

---

Sistema Experto de  
recomendación de  
lenguaje de programación

---

**Victor Gualdras de la Cruz**  
**Juan Carlos Fernández Durán**

## Table of Contents

1	Introducción .....	3
1.1	Planteamiento del Problema .....	3
1.2	Objetivos y Alcance .....	3
1.3	Los Expertos .....	4
1.4	Los usuarios finales del sistema .....	4
2	Estado del Arte .....	4
2.1	Conceptos y Procedimientos principales del área de aplicación del Sistema Experto .....	5
2.2	Conceptos y Procedimientos principales de Informática que serán utilizados .....	5
2.3	Revisión Bibliográfica .....	6
3	Estudio de viabilidad. Test de Slagle .....	7
3.1	Definición de las características .....	7
3.2	Asignación de pesos .....	8
3.3	Evaluación de la aplicación .....	8

## 1 Introducción

En esta sección realizaremos una introducción acerca del Sistema Experto de recomendación de lenguaje de Programación en el que quedará definido nuestro problema, definiendo a su vez el alcance que abarcará nuestro sistema experto y de cuáles son los objetivos que satisfará el mismo. Presentaremos los expertos de los cuales se obtendrá el conocimiento experto y a qué tipo de usuario estará dirigido nuestro 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 dedicamos nuestra vida diaria a este campo, entre ellos, los informáticos. Muchas veces, nos vemos en la situación de querer abordar un nuevo problema, queremos especializarnos en un campo específico, pero no conocemos las herramientas apropiadas para ello, en nuestro 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 empujados, la programación de un sistema experto o incluso un lenguaje que sea más didáctico para aprender a programar.

En base a unas características recomendaremos el lenguaje de programación que más adecuado al usuario según sus necesidades o preocupaciones. En este sistema experto nos centraremos 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 estudiaremos 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 nuestro 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 nuestro 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 presentaremos 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. Abordaremos brevemente cada uno de estos artículos con un breve resumen.

También introduciremos los conceptos más importantes tanto de nuestro 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 area 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 utilizaremos para construir nuestro 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, obtendremos 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 tendremos nuestras reglas y hechos, y por otra parte, tendremos 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 encontramos 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 estamos tratando, si es para una inteligencia artificial, una aplicación de escritorio, móvil, o web, tiene en cuenta para que plataformas queremos trabajar, si nos importa que sea multiplataforma...

Podemos 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 nuestro 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 utilizaremos 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] podemos 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 contamos con 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 video juegos 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

Vamos a analizar si el sistema 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. Nos serviremos de la leyenda [Table 1] mostrada a continuación para seguir el procedimiento:

Table 1. Leyenda

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

Procedamos 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.

**Table 2.** Categorías

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

**Table 3.** Identificación de las Características

<b>Pi</b>	Característica de la dimensión Plausibilidad
<b>Ji</b>	Característica de la dimensión Justificación
<b>Ai</b>	Característica de la dimensión Adecuación
<b>Ei</b>	Característica de la dimensión Éxito

**Table 4.** Tipos de características

<b>E</b>	Esencial
<b>D</b>	Deseable

**Tablas.** Mediante las siguientes tablas evaluaremos si nuestro sistema es viable.

**Table 5.** Dimension 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

**Table 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

Table 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

Table 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 calculamos el valor total de referencia del sistema:

$$VC = \frac{80.79358 + 74.42169 + 61.69024 + 62.84026}{4} = 69.93644$$

Unicamente nos queda calcular el Valor Normalizado. Valor Obtenido 69.93644  
- Valor Maximo Posible 76,21 Valor Real X - 100

$$X = \frac{69.94 * 100}{76,21} = 91.38$$

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

## References

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/>