
Sistema Experto de
recomendación de
lenguaje de programación

Victor Gualdras de la Cruz
Juan Carlos Fernández Durán

Índice general

Sistema Experto de recomendación de lenguaje de programación	3
<i>Juan Carlos Fernández Durán and Victor Gualdrás de la Cruz</i>	
1. Introducción	4
1.1. Planteamiento del Problema	4
1.2. Objetivos y Alcance	4
1.3. Los Expertos	5
1.4. Los usuarios finales del sistema	5
2. Estado del Arte	5
2.1. Conceptos y Procedimientos principales del área de aplicación del Sistema Experto	6
2.2. Conceptos y Procedimientos principales de Informática que serán utilizados	6
2.3. Revisión Bibliográfica	7
3. Estudio de viabilidad. Test de Slagle	8
3.1. Definición de las características	8
3.2. Asignación de pesos	9
3.3. Evaluación de la aplicación	9
4. Adquisición del conocimiento	14
4.1. El Proceso de las Entrevistas	15
4.2. Entrevistas y sus resultados	15
4.3. El proceso de conceptualización	16
4.4. Conceptualización y sus resultados	16
5. Implementación	30
5.1. Interfaz Gráfica de Usuario	30
5.2. Herramienta de Diseño de Interfaz seleccionada	31
5.3. Diseño de la Interfaz Gráfica de Usuario del Sistema Experto . . .	31
6. Conclusiones y trabajo futuro	32
A. Entrevista David Villa Alises	35
B. Entrevista Pascual Julián Iranzo	37
C. Entrevista Ismael Caballero	39

Sistema Experto de recomendación de lenguaje de programación

Juan Carlos Fernández Durán and Victor Gualdrás de la Cruz

Universidad Castilla la Mancha, Escuela Superior de Informática
Ciudad Real, España

Resumen En este documento se expone el objetivo que pretende cumplir el sistema de recomendación de lenguajes de programación. Se analizará la necesidad de un sistema como este, lo que quiere abarcar y las utilidades que tendrá. Posteriormente se presentarán los expertos de los que se recabará el conocimiento necesario para el sistema, mostrando sus habilidades y conocimientos más específicos, y lo que aportará cada uno al proyecto. Se mostrarán también los potenciales usuarios que tendrá el SE. A continuación se mostrará en que situación se encuentra este tipo de sistemas, analizando si existen otros sistemas similares, o que información de provecho existe que se pueda obtener. También se mostrarán algunos conceptos que serán usados a lo largo de este documento referentes tanto al campo de aplicación como a los Sistemas Expertos en general. Para justificar y comprobar que el sistema es viable y merece la pena su desarrollo se realizará el test de Slagle.

Abstract In this document is exposed the goal that aims to accomplish the programming language recommender system. It would be analyzed the requirement of a system like that, what it wants to cover and the utilities it will have. After this, the experts whose knowledge we will take of will be presented. The skills and knowledge of this experts will be shown, and it will show which give each other to the project. Also the potential users will be presented.

After that, it will be shown in which situation it's found this kind of systems, analyzing if there are another similars works, o what helpful information we can take. Next some related concepts are going to be explained, referred to the application field and so to the field of Expert System. To warrant and check the viability and whether it is worth to go away with the project the Slagle Test is going to be done.

1. Introducción

En esta sección se va a realizar una breve introducción acerca del Sistema Experto de recomendación de lenguaje de Programación en el que quedará definido el problema, definiendo a su vez el alcance que abarcará dicho sistema experto y de cuáles son los objetivos que satisfará el mismo. Posteriormente se presentarán los expertos de los cuales se obtendrá el conocimiento y a qué tipo de usuario estará dirigido este Sistema Experto

1.1. Planteamiento del Problema

El mundo de la informática es demasiado grande y muchas veces la tecnología y tendencias actuales suelen variar con rapidez, esto puede ser desconcertante para las personas que dedican su vida diaria a este campo, entre ellos, los informáticos. Muchas veces, hay quien se puede encontrar en la situación de querer abordar un nuevo problema, o bien uno quiere especializarse en un campo específico, pero no conoce las herramientas apropiadas para ello, en este caso, los lenguajes de programación, los cuales se caracterizan porque no existe uno que predomine sobre todos, ya que cada uno se adapta mejor a unos problemas que otros abordan peor.

Sin embargo, en la vida real, tampoco hay 1 lenguaje específico para cada tipo de problema, si es cierto que, existen unos determinados lenguajes de programación más importantes que otros a la hora de abordar una temática concreta, por ejemplo, la programación en dispositivos móviles, la programación de sistemas empotrados, la programación de un sistema experto o incluso un lenguaje que sea más didáctico para aprender a programar.

En base a unas características se procederá a recomendar el lenguaje de programación que más adecuado al usuario según sus necesidades o preocupaciones. Este sistema experto se centrará en las necesidades más comunes, como por ejemplo, necesidades empresariales que van fuertemente ligadas a una tecnología, como los dispositivos Smartphone, de entre los que se pueden diferenciar principalmente los sistemas Android y los sistemas iOS. También se estudiarán los lenguajes recomendados para aplicaciones convencionales de escritorio por ejemplo de gestión, programación de microcontroladores, campo de los videojuegos o desarrollo de páginas web entre otros.

1.2. Objetivos y Alcance

El objetivo de este sistema experto es proporcionar al usuario un lenguaje de programación que se adapte a sus problemas y necesidades, pudiendo estas diferir de varias formas, los campos de aplicación para el sistema experto serán:

- Aplicaciones de sobremesa
- Aplicaciones multiplataforma

- Páginas Web
- Aprendizaje
- Sistemas Expertos
- Videojuegos
- Aplicaciones para móviles
- Microcontroladores
- Conocer un determinado paradigma (POO por ejemplo)

1.3. Los Expertos

El conocimiento será adquirido principalmente de 3 expertos, siendo todos profesores con actividades docentes en la Escuela Superior de Informática de la UCLM

- **Pascual Julián Iranzo:** Doctorado en Ciencias de la Computación, con actividad docente en Lógica y Programación declarativa durante 18 años, experto en Sistemas Inteligentes proporcionará el conocimiento necesario acerca de la rama de la Programación Declarativa, sus ventajas frente a la Programación Imperativa así como de Sistemas Inteligentes.
- **David Villa Alises:** Doctorado en Ingeniería Informática, miembro del grupo de investigación **Arco** con actividad docente en Redes de Computadores y Sistemas Distribuidos.
- **Ismael Caballero Muñoz-Reja:** Experiencia profesional en el mundo de la informática, proporcionará la visión más empresarial acerca de los lenguajes de programación, actualmente desempeña su actividad docente en la Escuela Superior de Informática de la UCLM en asignaturas como Ingeniería del Software II

1.4. Los usuarios finales del sistema

Los usuarios serán, en su mayoría, personas que tienen un interés especial en la informática para realizar un determinado desarrollo, pudiendo ser para fines lucrativos, mero aprendizaje, práctica, etc.

Tras recibir las necesidades del usuario, el sistema responderá realizando una simulación del juicio del experto recomendando el lenguaje de programación que crea más apropiado

2. Estado del Arte

En el estado del arte se presenta un breve estudio sobre la presencia de cómo se ha llevado este problema al campo de los sistemas expertos por parte de otras personas, así como artículos en los que se describa esta problemática. Se abordará brevemente cada uno de estos artículos con un breve resumen.

También se introducirán los conceptos más importantes tanto del sistema experto como de los lenguajes de programación.

2.1. Conceptos y Procedimientos principales del área de aplicación del Sistema Experto

A continuación se presentarán los conceptos más importantes del área de los lenguajes de Programación

- **Lenguaje de Programación** es un lenguaje formal diseñado para comunicar instrucciones a un computador.
- **Lenguaje Compilado:** El programa escrito en el determinado lenguaje pasa por un proceso de compilación, de manera que dicho texto en el determinado lenguaje es convertido normalmente a un lenguaje intermedio como por ejemplo lenguaje ensamblador, y en última instancia, a un archivo binario que el procesador puede entender directamente.
- **Lenguaje interpretado:** El programa es escrito en el determinado lenguaje pero no se llega a compilar, en lugar de eso, un programa carga secuencialmente instrucción por instrucción.
- **Programación Orientada a Objetos:** Paradigma que un lenguaje de programación puede adoptar o no, que consiste en agrupar los datos de un determinado problema, quedando asociada con la lógica del propio programa.
- **Flujo de Control:** Orden en el que las instrucciones de programación se van ejecutando.
- **Programación Declarativa:** Paradigma de programación en el que se indica qué se debe resolver en lugar de cómo se debe resolver. El programador no tiene porqué especificar el flujo de control.

2.2. Conceptos y Procedimientos principales de Informática que serán utilizados

A continuación se presentará brevemente los conceptos principales que se van a utilizar para construir el sistema experto.

Ingeniería del Conocimiento. Se refiere a todo el proceso de uso de técnicas para la construcción, mantenimiento y uso de un Sistema basado en el conocimiento. El objetivo principal es construir modelos de conocimiento, dicho conocimiento es extraído principalmente por 2 vías, que puede ser la manual o la automática, en la manual, el ingeniero del conocimiento aplicará diversas técnicas para extraer el conocimiento de uno o varios expertos como por ejemplo, las entrevistas.

En el proceso de la obtención de conocimiento de forma automática, también llamado **Knowledge Discovery from Data** abreviado como KDD, se obtiene la información a partir de un gran número de datos.

Sistemas Expertos. Un sistema experto es un sistema que trata de emular la toma de decisión que tomaría un humano en un campo determinado. Estos sistemas expertos están divididos en 2 principales partes, una base del conocimiento,

en la que estarán las reglas y los hechos, y por otra parte, existirá un motor de inferencia, que se encargará de disparar los hechos con las reglas obteniendo la información necesaria.

