Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 82-96
http://publicaciones.uci.cu

Tipo de artículo: Artículo original
Temática: Soluciones Informáticas
Recibido: 10/04/2020 | Aceptado: 06/06/2020 | Publicado: 01/09/2020

Algoritmo de búsqueda fonética para el Sistema de Gestión Integral de Aduanas

Phonetic Searching Algorithms for the Integral Customs Management System.

Selianne Labañino Urbina[1*], Orlando Grabiel Toledano López[2], Camileidis Labañino Gainza[1]

[1] Departamento de Informática, Facultad 3, Universidad de las Ciencias Informáticas. Carretera de San Antonio de los Baños km 2 ½, Torrens, Boyeros, La Habana, Cuba. {slabanino, camil}@estudiantes.uci.cu

[2] Departamento de Informática, Facultad 4, Universidad de las Ciencias Informáticas. Carretera de San Antonio de los Baños km 2 ½, Torrens, Boyeros, La Habana, Cuba. ogtoledano@uci.cu

[*] Autor para correspondencia: ogtoledano@uci.cu

Resumen

La correspondencia automática de palabras con pronunciaciones similares es uno de los mecanismos cruciales para el mundo de la informática. En esta área existen diversos algoritmos, que se desarrollan para resolver el problema de la pronunciación. Pero su estudio permitió arribar a la conclusión de que estos no consideran las particularidades del idioma español, lo que los hace inapropiados en tareas de detección de duplicados originadas por errores ortográficos en este idioma. En el Sistema de Gestión Integral de Aduanas se está utilizando un algoritmo fonético que no brinda resultados correctos en correspondencia con el idioma español, lo que trae consigo errores de naturaleza ortográfica y tipográfica, en los datos procesados de las personas que interactúan con la aduana. Se realizó una revisión de la literatura científica sobre el uso de este tipo de algoritmo, identificando cuáles son las características más relevantes de los mismos y ventajas de su utilización. En la presente investigación se describe el desarrollo de un algoritmo fonético, que fue desarrollado mediante la metodología AUP_UCI, basada en el framework Symfony y el lenguaje de programación PHP en su versión 5.6. Además, de que se realizaron los flujos definidos por la metodología empleada: Modelo, Implementación, y Prueba. El algoritmo propuesto, contribuirá a una mejor toma de decisiones, ayudando así a eliminar los errores que se pudieran cometer en el proceso de gestión de información en esta institución.

Palabras clave: *algoritmos fonéticos, Spanish_Soundex, Spanish_Metaphone.*

Abstract

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 82-96
http://publicaciones.uci.cu

Automatic matching of words with similar pronunciations is one of the crucial mechanisms for the world of computing. In this area there are several algorithms, which are developed to solve the problem of pronunciation. But their study allowed arriving at the conclusion that these do not consider the particularities of the Spanish language, which makes them inappropriate in tasks of detection of duplicates originated by spelling errors in this language. In the Integral Customs Management System, a phonetic algorithm is being used that does not provide correct results in correspondence with the Spanish language, which brings about errors of an orthographic and typographical nature, in the processed data of the persons who interact with the customs. A review of the scientific literature on the use of this type of algorithm was carried out, identifying the most relevant characteristics of the algorithm and the advantages of its use. This research describes the development of a phonetic algorithm, which was developed using the AUP_UCI methodology, based on the Symfony framework and the PHP programming language in its version 5.6. In addition, that the flows defined by the methodology used were carried out: Model, Implementation, and Testing. The proposed algorithm will contribute to a better decision making, helping to eliminate the errors that could be made in the process of information management in this institution.

