

# Innovación y Emprendimiento

## Innovation and Entrepreneurship.

Autor 1: David Esteban Cediél Gómex

*Computación Blanda, Universidad Tecnológica de Pereira, Pereira, Colombia*

Correo-e: david.cediél@utp.edu.co

**Resumen—** El paper puesto a consideración es una apuesta técnica hacia la aplicación de tecnologías de computación blanda, con el propósito de servir a propósitos de carácter académico, y buscar su potencial aplicación en entornos de innovación y emprendimiento, soportados por la institución de educación superior y la plataforma EMPeCé, puesta a disposición por la Secretaría de Desarrollo Económico y Competitividad de la Alcaldía de Pereira.

**Palabras clave—** Investigación, Producto, Innovación, Tecnología, Informática, Emprendimiento.

**Abstract—** The paper put into consideration is a technical commitment towards the application of soft computing technologies, with the purpose of serving academic purposes, and seeking its potential application in innovation and entrepreneurship environments, supported by the institution of higher education and EMPeCé platform, made available by the Ministry of Economic Development and Competitiveness of the Municipality of Pereira..

**Key Word —** Research, Product, Innovation, Technology, Information Technology, Entrepreneurship.

## I. INTRODUCCIÓN

El presente paper (documento técnico estándar) está orientado al diseño de una iniciativa de proyecto o producto que permita aplicar los conocimientos adquiridos en la Universidad Tecnológica de Pereira, específicamente en las materias de Inteligencia Artificial, para llevar a cabo una propuesta que sirva como base para el desempeño académico, pero que vaya mucho más allá al concebir la solución a un problema de interés para la sociedad y para el ambiente académico y de proyección para la competitividad.

Para lograr lo anterior, se utilizará la plataforma EMPeCé, especialmente diseñada por la Secretaría de Desarrollo Económico y Competitividad de la Alcaldía de Pereira, la cual se pone a disposición en el marco de una iniciativa regional para promover el emprendimiento de base tecnológica.

EMPeCé es un sistema web basado en herramientas universales, integrando tecnologías estándar, como las siguientes: HTML5, CSS3, JavaScript, JQuery, Node.js, PHP y MySQL. El sistema está instalado en los servidores de la Alcaldía, cumpliendo con los parámetros establecidos por la institución en cuanto a la infraestructura física, ancho de banda y sistema de almacenamiento y protección de la información.

La plataforma EMPeCé permite organizar a los usuarios en Grupos de Gestión, cada uno de los cuales se corresponde con un programa específico, entre los cuales se pueden citar: Hecho en Pereira, Banca para Todos, CEDES, C+T+i, entre otros. Cada programa posee un líder que orienta la creación de usuarios y supervisa las actividades globales de los usuarios ligados al programa. El líder dispone de herramientas para validar las tareas realizadas y generar indicadores y reportes de las actividades desarrolladas. Esta es la gran oferta que la Secretaría pone a disposición de los usuarios, y que aquí se hace visible para los estudiantes que conforman programas académicos de base tecnológica.

Los emprendedores, por el sólo hecho de pertenecer a la plataforma EMPeCé, disponen de un dashboard en el cual pueden registrar sus proyectos y productos, facilitando la gestión integral de los mismos. Cada usuario emprendedor dispone de una clave de acceso única, y la traza de sus actividades queda registrada en la plataforma.

La plataforma EMPeCé, llena un vacío existente en los sistemas de información en la región, al permitir el registro digital de todos y cada uno de los emprendedores, de sus proyectos y de los productos que ofertan. Esta información, vital para la gestión de los emprendedores y para la generación de políticas de la Secretaría, se complementa con la disponibilidad de herramientas y módulos que facilitan el registro y seguimiento de cada producto y proyecto, facilitándole a los emprendedores optimizar sus recursos y promover el producido de sus actividades, generando oportunidades de comercialización y venta. Adicionalmente, le facilita a los emprendedores el cumplimiento de los requerimientos formales establecidos por el estado, y a la Secretaría el disponer de un mapa global de las actividades de emprendimientos, organizados por programa y meta datos ligados al proceso. Por último, resaltar que esta es una manera de promover el crecimiento regional, gracias al apoyo que desde la Alcaldía se brinda a esta iniciativa.

