separar los componentes de la aplicación dependiendo de la responsabilidad que tienen, esto significa que cuando hacemos un cambio en alguna parte de nuestro código, esto no afecte otra parte del mismo (MVC (Model, 2018).



Fig. 5 Funcionamiento del MVC

En Ruby on Rails esta arquitectura está estructurada de la siguiente forma:

Modelo: mantiene la relación entre los objetos y la base de datos, maneja la validación, la asociación y las transacciones. Este subsistema se implementa en la biblioteca ActiveRecord, que proporciona una interfaz y enlace entre las tablas de una base de datos relacional y el código de programa Ruby que manipula los registros de la base de datos. Los nombres de los métodos Ruby se generan automáticamente a partir de los nombres de campos de las tablas de la base de datos.

Vista: es una presentación de datos en un formato particular, desencadenado por la decisión de un controlador de presentar los datos. Son sistemas basados en scripts como JSP, ASP, PHP y muy fáciles de integrar con la tecnología AJAX. Este subsistema se implementa en la biblioteca ActionView, que es un sistema basado en Ruby incrustado (ERb) para la presentación de datos. Cada conexión Web a una aplicación de Rails da como resultado la visualización de una vista.

Controlador: por un lado, consulta los modelos sobre datos específicos y por el otro lado, organiza los datos (búsqueda, clasificación, mensajería) en una forma que se ajuste a las necesidades de una determinada vista. Este subsistema se implementa en ActionController, que es un intermediario de datos entre ActiveRecord (la interfaz de la base de datos) y ActionView (el motor de presentación).

2.9.2. Patrones de diseño.

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Representan las mejores prácticas utilizadas por los desarrolladores de software orientados a objetos experimentados. Los patrones de diseño son soluciones a los problemas generales que enfrentan los desarrolladores de software durante

el desarrollo del software (pdf, 2017). A continuación, se relacionan algunos de los patrones de diseño empleados en la implementación del módulo.

Patrones GRASP: representan los principios básicos de la asignación de responsabilidades a objetos, expresados en forma de patrones. GRASP es el acrónimo para *General Responsibility Assignment Software Patterns* (Patrones Generales de *Software* para Asignar Responsabilidades).

Entre estos se toman los siguientes patrones:

- **Experto:** la responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada. Se hace uso de este patrón en la clase *GesproCompetenceUser*, ya que la misma contiene los atributos necesarios para almacenar el nombre y la evaluación de las competencias.
- **Creador:** se aplica para la asignación de responsabilidades a las clases relacionadas con la creación de objetos, de forma tal que una instancia de un objeto sólo puede ser creada por el objeto que contiene la información necesaria para ello. En el módulo se evidencia el uso de este patrón en la clase *GesproCompetenceUser*, ya que esta clase posee la información para crear objetos de tipo *GesproCompetenceUser*.
- **Controlador:** las clases controladoras son las encargadas de gestionar el acceso a la lógica del negocio y controlar todo el proceso. En el caso de que sean muy extensas estas se dividen en varias clases controladoras, para separar la carga de procesamiento, disminuir la complejidad y responsabilidad de las mismas. Se hace uso de este patrón en la clase *GesproRrhhProjectCompetenceUserTwoTuplesController*, la cual maneja las peticiones realizadas por el usuario para evaluar las competencias. Esta clase puede acceder a todas las clases del sistema permitiendo que se controle todo el proceso.
- **Bajo acoplamiento:** pretende asignar una responsabilidad para mantener el bajo acoplamiento, es decir, el diseño de clases más independientes, que no se relacionen con muchas otras, que reduzcan el impacto de los cambios, que sean más reutilizables y acrecienten la oportunidad de una mayor productividad. Se evidencia el uso de este patrón en la clase *GesproRrhhProjectCompetenceUserTwoTuplesController*, ya que la misma solo se relaciona con la clase controladora, evitando así que una modificación en alguna de ellas suponga una gran repercusión en las restantes.
- **Alta cohesión:** el diseño y la dependencia entre clases están elaborados a partir de las funcionalidades que realizan, presentando alta relación de afinidad en las operaciones que controlan. En el caso de las actividades de alta complejidad comparten relación con otros objetos,

disminuyendo la carga de transacciones entre ellas. En el módulo se evidencia el uso de este patrón en la clase `GesproRrhhProjectCompetenceUserTwoTuplesController` que ayuda a no cargar de código innecesario a la controladora.

Patrón de acceso a datos: Active Record

Active Record (AR) es un patrón que facilita la creación y manipulación de objetos de negocio quienes requieren ser almacenados persistentemente en una base de datos. Minimiza la cantidad de configuraciones mediante el uso de un conjunto de convenciones. Cada clase AR representa una tabla de la base de datos cuyos atributos son representados como las propiedades de la clase y una instancia AR representa una fila en esa tabla (Record, 2018). Es conveniente utilizar este patrón en el desarrollo del módulo ya que como se evidencia en el diagrama de clases representado en la Figura 6, donde se muestra que hay una clase nombrada Active Record y de ellas heredan tres clases.

2.10. Diagrama de clases del diseño

Es una representación gráfica que sirve para representar la estructura de un sistema que será implementado utilizando un lenguaje orientado a objetos. Los diagramas de clases se realizan en la fase de diseño del software después de la fase de requisitos. El objetivo de estos diagramas es representar las clases que tendrán el sistema así como su contenido y sus relaciones con otras clases. La implementación de sistemas medianamente grandes no sería abordable sin este tipo de diagramas y aunque fuera abordable se tardaría mucho más y sería más fácil cometer errores (Sistemas, 2018).

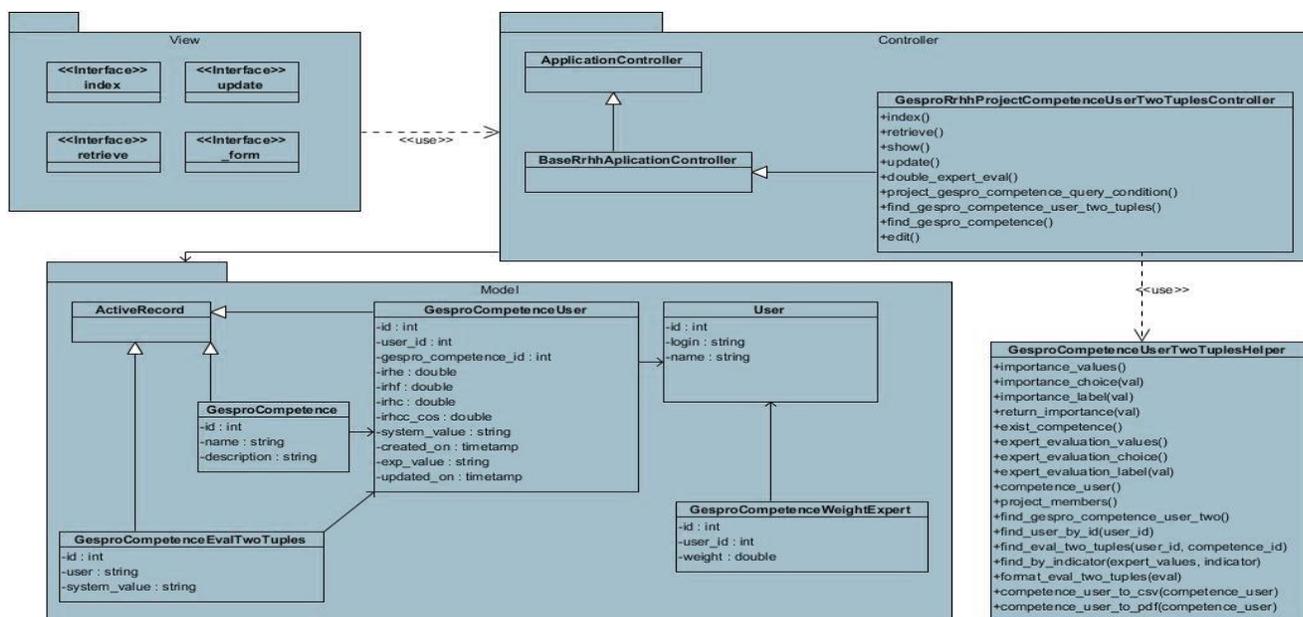


Fig. 6 Diagrama de clases.

