

DESARROLLO DE UN PROTOTIPO DE SISTEMA EXPERTO PARA EL APOYO
EN LA TOMA DE DECISIONES DEL PROCESO DE SELECCIÓN DE PERSONAL

ESTEBAN ANDRÉS PÉREZ VARGAS

UNIVERSIDAD EAFIT
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS
MEDELLÍN
2011

DESARROLLO DE UN PROTOTIPO DE SISTEMA EXPERTO PARA EL APOYO
EN LA TOMA DE DECISIONES DEL PROCESO DE SELECCIÓN DE PERSONAL

ESTEBAN ANDRÉS PÉREZ VARGAS

Código: 200029101010

Proyecto de grado presentado como requisito para optar al título de
Ingeniero de Sistemas

Asesor

ÁLVARO JARAMILLO FLÓREZ

Psicólogo, especialista en sistemas de información

UNIVERSIDAD EAFIT
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS
MEDELLÍN
2011

Nota de aceptación

Presidente del jurado

Jurado

Jurado

Medellín, Agosto 8 de 2011

CONTENIDO

	pág.
INTRODUCCIÓN	14
1. JUSTIFICACIÓN	16
2. OBJETIVOS	18
2.1 OBJETIVO GENERAL	18
2.2 OBJETIVOS ESPECÍFICOS	18
3. ALCANCE Y ENTREGABLES	19
3.1 ALCANCE	19
3.2 ENTREGABLES	19
4. MARCO TEÓRICO	20
4.1 LA SELECCIÓN DE PERSONAL	20
4.1.1 Flujo de trabajo y tareas en el proceso.	20
4.1.2. Antecedentes en la aplicación de la inteligencia artificial en la selección de personal	23
4.2 LA INTELIGENCIA ARTIFICIAL	23
4.2.1 Historia de la inteligencia artificial	25
4.2.2 Fundamentos de la inteligencia artificial	28
4.2.3 Ramas de la inteligencia artificial.	30
4.3 LOS SISTEMAS EXPERTOS	30
4.3.1 Historia de los sistemas expertos	31
4.3.2 Estructura básica de un sistema experto	33
4.3.3 Ventajas y desventajas	34
4.3.4 Aplicabilidad	35
4.3.5 Sistemas expertos con lógica difusa	36
4.3.6 Ingeniería del conocimiento	38
4.4 PARADIGMAS Y LENGUAJES DE PROGRAMACIÓN	39
4.4.1 Programación declarativa	39
4.4.2 Programación imperativa	40

4.4.3 Computación en la nube	40
4.4.4 Ambiente de desarrollo usado en el proyecto	44
4.5 LA METODOLOGÍA COMMONKADS PARA INGENIERÍA DEL CONOCIMIENTO	48
4.5.1 Ciclo de vida.	50
4.5.2 Estructura de CommonKADS	51
4.5.3 Modelo de la Organización	53
4.5.4 Modelo de Tareas	55
4.5.5 Modelo de Agentes	58
4.5.6 Otras plantillas en el modelado del contexto	58
4.5.7 Modelo de Conocimientos	59
4.5.8 Modelo de Comunicaciones	61
4.5.9 Modelo de Diseño	63
5. METODOLOGÍA DE TRABAJO	67
5.1 ELICITACIÓN DE REQUISITOS CON LA HERRAMIENTA REM	69
5.2 MODELADO DEL SISTEMA USANDO UML EN COMMONKADS	71
5.2.1 Equipo de trabajo	72
5.3 ELECCIÓN DE LA HERRAMIENTA DE DESARROLLO (LENGUAJE DE PROGRAMACIÓN DECLARATIVA)	72
5.3.1 Parámetros obligatorios para elegir el lenguaje	73
5.3.2 Parámetros adicionales	74
5.3.3 ¿Por qué no se usó un lenguaje declarativo en el desarrollo del sistema?	76
6. RESULTADOS	77
6.1 ELICITACIÓN DE REQUISITOS CON LA HERRAMIENTA REM	77
6.2 INGENIERÍA DEL CONOCIMIENTO CON COMMONKADS	77
6.2.1 OM-1: Identificación de problemas y oportunidades orientados al conocimiento	77
6.2.2 OM-2: Descripción de los aspectos de la organización que tienen impacto en o están afectados por el problema elegido	79
6.2.3 OM-3: Descripción del proceso desde el punto de vista de las tareas en que está compuesto y sus características principales	83
6.2.4 OM-4: Descripción del componente de conocimiento del modelo de la organización y sus principales características	84
6.2.5 OM-5: lista de chequeo para el documento de viabilidad de la decisión	86
6.2.6 TM-1: Descripción refinada de las tareas dentro del proceso objetivo	87
6.2.7 TM-2: Especificación del conocimiento empleado por una tarea y posibles cuellos de botella y áreas para mejorar	88

6.2.8. AM-1: Especificaciones del agente	89
6.2.9 OTA-1: Lista de chequeo para decisiones sobre impacto y mejoras	90
6.2.10 Modelado del conocimiento para la tarea ECO - Evaluar competencias personales del candidato	91
6.2.11 Modelo de Comunicaciones	101
6.2.12 Modelo de Diseño	105
6.3 ELECCIÓN DE UN LENGUAJE DE PROGRAMACIÓN DECLARATIVA	110
6.3.1 Lenguajes a considerar	110
6.3.2 Selección del Lenguaje de Programación	110
6.3.3 Lenguaje seleccionado: CLIPS	112
7. CONCLUSIONES	114
BIBLIOGRAFÍA	116

LISTA DE ANEXOS

	pág.
ANEXO A. ELICITACIÓN DE REQUISITOS	121
ANEXO B. INTERFACES DEL SISTEMA DESARROLLADO	131
B.1 Administración de Competencias	132
B.2 Administración de Perfiles de Cargos	134
B.3. Administración de Candidatos y sus Perfiles Psicológicos	136
B.4. Evaluación de Competencias Para Un Cargo	137
ANEXO C. CASOS DE PRUEBA	140

LISTA DE TABLAS

	pág.
Tabla 1. Aplicaciones de los Sistemas Expertos	36
Tabla 2. Ventajas y desventajas de la Computación en la Nube	42
Tabla 3. OM-2: Descripción de los aspectos de la organización que tienen impacto en o están afectados por el problema elegido.	53
Tabla 4. TM-1: Descripción refinada de las tareas dentro del proceso objetivo	56
Tabla 5. TM-2: Especificación del conocimiento empleado por una tarea y posibles cuellos de botella y áreas para mejorar.	57
Tabla 6. AM-1: Especificaciones del Agente.	58
Tabla 7. CM-1: Especificación de las transacciones que posibilitan el diálogo entre dos Agentes.	62
Tabla 8. CM-2: Especificación de los mensajes y los puntos de información que hacen una transacción individual	63
Tabla 9. DM-1: Descripción de la arquitectura del sistema.	64
Tabla 10. DM-2: Especificación de las facilidades ofrecidas por y en el sistema que será implantado	64
Tabla 11. DM-3: Lista de chequeo de decisiones en relación con la especificación de la arquitectura.	65
Tabla 12. DM-4: Decisiones del diseño de la aplicación.	66
Tabla 13. Iteraciones del modelo de desarrollo en espiral	68
Tabla 14. Descripción del proceso de Selección de Personal desde el punto de vista de las tareas en que está compuesto.	84
Tabla 15. Descripción de los activos de conocimiento organizacional para la Selección de Personal.	85
Tabla 16. TM-1 para la Tarea ECO	87
Tabla 17. TM-2 para el activo de conocimiento EVA	88
Tabla 18. AM-1 para el Agente Seleccionador de Personal	90
Tabla 19. Ejemplo de Perfiles Psicológicos de tres Candidatos.	92

Tabla 20. Ejemplo de cómo obtener los parámetros para hallar el Cumplimiento de cada Competencia mediante la Función Lambda.	94
Tabla 21. Ejemplo del cálculo del Cumplimiento de cada Candidato para cada Competencia.	94
Tabla 22. Ejemplo de definición de tres Perfiles de Cargo, mediante CDVIC para cada Competencia en cada Cargo	95
Tabla 23. CDVIC correspondiente al Perfil del Cargo de Gerente	99
Tabla 24. Tres CDVIC correspondientes a los Cumplimientos para cinco Competencias	100
Tabla 25. Cálculo de la Distancia entre cada uno de los tres Candidatos y el Cargo de Gerente, mediante la Distancia Normalizada de Hamming usando la Métrica de Hausdorff.	100
Tabla 26. CM-1: Transacción para ordenar la evaluación de competencias personales para un cargo.	103
Tabla 27. CM-1: Transacción para solicitar a la base de datos los perfiles psicológicos de los candidatos y el perfil del cargo.	103
Tabla 28. CM-1: Transacción para obtener el perfil del cargo y los perfiles psicológicos de los candidatos desde la base de datos	104
Tabla 29. CM-1: Transacción para reportar los resultados arrojados por el sistema al Seleccionador	104
Tabla 30. DM-1: Descripción de la arquitectura del sistema.	106
Tabla 31. DM-2: Especificación de las facilidades ofrecidas por y en el sistema que será implantado.	107
Tabla 32. Lista de controladores y las acciones y vistas asociados a cada uno.	108
Tabla 33. Comparación de lenguajes de programación declarativa para Inteligencia Artificial	111
Tabla 34. Ejemplo de definición de competencias y la función para determinar el Cumplimiento a partir del puntaje del candidato en una Prueba Psicológica C8	140
Tabla 35. Ejemplo de definición de Perfiles de Cargos	140
Tabla 36. Ejemplo de las Pruebas Psicológicas de los Candidatos.	141

Tabla 37. Ejemplo del Cumplimiento de cada Candidato con cada Competencia.	142
Tabla 38. Ejemplo de Evaluación de Competencias para el Cargo Extremo 1.	143
Tabla 39. Ejemplo de Evaluación de Competencias para el Cargo Extremo 2.	143
Tabla 40. Ejemplo de Evaluación de Competencias para el Cargo de Director de Mercadeo.	144
Tabla 41. Ejemplo de Evaluación de Competencias para el Cargo de Ingeniero de Software.	144

LISTA DE ILUSTRACIONES

	pág.
Ilustración 1. Estructura general del prototipo desarrollado.	15
Ilustración 2. Estructura básica de un Sistema Experto.	33
Ilustración 3. Ejemplo de pertenencia a un conjunto difuso	37
Ilustración 4. Modelos de CommonKADS.	51
Ilustración 5. Mapa de las Tareas Intensivas en Conocimiento de CommonKADS	61
Ilustración 6. Iteraciones del modelo de desarrollo en espiral	68
Ilustración 7. Tareas en la elicitación de requisitos	70
Ilustración 8. Roles y responsables en el proyecto	72
Ilustración 9. Mapa Conceptual de Métodos Expertos	78
Ilustración 10. Diagrama de actividades (UML) del proceso de Selección de Personal.	82
Ilustración 11. Diagrama de casos de uso (UML) en el proceso de Selección de Personal	83
Ilustración 12. Diagrama de clases UML para la evaluación de competencias personales	91
Ilustración 13. Aplicación de la Función Lambda (Γ) para calcular el Cumplimiento de un Candidato, partiendo del puntaje o medida obtenido por el Candidato, para cada una de las Competencias a evaluar.	93
Ilustración 14. Esquema del dominio para la tarea ECO	96
Ilustración 15. Conocimiento de inferencia para la tarea ECO.	97
Ilustración 16. Distancia de Hamming Normalizada entre dos CDVIC, mediante la Métrica de Hausdorff.	98
Ilustración 17. Aplicación de la Métrica de Hausdorff para determinar la distancia entre las formas de dos objetos X y Y.	99
Ilustración 18. Conocimiento de tarea para ECO.	100
Ilustración 19. Plan de comunicaciones representado en un Diagrama de estados UML.	102

Ilustración 20. Descripción del MVC según CommonKADS.	105
Ilustración 21. Subsistemas a implementar en el MVC, representando los elementos del modelado de conocimientos.	106
Ilustración 22. Secciones de la interfaz del sistema.	131
Ilustración 23. Despliegue de la administración de competencias en el Panel Principal.	132
Ilustración 24. Cuadro de diálogo para crear y editar competencias.	133
Ilustración 25. Despliegue de la administración de perfiles de cargos en el Panel Principal.	134
Ilustración 26. Cuadro de diálogo para crear y editar perfiles de cargos.	135
Ilustración 27. Despliegue de la administración de candidatos en el Panel Principal.	136
Ilustración 28. Cuadro de diálogo para crear y editar candidatos.	137
Ilustración 29. Cuadro de diálogo para seleccionar cargo para el cual se va a evaluar las competencias de los candidatos.	138
Ilustración 30. Evaluación de competencias para un cargo.	139

AGRADECIMIENTOS

Quiero agradecer a mis padres Gloria y Robinson, familiares y amigos; todas esas personas que han estado presentes para apoyarme incondicionalmente en lo profesional, académico y afectivo durante todos estos años.

Naturalmente, deseo manifestar mi agradecimiento al asesor de este proyecto de grado, Álvaro Jaramillo Flórez, quien me dio la idea para este proyecto y me apoyó para convertirla en realidad.

INTRODUCCIÓN

En el contexto económico mundial se requiere más que nunca que los cargos de las organizaciones estén ocupados por las personas más convenientes, evaluadas desde una perspectiva integral que incluya lo psicológico, lo académico y lo laboral. La Selección de Personal (SP) es el proceso donde se determina, para cada cargo empresarial, quién es la persona más conveniente para ese cargo.

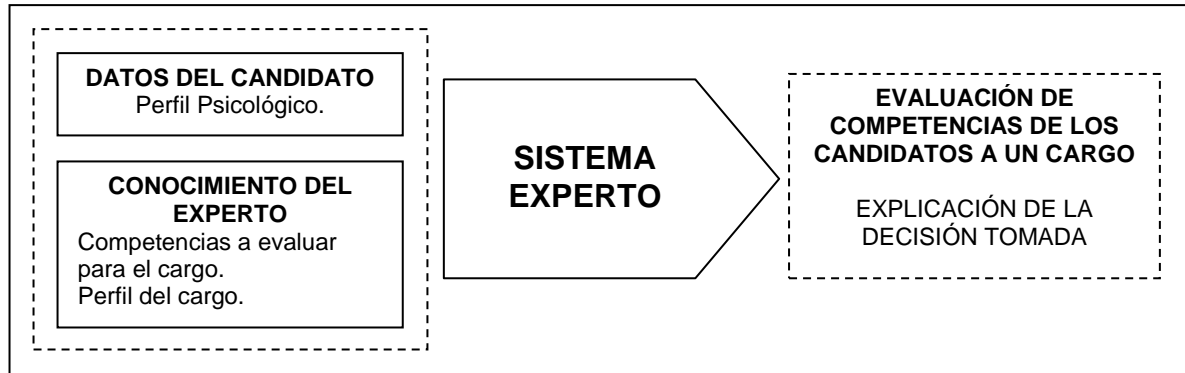
Es muy importante encontrar mecanismos que ayuden a las empresas a hacer procesos de SP de muy alta calidad, pero la complejidad de las organizaciones y los cargos hace que esto no sea fácil. Después de muchas teorías postuladas por estudiosos de diversas áreas de la Psicología, aún no existe la última palabra sobre el modelo más conveniente para la Selección de Personal. El proceso se desarrolla en un alto nivel de incertidumbre y requiere de la experiencia de expertos para tomar decisiones en contextos muy variados.

Mediante este proyecto se busca apoyar el proceso de SP en las organizaciones, desde el conocimiento adquirido en el pregrado de Ingeniería de Sistemas, para lo cual se ha desarrollado un sistema de apoyo a la toma de decisiones en SP. Siendo la SP un área tan dependiente de la inteligencia humana, se consideró que la Inteligencia Artificial podría dar una serie de herramientas muy útiles para poder simular el razonamiento de un experto en SP. Para abstraer el conocimiento de los expertos a un nivel computacional, se ha trabajado en dos de las ramas de la Inteligencia Artificial: los Sistemas Expertos y la Lógica Difusa.

Es importante mencionar algunas consideraciones sobre el papel cada vez más relevante de la Inteligencia Artificial en la vida de los seres humanos. La capacidad de adquisición, almacenamiento y procesamiento de datos crece de forma exponencial, con tecnologías cada vez más sofisticadas y baratas. En esta abundancia de recursos informáticos hay grandes posibilidades de extraer información valiosa, pero para ello se requiere conocer los procesos y razonamientos humanos para dar "inteligencia" a los métodos computacionales que analizan los datos y la información para extraer conocimiento que apoye la toma de decisiones a nivel operativo y estratégico.

Aunque no se puede modelar aún una herramienta que pueda reemplazar completamente al experto de SP, un ser humano, el sistema desarrollado puede proveer un soporte importante al experto en el desempeño de su actividad laboral, específicamente en la Evaluación de Competencias de los candidatos a un cargo específico en la organización.

Ilustración 1. Estructura general del prototipo desarrollado.



El sistema procesa datos correspondientes a puntajes de cada candidato en un perfil psicológico, el cual está definido en términos de puntajes para varias competencias. También se debe conocer el perfil del cargo, definido en términos de las competencias medidas en los perfiles psicológicos de los candidatos.

Con el fin de simular el procesamiento de los datos para obtener la Evaluación de Competencias, fue necesario modelar conceptos que son muy subjetivos y difusos, propios del pensamiento humano. Este conocimiento en SP se modeló mediante funciones y operaciones matemáticas que pueden ayudar a simular el razonamiento del experto.

Es importante mencionar que una de las metas personales del autor de este proyecto es demostrar que la adopción de los Sistemas Expertos es viable en el sector empresarial, aún para las micro, pequeñas y medianas empresas. Esto se puede lograr mediante la revolución tecnológica que trae la llamada Computación en la Nube (Cloud Computing). Los Sistemas Expertos orientados a problemas bien definidos, que no requieren grandes inversiones de tiempo y dinero para su desarrollo e implementación, pueden dar un aporte significativo a las organizaciones.

1. JUSTIFICACIÓN

La Selección de Personal es uno de los procesos más importantes y complejos para las empresas. Es importante porque el entorno cambiante y competitivo de las empresas demanda que cada cargo organizacional sea ocupado por individuos con características especiales, indispensables para ejecutar adecuadamente y sobresalir en un cargo específico. Es compleja porque requiere que se trabaje con varios de los elementos más difíciles de determinar: las motivaciones, conocimientos, la personalidad y la experiencia de una persona, la cultura y las políticas internas organizacionales, la cultura del aspirante, la presión a la que será sometido, ente otros. La Selección de Personal es un proceso estratégico en cualquier organización.

Las organizaciones utilizan con relativo éxito estrategias de Selección de Personal que dan una buena aproximación a que se obtenga un empleado adecuado para un cargo específico. El problema está en la subjetividad de lo que significa encontrar la persona “apropiada” para el cargo, y en que el potencial error humano del seleccionador siempre está presente. A la naturaleza subjetiva de la Selección de Personal debe sumarse que un seleccionador de personal puede verse afectado por la falta de tiempo y la presión laboral, entre otros factores [6].

Mediante este proyecto se busca determinar si es posible integrar en un Sistema Experto los conocimientos y procedimientos del seleccionador; esto con el fin de minimizar la posibilidad de error y hacer más efectivo el proceso de selección. Con este sistema se podría lograr mayor precisión, rapidez, objetividad y confiabilidad en el proceso.

El desarrollo de las tecnologías de la información y las comunicaciones ha traído la oportunidad de desarrollar sistemas con menor complejidad y menor costo, pero es necesario generar conocimiento y educar sobre cómo aprovechar esas tecnologías para tomar provecho de los avances de la Inteligencia Artificial (IA)¹ [13]. Athappilly sugiere que el uso de las tecnologías emergentes para entornos web hicieron posible el desarrollo de sistemas de IA de una forma mucho menos compleja y menos costosa. Sin embargo muchos usuarios del sector empresarial no conocen o no tienen conciencia de la oportunidad que esto representa. En consecuencia, se hace necesario educar y generar conocimiento sobre el potencial que tienen las nuevas tecnologías para lograr negocios más eficientes y competitivos [38].

¹ DUAN, Y.; EDWARDS, J.S. *Web-based Expert Systems: benefits and challenges*. Information & Management, 42. 2005.

La Ingeniería del Conocimiento (IC) busca el desarrollo estructurado de sistemas que permiten una gestión automatizada del conocimiento para solucionar problemas complejos en las organizaciones. Es necesario explorar y desarrollar las grandes posibilidades que esto ofrece. Sommerville [8] menciona que la IA, y específicamente la IC, complementan a la Ingeniería de Software (IS) tradicional en la búsqueda de soluciones a problemas donde la IS ha tenido un éxito limitado, que requieren una aproximación más acorde al conocimiento de los negocios. Este proyecto busca emplear recursos que ofrece la IC para apoyar la toma de decisiones en un proceso tan importante para las organizaciones como es la Selección de Personal.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Desarrollar un prototipo de Sistema Experto que contenga el conjunto de procesos de razonamiento y conocimiento requeridos por un experto en selección de personal, utilizando lenguajes de programación declarativa de libre distribución y que sirva de apoyo para la toma de decisiones.

2.2 OBJETIVOS ESPECÍFICOS

2.2.1 Elicitar los requisitos del sistema con la herramienta REM (Requirements Management), para conocer los requerimientos y necesidades del usuario final y la funcionalidad del sistema.

2.2.2 Modelar la estructura de la solución del sistema con UML (Unified Modeling Language), con el fin de orientar el desarrollo de los elementos del software.

2.2.3 Elegir la herramienta apropiada para el desarrollo del prototipo, por medio de un estudio de las características de los principales lenguajes de programación de Inteligencia Artificial (IA) de libre distribución.

2.2.4 Desarrollar la aplicación con el lenguaje seleccionado y de acuerdo con los requisitos y el modelo ya establecidos, para crear el motor de inferencia y elaborar la base de conocimientos del experto.

2.2.5 Realizar las pruebas que permitan comprobar la confiabilidad en el resultado arrojado por el prototipo.

3. ALCANCE Y ENTREGABLES

3.1 ALCANCE

El resultado final del proyecto es el desarrollo en software del prototipo del Sistema Experto (SE) propuesto, orientado a apoyar la toma de decisiones en el proceso de Selección de Personal en una organización.

En este caso Prototipo se refiere a un sistema de software que permitirá demostrar la funcionalidad de un SE mediante un conjunto de operaciones matemáticas que simulan el *razonamiento* del sistema. En este trabajo el término Prototipo no se refiere conjunto de interfaces gráficas de un sistema de información.

El Prototipo que se entregará funcionará con conocimientos de un experto en Selección de Personal por lo que alcanzará un nivel de funcionalidad aceptable para demostración real.

3.2 ENTREGABLES

- Documento de la Investigación.
- Elicitación de requisitos del sistema en formato de la herramienta REM (REquisite Management).
- Modelado del sistema mediante la metodología CommonKADS, usando diagramas UML (Lenguaje de Modelado Unificado).
- Prototipo funcional del Sistema Experto, conteniendo Casos de Prueba que demuestren el funcionamiento de la aplicación.

4. MARCO TEÓRICO

4.1 LA SELECCIÓN DE PERSONAL

La selección de personal es uno de los principales procesos de cualquier empresa u organización, ya que suministra los trabajadores adecuados para ejercer las funciones de cada uno de los cargos. Este proceso es llevado a cabo por personal especializado en el tema, pero los profesionales de todas las áreas se ven involucrados en algún momento del proceso.

La vinculación de una persona a la organización le permitirá hacer parte de los grupos de trabajo e influirá sobre el ambiente organizacional y las relaciones interpersonales.

Los procesos de selección en las empresas por lo general son costosos y toman mucho tiempo. Esto conlleva la necesidad de que se haga la mejor selección posible para evitar pérdidas de tiempo y logística.

4.1.1 Flujo de trabajo y tareas en el proceso. A continuación se enuncian en orden todas las tareas involucradas en el proceso de selección de personal. El proceso se inicia con la llegada de una solicitud para llenar un cargo vacante en la organización.

- **Recibir requerimiento para un Cargo.** Se recibe una solicitud formal para llevar un cargo vacante. La solicitud puede provenir de un empleador y/o un área específica de una organización
- **Definir perfil del cargo.** El perfil del cargo describe las funciones y responsabilidades que tendrá el nuevo ocupante, así como las competencias requeridas para un buen desempeño. El perfil del cargo determina las pruebas de habilidad que se van a presentar, la forma como se conducen las entrevistas y las pruebas técnicas o de conocimientos a que haya lugar.

El perfil es donde se definen las tareas o funciones que se van a realizar en el cargo, el tipo de relaciones que se deben tener con los demás, el tipo de proveedores de insumos o información que tendrá en su cargo, el tipo de productos y el tipo de clientes que su producto tendrá. Es importante determinar, por ejemplo, si se requiere alguien con gran capacidad de concentración y trabajar largos períodos de tiempo sin contacto con otras personas o si por el contrario se

requiere de alguien con gran habilidad para relacionarse con los demás y con gran iniciativa para ello.

Cuando se levanta el perfil del cargo también se especifica la experiencia que deben tener los aspirantes. Esto se determina de acuerdo con el tipo de función que se espera que realice la persona, lo particular que sea esa función y las políticas que al respecto se tengan en la empresa.

- **Buscar Hojas de Vida.** Es importante determinar el ámbito de búsqueda de quien va a llenar la vacante. Si se requiere que los candidatos a un cargo provengan exclusivamente del interior de la organización, se debe hacer una búsqueda interna de hojas de vida (Proceso Interno). Si cualquier candidato puede aplicar al cargo, se debe hacer una búsqueda externa (Proceso Externo). Esto puede marcar diferencias en el proceso de selección, porque al ser un proceso interno podrían omitirse pasos en el proceso (la visita domiciliaria, la búsqueda de referencias o algunas de las pruebas, por ejemplo) lo que a su vez bajaría costos al proceso.
- **Aplicar pruebas psicológicas al candidato.** La cantidad y tipo de pruebas que se lleven a cabo dependerá del interés de la empresa, sus políticas, el tiempo disponible para la selección, el tipo de cargo a proveer, etc. Por lo general se consideran tres tipos de pruebas de corte psicológico:
 - Evaluación de personalidad, para establecer la forma de ser del candidato y su conducta más típica.
 - Evaluación de motivación, para establecer los intereses que lo mueven al trabajo y a la vida cotidiana.
 - Evaluación de habilidades, para establecer las habilidades específicas para su mejor desempeño.
- **Evaluar competencias del candidato para el cargo.** Cada una de las pruebas psicológicas servirá para medir qué tanto cumple el candidato con una serie de **Competencias**. Una Competencia ser definida como una característica subyacente de un individuo, que está causalmente relacionada con un rendimiento efectivo o superior en una situación o trabajo, definido en términos de un criterio. Una Competencia también puede ser un conjunto de patrones de conducta, que la persona debe llevar a un cargo para rendir eficientemente en sus tareas y funciones. En síntesis, una Competencia puede ser un conjunto de patrones compuestos de características subyacentes a la persona (conocimientos, habilidades, disposiciones, conductas, etc.) que permiten al individuo alcanzar un rendimiento efectivo o superior en una actividad o trabajo. Las Competencias son individuales y no transferibles ni imitables, constituyendo un recurso valioso para la empresa [47].

Se debe evaluar las Competencias del candidato de tal forma que se pueda saber cuál de todos los candidatos se adapta mejor al Perfil del Cargo, desde el punto de vista de las Competencias necesarias.

Este trabajo de grado busca dar un apoyo a la toma de decisiones en la ejecución de esta tarea.

- **Aplicar Entrevista Estructurada al Candidato.** Se puede aplicar una o varias entrevistas al candidato, dependiendo de las políticas de la empresa, de las necesidades y de las variables de cada proceso (Interno o Externo, por ejemplo). Las entrevistas las hacen normalmente el psicólogo y el jefe directo. En ocasiones, también participa el gerente de la empresa. La entrevista tiene por objetivo aclarar dudas, medir empatía, observar ajuste del candidato a las normas y costumbres de la empresa, etc. Con frecuencia esta entrevista es decisoria para la vinculación.
- **Verificar referencias laborales.** Este paso tiene por objetivo medir la percepción que sobre el aspirante tienen personas que lo conocen en el plano laboral. Se busca saber cómo ha sido el desempeño y cómo han sido las relaciones con las personas con las que ha trabajado recientemente, y cuáles han sido los motivos o las circunstancias en que ha terminado esos trabajos.
- **Visitar al candidato en su domicilio.** Es una forma de conocer de primera mano sobre el nivel de vida, la calidad de vida y las relaciones del aspirante en su hogar. También son observados la vida social, el impacto que ejerce, la forma en que se relaciona con el entorno y el estilo de vida y cultura personal.
- **Evaluar ajuste del candidato al cargo.** En este punto se integran todos los datos recogidos y se producen unas conclusiones sobre ellos. Esta evaluación puede incluir recomendaciones y observaciones que servirán para el mejoramiento del candidato en caso de ser elegido.

La selección de personal es un campo aún no totalmente determinado ni se tienen sobre él verdades que se puedan aplicar sin temor al error. En vista de la complejidad psicológica de cada persona, puede haber errores y se puede elegir equivocadamente. Por este motivo psicólogos y empresas que se dedican a asesorías en el reclutamiento están permanentemente en la búsqueda de nuevos elementos que permitan determinar con mayor precisión las capacidades, aptitudes y actitudes de las personas.

Dependiendo de las orientaciones que tengan los encargados de los procesos, se dará mayor o menor relevancia a las pruebas psicológicas, las entrevistas, las referencias, la experiencia académica y laboral, etc. Cada uno de estos elementos tendrá entonces un peso específico a la hora de la decisión final.

- **Aceptación o rechazo del candidato y notificación.** Con base en la evaluación del ajuste del candidato y su comparación con los demás candidatos en el proceso de selección, se rechaza o se acepta el candidato y se notifica la decisión.

4.1.2. Antecedentes en la aplicación de la inteligencia artificial en la selección de personal. La Inteligencia Artificial ha sido ampliamente aplicada al desarrollo de sistemas que soportan la Selección de Personal. Han sido implementados desde sistemas para la minería de datos de personal para trabajar en empresas de alta tecnología, hasta sistemas que filtran hojas de vida para seleccionar los candidatos más aptos para un cargo. Otros sistemas buscan establecer modelos para comparar candidatos y seleccionar al más apropiado. Gran parte del desarrollo de sistemas basados en Inteligencia Artificial se dedican a recolectar y clasificar las hojas de vida de los candidatos de acuerdo a especificaciones de los empleadores. Aunque se cuenta con varias aproximaciones desde la Inteligencia Artificial, aún es un reto implementar una evaluación global de las capacidades de cada candidato en comparación con los demás candidatos y teniendo en cuenta todas las características del cargo que piensa ocupar [47, 50, 54].

Dentro del proceso de Selección de Personal, este proyecto está orientado específicamente a la tarea de evaluación de las competencias de los candidatos. En esta en concreto, es necesario un enfoque que permita tener en cuenta la variabilidad de los aspectos personales y culturales de los candidatos, los cargos que ocupan, y las organizaciones. Esto significa que se requieren enfoques adaptables a diversos contextos. Lo anterior sumado a la incertidumbre y la complejidad de los factores que afectan la selección de personal, incluyendo los métodos y la cultura de cada organización, hacen difícil definir un modelo único para solucionar todos los casos que se presenten a los seleccionadores. Se requieren aproximaciones más flexibles o adaptables a contextos específicos² [52].

4.2 LA INTELIGENCIA ARTIFICIAL

Inteligencia puede ser definida como la capacidad (orgánica o computacional) de lograr metas en el mundo. Esto incluye (pero no se limita a) la capacidad de razonar, abstraer problemas complejos y plantear soluciones. Los expertos hablan

² ARIAS, Yanet; ROSETE, Alejandro; MARTÍNEZ, Raúl. *Propuesta informática para seleccionar personal por competencias utilizando técnicas de Inteligencia Artificial*. Centro de Estudios de Ingeniería de Sistemas. Instituto Superior Politécnico José Antonio Echeverría. Cuba. 2006.

de varios grados de inteligencia en los seres humanos, varios animales y algunas computadoras. Por su parte, diversas fuentes definen Inteligencia como la capacidad que tienen muchos seres vivos de planear, resolver problemas, pensar en abstracto, aprender y comprender ideas. La inteligencia involucra mecanismos que, si son entendidos correctamente, pueden ser implementados por computadora y ser ejecutados con un buen desempeño.

Sin embargo, hay otros mecanismos relacionados con la inteligencia que aún no han sido entendidos, por lo que no han podido ser programados [9, 10]. La posibilidad de simular la mente humana en una máquina fue una obsesión para filósofos, científicos, humanistas y un gran grupo de personas que sospechaban el poder del pensamiento humano. A lo largo de la historia se invirtieron grandes esfuerzos en la búsqueda de este objetivo. Sin embargo, la tecnología existente antes del siglo XX no permitía hacer grandes avances [1, 3, 7].

Aún el día de hoy no se conoce con exactitud el mecanismo cerebral que permite al ser humano ser inteligente. Sin embargo, el nacimiento y auge de las Ciencias de la Computación en la década de 1950 fue vista como una oportunidad para quienes creen que simular la mente en una máquina es posible. La Inteligencia Artificial (IA) surge con el fin de comprender cómo funciona la mente, de tal forma que se pueda hacer computable. En 1950 Alan Turing presenta su Test de Turing, fundamentado en la hipótesis de que, si una máquina se comporta en todos los aspectos como inteligente, entonces debe ser inteligente³ [2].

No hay una definición unificada para Inteligencia Artificial. Una definición muy extendida es la de Patrick Hayes, quien consideró que la Inteligencia Artificial es la implementación de razonamientos inteligentes mediante técnicas propias de la Computación. Otra definición que también se acepta es la de Marvin Minsky, quien indica que la Inteligencia Artificial es una ciencia dedicada a la construcción de máquinas que implementan tareas propias de humanos dotados de Inteligencia⁴ [1].

La IA está estrechamente relacionada con tareas que impliquen el uso de computadoras para entender la inteligencia humana, lo cual tiene como objetivo la creación de programas para máquinas que imiten el comportamiento y la comprensión de los seres humanos. La investigación en el campo de la IA se caracteriza por la producción de máquinas para la automatización de tareas que requieran un comportamiento inteligente [9].

³ REINGOLD, E. *The Turing Test*. University of Toronto. 1999.

⁴ BUCHANAN, B. G. *History of Artificial Intelligence*. American Association for Artificial Intelligence. 2005.

La IA se ha convertido en una disciplina científica, enfocada en proveer soluciones a problemas de la vida cotidiana de los seres humanos. Los sistemas de IA actualmente son parte de la rutina en campos como Economía, Medicina, Ingeniería y la Milicia, y se ha usado en gran variedad de aplicaciones de software, juegos de estrategia como ajedrez de computador y otros videojuegos. La gran necesidad de llevar servicios inteligentes a través de las herramientas tecnológicas modernas hace que se incentive la investigación y el desarrollo de la IA [4]. Esta tendencia es ampliamente fortalecida por los recientes avances en Neurociencias, Psicología y las Ciencias de la Computación.

4.2.1 Historia de la inteligencia artificial. Podría afirmarse que el primer indicio conocido del estudio de la IA ocurre en el año 450 a. C., cuando Platón cita un diálogo en el que Sócrates le pregunta a Eutidemo: “Desearía saber cuál es la característica de la piedad que hace que una acción se pueda considerar como pía... y así la observe y me sirva de norma para juzgar tus acciones y las de otros.”

En la Antigua Grecia, los mitos de Hefesto y Pigmalión incorporan la idea de robots inteligentes. En el siglo V A.C. Aristóteles inventa la Lógica de Silogismos, el primer sistema formal de razonamiento deductivo.

En el siglo XIII, Ramón Llull, teólogo español, inventa una máquina para descubrir verdades no matemáticas mediante la combinación de palabras de una lista.

Durante los siglos XV y XVI se inventa el reloj mecánico, el primer mecanismo de medición moderno. Varios fabricantes de relojes se dedican entonces a inventar animales mecánicos y otros inventos que buscan simular seres vivos.

A principios del siglo XVII, Rene Descartes propone que los cuerpos de los animales no son más que máquinas complejas. Muchos otros pensadores de ese siglo presentaron variaciones y elaboraciones del mecanismo cartesiano. Thomas Hobbes publica *Leviatán*, que contiene una nueva aproximación a la Teoría del Pensamiento.

En 1623, Wilhelm Schickard crea la primera calculadora mecánica. En 1642, Blaise Pascal crea la segunda calculadora mecánica.