## II. CONTENIDO

### 1) Áreas de conocimiento

Técnicas de computación blanda:

- Redes Neuronales
- Lógica Difusa
- Sistemas Expertos
- Algoritmos Genéticos
- Agentes Inteligentes

### 2) Redes Neuronales

#### Descripción

Las redes neuronales son un campo de la inteligencia artificial el cual está inspirado en el funcionamiento del sistema nervioso, este modelo trata de emular tres factores, los cuales son: procesamiento paralelo, memoria distribuida y adaptabilidad al entorno.

La unidad principal de la red neuronal es la neurona, donde muchas de ellas se pueden organizar en capas.

Existe una gran variedad de redes neuronales, las cuales tienen diferentes propósitos, los más conocidos son el análisis de regresión, el cual se trata de predicción de series y funciones; y la clasificación, que engloba el reconocimiento de patrones.

#### Arquitectura

Una red neuronal está compuesta de varias neuronas, que al organizarlas se convierten en capas; al interconectar varias capas con otras se convierte en una red y esto es lo que se conoce como sistema neuronal.

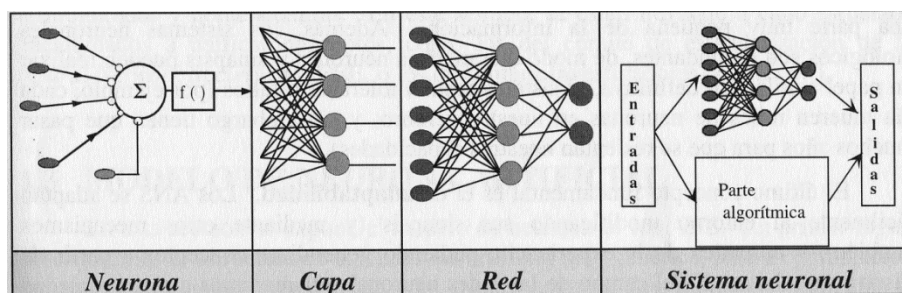


Imagen 1. Estructura de una Red Neuronal

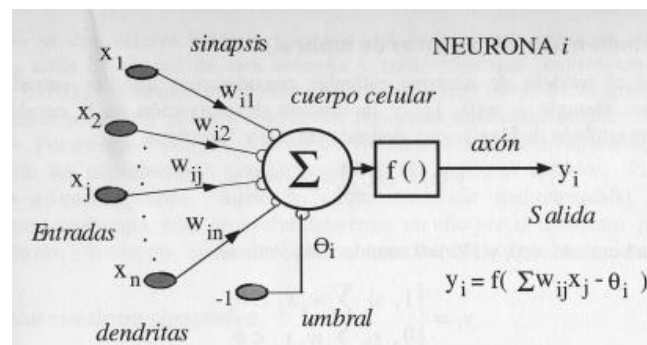


Imagen 2. Estructura de una Neurona

La forma en que las redes neuronales ajustan sus parámetros para dar una salida deseada es usando el algoritmo de Backpropagation, el cual, a partir de una función de error (normalmente el error de mínimos cuadrados), lleva hacia atrás el error para así usar el método del gradiente descendiente. Las salidas de error se propagan desde la última capa a la capa hacia capas anteriores, este proceso se realiza hasta que todas las neuronas que contribuyeron en la salida.