Keywords: *Spanish_Soundex, Spanish_Metaphone, phonetic algorithms.*

Introducción

Actualmente las organizaciones acumulan grandes volúmenes de datos, que sirven de base para generar información y conocimiento. Los datos incorrectos pueden llevar a tomar decisiones equivocadas con la consiguiente pérdida de tiempo, dinero y credibilidad. Uno de los inconvenientes que se pueden presentar en los datos almacenados se da cuando en una misma entidad del mundo real se almacena más de una vez de diferentes formas. El proceso para detectar este tipo de problemas se conoce como *record linkage* en el área de la estadística; *database hardening* en el área de la inteligencia artificial; *merge - purge*, o algoritmos fonéticos también se usan con frecuencia (Amón et al., 2012). Se han desarrollado funciones de similitud para la detección de duplicados; algunas de ellas son de tipo fonético como *Soundex* (Bhatti et al., 2014), *Metaphone* (P.Parmar & K Kumbharana, 2014), *Double Metaphone* (Koneru et al., 2016), Onca, Nysiis (Gálvez, 2007). Estas técnicas, se basan en la forma cómo se pronuncian las palabras en un idioma en particular y no están orientadas al idioma español.

Los algoritmos fonéticos han sido aplicados en sistemas que requieren mecanismos de recuperación de información en diferentes contextos o dominios de problemas (Benitez, 2017). Estos constituyen una alternativa para la búsqueda de términos donde las coincidencias no deben ser exactas pero si aproximadas, las cuales permiten detectar semejanzas entre pares de palabras en menor tiempo con respecto a otras técnicas de búsqueda aproximadas, tales como: difusa o parcial (Cámara, 2016).

En el Sistema de Gestión Integral de Adunas (GINA), se está utilizando un algoritmo fonético que no está diseñado para el idioma inglés con el objetivo de lograr el reconocimiento fonético, el cual, en muchas ocasiones, no

corresponde a la perfección con el uso de caracteres especiales y pronunciaciones específicas del idioma español. Esto trae consigo que se detecten falsos positivos, representando un fiasco en la gestión de la información y costos adicionales para las personas que han sido detectadas falsamente, provocando una mayor carga de trabajo para la institución.

A partir de lo anterior en esta investigación se plantea un nuevo algoritmo con principios fonéticos, considerando las características propias de la lengua española. El mismo basa su funcionamiento en dos algoritmos predecesores a este el algoritmo *Soundex* y el *Metaphone*, encaminados a cubrir las reglas de pronunciación del idioma español.

Materiales y métodos o Metodología computacional

Para el desarrollo de la propuesta se realizó un estudio de los principales algoritmos fonéticos utilizados en sistemas similares y algunos adaptados a la etimología del idioma español, donde se modifican algunas de las reglas de transformación para grupos fonéticos específicos distinguibles y característicos del idioma. El estudio y análisis del funcionamiento de cada uno, permitió identificar elementos importantes para el desarrollo de un algoritmo híbrido que combina varias características de los analizados y mejora el desempeño con respecto al que utilizaba el sistema GINA.

Algoritmo fonético *Spanish Soundex*

Las limitaciones del algoritmo Soundex han sido documentadas extensamente y han resultado en varias mejoras, pero ninguna orientada al idioma español. El estudio de las bibliografías sobre este tema, permitió determinar que el *Soundex* no está orientado al idioma español, incluso contemplando los caracteres españoles. Además, la dependencia de la letra inicial, el punto de la articulación del agrupamiento de la lengua inglesa, y el límite de codificación de cuatro caracteres no son eficientes para detectar errores ortográficos comunes en la lengua española (Koneru, 2016).

Una variante de esta codificación fonética orientada al idioma español, fue propuesta en (Amón et al., 2012), donde se realiza una codificación extendida del algoritmo *Soundex*. En esta modificación el autor agregó los caracteres españoles, y eliminó la dependencia a la primera letra, logrando así que la codificación final sea de la palabra completa en dígitos. Como resultado, el *Spanish Soundex* es más exacto que el algoritmo original en la búsqueda de coincidencias cercanas para las palabras españolas (Koneru et al., 2016).

Años más tarde en (Del Pilar Angeles et al., 2015) se realiza otra modificación al algoritmo fonético español para que el código de codificación sea redimensionable, logrando que todos los espacios en blanco se eliminen durante la codificación.

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 82-96
http://publicaciones.uci.cu

A continuación, se muestran los pasos del algoritmo (Koneru, 2016).