En 1670, Gottfried Leibniz mejora las dos calculadoras anteriores agregándoles funcionalidades para la multiplicación y la división. Leibniz también inventa el sistema binario y tuvo la idea de un Alfabeto del Pensamiento Humano que pudiese ser mecanizado [3].

Durante los siglos XVIII y XIX surgen nuevos inventos que buscan mecanizar operaciones matemáticas. George Boole desarrolla un algebra binaria (Algebra

Booleana) para representar algunas *leyes del pensamiento*. Charles Babbage y Ada Lovelace trabajaron en *máquinas calculadoras mecánicas programables*.

A principios del siglo XX, Bertrand Russell y Alfred North Whitehead publican *Principia Probabilis*, revolucionando la Lógica Formal. Russell, Ludwig Wittgenstein, y Rudolf Carnap llevaron la filosofía al análisis lógico del conocimiento.

En 1923, el término Robot es usado por primera vez en inglés en una obra de teatro.

En 1931, Gödel demuestra que un sistema lógico formal consistente permite la formulación de teoremas verdaderos que han de derivarse de una serie de axiomas (verdades no demostrables). Para lograr esto, Gödel construyó un lenguaje de programación basado en números enteros, por lo que se le conoce como el padre teórico de la ciencia de la computación.

En 1941, Konrad Zuse construye los primeros computadores controlados por programas.

En 1943, Warren Sturgis McCulloch y Walter Pitts publican *Un Cálculo Lógico de las Ideas inmanentes en la Actividad Nerviosa*, proponiendo una teoría inicial para el desarrollo de Redes Neuronales [3].

Los primeros desarrollos en inteligencia artificial comenzaron a mediados de los años 1950 con el trabajo de Alan Turing.

El **Test de Turing**, expuesto en 1950 en un artículo (*Computing machinery and intelligence*) para la revista Mind, sigue siendo hoy día una de las cabezas de lanza de los defensores de la Inteligencia Artificial. Está fundamentado en la hipótesis positivista de que, si una máquina se comporta en todos los aspectos como inteligente, entonces debe ser inteligente.

La prueba consiste en un desafío. La máquina debe hacerse pasar por humana en una conversación con un hombre a través de una comunicación de texto estilo chat. Al sujeto no se le avisa si está hablando con una máquina o una persona. Si el sujeto es incapaz de determinar si la otra parte de la comunicación es humana o máquina, entonces se considera que la máquina ha alcanzado un determinado nivel de madurez: es inteligente. Todavía ninguna máquina puede pasar este examen en una experiencia con método científico.

Una de las aplicaciones de la prueba de Turing es el control de spam. Dado el gran volumen de correos electrónicos enviados, el spam es, por lo general, enviado automáticamente por una máquina. Así la prueba de Turing puede usarse

para distinguir si el correo electrónico fue enviado por un remitente humano o por una máquina [3, 11, 12].

Después de la presentación de la Prueba de Turing, varias instituciones académicas entre las que se destacan la Universidad de Manchester y el Instituto Tecnológico de Carnegie (ahora Universidad de Carnegie-Mellon), se dedican a investigar las ciencias de la computación buscando que las máquinas puedan pensar como lo hace un ser humano.

En 1956, durante la Conferencia de Darmouth, John McCarthy emplea por primera vez en la historia el término Inteligencia Artificial. En este congreso se hicieron previsiones triunfalistas a diez años que jamás se cumplieron.

A partir del grupo que postulo la Inteligencia Artificial en Darmouth, se formaron dos grandes “escuelas” de I.A.: Allen Newell y Herbert Simon lideraron el equipo de la Universidad de Carnegie-Mellon, proponiéndose desarrollar modelos de comportamiento humano con aparatos cuya estructura se pareciese lo más posible a la del cerebro (lo que posteriormente derivó en los trabajos sobre “redes neuronales” artificiales).

John McCarthy y Marvin Minsky formaron otro equipo en el Instituto Tecnológico de Massachusetts (MIT), centrándose más en que los productos del procesamiento tengan el carácter de inteligente, sin preocuparse por que el funcionamiento o la estructura de los componentes sean parecidas a los del ser humano.

Ambos enfoques sin embargo corresponden a los mismos objetivos prioritarios de la I.A.: “entender la inteligencia natural humana, y usar máquinas inteligentes para adquirir conocimientos y resolver problemas considerados como intelectualmente difíciles”.

La historia de la IA ha sido testigo de ciclos de éxito, injustificado optimismo y la consecuente desaparición de entusiasmo y apoyos financieros. También ha habido ciclos caracterizados por la introducción de nuevos y creativos enfoques y de un sistemático perfeccionamiento de los mejores.

Fue en los años 60 cuando Alan Newell y Herbert Simon, que trabajando la demostración de teoremas y el ajedrez por ordenador logran crear un programa llamado GPS (General Problem Solver: solucionador general de problemas). Este era un sistema en el que el usuario definía un entorno en función de una serie de objetos y los operadores que se podían aplicar sobre ellos. Este programa era capaz de trabajar con las torres de Hanoi, así como con criptoaritmética y otros problemas similares, operando, claro está, con microcosmos formalizados que representaban los parámetros dentro de los cuales se podían resolver problemas. Lo que no podía hacer el GPS era resolver problemas ni del mundo real, ni médicos ni tomar decisiones importantes. El GPS manejaba reglas heurísticas que

la conducían hasta el destino deseado mediante el método del ensayo y el error. También aprendía de sus propios descubrimientos.

En los años 70, un equipo de investigadores dirigido por Feigenbaum comenzó a elaborar un proyecto para resolver problemas de la vida cotidiana o que se centrara, al menos, en problemas más concretos. Así es como nació el sistema experto.

El primer sistema experto fue el denominado DENDRAL, un intérprete de espectrograma de masa construido en 1967, pero el más influyente resultaría ser el MYCIN de 1974. MYCIN era capaz de diagnosticar trastornos en la sangre y recetar la correspondiente medicación, todo un logro en aquella época que incluso fueron utilizados en hospitales (como el Puff, variante de MYCIN de uso común en el Pacific Medical Center de San Francisco, EEUU)

Ya en los años 80, se desarrollaron lenguajes de programación especiales para utilizar con la Inteligencia Artificial, tales como el LISP o el PROLOG. En esta época se desarrollan sistemas expertos más refinados, como por ejemplo el EURISKO. Este programa perfecciona su propio cuerpo de reglas heurísticas automáticamente, por inducción [3, 14, 15].

A principios del nuevo milenio fue postulada la primera *Teoría Matemática de Inteligencia Artificial Emergente*, que combina la teoría de las ciencias de la computación con la teoría de probabilidades. Todo esto para derivar en el comportamiento óptimo de agentes racionales computarizados. Esta es una señal del avance de la IA y su formación como una ciencia formal.

Se ha diseñado algoritmos que permiten a las máquinas un aprendizaje programado y secuencial. Sin embargo, aún no existe el hardware necesario para implementarlos. Uno de los grandes expertos de la Inteligencia Artificial, Raymond Kurzweil, espera que en el año 2029 ya existan computadores capaces de sobrepasar la capacidad de computación del cerebro humano.

4.2.2 Fundamentos de la inteligencia artificial. La Inteligencia Artificial es una ciencia relativamente nueva que aborda tantas cuestiones confundibles en un nivel fundamental y conceptual que, adjunto a lo científico, es necesario hacer consideraciones desde el punto de vista de la filosofía. Por ejemplo:

- ¿En qué consiste la inteligencia? ¿Cómo la reconoceríamos en un objeto no humano, si la tuviera?
- ¿Qué sustancia y organización se requiere? ¿Es algo estrictamente biológico u orgánico? ¿Es posible que una criatura hecha de metal, por ejemplo, posea una inteligencia comparable a la humana?

- Aunque una criatura no orgánica pudiera solucionar problemas de la misma manera que un humano, ¿tendría o podría tener consciencia y emociones?
- Suponiendo que es posible hacer robots con una inteligencia comparable a la humana, ¿se deberían hacer?

Durante más de 2000 años de tradición en filosofía, han ido surgiendo diversas teorías del razonamiento y del aprendizaje, simultáneamente con el punto de vista de que la mente se reduce al funcionamiento físico. La Psicología ofrece herramientas que permiten la investigación de la mente humana, así como un lenguaje científico para expresar las teorías que se van obteniendo. La lingüística ofrece teorías para la estructura y significado del lenguaje, así como la Ciencia de la Computación, de la que se toman las herramientas matemáticas y tecnológicas que permiten que la Inteligencia Artificial sea una realidad.

Los filósofos clásicos delimitaron importantes ideas relacionadas con la IA, pero para pasar de allí a una ciencia formal era necesario contar con una formalización matemática en tres áreas principales: la computación, la lógica y la probabilidad. La idea de expresar un cálculo mediante un algoritmo formal se remota a la época del Jwarizmi, matemático árabe del siglo IX, con cuyas obras se introdujeron en Europa los números arábigos y el álgebra.

El hombre se ha aplicado a sí mismo el nombre científico de Homo Sapiens como una valoración de la trascendencia de nuestras habilidades mentales tanto para nuestra vida cotidiana como para nuestro propio sentido de identidad. Los esfuerzos del campo de la IA se enfocan en lograr la comprensión de entidades inteligentes. Una de las razones de su estudio es el aprender más de nosotros mismos. A diferencia de la Filosofía y de la Psicología, que también se ocupan de la inteligencia, los esfuerzos de la IA están encaminados tanto a la construcción de entidades como a su comprensión. Nadie podría pronosticar con toda precisión lo que se podría esperar en el futuro, pero es evidente que las computadoras que posean una inteligencia a nivel humano tendrán repercusiones muy importantes en la vida diaria así como el devenir de la civilización.

El problema que aborda la IA es uno de los más complejos: ¿Cómo es posible que un diminuto y lento cerebro, sea biológico o electrónico, tenga capacidad de percibir, comprender, predecir y manipular un mundo que en tamaño y complejidad lo excede con creces? A diferencia de la investigación en torno al desplazamiento mayor que la velocidad de la luz o de un dispositivo antigravitatorio, el investigador del campo de la IA cuenta con pruebas contundentes (desde las Matemáticas y la Computación) de que tal búsqueda es factible.

La IA permite al hombre emular en las máquinas el comportamiento humano, tomando como base el cerebro y su funcionamiento, de manera tal que se pueda alcanzar cierto razonamiento [3].

4.2.3 Ramas de la inteligencia artificial. No se puede decir que hay una lista generalmente aceptada de ramas de la IA. Esta es una ciencia en constante evolución y cada cierto tiempo aparece un nuevo campo de estudio, de modo que nunca hay un número fijo de divisiones de la IA. Además es común que dos o varias ramas conocidas interactúen para solucionar problemas específicos que no podrían resolver cada una de ellas por separado.

Sin embargo, hay dos escuelas o corrientes de pensamiento dentro de la IA. Cada una de estas escuelas tiene a su vez, sus propias ramas. Las dos corrientes dentro de la IA son la *Inteligencia Artificial Convencional* y la *Inteligencia Artificial Computacional* [3, 18].

- **IA Convencional:** Estudia métodos que actualmente se conocen como máquinas de aprendizaje. Por un lado, se caracteriza por el formalismo matemático. Un sistema matemático formal o un sistema axiomático es un artificio matemático compuesto de símbolos que se unen entre sí formando cadenas que a su vez pueden ser manipuladas según reglas para producir otras cadenas. De esta manera, el sistema formal es capaz de representar cierto aspecto de la realidad. La IA Convencional también se caracteriza por el análisis estadístico. Se busca que los sistemas logren predecir y tomar decisiones mediante extraer una conclusión en el efecto que algunos cambios en los valores de variables independientes tienen sobre una respuesta o variables dependientes [3, 18].
- **IA Computacional:** La IA Computacional implica desarrollo o aprendizaje iterativo. Se centra en el estudio de mecanismos adaptativos para permitir el comportamiento inteligente de sistemas complejos y cambiantes. El aprendizaje se realiza basándose en datos empíricos.

A su vez, estas dos escuelas tienen diferentes ramas de la IA.

4.3 LOS SISTEMAS EXPERTOS

Los Sistemas Expertos son una de las ramas de la Inteligencia Artificial Convencional. Un Sistema Experto (SE) es un programa informático que contiene el conocimiento temático de uno o más Expertos (personas). Un **Experto** en un área o problema es un individuo con un conocimiento especializado sobre dicho problema. A este tipo de conocimiento se le llama **Dominio**.

Los SE ejecutan tareas que normalmente deberían ser ejecutadas por un experto. Un SE busca razonar de la misma forma que lo hace un experto, para llegar a las mismas conclusiones a las que él llegaría partiendo de una serie de datos (hechos). La forma más común de un SE es un programa hecho con un grupo de reglas que analizan una serie de datos (usualmente suministrados por el usuario final del sistema) sobre un tipo específico de problemas, proveyendo información resultante que puede incluir diagnósticos, datos matemáticos y recomendaciones finales para el usuario.

Para que un SE sea herramienta efectiva, los usuarios deben interactuar de una forma fácil, reuniendo dos capacidades para poder cumplirlo:

- **Explicación de razonamientos:** los SE se deben desarrollar siguiendo ciertas reglas o pasos comprensibles de manera que se pueda generar la explicación para cada una de estas reglas o pasos, que a la vez se basan en hechos.
- **Adquisición de nuevos conocimientos:** son mecanismos de razonamiento que sirven para modificar los conocimientos anteriores. Sobre la base de lo anterior se puede decir que los sistemas expertos son el producto de investigaciones en el campo de la inteligencia artificial ya que esta no intenta sustituir a los expertos humanos, sino que se desea ayudarlos a realizar con más rapidez y eficacia todas las tareas que realiza con menor dificultad.

Actualmente se están mezclando diferentes técnicas o aplicaciones que permiten aprovechar las ventajas que cada una de estas ofrece para maximizar los beneficios que las empresas hacen en otros conocimientos y tecnologías que pueden servir como base para la construcción de SE. Es importante tener en cuenta la preponderancia cada vez mayor de los agentes en línea, basados en web, que cada vez toman un papel más preponderante en la adopción de los SE [13, 21, 22].

Es bastante común que se le dé igual significado a los conceptos Sistema Basado en el Conocimiento (SBC) y Sistema Experto (SE). Sin embargo, diversos autores dan diferentes significados, haciendo énfasis en el origen del conocimiento que sirve al sistema [20]. Para Gutiérrez [19] los SE son en realidad un subconjunto de los SBC. La razón principal para esto es que los SBC son todos aquellos sistemas donde esté implicada una manipulación o uso del conocimiento, cualquiera que sea su origen; en cambio, los SE usan conocimiento proveniente única y exclusivamente proveniente de un experto. Todo SE es un SBC pero no todo SBC es un SE. Un SE es entonces un SBC que es capaz de ejecutar una tarea que normalmente requiere la experticia de un humano.

4.3.1 Historia de los sistemas expertos. Los primeros SE conocidos surgieron en la década de 1960. Durante esta década los investigadores Alan Newell y

Herbert Simon desarrollaron un programa llamado GPS (General Problem Solver; solucionador general de problemas). Podía trabajar con criptoaritmética, con las torres de Hanoi y con otros problemas similares. Lo que no podía hacer el GPS era resolver problemas del mundo real, tales como un diagnóstico médico.

Algunos investigadores decidieron entonces cambiar por completo el enfoque del problema restringiendo su ambición a un dominio específico e intentando simular el razonamiento de un experto humano. En vez de dedicarse a computarizar la inteligencia general, se centraron en dominios de conocimiento muy concretos. De esta manera nacieron los SE.

A partir de 1965, un equipo dirigido por Edward Feigenbaum, comenzó a desarrollar SE utilizando bases de conocimiento definidas minuciosamente. Dos años más tarde se construye DENDRAL, el cual es considerado como el primer SE. La función de dicho SE era identificar estructuras químicas moleculares a partir de su análisis espectrográfico.

En la década de 1970 se desarrolló MYCIN para consulta y diagnóstico de infecciones de la sangre. Este sistema introdujo nuevas características: utilización de conocimiento impreciso para razonar y posibilidad de explicar el proceso de razonamiento. Lo más importante es que funcionaba de manera correcta, dando conclusiones análogas a las que un ser humano daría tras largos años de experiencia.

En MYCIN aparecen claramente diferenciados motor de inferencia (basado en reglas de razonamiento) y base de conocimientos (basado en hechos y datos). Al separar esas dos partes, se puede estudiar el motor de inferencias aisladamente. Esto da como resultado un sistema vacío o shell (concha). Así surgió EMYCIN (MYCIN Esencial) con el que se construyó SACON, utilizado para estructuras de ingeniería, PUFF para estudiar la función pulmonar y GUIDON para elegir tratamientos terapéuticos.

En esa época se desarrollaron también: HERSAY, que intentaba identificar la palabra hablada, y PROSPECTOR, utilizado para hallar yacimientos de minerales. De este último derivó el shell KAS (Knowledge Acquisition System).

En la década de 1980 hay un *boom* de los SE. Numerosas empresas de alta tecnología investigan en este área de la Inteligencia Artificial, desarrollando SE para su comercialización. Se llega a la conclusión de que el éxito de un SE depende casi exclusivamente de la calidad de su base de conocimiento. El inconveniente es que codificar la pericia de un experto humano puede resultar difícil, largo y laborioso.

Un ejemplo de SE moderno es CASHVALUE, que evalúa proyectos de inversión y VATIA, que asesora acerca del impuesto sobre el valor añadido o IVA [23].

4.3.2 Estructura básica de un sistema experto.

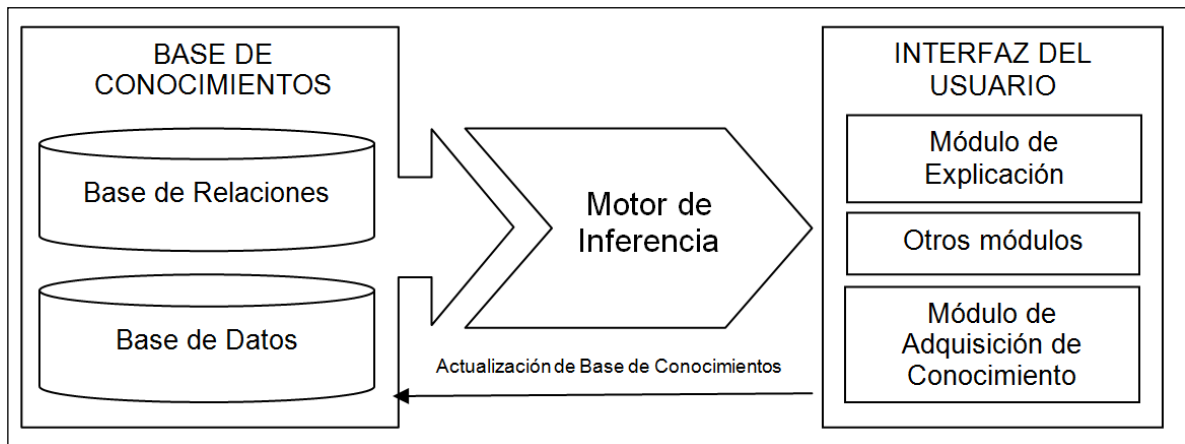


Ilustración 2. Estructura básica de un Sistema Experto.

▪ Base de Conocimientos

Se divide en dos componentes:

- **Base de Relaciones:** Es la parte del sistema experto que contiene el conocimiento sobre el dominio. Hay que obtener el conocimiento del experto y codificarlo en la base de relaciones. Una forma clásica de representar el conocimiento en un SE son las reglas, aunque se puede usar otros tipos de representación matemática.
- **Base de Datos:** Contiene los hechos sobre un problema que se han descubierto antes y durante una consulta. Durante una consulta con el SE, el usuario introduce la información del problema actual en la base de hechos. El sistema empareja esta información con el conocimiento disponible en la base de conocimientos para deducir nuevos hechos.
- **Motor de Inferencia.** El SE modela el proceso de razonamiento humano con un módulo conocido como el Motor de inferencia. Simula el razonamiento del experto en el dominio, y tiene como objetivo el generar nueva información con base en el conocimiento que se le provee.

Está formado por los algoritmos o programas que reflejan algún tipo de inferencia, manejan (seleccionan, deciden, interpretan y aplican) los conocimientos de la Base de conocimientos y coordinan las acciones que el sistema, como un todo, debe realizar.

Trabaja con la información contenida en la Base de relaciones y la Base de datos para deducir nuevos hechos. Su razonamiento puede ser por Encadenamiento hacia adelante, o Encadenamiento hacia atrás.

- **Módulo de Explicación.** Una característica fundamental de los SE es su habilidad para explicar su razonamiento. Usando el Módulo de Explicación, un SE puede proporcionar una explicación al usuario de por qué está haciendo una pregunta y cómo ha llegado a una conclusión. Este módulo proporciona beneficios tanto al diseñador del sistema como al usuario. El diseñador puede usarlo para detectar errores y el usuario se beneficia de la transparencia del sistema.
- **Módulo de Adquisición de Conocimiento.** Permite al Experto o al Ingeniero del Conocimiento actualizar el conocimiento de la Base de Conocimientos. Por este medio se ingresan los Hechos (datos) y las Relaciones al sistema. Facilita probar y depurar los cambios realizados.
- **Interfaz de Usuario.** Es donde se da la interacción entre un SE y su usuario. Se realiza en lenguaje natural, con el apoyo de facilidades modernas para interacción con computadoras (teclado, ratón, pantalla táctil, etc). Lo importante es que sea altamente interactiva, facilitando la “conversación” con seres humanos. Un requerimiento básico del interfaz es la habilidad de hacer preguntas y recibir respuestas, para poder finalmente desplegar una conclusión o resultado al usuario. [21, 23].

4.3.3 Ventajas y desventajas. Un SE es una ventaja importante cuando no se cuenta permanentemente con el apoyo de un experto humano. Por ejemplo en Medicina los SE tienen gran relevancia pues permiten dar atención a pacientes en zonas remotas o sin personal médico, lo que permite prestar atención inmediata en algunas situaciones.

Un SE puede explicar claramente cuál fue su razonamiento para llegar a una conclusión basado en los hechos (o datos) que ingresa el usuario y en general el contenido de la Base de Conocimientos (BC). Además es posible modificar la información contenida en la BC, lo cual permite agregar nuevo conocimiento del experto. Se modifica el comportamiento del sistema pero sin modificar el código fuente de la aplicación. El experto humano puede dejar la organización, pero su conocimiento permanecerá de todas formas en el SE.

Los SE pueden ser usados para acceder a una base de datos de manera inteligente [20] mediante técnicas de Minería de datos (MD) y Descubrimiento de conocimiento en bases de datos (KDD). Un SE puede procesar gran cantidad de datos, lo cual le facilita agregar al razonamiento nuevos hechos que hagan más sofisticado el resultado final.

Por otro lado, es posible que un SE tenga fallas en la definición del conocimiento del experto. Esto se puede dar debido a las dificultades que implica la naturaleza muchas veces abstracta y subjetiva del conocimiento que debe ser codificado en la BC. La administración de los cambios en la BC también puede ser un reto a nivel tecnológico y organizacional.

Aunque los SE se distinguen de las demás divisiones de la Inteligencia Artificial por proveer soluciones prácticas a distintas áreas, su aplicabilidad aún es limitada. Para un SE nada es obvio y si no se programa teniendo en cuenta absolutamente todo el razonamiento de un experto y hasta situaciones del día a día, lo más probable es que pueda llegar a conclusiones erradas. Por ejemplo, un investigador utilizó un SE que sirve para diagnosticar enfermedades de la piel y logró que el sistema diagnosticara sarampión a un carro. Falsedades que serían fácilmente identificables por seres humanos pueden ser aceptados como verdades por un SE.

En muchos casos aún no es posible reemplazar a un experto por un SE para resolver todos los problemas, pero un SE sí puede ser un apoyo importante para la toma de decisiones en diversos aspectos de la vida cotidiana.

4.3.4 Aplicabilidad. Un problema que se puede solucionar con un SE es aquel que no tiene una solución algorítmica clara y eficiente, estos se llaman **Problemas de estructura nociva**. Estos se presentan en situaciones donde solo el razonamiento puede dar un resultado dentro de muchos posibles resultados, dada una situación [20].

Tareas como hacer una operación matemática compleja pueden ser resueltas más fácilmente por sistemas tradicionales que por SE. En cambio, los SE son más pertinentes cuando se trata de sistematizar el conocimiento de un experto para generar diagnósticos donde es muy difícil definir una “solución correcta”.

Los SE tienen más valor en organizaciones donde hay un alto nivel de conocimientos y experiencia know-how que no puede ser fácilmente transmitida a otras personas. Son diseñados para recoger la experiencia, conocimientos e inteligencia de expertos, y hacer estos elementos accesibles para otros miembros de la organización a la hora de solucionar problemas.

Uno de los SE más famosos es XCON, para la configuración de sistemas de cómputo en la Digital Equipment Corporation (DEC). La configuración requiere que se suministren partes específicas de hardware, software y documentación. Este es un procedimiento complejo para sistemas de cómputo de especificaciones específicas requeridos por grandes corporaciones. XCON realiza una configuración en dos minutos, quince veces más rápido que un ser humano. Su

eficiencia es de 98%, contra un 70% de un experto humano. Esto ha resultado en que XCON le genera a DEC un ahorro de millones de dólares al año.

Los SE son diseñados y creados para facilitar tareas principalmente en Auditoría, Contabilidad, Medicina, Control de Procesos, Servicios Financieros, Recursos Humanos, Química, Electrónica, Ingeniería, entre otros.

Tabla 1. Aplicaciones de los Sistemas Expertos

Aplicación	Descripción
Configuración	Ensamblaje correcto de los componentes específicos de un sistema.
Diagnóstico	Identificación de situaciones problemáticas específicas, con base en la evidencia.
Instrucción	Enseñanza inteligente basada en preguntas, como lo haría un maestro humano.
Interpretación	Explicación de una situación observada.
Supervisión	Comparación de situaciones esperadas contra situaciones obtenidas reales.
Planeación	Diseño de procedimientos para obtener un resultado específico.
Pronóstico	Predicción del resultado futuro de una situación presente.
Remedio	Prescripción del tratamiento a un problema.
Control	Regulación de un proceso.

La tabla anterior describe algunos de los usos más comunes que se ha dado a los SE.

4.3.5 Sistemas expertos con lógica difusa. Aunque un SE no necesariamente requiere la Lógica Difusa para el diseño e implementación de este proyecto de grado, se propone simular el pensamiento del experto en Selección de Personal mediante Lógica Difusa. Con esto se busca aproximarse a la forma de pensar del experto, cuyo razonamiento no necesariamente se basa en valores de *verdadero* y *falso* típicos de la Lógica Clásica, sino que tiene un manejo amplio de ambigüedades e incertidumbres propias de la psique humana.

- Definición de Lógica Difusa. La Lógica como área del conocimiento, se puede definir como la ciencia que estudia los principios formales del razonamiento. En ese sentido la Lógica Difusa (LD) se define como el área que estudia los principios formales del razonamiento basado en aproximaciones empíricas, donde el razonamiento basado en certezas (verdadero o falso) es considerado

como una limitación [45]. La LD es una rama de la Inteligencia Artificial Computacional.

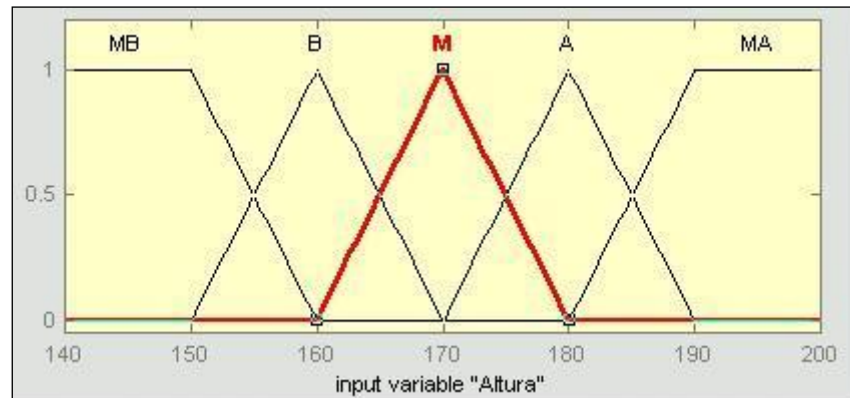
La LD surgió en 1965 en la Universidad de California en Berkeley gracias a los estudios del doctor Lotfi A. Zadeh. La LD ha encontrado una amplia aplicabilidad en sistemas de control electrónico, y ha sido bastante útil en Sistemas Expertos y muchas otras aplicaciones de Inteligencia Artificial.

En la actualidad la LD se encuentra en una dinámica evolución dado su éxito en aplicaciones informáticas donde los datos de entrada y las inferencias realizadas por el sistema tienen un alto nivel de ambigüedad, algo muy propio del razonamiento natural del ser humano [43]. A lo anterior se suma la evolución y democratización masiva de las tecnologías informáticas, una de las grandes revoluciones humanas.

▪ Teoría de Conjuntos Difusos

La LD se basa en la Teoría de Conjuntos Difusos (Fuzzy Sets), cuya primera aproximación fue hecha por Zadeh en 1965 [46]. Para cada conjunto difuso, existe asociada una **función de pertenencia** para sus elementos, que indican en qué medida el elemento forma parte de ese conjunto difuso. La pertenencia se mide en una escala que varía entre 0 y 1.

Ilustración 3. Ejemplo de pertenencia a un conjunto difuso



En la figura anterior se ilustra el concepto de pertenencia a un conjunto difuso. Para la variable "altura" hay cinco conjuntos difusos, cada uno de ellos con una función de pertenencia. Estos conjuntos son "muy bajo (MB)", "bajo (B)", "mediano (M)", "alto (A)", y "muy alto (MA)". Se puede decir que estas funciones determinan que una persona que mida 167 cm, tiene una pertenencia de 0.75 en el conjunto B y una pertenencia de 0.25 en el conjunto M.

Los Conjuntos Difusos soportan operaciones que se dan en la Lógica Clásica de Conjuntos. Estas operaciones pueden ser:

- **Complemento:** la función de “no pertenencia” de un elemento en un conjunto.
- **Unión:** la función que determina la pertenencia del elemento a uno u otro conjunto. La fórmula matemática para la Unión puede variar dependiendo del significado de la pertenencia del elemento a los conjuntos de la Unión.
- **Intersección:** la función que determina la pertenencia del elemento a uno y otro conjunto al mismo tiempo. La fórmula matemática para la Intersección puede variar dependiendo del significado de la pertenencia del elemento a los conjuntos de la Intersección.

Las operaciones de Conjuntos Difusos no se restringen necesariamente a estas tres opciones, aunque son las más comunes.

- **Conjuntos Difusos valorados por intervalos de confianza.** Se llama intervalo de confianza a un par de números entre los cuales se estima que estará cierto valor desconocido con una determinada probabilidad de acierto.

La incertidumbre puede ser representada mediante estos intervalos, de modo que el resultado de una función de pertenencia no de un solo número escalar sino que la pertenencia pueda estar entre dos puntos a_1 y b_1 .

Para mayor información se puede consultar el trabajo de Merigó⁵ [49].

4.3.6 Ingeniería del conocimiento. La Ingeniería del Conocimiento (IC) es una disciplina dentro de la Inteligencia Artificial y cuyo fin es la aplicación de una aproximación sistemática, disciplinada y cuantificable al desarrollo, funcionamiento y mantenimiento de Sistemas Expertos. En otras palabras, el objetivo último de la IC es el establecimiento de metodologías que permitan abordar el desarrollo de SE de una forma más sistemática [36].

Busca la transferencia y transformación del conocimiento y de la habilidad de entender y de resolver problemas desde alguna fuente de conocimiento a un programa. El SE intentará aproximarse en lo posible a la forma en la que dicho experto humano alcanza las soluciones [17].

Los Ingenieros del Conocimiento extraen el conocimiento de los expertos humanos en una determinada área, y codifican dicho conocimiento de manera que pueda ser procesado por un sistema. El Ingeniero del Conocimiento no es un

⁵ MERIGÓ, José. *Nuevas extensiones a los operadores OWA y su aplicación en los métodos de decisión*. Facultad de Economía y Empresa. Universidad de Barcelona. España. 2008.

experto en el campo que intenta modelar, mientras que el experto en el tema no tiene experiencia modelando su conocimiento (basado en la heurística) de forma que pueda ser representado de forma genérica en un sistema.

La Ingeniería del Conocimiento engloba a los científicos, tecnología y metodología necesarios para procesar el conocimiento. Puede basarse en metodologías que permitan obtener una mejor calidad en el conocimiento extraído y modelado. En este trabajo de grado se empleó la metodología CommonKADS, que se detallará en la Metodología de Trabajo.

4.4 PARADIGMAS Y LENGUAJES DE PROGRAMACIÓN

Un paradigma de programación es una forma o filosofía para la construcción de un software. No se puede decir que un paradigma es mejor que otro, dado que cada uno de ellos tiene sus ventajas y desventajas. La elección del paradigma depende del software a ser desarrollado, e incluso la mayoría de las veces se elige más de un paradigma para el desarrollo de una aplicación [24].

A continuación son presentados los paradigmas que fueron considerados para el desarrollo del software objeto de este proyecto.

4.4.1 Programación declarativa. La Programación Declarativa es un paradigma de programación que está basado en el desarrollo de programas especificando o "declarando" un conjunto de condiciones, proposiciones, afirmaciones, restricciones, ecuaciones o transformaciones que describen el problema y detallan su solución. La solución es obtenida mediante mecanismos internos de control, sin especificar exactamente cómo encontrarla (tan sólo se le indica a la computadora que es lo que se desea obtener o que es lo que se está buscando).

Un paradigma de programación declarativo está basado en el desarrollo de programas especificando o "declarando" un conjunto de condiciones, proposiciones, afirmaciones, restricciones, ecuaciones o transformaciones que describen el problema y detallan su solución. La solución es obtenida mediante mecanismos internos de control, sin especificar exactamente cómo encontrarla.

La programación declarativa es muy poderosa a la hora de verificar un programa, es decir, demostrar matemáticamente que cumple la especificación determinada por el programador [28].

La programación declarativa cuenta entre sus ramas más importantes la Programación Lógica y la Programación Funcional. La programación lógica se usa principalmente cuando se busca aplicar un conocimiento en forma de reglas

lógicas a la solución del problema. La programación funcional se basa en el concepto de función matemática para modelar la solución [25].

4.4.2 Programación imperativa. La programación imperativa describe la programación en términos del estado del programa y sentencias que cambian dicho estado. Los programas imperativos son un conjunto de instrucciones que le indican al computador cómo realizar una tarea.

La implementación de hardware de la mayoría de computadores es imperativa; prácticamente todo el hardware de los computadores está diseñado para ejecutar código de máquina, que es nativo al computador, escrito en una forma imperativa. Esto se debe a que el hardware de los computadores implementa el paradigma de las Máquinas de Turing. Desde esta perspectiva de bajo nivel, el estilo del programa está definido por los contenidos de la memoria, y las sentencias son instrucciones en el lenguaje de máquina nativo del computador (por ejemplo el lenguaje ensamblador).

Los primeros lenguajes imperativos fueron los lenguajes de máquina de los computadores originales. En estos lenguajes, las instrucciones fueron muy simples, lo cual hizo la implementación de hardware fácil, pero obstruyendo la creación de programas complejos. Fortran, cuyo desarrollo fue iniciado en 1954 por John Backus en IBM, fue el primer gran lenguaje de programación en superar los obstáculos presentados por el código de máquina en la creación de programas complejos [29].

En la programación imperativa se describe paso a paso un conjunto de instrucciones que deben ejecutarse para variar el estado del programa y hallar la solución, es decir, un algoritmo en el que se describen los pasos necesarios para solucionar el problema. A su vez, en la programación declarativa las sentencias que se utilizan lo que hacen es describir el problema que se quiere solucionar, pero no las instrucciones necesarias para solucionarlo. Esto último se realizará mediante mecanismos internos de inferencia de información a partir de la descripción realizada [26].

4.4.3 Computación en la nube. La Computación en la Nube, del inglés "Cloud Computing", es un paradigma que permite ofrecer servicios de computación a través de Internet. La "nube" es una metáfora de Internet. Es un paradigma de computación que surge en la primera década del siglo XXI donde los datos y los servicios residen en centros de datos masivamente escalables y se puede acceder desde cualquier dispositivo conectado a través de Internet.

En este tipo de computación todo lo que puede ofrecer un sistema informático se ofrece como **servicio**, de modo que los usuarios puedan acceder a los servicios

disponibles en Internet sin conocimientos, o por lo menos sin ser expertos en la gestión de los recursos que usan.