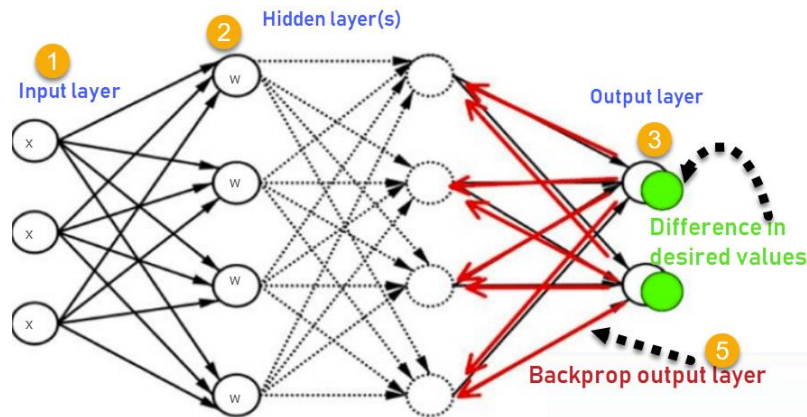


Imagen 3. Funcionamiento de Backpropagation

### Ejemplos de Uso

1. Reconocimiento de objetos en imágenes
2. Reconocimiento de calles con Google Street View
3. Mejorar el reconocimiento de voz actual
4. Aproximación de cualquier función continua
5. Detección de cambios en la superficie terrestre

### Herramientas disponibles

- Tensor Flow - Python
- Caffe / Caffe 2
- ONNX (Open Neural Network eXchanged) - Python
- Pytorch - Python
- Keras - Python
- Sklearn - Python
- Blocks - Python
- DeepLearnToolbox - MatLab

## Enlaces de referencia

<http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/t8neuronales.pdf>  
<http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/DM/tema3dm.pdf>  
[https://www.ibm.com/support/knowledgecenter/es/SS3RA7\\_sub/modeler\\_mainhelp\\_client\\_ddita/components/neuralnet/neuralnet\\_model.html](https://www.ibm.com/support/knowledgecenter/es/SS3RA7_sub/modeler_mainhelp_client_ddita/components/neuralnet/neuralnet_model.html)  
[https://www.ibm.com/support/knowledgecenter/es/SS3RA7\\_sub/modeler\\_mainhelp\\_client\\_ddita/components/neuralnet/neuralnet\\_model.html](https://www.ibm.com/support/knowledgecenter/es/SS3RA7_sub/modeler_mainhelp_client_ddita/components/neuralnet/neuralnet_model.html)  
<https://www.guru99.com/backpropagation-neural-network.html>  
<http://cipaponalredesneuronalesut.blogspot.com/2017/05/redes-de-propagacion-hacia-atras.html>  
<https://blog.es.logicalis.com/analytics/redes-neuronales-artificiales-funcionamiento-y-aplicacion-al-negocio>  
[http://atc.ugr.es/pages/personal/propia/alberto\\_prieto/conferencias\\_pdfs/redes-neuronales-modelos-y-aplicaciones%20%21](http://atc.ugr.es/pages/personal/propia/alberto_prieto/conferencias_pdfs/redes-neuronales-modelos-y-aplicaciones%20%21)

## 3) Lógica Difusa

La lógica difusa es un método de razonamiento aproximado no probabilista, esta permite representar de forma matemática la incertidumbre. Un elemento importante en esta técnica es el grafo de pertenencia difuso, el cual muestra la similitud de un evento con relación a otro.

La lógica difusa es usada cuando no existe un modelo matemático preciso o algunas partes del sistema son desconocidas.

Las características más importantes de la lógica difusa son:

1. Representación de la información imprecisa a través de la teoría de conjuntos imprecisos o difusos. En los conjuntos difusos la pertenencia a un conjunto puede tomar valores entre 0 y 1, a su vez, el cambio es gradual, a diferencia de los conjuntos clásicos que es un cambio inmediato