1. Los caracteres son convertidos en mayúscula, se deben ignorar todos los signos de puntuación.
2. Se eliminan las siguientes letras: 'A','E','I', 'O','U','H','W'.
3. Se cambian las letras en dependencia del grupo al que corresponda, según la siguiente tabla:

Tabla 1: Transformaciones del algoritmo Spanish Soundex (Koneru, 2016)

Letra a cifrar	**Dígito codificado**	**Letra a cifrar**	**Dígito codificado**
P	0	L,Y,LL	5
B,V	1	M,N,Ñ	6
F,H	2	Q,K	7
T,D	3	G,J	8
S,C,Z,X	4	R,RR	9

Algoritmo fonético *Spanish Metaphone*

Este algoritmo fue desarrollado por Alejandro Mosquera en el 2012 para el idioma español. Se basa en una adaptación de las reglas usadas en el *Metaphone* para el idioma inglés. A diferencia del *Spanish Soundex*, el algoritmo desarrollado retiene la información proveniente de las vocales, resultando el código final, un conjunto de caracteres (Koneru, 2016). A continuación, se muestran algunas de las transformaciones realizadas al algoritmo:

Tabla 2: Transformaciones del algoritmo Spanish Metaphone (Koneru, 2016)

á	ch	Ç	é	í	ó	ú	ñ	gü	ü	b	ll
A	X	S	E	I	O	U	NY	W	U	V	Y

Tabla 3: Secuencia de pasos del algoritmo Spanish Metaphone (Koneru, 2016)

1. Convertir todas las letras en mayúscula.
2. Si la primera letra es una vocal, se retiene la primera letra.
3. Se deben eliminar todas las letras repetidas, excepto la letra C.
4. Transformar las letras según estas reglas:

• CC→ X, • CE, CI → Z, • C → K.
5. Transformar GE, GI → J, sino la G es retenida.
6. Hv' es transformada a 'v' donde v es una vocal. En otro caso la 'H' es preservada.
7. Q→K, sino es seguida por una U. Entonces 'QU' es eliminada.
8. Se debe cambiar la W→U.
9. S → ES, si está al inicio de una palabra y seguida por una vocal, en otro caso la S es retenida.
10. X →EX, si está al inicio de la cadena y seguida por una vocal. En otro caso la X es retenida.

Algoritmo fonético *Meta_Soundex* como solución propuesta

Teniendo en cuenta las desventajas de los algoritmos tratados anteriormente, se desarrolló un algoritmo llamado *Meta_Soundex*. El algoritmo propuesto mejora su precisión sobre *Soundex*, ya que incluye la codificación de los sonidos vocálicos y los sonidos fonéticos combinados antes de la agrupación de letras individuales. La precisión del *Meta_Soundex* es mayor que la del *Metaphone* transformándolo de manera más eficiente que otros algoritmos.

Pasos para la implementación del algoritmo (Koneru, 2016):

1. Convertir todas las letras en mayúscula.
2. Codificar usando *Spanish Metaphone* para retener los sonidos vocálicos y las combinaciones de diptongos.
3. Codificar lo obtenido usando el *Spanish Soundex*.

El algoritmo anterior genera un código *Meta_Soundex* de longitud variable para el idioma español.

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 82-96
http://publicaciones.uci.cu