Según el IEEE Computer Society, la Computación en la Nube es un paradigma en el que la información se almacena de manera permanente en servidores de Internet y se envía a cachés temporales de cliente, lo que incluye equipos de escritorio, centros de ocio, portátiles, etc. Esto se debe a que, pese a que las capacidades de los PC han mejorado sustancialmente, gran parte de su potencia se desaprovecha, al ser máquinas de propósito general.

El cambio paradigmático que ofrece la Computación en la Nube es que permite aumentar el número de servicios basados en la red. Esto genera beneficios tanto para los proveedores, que pueden ofrecer, de forma más rápida y eficiente, un mayor número de servicios, como para los usuarios que tienen la posibilidad de acceder a ellos, disfrutando de la 'transparencia' e inmediatez del sistema y de un modelo de pago por consumo

La Computación en la Nube consigue aportar estas ventajas, apoyándose sobre una infraestructura tecnológica dinámica que se caracteriza, entre otros factores, por un alto grado de automatización, una rápida movilización de los recursos, una elevada capacidad de adaptación para atender a una demanda variable, así como virtualización avanzada y un precio flexible en función del consumo realizado. "Cloud Computing" evoluciona hacia un nuevo modelo de prestación de servicios de negocio y tecnología, que permite al usuario acceder a un catálogo de servicios estandarizados y responder a las necesidades del negocio, de forma flexible y adaptativa, en caso de demandas no previsibles o de picos de trabajo, pagando únicamente por el consumo efectuado.

La computación en nube es un concepto que incorpora el software como servicio, como en la Web 2.0 y otros conceptos recientes, también conocidos como tendencias tecnológicas, que tienen en común el que confían en Internet para satisfacer las necesidades de cómputo de los usuarios.

El concepto de la computación en la nube empezó en proveedores de servicio de Internet a gran escala, como Google, Amazon AWS y otros que construyeron su propia infraestructura. De entre todos ellos emergió una arquitectura: un sistema de recursos distribuidos horizontalmente, introducidos como servicios virtuales de TI escalados masivamente y manejados como recursos configurados y mancomunados de manera continua. Este modelo de arquitectura fue inmortalizado por George Gilder en su artículo de octubre 2006 en la revista Wired titulado Las fábricas de información. Las granjas de servidores, sobre las que escribió Gilder, eran similares en su arquitectura al procesamiento "grid" (red, parrilla), pero mientras que las redes se utilizan para aplicaciones de procesamiento técnico débilmente acoplados (loosely coupled, un sistema compuesto de subsistemas con cierta autonomía de acción, que mantienen una

interrelación continua entre ellos), este nuevo modelo de nube se estaba aplicando a los servicios de Internet.

- **Ventajas y desventajas.** A continuación se enumeran algunos de los beneficios e inconvenientes de la Computación en la Nube.

Tabla 2. Ventajas y desventajas de la Computación en la Nube

Ventajas	Desventajas
<ul style="list-style-type: none"> • Integración de servicios en Red. Por su naturaleza, la tecnología de "Cloud Computing" se puede integrar con mucha mayor facilidad y rapidez con el resto de las aplicaciones empresariales (tanto software tradicional como Cloud Computing basado en infraestructuras), ya sean desarrolladas de manera interna o externa. • Prestación de servicios a nivel mundial. Las infraestructuras de "Cloud Computing" proporcionan mayor capacidad de adaptación, recuperación de desastres completa y reducción al mínimo de los tiempos de inactividad. Una infraestructura 100% de "Cloud Computing" no necesita instalar ningún tipo de hardware. • Implementación más rápida y con menos riesgos. La belleza de la tecnología de "Cloud Computing" es su simplicidad, y el hecho de que requiera mucha menor inversión para empezar a trabajar. Se puede empezar a desarrollar proyectos muy rápidamente gracias a una infraestructura de "Cloud Computing". 	<ul style="list-style-type: none"> • Dependencia de la disponibilidad y acceso a Internet. La centralización de las aplicaciones y el almacenamiento de los datos originan una interdependencia de los proveedores de servicios. La disponibilidad de las aplicaciones depende de la disponibilidad de acceso a internet. • Seguridad. Los datos "sensibles" del negocio no residen en las instalaciones de las empresas por lo que podría generar un contexto de alta vulnerabilidad para la sustracción o robo de información. La información de la empresa debe recorrer diferentes nodos para llegar a su destino, cada uno de ellos (y sus canales) son un foco de inseguridad. Si se utilizan protocolos seguros, HTTPS por ejemplo, la velocidad total disminuye debido a la sobrecarga que requieren estos protocolos. • Proveedores y escalabilidad a largo plazo. La confiabilidad de los servicios depende de la viabilidad tecnológica y financiera de los proveedores de servicios en nube. Empresas emergentes o alianzas entre empresas podrían crear un

Ventajas	Desventajas
<ul style="list-style-type: none"> • Actualizaciones automáticas que no afectan negativamente a los recursos de TI. Si se actualiza una aplicación a su última versión, en la computación tradicional es necesario dedicar tiempo y recursos a volver a crear personalizaciones e integraciones. La tecnología de "Cloud Computing" no obliga a decidir entre actualizar y conservar las personalizaciones, porque esas personalizaciones e integraciones se conservan automáticamente durante la actualización. 	<p>ambiente propicio para el monopolio en los servicios. A medida que más usuarios empiecen a compartir la infraestructura de la nube, la sobrecarga en los servidores de los proveedores aumentará, si la empresa no posee un esquema de crecimiento óptimo puede llevar a degradaciones en el servicio.</p>

Cabe mencionar que aunque la Computación en la Nube es un concepto relativamente joven en el mundo de la tecnología, sus ventajas están promoviendo que cada vez más empresas la adopten como un soporte a la estrategia gerencial.

- **Capas.** A continuación se enumeran las diferentes capas de prestación de servicios tecnológicos para la nube. Estas capas van desde la aplicación desplegada al usuario hasta la infraestructura de hardware requerida, todas ellas soportadas por Internet.

El **Software como servicio** (en inglés *Software as a Service*, SaaS) se encuentra en la capa más alta y caracteriza una aplicación completa ofrecida como un servicio, en en una sola instancia del software que corre en la infraestructura del proveedor y sirve a múltiples organizaciones de clientes. El ejemplo de SaaS conocido más ampliamente es *Salesforce.com*, pero ahora ya hay muchos más, incluyendo las *Google Apps* que ofrecen servicios básicos de negocio como el e-mail y la suite de oficina.

La capa del medio, que es la **Plataforma como servicio** (en inglés *Platform as a service*, PaaS), es el empaquetamiento de un ambiente de desarrollo y despliegue de soluciones, orientados específicamente a la prestación de servicios SaaS. Las ofertas de PaaS pueden soportar todas las fases del ciclo de desarrollo y pruebas del software, o pueden estar especializadas en cualquier área en particular, tal como la administración del contenido. Los ejemplos comerciales incluyen *Google Apps Engine*, que sirve aplicaciones de la infraestructura *Google*. Servicios PaaS tales como éstos permiten gran flexibilidad, pero pueden ser restringidos por las capacidades que están disponibles a través del proveedor. *Microsoft* apuesta por

Windows Azure, que es una plataforma de desarrollo en la nube que permite crear y ejecutar aplicaciones codificadas en varios lenguajes y tecnologías, como .NET, Java y PHP.

La **Infraestructura como servicio** (*Infrastructure as a Service, IaaS*) se encuentra en la capa inferior y es un medio de entregar almacenamiento básico y capacidades de cómputo estandarizados en la red. Servidores, sistemas de almacenamiento, conexiones, enrutadores, y otros sistemas se concentran (por ejemplo a través de la tecnología de virtualización) para manejar tipos específicos de cargas de trabajo. El ejemplo comercial mejor conocido es Amazon Web Services, cuyos servicios EC2 y S3 ofrecen cómputo y servicios de almacenamiento esenciales (respectivamente). Otro ejemplo es Joyent cuyo producto principal es una línea de servidores virtualizados, que proveen una infraestructura en-demanda altamente escalable para manejar sitios Web, incluyendo aplicaciones Web complejas escritas en Ruby on Rails, PHP, Python, y Java.

▪ Tipos de nube

Las **nubes públicas** se manejan por terceras partes, y los trabajos de muchos clientes diferentes pueden estar mezclados en los servidores, los sistemas de almacenamiento y otras infraestructuras de la nube. Los usuarios finales no conocen qué trabajos de otros clientes pueden estar corriendo en el mismo servidor, red, discos como los suyos propios.

Las **nubes privadas** son una buena opción para las compañías que necesitan alta protección de datos y ediciones a nivel de servicio. Las nubes privadas están en una infraestructura en-demanda manejada por un solo cliente que controla qué aplicaciones debe correr y dónde. Son propietarios del servidor, red, y disco y pueden decidir qué usuarios están autorizados a utilizar la infraestructura.

Las **nubes híbridas** combinan los modelos de nubes públicas y privadas. El cliente es propietario de unas partes y comparte otras, aunque de una manera controlada. Las nubes híbridas ofrecen la promesa del escalado aprovisionada externamente, en-demanda, pero añaden la complejidad de determinar cómo distribuir las aplicaciones a través de estos ambientes diferentes. Las empresas pueden sentir cierta atracción por la promesa de una nube híbrida, pero esta opción, al menos inicialmente, estará probablemente reservada a aplicaciones simples sin condicionantes, que no requieran de ninguna sincronización o necesiten bases de datos complejas.

4.4.4 Ambiente de desarrollo usado en el proyecto. El prototipo de SE fue implementado en un entorno que permite el despliegue de aplicaciones bajo el modelo SaaS. Este ambiente se llama “hosting compartido” que es básicamente

una nube pública donde las principales herramientas son Linux (sistema operativo), Apache (motor web), MySQL (motor de bases de datos) y PHP (lenguaje de programación). Este tipo de ambiente se le conoce popularmente como LAMP.

- **Linux.** Un sistema operativo es el programa o conjunto de programas que efectúan la gestión de los procesos básicos de un sistema informático, y permite la normal ejecución del resto de las operaciones.

GNU/Linux es uno de los términos empleados para referirse a la combinación del núcleo o kernel libre similar a Unix denominado Linux, que es usado con herramientas de sistema GNU (iniciativa orientada a desarrollar un sistema operativo con código fuente completamente abierto). Su desarrollo es uno de los ejemplos más prominentes de software libre; todo su código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la GPL (Licencia Pública General de GNU, en inglés: General Public License) y otra serie de licencias libres.

A pesar de que Linux (núcleo) es, en sentido estricto, el sistema operativo, parte fundamental de la interacción entre el núcleo y el usuario (o los programas de aplicación) se maneja usualmente con las herramientas del proyecto GNU o de otros proyectos como GNOME. Sin embargo, una parte significativa de la comunidad, así como muchos medios generales y especializados, prefieren utilizar el término Linux para referirse a la unión de ambos proyectos.

A las variantes de esta unión de programas y tecnologías, a las que se les adicionan diversos programas de aplicación de propósitos específicos o generales se las denomina distribuciones. Su objetivo consiste en ofrecer ediciones que cumplan con las necesidades de un determinado grupo de usuarios. Algunas de ellas son especialmente conocidas por su uso en servidores y supercomputadoras donde tiene la cuota más importante del mercado. Según un informe de IDC, GNU/Linux es utilizado por el 78% de los principales 500 servidores del mundo. Otro informe le da una cuota de mercado de 89% en los 500 mayores supercomputadores. Con menor cuota de mercado el sistema GNU/Linux también es usado en el segmento de las computadoras de escritorio, portátiles, computadoras de bolsillo, teléfonos móviles, sistemas embebidos, videoconsolas y otros dispositivos.

- **Apache.** Un servidor web o servidor HTTP es un programa que procesa cualquier aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando o cediendo una respuesta en cualquier lenguaje o Aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se utiliza el protocolo HTTP para estas

comunicaciones, perteneciente a la capa de aplicación del modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa.

Apache es un servidor web de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual. El servidor Apache se desarrolla dentro del proyecto HTTP Server de la Apache Software Foundation.

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en Internet: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años.

- **MySQL.** Una base de datos o banco de datos (BD) es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Los sistemas de gestión de bases de datos (en inglés database management system, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de los sistemas de gestión de bases de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización.

MySQL es un DBMS con más de seis millones de instalaciones. MySQL AB (subsidiaria de Sun Microsystems, que a su vez es subsidiaria de Oracle) desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

- **PHP.** PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente para la interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos

o en la creación de otros tipos de programas incluyendo aplicaciones de escritorio con interfaz gráfica.

La implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. El lenguaje PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, el número de sitios en PHP ha compartido algo de su preponderante sitio con otros nuevos lenguajes no tan poderosos desde agosto de 2005. Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Cuando el cliente (computadora del usuario) hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos.

Para agilizar el desarrollo de este proyecto se utilizó **Zend Framework** (ZF), un framework de código abierto para desarrollar aplicaciones web y servicios web con PHP 5. ZF es una implementación que usa código 100% orientado a objetos. La estructura de los componentes de ZF es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado.

Aunque se pueden utilizar de forma individual, los componentes de la biblioteca estándar de ZF conforman un potente y extensible framework de aplicaciones web al combinarse. ZF ofrece un gran rendimiento y una robusta implementación de arquitecturas basadas en el Modelo-Vista-Controlador (MVC), una abstracción de base de datos fácil de usar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores

puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos.

El principal patrocinador del proyecto ZF es Zend Technologies, pero muchas empresas han contribuido con componentes o características importantes para el marco. Empresas como Google, Microsoft y Strikelron se han asociado con Zend para proporcionar interfaces de servicios web y otras tecnologías que desean poner a disposición de los desarrolladores de ZF.

- **HTML y Javascript.** HTML, siglas de HyperText Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo Javascript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

Javascript se utiliza principalmente en su forma del lado del cliente (client-side) implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de Javascript del lado del servidor (Server-side Javascript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo. Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web.

Para sacar mayor provecho de Javascript para enriquecer la interfaz con el usuario, se ha usado **Dojo Toolkit** como framework para Javascript. Este framework contiene APIs y widgets (controles) para facilitar el desarrollo de aplicaciones Web que utilicen tecnología AJAX. Resuelve asuntos de usabilidad comunes y proporciona una gama amplia de opciones en una sola biblioteca JavaScript.

4.5 LA METODOLOGÍA COMMONKADS PARA INGENIERÍA DEL CONOCIMIENTO

CommonKADS es una metodología para la construcción de Sistemas Basados en el Conocimiento (SBC). Fue desarrollada en la Universidad de Ámsterdam en cooperación con varios socios europeos, como universidades, organizaciones de investigación, casas de software y de consultoría. Actualmente es considerada por muchas compañías y organizaciones alrededor del mundo como un estándar para la ingeniería del conocimiento y de los SBC.

Se busca observar el problema desde el punto de vista del proceso organizacional que se va a intervenir, de modo que el modelado de la solución como SBC se adapte de la mejor manera al proceso y las tareas involucradas.

CommonKADS está fundamentada en los siguientes principios:

- CommonKADS provee una plataforma para llevar a cabo el modelado del problema. Un proyecto de conocimiento en CommonKADS incluye la construcción de una serie de modelos que constituyen parte del producto entregado. Estos modelos reflejan diferentes puntos de vista del conocimiento inmerso en un problema y en su solución. Cada uno tiene un propósito específico, unos productos asociados y unas estrategias para su desarrollo. Es importante recordar que un modelo es una abstracción útil de alguna parte de la realidad que hace posible focalizar ciertos aspectos e ignorar otros.
- Se conceptualiza el sistema independientemente de su estructura a nivel de programación. Según Alan Newell (1982) “para que el conocimiento pueda ser modelado en un nivel conceptual debe ser independiente de las construcciones informáticas específicas y de la implantación del software”.
- Se modela el conocimiento de tal forma que su análisis permita la detección de tipos, patrones, roles y estructuras, logrando una visión sistémica donde hay usuarios y actores que juegan diferentes roles restrictivos y especializados en la solución de problemas.
- Desde un punto de vista administrativo, un proyecto CommonKADS es de aprendizaje basado en la experiencia, siguiendo un modelo de desarrollo en espiral. CommonKADS de esta forma favorece una gestión de proyectos ordenable, balanceada y que permite un aprendizaje estructurado. Los resultados permiten un monitoreo estructurado que favorece ampliamente la toma de decisiones.
- En CommonKADS se hace visible el aporte de diversos paradigmas avalados y reconocidos internacionalmente. Entre estos se cuentan el análisis y el diseño estructurado, la orientación por objetos, la teoría de las organizaciones, la reingeniería de procesos y la gerencia de la calidad. Se toman en cuenta tanto la experiencia como las estructuras de información existentes en la organización.
- CommonKADS añade una especificidad a lo que debe entenderse como Sistema Basado en el Conocimiento: es un modelo operacional que exhibe los comportamientos deseados que se han especificado u observado en el mundo real.

- Algunos autores definen que el desarrollo de un SBC, desde el punto de vista de CommonKADS, es el desarrollo estructurado de modelos de comportamiento de solución de problemas en su contexto organizacional y de aplicación específico. En estos modelos se incluyen tanto los conocimientos de los expertos como los de otros sistemas del entorno, tales como la organización, el usuario y la interacción entre éste y el sistema. Un SBC es una realización computacional asociada con estos modelos.
- CommonKADS también ofrece una serie de formularios que facilita la labor de construcción del sistema y que permite obtener las especificaciones y los requerimientos de un problema y su solución, desde el punto de vista de su relación con el resto de la organización, de los entes que participan en el problema y del conocimiento que se requiere para llegar al sistema final. Más adelante se explicarán los modelos. Estos formularios son su forma estructural.

4.5.1 Ciclo de vida. El ciclo de vida en CommonKADS tiene varias fases, cada una con unas tareas y productos (elicitación de requisitos, documentos de pruebas, etc.) asociados. Estas fases se pueden ser integradas dentro del modelo de Desarrollo en Espiral (ver Metodología de Trabajo), son listadas a continuación:

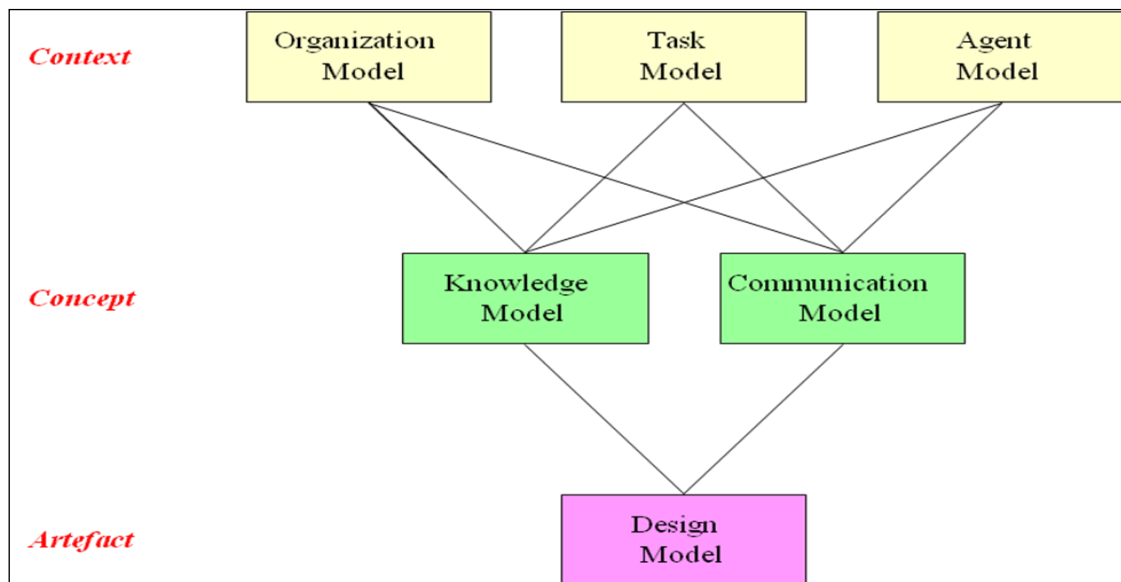
- **Análisis:** Se realiza para comprender el problema desde el punto de vista de la solución que se piensa desarrollar. Está formado por la especificación de los requerimientos externos del sistema basado en el conocimiento y por un análisis del problema específico. Los productos que se deben obtener en esta etapa son: un documento del proyecto, un documento de los requerimientos, un documento del modelo (modelo conceptual), un documento de viabilidad y un documento de apoyo.
- **Diseño:** En el cual se hace una descripción del comportamiento del sistema (descripción funcional) y una descripción física en la que se especifica detalladamente cada uno de sus componentes. De esta etapa debe salir toda la especificación modular del sistema y la descripción detallada de cómo debe ser, desde el punto de vista computarizado.
- **Implantación del sistema:** En esta etapa se considera tanto la integración del software desarrollado como su adaptación en la organización.
- **Instalación:** Consiste en la puesta en marcha del sistema con el fin de que comience a operar en la empresa, iniciándose su proceso productivo.
- **Uso:** Se plantean actividades relacionadas con el manejo mismo del sistema y de las salidas o resultados que éste proporciona.

- **Mantenimiento y refinamiento del conocimiento.**

4.5.2 Estructura de CommonKADS. La metodología CommonKADS abarca todo el ciclo de desarrollo de software de Sistemas Basados en el Conocimiento (SBC). Esto lo hace mediante un número de modelos interrelacionados que capturan los principales rasgos del sistema y de su entorno. Una descripción detallada de CommonKADS puede encontrarse en⁶ [40], a continuación solo se enunciarán aspectos generales.

CommonKADS define modelos que representan distintas perspectivas de los procesos organizacionales. El proceso de desarrollo del SBC consiste en recolectar información con base en un conjunto de *plantillas* para los modelos. CommonKADS define seis modelos enmarcados en tres niveles.

Ilustración 4. Modelos de CommonKADS.



En el nivel de **Contexto** se busca un mejor conocimiento de la organización y el entorno donde se va a implementar el sistema. Los objetivos principales son: Identificar problemas y oportunidades; Decidir entre distintas soluciones posibles dada la conveniencia de cada una de ellas; Mejorar el conocimiento relacionado

⁶ SCHREIBER, G.; AKKERMANS, H.; ANJEWIERDEN, A.; DE HOOG, R.; SHADBOLT, N.; VAN DE VELDE, W.; WIELINGA, B. *Knowledge Engineering and Management: The CommonKADS Methodology*. The MIT Press. 2000.

con las tareas que se llevan a cabo; y Proyectar los cambios que necesita la organización. Los modelos dentro de este nivel son:

- **Modelo de la Organización (OM):** es una herramienta para analizar la organización donde el SBC va a ser introducido, y pretende descubrir problemas y oportunidades.
- **Modelo de Tareas (TM):** describe a un nivel general las tareas que son realizadas o serán realizadas en el entorno organizativo en que se propone instalar el SBC y proporciona el marco para la distribución de tareas entre agentes.
- **Modelo de Agentes (AM):** un agente es un ejecutor de una tarea. Puede ser humano, software o cualquier otra entidad capaz de realizar una tarea. Este modelo describe las competencias, características, autoridad y restricciones para actuar de los agentes.

En el nivel de **Concepto** se detalla la estructura del conocimiento y el flujo de información necesario para solucionar un problema específico de la organización. Los modelos asociados son:

- **Modelo de Comunicaciones (CM):** detalla el intercambio de información entre los diferentes agentes involucrados en la ejecución de las tareas descritas en el modelo de tarea.
- **Modelo de Conocimiento (KM):** este es el corazón de la metodología CommonKADS y modela el conocimiento de resolución de problemas empleado por un agente para realizar una tarea. El modelo de la experiencia distingue entre el conocimiento de la aplicación y el conocimiento de resolución del problema. El conocimiento de la aplicación se divide en tres subniveles: nivel del dominio (conocimiento declarativo sobre el dominio), nivel de inferencia (una biblioteca de estructuras genéricas de inferencia) y nivel de tarea (orden de las inferencias).

En el nivel de **Artefacto** se describe la implementación tecnológica del sistema. Solo tiene un modelo asociado:

- **Modelo de Diseño (DM):** mientras que los otros cinco modelos tratan del análisis del SBC, este modelo se utiliza para describir la arquitectura y el diseño técnico del SBC como paso previo a su implementación. En general produce la especificación técnica en términos de arquitectura, plataforma de implementación, módulos de software, construcciones de representación, y mecanismos computacionales para la implementación del sistema.

4.5.3 Modelo de La Organización. El modelo de la organización incluye 5 plantillas que se describen a continuación.

- OM-1: Identificación de problemas y oportunidades orientados al conocimiento

Durante esta fase se adquiere un conocimiento de las características y divisiones generales de la organización, con el fin de entender las estructuras y procesos de la misma que servirán como elementos constituyentes del análisis de viabilidad del proyecto.

Este formulario tiene mucha utilidad cuando se quieren conocer los procesos de la organización, sus problemas, las personas o entidades que se ven involucradas en ellos, entre otras. En general, el resultado de aplicar OM-1 será el de obtener una visión global del estado actual de la organización. Pero, en caso que esto no sea necesario, se puede omitir o reducir su alcance, como por ejemplo en la situación en que ya se haya identificado claramente un proceso específico como problemático o como susceptible de mejoría.

- OM-2: Descripción de los aspectos de la organización que tienen impacto en o están afectados por el problema elegido.

Un análisis orientado a profundizar en el contexto organizacional donde se da el problema.

Tabla 3. OM-2: Descripción de los aspectos de la organización que tienen impacto en o están afectados por el problema elegido.

Estructura	¿Cómo está conformada la organización? Estructura de divisiones, jerarquías, unidades, etc.
Proceso	Se debe dar una visión del flujo de información y control del proceso al que se integrará en sistema, teniendo en cuenta la estructura organizacional definida. Se recomienda hacer uso de herramientas UML como Diagramas de Actividades. El Modelo OM-3 profundiza específicamente en la descripción del proceso.
Personas	Miembros de la organización involucrados (actores) en el proceso. Se deben incluir tomadores de decisiones, agentes externos, proveedores, usuarios o beneficiarios (clientes) del conocimiento. Estas personas no necesitan ser personas reales, sino que pueden ser roles funcionales jugados por personas en la organización. Se recomienda hacer uso de herramientas UML como Diagramas de Casos de Uso.

Recursos	Información sobre Infraestructura Tecnológica, Equipos, Material de Trabajo, y habilidades no del conocimiento (habilidades interpersonales, sociales, etc).
Conocimientos	Profundización sobre la naturaleza del conocimiento que tiene la organización en el dominio, siendo este un aspecto fundamental en el dimensionamiento del sistema. El Modelo OM-4 profundiza específicamente en la descripción del conocimiento.
Cultura y Poder	Estilos de trabajo, comunicación, relaciones informales, etc. Las reglas no escritas.

Se busca hacer un análisis general de la organización buscando comprender su realidad y objetivos.

- OM-3: Descripción del proceso desde el punto de vista de las tareas en que está compuesto y sus características principales

Se describe en forma más concreta el proceso considerado en OM-2, dividiéndolo en tareas. Una **Tarea** es cada una de las funciones o divisiones de un proceso, representando una serie de actividades orientadas a lograr un objetivo específico.

La tarea es ejecutada por uno o varios **Agentes** que controlan la calidad en su ejecución. La tarea puede ser vista como un sistema que recibe unos insumos (entradas) y entrega productos (salidas) de una forma estructurada y controlada, consumiendo y proveyendo recursos dentro de un contexto.

Cada una de las tareas es listada incluyendo la siguiente información:

- Identificador y nombre.
- Agente: persona o recurso que ejecuta la tarea.
- Ubicación en la estructura organizacional donde se ejecuta la tarea.
- ¿Es la tarea intensiva en conocimiento? Conocimiento requerido para ejecutar la tarea.
- Significancia: qué tan importante es la tarea, representada en una escala.

- OM-4: Descripción del componente de conocimiento del modelo de la organización y sus principales características

Se describe en forma más concreta los activos de conocimiento considerados en OM-2.

Cada uno de los activos de conocimiento es listado incluyendo la siguiente información:

- Nombre (debe ser acorde a OM-3).
 - Agente que posee el activo (debe ser acorde a OM-3).
 - Tarea en la que es usado (debe ser acorde a OM-3).
 - ¿Se usa correctamente?
 - ¿Se usa en el lugar apropiado dentro de la organización?
 - ¿Se usa en el momento apropiado?
 - ¿Es un activo de buena calidad?
- OM-5: Lista de chequeo para el documento de viabilidad de la decisión

Análisis sobre la viabilidad del proyecto como solución a un problema, desde el punto de vista de Negocio:

- Beneficios tangibles e intangibles.
- ¿Qué tan grande es el impacto?
- Costos esperados.
- Comparación con otras soluciones alternativas.
- Cambios organizacionales requeridos.
- Riesgos económicos y de negocio.

Análisis sobre la viabilidad del proyecto como solución a un problema, desde el punto de vista Técnico:

- En términos del conocimiento almacenado y los procedimientos de razonamiento, ¿qué tan compleja es la tarea a ser desempeñada por el sistema? Si hay mucha complejidad, revisar estado del arte.
- Aspectos críticos relacionados con tiempo, calidad, recursos requeridos, etc.
- Complejidad de la interacción con los usuarios finales.
- Complejidad de la interacción con otros sistemas (interoperabilidad, integración de sistemas).
- Riesgos técnicos.

Además se debe incluir información adicional sobre la viabilidad del proyecto, y qué acciones se proponen para continuar el desarrollo.

4.5.4 Modelo de tareas

El modelo de tareas incluye 2 plantillas que se describen a continuación.

- TM-1: Descripción refinada de las tareas dentro del proceso objetivo

TM-1 contiene detalles sobre el flujo de información, datos involucrados, objetivo de la tarea, requerimientos de desempeño, criterios de calidad, restricciones, entre otros.

Tabla 4. TM-1. Descripción refinada de las tareas dentro del proceso objetivo

Tarea	ID y nombre de la tarea.
Organización	Proceso de negocio y dependencia organizacional donde se ejecuta la tarea.
Objetivo y valor	Describe el objetivo de la tarea y cómo al lograrlo se agrega valor al negocio.
Dependencias y flujo	Tareas de entrada (envían insumos). Tareas de salida (reciben producto de esta tarea).
Objetos manejados	Objetos de entrada (insumos). Objetos de salida (productos). Objetos internos (usados internamente por la tarea).
Tiempo y control	Frecuencia y duración de la tarea, y cómo se coordina con otras tareas. Restricciones de control: pre-condiciones y post-condiciones.
Agentes	Agentes involucrados en la ejecución de la tarea.
Conocimientos y competencias	Competencias requeridas para una adecuada ejecución de la tarea. Competencias entregadas a la organización. Los conocimientos se detallan en TM-2.
Recursos	Recursos requeridos por la tarea, profundizando OM-2.
Calidad y desempeño	Medidas que usa la organización para controlar la calidad y desempeño en la ejecución de la tarea.

Este formulario debe llenarse para cada una de las tareas del proceso que están siendo afectadas por el desarrollo del sistema.

- TM-2: Especificación del conocimiento empleado por una tarea y posibles cuellos de botella y áreas para mejorar

TM-2 se concentra en identificar los cuellos de botella y oportunidades de mejora relativos a las áreas intensivas en el uso de conocimiento.

Tabla 5. TM-2: Especificación del conocimiento empleado por una tarea y posibles cuellos de botella y áreas para mejorar.

Activo de conocimiento	Nombre del activo de conocimiento.	
Agentes	Agentes que poseen el activo de conocimiento.	
Tareas	Tareas donde es usado el activo de conocimiento.	
Dominio	Área donde está embebido el activo de conocimiento.	
Naturaleza del conocimiento	Cuello de botella a ser mejorado	
Formal, riguroso		
Empírico, cuantitativo		
Heurístico, por reglas		
Altamente especializado, dominio específico		
Basado en la experiencia		
Basado en la acción		
Incompleto		
Incierto, podría ser incorrecto		
Cambia rápidamente		
Difícil de verificar		
Tácito, difícil de transferir		
Forma del conocimiento		
Mente		
Papel		
Electrónica		
Competencia de acción		
Otra		
Disponibilidad del conocimiento		
Limitaciones en tiempo		
Limitaciones en espacio		
Limitaciones en acceso		
Limitaciones en calidad		
Limitaciones en forma		

Este formulario debe ser llenado para todos los activos de conocimiento especificados en OM-4.

4.5.5 Modelo de agentes

El modelo de tareas incluye 1 plantilla que se describe a continuación.

- AM-1: Especificaciones del Agente

El propósito es entender los roles y competencias que aportan los diversos actores en la organización para resolver las tareas consideradas.

Tabla 6. AM-1: Especificaciones del Agente.

Nombre	Nombre del agente.
Organización	Indicar procesos de negocio y dependencias organizacionales donde se encuentra el agente.
Tareas	Tareas donde está involucrado el agente.
Comunicaciones	Otros agentes con los que interactúa el agente que está siendo descrito.
Conocimientos	Activos de conocimiento que posee el agente.
Otras competencias	Otras competencias que tiene el agente.
Responsabilidades y restricciones	Responsabilidades y restricciones que tiene el agente al ejecutar tareas.
Recursos	Recursos requeridos por la tarea, profundizando OM-2.

Este formulario se debe llenar para cada Agente involucrado en las tareas y activos de conocimiento involucrados en el desarrollo.

4.5.6 Otras plantillas en el modelado del contexto. Se incluye 1 plantilla que se describe a continuación.

- OTA-1: Lista de chequeo para decisiones sobre impacto y mejoras

Se dan recomendaciones y acciones para continuar con el desarrollo del proyecto, teniendo en cuenta toda la información del nivel de Contexto de CommonKADS.

Se tiene en cuenta: Impactos y cambios que traería el sistema a la organización; Cambios específicos relativos a las tareas y agentes; Actitudes y compromisos; y Acciones propuestas.

4.5.7 Modelo de conocimientos. El Modelo de Conocimientos solo incluye la plantilla KM-1. Para modelar el conocimiento a emplear en la solución a un problema, se debe hacer desde tres perspectivas, como se mostrará a continuación.

- KM-1: Lista de comprobación de la documentación del modelo del conocimiento

El formulario KM-1 debe servir para verificar la siguiente información:

- Modelo de conocimientos. Más adelante se explican los aspectos que debe contemplar la especificación.
- Fuentes de información utilizadas.
- Componentes del modelo que fueron considerados para reutilización.
- Escenarios para solucionar el problema objeto de la aplicación.
- Resultado de las simulaciones hechas durante la validación del conocimiento.
- Elicitación de los requerimientos de conocimiento.

El Modelo de conocimientos especifica los requerimientos de conocimiento y razonamiento del sistema. Se analizan las tareas especificadas en OM-3 que luego fueron detalladas en el Modelo de tareas.

CommonKADS distingue entre los siguientes tipos de conocimiento: Esquema del dominio, Conocimiento de inferencia, y Conocimiento de tarea. El conocimiento de inferencia y el de tarea definen el *Modelo de Interpretación* del sistema. El conocimiento debe ser modelado en el lenguaje CML2, que está definido en [40] y [41].

- **Esquema del dominio.** Tiene como propósito definir la conceptualización del dominio y debe ser representado en forma independiente de cómo será usado por el sistema.

Se contemplan los siguientes tipos de conocimientos del dominio:

- **Concepto:** grupo de objetos o instancias que comparten características similares y hacen parte en el dominio. Es muy similar al concepto de *Clase* de un diagrama de clases UML. Se tienen en cuenta los atributos, valores, Y tipos de valor.
- **Relaciones:** se refiere a las relaciones entre dos o más conceptos. Son definidas mediante *Argumentos*, para los cuales se pueden especificar atributos como la *Cardinalidad* y el *Rol* del argumento dentro de la relación. Se puede comprender bajo la idea de *Asociación* de los diagramas de clase de UML.

- **Tipos de regla:** indican una relación entre dos predicados lógicos, que son normalmente expresiones del tipo atributo-valor de un concepto. Son entonces un tipo especial de relación, conexiones lógicas que un experto describe en el dominio.
- **Sub-tipo y Super-tipo:** definen jerarquías entre los conceptos. Se pueden entender bajo la definición de *Generalización* de un diagrama de clases UML.

Además se especifica una *Base de Conocimientos* (no necesariamente relacionada con la BC de la estructura de un SE), que contiene instancias de Conceptos, Relaciones, Tipos de regla, y demás elementos del conocimiento del dominio.

- **Conocimiento de inferencia.** Describe los pasos básicos para llegar a una inferencia. Es decir que define un método de solución del problema, la forma como se usa el conocimiento del dominio en las inferencias y los roles del conocimiento que modelan las premisas, y las conclusiones de las deducciones. Todas las inferencias básicas descritas constituyen los bloques con los que se construye el sistema de razonamiento.