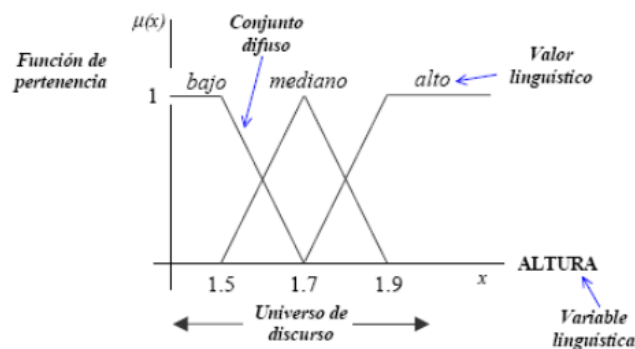


Imagen 4. Ejemplo de conjuntos difusos

2. El conocimiento se plantea como un conjunto de restricciones difusas sobre un conjunto de variables

## Arquitectura

La lógica difusa se basa en reglas de la forma “si antecedente entonces consecuente”, donde el antecedente y el consecuente son conjuntos difusos.

Los conceptos se relacionan con conjuntos difusos en un método llamado *fuzzificación*, luego de tener estos valores, se trabajan con reglas lingüísticas para producir una salida difusa, la cual puede ser defuzzificada para tener un valor discreto.

Cada elemento tiene un grado de pertenencia a un conjunto difuso.

Otro elemento importante es el conjunto de variables lingüísticas, las cuales son palabras de un lenguaje normal, su función es representar un elemento que no pueda describirse en términos numéricos. Cada variable lingüística tiene el nombre, el conjunto de etiquetas lingüísticas, el universo del discurso, una gramática libre de contexto y una regla semántica que asocia el valor a su significado.

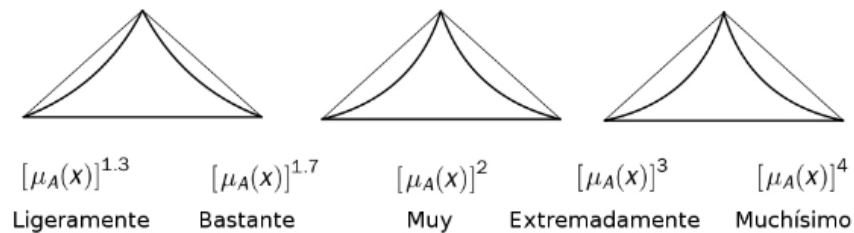


Imagen 5. Ejemplo de variables lingüísticas

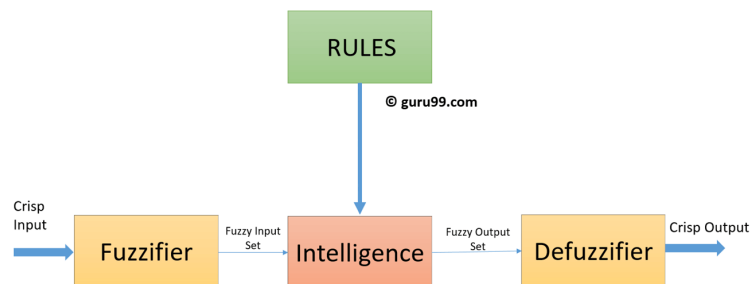


Imagen 6. Arquitectura de la lógica difusa

### Ejemplos de Uso

1. Sistemas de control de aire acondicionado
2. Mejorar la eficiencia de consumo de combustible en motores
3. Control crucero adaptativo en automóviles
4. Reconocer caracteres kanjis manuscritos
5. Selección automática de velocidades

### Herramientas Disponibles

- Fuzzy Logic Tools – C++
- Scikit-fuzzy – Python
- FuzzyWuzzy – Python
- PyFuzzy – Python

### Enlaces de referencia

[https://www.palermo.edu/ingenieria/pdf2014/13/CyT\\_13\\_24.pdf](https://www.palermo.edu/ingenieria/pdf2014/13/CyT_13_24.pdf)  
[https://www.esi.uclm.es/www/cglez/downloads/docencia/2011\\_Softcomputing/LogicaDifusa.pdf](https://www.esi.uclm.es/www/cglez/downloads/docencia/2011_Softcomputing/LogicaDifusa.pdf)  
[https://www.iit.comillas.edu/aramos/simio/transpa/t\\_se\\_as.pdf](https://www.iit.comillas.edu/aramos/simio/transpa/t_se_as.pdf)  
[https://www.palermo.edu/ingenieria/pdf2014/13/CyT\\_13\\_24.pdf](https://www.palermo.edu/ingenieria/pdf2014/13/CyT_13_24.pdf)  
<https://www.guru99.com/what-is-fuzzy-logic.html>  
[https://es.wikipedia.org/wiki/L%C3%B3gica\\_difusa#Aplicaciones](https://es.wikipedia.org/wiki/L%C3%B3gica_difusa#Aplicaciones)