Las clases que intervienen en el Diagrama de clases mostrado se describen a continuación:

ApplicationController: la aplicación Rails hereda de esta clase y define las acciones necesarias como métodos, que son invocados desde la web.

BaseRrhApplicationController: clase padre de todas las clases controladoras. Contiene todos los métodos genéricos heredados por otras clases.

GesproRrhProjectCompetenceUserTwoTuplesController: en esta clase se manejan las peticiones realizadas por el usuario para evaluar las competencias para que posteriormente muestre los resultados en la vista.

ActiveRecord: esta clase representa las tablas de la base de datos. Se hereda de la clase ActiveRecord::Base y se identifica automáticamente qué tabla usar y qué columnas tiene.

GesproCompetenceEvalTwoTuples: En esta clase se maneja la información sobre la evaluación final dada por los expertos.

GesproCompetence: en esta clase se maneja la información de las competencias.

GesproCompetenceUser: en esta clase se maneja la información referente a todas las evaluaciones dadas por los expertos y los diferentes indicadores de eficiencia, eficacia, consistencia.

User: se encarga de manejar los objetos referentes a los usuarios.

Update: esta clase está presente dentro de las vistas, su función es ejercer de formulario y enviar al controlador las peticiones realizadas por el usuario.

Index: esta clase perteneciente a las vistas, es la encargada de mostrar los usuarios pertenecientes a un proyecto que tienen competencias.

Retrieve: es la encargada de mostrar al usuario su evaluación. Esta clase está presente dentro de las vistas.

GesproRrhProjectCompetenceUserTwoTuplesHelper: esta clase provee métodos auxiliares para las funcionalidades asociadas a la evaluación de los usuarios.

GesproCompetenceWeightExpert: en esta clase se maneja todo lo referente a los pesos que se les asignan a los usuarios que tienen permiso de evaluador.

2.11. Modelo de datos.

Un modelo de datos es un sistema formal y abstracto que permite describir los datos de acuerdo con reglas y convenios predefinidos o podríamos decir que es un conjunto de concepto que permiten describir, a distintos niveles de abstracción, la estructura de una base de datos (Datos, 2010).

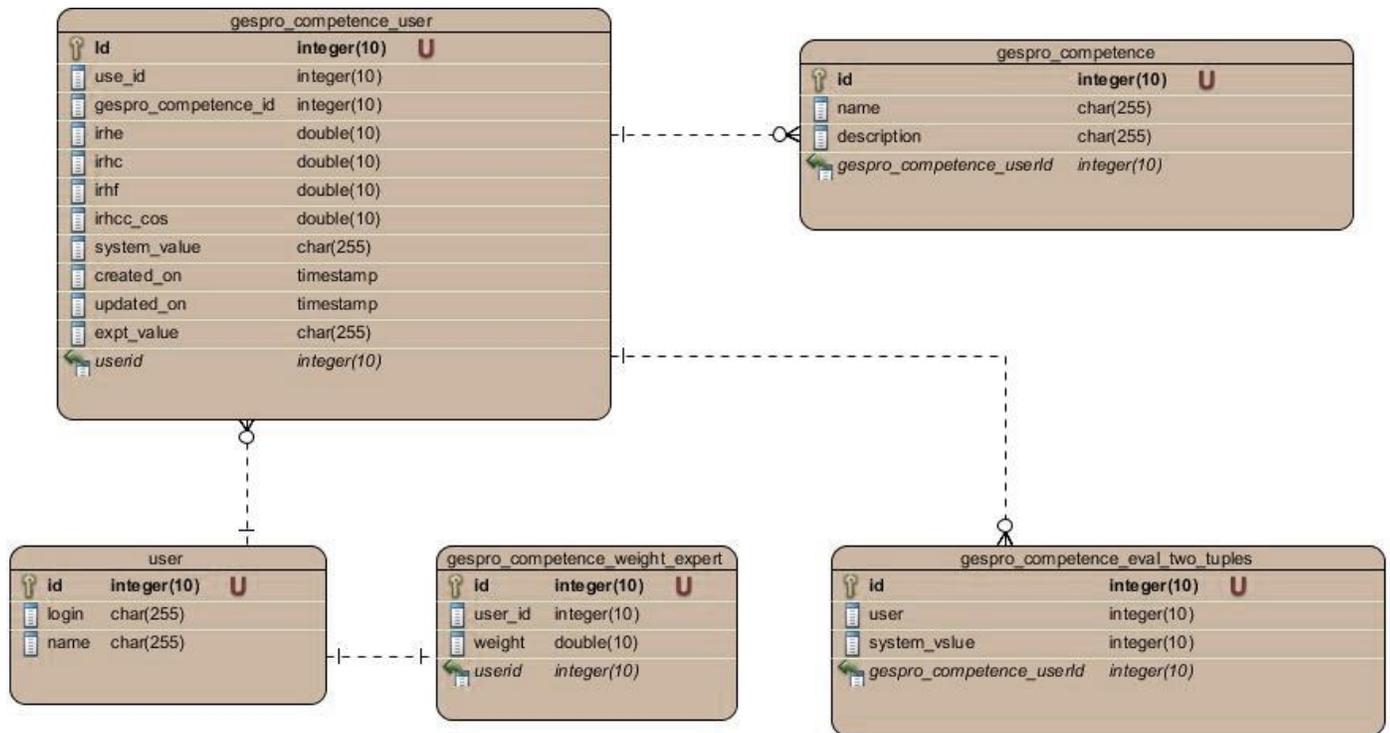


Fig. 7 Modelo de datos.

gespro_competence_user: en esta tabla se almacenan las diferentes competencias registradas en XEDRO-GESPRO. Contiene una relación de uno a muchos con las tablas gespro_competence y gespro_competence_eval_two_tuples.

gespro_competence: en esta tabla se almacenan las competencias y las descripciones de las mismas.

gespro_competences_eval_two_tuples: en esta tabla se almacenan las evaluaciones obtenidas de aplicar el método 2-tuplas con respecto a cada usuario.

user: en esta tabla se almacenan los datos de los usuarios registrados en la plataforma XEDRO-GESPRO. Contiene una relación de uno a muchos con la tabla gespro_competence_user_two_tuples.

GesproCompetenceWeightExpert: esta tabla almacena los pesos que se le asignan a cada experto.

2.12. Conclusiones parciales.

Tras la definición y análisis de cada uno de los temas abordados en este capítulo, se ha llegado a las siguientes conclusiones:

- Se describe la solución propuesta que permite evaluar las competencias por el método de 2-tuplas considerando criterios de diferentes expertos.
- El modelo de dominio elaborado permite un mejor análisis e interpretación de los conceptos asociados al dominio del problema.

- Los requisitos funcionales y no funcionales identificados permiten definir las funciones que deben realizarse y las restricciones a cumplir por el método elaborado.
- La definición de la arquitectura del sistema y del diagrama de clases del diseño permite establecer la base estructural de la implementación del método de 2-tuplas.
- El método de evaluación permite describir la estructura de la base de datos con el fin de garantizar una correcta gestión de los datos.

Capítulo 3: Implementación y pruebas a la solución propuesta.