Por lo general se definen *Roles de conocimiento*, cada uno de los cuales es un nombre abstracto para un objeto de datos, que identifica el rol de este último dentro del proceso de razonamiento. Hay *Roles de conocimiento estáticos* que no varían mucho en el tiempo y *Roles de conocimiento dinámicos* que varían cada vez que se ejecuta la aplicación del sistema.

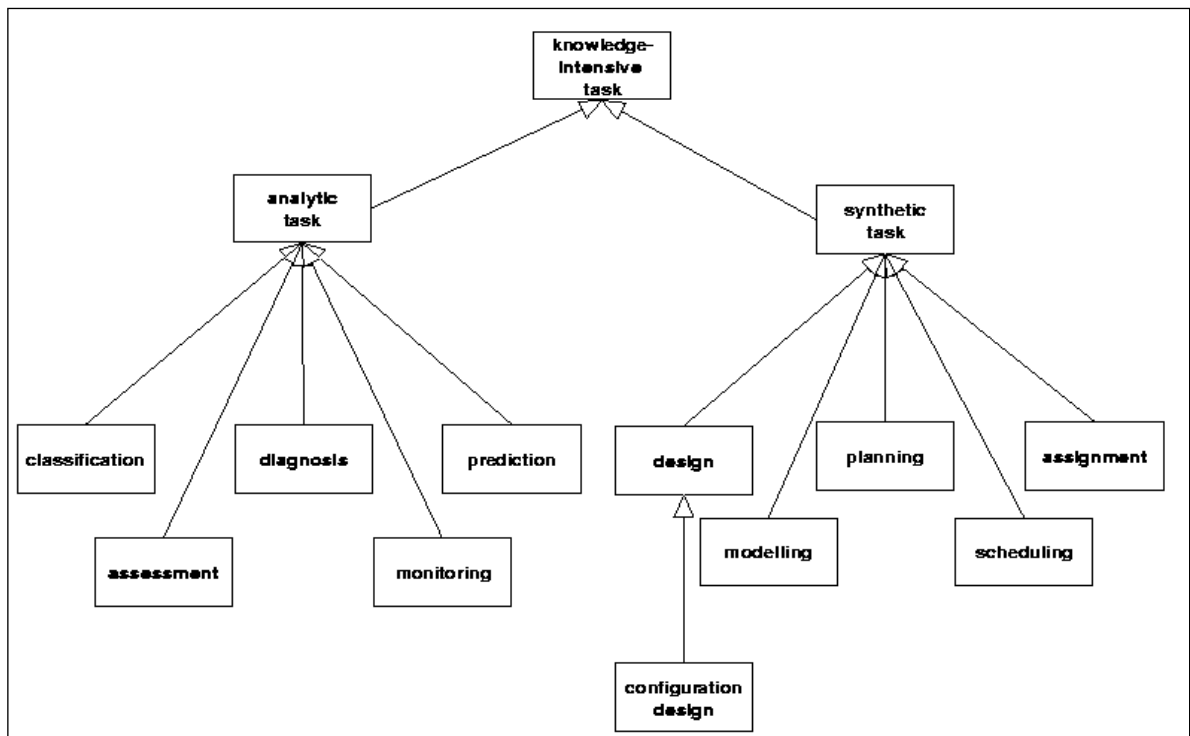
- **Conocimiento de tarea.** Describe qué objetivos busca la tarea, y cómo se pueden lograr mediante objetivos y sub-tareas que finalmente serán inferencias. Se representa una estrategia fija para alcanzar las metas de la solución del problema. Especifica el registro de la ejecución de los pasos de inferencia básicos definidos en el conocimiento de inferencia.

- **Plantillas para Modelado del Conocimiento.** El modelado del conocimiento es un trabajo de creación que puede requerir detalles muy específicos en cuanto a la descripción del dominio y los razonamientos del sistema. Por esto no se hacen exigencias estrictas sobre el modelado, aunque hay *Plantillas* que facilitan enormemente el trabajo mediante la reutilización del modelado del conocimiento para procesos básicos de razonamiento llamados **Tareas Intensivas en Conocimiento**. Para cada tarea se genera parcialmente un modelado que se puede reutilizarse en diversos escenarios, facilitando el trabajo del Ingeniero del Conocimiento.

Las tareas intensivas en conocimiento pueden clasificarse en dos grupos:

- **Tareas Analíticas:** La solución preexiste aunque esta sea desconocida. Los datos de entrada son acerca del problema que está siendo analizado, y los datos de salida son una caracterización del problema que lleva a la solución. Estas tareas son **Clasificación, Evaluación, Diagnóstico, Monitoreo, Predicción**.
- **Tareas Sintéticas:** La solución no existe antes de su desarrollo. Recibe como datos de entrada los requisitos para el sistema que se quiere construir, y los datos de salida son la descripción del sistema solicitado. Estas tareas son **Diseño, Modelado, Planeación, Programación y Asignación**.

Ilustración 5. Mapa de las Tareas Intensivas en Conocimiento de CommonKADS



Para este trabajo de grado se utilizará la plantilla de *Evaluación (o Valoración)*.

4.5.8 Modelo de Comunicaciones. Se debe partir por definir un Plan de Comunicaciones entre el Sistema y los Agentes. El Modelo de Comunicaciones tiene las plantillas CM-1 y CM-2.

- Plan de comunicaciones.

En el Modelo de Comunicaciones son fundamentales las transacciones. Una **Transacción** describe los actos de comunicación entre los diferentes agentes que participan en una tarea en el sistema. El Plan de comunicaciones da una descripción de todas las transacciones necesarias en el desarrollo del sistema.

Por lo general se puede representar con notaciones que muestren la comunicación de objetos de información entre los distintos agentes, de acuerdo a las tareas que cada uno de ellos ejecuta. Se recomienda usar un Diagrama de diálogo y un Diagrama de estados.

- CM-1: Especificación de las transacciones que posibilitan el diálogo entre dos agentes.

Se debe definir una transacción para cada objeto de información que surge como producto de una de las tareas identificadas en el Modelo de tareas, o en el Modelo de conocimientos. Estas son definidas en el formulario CM-1.

Tabla 7. CM-1: Especificación de las transacciones que posibilitan el diálogo entre dos Agentes.

ID y nombre	ID y nombre de la transacción. Debe representar claramente el objetivo de la transacción.
Objeto de información	Principal objeto de información de información generado, y entre qué tareas de transmite.
Agentes involucrados	Especificar qué agente envía y qué agente recibe la información.
Plan de comunicaciones	Referirse a algún Plan de comunicaciones ya especificado.
Restricciones	Pre-condiciones para que se pueda llevar a cabo la transacción, y post-condiciones que se asumen válidas después de la transacción.
Especificación de Intercambio de Información	Cada transacción puede tener una diferente estructura interna, o incluir objetos de información no generado por la tarea que envía el objeto principal. Estos detalles se referencian aquí pero se profundizan en CM-2.

Para cada transacción en el sistema, CM-1 ayuda a especificar sus objetivos y la forma como se debe implementar.

- CM-2: Especificación de los mensajes y los puntos de información que hacen una transacción individual

Este formulario tiene como objetivo profundizar la naturaleza de cada transacción, donde se requiera un nivel de detalle más avanzado.

Tabla 8. CM-2: Especificación de los mensajes y los puntos de información que hacen una transacción individual

Transacción	ID de la transacción
Agentes involucrados	Especificar qué agente que envía y qué agente que recibe la información.
Ítems de información	Listar todos los elementos de la información a ser transmitida, incluyendo el objeto principal y otros objetos de soporte. Para cada ítem se debe especificar el <i>Rol</i> (principal o soporte), <i>Forma</i> (formato del ítem), y <i>Medio</i> (medio como se despliega la información al agente receptor).
Especificación de mensajes	Describe todos los mensajes que componen la transacción. Para cada mensaje se debe especificar el <i>Tipo de comunicación</i> (intención), <i>Contenido</i> , y en algunos caso una <i>Referencia</i> a una descripción más precisa del contenido del mensaje.
Control sobre los mensajes	Si es necesario, se da una especificación de cómo se controlan los mensajes. Se puede utilizar un Diagrama de estados.

CM-A profundiza en los mensajes que se envían entre agentes para lograr una transacción, aunque no siempre es necesario llegar a este nivel de detalle.

4.5.9 Modelo de Diseño

El Modelo de Diseño tiene 4 plantillas que se detallan a continuación.

- DM-1: Descripción de la arquitectura del sistema.

La definición de una arquitectura generalmente tiene tres elementos: la descomposición del sistema en *subsistemas*; el *régimen de control*; y la descomposición del sistema en *módulos de software*. CommonKADS tiene una arquitectura predefinida, basada en la arquitectura conocida como Modelo-Vista-Controlador (MVC), que generalmente puede ser usada en la descripción de la arquitectura del sistema que se está usando.

MVC contempla tres subsistemas:

- **Modelo:** especifica las funciones y datos que proporcionan la funcionalidad de la aplicación. En CommonKADS el modelo debe incluir las funciones de razonamiento.
- **Vistas:** vistas externas para la interacción con el usuario del sistema. Se despliegan los objetos necesarios de acuerdo a las solicitudes del usuario.
- **Controlador:** es la unidad que controla la navegación en la aplicación. Generalmente es orientado a eventos generados por el usuario y/o el sistema para ejecutar comandos específicos.

Tabla 9. DM-1: Descripción de la arquitectura del sistema.

Decisión	Descripción
Estructura de subsistemas	Diagrama especificando los subsistemas de la aplicación. Puede referirse a una estructura estándar como el MVC.
Modelo de control	Caracterización general del régimen de control del sistema (por eventos, retorno de funciones, etc.).
Descomposición de subsistemas	Referencia a diagramas donde se explican los subsistemas por separado.

Es posible especificar otro tipo de arquitecturas mediante este formulario, aunque el MVC es una de las más usadas y ya existe un modelado para MVC reutilizable que provee CommonKADS.

- DM-2: Especificación de las facilidades ofrecidas por y en el sistema que será implantado

En DM-2 se detalla la plataforma donde será instalada la aplicación a desarrollar.

Tabla 10. DM-2: Especificación de las facilidades ofrecidas por y en el sistema que será implantado

Paquete de Software	Donde viene empaquetado el producto.
Hardware potencial	Plataformas opcionales de Hardware donde podría ser desplegado el sistema.
Hardware seleccionado	Plataforma de Hardware donde será desplegado el sistema.

Librería de vistas	Especificar si ya existen librerías de vistas para reutilización en el diseño de interfaces.
Tipado del lenguaje de programación	Hasta qué nivel el lenguaje no permite violaciones en el manejo de los datos.
Representación del conocimiento	Facilidades que da el lenguaje para emular el razonamiento del experto (si es declarativo o procedimental, si se pueden declarar reglas, etc.).
Protocolos de interacción con sistemas externos	Especificar protocolos de interacción.
Flujo de control	Especificar cómo se ejecutarán los mecanismos de control del sistema (mensajes, múltiples hilos, etc.).
Soporte a CommonKADS	Especificar si el lenguaje a usar provee arquitectura predefinida para dar soporte a CommonKADS.

Se busca definir el entorno tecnológico donde se desplegará la aplicación.

- DM-3: Lista de chequeo de decisiones en relación con la especificación de la arquitectura

En DM-3 se definen con mayor detalle los componentes de la arquitectura del sistema.

Tabla 11. DM-3: Lista de chequeo de decisiones en relación con la especificación de la arquitectura.

Controlador	Mecanismos para el manejo de eventos internos y externos.
Tarea	Cómo inicializar una tarea, qué tipo de estados puede tener.
Método de tarea	Mecanismos de control para ejecutar tareas, desde el lenguaje de programación. Especificar si estos son declarativos o imperativos.
Inferencia	Definir operaciones para controlar el estado de la inferencia: estado, si puede hallar una respuesta, etc.
Método de inferencia	Algoritmos para ejecutar las inferencias.
Roles dinámicos	Tipos de datos que se podrían instanciar durante la ejecución del sistema. Definir cómo se acceden y cómo se modifican.
Roles estáticos	Tipos de datos que no son modificados durante la

	ejecución del sistema. Definir cómo se acceden.
Base de conocimientos	Definir cómo se va a representar, acceder y modificar el conocimiento del experto dentro del sistema.
Vistas	Interfaces a utilizar.

Se busca definir si la arquitectura soporta todo lo necesario para garantizar un flujo de trabajo apropiado para los agentes usuarios de la solución.

- DM-4: Decisiones del diseño de la aplicación

En DM-4 se terminan de definir las decisiones de diseño del sistema.

Tabla 12. DM-4: Decisiones del diseño de la aplicación.

Controlador	Definir cómo se manejarán los eventos en el sistema, siguiendo las especificaciones del Plan de Comunicaciones y sus transacciones, ya definidos en el Modelo de Comunicaciones
Métodos de tarea	Formalizar las estructuras de control a usar.
Roles dinámicos	Escoger un tipo de dato para cada uno de los roles.
Inferencias	Mostrar un ejemplo de cómo se invocarán las inferencias a implementar en el sistema. Los roles dinámicos y estáticos pasan a ser parámetros de los métodos de inferencia.
Métodos de inferencia	Especificar los métodos de inferencia a aplicar.
Modelos del dominio	Todo el modelado del dominio es representado de acuerdo a las especificaciones de la arquitectura.
Objetos de vistas	Seleccionar las interfaces visuales más apropiadas para interactuar con el usuario.

Con DM-4 se completa la descripción formal de los módulos de software que compondrán la solución.

5. METODOLOGÍA DE TRABAJO

Este proyecto busca demostrar la posibilidad de reproducir los procesos de razonamiento de un experto en Selección de Personal. Por ser este un desarrollo que siempre estará propenso a errores y/o ampliaciones, la metodología para el desarrollo del software tenía que permitir el continuo crecimiento y control de errores en la aplicación, a partir del juicio del cliente y/o experto en selección de personal.

Los aspectos claves de la metodología son listados a continuación, y serán profundizados a lo largo de este capítulo:

- La elicitación de requisitos del sistema se hizo mediante la herramienta REM (REquirements Management), siguiendo una metodología propuesta por los creadores de esta aplicación.
- El Desarrollo en Espiral fue el ciclo de vida inicialmente seleccionado en el anteproyecto para el desarrollo de este prototipo porque permite un mejor control del producto final por parte del cliente del desarrollo. Dado que este el objetivo es obtener un Sistema Basado en el Conocimiento, se consideró altamente conveniente utilizar la metodología CommonKADS, la cual proporciona ventajas que facilitan la Ingeniería del Conocimiento. CommonKADS es completamente compatible con el Desarrollo en Espiral, por lo que fue la metodología seleccionada para modelar el sistema.
- El modelado mediante UML fue incluido dentro del modelado obtenido en CommonKADS. Con esto se espera haber logrado un mejor nivel de detalle en las especificaciones del problema y la solución a implementar.
- La elección de la herramienta de desarrollo también fue hecha en el modelado de la organización de CommonKADS. Se escogió el lenguaje de programación PHP, el cual es imperativo, porque entre otras razones ya la organización cuenta con infraestructura y personal capacitado para soportar un ambiente de desarrollo en PHP, lo cual es un gran activo que facilita el despliegue de aplicaciones en Internet (Computación en la Nube). Sin embargo, fuera del modelado se hace un análisis breve de algunos de los lenguajes de programación declarativa más comunes en Inteligencia Artificial, eligiendo uno suponiendo que se usará para el desarrollo del sistema (lo cual no va a suceder).
- El desarrollo de la aplicación evolucionaba con cambios y nuevos aspectos tenidos en cuenta a medida que evolucionaba el modelado en CommonKADS.

El modelado que se encuentra en los RESULTADOS define el sistema desarrollado.

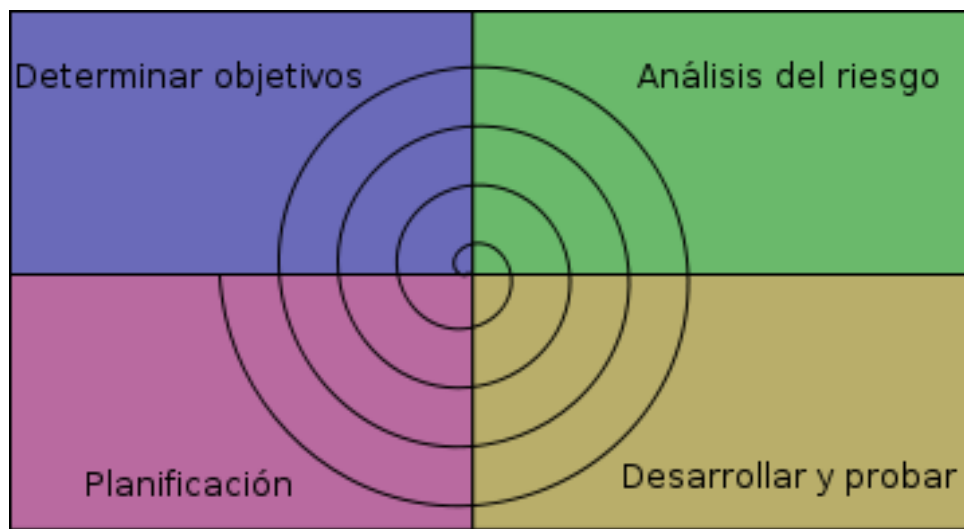
- En las bases de datos y base de conocimientos del sistema se agregaron los casos de prueba para demostrar la confiabilidad en el sistema experto. Estos Casos de Prueba también pueden hallarse en el Anexo C.

El ciclo de vida del Desarrollo en Espiral es un enfoque evolutivo que permite evaluar el desempeño y los riesgos asociados al resultado final para diseñar estrategias que permitan llegar a lo deseado. Permite el crecimiento y la corrección del software en ciclos iterados. Contempla las siguientes actividades principales, que se repiten hasta que el producto sea terminado:

Tabla 13. Iteraciones del modelo de desarrollo en espiral

Etapa	Descripción
Determinar objetivos	Identificación de objetivos a corto y largo plazo con base en las necesidades que señale el ingeniero del conocimiento a medida que se desarrolle el proyecto. Diseño de las actividades a realizar en base a esos objetivos.
Análisis de riesgo	Evaluación de las distintas alternativas para el desarrollo de la aplicación, teniendo en cuenta los objetivos que se determinaron. Se hace también un análisis de la ingeniería de requisitos.
Desarrollo y pruebas	Desarrollo de una nueva versión de la aplicación y/o del modelado del sistema. Se prueba la aplicación y se hace retroalimentación.
Planificación	Con base en los resultados obtenidos en la fase anterior, se planifican las actividades futuras para continuar el proyecto y/o definir cambios necesarios.

Ilustración 6. Iteraciones del modelo de desarrollo en espiral



Con cada iteración alrededor de la espiral (comenzando en el centro y siguiendo hacia el exterior), se construyen sucesivas versiones del software, cada vez más completa y, al final, al propio sistema operacional [30].

5.1 ELICITACIÓN DE REQUISITOS CON LA HERRAMIENTA REM

La **Ingeniería de Requisitos** comprende todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer para el sistema a desarrollar, tomando en cuenta los diversos requisitos de los potenciales demandantes del software. El propósito de la ingeniería de requisitos es hacer que los mismos alcancen un estado óptimo antes de alcanzar la fase de diseño en el proyecto. Los buenos requisitos deben ser medibles, comprobables y sin ambigüedades o contradicciones.

Dentro de la ingeniería de requisitos, la etapa de **Elicitación de Requisitos** abarca la primera y quizás más importante fase dentro del desarrollo de un sistema informático. Uno de los retos más importantes de la elicitación de requisitos es garantizar que los requisitos del sistema sean consistentes con las necesidades de la organización donde se utilizará el mismo y con las futuras necesidades de los usuarios. Por otra parte los modelos actuales de competitividad en gestión de las organizaciones incluyen como condición que la gestión esté basada en procesos más que en el establecimiento jerárquico de los cargos [33, 34].

REM es una herramienta experimental gratuita diseñada para soportar la fase de Ingeniería de Requisitos de un proyecto de desarrollo de software. REM se basa en la metodología definida en la tesis doctoral "Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información", presentada por Amador Toro Durán, de la Universidad de Sevilla, en septiembre de 2000. Esta metodología es revisada y perfeccionada en 2002 [35].

La metodología soportada por REM contempla seis tareas en la elicitación de requisitos:

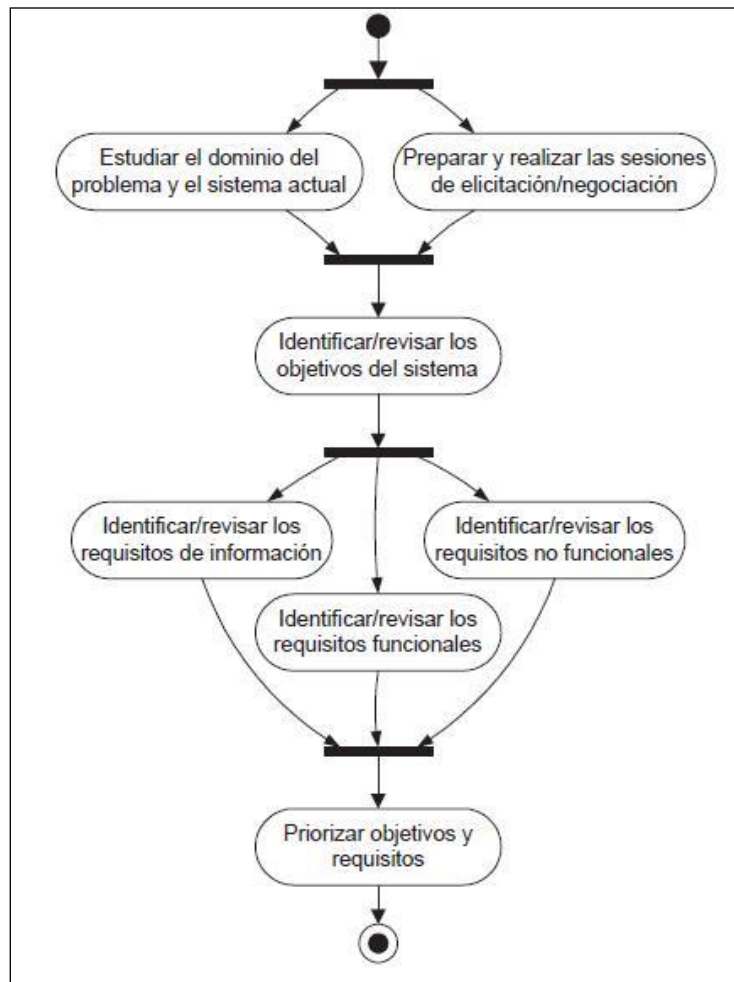
- Tarea 1: Obtener información sobre el dominio del problema y el sistema actual
- Tarea 2: Preparar y realizar las sesiones de elicitación/negociación
- Tarea 3: Identificar/revisar los objetivos del sistema
- Tarea 4: Identificar/revisar los requisitos de información

- Tarea 5: Identificar/revisar los requisitos funcionales
- Tarea 6: Identificar/revisar los requisitos no funcionales

El documento generado por REM como resultado de la Elicitación de Requisitos es el Anexo A.

Los requisitos de la primera iteración en el proceso de desarrollo, fueron obtenidos mediante una entrevista semanal durante un tiempo aproximado de dos meses, para un total de **8 entrevistas**. En estas estuvo involucrado el autor y el asesor de este proyecto.

Ilustración 7. Tareas en la elicitación de requisitos



5.2 MODELADO DEL SISTEMA USANDO UML EN COMMONKADS

El 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 "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar 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.

Se consideró que UML por sí solo no es suficiente para modelar el sistema a desarrollar, dado que es importante abstraer un conocimiento preciso sobre el ambiente empresarial donde se va a desplegar la solución y el modelado del conocimiento de un experto en Selección de Personal. Sin embargo UML es bastante útil para definir los conceptos importantes mediante herramientas como los Diagramas de clases.

UML se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software. En este proyecto UML soporta a CommonKADS.

CommonKADS es una metodología para la construcción de Sistemas Basados en el Conocimiento (SBC) y donde es fácil incluir el modelado que se haga en UML, por lo que se tomó como marco principal de referencia para modelar el sistema.

A lo largo de todos estos modelos es posible y conveniente utilizar Diagramas UML, lo cual está directamente relacionado con uno de los objetivos de este proyecto. Por eso en el modelado CommonKADS se hará uso de UML.

En el modelado del conocimiento se prescindirá del modelado en el lenguaje CML2. Se hará mayor uso de las Plantillas de Conocimiento que provee CommonKADS como "bases" del modelado general. En específico se usará la Plantilla para la **Tarea Intensiva en Conocimiento de Evaluación (Assessment)**.

Después de la elicitación de requisitos con REM, se inició el desarrollo del sistema y esto requirió que se llevaran a cabo entrevistas a medida que se avanzaba en la

metodología CommonKADS. Se llevaron a cabo aproximadamente **25 a 30 entrevistas** a lo largo de un año. Estas reuniones involucraron al autor del proyecto y el asesor del proyecto. Un experto en Selección de Personal fue invitado a participar en 3 de esos encuentros. Los requisitos adicionales obtenidos durante estas iteraciones no fueron levantados en REM.

5.2.1 Equipo de trabajo. Se mencionan los roles a tener en cuenta en un proyecto CommonKADS, y se detallan las personas que ejercerán cada uno de esos roles.

Ilustración 8. Roles y responsables en el proyecto

Rol	Descripción	Responsable
Proveedor de conocimiento	Experto tradicional, con una amplia experiencia en su dominio de trabajo. Su cooperación es vital.	Álvaro Jaramillo y Juliana Trujillo
Ingeniero del conocimiento	Analista de sistemas especial que busca enlazar el dominio del experto y el sistema que se construye. Debe evitar convertirse en un experto (proveedor de conocimiento), sino representar el conocimiento de tal forma que su uso sea automatizable.	Esteban Pérez
Desarrollador del sistema experto	Implementa el sistema en una plataforma tecnológica seleccionada. Debe comprender el análisis del conocimiento del dominio. Es necesario que tenga habilidades para el diseño e implementación de sistemas.	Esteban Pérez
Usuario del conocimiento	Interactúa y/o es afectado por el sistema. Es importante tener en cuenta su nivel de experticia en el uso de herramientas tecnológicas y el conocimiento del dominio de la aplicación.	Seleccionadores de Personal
Gerente de proyecto	Planea, programa y monitoriza el trabajo de desarrollo. Es responsable ante el cliente y/o usuario del sistema.	Esteban Pérez
Administrador del conocimiento	Monitorea el cumplimiento del propósito del sistema a nivel organizacional, y los activos de conocimiento generados.	Álvaro Jaramillo

Nótese que algunas personas pueden asumir dos o más roles, dependiendo del personal disponible y la conveniencia que así sea para el proyecto.

5.3 ELECCIÓN DE LA HERRAMIENTA DE DESARROLLO (LENGUAJE DE PROGRAMACIÓN DECLARATIVA)

No hay un método reconocido y avalado para elegir un lenguaje de programación, puesto que la mayoría de estos son hechos para satisfacer necesidades específicas de un mercado para proyectos con necesidades específicas. Es cierto que hay algunas propuestas sobre los criterios que se deben tener en cuenta, pero esto depende del tipo de desarrollo que se va a hacer.

Por lo tanto, para el desarrollo de este proyecto se eligió una serie de parámetros que se deben tener en cuenta para desarrollar el prototipo de sistema experto. Algunos de estos parámetros serán tenidos en cuenta por las necesidades de este sistema en específico y serán obligatorios. Otros parámetros, más generales para cualquier tipo de sistema, serán tenidos en cuenta según lo descrito por Lawlis [32] en su propuesta de parámetros a tener en cuenta para elegir un lenguaje de programación.

Es importante mencionar que el lenguaje de programación no es la única herramienta a elegir en el desarrollo de un sistema, aunque generalmente es una de las más importantes. Otros aspectos que pueden ser cruciales son el motor de la base de datos, las facilidades para interoperar, el proveedor del servicio de hosting, entre otros.

5.3.1 Parámetros obligatorios para elegir el lenguaje. Los siguientes requisitos son indispensables para la selección del lenguaje de programación a utilizar.

- **Encadenamiento del razonamiento.** En este sistema lo que se busca es encontrar hechos (datos) a partir de otros hechos y reglas de inferencia. Por lo tanto, **para desarrollar este prototipo de este proyecto es necesario utilizar un lenguaje declarativo que soporte el encadenamiento hacia adelante** por que se parte de hechos que son el resultado de las mediciones del candidato y de las características del perfil del cargo dentro de la Selección de Personal.

El lenguaje de programación puede actuar de dos maneras para encontrar los valores de verdad que necesita: por encadenamiento hacia adelante, o por encadenamiento hacia atrás.

El encadenamiento hacia adelante es un razonamiento dentro de un sistema lógico que avanza desde los datos o premisas hacia las conclusiones cuya validez aún no son conocidas. El encadenamiento hacia adelante no es más que el uso aplicado de la implicación tautológica Modus Ponens.

Modus Ponens puede ser explicado mediante la siguiente fórmula:
$$[(p \rightarrow q) \wedge p] \rightarrow q.$$

Esto quiere decir que si p implica a q , y p es verdadera, entonces q es verdadera.

Por su parte, el encadenamiento hacia atrás tiene como objetivo “sustentar” un teorema, y para ello se hace la búsqueda de las premisas y otra información que permita sustentarlo.

El encadenamiento hacia atrás es una forma de razonar dentro de un sistema lógico que busca seguir la ruta desde lo que se quiere sustentar en su validez (un teorema) hacia los datos (las premisas). Aquí se parte de algo que se quiere demostrar y se buscan las oraciones que permitirían llegar a esa conclusión. Por último, se busca establecer las premisas correspondientes. Supóngase que dado A se presentará B, y que dado B se presentará C ($A \rightarrow B$ y $B \rightarrow C$). Se le debe dar al sistema una premisa a evaluar, supóngase C. El sistema encontrará que para que se dé C se debe dar B. Luego evalúa si se da B, pero encontrará que se dará solo si se presenta A. El sistema podrá preguntar al usuario si A es cierta o falsa, y a partir de allí definir la validez o falsedad de B y finalmente confirmar la validez o falsedad de C; en otros casos el sistema simplemente no encuentra la premisa A, de modo que acepta la falsedad de A y por lo tanto de B y de C.

▪ **Tipo de licencia.** El lenguaje escogido debe ser de libre distribución, por lo menos para usos académicos. No es posible escoger un lenguaje de licencia protegida para cualquier uso, dado que elevaría considerablemente los costos del desarrollo y mantenimiento del sistema.

5.3.2 Parámetros adicionales. Lawlis [32] expone aspectos importantes a tener en cuenta a la hora de elegir un lenguaje de programación, y serán explicados a continuación.

- **Claridad del código fuente:** La legibilidad es la propiedad del lenguaje de la que depende que el código escrito sea lo suficientemente claro y entendible no solo para el desarrollador, sino para otras personas que quieran leer o modificar el código programado. A juicio del autor de este documento, la claridad del código fuente programado depende de lo oportunos y claros que sean los comentarios que se encuentren a lo largo del código. Hay propuestas sobre cómo escribir el código fuente en sí, pero dado que la legibilidad depende también de quien lee el código, todas esas propuestas son controversiales. Lo que se puede garantizar es que el código fuente tendrá comentarios que de verdad ayuden a entenderlo y se usarán tabulaciones y espacios que hagan el código tan legible como sea posible. En este estado de cosas, este parámetro no se tendrá en cuenta. Pero queda el compromiso por parte del desarrollador de hacer el código tan claro como sea posible.
- **Administración de desarrollos complejos:** Se refiere a las facilidades que ofrece el lenguaje para gestionar la complejidad que se podría presentar al desarrollar un sistema. Este prototipo no tendrá una complejidad muy grande, por lo que este parámetro no será tenido en cuenta.
- **Soporte de concurrencia:** Se refiere a la capacidad del lenguaje para soportar dos o más tareas al mismo tiempo usando los mismos recursos

(procesador, memoria, etc.). En este proyecto esto no es crítico, por lo que no se tendrá en cuenta.

- **Soporte de sistemas distribuidos:** Se refiere a la capacidad del sistema para trabajar en forma distribuida en dos o más sistemas para llevar a cabo una tarea única. En este proyecto esto no es crítico, por lo que no se tendrá en cuenta.
- **Facilidad de mantenimiento:** Se refiere a las facilidades que ofrece el lenguaje para hacer cambios en el código fuente cuando sea necesario. Tiene mucho que ver con la Claridad del código fuente, por lo que en realidad esta característica no es tan dependiente del lenguaje en sí, sino de las buenas prácticas de programación. No se tendrá en cuenta para la selección del lenguaje por las mismas razones que no se tendrá en cuenta la Claridad del código.
- **Facilidad de integración:** Es la capacidad de un sistema para ser integrado a otras tecnologías. Por ejemplo, JESS puede ser muy bien integrado a Java y CLIPS se conecta muy bien con C. En algunos casos la integración está muy relacionada con el sistema operativo que se está usando, arruinando la portabilidad de la aplicación.
- Sin embargo, este prototipo de sistema basado en el conocimiento no tiene ninguna necesidad de ser conectado al núcleo de algún sistema operativo en específico, por lo que la facilidad de integración no riñe con la portabilidad.
- **Portabilidad:** Portabilidad es la propiedad que tiene un sistema de poder ser ejecutado en distintas plataformas (sistemas operativos). Incluso si se piensa que el sistema solo va a ser ejecutado bajo ambientes Windows, no se debe descartar la posibilidad que en algún momento tendrá que ejecutarse la aplicación en plataformas Unix, Linux, etc.
- **Soporte a ejecución del sistema en tiempo real:** Se refiere a la capacidad del sistema de llevar a cabo instrucciones en el mismo momento en el que está siendo ejecutado. Dependiendo de especificaciones de funcionalidad, varios lenguajes “aparentan” llevar a cabo las instrucciones pero en realidad no lo hacen. Lawlis propone que el lenguaje elegido debe poder llevar a cabo las instrucciones en tiempo real. En este proyecto esto no es crítico, por lo que no se tendrá en cuenta.
- **Confiabilidad y Seguridad:** El lenguaje elegido debe permitir al sistema desarrollado llevar a cabo sus funciones de forma satisfactoria a lo largo de su ejecución, de acuerdo a lo que se espera que haga la aplicación. Además el lenguaje debe ser libre de fallas tanto como sea posible.

- **Reusabilidad:** El lenguaje debe permitir la adaptación del código fuente en otra aplicación.
- **Estandarización:** La definición del lenguaje debe ser formalmente estandarizada, reconocida por entidades tales como la ANSI o la ISO. El lenguaje por lo tanto debe poder ser interpretado en plataformas también estandarizadas.
- **Soporte:** El soporte es esencial, pues permite al desarrollador solucionar los problemas que ocurran durante la codificación en lo referente a la sintaxis del lenguaje. Se espera que el lenguaje seleccionado tenga una documentación de su sintaxis, una página Web donde se proporcione más documentación relacionada. En lo posible, también debe haber un lugar en el ciberespacio donde sea posible que otros programadores ayuden a solucionar problemas de codificación. Un buen soporte es tal vez la característica más importante que debe tener el lenguaje de programación elegido.

5.3.3 ¿Por qué no se usó un lenguaje declarativo en el desarrollo del sistema?

En CommonKADS, en el Modelado de la Organización se encontró que el lenguaje elegido para el desarrollo del sistema fue PHP, el cual es un **lenguaje imperativo**. Esto se debió, entre otras razones, a las ventajas que ofrece PHP para soportar el despliegue de aplicaciones en la infraestructura tecnológica de la organización. PHP es además una herramienta de libre distribución y se puede desarrollar una aplicación de tal forma que “simule” un encadenamiento hacia adelante en la resolución de un problema de razonamiento.

Sin embargo, para efectos de cumplir con uno de los objetivos iniciales de este trabajo de grado, la metodología presentada sí fue usada para elegir un **lenguaje declarativo** de programación. Este lenguaje declarativo no fue usado en el desarrollo del sistema, simplemente se seleccionó como ejercicio académico.

6. RESULTADOS

6.1 ELICITACIÓN DE REQUISITOS CON LA HERRAMIENTA REM

La elicitación de requisitos con REM se hizo solo una vez, antes de iniciar el modelado y desarrollo formal del sistema.

Se obtuvo el listado de requerimientos iniciales antes de empezar el desarrollo del sistema. El resultado puede ser visto en el ANEXO A.

6.2 INGENIERÍA DEL CONOCIMIENTO CON COMMONKADS

A continuación se muestra el resultado de todas las plantillas para todos los modelos de CommonKADS.