Más tarde el usuario definirá su situación y el sistema experto simulará el juicio del experto dando como resultado un lenguaje de programación recomendado

2.3. Revisión Bibliográfica

Tras una búsqueda exhaustiva, solo se ha encontrado un único sistema experto [1] el cual en base a unas preguntas cortas determina que lenguaje de programación es el que debes usar, su código fuente está publicado, y se puede observar que es bastante sencillo, el sistema está compuesto por unas preguntas que según la respuesta te redirige a otra pregunta o directamente a una respuesta, toma en consideración varias cosas, como por ejemplo que tipo de desarrollo se está tratando, si es para una inteligencia artificial, una aplicación de escritorio, móvil, o web, tiene en cuenta para en que plataformas se quiere trabajar, si tiene importancia que sea multiplataforma...

Se puede encontrar un artículo[2] acerca de qué lenguaje de programación elegir como primer lenguaje de programación para aprender a programar, esta será otra perspectiva que este sistema experto abordará, y por lo tanto es interesante analizarla. En este artículo se habla de cómo Pascal fue inicialmente concebido como un lenguaje de programación para la enseñanza, además, tuvo mucho éxito y se utilizó con propósitos comerciales, sin embargo ha quedado muy desfasado en el tiempo. Se realiza un estudio de los requisitos que debería tener el lenguaje de programación que se utilizará como primer lenguaje, y se hace también un estudio de qué tipo de aplicaciones tienen unos determinados lenguajes candidatos a ser el primer lenguaje de programación, siendo C# y Java los lenguajes con mayor aplicación. El artículo concluye que aunque Pascal fuera ideal, se ha quedado muy desfasada y recomienda como primer lenguaje Python, ya que cumple con todos los requisitos, soporta varios paradigmas de programación como la orientada a objetos, estructural o funcional. además de ser un lenguaje muy utilizado.

Dado que en el sistema experto se abordará el tema empresarial, es interesante ver una comparativa de los lenguajes de programación mejor pagados [3] ya que algunas personas pueden interesarles únicamente el punto de vista económico, visto desde esta perspectiva, los desarrolladores de Ruby on Rails, Objective C y Python son los lenguajes mejor pagados de la industria. Por parte de Ruby on Rails y Python, se debe a la enorme productividad que consiguen con poco tiempo, por parte de Objective C se debe a la exclusividad impuesta de la plataforma Apple para programar en dicho lenguaje para sus propias aplicaciones.

Otra de las grandes áreas de aplicación de la programación es la programación web, en este artículo [4] se puede observar una comparativa de algunos de

los lenguajes de programación más interesantes para afrontar un desarrollo web, siendo HTML y CSS los más básicos, luego se ve diferenciada por los que se ejecutan desde la aplicación cliente y desde la servidora. En el caso del cliente se dispone de JavaScript y ActionScript, desde el servidor se recomienda utilizar PHP, Java, Python o Ruby. El autor concluye con que la decisión depende del problema, un problema se puede ajustar mejor a un determinado lenguaje de programación que otro.

Es especialmente interesante este artículo [5] donde además de hacer una ligera pincelada de cada lenguaje, muestra en imágenes varios perfiles de desarrollador típicos en la industria, habilidades asociadas a esos perfiles y lenguajes de programación que también están relacionados. Permite ver de una forma clara y sencilla que se espera de cada perfil además de una breve descripción de los lenguajes, por ejemplo, los desarrolladores de videojuegos tienden a usar más los lenguajes C, C++, C# y Java, trabajan bien bajo presión, y suelen tener otras habilidades relacionadas con el diseño.

3. Estudio de viabilidad. Test de Slagle

Se procede a analizar si el proyecto previsto es viable de cara a realizarlo mediante un Sistema Experto. Para ello se usará el conocido como Test de Slagle. En primer lugar se definirán las características del sistema y de los involucrados o stakeholders. A continuación se asignarán los pesos para finalmente evaluar si el sistema cumple con los objetivos necesarios.

3.1. Definición de las características

Se contemplan las características del sistema desde cuatro dimensiones: Plausibilidad, Justificación, Adecuación, Éxito, sobre cada una de estas dimensiones se establecen tres categorías diferentes: Directivos o Usuarios, los expertos y la tarea. Las características pueden ser esenciales o deseables y se ha establecido un valor umbral de 7 que debe ser superado por las primeras.

- **Plausibilidad:** Se analiza si la tarea es susceptible de poder abordarse desde el punto de vista de la Ingeniería del Conocimiento, y si se cuenta con los medios para hacerlo.
- **Justificación:** Se justifica o intenta justificar la necesidad de realizar este sistema desde la perspectiva de la Ingeniería del Conocimiento. Puede haber muchos motivos de justificación, como el económico, el social etc.
- **Adecuación:** Se analiza lo adecuado del problema para ser resuelto mediante técnicas de IC. El problema podría ser mejor abordado mediante la algoritmia u otros procedimientos.
- **Éxito:** Se estima y determina el éxito que podría llegar a tener el sistema.

3.2. Asignación de pesos

En función de su importancia relativa cada característica recibirá un peso de 0 a 10. Este peso deberá ser constante evitando así que se ajuste la evaluación a conveniencia. Al contrario de lo que se pudiese pensar, una característica esencial no debe tener un peso de 10, si bien deberá tener un valor alto.

3.3. Evaluación de la aplicación

A continuación se procede de la siguiente manera para evaluar el sistema. Se utilizará la leyenda [Table 1] mostrada a continuación para seguir el procedimiento:

Cuadro 1. Leyenda

Abreviatura	Significado	RANGO
P	Peso de las características (fijo a priori)	0...10
Ppi	Peso de una característica de la dimensión Plausibilidad	Pp1 ... Pp10
Pji	Peso de una característica de la dimensión Justificación	Pj1 ... Pj7
Pai	Peso de una característica de la dimensión Adecuación	Pa ... Pa12
Pei	Peso de una característica de la dimensión Éxito	Pe1 ... Pe17
V	Valor de las características (lo asignamos nosotros)	0 ... 10
Vpi	Valor de una característica de posibilidad (plausibilidad)	Vp1 ... Vp10
Vji	Valor de una característica de justificación	Vj1 ... Vj7
Vai	Valor de una característica de adecuación	Va1 ... Va12
Ve1	Valor de una característica de éxito	Ve1 ... Ve17
VC	Valor total de una aplicación candidata	0 ... 100
Vci	Valor global de una aplicación en una dimensión (VC1,...)	0 ... 100
Vu	Valor umbral de una aplicación (habitualmente se considera 7)	0 ... 10
//	División entera	

Se va a proceder ahora con el método a seguir.

1. Asignación de un valor V a cada característica de 0 (que se considera nula) a 10 (totalmente cumplida). Si el valor de una característica esencial no alcanza el umbral exigido, su cómputo es cero y la aplicación queda rechazada.
2. Se multiplica cada valor de una característica por su correspondiente peso para obtener los valores ponderados de las características. Este cálculo se efectuará para cada una de las dimensiones en que se han establecido las características.
3. Multiplicar, para cada dimensión, estos valores ponderados de las características.
4. Obtener para cada dimensión la pseudo-media geométrica de los valores ponderados de las características. Para ello se calculará la raíz n -ésima del producto obtenido en el apartado anterior, utilizando como índice de la raíz el

valor máximo de los índices usados en cada dimensión.

$$VC1 = \prod_{i=1,2,5} (Vp_i/Vu_i) \left(\prod_{i=1}^{10} (Pp_i * Vp_i) \right)^{1/10}$$

$$VC2 = \prod_{i=1,4,5,7} (Vj_i/Vu_i) \left(\prod_{i=1}^7 (Pj_i * Vj_i) \right)^{1/7}$$

$$VC3 = \prod_{i=4,7,9,10} (Va_i/Vu_i) \left(\prod_{i=1}^{12} (Pa_i * Va_i) \right)^{1/12}$$

$$VC4 = \prod_{i=6,10,12,17} (Ve_i/Vu_i) \left(\prod_{i=1}^{17} (Pe_i * Ve_i) \right)^{1/17}$$

5. Dividir la suma de los valores globales de cada aplicación candidata, en las cuatro dimensiones, entre cuatro. Se obtendrá el valor total general de la aplicación.
6. Elegir la aplicación candidata con mejor valor, generalmente. De todas maneras, se establecerá un valor umbral para la elección de la tarea. Si el valor resultante es menor que el umbral exigido, se rechazará la tarea. Este proceso se realiza tanto para una sola tarea como para un conjunto de ellas.

$$VC = \begin{cases} \sum_{i=1}^4 (VC_i/4) & \text{si } \prod_{i=1}^4 VC_i \neq 0 \\ \text{en otro caso} & 0 \end{cases} \quad (1)$$

Leyenda Aquí se encuentra una guía de las abreviaturas usadas y su significado.

Cuadro 2. Categorías

CAT	Categoría
EX	Expertos
TA	Tarea
DU	Directivos y/o usuarios

Cuadro 3. Identificación de las Características

Pi	Característica de la dimensión Plausibilidad
Ji	Característica de la dimensión Justificación
Ai	Característica de la dimensión Adecuación
Ei	Característica de la dimensión Éxito

Cuadro 4. Tipos de características

E	Esencial
D	Deseable

Tablas. Mediante las siguientes tablas se va a evaluar si el sistema es viable.