3.1. Introducción.

En el presente capítulo se aborda la fase de implementación y pruebas. Se muestra el diagrama de despliegue implementado, así como el modelo de implementación, y la descripción de cómo los elementos del modelo de diseño se implementan en términos de componentes. Se diseñan y aplican las pruebas para comprobar la calidad de la aplicación y el correcto funcionamiento del método y se muestran los resultados obtenidos en las mismas.

3.2. Diagrama de despliegue.

Se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Un diagrama de despliegue es la forma de mostrar la configuración de nodos de procesamientos en tiempo de ejecución y los componentes que en ellos residen. Estos nodos forman la topología de hardware sobre el que se ejecuta el sistema. Este diagrama se preocupa principalmente de la distribución, entrega e instalación de las partes que constituye el sistema físico (Despliegue, 2018).

En la Figura 8 Diagrama de despliegue se muestran los componentes de *hardware* y sus relaciones para realizar el despliegue del sistema. Este modelo de despliegue está compuesto por tres nodos que representan al menos una computadora del usuario, el servidor de la plataforma XEDRO-GESPRO y el de base de datos. La plataforma XEDRO-GESPRO está instalada en un Servidor Web Apache, utilizando PostgreSQL como Sistema Gestor de Base de Datos encargado de almacenar los valores con que trabajará el sistema. Todas las PC clientes podrán acceder a dicho servidor mediante el protocolo de comunicación HTTPS, logrando que en cada estación de trabajo los usuarios tengan acceso a la plataforma.

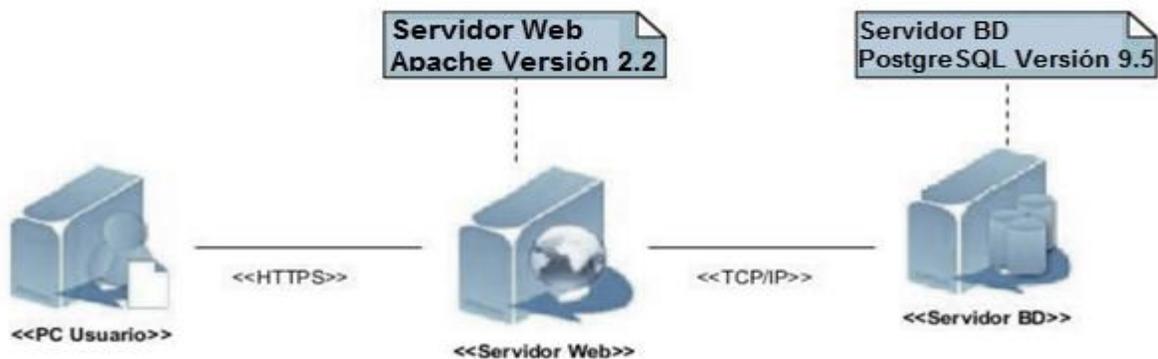


Fig. 8 Diagrama de despliegue.

3.3. Implementación.

La etapa de implementación del desarrollo de software es el proceso de convertir una especificación del sistema en un sistema ejecutable. Siempre implica los procesos de diseño y programación de software, pero, si se utiliza un enfoque evolutivo de desarrollo, también puede implicar un refinamiento de la especificación del software (Sommerville, 2005).

3.4. Estándares de codificación.

Los estándares de codificación son definidos por el equipo de desarrollo para lograr una estandarización en la programación del *software*. Engloban todos los aspectos relacionados con la generación de código, de tal manera que sea prudente, práctico y entendible para todos los programadores. Un código fuente en su totalidad debe reflejar un estilo armonioso, de forma tal que aparente ser realizado por un único programador. Si se logra la mantenibilidad del código, el sistema puede modificarse para añadir nuevas características, modificar las existentes, depurar errores, o mejorar el rendimiento (Microsoft, 2018).

En la Figura 6 se muestra la aplicación de algunos de los estándares de codificación utilizados en la implementación del sistema.

```
118
119 def project_gespro_competences_query_condition
120   result = " user_id=#{@competence_user_id}" if has_value_params(@competence_user_id)
121   result
122 end
123
124 def find_gespro_competence_user_two_tuples
125   @gespro_competence_user_two_tuples = GesproCompetenceUserTwoTuples.find(params[:id])
126   rescue ActiveRecord::RecordNotFound
127     render_404
128 end
129
130 def find_gespro_competence
131   @gespro_competence = GesproCompetence.find params[:id]
132   rescue ActiveRecord::RecordNotFound
133     render_404
134 end
135
136 def find_dimensions
137   @gespro_dimensions = GesproDimension.where(:gespro_competence_id => params[:id]).to_a
138   rescue ActiveRecord::RecordNotFound
139     render_404
140 end
```

Fig. 9 Ejemplo de uso de estándares de codificación.

3.4.1. Convenciones de nomenclatura.

Las convenciones de codificación son un conjunto de normas para un lenguaje de programación específico que recomiendan estilos de programación, buenas prácticas y métodos para mantener el aspecto del código

fuente. Estas convenciones incluyen la organización de archivos, los comentarios, las declaraciones los espacio en blanco, las llaves de apertura y cerrado (Vega, 2017). Para el desarrollo de este sistema se utilizaron los estándares de codificación empleado para la elaboración de la plataforma XEDRO-GESPRO, con el propósito de estandarizar las nomenclaturas en el desarrollo del sistema. A continuación, se muestran algunos de las convenciones empleadas en la implementación:

- **Snake case** o **snake_case**: es la habilidad de escribir palabras o frases compuestas en las que los elementos se separan con un carácter de subrayado (`_`), sin espacios, y con la primera letra de cada palabra en minúscula. Este formato se utilizó para escribir los métodos y las variables.
- **CamelCase**: es la práctica de escribir palabras o frases compuestas de modo que cada palabra o abreviatura en el centro de la frase empiece con una letra mayúscula, sin espacios intermedios ni signos de puntuación. Fue utilizado en el nombre de las clases.
- **SCREAMING_SNAKE_CASE**: se utiliza para las constantes y define que todos los elementos se escriben con letra mayúscula separadas por un carácter de subrayado (`_`) y sin espacios.
- Antes y después de los operadores y separadores utilizar un espacio.
- Cuando hay argumentos utilizar definición con paréntesis.
- Eliminar los paréntesis innecesarios al llamar a los métodos.
- Utilizar para comentario simple `#` y para bloques de comentarios `=begin [comentarios] =end`.
- No usar una línea en blanco entre el bloque de comentario y de la definición.

3.5. Pruebas al sistema.

Las pruebas al software son un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación (Pressman, 2009). Buscando cumplir con las funcionalidades formuladas por el cliente tratando de corregir cada error antes de entregar la versión final del producto. El principal objetivo del diseño de casos de prueba es obtener un conjunto de pruebas que tengan la probabilidad de revelar la mayor suma de defectos del software.

3.5.1. Pruebas de caja negra.

Las pruebas de caja negra son una manera de encontrar casos específicos en ese módulo que atiendan a su especificación. Se centran en los requisitos funcionales del software intentando descubrir errores como funciones incorrectas o ausentes, errores de la interfaz, en la estructura de datos o en accesos a bases de datos externas, de rendimiento, de inicialización o de terminación. Estas pruebas no utilizan ninguna información interna de los componentes de software o sistemas que se van a probar, sino que consideran

el comportamiento del software desde un punto externo. Para la aplicación de las pruebas de caja negra, se encuentra la técnica partición equivalente, la cual se utiliza para validar las funcionalidades del producto.

Partición equivalente.

Una partición equivalente divide el dominio de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. El diseño de estos casos de prueba para la partición equivalente se basa en la evaluación de las clases de equivalencia, las cuales representan un conjunto de estados válidos o inválidos para condiciones de entrada. Regularmente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica (Jaramillo, 2018).

Diseño de casos de pruebas.