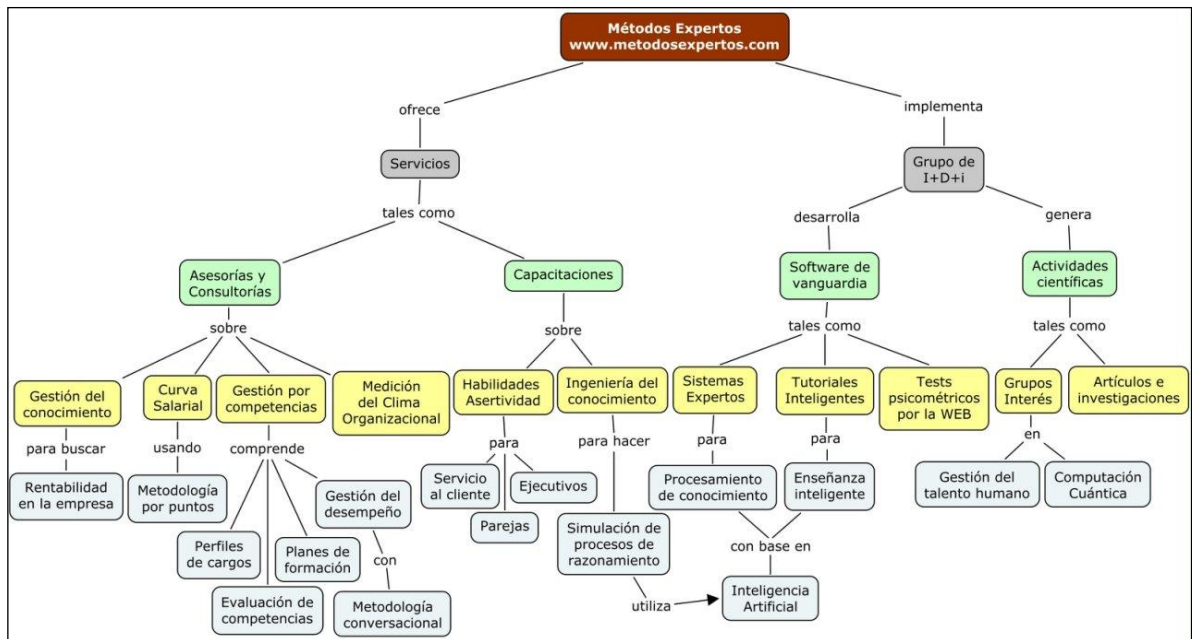
6.2.1 OM-1: Identificación de problemas y oportunidades orientados al conocimiento

Métodos Expertos, es una organización que se dedica a promover la productividad empresarial de sus clientes, aprovechando el conocimiento organizacional de sus empleados. Esto se logra mediante metodologías estructuradas para integrar ese conocimiento a sistemas de computación, en lo que se conoce como Ingeniería del Conocimiento.

Métodos Expertos busca, desde las distintas áreas profesionales representadas en su equipo, generar soluciones tecnológicas que hagan uso y aprovechen al máximo el conocimiento que se requiere dentro de las empresas. Específicamente en Psicología, se está buscando optimizar los procesos de Selección de Personal en las organizaciones, siendo estos críticos para administrar de la mejor forma el talento humano de las empresas.

La Selección de Personal (SP) es el primer asunto a resolver que en relación con el personal se le plantea a cualquier empresa. El proceso de SP es aquel en el que se decide si se contratará o no a los candidatos encontrados en la búsqueda realizada previamente.

Ilustración 9. Mapa Conceptual de Métodos Expertos



Cuando se planifica el proceso de SP se debe tener en cuenta la importancia de la confiabilidad en los instrumentos de medición de las capacidades de los posibles candidatos. Para realizar el proceso de selección de personal se deben diseñar distintas pruebas confiables donde el postulante demuestre si es capaz de realizar el trabajo. En las pruebas se deben tener en cuenta elementos como los títulos obtenidos, conocimientos gerenciales y técnicos, experiencia laboral, entrevistas, perfil psicológico, perfil del cargo dentro de una empresa, etc.

Las organizaciones necesitan seleccionar a los mejores candidatos para asegurar alto desempeño en el trabajo. Los procesos de selección comprenden procesos de evaluación de competencias para identificar los empleados más preparados para desarrollar las actividades. Las evaluaciones requieren de expertos en el tema para evaluar, analizar y tomar decisiones acertadas en la selección de personal calificado.

Es frecuente encontrar inconsistencias en las decisiones por parte de los expertos en selección ya que no utilizan un procedimiento formalizado y los conocimientos del dominio están influenciados por criterios personales que difieren de experto a experto. Además, los costos de evaluación son altos causados por desplazamientos y por la logística de aplicación con lápiz y papel.

Para resolver esta situación se busca obtener el conocimiento de un experto con la suficiente experiencia en seleccionar personal, de tal manera que utilice los mismos parámetros de evaluación para aumentar consistencia en las decisiones. También se busca, mediante el uso de Internet, permitir el acceso a los candidatos y a las empresas empleadoras de forma remota, con el fin de facilitar la logística y el ahorro en costos tanto para los candidatos como para los seleccionadores.

Se podría ofrecer a las empresas la posibilidad de evaluar a sus candidatos y a sus empleados de forma remota, obteniendo resultados altamente confiables sobre las competencias que poseen, permitiendo tener información consistente y confiable y ahorrando costos y tiempo por la gestión logística.

La Organización Internacional del Trabajo OIT (1993) define el concepto de "Competencia como la idoneidad para realizar una tarea o desempeñar un puesto de trabajo eficazmente por poseer las calificaciones requeridas para ello". Según el Ministerio de Educación de Colombia, "las Competencias Laborales Generales son aquellas que se aplican a cualquier clase de trabajo y sector económico. Son las que se utilizan en cualquier espacio laboral y que preparan para cualquier clase de trabajo. Las competencias laborales comprenden todos aquellos conocimientos, habilidades y actitudes que son necesarios para desempeñarse con eficiencia como personas productivas". Una persona es competente cuando ha logrado desarrollar la destreza para el trabajo y las habilidades personales para adaptarse a las exigencias del entorno. El desarrollo de las competencias es un asunto de tiempo y dedicación.

Leonard Mertens (1996) diferencia la competencia laboral de la competencia personal. La primera requiere de la aplicación de los aspectos personales como habilidades, aptitudes, actitudes, conocimientos. Estos aspectos permiten la contextualización en el mercado y preparan para la competitividad. "La empresa y la persona son competentes no tanto por cumplir con un desempeño mínimo aceptado sino por tener la capacidad de destacarse en el medio".

La **Competencia Personal** permite un funcionamiento de la persona en el contexto profesional con características como perseverancia, flexibilidad, autonomía, responsabilidad, etc. que la hacen apto para un eficiente desempeño. Esta cualidad de la competencia personal es considerada por el Análisis Ocupacional y Funcional del Trabajo (IBERPROF, Organización de Estados Iberoamericanos, 2000) cuando plantea: "Competente: persona que posee un repertorio de habilidades, conocimientos y destrezas y la capacidad para aplicarlas en una variedad de contextos".

6.2.2 OM-2: Descripción de los aspectos de la organización que tienen impacto en o están afectados por el problema elegido

A continuación se detalla la información de esta plantilla.

▪ **Estructura, Cultura y Poder.** Métodos Expertos está dirigido por un Director quien es el principal tomador de decisiones organizacionales. Se cuenta con personas expertas en las áreas mostradas en el mapa conceptual, incluido el Director, para llevar a cabo el trabajo cotidiano que es intensivo en conocimiento. Más que una jerarquía, el grupo tiene una estructura que privilegia el trabajo en red entre los distintos miembros donde cada uno aporta conocimientos y soluciones a problemas desde el punto de vista de cada una de las áreas representadas.

Esta estructura una buena fluidez de la comunicación dentro del grupo. Los miembros del equipo generalmente participan activamente en la toma de decisiones importantes junto con el Director.

▪ **Proceso, Actores y Conocimiento en Selección de Personal.** Para llevar a cabo la SP es necesario que se reciban solicitudes por parte de las empresas que necesitan llenar cargos vacantes, y **Candidatos** que busquen ser elegidos para esos cargos. El cargo que busca llenar el **Empleador** debe ser definido de forma estructurada por los profesionales que ejecutan el proceso SP, obteniendo un **Perfil del Cargo** (PC). Para poder realizar una evaluación de competencias, este perfil debe tener una descripción de las competencias personales con las que debe cumplir el candidato ideal.

El Empleador es el departamento organizacional interno que tiene el cargo vacante.

Una vez obtenido el Perfil del Cargo, se debe buscar entre las **Hojas de Vida** presentadas por personas que quieren ocupar el cargo vacante. En la hoja de vida de cada Candidato se busca si el Candidato cumple o no con los requerimientos básicos del Cargo. Esta categorización puede basarse en Competencias Laborales, Experiencia Académica, Edad, entre otros. Si el Candidato cumple con esos requerimientos básicos, sigue en el proceso de selección.

A continuación se deben ejecutar los siguientes pasos, dependiendo del diseño del proceso de selección de la organización. Algunos pasos se pueden omitir, y en algunos casos se pueden ejecutar de forma paralela:

- El Candidato presenta una o varias **Pruebas Psicológicas** que sirven para definir su **Perfil Psicológico** en un contexto dado, que provee la medida del cumplimiento de candidato con una serie de competencias personales. Después se hace una **Evaluación de Competencias** que requiere el conocimiento de un **Seleccionador**.
- Se aplica una o varias Entrevistas Estructuradas al Candidato.

- Se realiza una Verificación de las Referencias Laborales del Candidato.
- **Se realiza una visita domiciliaria al Candidato.** Al tener datos de las etapas anteriores se puede realizar la tarea **Evaluar ajuste del Candidato para el Cargo** que usualmente la lleva a cabo el Seleccionador. Se obtiene una decisión sobre si el aspirante se ajusta a un cargo específico, lo cual se debe notificar al Empleador para que pueda proseguir con el proceso de contratación del Candidato o para continuar con otros procesos de selección.
- **Recursos.** La Infraestructura Tecnológica de Métodos Expertos para el soporte de las BD y las aplicaciones está en un servidor de hosting compartido por distintos usuarios, administrado por una empresa dedicada a prestar este tipo de servicios, operando bajo un ambiente LAMP (Linux, Apache, MySQL y PHP). Métodos Expertos tiene una cuenta privada en ese servidor, lo que le ha permitido desarrollar aplicaciones en este ambiente y estar familiarizado con él. La seguridad está administrada por la empresa prestadora del servicio de hosting en el servidor compartido.

Este tipo de hosting permite tener diversos beneficios. Las aplicaciones y BD tienen un riesgo mínimo de ser perdidas, pues están bajo una infraestructura que contempla altos estándares de seguridad para el almacenamiento de datos. En un computador de escritorio o un servidor propiedad adquirido, los datos correrían un riesgo mucho mayor de ser accedidos y/o borrados. Además el acceso a las aplicaciones y la BD puede hacerse desde cualquier equipo de cómputo con conexión a Internet, mientras se tengan las credenciales de acceso apropiadas (usualmente un nombre de usuario y una contraseña).

Los miembros de Métodos Expertos ya están familiarizados con el uso de las herramientas que tienen en su cuenta de hosting, pues acceden a los datos y las aplicaciones en su trabajo del día a día. Idealmente se busca que el nuevo sistema se integre a ese entorno basado en web, con el fin de maximizar el aprovechamiento de los recursos ya existentes.

Los miembros de Métodos Expertos son profesionales con habilidades en el uso de herramientas tecnológicas modernas como navegadores web, procesadores de texto, correo electrónico, etc. Cada uno de ellos cuenta con una computadora personal, y están muy familiarizados con el uso de Internet en el día a día de su trabajo, aprovechando las ventajas que ofrece la red para adquisición de nuevos conocimientos y competencias.

Ilustración 10. Diagrama de actividades (UML) del proceso de Selección de Personal.

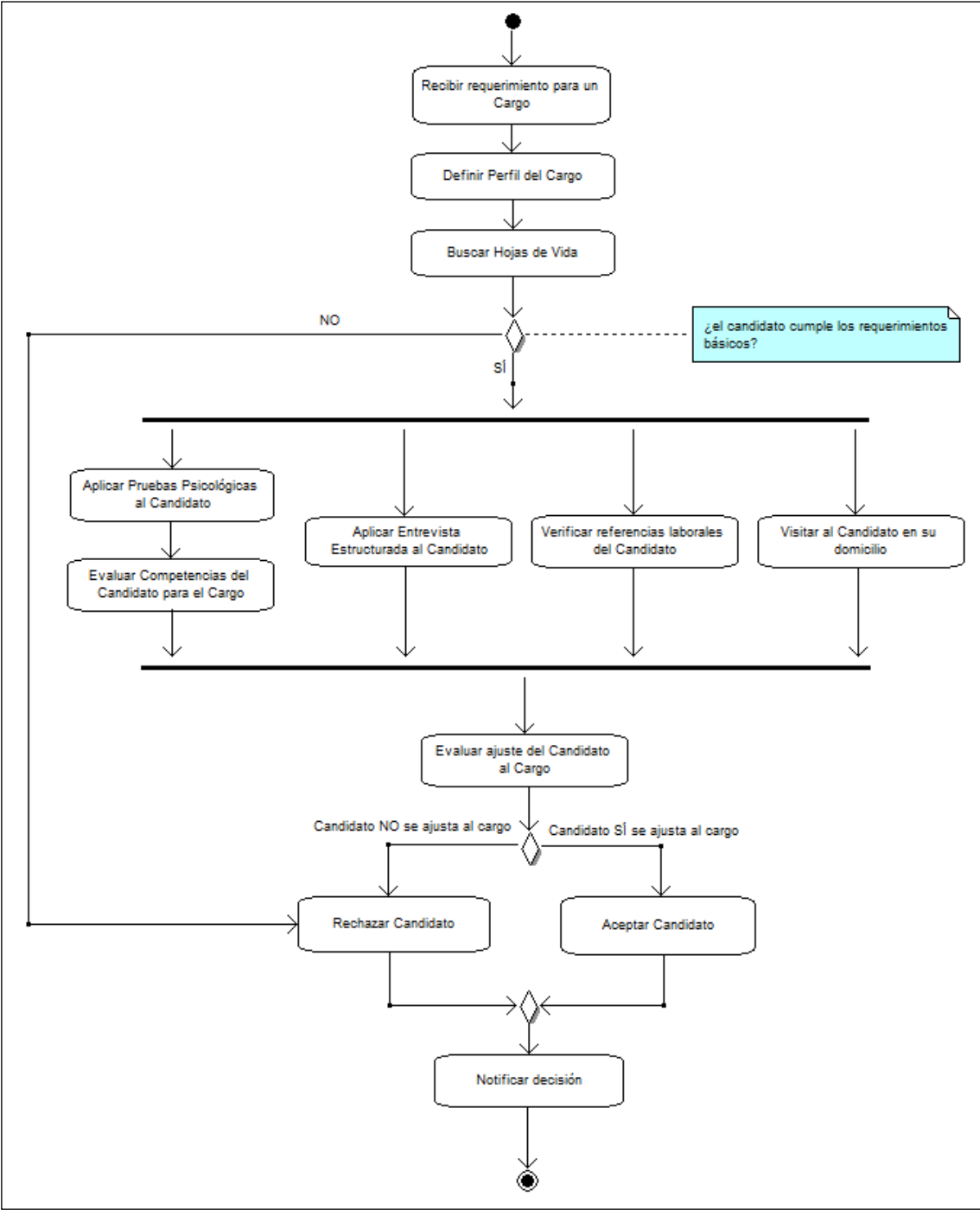
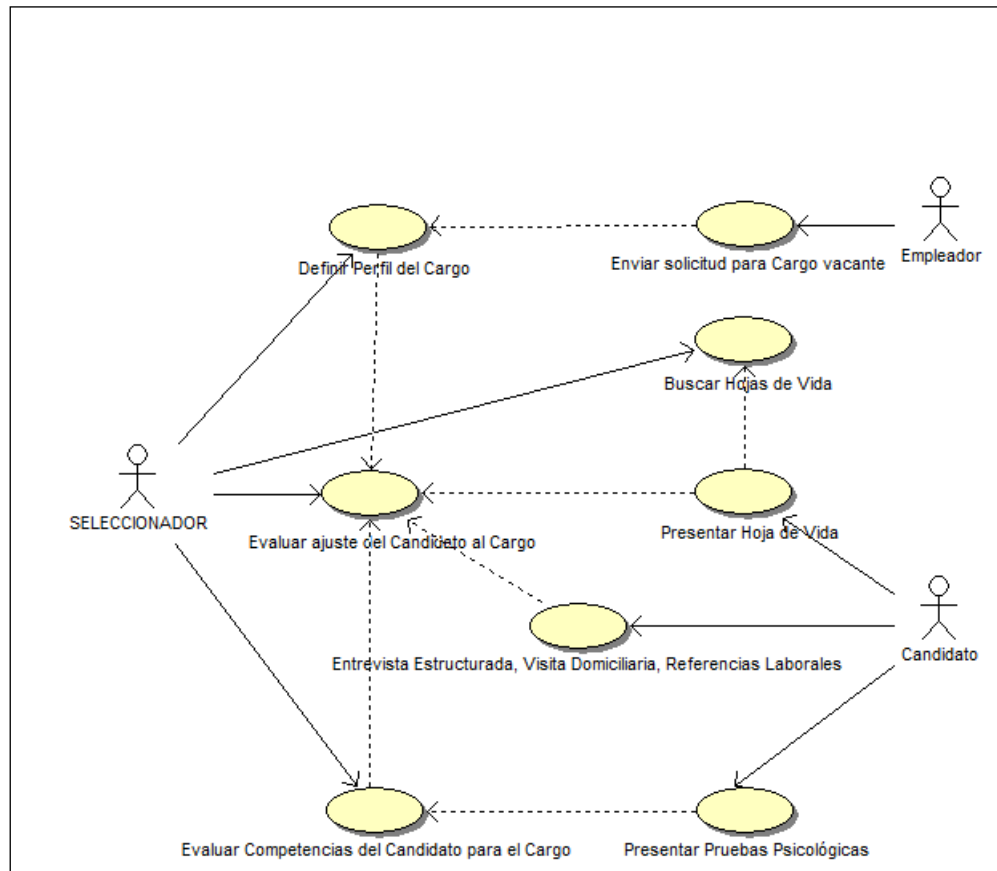


Ilustración 11. Diagrama de casos de uso (UML) en el proceso de Selección de Personal



6.2.3 OM-3: Descripción del proceso desde el punto de vista de las tareas en que está compuesto y sus características principales

En la Tabla 14 se detallan las tareas del proceso, haciendo énfasis en las tareas que serán relevantes en el proyecto.

Es importante mencionar que, de todo el proceso, la tarea a ser intervenida mediante este sistema es la de Evaluación de las Competencias del Candidato. Sin embargo, se considera que se debe hacer énfasis en los procesos de selección que evalúan **Competencias Personales**. Este tipo de proceso es muy extendido y, adicionalmente, mediante este sistema se le podrá hacer asequible a pequeñas y medianas empresas. Por lo tanto, de ahora en adelante se considerará que en esta tarea se hará una **Evaluación de Competencias Personales**.

6.2.4 OM-4: Descripción del componente de conocimiento del modelo de la organización y sus principales características

Tabla 14. Descripción del proceso de Selección de Personal desde el punto de vista de las tareas en que está compuesto.

ID	Nombre	Agente que la ejecuta	Lugar en estructura	Activo de conocimiento	¿Intensiva en conocimiento?	Significancia
REQ	Recibir requerimiento para un Cargo	Seleccionador	Selección de Personal		No	5
PER	Definir perfil del cargo	Empleador, Seleccionador	Selección de Personal, Área del Empleador	Definición formal de cargos mediante perfiles	Sí	5
BUS	Buscar Hojas de Vida	Seleccionador	Selección de Personal		No	4
PRU	Aplicar Pruebas Psicológicas al Candidato	Candidato	Selección de Personal		Sí	4
ECO	<i>Evaluar Competencias Personales del Candidato</i>	<i>Seleccionador</i>	<i>Selección de Personal</i>	<i>Evaluación de Competencias Personales</i>	<i>Sí</i>	<i>5</i>
ENT	Aplicar Entrevista Estructurada al Candidato	Empleador, Seleccionador	Selección de Personal, Área del Empleador	Diseño de Entrevista Estructurada	Sí	4
VER	Verificar Referencias Laborales del Candidato	Seleccionador	Selección de Personal		No	3
VIS	Visitar Candidato en su Domicilio	Seleccionador	Selección de Personal		No	3
AJU	Evaluar ajuste de Candidato a un Cargo	Seleccionador	Selección de Personal	Evaluación del candidato para el cargo	Sí	5
NOT	Aceptar / rechazar Candidato, y notificar	Seleccionador	Selección de Personal		No	5

Significancia se mide en escala del 1 al 5.
Las tareas relevantes en este proyecto están resaltadas en negrilla cursiva.

En la Tabla 15 se detalla el conocimiento organizacional relevante en el desarrollo del proyecto.

Tabla 15. Descripción de los activos de conocimiento organizacional para la Selección de Personal.

ID	Activo de conocimiento	Agente que lo posee	Usado en	¿Forma correcta?	¿Lugar correcto?	¿Tiempo correcto?	¿Calidad correcta?
DPC	Definición formal de Cargos mediante perfiles	Seleccionador	PER	Sí, aunque la información se podría manejar en una base de datos	Sí	Sí	Sí
<i>EVA</i>	<i>Evaluación de Competencias Personales</i>	<i>Seleccionador</i>	<i>ECO</i>	<i>No, el Seleccionador debe estudiar las pruebas psicotécnicas de cada candidato, uno por uno</i>	<i>Sí</i>	<i>Sí, pero se podría optimizar el tiempo de entrega</i>	<i>Sí</i>
DEE	Diseño de Entrevista Estructurada	Seleccionador	ENT	No, requiere diseños más formales	Sí	Sí	Sí
ECC	Evaluación del Candidato para el Cargo	Seleccionador	AJU	No, el Seleccionador debe estudiar la evidencia disponible de cada candidato, uno por uno	Sí	Sí, pero se podría optimizar el tiempo de entrega	Sí
Los activos de conocimiento relevantes en este proyecto están resaltados en negrilla cursiva.							

6.2.5 OM-5: lista de chequeo para el documento de viabilidad de la decisión

A continuación se detalla la información de esta plantilla.

▪ **Punto de vista de negocio.** El Seleccionador necesita mucho tiempo para ejecutar las tareas ECO y AJU, pero se ha identificado que es posible y necesario usar las tecnologías existentes para automatizar las tareas parcialmente y hacer más eficiente el trabajo del Seleccionador. Por eso se considera que este proyecto es necesario para poder ofrecer un servicio de Selección de Personal más competitivo, que sea más atractivo para los clientes y al mismo tiempo maximice la productividad del Seleccionador.

No se requieren cambios organizacionales significativos mientras se siga usando la Infraestructura Tecnológica con la que actualmente cuenta Métodos Expertos (ver OM-2), el cual ya está familiarizado con un ambiente de trabajo en la web para la administración aplicaciones y datos. Además Métodos Expertos ya cuenta con un experto técnico en las herramientas que ofrece el ambiente web: PHP y MySQL, mientras que utilizar otras herramientas implica la contratación de nuevo personal o servicio de soporte a las aplicaciones.

Por otro lado se espera tener reconocimiento empresarial y académico pues se está siguiendo una metodología de Ingeniería del Conocimiento avalada que demuestra la aplicabilidad de la Inteligencia Artificial en el quehacer cotidiano de las organizaciones.

Por estas razones se debe dar prioridad a maximizar el aprovechamiento de este ambiente web, de modo que no sea necesario incurrir en más gastos adquiriendo nueva infraestructura.

▪ **Punto de vista técnico.** A criterio del desarrollador del proyecto, el conocimiento del experto puede ser representado mediante descripciones matemáticas y reglas relativamente sencillas, fáciles de codificar y almacenar en cualquier herramienta de desarrollo empleada incluyendo una base de datos relacional.

Aunque inicialmente se pensó en utilizar un lenguaje declarativo para desarrollar el sistema, se hará uso de PHP para desarrollar un Motor de Inferencia prototipo dado que es el lenguaje utilizado para desarrollar las demás aplicaciones para manipular los datos pertinentes en el trabajo de la organización en SP. Se agregarán elementos propios de la programación declarativa (típica de IA) dentro de la programación imperativa de PHP, con el fin de obtener un Motor de Inferencias integrable con las reglas codificadas en la Base de Conocimientos del sistema. Este enfoque híbrido entre programación declarativa e imperativa ha sido utilizado en diversas ocasiones en ámbitos académicos y empresariales con

buenos resultados [13, 37, 38, 39]. Esto es beneficioso en el sentido que acerca más la IA con el día a día de la academia y la empresa, generando nuevas oportunidades para desarrollar productos y servicios innovadores en el sector software.

Las aplicaciones en PHP pueden ser mantenidas por personal que ya está en Métodos Expertos y son fácilmente integrables a las aplicaciones actuales con las que cuenta la organización (entorno de trabajo), como se explicó en B.5.1. Esto facilitará enormemente la integración de esta nueva aplicación en el trabajo cotidiano del Seleccionador de Personal de las empresas.

La Base de Conocimientos será almacenada en una base de datos MySQL, donde también están almacenados los datos de la Base de Hechos. MySQL tiene amplias facilidades para su integración (interoperabilidad) con PHP.

Esto facilitará el desarrollo de la aplicación dentro de un marco de tiempo favorable para el desarrollador, sin riesgos técnicos considerables a tener en cuenta.

6.2.6 TM-1: Descripción refinada de las tareas dentro del proceso objetivo

A continuación se diligencia el formulario TM-1 para la tarea objetivo de este proyecto:

Tabla 16. TM-1 para la Tarea ECO

Tarea	ECO - Evaluar Competencias Personales del Candidato
Organización	Selección de Personal.
Objetivo y valor	Esta tarea es necesaria para establecer en qué Competencias Personales específicas se destaca el candidato, y qué tan aproximadas son a los requerimientos para el Cargo. Como se explicó, las competencias afectan el desempeño de una persona en cualquier cargo.
Dependencias y flujo	Tareas de entrada: PRU - Aplicar Prueba Psicológica a Candidato. Tareas de salida: AJU - Evaluar ajuste de Candidato a un Cargo.
Objetos manejados	Objetos de entrada: Perfil Psicológico del Candidato. Objetos de salida:

Tarea	ECO - Evaluar Competencias Personales del Candidato
	Qué tanto cumple el Candidato con las competencias requeridas para el Cargo.
Tiempo y control	Esta tarea es posible ejecutarla una vez se tiene el Perfil Psicológico del Candidato, obtenido con la tarea PRU. Al terminar esta tarea, el Seleccionador debe obtener una evaluación del candidato que mida qué tanto se acercan sus competencias a las competencias requeridas. Usualmente esta tarea puede tomar al Seleccionador alrededor de 1-2 horas, pero se espera que con el sistema a desarrollar tome 15 minutos.
Agentes	Seleccionador de Personal.
Conocimientos y competencias	EVA - Evaluación de Competencias Personales
Recursos	Esta es una tarea intensiva en conocimiento que requiere el conocimiento del experto en SP y el Perfil Psicológico de cada candidato.
Calidad y desempeño	Se debe recibir una retroalimentación por parte de la empresa contratante del Candidato seleccionado para el cargo que estaba vacante, con el fin de establecer si se hizo una clasificación apropiada de la personalidad del Candidato. Cabe anotar que el Seleccionador puede tener varios años de experiencia que le permiten conocer qué tan acertada fue su predicción sobre la personalidad de los Candidatos y qué tan acertada es la predicción de su ajuste a un cargo.

6.2.7 TM-2: Especificación del conocimiento empleado por una tarea y posibles cuellos de botella y áreas para mejorar

A continuación se diligencia el formulario TM-2 para el activo de conocimiento necesario para ejecutar la tarea especificada en TM-1.

Tabla 17. TM-2 para el activo de conocimiento EVA.

Activo de conocimiento	EVA - Evaluación de Competencias Personales
Agentes	Seleccionador de Personal
Tareas	ECO - Evaluar Competencias Personales del Candidato
Dominio	Selección de Personal

Naturaleza del conocimiento		Cuello de botella a ser mejorado
Formal, riguroso		
Empírico, cuantitativo		
Heurístico, por reglas		
Altamente especializado, dominio específico	x	El conocimiento es amplio pero no está representado en formatos que permitan un uso por parte de personas no expertas. Se busca poder plasmar la experticia del Seleccionador de Personal en fórmulas matemáticas y reglas que permitan evaluar la Personalidad de un candidato de forma automatizada.
Basado en la experiencia	x	
Basado en la acción	x	
Incompleto		
Incierto, podría ser incorrecto		
Cambia rápidamente		
Difícil de verificar		
Tácito, difícil de transferir		
Forma del conocimiento		
Mente	x	Se busca poder plasmar el conocimiento de forma electrónica con el fin de poder almacenarlo como activo organizacional.
Papel		
Electrónica		
Competencia de acción	x	El sistema podrá ser usado para simular el razonamiento de un experto con amplia experiencia ejecutando evaluación de competencias, por lo que daría soporte al experto para evaluar competencias. No siempre se podrá prescindir del experto completamente.
Otra		
Disponibilidad del conocimiento		
Limitaciones en tiempo		
Limitaciones en espacio		
Limitaciones en acceso	x	Depende de la presencia del experto.
Limitaciones en calidad		
Limitaciones en forma		

6.2.8. AM-1: Especificaciones del agente

Se detalla el formulario AM-1 para el único agente involucrado en las tareas intensivas en conocimiento que son objeto de análisis en este proyecto.

Tabla 18. AM-1 para el Agente Seleccionador de Personal

Nombre	Seleccionador de Personal
Organización	Selección de Personal. Descripción de Perfiles de Cargos.
Tareas	PER - Definir Perfil del Cargo. ECO - Evaluar Competencias Personales del Candidato. VER - Verificar Referencias Laborales del Candidato. VIS - Visitar Candidato en su Domicilio. AJU - Evaluar ajuste de candidato a un cargo. NOT - Aceptar / rechazar candidato, y notificar.
Comunicaciones	N/A.
Conocimientos	DPC - Definición formal de Cargos mediante perfiles. EVA - Evaluación de Competencias Personales. DEE - Diseño de Entrevista Estructurada. EVA - Evaluación del Candidato para el Cargo.
Otras competencias	Profesional en Psicología. Experiencia en Gestión del Recurso Humano.
Responsabilidades y restricciones	El Agente debe tener consideraciones éticas en la búsqueda imparcial de oportunidades laborales para los Candidatos disponibles.
Recursos	Se debe tener la información necesaria sobre los Candidatos y los Cargos para los que serán evaluados.

6.2.9 OTA-1: lista de chequeo para decisiones sobre impacto y mejoras

Se considera que un proyecto de esta naturaleza, orientado a la Selección de Personal, constituye un paso importante para la organización en un mejor aprovechamiento del conocimiento organizacional como activo administrable.

Cabe anotar también que el soporte tecnológico a la tarea CAL sería un gran apoyo a la organización, pues permite reducir el tiempo que toma analizar toda la información psicotécnica de un Candidato. Aunque no se automatiza la tarea completamente, se da un apoyo significativo al Seleccionador de Personal en la toma de sus decisiones.

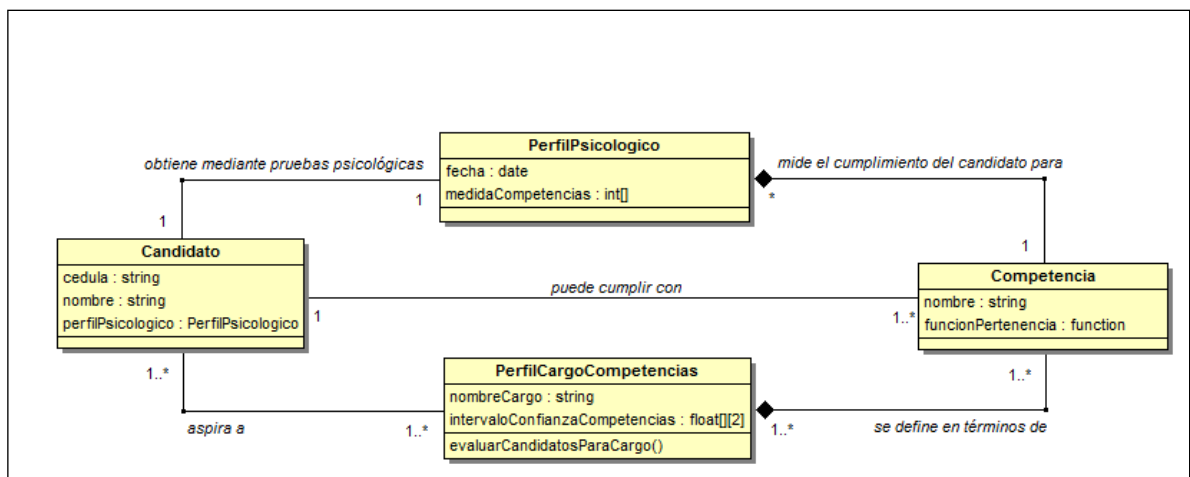
Se recomienda entonces continuar con el proyecto, pero aprovechando la infraestructura tecnológica existente y los aplicativos que ya están siendo usados por la organización, como se detalló en el formulario OM-5.

6.2.10 Modelado del conocimiento para la tarea ECO - Evaluar Competencias Personales del Candidato

Esta es la etapa más crítica del modelado del sistema, pues define en detalle cómo se va a simular el razonamiento del experto.

Es necesario hacer una primera aproximación a los elementos de datos que acostumbra usar el experto en Selección de Personal para poder evaluar las competencias personales de los Candidatos. Para lograr esto se utilizó un diagrama de clases UML.

Ilustración 12. Diagrama de clases UML para la evaluación de competencias personales



A continuación se detalla cómo se interrelacionan estos elementos.

- **El Perfil Psicológico como la medición de las Competencias de un Candidato.** Para cada ser humano se puede medir una serie de Competencias, las cuales en contextos específicos son definidas por un experto en Selección de Personal o Psicología. Cada una de las Competencias representa un aspecto del ser humano que es relevante para el proceso de Selección. Se diseña una Prueba Psicológica para medir esas competencias en cada candidato, de modo que la información obtenida sirva para soportar la toma de decisiones en el proceso de Selección.

Por ejemplo, para tres Candidatos P1, P2 y P3, se puede medir las Competencias C1, C2, C3, C4 y C5. Estas medidas se hacen en una escala de 1 a 100. Así se abstrae el Perfil Psicológico de cada Candidato.

Tabla 19. Ejemplo de Perfiles Psicológicos de tres Candidatos.

Candidato	Flexibilidad (C1)	Sociabilidad (C2)	Audacia (C3)	Conversación (C4)	Curiosidad (C5)
P1	13	55	63	40	89
P2	76	27	48	91	34
P3	26	48	83	77	51

- El Cumplimiento de un Candidato con una Competencia definido como un Conjunto Difuso

Aunque las Competencias son medidas en una escala creciente continua entre 1 y 100, no necesariamente se tiene que las Competencias sean “valoradas” de forma tan uniforme. Aparece el concepto de **Cumplimiento de un Candidato con una Competencia** que se busca ilustrar con dos ejemplos.

Por ejemplo, para la competencia C1, los candidatos P1 y P3 toman valores muy bajos (13 y 26) que un experto podría juzgar como no significativos pues la flexibilidad baja o media no es necesariamente mala al estar evaluando competencias para cargos operativos o rutinarios en una empresa. El experto podría considerar que C1 puede ser significativo al evaluar cargos de dirección estratégica si el Candidato obtuvo una medida de 80 o más, sin importar si se obtuvo cualquier valor entre 80 y 100, pues en general se consideraría que de todas formas el Candidato es una persona “flexible”.