Cuadro 5. Dimensión de Plausibilidad

CAT.	IDEN	PESO(P)	VALOR(V)	DENOMINACIÓN DE LA CARACTERÍSTICA	TIPO
EX	P1	10	10	Existen expertos.	E
EX	P2	10	9	El experto asignado es genuino.	E
EX	P3	8	10	El experto es cooperativo.	D
EX	P4	7	8	El experto es capaz de articular sus métodos pero no categoriza.	D
TA	P5	10	9	Existen suficientes casos de prueba; normales, típicos, ejemplares, correosos, etc	E
TA	P6	10	10	La tarea está bien estructurada y se entiende	D
TA	P7	10	8	Sólo requiere habilidad cognoscitiva.	D
TA	P8	9	9	No precisan resultados verdaderamente comprometidos con el proyecto.	D
TA	P9	9	8	La tarea no requiere sentido común.	D
DU	P10	7	10	Los directivos están verdaderamente comprometidos con el proyecto.	D

Cuadro 6. Dimensión de la Justificación

CAT.	IDEN.CAT	PESO(P)	VALOR(V)	DENOMINACIÓN DE LA CARACTERÍSTICA	TIPO
EX	J1	10	8	El experto NO esta disponible	E
EX	J2	10	7	Hay escasez de experiencia humana	D
TA	J3	8	10	Existe necesidad de experiencia simultánea en muchos lugares	D
TA	J4	10	7	Necesidad de experiencia en entornos hostiles, penosos y/o poco gratificantes	E
TA	J5	8	9	No existen soluciones alternativas admisibles	E
DU	J6	7	10	Se espera una alta tasa de recuperación de la inversión	D
DU	J7	8	10	Resuelve una tarea útil y necesaria	E

Cuadro 7. Dimensión de Adecuación

CAT.	IDEN. CAT.	PESO(P)	VALOR(V)	DENOMINACIÓN DE LA CARACTERÍSTICA	TIPO
EX	A1	5	10	La experiencia del experto está poco organizada	D
TA	A2	6	10	Tiene valor practico	D
TA	A3	7	9	Es más táctica que estratégica	D
TA	A4	7	10	Sirve a necesidades de largo plazo	E
TA	A5	5	10	La tarea, que no es demasiado fácil, pero es de conocimiento intensivo, tanto propio del dominio, como de manipulación de la información.	D
TA	A6	6	10	Es de tamaño manejable, y/o es posible un enfoque gradual y/o, una descomposición en subtareas independientes.	D
EX	A7	7	10	La transferencia de experiencia entre humanos es factible.	E
TA	A8	6	10	Estaba identificada como un problema en el área y los efectos de la introducción de un SE pueden planificarse.	D
TA	A9	9	10	No requiere respuestas en tiempo real inmediato".	E
TA	A10	9	10	La tarea no requiere investigación básica y usa, si alguna, poca generación y entendimiento el lenguaje natural.	E
TA	A11	5	10	El experto usa básicamente razonamiento simbólico que implica factores subjetivos.	D
TA	A12	5	9	Es esencialmente de tipo heurístico.	D

Cuadro 8. Dimensión de Éxito

CAT.	IDEN. CAT.	PESO(P)	VALOR(V)	DENOMINACIÓN DE LA CARACTERÍSTICA	TIPO
EX	E1	8	10	No se sienten amenazados por el proyecto, son capaces de sentirse intelectualmente unidos al proyecto.	D
EX	E2	6	10	Tienen un brillante historial en la realización de esta tarea	D
EX	E3	5	10	Hay acuerdos en lo que constituye una buena solución a la tarea.	D
EX	E4	5	9	La única justificación para dar un paso en la solución es la calidad de la solución final.	D
EX	E5	6	9	No hay un plazo de finalización estricto, ni ningún otro proyecto depende de esta tarea.	D
TA	E6	7	10	No está influenciada por vaivenes políticos.	E
TA	E7	8	8	Existen ya SS.EE. que resuelvan esa o parecidas tareas.	D
TA	E8	8	10	Hay cambios mínimos en los procedimientos habituales.	D
TA	E9	5	10	Las soluciones son explicables o interactivas.	D
TA	E10	7	9	La tarea es de I+D de carácter práctico, pero no ambas cosas simultáneamente.	E
DU	E11	6	10	Están mentalizados y tienen expectativas realistas tanto en el alcance como en las limitaciones.	D
DU	E12	7	10	No rechazan de plano esta tecnología.	E
DU	E13	6	10	El sistema interactúa inteligente y amistosamente con el usuario.	D
DU	E14	9	9	El sistema es capaz de explicar al usuario su razonamiento.	D
DU	E15	8	10	La inserción del sistema se efectúa sin traumas; es decir, apenas se interfiere en la rutina cotidiana de la empresa.	D
DU	E16	6	8	Están comprometidos durante toda la duración del proyecto, incluso después de su implantación.	D
DU	E17	8	9	Se efectúa una adecuada transferencia tecnológica.	E

Así pues, aplicando el método anteriormente descrito:

$$VC1 = \prod_{i=1,2,5} (Vp_i // Vu_i) \left(\prod_{i=1}^{10} (Pp_i * Vp_i) \right)^{1/10} = 80,79358$$

$$VC2 = \prod_{i=1,4,5,7} (Vj_i // Vu_i) \left(\prod_{i=1}^7 (Pj_i * Vj_i) \right)^{1/7} = 74,42169$$

$$VC3 = \prod_{i=4,7,9,10} (Va_i // Vu_i) \left(\prod_{i=1}^{12} (Pa_i * Va_i) \right)^{1/12} = 61,69024$$

$$VC4 = \prod_{i=6,10,12,17} (Ve_i // Vu_i) \left(\prod_{i=1}^{17} (Pe_i * Ve_i) \right)^{1/17} = 62,84026$$

Una vez calculados todos los valores se calcula el valor total de referencia del sistema:

$$VC = \frac{80,79358 + 74,42169 + 61,69024 + 62,84026}{4} = 69,93644$$

Únicamente queda calcular el Valor Normalizado. Valor Obtenido 69.93644
- Valor Máximo Posible 76,21 Valor Real X - 100

$$X = \frac{69,94 * 100}{76,21} = 91,38$$

Así pues, este sistema de recomendación de lenguajes de programación tiene una fiabilidad del **91.38**, el cual es muy alto, asegurando en gran medida la viabilidad del proyecto.

4. Adquisición del conocimiento

La adquisición del conocimiento es la extracción o recogida del conocimiento de la fuente experta para obtener el conocimiento heurístico y la experiencia en la resolución de problemas de un determinado dominio por parte del ingeniero, con el fin de que sea usado por un programa. Requiere de una larga e intensa interacción. En este caso las fuentes de las que se extraerá dicho conocimiento serán las fuentes citadas en las referencias y los expertos nombrados antes.

4.1. El Proceso de las Entrevistas

En el caso de este proyecto se ha decidido optar por un método manual, como son las entrevistas, como método principal de extracción del conocimiento a los expertos, si bien, se ha recurrido también a libros, documentos y fuentes Web como complemento a estas entrevistas. El proceso de las entrevistas es un proceso bastante complicado, especialmente por las diferencias que puedan existir con el experto que pueden ser de muchos tipos: El tiempo que el experto esta disponible, inexistencia de un experto adecuado, el ingeniero del conocimiento puede no tener las bases necesarias sobre el tema para poder comunicarse adecuadamente con este, el experto puede ser reticente a contar secretos o experiencias que le sean exclusivas etc.

4.2. Entrevistas y sus resultados

En este caso, se han utilizado tanto entrevistas no estructuradas como estructuradas. A pesar de que existen varios expertos se ha delimitado el contenido y finalidad del proyecto de tal forma que cada experto aporte su conocimiento sobre su campo específico sin entrar en conflicto o inconsistencias con lo que aporta otro experto. Por esta razón no se ha considerado necesario realizar técnicas de entrevistas múltiples. Las entrevistas han sido determinantes en el proceso de extracción del conocimiento. Sin estas entrevistas no se hubiese podido llevar a cabo gran parte del trabajo ya que existen pocos recursos sobre este tema, y mucho menos opiniones y experiencias de personas con experiencia en esta materia. Este es un campo que necesita en gran medida del factor de la experiencia, ya que teóricamente no se es capaz de analizar y valorar gran parte de las variables que presentan y que diferencia a los distintos lenguajes o tecnologías para la programación. Las entrevistas se han realizado de la siguiente manera:

- Primera reunión con el experto en la que se ha explicado en que iba a consistir del sistema y lo que se iba a necesitar de él, y se han establecido los principales puntos que se deberían tratar. Además se ha corroborado con el experto su experiencia y conocimiento en el tema. Esta primera entrevista correspondería al grupo de entrevistas no estructuradas.
- Se le han proporcionado al experto una serie de preguntas que iban a ser la espina dorsal de la entrevista para que pudiese preparárselas.
- Se ha procedido a una entrevista estructurada con unas preguntas bien definidas que permitían extraer lo necesario para el proyecto.
- Finalmente se ha contrastado lo comentado por los expertos con otras fuentes de conocimiento, especialmente con datos estadísticos aportados por organizaciones e instituciones de gran prestigio y confianza en el campo del sistema.

Respecto a las variantes de entrevistas utilizadas, decir que han sido mayoritariamente orientadas a una discusión focalizada. Se planteaban datos objetivos tales como "¿Cuál es el lenguaje de programación mas demandado actualmente por las empresas?" mostrando y estudiando algunas estadísticas. También se ha usado el método de la reclasificación o descomposición

de metas, se le han puesto casos generales de como lograr un objetivo, y el experto ha planteado distintas posibilidades, por ejemplo partiendo desde varias tecnologías o especialidades distintas llegar a ser un buen profesional en una determinada área.

4.3. El proceso de conceptualización

Antes de pasar a la etapa de representación del conocimiento es necesario una etapa intermedia denominada conceptualización, que consiste en hacer explícitos los conceptos claves y las relaciones relevantes entre estos. En esta etapa se establecen que conocimientos maneja el experto, cómo los utiliza, dónde los emplea y cuándo los usa, permitiendo de esta manera obtener una descripción detallada del problema mediante modelos conceptuales. Estos modelos conceptuales son una especificación de lo que debe hacer el sistema pero no especifican el cómo.