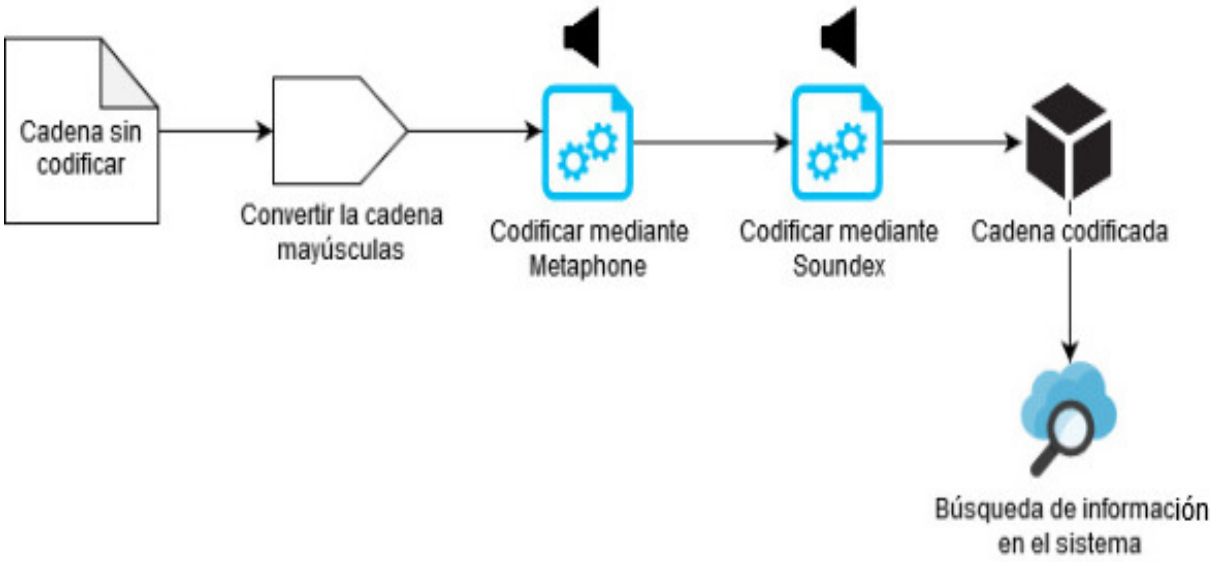


Figura 1: Secuencia de pasos del algoritmo propuesto.

Resultados y discusión

Las pruebas se aplican en diferentes niveles teniendo en cuenta una serie de objetivos en diversos escenarios y niveles de trabajo, debido a que las pruebas son agrupadas por niveles que se encuentran en distintas etapas del proceso de desarrollo. Se muestran a continuación algunos niveles de pruebas para valorar cuales serán utilizados:

- **Pruebas unitarias:** se ponen a prueba unidades de programa o clases de objetos individuales. Las pruebas de unidad deben enfocarse en comprobar la funcionalidad de objetos o métodos (Sommerville, 2011).
- **Pruebas del componente:** donde muchas unidades individuales se integran para crear componentes compuestos. La prueba de componentes debe enfocarse en probar interfaces del componente (Pressman, 2010).
- **Prueba de integración:** técnica sistemática para construir la arquitectura del software, mientras que al mismo tiempo, se van ejecutando pruebas para encontrar errores asociados con la interfaz (Sommerville, 2011). Es realizada también por el desarrollador de software con el propósito de detectar errores de interfaces y

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 82-96
http://publicaciones.uci.cu

relaciones entre componentes. Consiste fundamentalmente en validar las conexiones e integración entre dos o más componentes de software haciendo uso de la técnica de caja blanca.

Con la ejecución de las actividades de prueba, se aplicaron pruebas unitarias para poder validar las funcionalidades con las que cuenta el algoritmo propuesto mediante el uso de *PHPUnit*, que es una biblioteca para realizar este tipo de test en el lenguaje de programación PHP. El resto de los niveles de pruebas aplicados, permitieron evaluar la calidad de la propuesta desarrollada y su integración con el resto de los componentes de GINA donde se invoca la búsqueda fonética de información y recuperación de personas.

Se aplicaron además como otro elemento de validación de la solución propuesta el método de pruebas de caja blanca, específicamente la técnica de camino básico, con el objetivo de probar procedimientos del algoritmo, derivando casos de pruebas, para asegurar que se ejecuten las instrucciones del mismo por lo menos una vez en la prueba, y que todas las condiciones lógicas se ejerciten.

Resultados obtenidos del método de caja blanca

Para la realización de las pruebas se aplicó el método de caja blanca sobre las clases principales del algoritmo propuesto, con la ejecución de las mismas, se pudo derivar un conjunto de pruebas que tienen la mayor probabilidad de descubrir errores en el software.

A continuación, se presenta el resultado de la prueba realizada a uno de los métodos principales del algoritmo diseñado:

Tabla 4: Resultado de la prueba de ruta básica en el método *Metaphone*.