#### 4) Sistemas Expertos

##### Descripción

Los sistemas expertos son un subconjunto de la inteligencia artificial los cuales usan conocimiento humano para resolver problemas, estos imitan el proceso de pensamiento de las personas expertas en un área de conocimiento.

Existen varios tipos de sistemas expertos, entre los cuales se encuentran los sistemas basados en reglas, basados en casos y basados en redes bayesianas

##### Arquitectura

Los sistemas expertos están compuestos por dos tipos de ambientes, de desarrollo y de consulta.

El sistema de desarrollo es donde se le entrega el conocimiento y el ambiente de consulta es donde otras personas pueden obtener consejos de este sistema.

En el ambiente de consulta se encuentran varios elementos muy importantes para el funcionamiento de un sistema experto, estos son: la base de conocimiento, el cual tiene toda la información pertinente para resolver el problema, la base de hechos, el cual tiene los datos que se desean analizar; y el motor de inferencia, el cual interpreta las reglas y controla los pasos para la solución del problema.

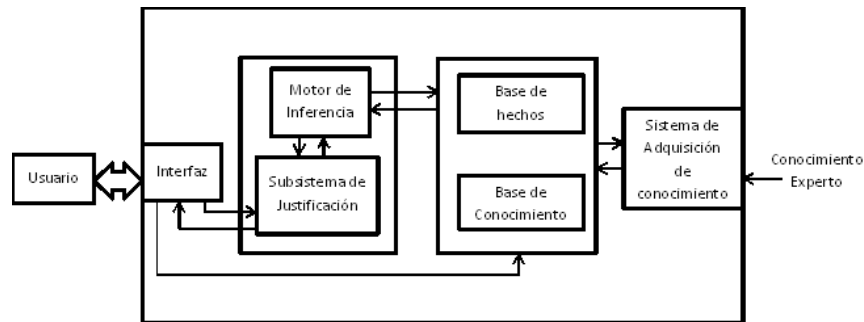


Imagen 7. Arquitectura de un Sistema experto

##### Ejemplos de uso:

1. Diagnóstico médico MYCIN
2. Guiado de vehículos y proyectiles de forma semiautomática
3. Diseño de circuitos de alto grado de integración
4. Diagnóstico de control de calidad
5. Simulación de Vuelos

##### Herramientas disponibles

- Amzi Prolog
- Lisp
- Jess – Java
- Pyke - Python

##### Enlaces de Referencia

[https://es.wikipedia.org/wiki/Sistema\\_experto#Ejemplos\\_importantes\\_de\\_los\\_sistemas\\_expertos](https://es.wikipedia.org/wiki/Sistema_experto#Ejemplos_importantes_de_los_sistemas_expertos)  
[https://www.academia.edu/10375975/Sistemas\\_expertos\\_y\\_sus\\_aplicaciones](https://www.academia.edu/10375975/Sistemas_expertos_y_sus_aplicaciones)  
[https://www.palermo.edu/ingenieria/pdf2014/13/CyT\\_13\\_24.pdf](https://www.palermo.edu/ingenieria/pdf2014/13/CyT_13_24.pdf)  
[https://www.ecured.cu/Sistemas\\_expertos](https://www.ecured.cu/Sistemas_expertos)  
<https://sites.google.com/site/proyecto-inteligencia-artificial/indice/los-sistemas-expertos>

## 5) Algoritmos Genéticos

### Descripción

Los algoritmos genéticos son técnicas adaptativas enfocadas en solucionar problemas de búsqueda y optimización, se basan en el proceso genético de los organismos vivos.