4.4. Conceptualización y sus resultados

Durante este proceso se determinan tres tipos de conocimientos, los Estratégicos que especifican qué hacer, dónde y porqué hacerlo, es decir estos conocimientos fijan la secuencia de pasos que el SE deberá seguir para ejecutar su tarea, los conocimientos Tácticos de Acción u Operativos, que especifican cómo y cuándo el SE puede añadir a sus conocimientos genéricos información actual acerca del caso y los conocimientos Fácticos o Declarativos, que especifican lo que es, o se cree que es verdad acerca del mundo en general y acerca del caso particular para el cual se está ejecutando la tarea. Finalmente también se presentan Metaconocimientos, estos registran la forma en que el experto usa los conocimientos para tomar una decisión, pueden existir metaconocimientos en cualquier nivel, es decir estos pueden ser metaconocimientos del tipo estratégicos, tácticos y fácticos.

Se ha realizado el siguiente esquema en el que se presenta una primera aproximación de la formalización del conocimiento

Conocimientos Factuales Especifican lo que es, o se cree que es verdad acerca del mundo en general y del caso particular para el cual se está ejecutando la tarea.

Primero de todo se ha procedido a la elaboración de un glosario, en el que se recogen los conceptos mas importantes del campo que nos ocupa que es el de la informática.

A

Administrador de base de datos: Persona encargada de velar por la integridad de los datos y sus asociaciones, así como de autorizar las modificaciones que se desee hacer.

Administrador de archivos (File Manager o Manejador de Archivos): Aplicación utilizada para facilitar distintas tareas con archivos como la copia, eliminación, movimiento entre otras. Algunos administradores de archivos permiten la asociación de las extensiones de los archivos con las aplicaciones preparados

para trabajar con los mismos, permitiendo abrir, , reproducir, modificar, etc. cada archivo con la aplicación asociada.

Algoritmo: Procedimiento lógico-matemático, aplicado para resolver un problema.

Análisis de sistemas: Estudio de una tarea o función para comprenderla y encontrar mejores maneras de realizarla.

Ancho de banda: Medida de la cantidad de información que puede pasar por una vía, expresada en bits/segundo (o algún múltiplo).

Aplicación: Programa diseñado para una determinada función

Apple®: Primera compañía fabricante de computadoras personales y creadora de la computadora Macintosh, cuyo sistema operativo incorporó la GUI de mayor éxito hasta la aparición de Microsoft Windows®.

Archivo: Conjunto de datos relacionados.

ASCII (American Standard Code for Information Interchange): Código estándar estadounidense para el intercambio de información. Código de siete bits adoptado como un estándar mundial para facilitar el intercambio de datos entre distintos sistemas y máquinas en ambientes conectados en red.

Assembler: Ver Lenguaje ensamblador.

Automatización: Realización de una combinación específica de acciones por una máquina, sin la ayuda de personas.

B

Backup: Copias de archivos, equipos de reemplazo o procedimientos alternativos disponibles para ser usados en caso de emergencias producidas por fallas totales o parciales de un sistema computacional.

Banco de datos: Colección de archivos de datos, de tipo histórico, utilizados para consultas específicas de algún tema en particular.

Base de datos relacional: Colección de datos organizada y relacionada, para evitar duplicaciones y permitir la obtención de datos combinados, satisfaciendo la necesidad de usuarios con diferentes necesidades de información.

BIOS (Basic Input/Output System): Sistema de entrada/salida básico. Código contenido por una computadora que proporciona un nexo entre el hardware y el sistema operativo. Generalmente contenido en un chip insertado en el motherboard. (o placa madre)

Bit (BInary digiT): Dígito binario. Unidad básica de información utilizada en un sistema de numeración binario. Un bit sólo puede ser cero o uno.

C

C: Lenguaje de programación usado principalmente para la programación de sistemas o sofisticadas aplicaciones.

C++: Lenguaje de programación orientado a objetos, basado en el lenguaje C.

Cache: Ver Memoria cache.

Circuito integrado: Circuito electrónico miniaturizado, capaz de realizar todas las funciones de uno convencional. Puede contener numerosos transistores, diodos, condensadores y resistencias que se fabrican y colocan en un simple chip.

Cliente: Programa que demanda servicios de otra computadora llamada servidor, y se hace cargo de la interacción necesaria con el usuario.

Código fuente: Programa escrito en un lenguaje de programación de alto nivel por un programador. Es solo un archivo de texto simple que contiene la secuencia de operaciones que la computadora deberá ejecutar, en una forma simple de entender por una persona que sepa programar en dicho lenguaje.

Código objeto: Programa expresado en lenguaje de máquina (ceros y unos), de manera que pueda ser ejecutado por una computadora.

Compatibilidad: Habilidad de usar sistemas y dispositivos de una computadora en otra, sin requerir cambios.

Compilador: Programa que traduce instrucciones escritas en un lenguaje de programación de alto nivel a un lenguaje de máquina.

Compilar: Generar un programa en lenguaje de máquina a partir de un lenguaje de programación de alto nivel.

Computadora: Una computadora es un sistema digital con tecnología microelectrónica capaz de procesar datos a partir de un grupo de instrucciones denominado programa. La estructura básica de una computadora incluye microprocesador (CPU), memoria y dispositivos de entrada/salida (E/S), junto a los buses que permiten la comunicación entre ellos.

Consola: Interfaz de comandos de un sistema operativo que permite el envío de ordenes a la computadora a través del teclado.

Correo electrónico: Servicio de intercambio de mensajes entre usuarios, que puede incluir texto y elementos multimedia.

D

Dato: Representación de un hecho o idea que puede ser manipulado y al cual se le puede asignar un significado.

Debugging: Ver Depuración.

Default: Ajustes por defecto. Lo que pasará si no se modifica nada.

Depuración: Detección, localización y eliminación de errores en un programa. También llamado debugging.

Diagrama de flujo: Representación gráfica de los tipos y secuencia de operaciones de un programa o proceso.

Dirección IP: Número que identifica cada una de las computadoras que se encuentran conectadas a Internet. Es única a nivel mundial. En su versión IPv4 (la más común todavía), la forman 4 conjuntos de números binarios, o su representación decimal (ej. 212.15.82.22)

Dominio: El nombre de dominio es un identificador único a través de la cual las computadoras se vinculan a Internet (por ej. para identificar sitios web y direcciones de correo electrónico). El sistema es jerárquico permitiendo la definición de subdominios de un dominio existente. A veces coloquialmente (y de modo incorrecto) se utiliza para referirse a las "direcciones Web") DNS (Domain

Name System): Sistema que almacena y asocia diferentes tipos de información con nombres de dominio. Su función más importante es traducir el nombre de dominio a una dirección IP.

E

Emulación: Proceso mediante el cual una computadora se hace funcionar como si fuera otra, para aceptar el mismo tipo de datos, ejecutar los mismos programas y obtener iguales resultados.

Extensiones de archivos: Es una cadena de caracteres anexada al nombre de un archivo, usualmente antecedita por un punto. Su función principal, es diferenciar el formato del archivo, de modo que el sistema operativo disponga el procedimiento correspondiente para ejecutarlo o interpretarlo.

F

FAQ (Frequently Asked Questions): Documentos informativos que recogen las respuestas a las preguntas formuladas más frecuentemente por los usuarios de un servicios determinado.

FORTRAN (FORmula TRANslator): El primer lenguaje de programación de alto nivel, diseñado para realizar cálculos matemáticos.

Firmware: Secuencia de comandos básicos, embebidos dentro del hardware. Generalmente estos comandos están en las memorias ROM.

Formato de archivo: Estructura de un archivo que define la forma en que se guarda y representa la información que contiene en la pantalla o en la impresora. El formato puede ser muy simple y común, como el de los archivos guardados como texto ASCII puro, o puede ser muy complejo e incluir varios tipos de instrucciones y códigos de control utilizados por programas, impresoras y otros dispositivos o el modo de compresión de los datos, como algunos formatos gráficos. En MS-DOS la extensión del nombre del archivo suele indicar el formato del archivo. Entre los ejemplos se cuentan el formato RTF (Rich Text Format), DCA (Document Content Architecture), PICT, DIF (Data Interchange Format), DXF, TIFF (Tag Image File Format) y EPSF (Encapsulated PostScript Format).

Freeware: Programa de uso sin costo siempre que se respeten las condiciones del propietario del mismo. No debe confundirse con Free Software.

Free Software: ver Software Libre.

FTP (File Transfer Protocol): Protocolo de transferencia de archivos de una computadora a otra a través de una red (usado en Internet).

G

Gigabyte (GB): 1.024 Megabytes, o aproximadamente mil millones de bytes (1.024 x 1.024 x 1.024 bytes).

GNU: Proyecto iniciado en 1984 por Richard Stallman que en la actualidad brinda la posibilidad de resolver, casi, la totalidad de los problemas de tratamiento informático con software libre. Esto incluye desde juegos hasta el núcleo

del sistema operativo. El núcleo más conocido del proyecto GNU es Linux. Las siglas GNU significan GNU's Not Unix (GNU No es Unix).

GUI (Graphical User Interface): Interfaz Gráfica de Usuario. Diseño amigable para la parte de un programa que interactúa con el usuario, basado en el uso de íconos y ventanas para representar sus distintas funciones. Generalmente asociada con el uso del dispositivo apuntador (ratón).

H

Hardware: Los componentes físicos de la computadora, así como sus periféricos.

Hipertexto: Sistema de organización y consulta de la información de manera no secuencial. La información se relaciona mediante enlaces que permiten vincular entre sí documentos o partes de documentos a través de "saltos".

Hipervínculo: Conexión en distintos puntos de una página de Internet, que lleva a otro punto determinado del mismo sitio o de otro dentro de la red.

HTML (HyperText Markup Language): Lenguaje de composición de páginas de hipertexto para la WWW.