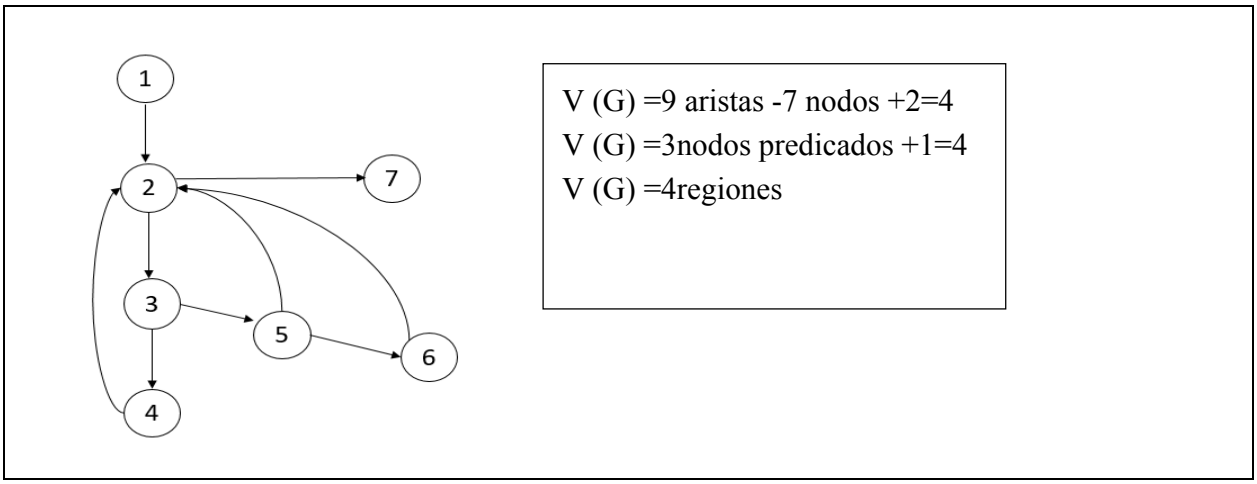
```
<?php
/**
 * Created by PhpStorm.
 * User: Selianne
 * Date: 10-feb-20
 * Time: 3:39 p.m.
 */class PhoneticAlgorithmsES
{
   public function Metaphone($string, $phonemes = 0)

   {
```

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 82-96
http://publicaciones.uci.cu

```
$meta_key = '';
$current_pos = 0;
$string_length = strlen($string);
$end_of_string_pos = $string_length - 1;
$original_string = $string. ' ';
$original_string = strtoupper($original_string);

while (0 === $phonemes || strlen($meta_key) < $phonemes){

  if (self::ContieneVocal($original_string, $current_pos)
      && 0 === $current_pos) {

    $meta_key .= $current_char;
    $current_pos += 1;
    } else
      {
       if (self::CadenaOriginal($original_string, $current_pos,
          2, ['CE','CI'])) {

          $meta_key .= 'Z';

          $current_pos += 2;
        }
          }
  $meta_key = trim($meta_key);

   return $meta_key;
  }
}
```

1 2 3 4 5 6 7

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 82-96
http://publicaciones.uci.cu



Resultados obtenidos con la realización de pruebas estadísticas. Medidas de calidad en los datos.

En este apartado se hará alusión a diferentes medidas para poder analizar toda la información que resulta de comparar los datos del sistema, con los generados por el algoritmo que se propone.

Estos datos se clasificarán en varias categorías (María del Pilar Ángeles, 2016):

- TP: Conocidos como positivos verdaderos, que son el registro de que los datos clasificados como un par, son los datos verdaderos.
- FP: Los casos negativos erróneamente clasificados como positivos por el algoritmo.
- TN: (*true negative*) Se refieren a los casos negativos que han sido clasificados, correctamente por el algoritmo.
- FN: (*false negative*) Los pares que han sido clasificados como falsos, pero son verdaderos, son considerados con igual valor que los positivos verdaderos (Fuentes et al., 2017).