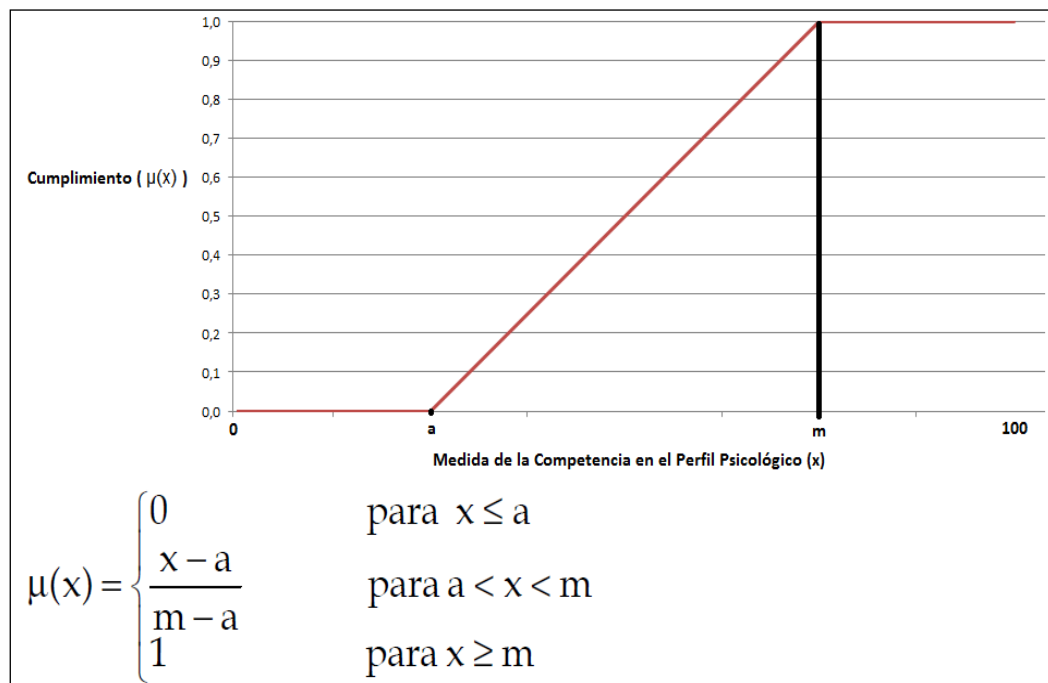
Así mismo, para la competencia C4 se puede considerar que los candidatos P2 y P3 tienen un muy buen nivel pues obtuvieron puntajes bastante altos (91 y 77) de modo que cualquiera de los dos tiene un Cumplimiento total con las necesidades de la organización en cuanto a Habilidades Conversacionales.

En síntesis, la medida de cada Competencia en el Perfil Psicológico no necesariamente determina quienes son menos aptos en habilidades y actitudes para esa Competencia. El criterio para evaluar estas aptitudes ni siquiera es igual para todas las Competencias. Por eso es importante incluir el Cumplimiento en el modelado del conocimiento, pues es un elemento que le da elasticidad al razonamiento del sistema. El gran beneficio obtenido es que el sistema podrá ser adaptado, dependiendo del contexto organizacional donde se vaya a usar.

Se considera que el Cumplimiento de un Candidato con una Competencia puede ser representado mediante una **Función de Pertenencia**, típica de la Lógica Difusa. Cada Competencia tiene asociada su propia función de pertenencia, simulando el hecho que el criterio para determinar el Cumplimiento no es el mismo para todas las Competencias.

El Cumplimiento de todas las Competencias es modelado matemáticamente con la un tipo de función llamada **Función Lambda (Γ)**. La función de pertenencia específica para cada Competencia puede ser entonces parametrizada con los valores de los puntos **a** y **m**, y como resultado se obtiene el Cumplimiento del Candidato para esa Competencia, en un intervalo entre 0.0 y 1.0.

Ilustración 13. Aplicación de la Función Lambda (Γ) para calcular el Cumplimiento de un Candidato, partiendo del puntaje o medida obtenido por el Candidato, para cada una de las Competencias a evaluar.



A continuación se da un ejemplo del procedimiento para hallar el Cumplimiento de un Candidato para cada Competencia, continuando con el caso previamente expuesto para los 3 Candidatos y las 5 Competencias.

Tabla 20. Ejemplo de cómo obtener los parámetros para hallar el Cumplimiento de cada Competencia mediante la Función Lambda.

Competencia	a	m	Explicación
Flexibilidad (C1)	30	80	Cumplimiento nulo para puntajes menores de 30, cumplimiento total para puntajes mayores de 80. Entre 30 y 80 el cumplimiento crece uniformemente con el puntaje.
Sociabilidad (C2)	0	70	No es posible ser completamente no social,. El cumplimiento crece uniformemente hasta un puntaje de 70. A partir de 70 el cumplimiento es total.
Audacia (C3)	0	100	El cumplimiento crece uniformemente con el puntaje.
Conversación (C4)	40	75	Cumplimiento nulo para puntajes menores de 40, cumplimiento total para puntajes mayores de 75. Entre 40 y 75 el cumplimiento crece uniformemente con el puntaje.
Curiosidad (C5)	20	100	Cumplimiento nulo para puntajes menores de 20. Cumplimiento crece uniformemente con el puntaje para puntajes mayores a 20.

Con los puntajes de los Candidatos y las funciones de Cumplimiento para cada Competencia, se obtiene el Cumplimiento de cada Candidato para cada Competencia.

Tabla 21. Ejemplo del cálculo del Cumplimiento de cada Candidato para cada Competencia.

Candidato	Flexibilidad (C1)	Sociabilidad (C2)	Audacia (C3)	Conversación (C4)	Curiosidad (C5)
P1	0.0	0.8	0.6	0.0	0.9
P2	0.9	0.4	0.5	1.0	0.2
P3	0.0	0.7	0.8	1.0	0.4

Hasta ahora se ha obtenido un Cumplimiento de cada Candidato para cada Competencia. Esto da al experto en Selección de Personal la posibilidad de asignar o quitar relevancia a los valores extremos (muy altos y muy bajos) en los puntajes obtenidos por los Candidatos para cada Competencia.

A continuación se explica cómo se relaciona el Cumplimiento con el Perfil del Cargo.

- El Perfil del Cargo definido en términos de las Competencias de un Candidato ideal para ese Cargo

Se debe definir el Perfil del Cargo con el fin de conocer en términos concretos qué se desea de la persona que ocupe el Cargo. Para evaluar las Competencias en el proceso de Selección, se hace necesario definir el Cargo en términos de un

Candidato con las Competencias ideales. Es decir, para la evaluación de competencias, el Perfil del Cargo se define en términos de un Perfil Psicológico especial, que corresponde a un **Candidato Ideal** para ese Cargo.

Esta definición del Candidato Ideal se hace a criterio del experto mediante *Intervalos de Confianza* para el Cumplimiento de cada Competencia. No siempre se requiere un Cumplimiento exacto con una Competencia, es decir que el Cumplimiento no puede ser un valor específico sino que se debe poder tener otro grado de flexibilidad. Por lo tanto se plantea que el Cumplimiento del Candidato Ideal para cada Competencia, puede representarse mediante un *Conjunto Difuso Valorado por Intervalos de Confianza (CDVIC)*. Un ejemplo a continuación.

Tabla 22. Ejemplo de definición de tres Perfiles de Cargo, mediante CDVIC para cada Competencia en cada Cargo

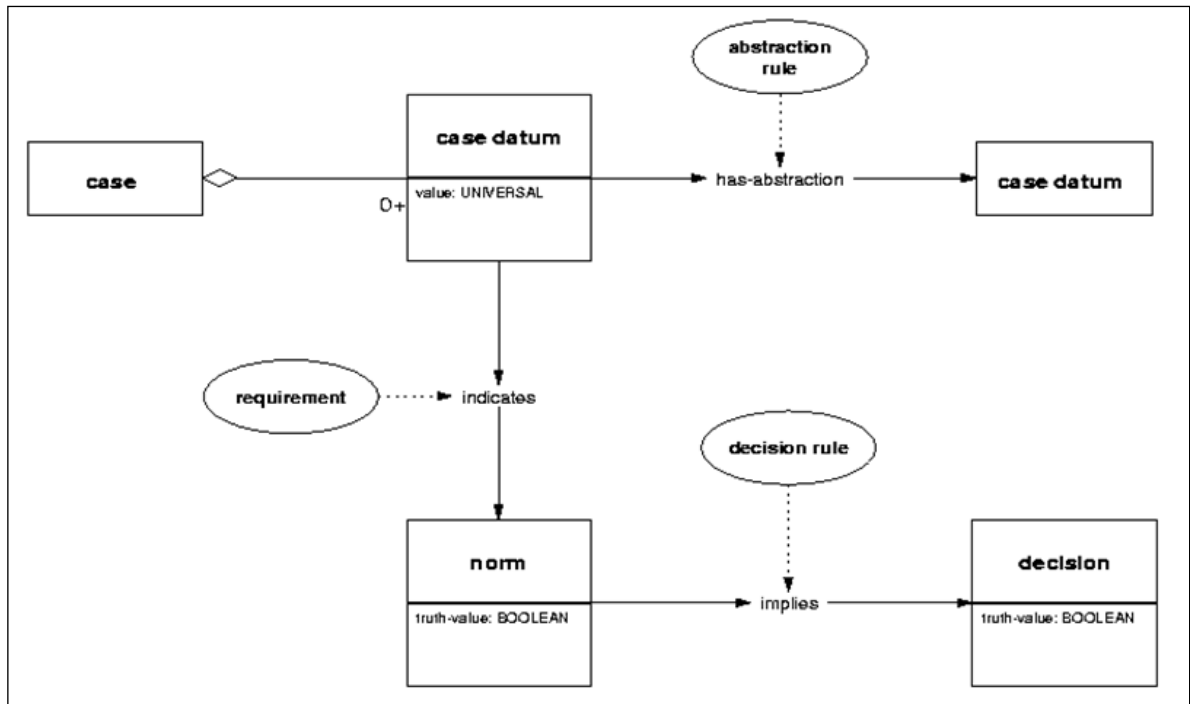
Cargo	Flexibilidad (C1)	Sociabilidad (C2)	Audacia (C3)	Conversación (C4)	Curiosidad (C5)
Programador	[0.6, 0.8]	[0.9, 1.0]	[0.2, 0.5]	[0.1, 0.3]	[0.4, 0.6]
Representante Comercial	[0.5, 0.7]	[0.3, 0.6]	[0.7, 1.0]	[0.7, 1.0]	[0.3, 0.6]
Gerente	[0.8, 1.0]	[0.5, 0.8]	[0.6, 0.9]	[0.5, 0.7]	[0.5, 1.0]

Así se ha obtenido una definición de cada Perfil del Cargo, para poder compararlo con las Competencias de cada Candidato a ser evaluado.

- **Esquema del dominio.** Para modelar la ejecución de la tarea ECO mediante un método computable, se ha elegido la plantilla de CommonKADS para la EVALUACIÓN (ASSESSMENT). A continuación se procede a detallar la tarea ECO mediante la plantilla mencionada.

El esquema del dominio puede explicarse mediante el siguiente gráfico que provee la plantilla de Evaluación de CommonKADS.

Ilustración 14. Esquema del dominio para la tarea ECO



Un **Caso** es el individuo o elemento que va a ser evaluado por el sistema. Los **Datos del Caso** deben ser abstraídos mediante una **Regla de Abstracción**, de modo que tengan un formato computable por el sistema. En este trabajo la Regla de Abstracción es cualquier mecanismo que permita obtener esos datos.

Se necesita un **Requerimiento** para evaluar los Datos del Caso, de modo que el Caso pueda ser evaluado. Para evaluar el Caso se debe seleccionar una **Norma**. Puede existir una o varias Normas, pero se debe seleccionar una para la evaluación. Esta Norma llevará a una **Decisión** mediante una **Regla de Decisión**.

Para la tarea ECO se identifican los siguientes conceptos:

- El Caso es cada Candidato con sus Competencias, que ha de ser evaluado para saber si se selecciona para un Cargo o no.
- La Regla de Abstracción se define en los mecanismos de la Prueba Psicológica presentada por el Candidato y la estrategia para hallar el Cumplimiento de los Candidatos para cada Competencia.
- Los Datos del Caso son lo que hemos llamado Perfil Psicológico y el Cumplimiento de cada Candidato para cada Competencia. Es importante

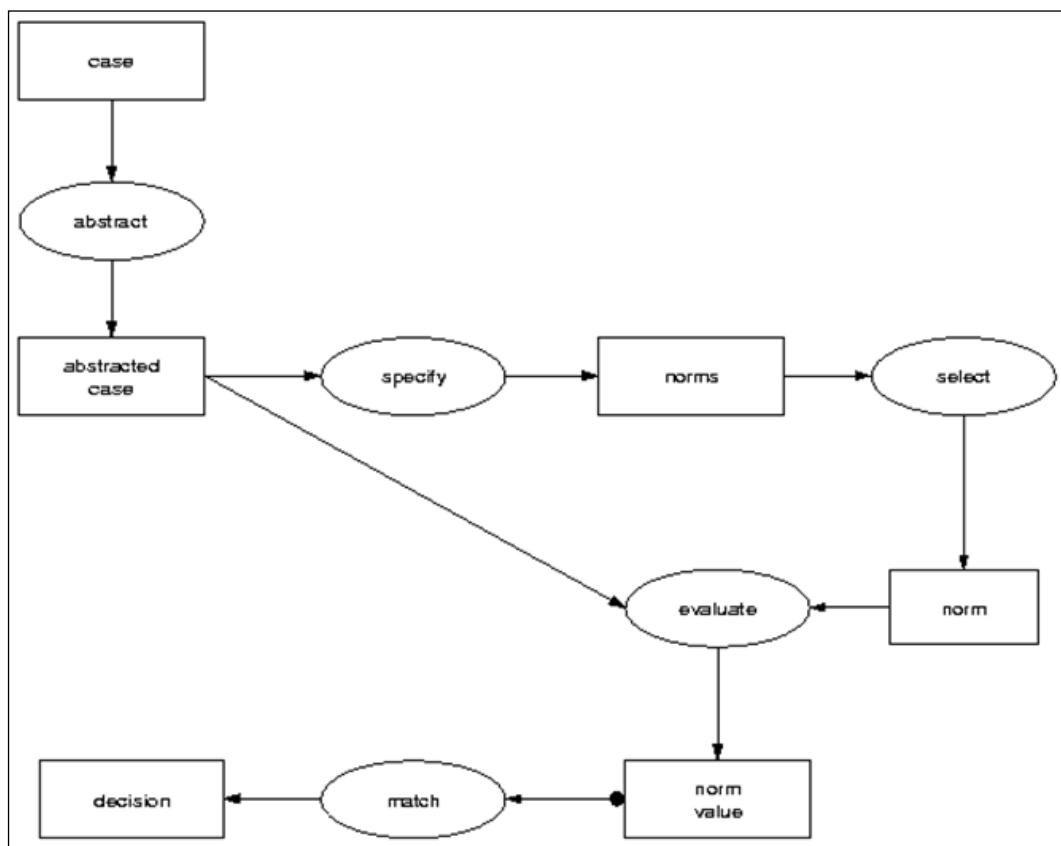
destacar que antes de la ejecución de la tarea ECO, lo más importante es el Cumplimiento de cada Candidato para cada Competencia.

- El Requerimiento ocurre cuando el usuario quiere evaluar las Competencias del Candidato, respecto al Perfil del Cargo.
- Roles estáticos: Competencias y las funciones de pertenencia de cada una; Perfiles de Cargos.
- Roles dinámicos: Perfiles Psicológicos de los Candidatos.

Queda pendiente por definir la Norma y la Regla de Decisión a aplicar para poder llegar a una decisión sobre la elegibilidad del Candidato. Esto se hará en el conocimiento de inferencia.

▪ **Conocimiento de Inferencia.** A continuación se describen los pasos necesarios en la inferencia para evaluar las competencias de un Candidato en un proceso de Selección de Personal, con ayuda de la plantilla de Evaluación de CommonKADS.

Ilustración 15. Conocimiento de inferencia para la tarea ECO.



1. **Abstraer el Caso:** El caso se *abstrae* mediante una definición del Perfil Psicológico en términos de datos, como se explicó previamente. Se parte de un Caso, para obtener un Caso Abstraído que son los datos a procesar en la tarea.
2. **Especificar Normas:** Se evalúan todos los Candidatos para un Perfil del Cargo específico, con una misma Norma para asegurar la imparcialidad en la decisión sobre la evaluación de competencias.
3. **Seleccionar Norma:** No se requiere este paso, pues solo se usará una Norma predeterminada para evaluar todos los Candidatos respecto al Cargo.
4. **Evaluar Norma:** Se determinó que la Norma a aplicar debe calcular la **Distancia** entre el Cumplimiento del Candidato para cada Competencia, y el Perfil del Cargo definido en términos de Competencias. Es decir que se va a buscar qué tanto difieren el Perfil del Cargo y el set de Competencias del Candidato y su set de Competencias.
5. **Encontrar solución:** El Candidato con la menor Distancia puede ser el seleccionado para el Cargo, desde el punto de vista de la evaluación por competencias. Si dos o más Candidatos tienen la menor Distancia, cualquiera de ellos puede ser seleccionado para el Cargo.

La Distancia será calculada mediante una función que se apoya en la función matemática conocida como **Distancia de Hamming Normalizada (DHN)** usando la **Métrica de Hausdorff** [57]. La DHN entre dos CDVIC, A y B, se define de la siguiente manera:

Sean

A = { [Intervalo confianza 1], [Intervalo confianza 2], ..., [Intervalo confianza n] }

= { **MA(xi) | i = 1,2,...,n** }

B = { [Intervalo confianza 1], [Intervalo confianza 2], ..., [Intervalo confianza n] }

= { **MB(xi) | i = 1,2,...,n** }

Ilustración 16. Distancia de Hamming Normalizada entre dos CDVIC, mediante la Métrica de Hausdorff.

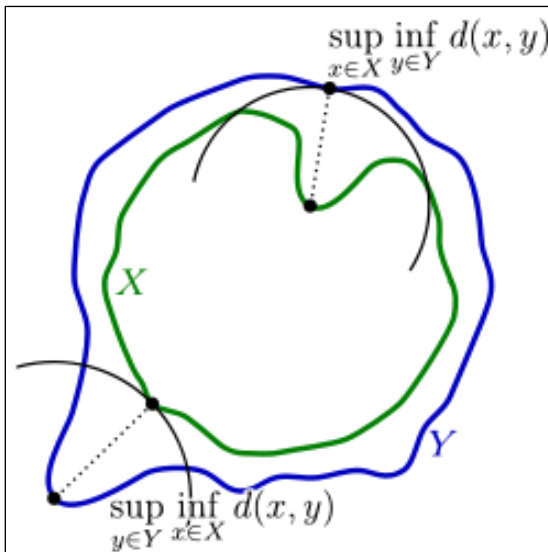
the Hamming distance $d_h(A, B)$:

$$d_h(A, B) = \sum_{i=1}^n \max\{|M_{AL}(x_i) - M_{BL}(x_i)|, |M_{AU}(x_i) - M_{BU}(x_i)|\},$$

the normalized Hamming distance $l_h(A, B)$:

$$l_h(A, B) = \frac{1}{n} \sum_{i=1}^n \max\{|M_{AL}(x_i) - M_{BL}(x_i)|, |M_{AU}(x_i) - M_{BU}(x_i)|\},$$

Ilustración 17. Aplicación de la Métrica de Hausdorff para determinar la distancia entre las formas de dos objetos X y Y.



Se usó este método para medir la distancia, pues se basa en un concepto matemático que busca medir qué tan “distantes” son dos objetos, analizando numéricamente la “forma” de ambos. La Métrica de Hausdorff determina la distancia para cada Competencia entre los objetos Candidato y Perfil del Cargo, mientras que la Distancia de Hamming integra todas las distancias de todas las Competencias.

A continuación se expone un ejemplo para calcular la Distancia entre el Cargo de Gerente, y cada uno de los Candidatos P1, P2 y P3. El Cumplimiento de un Candidato con una Competencia es un número escalar, no es un intervalo, por lo que será convertido en intervalo. Si el valor es 0.6, entonces su intervalo correspondiente es [0.6, 0.6].

Tabla 23. CDVIC correspondiente al Perfil del Cargo de Gerente

Cargo	Flexibilidad (C1)	Sociabilidad (C2)	Audacia (C3)	Conversación (C4)	Curiosidad (C5)
Gerente	[0.8, 1.0]	[0.5, 0.8]	[0.6, 0.9]	[0.5, 0.7]	[0.5, 1.0]

Tabla 24. Tres CDVIC correspondientes a los Cumplimientos para cinco Competencias

Candidato	Flexibilidad (C1)	Sociabilidad (C2)	Audacia (C3)	Conversación (C4)	Curiosidad (C5)
P1	[0.0, 0.0]	[0.8, 0.8]	[0.6, 0.6]	[0.0, 0.0]	[0.9, 0.9]
P2	[0.9, 0.9]	[0.4, 0.4]	[0.5, 0.5]	[1.0, 1.0]	[0.2, 0.2]
P3	[0.0, 0.0]	[0.7, 0.7]	[0.8, 0.8]	[1.0, 1.0]	[0.4, 0.4]

Tabla 25. Cálculo de la Distancia entre cada uno de los tres Candidatos y el Cargo de Gerente, mediante la Distancia Normalizada de Hamming usando la Métrica de Hausdorff.

Candidato	Distancia entre el Candidato y el Cargo de Gerente
P1	0.54
P2	0.44
P3	0.50

En el caso expuesto el Candidato seleccionado como Gerente sería P2.

- **Conocimiento de tarea.** El conocimiento de tarea se describe mediante la plantilla que provee CommonKADS en la plantilla de la Evaluación.

Ilustración 18. Conocimiento de tarea para ECO.

```

while new-solution abstract(case -> abstracted-feature) do
    case := abstracted-feature union case;
end while
specify(case -> norms);
repeat
    select(norms -> norm);
    evaluate(case + norm -> norm-value);
    evaluation-results := norm-value union evaluation-results;
until has-solution match(evaluation-results -> decision);

```

Antes de calcular una nueva solución, se debe agregar los datos que definen el Caso mediante la abstracción de cada uno de los Datos del Caso. Luego se especifican las Normas y selecciona una de ellas, aunque en este proyecto solo se usará una Norma, como ya se había mencionado. El Caso se evalúa para la Norma establecida, obteniendo un valor concreto (la Distancia). Se calcula la solución estableciendo qué Candidato obtuvo la menor Distancia.

▪ **KM-1: Lista de comprobación de la documentación del modelo del conocimiento**

Se presenta el formulario KM-1 para el modelado del conocimiento de la tarea ECO.

- **Modelo de conocimientos y elicitación de los requisitos de conocimiento:** descrito en las páginas anteriores.
- **Fuentes de información:** Álvaro Jaramillo, [45, 46, 47, 48, 49, 51, 52, 57].
- **Componentes del modelo que fueron considerados para reutilización:** Plantilla para la tarea analítica Evaluación. Esta plantilla la provee CommonKADS.
- **Escenarios para solucionar el problema objeto de la aplicación:** Evaluación de competencias en un proceso de Selección de Personal.
- **Resultado de las simulaciones hechas durante la validación del conocimiento:** en las páginas anteriores hay un ejemplo, pero se puede hallar Casos de Prueba completos en el ANEXO C.

Con esto se concluye el modelado del conocimiento para la tarea ECO.

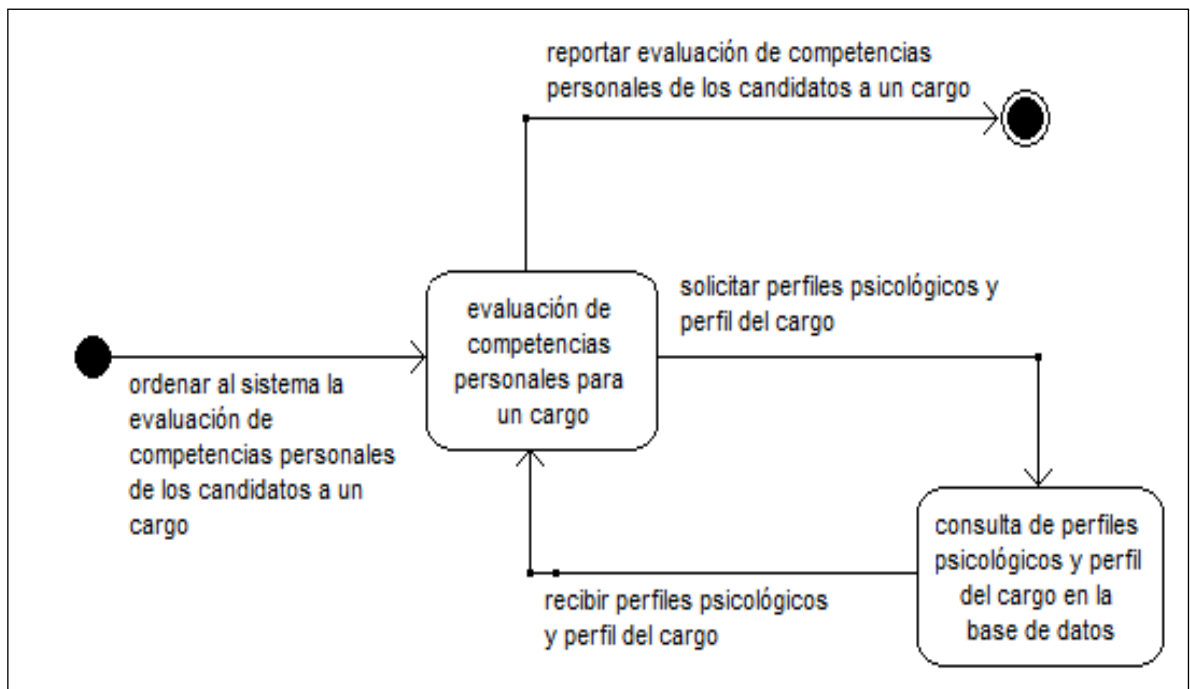
6.2.11 Modelo de comunicaciones

A continuación se describen los elementos a tener en cuenta para el Modelo de Comunicaciones para la ejecución de la tarea ECO.

El Ingeniero de Conocimiento actualiza la Base de Conocimientos mediante operaciones de lectura, creación, actualización y borrado (CRUD) para los Perfiles de los Cargos y los Perfiles Psicológicos de cada Candidato. No se hará modelado de comunicaciones para estas operaciones dado que son tareas no intensivas en conocimiento que son rutinarias en cualquier sistema de software.

- **Plan de comunicaciones.** El Plan de comunicaciones inicia su ejecución cuando el agente Seleccionador genera la orden de evaluar las competencias personales de los Candidatos para un Cargo específico. La evaluación necesita el Perfil Psicológico de cada Candidato y el Perfil del Cargo definido como un CDVIC en todas las Competencias evaluadas. Ambos ítems de información deben ser solicitados en la base de datos. Luego se procede a ejecutar la evaluación según lo expuesto en el modelado del conocimiento. El sistema reporta el conocimiento para soportar la Selección de Personal desde la tarea ECO.

Ilustración 19. Plan de comunicaciones representado en un Diagrama de estados UML.



En el Plan de comunicaciones, se considerará que el Sistema y la Base de datos son dos agentes que participan en las transacciones junto al agente Seleccionador.

- **CM-1: Especificación de las transacciones que posibilitan el diálogo entre Agentes**

A continuación se especifican todas las transacciones mencionadas en el Plan de comunicaciones.

Tabla 26. CM-1: Transacción para ordenar la evaluación de competencias personales para un cargo.

ID y nombre	ORDER – ordenar al sistema la evaluación de competencias personales de los candidatos a un cargo.
Objeto de información	Requerimiento enviado al sistema desde el navegador del usuario.
Agentes involucrados	Seleccionador (envía), Sistema (recibe).
Plan de comunicaciones	Ver Plan de comunicaciones definido previamente.
Restricciones	Pre-condiciones: Conexión a Internet debe estar disponible. Post-condiciones: Inicia ejecución de evaluación de competencias personales.
Especificación de Intercambio de Información	La solicitud debe incluir un identificador del Perfil del Cargo, de modo que sea posible encontrarlo en la base de datos. No se requiere llenar CM-2.

Tabla 27. CM-1: Transacción para solicitar a la base de datos los perfiles psicológicos de los candidatos y el perfil del cargo.

ID y nombre	REQUEST – solicitar perfiles psicológicos y perfil del cargo.
Objeto de información	Solicitud enviada por la aplicación a la base de datos incluyendo identificador del perfil del cargo.
Agentes involucrados	Sistema (envía), Base de datos (recibe).
Plan de comunicaciones	Ver Plan de comunicaciones definido previamente.
Restricciones	Pre-condiciones: Se debe tener el identificador del perfil del cargo, con el fin de poder buscarlo en la base de datos. Post-condiciones: La consulta a la base de datos genera datos de perfiles psicológicos de los candidatos y perfil del cargo.
Especificación de Intercambio de Información	Se genera una consulta SQL que busca el identificador del perfil del cargo y los perfiles psicológicos de los candidatos. No se requiere llenar CM-2.

Tabla 28. CM-1: Transacción para obtener el perfil del cargo y los perfiles psicológicos de los candidatos desde la base de datos

ID y nombre	OBTAIN – recibir perfiles psicológicos y perfil del cargo.
Objeto de información	Perfil Psicotécnico del Candidato.
Agentes involucrados	Base de datos (envía), Sistema (recibe).
Plan de comunicaciones	Ver Plan de comunicaciones definido previamente.
Restricciones	Pre-condiciones: La búsqueda del perfil del cargo, basado en un identificador, debe retornar un resultado. Post-condiciones: Se obtiene datos del perfil del cargo y los perfiles psicológicos de los candidatos.
Especificación de Intercambio de Información	La base de datos envía todos los registros correspondientes a los perfiles psicológicos de los candidatos y el perfil del cargo. No se requiere llenar CM-2.

Tabla 29. CM-1: Transacción para reportar los resultados arrojados por el sistema al Seleccionador

ID y nombre	REPORT – reportar evaluación de competencias personales de los candidatos a un cargo.
Objeto de información	Evaluación de competencias según lo detallado en el modelado del conocimiento.
Agentes involucrados	Sistema (envía), Seleccionador (recibe).
Plan de comunicaciones	Ver Plan de comunicaciones definido previamente.
Restricciones	Pre-condiciones: Se tiene perfil del cargo, perfiles psicológicos de los candidatos, y base de conocimientos con los registros necesarios. Post-condiciones: El reporte se despliega al Seleccionador.
Especificación de Intercambio de Información	Se reporta evaluación de competencias para el cargo especificado, con la explicación correspondiente. No se requiere llenar CM-2.

Mediante el formulario CM-1 se han especificado las transacciones necesarias para desplegar los resultados de la evaluación de competencias. Las operaciones CRUD en la base de datos no han sido modeladas pues son operaciones básicas del software, como se había mencionado.

- **CM-2: Especificación de los mensajes y los puntos de información que hacen una transacción individual**

Este formulario no ha sido necesario para especificar detalles de las transacciones, por lo que no será diligenciado.

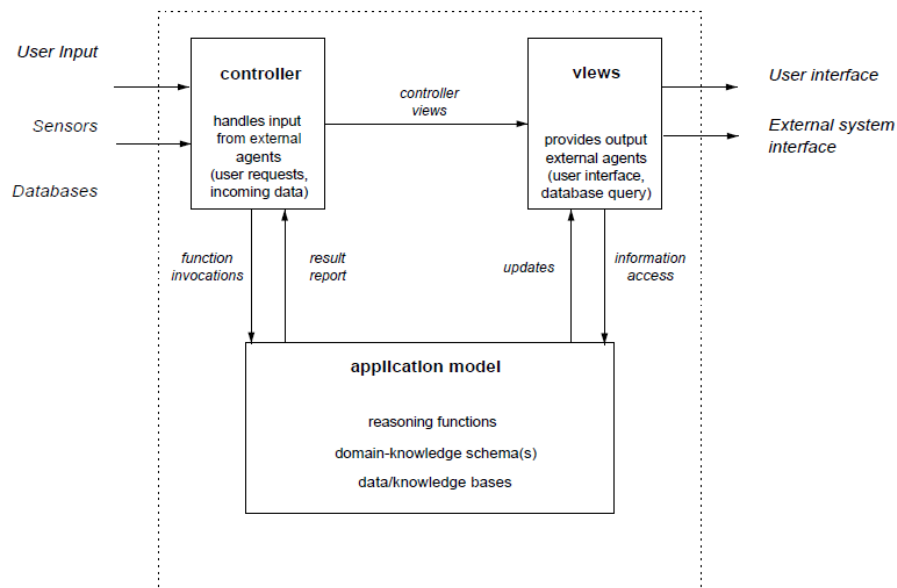
6.2.12 Modelo de diseño

El Modelo de Diseño se detalla a continuación.

- **DM-1: Descripción de la arquitectura del sistema**

El sistema se basa en la arquitectura conocida como Modelo-Vista-Controlador (MVC). A continuación una representación que provee CommonKADS para esta arquitectura.

Ilustración 20. Descripción del MVC según CommonKADS.

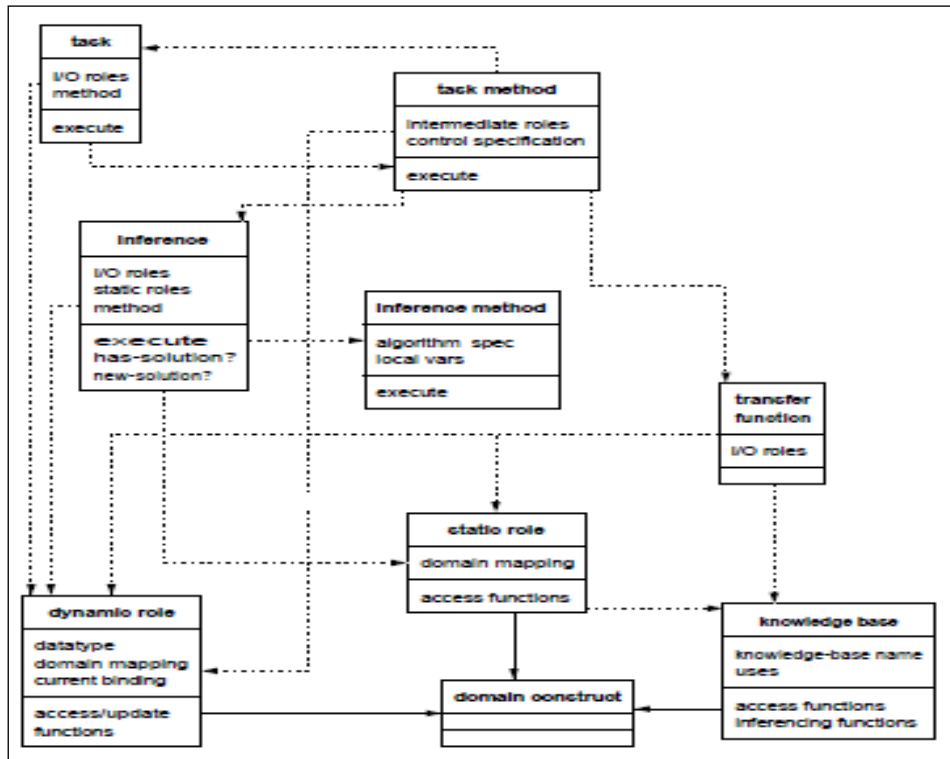


Este es el formulario DM-1, que describe con más detalle los aspectos relevantes de esta arquitectura para este proyecto.

Tabla 30. DM-1: Descripción de la arquitectura del sistema.

Decisión	Descripción
Estructura de subsistemas	La arquitectura Modelo-Vista-Controlador (MVC).
Modelo de control	En el sistema cada Controlador tendrá asociadas sus propias vistas para desplegar la interfaz con el usuario. El usuario maneja el sistema mediante el envío de requerimiento encriptado en una URL (web). El control del sistema se hace respondiendo a las solicitudes del usuario.
Descomposición de subsistemas	En la figura que aparece abajo se representa el modelado del conocimiento en la arquitectura, dividiéndolo en distintos subsistemas. Las líneas punteadas representan rutas en los mecanismos de invocación.

Ilustración 21. Subsistemas a implementar en el MVC, representando los elementos del modelado de conocimientos.



Nótese que se ha introducido un nuevo elemento llamado *inference method* para describir algorítmicamente el conocimiento de inferencia.

- **DM-2: Especificación de las facilidades ofrecidas por y en el sistema que será implantado**

En DM-2 se detalla la plataforma donde será instalada la aplicación a desarrollar.

Tabla 31. DM-2: Especificación de las facilidades ofrecidas por y en el sistema que será implantado.

Ambiente de Software	El sistema puede operar en cualquier plataforma de hosting web que cumpla con las siguientes características: Linux, kernel versión 2.6.18-194.17.1.el5. Apache, versión 2.2.19. PHP, versión 5.2.17. MySQL, versión 5.1.56. Este ambiente lo proveen muchas empresas de hosting web.
Hardware potencial	Cualquier tipo de servidor que soporte el ambiente de software.
Hardware seleccionado	Servidor de hosting que tiene la empresa proveedora del hosting web.
Librería de vistas	Se hará uso de las facilidades que proveen Zend Framework y Dojo Toolkit para el despliegue de interfaces web.
Tipado del lenguaje de programación	PHP es un lenguaje débilmente tipado, es decir que no es estricto en la definición de tipos de datos para las variables.
Representación del conocimiento	PHP es un lenguaje imperativo, pero provee muy buenas facilidades para implementar el modelado del razonamiento mediante funcionalidades matemáticas.
Protocolos de interacción con sistemas externos	Para este prototipo no será necesario que el sistema sea interoperable con otros sistemas. Sin embargo a futuro se buscará que otros sistemas alimenten la base de datos de perfiles de los candidatos.
Flujo de control	Se usará solicitudes del usuario mediante URL y formularios HTML. El sistema internamente operará según lo descrito en DM-1.
Soporte a CommonKADS	PHP no da soporte a CommonKADS.