HTTP (HyperText Transfer Protocol). Protocolo de transferencia de hipertexto. Es el protocolo que permite navegar por la WWW.

I

Ícono: Símbolo que representa un programa, archivo o aplicación y que sirve para ejecutar al mismo.

Información: Es el resultado del procesamiento de datos. Todo aquello que permite adquirir cualquier tipo de conocimientos.

Informática: Es la ciencia del tratamiento automático de la información mediante una computadora. La informática es un amplio campo que incluye los fundamentos teóricos, el diseño, la programación y el uso de las computadoras (ordenadores).

Instrucción: Conjunto de caracteres que especifica una operación a realizarse y el valor o ubicación de uno o más operandos requeridos.

Inteligencia artificial: Programas diseñados para que su funcionamiento imite los procesos humanos de toma de decisiones y para que aprenda de los eventos pasados.

Interfaz: Conexión entre dos componentes de hardware, entre dos aplicaciones o entre un usuario y una aplicación. También llamada por el término en inglés interfase.

Internet: Red mundial de computadoras conectadas a través del protocolo TCP/IP.. Es la más grande e importante red de redes interconectadas a través de routers. .

Intranet: Denominación utilizada para referirse a la red interna de una empresa o institución.

K

Kilobyte (KB): Medida de información. Contiene 1.024 bytes.

L

LAN (Local Area Network): Red de área local. Es la forma en la cual se interconectan computadoras ubicadas en un mismo lugar a través de un cable de red.

Lenguaje de programación: Conjunto de sentencias utilizadas para escribir secuencias de instrucciones para ser ejecutadas en una computadora.

Lenguaje de programación de alto nivel: Lenguaje de programación cercano a la notación utilizada en problemas o procedimientos. Por ejemplo FORTRAN, BASIC, C, PASCAL o Logo.

Lenguaje de programación de bajo nivel: Lenguaje de programación orientado a la máquina. Como los lenguajes de máquina y ensambladores.

Lenguaje ensamblador: Lenguaje de programación simbólico de bajo nivel. Tiene una correspondencia uno a uno en las instrucciones y formato de datos con el lenguaje de máquina. Entre más cercano es el lenguaje de programación al lenguaje de máquina, más rápidas resultan las aplicaciones, pero los programas resultan más difíciles de entender para un programador.

Linux: Núcleo o kernel de sistema operativo compatible con UNIX®, que se puede utilizar en casi cualquier plataforma de computadora sin pagar costo de licencia y con libre acceso y modificación de su código fuente. Muchas veces se utiliza erróneamente el término para nombrar a todo el sistema operativo.

Lisp: Lenguaje de programación funcional. Su nombre deriva del término "procesamiento de listas" en inglés ("List Processing"). Fue un lenguaje originalmente utilizado en inteligencia artificial.

Login: Acción de conectarse a un sistema ingresando un nombre de usuario y una contraseña.

Logo: Lenguaje de programación creado por Wally Feurzeig y Seymour Papert. Basado en el lenguaje Lisp, fue concebido como un lenguaje apropiado para el aprendizaje y por eso está traducido a muchos idiomas. Muy usado cuando se empezaron a introducir computadoras en los centros educativos. Su característica más conocida es el uso de los llamados gráficos de tortuga, un sistema gráfico muy intuitivo encaminado a la exploración de la geometría.

M

Macro: Instrucción de un programa fuente que realiza un conjunto de operaciones en otro programa que lo contiene.

Megabyte (MB): Medida de información equivalente a 1.024 kilobytes.

Memoria: Almacenamiento primario de una computadora, como la RAM o la ROM.

Memoria auxiliar o secundaria: Memoria que suplementa la memoria principal o primaria, generalmente utilizando parte del disco duro.

Memoria caché: Un área de memoria de alta velocidad en el procesador donde se almacenará la información de uso más frecuente. Es una porción relativamente

pequeña de memoria, muy rápida y reservada para el almacenamiento temporal de datos o instrucciones que el procesador va a utilizar próximamente.

Memoria principal: Lugar en el cual se almacenan datos e instrucciones en una computadora antes y durante su ejecución.

Memoria virtual: Una técnica de administración de memoria que permite utilizar un espacio del disco duro como si se tratase de memoria RAM. Esta técnica proporciona a las aplicaciones la posibilidad de utilizar más memoria de la que el sistema dispone.

Menú: Lista de opciones mostrada sobre una pantalla de las cuales el usuario puede seleccionar.

Multimedia: Forma de presentar información a través de una computadora, usando texto, gráficos, sonido o video.

Multiprocesamiento: Técnica para ejecutar dos o más secuencias de instrucciones simultáneamente en una misma computadora. Se necesita mas de un procesador (máquinas grandes) o microprocesadores especiales.

Multitarea: Ejecución simultánea, en una computadora, de más de un programa. Las tareas se alternan en la ejecución a tanta velocidad que el usuario no llega a percibir su interrupción.

N

Net: Apócope de Internet.

Nodo: Computadora o cualquier otro dispositivo conectado a una red.

O

OCR (Optical Character Recognition): Reconocimiento óptico de caracteres. Técnica de registro y lectura de caracteres u otros símbolos en un archivo de imagen de mapa de bits.

Offline: Equipos o dispositivos que no están en comunicación directa o apagados.

Online: Equipos o dispositivos que están en comunicación directa o encendidos.

Ordenador : Término usado en España y en algunos países de latinoamérica para referirse a una computadora.

P

Palabra reservada: Palabra que no puede usarse para propósitos distintos de los establecidos por el programa en uso.

PASCAL: Lenguaje de programación especialmente apto para construir programas estructurados. Diseñado por Niklaus Wirth en el instituto ETH Zürich (Suiza) alrededor de 1968, la primera implementación estuvo disponible en 1970.

Password: Contraseña utilizada para ingresar en una red o en un sistema de manera segura. Conjunto de caracteres alfanuméricos requeridos pra accedr a una determinada red, sistema, aplicación o recurso.

Placa madre (o Motherboard): Placa de circuito impreso donde se instalan el procesador, la ROM, la RAM, los buses y otros elementos de una computadora.

Plugin (o plug-in): Es un programa que interactúa con otro programa para aportarle una función o utilidad específica, generalmente muy específica. Este programa adicional es ejecutado por la aplicación principal. Los plugins típicos tienen la función de reproducir determinados formatos de gráficos, reproducir datos multimedia, codificar/decodificar emails, filtrar imágenes de programas gráficos, etc.

Procesador de textos: Programa que permite la manipulación de textos con formato y que permite generar archivos que conserven el estilo realizado.

Procesamiento de datos: Secuencia sistemática de operaciones realizadas sobre datos para obtener un resultado deseado.

Procesamiento en tiempo real: Técnica de procesamiento en que la actualización de los datos afectados por un evento se realiza a medida que sucede el evento causante.

Proceso: Manipular datos o realizar otras operaciones de acuerdo a un programa.

Programación: Se llama programación a la creación de un programa informático, un conjunto concreto de instrucciones que una computadora u otro dispositivo informático puede ejecutar. El programa se escribe en un determinado lenguaje de programación, (con dificultad se puede escribir directamente en lenguaje de máquina). Un programa puede estar dividido en diversas partes, que pueden estar escritas en lenguajes distintos.

Programa: Secuencia de instrucciones que dirige a la computadora a realizar operaciones específicas para obtener un resultado deseado.

Programa de control: Programa del sistema operativo que lee instrucciones de control.

Programa fuente: Ver código fuente.

Programa intérprete: Programa de computadora que procesa instrucciones de lenguajes de programación de alto nivel instrucción por instrucción, determinando las operaciones requeridas y haciendo que la computadora las realice.

Programa objeto: Ver código objeto.

Programador: Persona que define la solución a un problema y escribe las instrucciones requeridas por una computadora para llevar a cabo esa solución. Un programador que también realiza análisis de sistemas y diseño, suele llamarse Analista/Programador.

Protocolo: Definición del sistema de comunicación de una computadora. Acuerdo entre diferentes sistemas para trabajar conjuntamente bajo un estándar común. Conjunto de normas que permiten estandarizar un procedimiento repetitivo.

R

RAM (Random-Access Memory): Memoria primaria de una computadora. En las PCs es accesible por el procesador a través del puente norte del chipset.

Recuperación: Habilidad para reiniciar el proceso, ante una falla del equipo, sin pérdida de datos o resultados.

Red: Interconexión de una o más computadoras a través de hardware y software.

Robot: Máquina programable que puede realizar varias tareas físicas bajo el control de un programa.

Router: Dispositivo que se encarga de gestionar y organizar el tránsito de datos entre diferentes redes.

S

Salida: Output. Resultado del procesamiento.

Servidor: Computadora o programa que proporciona recursos y servicios a las computadoras conectadas a una red y al mismo tiempo gestiona el uso de esa red..

Simulación: Representación del funcionamiento de un sistema por otro. Por ejemplo, la representación de un sistema físico por un modelo matemático.

Sistema: Conjunto de elementos interrelacionados que trabajan juntos para obtener un resultado deseado.

Sistema de Archivo: Un sistema de archivos consta de tipos de datos abstractos, que son necesarios para el almacenamiento, organización jerárquica, manipulación, navegación, acceso y consulta de datos. La mayoría de los sistemas operativos poseen su propio sistema de archivos. Los sistemas de archivos son representados ya sea textual o gráficamente utilizando gestores de archivos o shells. En modo gráfico a menudo son utilizadas las metáforas de carpetas (directorios) conteniendo documentos, archivos y otras carpetas. Un sistema de archivos es parte integral de un sistema operativo moderno. Los sistemas de archivos más comunes utilizan dispositivos de almacenamiento de datos que permiten el acceso a los datos como una cadena de bloques de un mismo tamaño, a veces llamados sectores, usualmente de 512 bytes de longitud. El software del sistema de archivos es responsable de la organización de estos sectores en archivos y directorios y mantiene un registro de qué sectores pertenecen a qué archivos y cuáles no han sido utilizados. En la realidad, un sistema de archivos no requiere necesariamente de un dispositivo de almacenamiento de datos, sino que puede ser utilizado también para acceder a datos generados dinámicamente, como los recibidos a través de una conexión de red

Sistema de manejo de base de datos: Software que maneja la organización, localización, catalogación, almacenamiento, recuperación y mantención de datos en una base de datos.