Un diseño de caso de pruebas es un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para determinar si las funcionalidades de una aplicación están parcial o completamente realizadas. Su propósito es especificar una forma de probar el sistema que incluya las entradas, los resultados esperados y las condiciones bajo las que ha de probarse. Los casos de prueba ayudan a validar que el sistema desarrollado realice las funciones para las que ha sido creado en base a los requisitos del usuario, por lo que resulta importante diseñar al menos un caso de prueba para que cada requisito del sistema, con el objetivo de asegurar que todas las funcionalidades sean probadas. A continuación, se muestra el diseño de los casos de pruebas de los requisitos funcionales.

A continuación, se muestra el diseño de caso de prueba del RF: Listar usuarios a evaluar. El resto de los diseños de casos de pruebas pueden ser consultados en los anexos del 7 al 12.

Caso de prueba del RF: Listar usuarios a evaluar.

Descripción general.

- Lista los usuarios existentes que van a ser evaluados por los expertos.

Condiciones de ejecución.

- El usuario debe estar autenticado en la plataforma.

Tabla 14 Caso de prueba del RF: Listar usuarios a evaluar.

Escenario	Descripción	Respuesta del sistema
Listar los usuarios.	V	Se muestra el listado de los usuarios que poseen competencias para evaluar.
	Muestra los usuarios que poseen competencias.	

3.5.2. Pruebas de aceptación.

Este tipo de pruebas se lleva a cabo generalmente por un usuario o cliente con el fin de verificar si el producto ha sido desarrollado de acuerdo con las normas y criterios establecidos y cumple con todos los requisitos especificados. Estas pruebas se realizan sobre el producto terminado e integrado o pudiera ser una versión del producto o una iteración funcional pactada previamente con el cliente (**Aceptación, 2018**).

A continuación, se muestra la prueba de aceptación sobre la Historia de usuario Listar usuarios a evaluar. El resto de las pruebas de aceptación pueden ser consultadas en los anexos del 13 al 18.

Tabla 15 Prueba de aceptación sobre la Historia de usuario *Listar usuarios a evaluar*

Caso de prueba	
Código: HU1-P1	Historia de usuario: 1
Nombre: Listar usuarios a evaluar.	
Descripción: Prueba para la funcionalidad listar usuarios a evaluar.	
Condiciones de ejecución: El usuario debe estar autenticado en la plataforma.	
Entradas/Pasos de ejecución: <ul style="list-style-type: none">• Acceder al sistema.• Escoger un proyecto.• Seleccionar la opción Ejecución/Recursos humanos en el menú.• Seleccionar la pestaña Gestión.• Seleccionar Evaluación Competencias 2-tuplas.	
Resultado de las pruebas: El sistema de mostrar el listado de los posibles usuarios a evaluar que posean competencias.	
Evaluación de la prueba: Satisfactorio.	

3.6. Resultados de las pruebas.

Como parte de las pruebas de aceptación realizadas en 3 iteraciones, fueron detectadas un total de 16 no conformidades, las cuales fueron agrupadas en 2 tipos: ortografía y diseño, las que se muestran en la Figura 10.

Una vez terminadas las pruebas de aceptación, las cuales arrojaron resultados satisfactorios, se verificó que el sistema desarrollado cumple con las especificaciones del cliente. El sistema permite evaluar al usuario en cada una de las competencias ofreciendo una medida real, obteniendo además, una evaluación más completa con mayor fiabilidad.

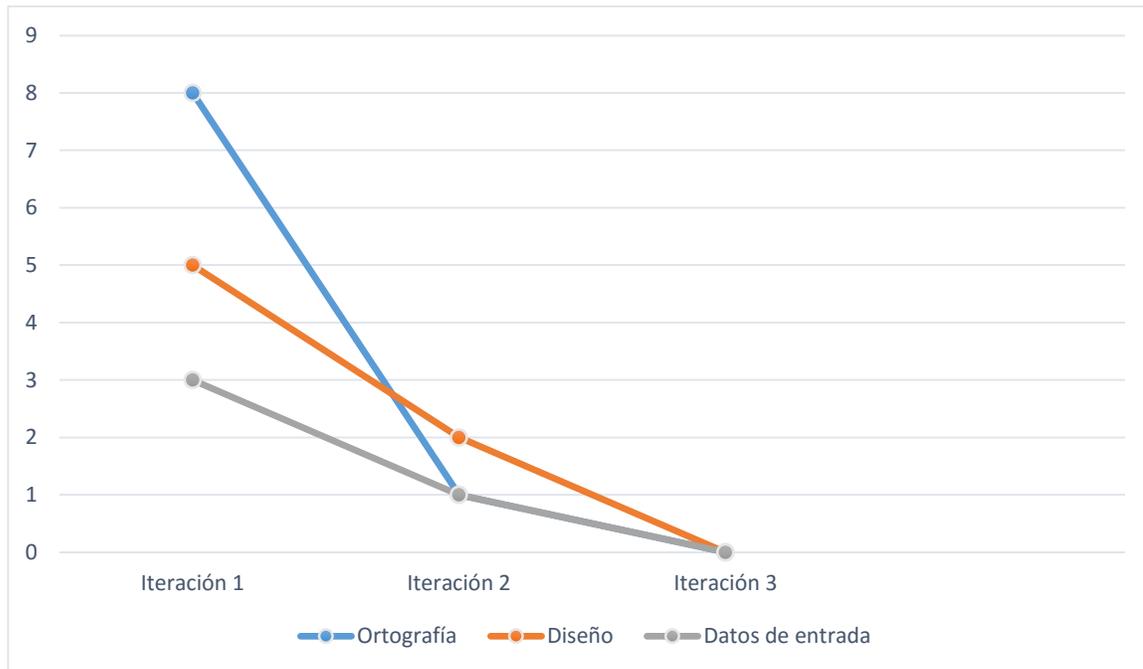


Fig. 10 Relación de no conformidades detectadas en las pruebas.

3.7. Técnicas de ladov.

La técnica de ladov en su versión original fue creada para el estudio de la satisfacción por la profesión en carreras pedagógicas. Está conformada por cinco preguntas: tres cerradas y 2 abiertas, constituye una vía indirecta para el estudio de la satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre las tres preguntas cerradas que se intercalan dentro de un cuestionario (ver anexo 19) y cuya relación el sujeto desconoce (**Díaz, 2018**). Estas tres preguntas se relacionan a través de lo que se denomina el "Cuadro Lógico de ladov", el cual se muestra adaptado a la presente investigación en el anexo 20.

El número resultante de la interrelación de las tres preguntas nos indica la posición de cada sujeto en la escala de satisfacción, o sea, su satisfacción individual. La escala de satisfacción utilizada es la siguiente:

1. Clara satisfacción.
2. Más satisfecho que insatisfecho.

3. No definida.
4. Más insatisfecho que satisfecho.
5. Clara insatisfacción.
6. Contradictoria.

Esta técnica también permite obtener el índice de satisfacción grupal (ISG), para lo cual se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1 como se muestra en el anexo 21. La satisfacción grupal se calcula por la siguiente fórmula:

$$\text{ISG} = A (+1) + B (+0,5) + C (0) + D (-0,5) + E (-1) N$$

En esta fórmula A, B, C, D, E, representan el número de sujetos con índice individual 1, 2, 3 o 6, 4, 5 y donde N representa el número total de sujetos del grupo. El índice grupal arroja valores entre + 1 y - 1. Los valores que se encuentran comprendidos entre - 1 y - 0,5 indican insatisfacción; los comprendidos entre - 0,49 y + 0,49 evidencian contradicción y los que caen entre 0,5 y 1 indican que existe satisfacción (**Fabre, 2017**). En el departamento de Gestión de Proyectos se aplicó la técnica de ladov a 15 trabajadores obteniendo los siguientes resultados:

- Clara satisfacción 10.
- Más satisfecho que insatisfecho 3.
- No definida 2.
- Más insatisfecho que satisfecho 0.
- Clara insatisfacción 0.
- Contradictoria 0.