Se define el entorno tecnológico donde se desplegará la aplicación.

- DM-3: Lista de chequeo de decisiones en relación con la especificación de la arquitectura y DM-4: Decisiones del diseño de la aplicación

En DM-3 y DM-4 se definen con mayor detalle los componentes de la arquitectura del sistema.

A continuación se listan los controladores y las diferentes vistas en el sistema.

Tabla 32. Lista de controladores y las acciones y vistas asociados a cada uno.

Controlador	Acción / Vista
IndexController Controlador para el despliegue de la arquitectura de la interfaz y vistas de apoyo del sistema.	index Despliegue de la arquitectura de la interfaz y los links principales.
	inicio Introducción al sistema.
	ayuda Información relevante sobre el sistema.
AtributosController Administración de las Competencias.	principal Despliegue de lista de competencias y opciones para crear, editar y borrar.
	form Despliegue de formulario para crear o editar competencias.
	save Guardado de los datos de la competencia, ingresados por el usuario en form.
	delete Borrar una competencia.
	datoslista Datos a mostrar en la lista de competencias en la vista <i>principal</i> .
CandidatosController Administración de los candidatos y sus perfiles psicológicos.	principal Despliegue de lista de candidatos y opciones para crear, editar y borrar.
	form Despliegue de formulario para crear o editar candidatos.
	save Guardado de los datos de los candidatos,

Controlador	Acción / Vista
	ingresados por el usuario en form. delete Borrar un candidato. datoslista Datos a mostrar en la lista de candidatos en la vista <i>principal</i> .
CargosController Administración de los perfiles de los cargos.	principal Despliegue de lista de cargos y opciones para crear, editar y borrar. form Despliegue de formulario para crear o editar cargos. save Guardado de los datos de los cargos, ingresados por el usuario en form. delete Borrar un cargo. datoslista Datos a mostrar en la lista de cargos en la vista <i>principal</i> .
EvaluacionesController Generar evaluación de competencias para un cargo específico.	seleccionarcargo Selección del cargo para el cual se va a generar la evaluación de competencias de los candidatos. Aquí se inicializa la tarea de evaluación de competencias (eco). eco Despliegue de los resultados de la evaluación de competencias. Aquí están contenidos la inferencia y el método de inferencia. ecolista Datos de los resultados de la evaluación de competencias para cada candidato, a ser mostrados en la vista <i>eco</i> .

Los roles estáticos son el modelo Competencias y el modelo Perfiles de los Cargos (modelos dentro de la arquitectura MVC). Ellos componen la Base de Conocimientos. El modelo Candidatos es un rol dinámico, pues durante la ejecución de la tarea ECO se agrega información relevante sobre cada candidato aunque no se ingrese a la base de datos MySQL.

Las vistas pueden ser consultadas en el Anexo B.

6.3 ELECCIÓN DE UN LENGUAJE DE PROGRAMACIÓN DECLARATIVA

Ya se explicó que el lenguaje de programación a usar será PHP, en combinación con una base de datos MySQL. Sobre la programación imperativa de PHP es posible implementar aspectos de la programación declarativa típica de IA, obteniendo resultados exitosos a nivel académico y empresarial [13, 37].

Sin embargo es válido hacer una consideración de lenguajes representativos en el desarrollo de sistemas de IA, dado que corresponde a uno de los objetivos de este trabajo de grado. Se hará un análisis de tres lenguajes de IA, para decidir brevemente cuál de ellos se usaría para el desarrollo de un SE en un caso hipotético. Se utiliza la metodología propuesta por Lawlis⁷ [32].

6.3.1 Lenguajes a considerar. Se deben descartar lenguajes que soporten los dos requisitos básicos: Encadenamiento hacia delante y Distribución Libre. A continuación se describe brevemente algunos de los lenguajes de IA más conocidos para el desarrollo de SE para entornos empresariales.

- **CLIPS (C Language Integrated Production System):** Herramienta que provee un ambiente de desarrollo para la producción y ejecución de sistemas expertos, creada en 1984 por la NASA, en Estados Unidos.
- **JESS (Java Expert System Shell):** Herramienta basada en CLIPS en el lenguaje Java, para integración con aplicaciones hechas en Java, creada en 1995 por Sandia National Labs, en Estados Unidos.
- **Common LISP:** Common LISP procede de LISP (List Processing Language), este último especificado originalmente en 1958 en el Instituto Tecnológico de Massachusetts. Common LISP es una de las variaciones del LISP que más éxito ha tenido siendo aplicado industrialmente, fuera de entornos académicos.

6.3.2 Selección del Lenguaje de Programación. Los tres lenguajes considerados han demostrado ser bastante útiles a las horas de desarrollar

⁷ LAWLIS, Patricia. *Guidelines for Choosing a Computer Language: Support for the Visionary Organization*. Ada Information Clearinghouse. 1997.

aplicaciones de IA y cumplen satisfactoriamente con la mayoría de los requerimientos de este proyecto. Sin embargo hay información importante que debe ser considerada para la selección hipotética del lenguaje a usar. Parte de esta selección también es basada en el juicio que da la experiencia del desarrollador.

Tabla 33. Comparación de lenguajes de programación declarativa para Inteligencia Artificial

	CLIPS	JESS	Common LISP
Facilidad de integración	Se integra a cualquier aplicación en C/C++, también tiene protocolos para integrarse a aplicaciones en otros lenguajes.	Se integra a cualquier aplicación Java, también tiene protocolos para integrarse a aplicaciones en otros lenguajes.	No es fácil integrarlo a otras tecnologías. Se necesitarían librerías especializadas en otros lenguajes para poder integrarlo.
Portabilidad	Puede ser llevado a cualquier sistema que tenga un compilador ANSI para C/C++ (Windows 95/98/NT, MacOS X, y Unix).	Puede ser llevado a todas las plataformas que soporten la Máquina Virtual de Java, de la que depende completamente.	Puede ser llevado a cualquier sistema que tenga un compilador ANSI para C/C++ (Windows 95/98/NT, MacOS X, y Unix).
Confiabilidad y Seguridad	Alta (Construido en C).	Alta (Construido en Java).	Alta (construido en LISP).
Reusabilidad	Sí	Sí	Sí
Estandarización	Reconocido por la ANSI.	Reconocido por la ANSI.	Reconocido por la ANSI.
Soporte	Documentación completa, y la más grande comunidad de desarrolladores.	Documentación completa, y gran comunidad de desarrolladores.	Documentación completa, y gran comunidad de desarrolladores.

CommonLISP es muy difícil de integrar a otras tecnologías, como es necesario en este caso. CLIPS y JESS presentan mejores facilidades para integrar con aplicaciones desarrolladas en otros lenguajes, como se necesita en este proyecto.

Por otro lado, JESS está basado en Java que es un lenguaje robusto y menos rápido en comparación con C, que es el lenguaje en el que está basado CLIPS. Es importante mencionar además que el desarrollador de la aplicación tiene un mejor conocimiento de C que de Java, lo que le haría más fácil integrar su aplicación con CLIPS.

Después de comparar las ventajas y desventajas de los lenguajes considerados, el lenguaje que cumple con tener encadenamiento hacia delante y ser de distribución libre, además de cumplir satisfactoriamente con los otros parámetros estipulados, es CLIPS. Por lo tanto el prototipo de Sistema Experto sería programado en CLIPS.

6.3.3 Lenguaje seleccionado: Clips. CLIPS es una herramienta que provee un ambiente de desarrollo para la producción y ejecución de sistemas expertos. Fue creado a partir de 1984, en el Lyndon B. Johnson Space Center de la NASA. Los fondos cesaron a principios de los años 1990, y hubo un mandato de la NASA para comprar software comercial.

CLIPS es un acrónimo de C Language Integrated Production System (Sistema de Producción Integrado en Lenguaje C). En la actualidad, entre los paradigmas de programación que soporta CLIPS se encuentran la Programación Lógica, la Programación Imperativa y la Programación Orientada a Objetos.

CLIPS probablemente es la herramienta para Sistemas Expertos más ampliamente usada debido a que es rápido, eficiente y gratuito. Aunque ahora es de dominio público, aún es actualizado y mantenido por su autor original, Gary Riley.

Las características principales de CLIPS son:

- **Representación del Conocimiento:** CLIPS permite manejar una amplia variedad de conocimiento, soportando tres paradigmas de programación: el declarativo, el imperativo, y el orientado a objetos. La programación lógica basada en reglas permite que el conocimiento sea representado como reglas heurísticas que especifican las acciones a ser ejecutadas dada una situación. La POO permite modelar sistemas complejos como componentes modulares. La programación imperativa permite ejecutar algoritmos de la misma manera que en C, Java, LISP y otros lenguajes.
- **Portabilidad:** CLIPS fue escrito en C con el fin de hacerlo más portable y rápido, y ha sido instalado en diversos sistemas operativos (Windows 95/98/NT, MacOS X, Unix) sin ser necesario modificar su código fuente. CLIPS puede ser ejecutado en cualquier sistema con un compilador ANSI de C, o un compilador de C++. El código fuente de CLIPS puede ser modificado en caso

que el usuario lo considere necesario, con el fin de agregar o quitar funcionalidades.

- **Integrabilidad:** CLIPS puede ser embebido en código imperativo, invocado como una sub-rutina, e integrado con lenguajes como C, Java, FORTRAN y otros. CLIPS incorpora un completo lenguaje orientado a objetos (COOL) para la elaboración de sistemas expertos. Aunque está escrito en C, su interfaz más próxima se parece a LISP. Pueden escribirse extensiones a CLIPS sobre C, y al contrario, CLIPS puede ser llamado desde C. CLIPS puede ser extendido por el usuario mediante el uso de protocolos definidos.
- **Desarrollo Interactivo:** La versión estándar de CLIPS provee un ambiente de desarrollo interactivo y basado en texto; este incluye herramientas para la depuración, ayuda en línea, y un editor integrado. Las interfaces de este ambiente tienen menús, editores y ventanas que han sido desarrollados para MacOS, Windows 95/98/NT, X Window, entre otros.
- **Verificación/Validación:** CLIPS contiene funcionalidades que permiten verificar las reglas incluidas en el sistema experto que está siendo desarrollado, incluyendo diseño modular y particionamiento de la Base de Conocimientos del sistema, chequeo de restricciones estático y dinámico para funciones y algunos tipos de datos, y análisis semántico de reglas para prevenir posibles inconsistencias.
- **Documentación:** En la página web oficial de CLIPS se encuentra una extensa documentación que incluye un Manual de Referencia y una Guía del Usuario.
- **Bajo Costo:** CLIPS es un software de dominio público.

7. CONCLUSIONES

La Selección de Personal es claramente uno de los procesos más importantes y críticos para las organizaciones, aunque tiene un alto nivel de incertidumbre y subjetividad dependiendo del contexto donde esté siendo ejecutado. Se espera que el sistema desarrollado mediante este trabajo pueda soportar el proceso en las organizaciones.

Es difícil modelar todo el razonamiento humano involucrado en la Selección de Personal, por lo que se concentraron los esfuerzos en el soporte a la toma de decisiones en la Evaluación de Competencias. Un modelado integral de la Evaluación de Competencias también puede ser difícil de obtener, pero se ha logrado establecer modelos matemáticos para simular el razonamiento de un experto. Este “razonamiento matemático” es, sin embargo, una aproximación que puede ser mejorada con nuevos elementos que flexibilicen el diseño del razonamiento del sistema, haciéndolo más parecido a la riqueza del razonamiento humano. De todas formas la Lógica Difusa fue un recurso muy importante para poder abstraer el razonamiento humano y llevarlo a algoritmos computables.

Aunque no se cuenta con la posibilidad inmediata de crear una máquina que demuestre inteligencia en cualquier área, es posible desarrollar aplicaciones para implementar computacionalmente razonamientos específicos. Es necesario continuar con los estudios que buscan comprender el cerebro humano desde un punto de vista sistémico que integre lo biológico, lo neurológico y lo psicológico. Así se podría llegar al diseño y desarrollo de un cerebro artificial que demuestre inteligencia igual o superior a la humana.

En todo caso, para hacer una aproximación al desarrollo de cualquier Sistema Experto, es muy importante comprender claramente los procesos y las tareas involucradas, así como los datos y recursos tecnológicos con los que se cuenta en la organización. Por eso ha sido muy importante usar la metodología CommonKADS.

También es muy importante conocer más modelos matemáticos que puedan ser adaptados para modelar el razonamiento de un experto. La fidelidad en la simulación de la mente del experto depende de qué tan preciso o conveniente es el modelo matemático elegido para implementar el razonamiento a nivel computacional.

Un objetivo era desarrollar el sistema mediante un lenguaje declarativo de programación, pero se determinó que lo mejor era hacerlo mediante el lenguaje PHP. Las razones principales son que el hosting web para Computación en la Nube es ofrecido a precios muy económicos y de alta calidad, y que ya se cuenta con programadores expertos en PHP. El razonamiento se puede llevar a cabo de forma imperativa, por lo que finalmente se consideró que la programación declarativa no era lo más práctico. Más que aferrarse a una idea de hacer las cosas, en este proyecto se dio prioridad a poder hacer de este sistema un recurso disponible para las organizaciones.

Desde el enfoque empresarial de este proyecto, se ha encontrado que la Computación en la Nube y la Inteligencia Artificial marcan dos tendencias que serán clave en los modelos de negocios para proveer servicios a nivel mundial con apoyo tecnológico. La Computación en la Nube aporta las herramientas para poder masificar el acceso a la tecnología informática. La Inteligencia Artificial ayuda a comprender y modelar los procesos organizacionales y razonamiento de los expertos, de modo que ese conocimiento pueda ser usado para sacar provecho de los datos e información disponible.

El conocimiento adquirido en el desarrollo de este trabajo puede ser aplicado en cualquier área de la economía. Sin embargo el autor del proyecto espera poder aplicar el conocimiento en Bioinformática e Informática Biomédica, dada la importancia que tienen en el mejoramiento de la calidad de vida de las personas.

BIBLIOGRAFÍA

- [1] BUCHANAN, B. G. *History of Artificial Intelligence*. American Association for Artificial Intelligence. 2005.
- [2] REINGOLD, E. *The Turing Test*. University of Toronto. 1999.
- [3] RUSSELL, Stuart J.; NORVIG, Peter. *Artificial Intelligence: A Modern Approach*. Pearson Education. 2003.
- [4] ENGLEMORE, R. S.; FEIGENBAUM, E. *Knowledge Based Systems in Japan*. Japanese Technology Evaluation Center. 1993.
- [5] ZALDIVAR, R.W.; MARRERO, C.; ORTEGA, L. *La Gestión del Personal de Apoyo en las Organizaciones*. Contribuciones a la Economía. 2009.
- [6] CABRERA, L.; GÁLVEZ, J. Capítulo Dos: *Reclutamiento y selección de personal*. Gestión de Personal. Suis. 2009.
- [7] GERSHENSON, C. *Filosofía de la Mente e Inteligencia Artificial*. Universidad Autónoma de México. 2001.
- [8] SOMMERVILLE, Ian. *Artificial Intelligence and Systems Engineering*. University of Saint Andrews. 1994.
- [9] MCCARTHY, John. *What is Artificial Intelligence?* Stanford University. 2004.
- [10] NEISSER, U.; BOODOO, G.; BOUCHARD Jr, T.J.; BOYKIN, A.W.; BRODY, N.; CECI, S.J.; HALPERN, D.F.; LOEHLIN, J.C.; PERLOFF, R.; STERNBERG, R.J. *Intelligence: Knowns and Unknowns. Annual Progress in Child Psychiatry and Child Development*. 1997.
- [11] SAYGIN, Ayse. *The Turing Test Page*. London's Global University. 2003.
- [12] TURING, A. M. Computing machinery and intelligence. *Mind*, 59, 433–460. 1950.
- [13] DUAN, Y.; EDWARDS, J.S. *Web-based Expert Systems: benefits and challenges*. *Information & Management*, 42. 2005.
- [14] GUZMAN, G.; AYUQUINA, M.; CRUZ, W.; ZAMBRANO, D. *Introducción de la Inteligencia Artificial*. Universidad de Guayas. 1998.

- [15] KENDALL, Graham. *Introduction to Artificial Intelligence*. University of Nottingham. 2001.
- [16] SCHMIDHUBER, Jürgen. *Artificial Intelligence. History Highlights and Outlook*. Istituto Dalle Molle di Studi sull'Intelligenza Artificiale. 2003.
- [17] RODRÍGUEZ, A.;HERNÁNDEZ, J.; PLÁCIDO, A. *Ingeniería del Conocimiento: Extracción del Conocimiento*. Facultad de Informática, Universidad de Las Palmas de Gran Canaria. Biblioteca Univesitaria. 2006.
- [18] ISCID. *Branches of Artificial Intelligence*. International Society for Complexity, Information and Design. 2005.
- [19] GUTIERREZ, Miguel. *Introducción a los Sistemas Basados en el Conocimiento*. Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Sevilla. 2006.
- [20] GIARRATANO, J.; RILEY, G. *Sistemas Expertos: Principios y Programación*. Thomson Editores. 2001.
- [21] DARLINGTON, K. *The Essence of Expert Systems*. Prentice Hall, England. 2000.
- [22] JACKSON, P. *Introduction to Expert Systems*. Prentice Hall. 1990.
- [23] ANTKARE, Ike. *An investigation of expert systems with Japer*. In Proceedings of the Workshop on Modular, Metamorphic Technology. 2009.
- [24] FLOYD, RW Floyd. *The Paradigms of Programming*. ACM Turing award lectures. 2007.
- [25] CHEN, W.; KIFER, M.; WARREN, D.S. *HiLog: A Foundation for Higher-order Logic Programming*. Journal of Logic Programming, 15, 187–230. 1993.
- [26] COENEN, F. *Characteristics of Declarative Programming Languages*. 1999.
- [27] PEYTON, S. *The Implementation of Functional Programming Languages*. Prentice Hall. 1987.
- [28] SONDERGAARD, H.; SESTOFT, P. *Referential transparency, definiteness and unfoldability*. Acta Informatica 27, 6, 505–517. 1990.
- [29] PRATT, T. W.; ZELKOWITZ, M. V. *Programming Languages: Design and Implementation*. Prentice Hall. 1996.

- [30] SCHACH, S. *Object-Oriented and Classical Software Engineering*. McGraw-Hill. 2006.
- [31] SCHELL, J. *Chapter Seven: The Game Improves Through Iteration*. The Art of Game Design. Elsevier. 2008.
- [32] LAWLIS, Patricia. *Guidelines for Choosing a Computer Language: Support for the Visionary Organization*. Ada Information Clearinghouse. 1997.
- [33] STELLMAN, A.; GREENE, J. *Applied Software Project Management*. O'Reilly Media. 2005.
- [34] VILLANUEVA, I.; SÁNCHEZ, J. Elicitación de requisitos en sistemas de gestión orientados a procesos. Universidad Politécnica de Valencia, Departamento de Computación. 2005.
- [35] TORO, A.; BERNÁRDEZ, B. Metodología para la Elicitación de Requisitos de Sistemas Software. Universidad de Sevilla, Departamento de Lenguajes y Sistemas Informáticos. 2002.
- [36] PALMA, J.T.; PANIAGUA, E.; MARTÍN, R.; MARTÍN, F. *Ingeniería del Conocimiento: de la Extracción al Modelado del Conocimiento*. Universidad de Murcia, Facultad de Informática. 2000.
- [37] HANUS, M. *Putting declarative programming into the web: Translating Curry to JavaScript*. Proceedings of the 9th ACM SIGPLAN international conference on Principles and practice of declarative programming. 2007.
- [38] ATHAPPILLY, K. *A dynamic web-based knowledge system for prototype development for extended enterprise*. Proceedings of the 3rd International Conference on the Practical Applications of Knowledge Management. Manchester 2000.
- [39] KOENDERICK, N.; BROEKSTRA, J.; TOP, J. *Bounded transparency for automated inspection in agriculture*. Computers and Electronics in Agriculture. Elsevier. 2010.
- [40] SCHREIBER, G.; AKKERMANS, H.; ANJEWIERDEN, A.; DE HOOG, R.; SHADBOLT, N.; VAN DE VELDE, W.; WIELINGA, B. *Knowledge Engineering and Management: The CommonKADS Methodology*. The MIT Press. 2000.
- [41] ANJEWIERDEN, A.; SCHREIBER, G. *CML2 syntax (2.2.1)*. SWI, University of Amsterdam. 1994.

- [42] SILLER, W.; BUCKLEY, J. *Fuzzy Expert Systems and Fuzzy Reasoning*. Wiley. 2005.
- [43] MORALES-LUNA, Guillermo. *Introducción a la Lógica Difusa*. Centro de Investigación y Estudios Avanzados. Instituto Politécnico Nacional. México. 2002.
- [44] POST, W.; WIELINGA, B.; DE HOOG, R.; SCHREIBER, G. *Organizational Modeling in CommonKADS: The Emergency Medical Service*. University of Amsterdam. IEEE Expert. 1997.
- [45] ZADEH, Lotfi A. *Fuzzy Logic*. Department of EECS, Computer Science Division. University of California, Berkeley. 1988.
- [46] ZADEH, Lotfi A. *Fuzzy Sets*. Department of EECS, Computer Science Division. University of California, Berkeley. 1965.
- [47] CANÓS, Lourdes; CASASÚS, Trinidad; LARA, Tomás; LIERN, Vicente; PÉREZ, Juan. *Modelos flexibles de selección de personal basados en la valoración de competencias*. Departamento de Organización de Empresas. Universidad Politécnica de Valencia. España. 2008.
- [48] CANÓS, Lourdes. *Gestión de recursos humanos basada en la lógica borrosa*. Departamento de Organización de Empresas. Universidad Politécnica de Valencia. España. 2005.
- [49] MERIGÓ, José. *Nuevas extensiones a los operadores OWA y su aplicación en los métodos de decisión*. Facultad de Economía y Empresa. Universidad de Barcelona. España. 2008.
- [50] CHIEN, Chen-Fu; CHIEN, Li-Fei. *Data mining to improve Personnel Selection and enhance human capital: A case study in high-technology industry*. Department of Industrial Engineering and Engineering Management. National Tsing-Hua University. China. 2006.
- [51] ZAREI, Hassan; REZA, Mohammad; KARIMI, Mohammad; AZIZOLLAHI, Sirous. *The Application of Fuzzy TOPSIS Approach to Personnel Selection for Padir Company, Iran*. Faculty of Management. University of Tehran. Irán. 2011.
- [52] ARIAS, Yanet; ROSETE, Alejandro; MARTÍNEZ, Raúl. *Propuesta informática para seleccionar personal por competencias utilizando técnicas de Inteligencia Artificial*. Centro de Estudios de Ingeniería de Sistemas. Instituto Superior Politécnico José Antonio Echeverría. Cuba. 2006.

- [53] LIN, Hung-Tso. *Personnel selection using analytic network process and fuzzy data envelopment analysis approaches*. Department of Distribution Management. National Chin-Yi University of Technology. China. 2010.
- [54] DARAMOLA, J. O.; OLADIPUPO, O. O.; MUSA, A. G. *A fuzzy expert system (FES) tool for online personnel recruitments*. Department of Computer and Information Sciences, College of Science and Technology. Covenant University. Nigeria. 2010.
- [55] JANTAN, Hamidah; HAMDAN, Abdul; OTHMAN, Zulaiha. *Intelligent Techniques for Decision Support System in Human Resource Management*. Faculty of Computer and Mathematical Science. Universiti Teknologi MARA, Terengganu. Malasia. 2010.
- [56] KURZWEIL, Raymond. *The Singularity is Near*. Viking Press. New York. 2005.
- [57] PARK, Jin; LIM, Ki; PARK, Jong; KWUN, Young. *Distances between Interval-valued Intuitionistic Fuzzy Sets*. Division of Mathematical Sciences. Pukyong National University. Corea del Sur. 2007.

ANEXOS

Anexo A. ELICITACIÓN DE REQUISITOS

ELICITACIÓN DE REQUISITOS

Versión 1.0

Fecha 19/04/2010

Preparado para:

[MÉTODOS EXPERTOS - GRUPO DE INVESTIGACIÓN Y DESARROLLO](#)

Preparado por:

[MÉTODOS EXPERTOS - GRUPO DE INVESTIGACIÓN Y DESARROLLO](#)

Organización	MÉTODOS EXPERTOS - GRUPO DE INVESTIGACIÓN Y DESARROLLO
Dirección	URL: www.metodosexpertos.com
Teléfono	N/A
Fax	N/A
Comentarios	OBJETO SOCIAL: Desarrollar e implementar proyectos de investigación que busquen aprovechar el conocimiento de las empresas para elevar la productividad y la rentabilidad en sus negocios.
Organización	E-GESTIÓN CONSULTORÍA ORGANIZACIONAL
Dirección	URL: www.egestionconsultoria.com
Teléfono	N/A
Fax	N/A
Comentarios	OBJETO SOCIAL: Acompañar a las empresas en el desarrollo de los procesos de gestión humana brindando rapidez de respuesta y alta calidad en los resultados para aprovechar las competencias del talento humano y contribuir al aumento de la productividad del negocio. Selección de personal

	<p>Descripción de perfiles de cargos Implementación de gestión por competencias Desarrollo de planes de formación Medición de desempeño Medición de clima organizacional Evaluación de potencial Visita domiciliaria Diseño de curva salarial</p>
Participante	ESTEBAN PÉREZ
Organización	<u>MÉTODOS EXPERTOS - GRUPO DE INVESTIGACIÓN Y DESARROLLO</u>
Rol	GERENTE DEL PROYECTO DE SOFTWARE
Es desarrollador	Sí
Es cliente	No
Es usuario	No
Comentarios	<p>TODOS LOS ROLES DE ESTE PARTICIPANTE: Gerente del Proyecto Ingeniero de Conocimiento Ingeniero de Software</p> <p>Candidato al título de Ingeniero de Sistemas de la Universidad EAFIT, con Bases de datos como área de énfasis. Su área de interés principal es la Inteligencia artificial aplicada a la Minería de datos en informática biomédica. Tiene una experiencia de cuatro años en emprendimiento y dos años como investigador en informática biomédica. Se ha desempeñado durante seis años en evaluación, diseño, desarrollo e implementación de tecnologías y bases de datos orientadas a Internet. Durante este tiempo ha trabajado con equipos multidisciplinarios de alrededor de 20 países. Actualmente Esteban es Director de Tecnología en Okorum Technologies, siendo el principal responsable de la vigilancia tecnológica y la administración estratégica de la información.</p>
Participante	ÁLVARO JARAMILLO
Organización	<u>MÉTODOS EXPERTOS - GRUPO DE INVESTIGACIÓN Y DESARROLLO</u>
Rol	EXPERTO EN EVALUACIÓN DE PERSONAL Y PSICOMETRÍA
Es desarrollador	No
Es cliente	Sí

Es usuario	Sí
Comentarios	TODOS LOS ROLES DE ESTE PARTICIPANTE: Administrador de Conocimiento Proveedor de Conocimiento Usuario de Conocimiento Director de Métodos Expertos.
Participante	JULIANA TRUJILLO
Organización	E-GESTIÓN CONSULTORÍA ORGANIZACIONAL
Rol	EXPERTA EN SELECCIÓN DE PERSONAL
Es desarrollador	No
Es cliente	Sí
Es usuario	Sí
Comentarios	TODOS LOS ROLES DE ESTE PARTICIPANTE: Proveedor de Conocimiento Usuario de Conocimiento Ha conducido más de de 4,000 procesos de Selección de Personal. Directora de E-gestión Consultoría Organizacional.
OBJ-0001	Apoyar la evaluación de competencias personales en el proceso de selección de personal
Versión	1.0 (19/10/2010)
Autores	<ul style="list-style-type: none"> • ESTEBAN PÉREZ
Fuentes	<ul style="list-style-type: none"> • ÁLVARO JARAMILLO
Descripción	El sistema deberá <i>apoyar la evaluación de competencias personales en el proceso de Selección de Personal.</i>
Subobjetivos	Ninguno
Importancia	vital
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta
Comentarios	La Organización Internacional del Trabajo OIT (1993) define el concepto de Competencia como “la idoneidad para realizar una tarea o desempeñar un puesto de trabajo eficazmente por poseer las calificaciones requeridas para ello”. Según el Ministerio de

	<p>Educación de Colombia, "las Competencias Laborales Generales son aquellas que se aplican a cualquier clase de trabajo y sector económico. Son las que se utilizan en cualquier espacio laboral y que preparan para cualquier clase de trabajo. Las competencias laborales comprenden todos aquellos conocimientos, habilidades y actitudes que son necesarios para desempeñarse con eficiencia como personas productivas". Una persona es competente cuando ha logrado desarrollar la destreza para el trabajo y las habilidades personales para adaptarse a las exigencias del entorno. El desarrollo de las competencias es un asunto de tiempo y dedicación.</p> <p>Leornad Mertens (1996) diferencia la competencia laboral de la competencia personal. La primera requiere de la aplicación de los aspectos personales como habilidades, aptitudes, actitudes, conocimientos. Estos aspectos permiten la contextualización en el mercado y preparan para la competitividad. "La empresa y la persona son competentes no tanto por cumplir con un desempeño mínimo aceptado sino por tener la capacidad de destacarse en el medio".</p> <p>La competencia personal permite un funcionamiento de la persona en el contexto profesional con características como perseverancia, flexibilidad, autonomía, responsabilidad, etc. que la hacen apto para un eficiente desempeño. Esta cualidad de la competencia personal es considerada por el Análisis Ocupacional y Funcional del Trabajo (IBERPROF, Organización de Estados Iberoamericanos, 2000) cuando plantea: "Competente: persona que posee un repertorio de habilidades, conocimientos y destrezas y la capacidad para aplicarlas en una variedad de contextos".</p>
NFR-0001	Costos y licencias de herramientas tecnológicas utilizadas en el desarrollo
Versión	1.0 (19/04/2010)
Autores	<ul style="list-style-type: none"> • ÁLVARO JARAMILLO
Fuentes	<ul style="list-style-type: none"> • ÁLVARO JARAMILLO
Dependencias	Ninguno
Descripción	<p>El sistema deberá <i>ser desarrollado usando únicamente con herramientas de libre distribución, con el fin de reducir los costos asociados al proyecto. Estas herramientas deben estar instaladas en el ambiente de hosting web del cliente que provee</i></p>

	<i>LunarPages.com. Este ambiente trae MySQL como motor de bases de datos y PHP como lenguaje para programación web.</i>
Importancia	vital
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta
Comentarios	Ninguno
NFR-0002	Marco organizacional para el uso del software
Versión	1.0 (19/04/2010)
Autores	<ul style="list-style-type: none"> • ÁLVARO JARAMILLO
Fuentes	<ul style="list-style-type: none"> • ÁLVARO JARAMILLO
Dependencias	Ninguno
Descripción	<i>El sistema deberá ser usado en el proceso de Selección de Personal en una organización cualquiera. Deberá ser administrado por el seleccionador de personal encargado.</i>
Importancia	vital
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta
Comentarios	Ninguno
FRQ-0001	Especificación del cumplimiento de un candidato con los requerimientos de un cargo
Versión	1.0 (19/04/2010)
Autores	<ul style="list-style-type: none"> • ESTEBAN PÉREZ
Fuentes	<ul style="list-style-type: none"> • ÁLVARO JARAMILLO
Dependencias	<input type="checkbox"/> [FRQ-0003] Modificación de reglas para determinar el cumplimiento de un candidato con las competencias requeridas para un cargo <input type="checkbox"/> [IRQ-0001] Descripción de las competencias personales <input type="checkbox"/> [IRQ-0002] Descripción de los cargos en términos de las competencias del candidato ideal <input type="checkbox"/> [FRQ-0002] Especificación de competencias personales <input type="checkbox"/> [IRQ-0003] Especificación de la prueba psicotécnica de cada

	candidato
Descripción	<p>El sistema deberá <i>determinar qué tanto cumple o tiene el candidato una serie de competencias personales específicas, a ser determinadas por el Seleccionador de Personal, que le permiten ajustarse a las características de un Cargo.</i></p> <p><i>No se puede afirmar que un candidato no cumple o cumple estrictamente para el cargo, sino que el sistema debe razonar para saber "qué tanto" cumple el candidato con los requerimientos del cargo.</i></p> <p><i>Cada decisión arrojada por el sistema debe ser explicada y/o justificada al usuario.</i></p>
Importancia	vital
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta
Comentarios	Ninguno
FRQ-0002	Especificación de competencias personales
Versión	1.0 (19/04/2010)
Autores	<ul style="list-style-type: none"> • ESTEBAN PÉREZ
Fuentes	<ul style="list-style-type: none"> • ÁLVARO JARAMILLO
Dependencias	<input type="checkbox"/> [FRQ-0003] Modificación de reglas para determinar el cumplimiento de un candidato con las competencias requeridas para un cargo <input type="checkbox"/> [IRQ-0002] Descripción de los cargos en términos de las competencias del candidato ideal
Descripción	<p>El sistema deberá <i>permitir la especificación de las competencias personales a ser tenidas en cuenta en la evaluación de competencias, las cuales serán medidas para cada candidato mediante una o varias pruebas psicológicas. Para cada cargo se debe definir las competencias personales necesarias y los valores que estas deben tomar.</i></p>
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	alta

Comentarios	Ninguno
FRQ-0003	Modificación de reglas para determinar el cumplimiento de un candidato con las competencias requeridas para un cargo
Versión	1.0 (19/04/2010)
Autores	<ul style="list-style-type: none"> • ESTEBAN PÉREZ
Fuentes	<ul style="list-style-type: none"> • ÁLVARO JARAMILLO
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir al usuario modificar las reglas y/o mecanismos con las que el sistema evalúa las competencias personales de los candidatos, que lleva a la selección del candidato más acorde a las necesidades de un cargo.</i>
Importancia	importante
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta
Comentarios	Las reglas no necesariamente son del tipo SI-ENTONCES. Principalmente se busca que las “reglas” ingresadas por el usuario soporten el modelo matemático que emula el razonamiento del experto.
IRQ-0001	Descripción de las competencias personales
Versión	1.0 (19/10/2010)
Autores	<ul style="list-style-type: none"> • ESTEBAN PÉREZ
Fuentes	<ul style="list-style-type: none"> • ÁLVARO JARAMILLO
Dependencias	<input type="checkbox"/> [IRQ-0002] Descripción de los cargos en términos de las competencias del candidato ideal <input type="checkbox"/> [IRQ-0003] Especificación de la prueba psicotécnica de cada candidato
Descripción	El sistema deberá almacenar la información correspondiente a <i>las competencias personales para cada candidato. La medida de cada competencia puede ser obtenida mediante pruebas psicológicas determinadas por el experto. En el contexto de este sistema, una prueba psicológica para cada candidato es la medida de cada uno de las competencias en una escala de 0 a 100.</i>
Datos	Ninguno

específicos		
Tiempo de vida	Medio	Máximo
	2 año(s)	5 año(s)
Ocurrencias simultáneas	Medio	Máximo
	30	50
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	
IRQ-0002	Descripción de los cargos en términos de las competencias del candidato ideal	
Versión	1.0 (19/04/2010)	
Autores	<ul style="list-style-type: none"> • ESTEBAN PÉREZ 	
Fuentes	<ul style="list-style-type: none"> • ÁLVARO JARAMILLO 	
Dependencias	Ninguno	
Descripción	<p>El sistema deberá almacenar la información correspondiente a <i>la descripción de cada cargo para el que se va a hacer la selección, en términos de los requerimientos de competencias personales.</i></p> <p><i>Normalmente esta descripción del cargo, en términos de las competencias personales que debe tener el candidato ideal, proviene de la descripción formal del Perfil del Cargo.</i></p>	
Datos específicos	Ninguno	
Tiempo de vida	Medio	Máximo
	2 año(s)	5 año(s)
Ocurrencias simultáneas	Medio	Máximo
	50	100
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

IRQ-0003	Especificación de la prueba psicológica de cada candidato					
Versión	1.0 (19/04/2010)					
Autores	<ul style="list-style-type: none"> • ESTEBAN PÉREZ 					
Fuentes	<ul style="list-style-type: none"> • ÁLVARO JARAMILLO 					
Dependencias	<input type="checkbox"/> [FRQ-0001] Especificación del cumplimiento de un candidato con los requerimientos de un cargo.					
Descripción	El sistema deberá almacenar la información correspondiente a <i>la prueba psicológica de cada candidato. Una prueba psicológica será el valor que tome cada competencia personal, en una escala de 0 a 100.</i>					
Datos específicos	Ninguno					
Tiempo de vida	Medio					Máximo
	6 mes(es)					1 año(s)
Ocurrencias simultáneas	Medio					Máximo
	200					10000
Importancia	vital					
Urgencia	inmediatamente					
Estado	validado					
Estabilidad	alta					
Comentarios	Ninguno					
TRM-0001	FRQ-0001	FRQ-0002	FRQ-0003	IRQ-0001	IRQ-0002	IRQ-0003
OBJ-0001			-	-	-	-
FRQ-0001	-					
FRQ-0002	-	-		-		-
FRQ-0003	-	-	-	-	-	-
IRQ-0001	-	-	-	-		
IRQ-0002	-	-	-	-	-	-
IRQ-0003		-	-	-	-	-

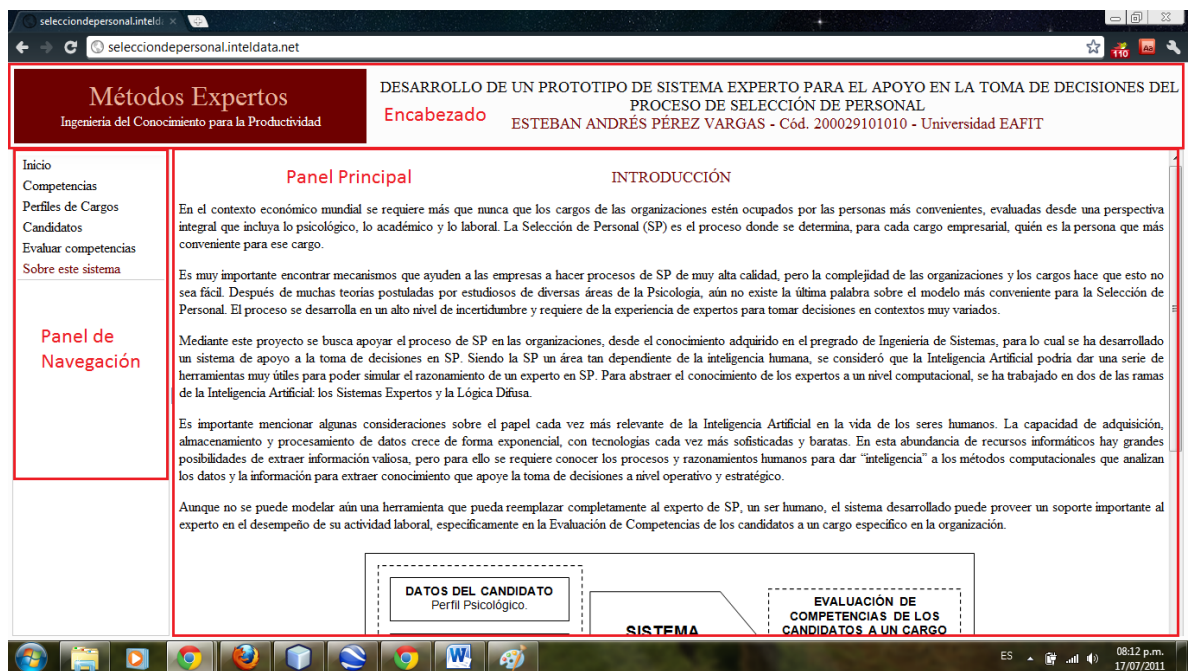
Matriz de rastreabilidad: Rastreo de requisitos funcionales y de información frente a los demás elementos de la elicitación

Anexo B. Interfaces del Sistema Desarrollado

Al acceder el sistema en un browser o navegador de Internet, el usuario puede ver una interface con tres secciones claramente diferenciadas:

- **Encabezado:** Se muestra información general y títulos. Su contenido nunca va a cambiar.
- **Panel de Navegación:** Se muestran los links a los diferentes elementos de datos importantes en el sistema. Al hacer click en cualquiera de esos links se abrirá un contenido en el Panel Principal. Su contenido nunca va a cambiar.
- **Panel Principal:** Se muestra los contenidos y los widgets más importantes con los que debe interactuar el usuario. El contenido varía dependiendo del comportamiento del usuario (qué link selecciona) en el Panel de Navegación.