Estos elementos generan una matriz de confusión como se muestran a continuación:

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 82-96
http://publicaciones.uci.cu

Figura 3: Matriz de confusión de registros de clasificación de datos.

Las medidas de calidad de datos son tomadas de (Christen & Goiser, 2007) por su estudio en la comparación de datos entre distintos conjuntos de información:

- Exactitud (*acc*): Indica la capacidad que tiene el algoritmo de detectar coincidencias entre los dos conjuntos de datos a evaluar, a menudo se utiliza de la misma forma que la precisión, la diferencia es que la exactitud toma en cuenta todos los valores de la matriz de confusión (Figura7), y está determinada por la siguiente fórmula:

$$acc = \frac{|TP| + |TN|}{|TP| + |FP| + |TN| + |FN|}$$

- Precisión (*prec*): Llamada también como valor positivo predictivo, es la proporción de datos clasificados como verdaderos. Se representa por la siguiente ecuación:

$$\text{prec} = \frac{|\text{TP}|}{|TP| + |FP|}$$

- Sensibilidad (*rec*): Conocida también como *recall* o tasa de valores positivos. Esta es la proporción de coincidencias actuales que han sido clasificadas correctamente, y su fórmula para el cálculo es la siguiente:

$$\text{rec} = \frac{|\text{TP}|}{|TP| + |FN|}$$

- Especificidad (*spec*): Tasa de valores negativos, este valor podría ser alterado si se cuenta con una tasa muy alta de verdaderos negativos, normalmente originado porque el conjunto de datos ha sido tratado previamente para quitar incongruencias. Su fórmula es como se describe a continuación:

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 82-96
http://publicaciones.uci.cu

$$\text{spec} = \frac{|\text{TN}|}{|TN| + |FP|}$$

- Tasa de Falsos Positivos (*fpr*): Considerada como la inversa de la especificidad y ayuda a determinar la tasa actual de falsos positivos de una evaluación dada. Está determinada por la siguiente fórmula:

$$\text{fpr} = \frac{|\text{FP}|}{|TN| + |FP|}$$

De igual manera se puede calcular de la siguiente forma:

$$fpr = 1 - spec$$

- *F-measure*: Conocida como la media armónica entre la sensibilidad y la precisión, esta tendrá un valor elevado cuando tanto la sensibilidad como la precisión, tengan valores elevados, y se encuentra determinada por la fórmula siguiente:

$$\text{f} - measure = 2\frac{\text{prec} * \text{rec}}{prec + rec}$$

Pruebas estadísticas

Para realizar las pruebas estadísticas se generaron un conjunto de datos compuestos por registros con atributos como nombres y apellidos, de las personas que están almacenados en la base de datos de la aduana. Los conjuntos de datos generados fueron de 25, 50, 100 y 1000 nombres tomados aleatoriamente.

Escenario #1

Se realizó una búsqueda en la base de datos de la aduana de 25 nombres tomados aleatoriamente, y verificando que ninguno de los nombres se repitiera en la muestra. De estos se encontraron 1.325 personas y 43 FP por el algoritmo utilizado en el sistema GINA. Sin embargo, por el algoritmo *Meta_Soundex*, se encontraron 1.282 personas y 26 FP.

Tabla 5: Resultado de la prueba estadística para el dataset de 25 nombres.

Medida de calidad\Algoritmo	GINA	*Meta_Soundex*
Total de personas a clasificar	1.325	1.282
Precisión	0.96	**0.98**
F_Measure	0.975	**0.989**

Escenario #2

Se realizó una búsqueda en la base de datos de la aduana de 50 nombres tomados aleatoriamente, y verificando que ninguno de estos nombres se repitiera en la muestra. De estos se encontraron 1.765 verdaderos positivos según el

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 82-96
http://publicaciones.uci.cu

algoritmo utilizado en el sistema y 294 FP. Utilizando el algoritmo *Meta_Soundex_Algorithm*, se encontraron 1.471 personas y 32 clasificadas como FP.

Tabla 6: Resultado de la prueba estadística para el dataset de 50 nombres.