Sistema numérico binario: Sistema de numeración de base 2, es decir, que sólo usa dos dígitos. Por lo general los dígitos utilizados son 0 y 1.

Sistema numérico hexadecimal: Sistema numérico de base 16, generalmente usando los dígitos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Sistema operativo: Programa de control que dirige el hardware de una computadora. Por lo general es, en realidad, una colección de programas que interactúan juntos.

Software: Programas escritos en un lenguaje que la computadora entiende y puede ejecutar para realizar una tarea.

Software de aplicación: Programas que realizan las tareas específicas de procesamiento de datos.

Software libre: Es el que respeta la libertad del usuario, ateniéndose a las 4 libertades que plantea la Free Software Foundation: De usarlo para el fin que se quiera; De realizar copias; De modificarlo para ajustarlo a nuestro gusto; De distribuir las mejoras. Adicionalmente se suele decir que la única restricción es que cada uno que reciba ese software, debe heredar esas libertades.

T

TCP/IP: Conjunto de protocolos que rigen la transmisión de información en Internet.

Terminal: Dispositivo en un sistema o red de comunicación en el cual los datos pueden ingresarse o salir, pero no procesarse.

Terminal inteligente: Es una terminal con capacidad de procesamiento en sí misma.

Testing: La prueba de un programa o un sistema para asegurar que funciona adecuadamente.

Transmisión asincrónica: Método de transferencia de datos en el cual las unidades emisoras y receptoras no tienen igual velocidad.

Transmisión digital: Transferencia eléctrica de datos por señales discretas.

Transmisión sincrónica: Método de transferencia eléctrica en el cual las velocidades de entrada y salida son iguales.

U

Unidad central de procesamiento (CPU): La Unidad Central de Proceso (UCP) o CPU (siglas de Central Processing Unit) es la unidad donde se ejecutan las instrucciones de los programas y se controla el funcionamiento de los distintos componentes de la computadora. Suele estar integrada en un chip denominado microprocesador.

V

Ventana: Parte de la pantalla usada independientemente del resto.

Virtual: Se dice de la representación en una computadora de algo que no tiene existencia material o no está presente en ese lugar.

Virus informático: Programa con finalidades destructivas o de interferencia del funcionamiento correcto de los sistemas informáticos. Volúmen: Entidad física utilizada para almacenar datos e instrucciones. Puede ser cinta o un disco magnético.

W

WAN (Wide Area Network): Conexión entre varias redes de área local, físicamente distantes. El ejemplo más conocido es Internet.

WLAN (Wireless Local Area Network): Red de área local inalámbrica.

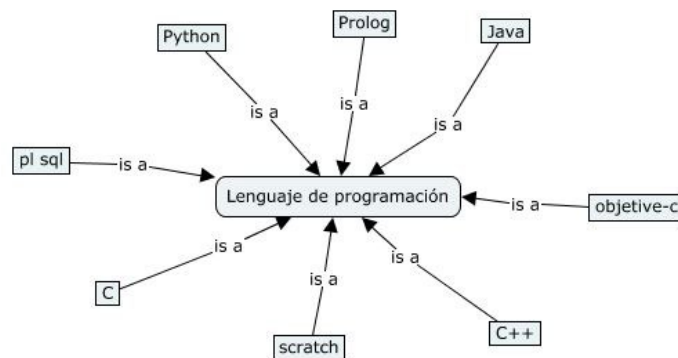
Wiki: Forma de sitio web en donde se acepta que usuarios creen, editen, borren o modifiquen el contenido de una página web, de una forma interactiva, fácil y rápida. Estas facilidades hacen de una wiki una herramienta efectiva para el trabajo colaborativo. Una wiki permite crear y mejorar las páginas de forma

instantánea por medio de una interfaz muy simple , dando una gran libertad al usuario

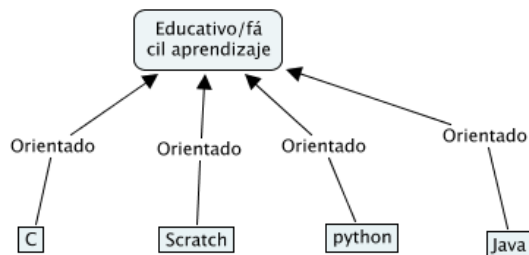
World Wide Web (www). Sistema de organización de la información de Internet a través de enlaces hipertexto. En sentido estricto es el conjunto de servidores que emplean el protocolo HTTP.

Aquí se muestran también algunas ontologías sobre los lenguajes de programación que se usarán.

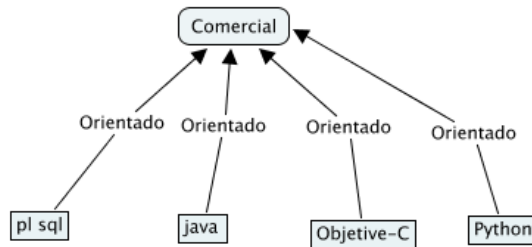
Estos son los lenguajes que vamos a tratar a lo largo del proyecto:



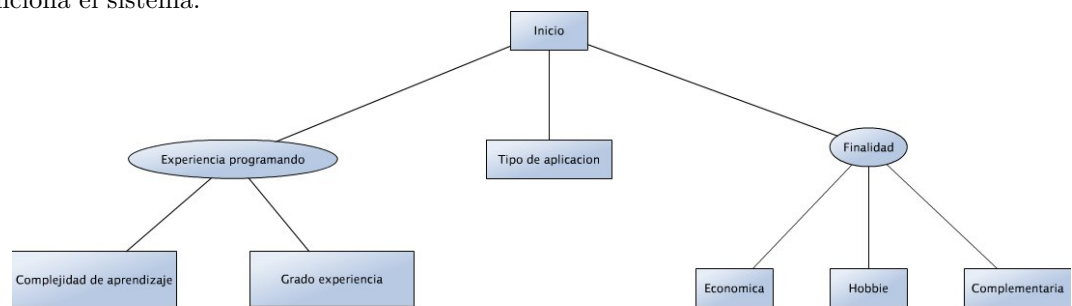
Aquí mostramos los lenguajes que son los indicados cuando queremos aprender, o para el ámbito educativo de niños o personas muy mayores sin conocimientos:



Por otra parte, estos son aquellos que tienen una orientación mas comercial, para aquellos que quieran hacer de la programación su oficio, o al menos un negocio relativamente rentable.



Conocimientos Estratégicos Especifican qué hacer, dónde y por qué hacerlo, es decir, fijan la secuencia de pasos que el SE deberá seguir para ejecutar su tarea. A continuación se especifican una serie de pseudoreglas que definen como funciona el sistema.



Conocimientos Tácticos De acción u operativos, especifican cómo y cuándo el SE puede añadir a sus conocimientos genéricos información actual acerca del caso. En este caso, se van a usar pseudoreglas, que representan en un lenguaje mas natural como funcionará nuestro sistema internamente. Consisten en una serie de condiciones que si se cumplen disparan una acción, que puede ser poner un elemento a verdadero que en un futuro será la condición de otra pseudoregla, o bien nos sugiere el lenguaje de programación que ha concluido debemos usar/aprender.

R1	
Si	tiene experiencia
Entonces	Finalidad

R2	
Si	no tiene experiencia
Entonces	NoExp

R3	
Si	NoExp
Y	Aprender fundamentos
Entonces	Aprender

R4	
Si	NoExp
Y	Finalidad concreta
Entonces	Finalidad

R5	
Si	Aprender
Y	curva de aprendizaje muy fácil o lenguaje para niños o ancianos
Entonces	<u>AppInventor</u> o <u>Scratch</u>

R6	
Si	Aprender
Y	curva de aprendizaje fácil
Entonces	<u>Python</u> o <u>RubyonRails</u>

R7	
Si	Aprender
Y	curva de aprendizaje normal
Entonces	<u>Java</u> o <u>.Net</u>

R8	
Si	Aprender
Y	curva de aprendizaje difícil
Entonces	<u>C</u> o <u>C++</u>

R9	
Si	Finalidad
Y	Finalidad es hobby o complementaria
Entonces	complementaria

R10	
Si	Complementaria
Y	Aprender paradigma
Entonces	AprenderParadigma

R11	
Si	AprenderParadigma
Y	Programación orientada a objetos
Entonces	<u>Java</u>

R12	
Si	AprenderParadigma
Y	Programación lógica
Entonces	<u>Prolog</u>

R13	
Si	AprenderParadigma
Y	Programación funcional
Entonces	<u>Haskell</u>

R14	
Si	Complementaria
Y	Aplicación específica
Entonces	Aplicación

R15	
Si	Aplicación
Y	Programación web
Entonces	Pweb

R16	
Si	Pweb
Y	Parte cliente
Entonces	<u>JavaScript</u>

R17	
Si	Pweb
Y	Parte servidor
Entonces	<u>JavaScript</u> o <u>Go</u> o <u>RubyonRails</u>

R18	
Si	Aplicación
Y	Videojuegos
Entonces	<u>C + +</u>

R19	
Si	Aplicación
Y	Aplicaciones móviles
Entonces	<u>Android</u> – > <u>Java</u> <u>iOs</u> – > <u>ObjectiveC</u>

R19	
Si	Aplicación
Y	Multiplataforma
Entonces	<u>Java</u>

R20	
Si	Aplicación
Y	Ámbito científico
Entonces	<u>Matlab</u> <u>Python</u>

R21	
Si	Aplicación
Y	Sistemas Expertos
Entonces	<u>Lisp</u> <u>Prolog</u>

R22	
Si	Aplicación
Y	Sistemas distribuidos
Entonces	<u>C</u>
R23	
Si	Aplicación
Y	Procesadores de lenguaje natural
Entonces	<u>Prolog</u>
R24	
Si	Aplicación
Y	Bases de Datos
Entonces	<u>SQL</u>
R25	
Si	Finalidad
Y	Finalidad es económica
Entonces	Económica
R26	
Si	Económica
Y	Lenguaje mejor pagado
Entonces	<u>PL – SQL</u>
R27	
Si	Económica
Y	Lenguaje con mayor demanda
Entonces	<u>Java o C# o .NET + Frameworkespecifico</u>

5. Implementación

Tras haber llevado a cabo todo el proceso de conceptualización toca el de implementación. Primero se mostrará un primer boceto de como se ha planteado el sistema de manera genérica haciendo uso de una herramienta de maquetado, y posteriormente se analizara la tecnología utilizada para el desarrollo de la propia interfaz y finalmente la interfaz propiamente dicha.