Ilustración 22. Secciones de la interfaz del sistema.



El primer contenido que aparece es una Introducción al proyecto desarrollado, que se puede acceder siempre desde el link Inicio en el Panel de Navegación. Hay información de ayuda en el link Sobre este sistema.

B.1 Administración de Competencias

Al hacer click en Competencias, en el Panel de Navegación, se despliega un listado de las competencias creadas, y se da la opción de crear una nueva competencia, o editar o borrar una competencia ya creada.



The screenshot displays the 'Métodos Expertos' web application interface. The header features the title 'Métodos Expertos' and the subtitle 'Ingeniería del Conocimiento para la Productividad'. The main content area is titled 'COMPETENCIAS' and includes a 'Crear competencia' button. Below this is a table listing various competencies with their descriptions and numerical values for 'a' and 'm'.

Opciones	Nombre	Descripción	a	m
Editar Borrar	ASERTIVIDAD	Define qué tan asertiva es la persona en su comunicación con los demás.	10	80
Editar Borrar	ASUNCIÓN DE RESGOS		30	75
Editar Borrar	AUDACIA		25	75
Editar Borrar	AUTOESTIMA	Qué tanto se estima la persona.	30	80
Editar Borrar	CURIOSIDAD		5	90
Editar Borrar	EXTROVERSIÓN		10	85
Editar Borrar	FLEXIBILIDAD		50	80
Editar Borrar	HABILIDADES CONVERSACIONALES	Es la capacidad de conversación de la persona.	40	65
Editar Borrar	NORMATIVIDAD		5	95
Editar Borrar	SOCIABILIDAD		30	80

Ilustración 23. Despliegue de la administración de competencias en el Panel Principal.

Si el usuario decide crear o editar una competencia, aparece un cuadro de diálogo para que pueda ingresar los datos correspondientes y guardarlos.

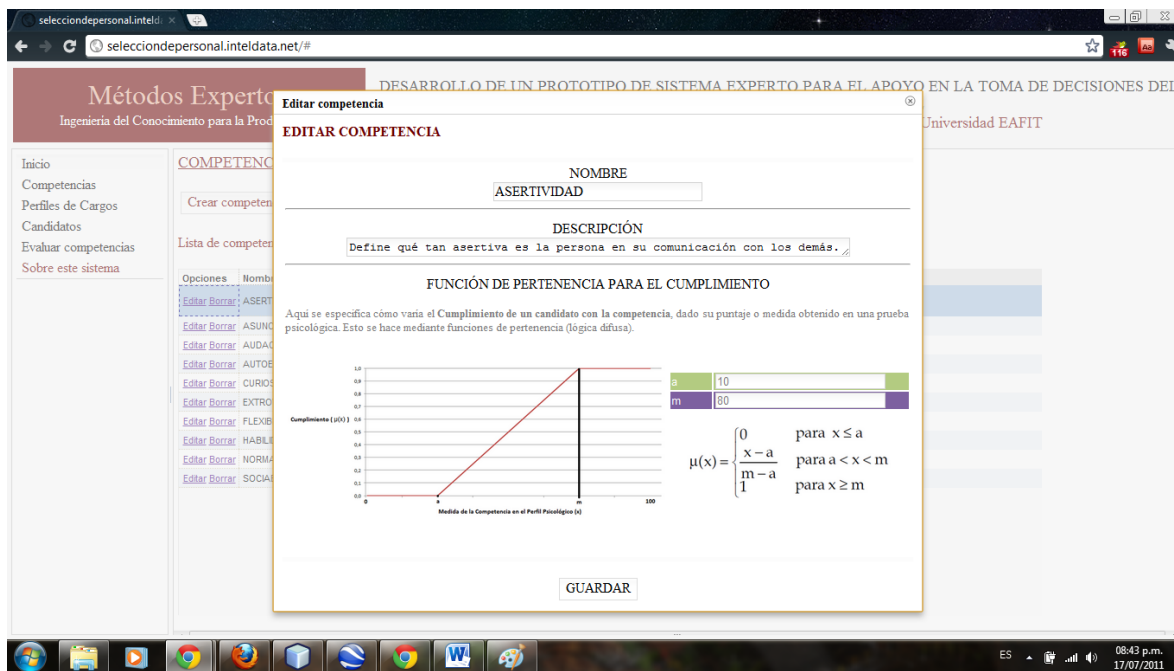


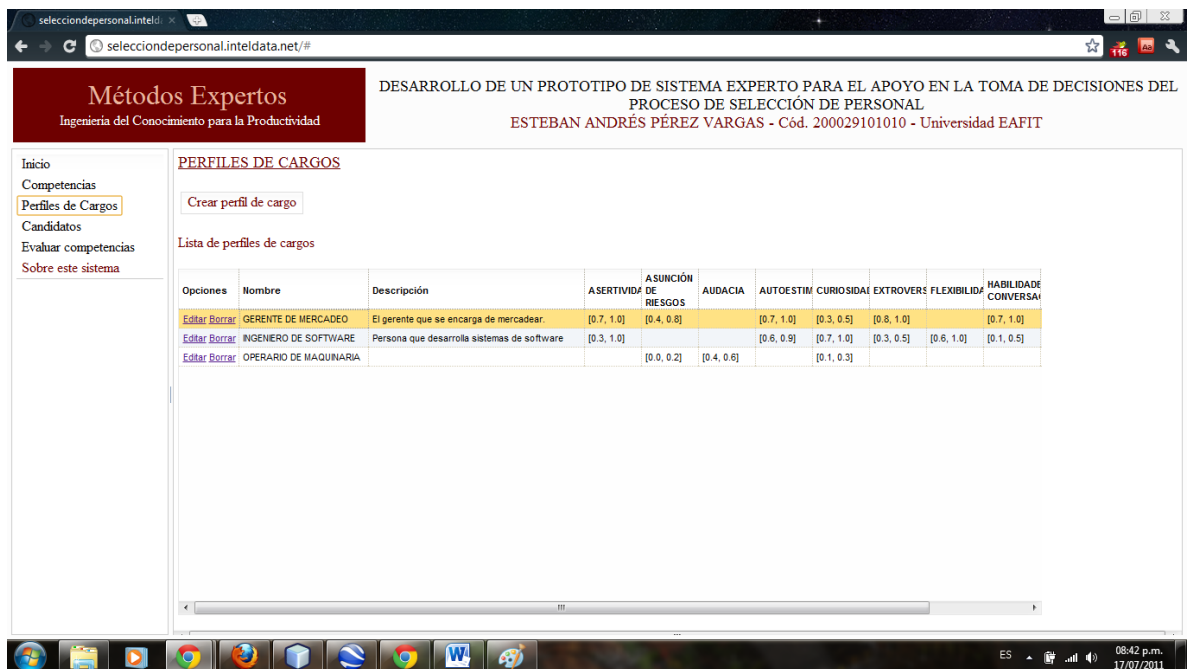
Ilustración 24. Cuadro de diálogo para crear y editar competencias.

Al hacer click en el botón GUARDAR, se procesa el guardado de los datos y se genera una notificación de éxito o fracaso de la operación.

B.2 Administración de Perfiles de Cargos

Al hacer click en Perfiles de Cargos, en el Panel de Navegación, se despliega un listado de los perfiles creados, y se da la opción de crear un nuevo perfil de cargo, o editar o borrar un perfil de cargo ya creado.

Ilustración 25. Despliegue de la administración de perfiles de cargos en el Panel Principal.

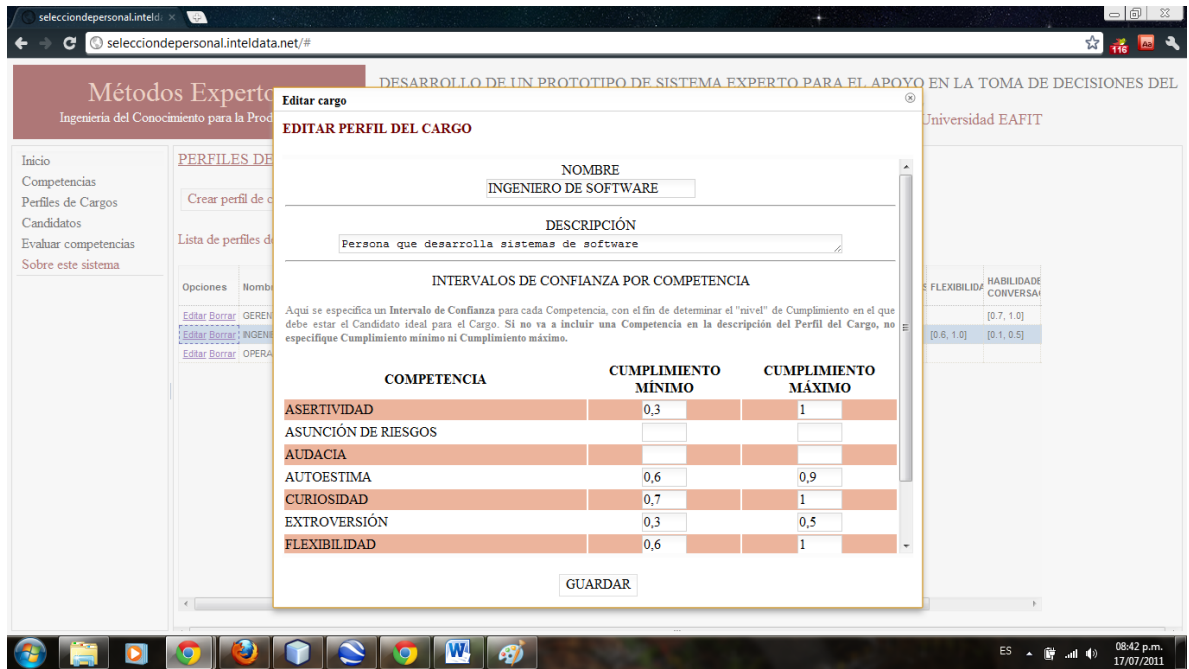


The screenshot shows a web browser window with the URL 'selecciondepersonal.inteldata.net/#'. The page title is 'Métodos Expertos' and the subtitle is 'Ingeniería del Conocimiento para la Productividad'. The main content area is titled 'PERFILES DE CARGOS' and contains a 'Crear perfil de cargo' button and a table of job profiles.

Opciones	Nombre	Descripción	ASERTIVIDAD	ASUNCIÓN DE RIESGOS	AUDACIA	AUTOESTIM	CURIOSIDAD	EXTROVERSI	FLEXIBILIDAD	HABILIDAD CONVERSACIONAL
Editar Borrar	GERENTE DE MERCADEO	El gerente que se encarga de mercader.	[0.7, 1.0]	[0.4, 0.8]		[0.7, 1.0]	[0.3, 0.5]	[0.8, 1.0]		[0.7, 1.0]
Editar Borrar	INGENIERO DE SOFTWARE	Persona que desarrolla sistemas de software	[0.3, 1.0]			[0.6, 0.9]	[0.7, 1.0]	[0.3, 0.5]	[0.6, 1.0]	[0.1, 0.5]
Editar Borrar	OPERARIO DE MAGUINARIA			[0.0, 0.2]	[0.4, 0.6]		[0.1, 0.3]			

Si el usuario decide crear o editar una competencia, aparece un cuadro de diálogo para que pueda ingresar los datos correspondientes y guardarlos.

Ilustración 26. Cuadro de diálogo para crear y editar perfiles de cargos.



Al hacer click en el botón GUARDAR, se procesa el guardado de los datos y se genera una notificación de éxito o fracaso de la operación. No es necesario especificar un intervalo de confianza para cada competencia.

B.3. Administración de Candidatos y sus Perfiles Psicológicos

Al hacer click en el link Candidatos, en el Panel de Navegación, se despliega un listado de los candidatos creados, y se da la opción de crear un nuevo candidato, o editar o borrar un candidato ya creado.

Ilustración 27. Despliegue de la administración de candidatos en el Panel Principal.

selecciondepersonal.inteldata.net/#

Métodos Expertos
Ingeniería del Conocimiento para la Productividad

DESARROLLO DE UN PROTOTIPO DE SISTEMA EXPERTO PARA EL APOYO EN LA TOMA DE DECISIONES DEL PROCESO DE SELECCIÓN DE PERSONAL
ESTEBAN ANDRÉS PÉREZ VARGAS - Cód. 200029101010 - Universidad EAFIT

Inicio
Competencias
Perfiles de Cargos
Candidatos
Evaluar competencias
Sobre este sistema

CANDIDATOS Y PERFILES PSICOLÓGICOS

Crear candidato y su perfil psicológico

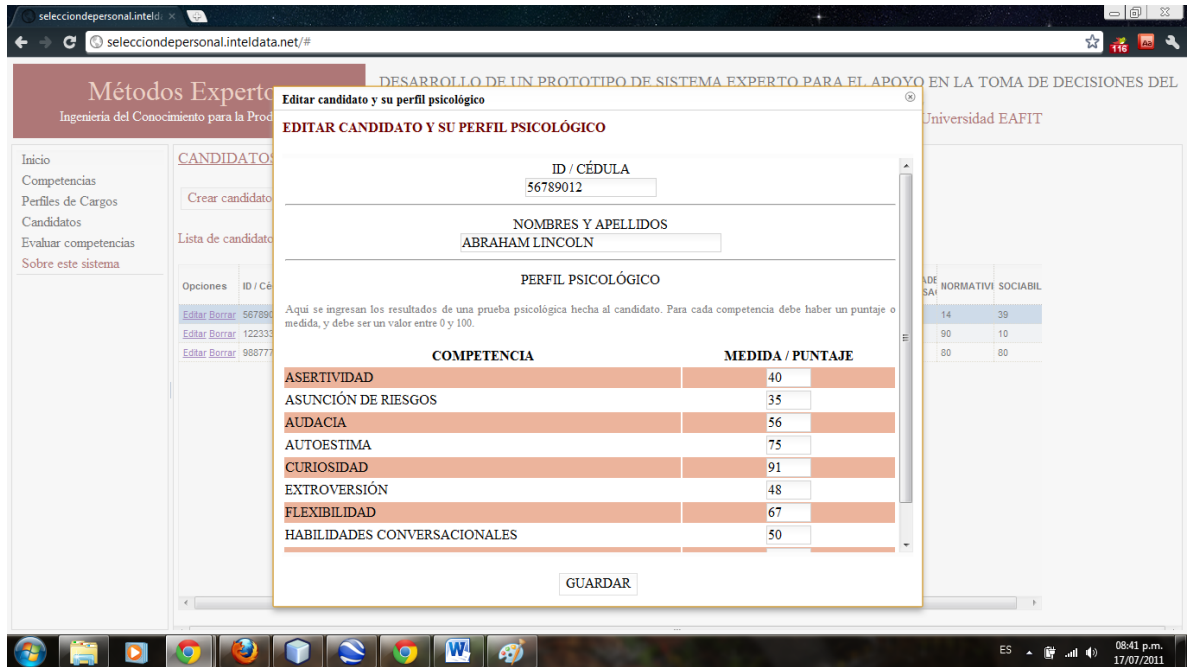
Lista de candidatos y sus perfiles psicológicos

Opciones	ID / Cédula	Nombre	ASERTIVIDAD DE RIESGOS	ASUNCIÓN	AUDACIA	AUTOESTIM	CURIOSIDAD	EXTROVERSI	FLEXIBILIDAD	HABILIDAD CONVERSACIONAL	NORMATIVIDAD	SOCIALIZACIÓN
Editar Borrar	56789012	ABRAHAM LINCOLN	40	35	56	75	91	48	67	50	14	39
Editar Borrar	122333	FRANKLIN D. ROOSEVELT	50	20	30	40	50	60	70	80	90	10
Editar Borrar	988777	GEORGE WASHINGTON	20	30	90	40	80	50	70	60	80	80

ES 08:42 p.m. 17/07/2011

Si el usuario decide crear o editar un candidato, aparece un cuadro de diálogo para que pueda ingresar los datos correspondientes y guardarlos.

Ilustración 28. Cuadro de diálogo para crear y editar candidatos.

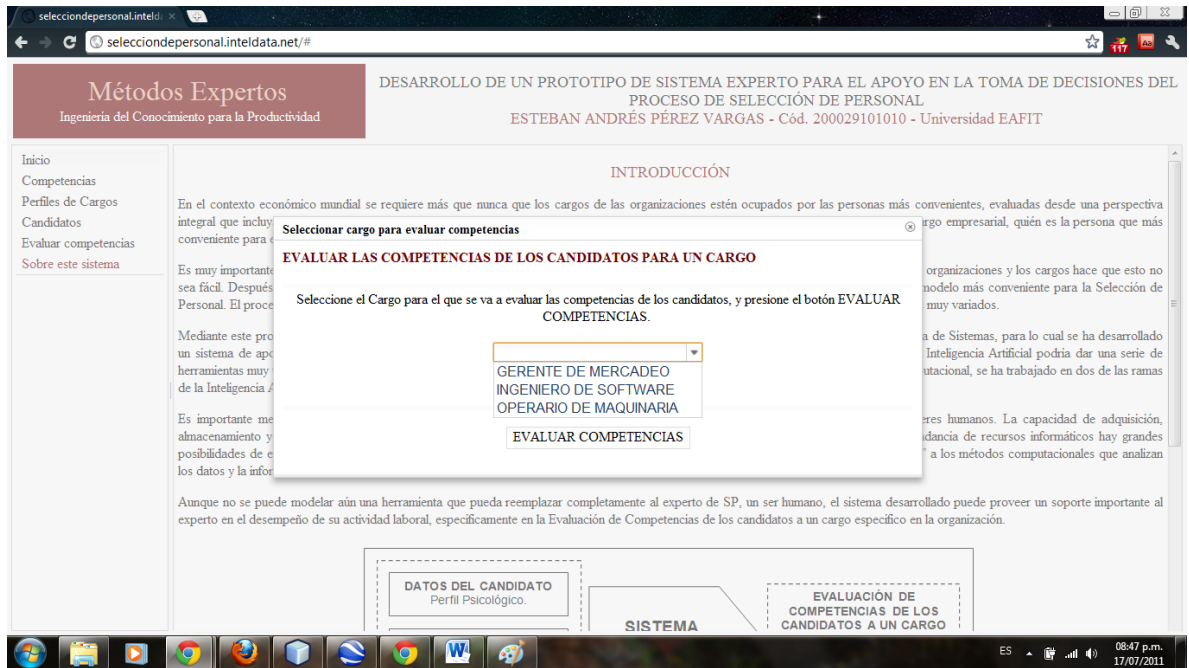


Al hacer click en el botón GUARDAR, se procesa el guardado de los datos y se genera una notificación de éxito o fracaso de la operación. Es necesario especificar un puntaje o medida para cada competencia.

B.4. Evaluación de Competencias Para Un Cargo

Al hacer click en Evaluar Competencias, se despliega un cuadro de diálogo para seleccionar el cargo para el cual se van a evaluar las competencias de todos los candidatos.

Ilustración 29. Cuadro de diálogo para seleccionar cargo para el cual se va a evaluar las competencias de los candidatos.



Al seleccionar el cargo, aparece la evaluación de las competencias con información que ayuda a explicar la decisión del sistema.

Ilustración 30. Evaluación de competencias para un cargo.

Métodos Expertos
Ingeniería del Conocimiento para la Productividad

DESARROLLO DE UN PROTOTIPO DE SISTEMA EXPERTO PARA EL APOYO EN LA TOMA DE DECISIONES DEL PROCESO DE SELECCIÓN DE PERSONAL
ESTEBAN ANDRÉS PÉREZ VARGAS - Cód. 200029101010 - Universidad EAFIT

INGENIERO DE SOFTWARE
EVALUACIÓN DE LAS COMPETENCIAS DE LOS CANDIDATOS PARA EL CARGO

Candidatos seleccionados

Distancia:
0.37

Candidatos:
ABRAHAM LINCOLN

Detalles de la Evaluación de Competencias de todos los candidatos

En la siguiente tabla aparecen listados todos los candidatos, especificando la distancia global entre cada uno de ellos y el cargo. También aparece la distancia por cada competencia.

ID / Cédula	Nombre	Distancia	ASERTIVIDAD	AUTOESTIMA	CURIOSIDAD	EXTROVERSIÓN	FLEXIBILIDAD	HABILIDAD CONVERSACIONAL	NORMATIVIDAD	SOCIABILIDAD
56789012	ABRAHAM LINCOLN	0.37	0.57	0.3	0.3	0.21	0.43	0.3	0.6	0.22
122333	FRANKLIN D. ROOSEVELT	0.53	0.43	0.7	0.47	0.37	0.33	0.9	0.64	0.4
988777	GEORGE WASHINGTON	0.55	0.86	0.7	0.18	0.23	0.33	0.7	0.53	0.9

Así se obtiene la ejecución de la tarea ECO de forma computarizada, para apoyar el proceso de Selección de Personal.

Anexo C. Casos de Prueba

Se empieza por definir las Competencias y sus funciones para determinar el Cumplimiento a partir del puntaje obtenido por el Candidato en una Prueba Psicológica. La función del Cumplimiento se basa en la Función Lambda y sus parámetros a y m.

Tabla 34. Ejemplo de definición de competencias y la función para determinar el Cumplimiento a partir del puntaje del candidato en una Prueba Psicológica

ID	Competencia	a	m
C1	ASERTIVIDAD	15	75
C2	ASUNCIÓN DE RIESGOS	30	75
C3	AUDACIA	25	75
C4	AUTOESTIMA	30	80
C5	CURIOSIDAD	15	90
C6	EXTROVERSIÓN	10	85
C7	FLEXIBILIDAD	50	80
C8	HABILIDADES CONVERSACIONALES	40	65
C9	NORMATIVIDAD	25	95
C10	SOCIABILIDAD	30	80

También se requiere que se determinen los Perfiles de los Cargos, basándose en Conjuntos Difusos Valorados en Intervalos de Confianza (CDVIC) para cada Competencia. Nótese que si no se requiere evaluar una Competencia específica en un Cargo, no se especifica un CDVIC para esa Competencia.

Tabla 35. Ejemplo de definición de Perfiles de Cargos

		CARGO EXTREMO 1	CARGO EXTREMO 2	DIRECTOR DE MERCADEO	INGENIERO DE SOFTWARE
C1	min	0	0,8	0,7	0,3
	max	0,2	1	1	1
C2	min	0	0,8	0,4	
	max	0,2	1	0,8	

		CARGO EXTREMO 1	CARGO EXTREMO 2	DIRECTOR DE MERCADEO	INGENIERO DE SOFTWARE
C3	min	0	0,8	0,6	
	max	0,2	1	0,9	
C4	min	0	0,8	0,7	0,6
	max	0,2	1	1	0,9
C5	min	0	0,8	0,3	0,7
	max	0,2	1	0,5	1
C6	min	0	0,8	0,8	0,3
	max	0,2	1	1	0,5
C7	min	0	0,8		0,7
	max	0,2	1		1
C8	min	0	0,8	0,7	0,1
	max	0,2	1	1	0,5
C9	min	0	0,8		0,3
	max	0,2	1		0,7
C10	min	0	0,8	0,6	0,1
	max	0,2	1	1	0,4

Hay dos de estos Perfiles de Cargo que merecen especial atención, pues se han creado como ayuda para probar la efectividad del sistema en casos extremos:

- **Cargo Extremo 1:** Busca identificar aquellos Candidatos con Cumplimientos bajos en cada Competencia.
- **Cargo Extremo 2:** Busca identificar aquellos Candidatos con Cumplimientos altos en cada Competencia.

Ahora se requiere los resultados de las Pruebas Psicológicas de todos los Candidatos a evaluar. Estas contienen el puntaje o medida obtenido por el Candidato en cada Competencia. Se hará ejemplos para 10 Candidatos. P1 y P2 tienen puntajes de 100 en todas las Competencias, P3 y P4 tienen puntajes de 10 en cada Competencia. Para los demás Candidatos los puntajes fueron generados aleatoriamente entre 0 y 100.

Tabla 36. Ejemplo de las Pruebas Psicológicas de los Candidatos.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
C1	100	100	10	10	95	79	19	28	47	12
C2	100	100	10	10	26	84	74	34	41	47
C3	100	100	10	10	31	42	34	82	60	41
C4	100	100	10	10	74	56	43	56	22	89

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
C5	100	100	10	10	22	74	78	14	11	65
C6	100	100	10	10	5	44	46	25	72	87
C7	100	100	10	10	63	29	10	91	59	50
C8	100	100	10	10	54	14	70	36	57	72
C9	100	100	10	10	53	8	83	31	17	49
C10	100	100	10	10	32	12	46	37	41	90

Se espera que P1 y P2 obtengan las mismas distancias en la evaluación de competencias para todos los Cargos. Lo mismo debe suceder entre P3 y P4.

Estos cuatro casos son un control para determinar si efectivamente se obtienen los mismos resultados para Candidatos en la misma situación.

Con la definición de la función del Cumplimiento con base en los puntajes de la Prueba Psicológica, se obtiene el Cumplimiento de cada Candidato con cada Competencia.

Tabla 37. Ejemplo del Cumplimiento de cada Candidato con cada Competencia.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
C1	1,00	1,00	0,00	0,00	1,00	1,00	0,07	0,22	0,53	0,00
C2	1,00	1,00	0,00	0,00	0,00	1,00	0,98	0,09	0,24	0,38
C3	1,00	1,00	0,00	0,00	0,12	0,34	0,18	1,00	0,70	0,32
C4	1,00	1,00	0,00	0,00	0,88	0,52	0,26	0,52	0,00	1,00
C5	1,00	1,00	0,00	0,00	0,09	0,79	0,84	0,00	0,00	0,67
C6	1,00	1,00	0,00	0,00	0,00	0,45	0,48	0,20	0,83	1,00
C7	1,00	1,00	0,00	0,00	0,43	0,00	0,00	1,00	0,30	0,00
C8	1,00	1,00	0,00	0,00	0,56	0,00	1,00	0,00	0,68	1,00
C9	1,00	1,00	0,00	0,00	0,40	0,00	0,83	0,09	0,00	0,34
C10	1,00	1,00	0,00	0,00	0,04	0,00	0,32	0,14	0,22	1,00

Ahora ya se tiene toda la información necesaria para evaluar competencias para los cuatro cargos especificados.

Para el CARGO EXTREMO 1, que sería apto para los Candidatos con los puntajes más bajos en todas las Competencias, se obtuvo la siguiente información.

Tabla 38. Ejemplo de Evaluación de Competencias para el Cargo Extremo 1.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
C1	1,00	1,00	0,20	0,20	1,00	1,00	0,13	0,22	0,53	0,20
C2	1,00	1,00	0,20	0,20	0,20	1,00	0,98	0,11	0,24	0,38
C3	1,00	1,00	0,20	0,20	0,12	0,34	0,18	1,00	0,70	0,32
C4	1,00	1,00	0,20	0,20	0,88	0,52	0,26	0,52	0,20	1,00
C5	1,00	1,00	0,20	0,20	0,11	0,79	0,84	0,20	0,20	0,67
C6	1,00	1,00	0,20	0,20	0,20	0,45	0,48	0,20	0,83	1,00
C7	1,00	1,00	0,20	0,20	0,43	0,20	0,20	1,00	0,30	0,20
C8	1,00	1,00	0,20	0,20	0,56	0,20	1,00	0,20	0,68	1,00
C9	1,00	1,00	0,20	0,20	0,40	0,20	0,83	0,11	0,20	0,34
C10	1,00	1,00	0,20	0,20	0,16	0,20	0,32	0,14	0,22	1,00
DIST.	1,00	1,00	0,20	0,20	0,41	0,49	0,52	0,37	0,41	0,61

Tal como se esperaba, los candidatos con las menores distancias son P2 y P3, dado que sus Cumplimientos son los más bajos para todas las Competencias, y se les está evaluando para un Cargo que premia todos los Cumplimientos bajos. También se esperaba que los Candidatos con las distancias más altas sean P1 y P2, pues todos sus Cumplimientos son altos.

Ahora en la evaluación para el CARGO EXTREMO 2, que favorece a los Candidatos con Cumplimientos altos en todas las Competencias, se obtiene los siguientes resultados.

Tabla 39. Ejemplo de Evaluación de Competencias para el Cargo Extremo 2.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
C1	0,20	0,20	1,00	1,00	0,20	0,20	0,93	0,78	0,47	1,00
C2	0,20	0,20	1,00	1,00	1,00	0,20	0,18	0,91	0,76	0,62
C3	0,20	0,20	1,00	1,00	0,88	0,66	0,82	0,20	0,30	0,68
C4	0,20	0,20	1,00	1,00	0,12	0,48	0,74	0,48	1,00	0,20
C5	0,20	0,20	1,00	1,00	0,91	0,21	0,16	1,00	1,00	0,33
C6	0,20	0,20	1,00	1,00	1,00	0,55	0,52	0,80	0,17	0,20
C7	0,20	0,20	1,00	1,00	0,57	1,00	1,00	0,20	0,70	1,00
C8	0,20	0,20	1,00	1,00	0,44	1,00	0,20	1,00	0,32	0,20
C9	0,20	0,20	1,00	1,00	0,60	1,00	0,17	0,91	1,00	0,66
C10	0,20	0,20	1,00	1,00	0,96	1,00	0,68	0,86	0,78	0,20
DIST.	0,20	0,20	1,00	1,00	0,67	0,63	0,54	0,71	0,65	0,51

Se observa que sucede lo contrario a lo sucedido en la evaluación de CARGO EXTREMO 1. P1 y P2 obtienen las menores distancias porque tienen puntajes y Cumplimientos altos para todas las Competencias. P3 y P4 resultan penalizados pues tienen puntajes y Cumplimientos bajos.

A continuación se muestra los resultados para la evaluación de competencias para los Cargos DIRECTOR DE MERCADEO e INGENIERO DE SOFTWARE.

Tabla 40. Ejemplo de Evaluación de Competencias para el Cargo de Director de Mercadeo.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
C1	0,30	0,30	1,00	1,00	0,30	0,30	0,93	0,78	0,47	1,00
C2	0,60	0,60	0,80	0,80	0,80	0,60	0,58	0,71	0,56	0,42
C3	0,40	0,40	0,90	0,90	0,78	0,56	0,72	0,40	0,20	0,58
C4	0,30	0,30	1,00	1,00	0,18	0,48	0,74	0,48	1,00	0,30
C5	0,70	0,70	0,50	0,50	0,41	0,49	0,54	0,50	0,50	0,37
C6	0,20	0,20	1,00	1,00	1,00	0,55	0,52	0,80	0,17	0,20
C8	0,30	0,30	1,00	1,00	0,44	1,00	0,30	1,00	0,32	0,30
C10	0,40	0,40	1,00	1,00	0,96	1,00	0,68	0,86	0,78	0,40
DIST.	0,40	0,40	0,90	0,90	0,61	0,62	0,63	0,69	0,50	0,45

Es coincidencia que nuevamente los Candidatos P1 y P2 obtuvieran las distancias más bajas en el cargo DIRECTOR DE MERCADEO. Esto sucede porque en general los CDVIC definidos premian Cumplimientos relativamente altos en todas las Competencias.

Tabla 41. Ejemplo de Evaluación de Competencias para el Cargo de Ingeniero de Software.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
C1	0,70	0,70	1,00	1,00	0,70	0,70	0,93	0,78	0,47	1,00
C4	0,40	0,40	0,90	0,90	0,28	0,38	0,64	0,38	0,90	0,40
C5	0,30	0,30	1,00	1,00	0,91	0,21	0,16	1,00	1,00	0,33
C6	0,70	0,70	0,50	0,50	0,50	0,15	0,18	0,30	0,53	0,70
C7	0,30	0,30	1,00	1,00	0,57	1,00	1,00	0,30	0,70	1,00
C8	0,90	0,90	0,50	0,50	0,46	0,50	0,90	0,50	0,58	0,90
C9	0,70	0,70	0,70	0,70	0,30	0,70	0,53	0,61	0,70	0,36
C10	0,90	0,90	0,40	0,40	0,36	0,40	0,22	0,26	0,18	0,90
DIST.	0,61	0,61	0,75	0,75	0,51	0,51	0,57	0,52	0,63	0,70

Los Candidatos con la menor distancia para el Cargo de INGENIERO DE SOFTWARE son P5 y P6, aunque P8 tiene una distancia ligeramente mayor. Nótese que a pesar de las distancias casi idénticas entre los tres candidatos, varían considerablemente en sus Cumplimientos para las distintas Competencias.