El resultado del ISG fue de aproximadamente 0,77 lo que refleja aceptación de la propuesta y un alto grado de satisfacción por parte de los trabajadores del departamento.

3.8. Conclusiones parciales.

Al finalizar este capítulo se obtienen las siguientes conclusiones:

- El diagrama de despliegue se utilizó para establecer la disposición física de los componentes del hardware y sus relaciones en la implementación del despliegue del sistema.

- Con la utilización de los estándares de codificación definidos el código generado logra que el desarrollo del método sea prudente, práctico y entendible para todos los programadores.
- Mediante las pruebas realizadas se comprobó la validez del sistema diseñado, que garantiza el cumplimiento de los requisitos definidos y la entrega de un producto de mejor calidad y mayor aplicación.

Conclusiones Generales

Una vez finalizado el presente investigación se obtuvieron resultados satisfactorios y se arribó a las siguientes conclusiones:

- La metodología, herramientas y tecnologías utilizadas en la solución del problema investigado facilitaron el desarrollo del software diseñado.
- Con las funcionalidades implementadas, teniendo en cuenta los requisitos establecidos en el sistema, la arquitectura y los patrones de diseños utilizados, se obtuvo un método de mayor funcionalidad y mejores resultados al emplear la plataforma XEDRO-GESPRO 18.05.
- La realización de todas las pruebas requeridas por el sistema para su corrección permitieron la entrega de un producto final con la calidad requerida.
- El método desarrollado contribuye a definir, mediante términos lingüísticos, el resultado de la evaluación por competencias de un individuo, combinando las opiniones de los evaluadores expresadas a través de diversas escalas cualitativas.

Recomendaciones

Durante el desarrollo del módulo surgieron nuevas ideas que serían recomendables tener en cuenta:

- Permitir organizar por roles las evaluaciones por competencias.
- Analizar la utilización de la gema whenever para la configuración del cálculo de los indicadores, lo cual debe facilitar el proceso de despliegue de la plataforma XEDRO-GESPRO 18.05.

Bibliografía

- Aceptación, Pruebas de. 2018.** https://prezi.com/_g8uk9zcyj4i/pruebas-de-aceptacion/. [En línea] 2018. [Citado el: 29 de mayo de 2018.]
- Alaimo, Martín. 2013.** *Proyectos ágiles con SCRUM*. 2013.
- Ampuero, M., Baldoquín, M. G., & Soler, J. M. 2007.** *Gestión de Recursos Humanos por Competencias en los Proyectos de Software*. s.l. : Revista Cubana de Ciencias Informáticas, I(4), 82-91., 2007.
- Andres, M. 2009.** *Un modelo para la asignación de recursos humanos a equipos de proyectos de software*. . Ciudad de la Habana, Cuba: Instituto Superior Politécnico “José Antonio Echeverría”, Facultad de Ingeniería Informática. : s.n., 2009.
- Andrés, R. 2009.** *Evaluación de desempeño: Nuevos enfoques desde las teorías de subconjuntos difusos y de la decisión multi-criterio*. Departamento de Economía Aplicada, Facultad de Ciencias Económicas y Empresariales. Valladolid : Universidad de Valladolid, 2009. Tesis doctoral.
- Anglada, Alain Abel Garófalo Hernández. 2013.** *Revista Cubana de Ciencias Informáticas*. *Revista Cubana de Ciencias Informáticas*. [En línea] abr-jun. de 2013. [Citado el: 26 de octubre de 2015.] http://scielo.sld.cu/scielo.php?pid=S2227-18992013000200006&script=sci_arttext. ISSN 2227-1899.
- Apache. 2016.** Una Introducción a APACHE. http://linux.ciberaula.com/articulo/linux_apache_intro. [En línea] 2016. [Citado el: 30 de mayo de 2018.]
- Calaña, Ernesto Soler. 2016.** *Diseno e implementación de un módulo para la gestión de activos para la plataforma GESPRO 16.05*. La habana : Universidad de las Ciencias Informáticas, Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE) , Facultad 5, 2016.
- Clemen. 1995.** *Making Hard Decisions. An Introduction to Decision Analysis*. Duxbury Press. 1995.
- Cruz, P. K. y Vega, G. M. 2001.** *La gestión por competencias: una nueva herramienta en la planificación estratégica del recurso humano*. 2001.
- Cuesta, A. 2011.** *Metodología de Gestión por Competencias Asumiendo la Norma Cubana de Gestión de Capital Humano*. [ed.] Joao Mauricio Gama. 2011. págs. 300-311. Vol. 13. ISSN 1806-4892.
- Datos, Definición de un modelo de. 2010.** <https://tombasededatos.wordpress.com/2010/08/28/2-1-definicion-de-un-modelo-de-datos/>. [En línea] 28 de agosto de 2010. [Citado el: 5 de marzo de 2018.]

- Despliegue, Diagrama de. 2018.** https://nanopdf.com/download/figura-1-diagrama-de-despliegue_pdf. [En línea] febrero de 2018. [Citado el: 24 de abril de 2018.]
- Diaz, O. F. Técnica de ladov. 2018.** <https://es.slideshare.net/ofebles/gc-37299112>. [En línea] 2018. [Citado el: 29 de mayo de 2018.]
- Disca. 2012.** disca. *disca.upv*. [En línea] upv, enero de 2012. <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.
- Espinilla, M. 2009.** *Nuevos Modelos de Evaluación Sensorial con Información Lingüística*. España : Escuela Politécnica Superior de Jaén: Tesis para optar al grado de Doctor en Informática., 2009.
- Fabre, M. S. A. F. D. C. 2017.** http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2071-00542014000300012. [En línea] 2017. [Citado el: 25 de mayo de 2018.]
- Gil, Javier. 2007.** *La evaluación de competencias laborales*. 2007. págs. 83-106.
- Herrera, F y Martínez, L. 2000.** *A 2-tuple Fuzzy Linguistic Representation Model for Computing with Words*. s.l. : IEEE, 2000. págs. 746-752. Vol. 8.
- Herrera, F., Martínez, L. y Sánchez, P. J. 2005.** *Managing non-homogeneous information in group decision making*. 2005. págs. 115-132. Vol. 166.
- Herrera, Francisco, y otros. 2001.** *A linguistic decision model for personnel management solved with a linguistic biobjective genetic algorithm*. 2001. págs. 47-64.
- Historias de usuarios, umh2018-TADS. 2016.** <http://umh2818.edu.umh.es/wp-content/uploads/sites/884/2016/02/Historias-de-usuario.pdf>. [En línea] febrero de 2016. [Citado el: 5 de marzo de 2018.]
- IBM. 2009.** IBM Knowledge Center. *IBM Knowledge Center*. [En línea] IBM, diciembre de 2009. [Citado el: 24 de octubre de 2015.] www-01.ibm.com/support/knowledgecenter/SSEPGG_8.2.0/com.ibm.db2.udb.doc/ad/c0009415.htm?lang=es.
- Inteco, Laboratorio Nacional de Calidad del Software. 2010.** *INGENIERÍA DEL SOFTWARE: METODOLOGÍAS Y CICLOS DE VIDA*. Gobierno de España : s.n., 2010.
- Jaramillo, N. 2018.** https://prezi.com/-jp_rqhov1nn/particiones-o-clases-de-equivalencia/. *Particiones o Clases de Equivalencia*. [En línea] 2018. [Citado el: 25 de abril de 2018.]