5.1. Interfaz Gráfica de Usuario

La interfaz de usuario es una parte muy importante de cualquier programa informático. Es la parte que el usuario aprecia de manera directa, y muchas veces sirve para juzgar cualquier sistema antes de verlo en funcionamiento. Algunos datos interesantes son "La interfaz constituye entre el 47 % y el 60 % de las líneas de código" (McIntyre) o "Actualmente más del 70 % del esfuerzo de desarrollo de las aplicaciones interactivas está dedicado a la interfaz" (Gartner Group).

5.2. Herramienta de Diseño de Interfaz seleccionada

Se hará uso de herramientas de prototipado o maquetación, en este caso se ha usado Balsamic Mockups. La ventaja que ofrece esta herramienta es permitir un prototipado muy rápido similar al que se haría dibujando sobre un papel directamente a lápiz, solo que con elementos de arrastrar y soltar, en cortos periodos de tiempo se puede hacer prototipos de la interfaz de forma eficaz y rápida.

Esto es muy importante, ya que si en algún momento, detectamos que algo podría fallar en nuestra interfaz, o incluso, a veces, en la funcionalidad de nuestra propia aplicación, los errores se detectarán en una fase muy temprana del proyecto, y corregir dichos costes es mucho más barato y rápido que una vez la aplicación se ha desarrollado por ejemplo.

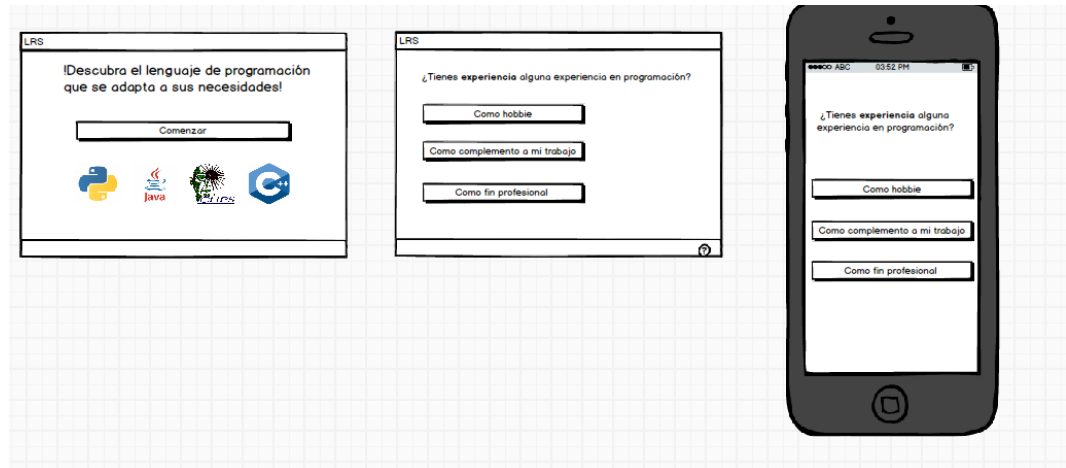
Balsamic Mockups además es una aplicación ligera, es decir, se puede trabajar directamente en la propia web, sin necesidad de descargar ninguna aplicación de escritorio, por lo tanto es aún más rápido y cómodo para el propio usuario.

5.3. Diseño de la Interfaz Gráfica de Usuario del Sistema Experto

Como al principio no se sabía la plataforma para la que se iba a desarrollar se hicieron dos versiones, una para móvil y otra de aplicación de escritorio.

Es imprescindible que esta interfaz sea intuitiva y fácil de entender para el usuario. Además es un proceso que debe realizarse junto con este, mostrando primero bocetos y prototipos y permitiéndole (en la medida de lo posible) hacer cambios o sugerir mejoras.

Nuestro sistema experto trata de recomendar un lenguaje a partir de una determinada situación, esta situación será la de el usuario en cuestión, una de las pocas formas que podemos recoger dicha información es mediante preguntas y respuestas, preguntaremos al usuario unas determinadas cuestiones de relevancia para la decisión para la recomendación del lenguaje, una vez tengamos la suficiente información, recomendaremos el lenguaje adecuado junto con una descripción del mismo. A continuación, presentamos en Mockup un primer prototipado de la interfaz:



Tanto la aplicación de escritorio como en la aplicación móvil, el usuario irá respondiendo una serie de cuestiones pulsando los botones con unas respuestas predeterminadas, de esta forma el sistema experto conocerá la situación del usuario con el que podrá recomendar el lenguaje que más se ajusta a sus necesidades.

6. Conclusiones y trabajo futuro

Como conclusiones, destacar que se ha conseguido llevar a cabo el proyecto previsto, proporcionando una herramienta que de manera sencilla aconseja o recomienda a un público muy variado el lenguaje de programación que mejor se adapte a sus necesidades. De la evaluación del estado del arte se ha podido comprobar que no existe ningún trabajo similar (al menos serio). Casi toda la información relacionada con este tipo de sistemas que una persona sin un alto grado de experticidad puede tener a su disposición esta recogida en blogs y es mayoritariamente de opiniones de usuarios amateurs. Por el contrario este proyecto cuenta con la participación de grandes expertos y con las fuentes mas fiables de las que se ha podido disponer.

Se ha hecho pruebas para comprobar que todos las posibles soluciones son alcanzables, por lo tanto, las reglas inicialmente propuestas se cumplen, dada la falta de tiempo, no se podrá contrastar el resultado final con la opinión de los expertos. Se mostrarán 2 casos de prueba concretos para poder observar un ejemplo de ejecución de nuestro Sistema Experto.

Leonardo es un estudiante de 2º curso de Ingeniería Informática, tiene ya unos conocimientos sólidos de los fundamentos de la programación y le gustaría dedicarse al mundo del desarrollo de Videojuegos, pero sin embargo no sabe cual es el lenguaje recomendado para este campo, que es C++.

Prueba 1	
Do you have a solid knowledge of Programming Skills?	Yes
Do you wish to improve your Programming Skills?	Yes
Do you wish to learn new programming paradigms?	No
Do you wish to program in a specific area like Web, Games, etc?	Yes
Do you wish to program in order to make Websites?	No
Do you wish to program Video-Games?	Yes
Use C++ in order to program Video-Games	—

Alberto es una persona con la carrera acabada y está interesado en llevar una buena vida gracias a los conocimientos que dispone, para eso decide especializarse en algún lenguaje que le permita cobrar bastante.

Prueba 2	
Do you have a solid knowledge of Programming Skills?	Yes
Do you wish to improve your Programming Skills?	No
Learn PL-SQL in order to get a well-paid job	—

Por supuesto al ser un mercado tan cambiante el mercado tecnológico con el que tan ligados están los lenguajes de programación, este sistema necesita de una actualización continua. Además estas actualizaciones se podrían automatizar extrayendo de alguna fuente de información fiable que lenguajes se usan mas y para que, aunque este seria un cambio completo de orientación del proyecto.

Referencias

1. Kish Waukee College. Expert System for Choosing a Programming Language
2. Vladyslav Kruglyk and Michael Lvov. Kherson State University. Ukraine. <http://ceur-ws.org/Vol-848/ICTERI-2012-CEUR-WS-paper-37-p-188-198.pdf>
3. Lisa Eadicicco Business Insider. (2014). <http://www.businessinsider.com/best-tech-skills-resume-ranked-salary-2014-11>
4. Yoshitaka Shiotsu. (2014). <https://www.odesk.com/blog/2014/03/web-development-101-top-web-development-languages-2014/>
5. Kasia Mikoluk. (2013). <https://blog.udemy.com/best-programming-language/>

A. Entrevista David Villa Alises

Documento de adquisición del conocimiento	Número:
Fecha: 9/04/2015	
Hora: 18:00	
Lugar: ESI Ciudad Real	
Asistentes: Juan Carlos Fernández Durán - David Villa Alises	
Fuentes de Conocimiento:	
Soporte:	
Conocimiento anterior a la entrevista:	
Objetivos: Entrevistar al experto	
Modo:	
Planteamiento de la sesión:	
Resultados de la sesión:	
Plan de análisis:	
Resultados de análisis:	
Tiempo de análisis: 45 minutos	
Recursos empleados:	
Otros datos complementarios:	
Comentarios:	

¿Como docente con gran experiencia y experto en lenguajes imperativos, que lenguaje recomendaría a alguien que quiere empezar con la programación como hobby?

Sin conocimientos técnicos relevantes recomendaría Scratch o AppInventor, sin embargo, para una persona con conocimientos técnicos o matemáticos recomendaría Ruby o Python ya que los lenguajes interpretados son menos ?encorsetados? y permite escribir código más libremente ya que no está fuertemente tipado.