Es una variante de la búsqueda de haz estocástica donde los estados siguientes son el resultado de combinar dos estados padres, así, se tiene información de dos estados, a diferencia de muchas otras técnicas (ascenso a la colina, temple simulado, etc.) las cuales solo tienen un estado padre

La principal ventaja de los algoritmos genéticos es la capacidad del cruce de estados para intercambiar información entre aquellos que han evolucionado de forma independiente.

### Arquitectura

Los algoritmos genéticos tienen varios elementos importantes para su funcionamiento, el primero es una población inicial que, en muchos problemas, puede ser una cadena inicializada con valores aleatorios. Luego, a partir de una función de idoneidad, que determina que tan bueno es el estado, se realiza un proceso de selección de los mejores estados, a continuación, se realiza un proceso de cruce entre los estados seleccionados para así crear nuevos estados, y, por último, para agregar un factor que lleve a explorar más el espacio de estados se realiza un proceso de mutación.

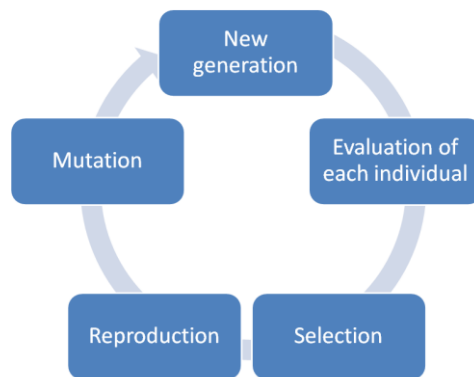


Imagen 8. Arquitectura del algoritmo genético

### Ejemplos de uso

1. Diseñar materiales y formas aerodinámicas para autos de carrera
2. Enrutamiento dinámico y anticipado de circuitos para telecomunicaciones
3. Cifrado de datos
4. Optimización de rutas vehiculares
5. Mejorar las mezclas de combustible para los motores

### Herramientas Disponibles

- PyEasyGA – Python
- DEAP – Python
- pySTEP – Python
- PyRobot – Python
- PyVolution

### Enlaces de referencia

---

<https://www.brainz.org/15-real-world-applications-genetic-algorithms/>  
<http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/temageneticos.pdf>  
<https://dzone.com/articles/11-deep-learning-with-python-libraries-and-frameworks>  
<https://la.mathworks.com/help/gads/what-is-the-genetic-algorithm.html>  
<https://www.geeksforgeeks.org/genetic-algorithms/>

## 6) Agentes Inteligentes

### Descripción

Para empezar a hablar de agentes inteligentes, es necesario enunciar la definición de agente, un agente es algo que razona, que puede percibir su entorno, que persiste durante un periodo de tiempo, que se adapte a los cambios y que sea capaz de alcanzar diferentes objetivos.

Un agente racional es un agente el cual busca obtener el mejor resultado esperado.

Existen varios tipos de agentes inteligentes: agentes reactivos simples, los cuales solo se guían por percepciones actuales y no por percepciones pasadas; agentes reactivos basados en modelos, los cuales tienen un estado interno que les permite guardar percepciones pasadas; agentes basados en objetivos, los cuales tienen información sobre la meta que deben alcanzar; agentes basados en utilidad, los cuales, además de las metas, tienen una noción de eficiencia; y por último, agentes que aprenden, los cuales tienen una retroalimentación de las críticas sobre las actuaciones pasadas.

### Arquitectura

Todos los agentes tienen elementos que son necesarios para su funcionamiento, como son los sensores y los actuadores, los agentes inteligentes tienen más elementos, como un elemento de aprendizaje, la crítica que servirá como retroalimentación, y un generador de problemas, que servirá para explorar más su entorno.