- Jiang, Yingjie, y otros. 2010.** *The Determination of Cognitive Behavior Mode Based on Fuzzy Logic System.* s.l. : IEE, 2010. págs. 80-84. 978-0-7695-4279-9/10.
- Jing, Han. 2009.** *Application of Fuzzy Data Mining Algorithm in Performance Evaluation of Human Resource.* s.l. : International Forum on Computer Science-Technology and Applications, 2009. 978-0-7695-3930-0/09.
- Kaneiwa, Ken, MIZOGUCHI, Riichiro y NGUYEN, Philip HP. 2015.** *A Logical and Ontological Framework for Compositional Concepts of Objects and Properties.* s.l. : New Generation Computing, 2015.
- Martínez, L. 1999.** *Un nuevo modelo de representación de información lingüística basado en 2-tuplas para la agregación de preferencias lingüísticas.* Granada : Universidad de Granada, España: Tesis para optar al grado de Doctor en Informática., 1999.
- Martínez, Marilé Lemus, et al. 2016.** *Sistema para la Informatización de la Gestión de Anuarios Estadísticos en la ONEI.* s.l. : Serie Científica-Universidad de las Ciencias Informáticas, 2016.
- MD. 2008.** Tecnología y Synergix. <https://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>. [En línea] 10 de julio de 2008. [Citado el: 15 de mayo de 2018.]
- Microsoft. 2018.** [https://msdn.microsoft.com/es-es/library/aa291591\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa291591(v=vs.71).aspx). *Revisiones de código y estándares de codificación.* [En línea] 2018. [Citado el: 24 de abril de 2018.]
- Moro, Alfonso Infante, Oscar Gallego Pérez, Amparo Sánchez Macías. 2013.** *Los gadgets en las plataformas de telefomación: el caso del proyecto DIPRO 2.0.* s.l. : Pixel-Bit: Revista de medios y educación, 2013.
- MVC (Model, View, Controller) explicado. 2018.** <https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>. [En línea] 2018. [Citado el: 5 de marzo de 2018.]
- OBE, Regina O. y HSU, Leo S. 2015.** *PostGIS in action.* s.l. : Manning Publications Co. 2015.
- ONN. 2007.** *Norma Cubana NC 3000:2007.* La Habana : Oficina Nacional de Normalización, 2007.
- Paradigm. 2010.** Paradigm, 2010. *Paradigm, 2010.* [En línea] www.visual-paradigm.com, 2010. [Citado el: 15 de noviembre de 2015.] <http://www.visual-paradigm.com/aboutus/newsreleases/vpuml80.jsp>.
- Payer, Mathias, BARRESI, Antonio y GROSS, Thomas R. 2015.** *Fine-grained control-flow integrity through binary hardening. En International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment.* s.l. : Springer International Publishing, 2015.

pdf, Libro Patrones de diseño en. 2017. <https://postparaprogramadores.com/libro-patrones-diseno/> . [En línea] noviembre de 2017. [Citado el: 24 de abril de 2018.]

Pereda, S. y Berrocal, F. 2004. *Gestión de recursos humanos por competencias*. Madrid : Centro de Estudios Ramón Areces., 2004.

PMOinformatica.com. 2016. <http://www.pmoinformatica.com/2013/11/plantillas-scrum-pila-producto-product.html>. <http://www.pmoinformatica.com/2016/11/plantillas-scrum-sprintbacklog.html#more>. [En línea] noviembre de 2016. [Citado el: 30 de mayo de 2018.]

Pressman, Roger. 2009. *Ingeniería del Software. Un enfoque práctico*. s.l. : McGraw-Hill/Interamericana, 2009.

Quintero. 2002. *Arquitectura de software dinámica basada en reflexión*. 2002.

Rails, Introducción a Ruby on. 2008. http://librosweb.es/libro/introduccion_rails/capitulo_2.html. http://www.volcanica.cat/wp-content/uploads/2008/08/introduccion_ruby_on_rails.pdf. [En línea] 2008. [Citado el: 28 de febrero de 2018.]

Record, Fundamentos de Active. 2018. http://www.guiasrails.es/active_record_basics.html. [En línea] 2018. [Citado el: 24 de abril de 2018.]

Ruby, El lenguaje de. 2018. <http://eudev2.uta.cl/rid=1GR0DSG4D-1Y1NH87-4RQ/ruby.pdf>. [En línea] 2018. [Citado el: 28 de febrero de 2018.]

RubyMine. 2000-2018. <https://www.jetbrains.com/ruby/features/>. [En línea] 2000-2018. [Citado el: 28 de febrero de 2018.]

Sistemas, Diagramas UML (Diseño de. 2018. <https://www.slideshare.net/josuesalas5/diagramas-uml-diseo-de-sistemas-66609390>. [En línea] 2018. [Citado el: 5 de marzo de 2018.]

Sommerville, I. 2005. *Ingeniería del software*. 2005.

Sosa, Ángel Gabriel Olivera. 2008. <https://es.scribd.com/doc/37187866/Requerimientos-funcionales-y-no-funcionales>. *Planificación y modelado*, Instituto Tecnológico Superior Escárcega. [En línea] 2008. [Citado el: 5 de marzo de 2018.]

Stewart, Claire R., et al. 2016. *Sensory symptoms and processing of nonverbal auditory and visual stimuli in children with autism spectrum disorder*. s.l. : Journal of autism and developmental disorders, 2016.

Tejada, J., & Navio, A. 2005. *El desarrollo y la gestión de competencias profesionales: una mirada desde la formación.* s.l. : Revista Iberoamericana de Educación., 2005.

uml-diagrams. 2010. uml-diagrams. *uml-diagrams.* [En línea] 2010. [Citado el: 14 de octubre de 2015.] <http://www.uml-diagrams.org/>.

Vega, Adrian Alonso. 2017. <https://medium.com/@alonsus91/convenci%C3%B3n-de-nombres-desde-el-camelcase-hasta-el-kebab-case-787e56d6d023> . *Convencion de nombres: desde el CamelCase hasta el kebab-case.* . [En línea] junio de 2017. [Citado el: 25 de abril de 2018.]

Wang, Xinchun y Jiang, Yubo. 2010. *The Study of Enterprise Human Resource Evaluation Research Based on Neural.* Qingdao : Third International Symposium on Information Processing, 2010.

Xiaofan, C., & Fengbin, W. 2010. *Application of Data Mining on Enterprise Human Resource Performance Management.* s.l. : Kunming: 3rd International Conference on Information Management, Innovation Management and Industrial Engineering., 2010.

Anexos

Anexo 1: Historia de usuario Evaluar usuario.

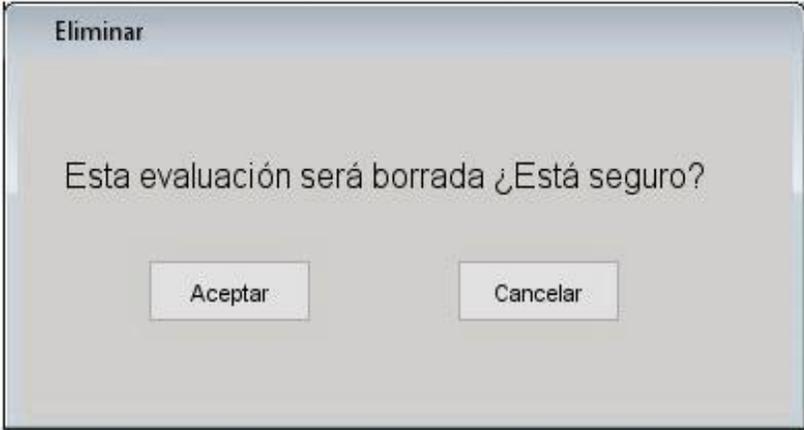
Tabla 16 Historia de usuario “Evaluar usuario”

Historia de usuario	
Número: 2	Nombre: Evaluar usuario
Cantidad de modificaciones: Ninguna	
Usuario: Shirley Núñez Mullings	Iteración asignada: 1
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2.1	
Descripción: Muestra un nuevo formulario con un listado de competencias pertenecientes a ese usuario y las posibles evaluaciones que pueden proporcionar los expertos.	
Observaciones: NA	
Prototipo de interfaz:	
	

Anexo 2: Historia de usuario Eliminar evaluación.