Es importante la interactividad, poder interaccionar con el programa desde una Shell, como Python y Ruby, se podría decir que tu programa funciona como espera que funcione.

¿Y para dedicarse al ámbito profesional?

Ruby o Python son los adecuados para empezar, su librería estándar se parece mucho a la librería estándar de C o C++, por lo tanto, más tarde al dar el salto a C o C++ es más fácil, además existe mucha documentación de librería estándar C++, tan solo viendo la signatura puedes usarla, todo es bastante parecido y es altamente recomendable.

Un ingeniero informático también debe conocer C, puedes hacer un compilador para cualquier cosa, lenguaje simple, compilador simple, toda persona que se quiera considerar un buen profesional debería conocer C, pasar de Python a C es bastante fácil.

¿Cuál cree que es el lenguaje con la curva de aprendizaje mas suave?

Recomendaría los Koans de Ruby y Python para aprender rápido, estos lenguajes te ocultan muchos detalles al comienzo, por lo tanto es más fácil empezar.

Y de los más usados o con mayor aplicación, ¿Cuál tiene la curva de aprendizaje mas pronunciada?

Python es muy versátil aunque te puedes encontrar de todo pero ya es forzar, por ejemplo el uso de PHP para aplicaciones de escritorio.

Python tiene una muy buena relación aceptable que cubra todo el espectro, LUA o Boo son interesantes también, aunque sean más enfocados a lenguaje de sistemas o de servidor

¿En el caso de niños mas pequeños, que vayan al colegio o estén en enseñanza secundaria, que lenguaje cree que deberían aprender?

Scratch o AppInventor es más amigable por el tema de arrastrar y soltar y elegir de un menu

Otra opción interesante son los lenguajes gráficos no tienen la suficiente expresividad aún, tienen mucho camino por recorrer pero sin duda es una opción que cobrará una mayor importancia en el futuro y a tener en cuenta.

Para un profesional que solo conoce un lenguaje imperativo funcional como C o Pascal, ¿qué lenguaje recomendaría para introducirlo a la programación orientada a objetos?

Aquel que soporte ambos paradigmas, aunque POO se puede simular en C o Pascal, no es elegante, también depende de la gente purista, por ejemplo Java lo consideran basado en objetos, cualquier lenguaje interpretado es más simple para hacer experimentos.

Bien, ahora le vamos a realizar una serie de cuestiones para que nos identifique el lenguaje que usted considera mas apropiados para las siguientes aplicaciones:

-Programación web.

Para el lado del cliente, sin duda, JavaScript.

Por el lado del servidor, siempre está cambiando la tecnología, por lo tanto es difícil recomendar uno, últimamente están cobrando importancia los servidores tipo node con JavaScript, sin embargo me parece algo forzado para servidor ya que es interpretado, por lo tanto no es ligero y en el servidor debes poder manejar cualquier cosa.

El lenguaje Go también esta cobrando más importancia, desarrollado por los creadores de C. Actualmente recomendaría Ruby on Rails, pero se debe tener cuidado ya que las tecnologías aquí cambian muy rápidamente.

-Videojuegos.

C++ debido a la eficiencia, sobre todo para el Motor de Juego, gráficos etc, sin embargo, otras cosas más externas, como IA, recomendaría otros lenguajes

como Python o LUA.

-Aplicaciones móviles.

Más que recomendar lenguajes, solo existen 2 lenguajes que son los impuestos por la propia plataforma, siendo iOS y Android, serían los lenguajes Objective-C y Java.

-Aplicaciones multiplataforma.

Si deseas multiplataforma, recomendaría Java

.NET no tiene tanta presencia en el mercado pero se recomienda también, su bytecode está mejor pensado, pero .NET debido a la restricciones de la plataforma no está siendo todo lo potente que podría llegar a ser.

-Ámbito Científico / Matemático

Python le está comiendo el terreno a MatLab ya que es un lenguaje que permite hacer lo mismo y además es un lenguaje de propósito general.

Vayamos ahora con la parte de sistemas distribuidos o empotrados. Un caso más específico de sistema empujado podría ser uno que requiriese de tiempo real ¿Cuál sería la recomendación en este caso?

Lenguaje C es sin duda la opción a elegir, podría ser interesante también Go pero este lenguaje aún le queda camino por recorrer.

B. Entrevista Pascual Julián Iranzo

Documento de adquisición del conocimiento	Número:
Fecha: 26/03/2015	
Hora: 12:00	
Lugar:	
Asistentes: Victor Gualdrás - Pascual Julián	
Fuentes de Conocimiento:	
Soporte:	
Conocimiento anterior a la entrevista:	
Objetivos: Entrevistar al experto	
Modo:	
Planteamiento de la sesión:	
Resultados de la sesión:	
Plan de análisis:	
Resultados de análisis:	
Tiempo de análisis:	
Recursos empleados:	
Otros datos complementarios:	
Comentarios:	

¿En qué casos cree que sería mas recomendable usar un lenguaje de programación lógico frente a uno imperativo, y uno funcional frente a uno también imperativo?

Los lenguajes de programación declarativa (ya sean lógicos o funcionales) son más adecuados en la solución de todos los problemas que requieran manipulación simbólica o algún tipo de inferencia lógica.

¿Qué lenguaje recomendaría a alguien que no tiene conocimientos de programación?

Le recomendaría un lenguaje de programación lógico como Prolog (que no es fuertemente tipado) o un lenguaje funcional como Haskell (que implementa un sistema de tipos muy rico, pero es simple de aprender y usar) para que experimentará la emoción de programar y resolver problemas (incluso complejos) sin las complicaciones sintácticas de los lenguajes imperativos y orientados a objetos.

Naturalmente, aunque estoy proponiendo el uso de un lenguaje declarativo como primer lenguaje de programación, eso no exime del aprendizaje de un lenguaje convencional en una segunda etapa de su formación.

¿Y a alguien que sabe programar pero únicamente conoce el paradigma imperativo?

Un ingeniero informático que únicamente conozca lenguajes convencionales (ya sean de corte imperativo u orientados a objetos) creo que puede tener una desventaja competitiva de cara al futuro que nos viene. Cada vez más, con la

llegada de las máquinas multi-core, la computación concurrente y paralela va a ser un tema primordial y los modelos de cómputo de los lenguajes declarativos se adaptan bien a la computación paralela (cosa que no ocurre con los lenguajes convencionales). Muestra del interés que despiertan los lenguajes declarativos y, en especial, los lenguajes funcionales, es que Java 8 incorpora nuevas clases que permiten un estilo de programación funcional en Java. También hay un auge de los lenguajes híbridos tipo Scala. Por otra parte, no olvidemos que, aunque de dominio específico, SQL es un lenguaje declarativo.

Por todo ello, yo recomendaría el estudio de un lenguaje de corte lógico y otro de corte funcional. Los estándares en estos campos son, respectivamente, Prolog y Haskell.

Le vamos a mostrar una serie de aplicaciones que pueden hacer uso de los lenguajes declarativos. Elija en cada caso un lenguaje para cada uno.

-IA de videojuegos

La llamada lógica de videojuegos es, en el momento actual, muy rudimentaria y puede ser ventajosamente implementada mediante simples instrucciones procedimentales (if-then-else). No tengo noticias de lenguajes declarativos empleados con estos fines.

-Procesadores de lenguaje natural

Prolog se ha empleado exhaustivamente para el procesamiento del lenguaje natural. De hecho Prolog tiene su origen en este campo.

-Bases de datos

El lenguaje de dominio específico SQL para bases de datos relacionales y Datalog en el ámbito de las bases de datos deductivas.

-Sistemas Expertos Prolog o Lisp, que son lenguajes de propósito general, o lenguajes basados en reglas, especialmente adaptados para el desarrollo de sistemas expertos, como Clips (un lenguaje híbrido que combina los paradigmas lógico, procedural y orientado a objetos).

C. Entrevista Ismael Caballero

Documento de adquisición del conocimiento	Número:
Fecha: 25/03/2015	
Hora: 16:00	
Lugar:	
Asistentes: Juan Carlos - Victor - Pascual Julián	
Fuentes de Conocimiento:	
Soporte:	
Conocimiento anterior a la entrevista:	
Objetivos: Entrevistar al experto	
Modo:	
Planteamiento de la sesión:	
Resultados de la sesión:	
Plan de análisis:	
Resultados de análisis:	
Tiempo de análisis:	
Recursos empleados:	
Otros datos complementarios:	
Comentarios:	

¿Cual es su experiencia en el mundo laboral? 2 años trabajando en el departamento de I+D en Indra - Software Labs, evaluando la calidad de los productos software, tratando con los lenguajes Java y COBOL

¿Cual es su experiencia en la docencia de la UCLM? Primer curso de programación en C desde los 16 años, cursos en Pascal, en Java, en Visual Basic .NET

¿Cual es el lenguaje que actualmente tiene más demanda por las empresas? En España, Java principalmente, JavaScript en programación web FrontEnd y C#

¿Cual es el lenguaje que considera mejor pagado actualmente? PL-SQL debido al grado de especialización alto

¿Cual considera que es el lenguaje con salida más inmediata tras terminar los estudios? Es más adecuado conocer los framework específicos, como Struts 2, Spring, para grandes proyectos, los lenguajes acaban siendo circunstanciales.

¿Consideraría algún lenguaje mejor para hacer trabajo como freelance? Especializarte en un lenguaje como por ejemplo .NET y dedicarte a ello, pero ya debes meterte a un nivel de profundidad.

¿Existe actualmente algún lenguaje de alguna tecnología específica que considere interesante para orientar tu carrera laboral? Tal y como avanzan las cosas, hay 2 alternativas, un trabajo de desarrollo en un lenguaje de propósito general tipo Java o tipo C# o aprender un lenguaje de programación que encaje bien con el BigData.