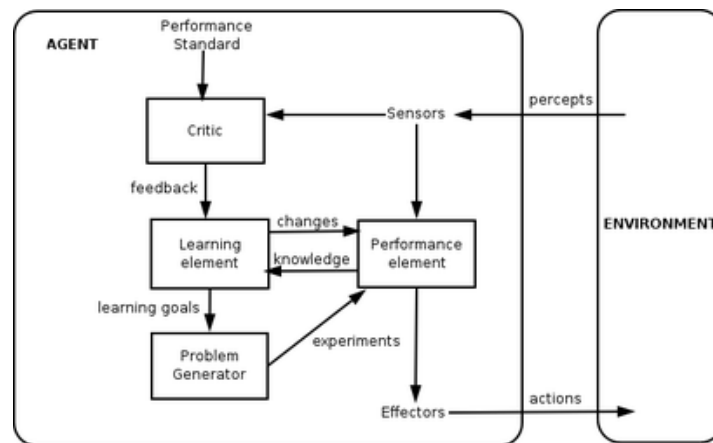


Imagen 9. Arquitectura de los agentes inteligentes

### Ejemplos de uso

1. Control de producción hortícola
2. Asistentes Virtuales
3. Manejo autónomo
4. Automatización de procesos industriales con robótica
5. Mercadotecnia



**Herramientas disponibles**

- AIMA – Python
- SimpleAI – Python
- Mxnet – Python
- Theano – Python
- Lasagne – Python

**Enlaces de referencia**

[https://en.wikipedia.org/wiki/Intelligent\\_agent](https://en.wikipedia.org/wiki/Intelligent_agent)

<https://www.geeksforgeeks.org/agents-artificial-intelligence/>

[http://www.mind.ilstu.edu/curriculum/ants\\_nasa/intelligent\\_agents.php](http://www.mind.ilstu.edu/curriculum/ants_nasa/intelligent_agents.php)

<https://www.techopedia.com/definition/28055/intelligent-agent>

**7) Listado de problemas potenciales tomados de todas las tecnologías enunciadas**

Nro.	Descripción del problema	Tecnologías a Utilizar
1		
2		
3		
4		
5		
6		
7		
8		

**8) Datos del emprendedor**

Nro.	Ítem	Dato
1	Nombres	David Esteban
2	Apellidos	Cediel Gómez
3	Correo	<a href="mailto:David.cediel@utp.edu.co">David.cediel@utp.edu.co</a>
4	Nombre de Usuario	Davidcediel12
4	Programa Académico	Ingeniería de Sistemas y Computación
5	Semestre actual	Octavo

**9) Caracterización del emprendedor**

Nro.	Característica	Descripción
1	Lenguajes de programación	Python, Java
2	Áreas de interés	Computación Blanda
3	Experiencia en Investigación	Ninguna
4	¿Emprendimientos previos?	Ninguno
5	Interés por el emprendimiento	Poco
6	Interés por la investigación	Elevado
7	Interés por la aplicación del conocimiento	Poco
8	Proyecto seleccionado	
9	Producto a desarrollar	
10	Estado actual del producto	

**10) Problema Seleccionado**

<Solución propuesta>

**11) Solución Propuesta**

<Teoría general del problema>

**12) Análisis de la solución**

<Mapa de requerimientos>

**13) Diseño de la solución**

<Arquitectura y solución propuesta>

**14) Implantación**

<Implantación: algoritmo y código>

**15) Modelo de pruebas**

<Pruebas del producto desarrollado>

**16) Modelo de comercialización**

<Comercialización del producto. E-commerce>

### III. CONCLUSIONES

<Conclusiones del proceso desarrollado e impactos obtenidos>.

### REFERENCIAS

Inteligencia Artificial 101. Disponible: <https://inteligenciaartificial101.wordpress.com/tag/perceptron/>

El perceptrón. Disponible: <file:///C:/Users/utp/Desktop/Redes%20de%20una%20capa.pdf>

Koldo Pina, (2018, marzo). Como entrenar a tú perceptrón. Disponible: <https://koldopina.com/como-entrenar-a-tu-perceptron/>