Tabla 17 Historia de usuario “Eliminar Evaluación”

Historia de usuario	
Número: 3	Nombre: Eliminar Evaluación
Cantidad de modificaciones: Ninguna	
Usuario: Shirley Núñez Mullings	Iteración asignada: 1

Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 1.8	
Descripción: Permite eliminar del sistema la evaluación del usuario.	
Observaciones: NA	
Prototipo de interfaz:	
	

Anexo 3: Historia de usuario Ver detalles.

Tabla 18 Historia de usuario “Ver detalles”

Historia de usuario	
Número: 4	Nombre: Ver detalles
Cantidad de modificaciones: Ninguna	
Usuario: Shirley Núñez Mullings	Iteración asignada: 1
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2.3	
Descripción: Muestra el usuario correspondiente todas las competencias evaluadas y la evaluación final.	
Observaciones: NA	
Prototipo de interfaz:	

Detalles								
Competencias	IRHE	IRHF	IRHC	IRHCC	Eval.Sistema	Eval.Multiexperto	Fecha Cort...	
Identificar servicios candidat...	0.7	0.2	0.6	0.5	Muy bien	bajo:-0.13	alto:0.2	Evaluar
Analizar requisitos, identificar...	0.2	0.4	0.8	0.1	Bien	alto:0.3	muy alto:0.9	Evaluar
Establecer, mantener registro...	0.5	0.8	0.3	0.9	Muy bien	muy alto:-0.13	bajo:0.4	Evaluar

Anexo 4: Historia de usuario Exportar a pdf

Tabla 19 Historia de usuario "Exportar a pdf"

Historia de usuario	
Número: 5	Nombre: Exportar a pdf
Cantidad de modificaciones: Ninguna	
Usuario: Shirley Núñez Mullings	Iteración asignada: 1
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 1.1	
Descripción: Permite exportar la información del sistema a un documento en pdf.	
Observaciones: NA	

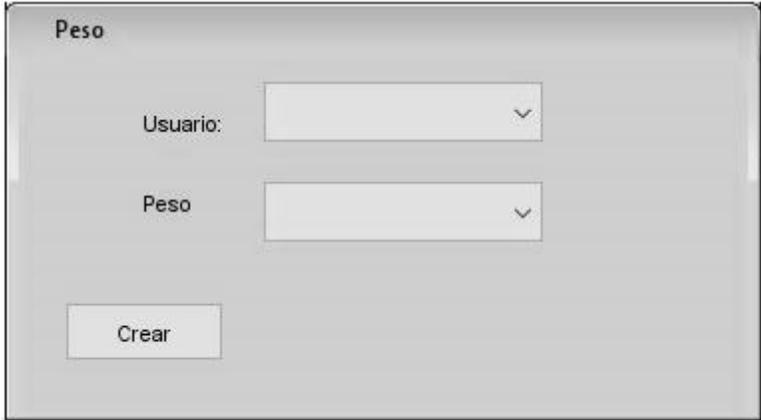
Anexo 5: Historia de usuario Exportar a csv.

Tabla 20 Historia de usuario "Exportar a csv"

Historia de usuario	
Número: 6	Nombre: Exportar a csv
Cantidad de modificaciones: Ninguna	
Usuario: Shirley Núñez Mullings	Iteración asignada: 1
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 1.1	
Descripción: Permite exportar la información del sistema a un documento en csv.	
Observaciones: NA	

Anexo 6: Historia de usuario Asignar peso a los expertos.

Tabla 21 Historia de usuario “Asignar pesos a los expertos”

Historia de usuario	
Número: 7	Nombre: Asignar pesos a los expertos
Cantidad de modificaciones: Ninguna	
Usuario: Shirley Núñez Mullings	Iteración asignada: 1
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2.3	
Descripción: Permite al usuario correspondiente la opción de asignarle un peso a cada experto según su nivel de experiencia.	
Observaciones: NA	
Prototipo de interfaz:	
	

Anexo 7: Prueba de caja negra. Evaluar a usuarios.

Caso de prueba del RF: Evaluar a usuarios.

Descripción general.

- Muestra un formulario con las posibles evaluaciones de los indicadores.

Condiciones de ejecución.

- El experto debe estar autenticado en la plataforma.
- Debe tener asignado el permiso de evaluador.

Tabla 22 Caso de prueba del RF: Evaluar a usuarios.

Escenario	Descripción	Sistema	Respuesta del sistema
-----------	-------------	---------	-----------------------

Evaluar a los usuarios de forma correcta.	V	V	El sistema muestra el mensaje “Actualizado correctamente”.
	Muestra los indicadores para poder formular la evaluación de la competencia.	Selecciona la evaluación para cada indicador.	

Anexo 8: Prueba de caja negra. Eliminar evaluación.

Caso de prueba del RF: Eliminar evaluación.

Descripción general.

- Permite eliminar del sistema la evaluación dada por el usuario autenticado.

Condiciones de ejecución.

- El experto debe estar autenticado en la plataforma.
- Debe tener asignado el permiso de evaluador.

Tabla 23 Caso de prueba del RF: Eliminar evaluación.

Escenario	Descripción	Confirmación del sistema	Respuesta del sistema
EC 1 Eliminar la evaluación del usuario de forma correcta.	V	V	Muestra el mensaje “Se borró satisfactoriamente”.
	Se elimina el perfil del usuario de forma correcta.	Muestra el mensaje “Esta evaluación será borrada. ¿Está seguro?”.	

Anexo 9: Prueba de caja negra. Ver detalles.

Caso de prueba del RF: Ver detalles.

Descripción general.

- Muestra un listado de las competencias correspondientes a cada usuario, los indicadores de eficacia, eficiencia y consistencia del conjunto, la evaluación del sistema, así como la evaluación multi-experto, la opción evaluar y eliminar

Condiciones de ejecución.

- El experto debe estar autenticado en la plataforma.

Tabla 24 Caso de prueba del RF: Ver detalles.

Escenario	Descripción	Respuesta del sistema
Ver detalles.	V	Muestra la evaluación del usuario para cada una de las competencias.
	Muestra el listado de las competencias correspondientes al usuario seleccionado, así como los valores de los indicadores de eficiencia, eficacia, efectividad y consistencia del conjunto.	

Anexo 10: Prueba de caja negra. Exportar a pdf.

Caso de prueba del RF: Exportar a pdf.

Descripción general.

- Permite exportar la información del sistema a un documento en pdf.

Condiciones de ejecución.

- El experto debe estar autenticado en la plataforma.

Tabla 25 Caso de prueba del RF: Exportar a pdf.

Escenario	Descripción	Respuesta del sistema
Exportar a pdf.	V	El sistema guarda las evaluaciones en pdf.
	Muestra la opción de exportar las evaluaciones a pdf.	

Anexo 11: Prueba de caja negra. Exportar a csv.

Caso de prueba del RF: Exportar a csv.

Descripción general.

- Permite exportar la información del sistema a un documento en csv.

Condiciones de ejecución.

- El experto debe estar autenticado en la plataforma.

Tabla 26 Caso de prueba del RF: Exportar a csv.

Escenario	Descripción	Respuesta del sistema
Exportar a csv.	Muestra la opción de exportar las evaluaciones a csv.	El sistema guarda las evaluaciones en csv.

Anexo 12: Prueba de caja negra. Asignar peso a los expertos.

Caso de prueba del RF: Asignar peso a los expertos.

Descripción general.

- Permite asignar pesos a los expertos que poseen permisos de evaluadores.

Condiciones de ejecución.

- El usuario debe estar autenticado en la plataforma como administrador.

Tabla 27 Caso de prueba del RF: Asignar pesos a los expertos.

Escenario	Descripción	Respuesta del sistema
EC 1 Asignar pesos a los expertos de forma correcta.	V Permite asignarle un peso a cada experto según su nivel de experiencia que posea permiso de evaluador.	Creado correctamente.

Anexo 13: Prueba de aceptación. Evaluar usuario.

Tabla 28 Prueba de aceptación sobre la Historia de usuario *Evaluar usuario*

Caso de prueba	
Código: HU2-P2	Historia de usuario: 2
Nombre: Evaluar usuario.	
Descripción: Prueba para la funcionalidad evaluar usuario.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El experto debe estar autenticado en la plataforma. • Debe tener asignado el permiso de evaluador. 	

Entradas/Pasos de ejecución:

- Acceder al sistema.
- Escoger un proyecto.
- Seleccionar la opción Ejecución/Recursos humanos en el menú.
- Seleccionar la pestaña Gestión.
- Seleccionar Evaluación Competencias 2-tuplas.
- Seleccionar la opción ver detalles.
- Seleccionar la opción evaluar.

Resultado de las pruebas: El sistema debe mostrar los indicadores para poder formular la evaluación de la competencia.

Evaluación de la prueba: Satisfactorio.

Anexo 14: Prueba de aceptación. Eliminar evaluación.

Tabla 29 Prueba de aceptación sobre la Historia de usuario *Eliminar evaluación*

Caso de prueba	
Código: HU3-P3	Historia de usuario: 3
Nombre: Eliminar evaluación.	
Descripción: Prueba para la funcionalidad eliminar evaluación.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El experto debe estar autenticado en la plataforma. • Debe tener asignado el permiso de evaluador. 	
Entradas/Pasos de ejecución:	
<ul style="list-style-type: none"> • Acceder al sistema. • Escoger un proyecto. • Seleccionar la opción Ejecución/Recursos humanos en el menú. • Seleccionar la pestaña Gestión. • Seleccionar Evaluación Competencias 2-tuplas. 	

<ul style="list-style-type: none"> • Seleccionar la opción ver detalles. • Seleccionar la opción eliminar.
Resultado de las pruebas: El sistema debe eliminar la evaluación del experto autenticado en el sistema.
Evaluación de la prueba: Satisfactorio.

Anexo 15: Prueba de aceptación. Ver detalles.

Tabla 30 Prueba de aceptación sobre la Historia de usuario Ver detalles

Caso de prueba	
Código: HU4-P4	Historia de usuario: 4
Nombre: Ver detalles.	
Descripción: Prueba para la funcionalidad ver detalles.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El experto debe estar autenticado en la plataforma. 	
Entradas/Pasos de ejecución:	
<ul style="list-style-type: none"> • Acceder al sistema. • Escoger un proyecto. • Seleccionar la opción Ejecución/Recursos humanos en el menú. • Seleccionar la pestaña Gestión. • Seleccionar Evaluación Competencias 2-tuplas. • Seleccionar la opción ver detalles. 	
Resultado de las pruebas: El sistema debe mostrar el listado de las competencias correspondientes al usuario seleccionado, así como los valores de los indicadores de eficiencia, eficacia, efectividad y consistencia del conjunto.	
Evaluación de la prueba: Satisfactorio.	

Anexo 16: Prueba de aceptación. Exportar a pdf.

Tabla 31 Prueba de aceptación sobre la Historia de usuario *Exportar a pdf*

Caso de prueba	
Código: HU5-P5	Historia de usuario: 5
Nombre: Exportar a pdf.	
Descripción: Prueba la funcionalidad de exportar a pdf.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El experto debe estar autenticado en la plataforma. 	
Entradas/Pasos de ejecución:	
<ul style="list-style-type: none"> • Acceder al sistema. • Escoger un proyecto. • Seleccionar la opción Ejecución/Recursos humanos en el menú. • Seleccionar la pestaña Gestión. • Seleccionar Evaluación Competencias 2-tuplas. • Seleccionar la opción ver detalles. • Seleccionar la opción exportar a pdf. 	
Resultado de las pruebas: El sistema guarda las evaluaciones en pdf.	
Evaluación de la prueba: Satisfactoria.	

Anexo 17: Prueba de aceptación. Exportar a csv.

Tabla 32 Prueba de aceptación sobre la Historia de usuario *Exportar a csv*

Caso de prueba	
Código: HU6-P6	Historia de usuario: 6
Nombre: Exportar a csv.	
Descripción: Prueba la funcionalidad de exportar a csv.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El experto debe estar autenticado en la plataforma. 	
Entradas/Pasos de ejecución:	

- Acceder al sistema.
- Escoger un proyecto.
- Seleccionar la opción Ejecución/Recursos humanos en el menú.
- Seleccionar la pestaña Gestión.
- Seleccionar Evaluación Competencias 2-tuplas.
- Seleccionar la opción ver detalles.
- Seleccionar la opción exportar a csv.

Resultado de las pruebas: El sistema guarda las evaluaciones en csv.

Evaluación de la prueba: Satisfactoria.

Anexo 18: Prueba de aceptación. Asignar peso a los expertos.

Tabla 33 Prueba de aceptación sobre la Historia de usuario *Asignar pesos a los expertos.*

Caso de prueba	
Código: HU7-P7	Historia de usuario: 7
Nombre: Asignar pesos a los expertos.	
Descripción: Prueba la funcionalidad de asignar peso a los expertos.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado en la plataforma como administrador. 	
Entradas/Pasos de ejecución:	
<ul style="list-style-type: none"> • Acceder al sistema. • Seleccionar la opción Configuración/Recursos humanos en el menú. 	
Resultado de las pruebas: El sistema asigna un peso al experto.	
Evaluación de la prueba: Satisfactoria.	

Anexo 19: Cuestionario aplicado a los trabajadores del departamento XEDRO-GESPRO.

Método para la evaluación multicriterio de recursos humanos basado en el modelo 2-tuplas para XEDRO-GESPRO 18.05.



Lea con cuidado cada pregunta antes de responder. En este cuestionario usted no tiene que poner su nombre. Le agradecemos su participación y honestidad.

1- ¿Se siente satisfecho con el resultado que obtuvo con la utilización del método 2tuplas sobre la evaluación por competencias?

Si No sé No

2- ¿Qué importancia le concede a la utilización de este método?

3- ¿Siente usted que este método le es útil para ser utilizado dentro de la plataforma GESPRO?

Si No sé No

4- ¿Qué aspectos, a su juicio, limitan el uso de este método?

5- ¿Le gusta la forma en que se muestra este método?

- Me gusta mucho.
- No me gusta tanto.
- Me da lo mismo.
- Me disgusta más de lo que me gusta.
- No me gusta nada.
- No sé qué decir.

Anexo 20: Cuadro Lógico de Iadov.

	1. ¿Se siente satisfecho con el resultado que obtuvo con la aplicación del método 2tuplas sobre la evaluación por competencia?								
	Si			No sé			No		
5. ¿Le gusta la forma en que se muestra el método?	3. ¿Siente usted que este método es útil para ser usado en la plataforma Gespro?								
	Si	No sé	No	Si	No sé	No	Si	No sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
No me gusta tanto	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No se que decir	2	3	6	3	3	3	6	3	4

Anexo 21: Niveles de satisfacción expresados en escala numérica para calcular ISG

Escala	Resultado
+1	Máximo de satisfacción
0,5	Más satisfecho que insatisfecho
0	No definido y contradictorio
- 0,5	Más insatisfecho que satisfecho
-1	Máxima insatisfacción