Medida de calidad\Algoritmo	GINA	*Meta_Soundex*
Total de personas a clasificar	1.767	1.471
Precisión	0.857	**0.978**
F_Measure	0.922	**0.988**

Escenario #3

Se realizó una búsqueda en la base de datos de la aduana de 100 nombres tomados aleatoriamente, y verificando que ninguno de estos nombres se repitiera en la muestra. De estos se encontraron 3.475 TP según el algoritmo utilizado en el sistema, y 373 personas fueron clasificadas como FP. Con el algoritmo *Meta_Soundex_Algorithm*, se clasificaron 3.100 y 71 como FP.

Tabla 7: Resultado de la prueba estadística para el dataset de 100 nombres.

Medida de calidad\Algoritmo	GINA	*Meta_Soundex*
Total de personas a clasificar	3.475	3.100
Precisión	0.903	**0.977**
F_Measure	0.949	**0.988**

Escenario#4

Se realizó una búsqueda en la base de datos de la aduana de 1000 nombres tomados aleatoriamente, y verificando que ninguno de estos nombres se repitiera en la muestra. De estos se encontraron 35.959 verdaderos positivos según el algoritmo utilizado en el sistema, y 2.188 como FP. Además se utilizó el algoritmo *Meta_Soundex_Algorithm*, donde se clasificaron 33.771 y 528 como FP.

Tabla 8: Resultado de la prueba estadística para el dataset de 1000 nombres.

Medida de calidad\Algoritmo	GINA	*Meta_Soundex*
Total de personas a clasificar	35.959	33.771
Precisión	0,942	**0.984**
F_Measure	0.9701	**0.9919**

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 82-96
http://publicaciones.uci.cu

Metodología para el análisis de las pruebas

- Determinación de las variables estadísticas:

  La variable dependiente está determinada por el número de aciertos o coincidencias fonéticas verdaderas.

  La variable independiente está determinada por el tipo de algoritmo utilizado, este afecta a la variable dependiente, conforme se utilice un algoritmo distinto de codificación fonética.

- Determinación del método estadístico:

  Para llevar a cabo las pruebas se utilizó el método estadístico ANOVA, para la comparación de las medias de las distintas variables a evaluar (precisión y *F_Measure*). Se crearon 2 grupos de datos con los resultados de las comparaciones entre los algoritmos.

La comparación posterior se realizó mediante el test de DUNCAN (Fallas, 2012).Esto con el fin de determinar los casos en los que hubo diferencia significativa en las medias, a esto se le conoce como estudio posterior. El experimento con la prueba ANOVA se describe a continuación:

Prueba de Análisis de medias (ANOVA)

El experimento tiene el siguiente modelo estadístico:

$$H_0 = \mu_1 = \mu_2 = \mu_k$$

$$H_a = por\ lo\ menos\ 1\ \mu\ diferente$$

Donde H_0 es la definición de la hipótesis nula y H_a es la hipótesis alternativa, y μ es el número de aciertos por grupos de algoritmos comparados, y k es el número de grupos evaluados, que en este caso son 2.

Si la significancia de F es menor que 0.05 se rechaza la hipótesis nula, y se dice que no hay suficiente evidencia para afirmar que existe diferencia significativa entre los dos algoritmos para la búsqueda fonética de nombres.

Resultados de las pruebas estadísticas

Si se considera la hipótesis planteada anteriormente, y evaluando cada valor de la media de los valores de la precisión y *F_Measure*, de los algoritmos GINA y *Meta_Soundex* se puede decir que:

- En la prueba ANOVA, se puede observar que los valores de las medias de cada valor calculado para cada algoritmo, era mayor que 0.05, por lo tanto no se rechaza H_0 y se puede afirmar que hay diferencia significativa entre los dos algoritmos analizados.
- Se concluye que el algoritmo *Meta_Soundex* tiene mayor precisión que el algoritmo que se está utilizando actualmente en el sistema, contribuyendo positivamente en la detección de coincidencias verdaderas, considerando casos verdaderos positivos y casos verdaderos negativos.
- Evaluando el balance armónico entre la precisión y la sensibilidad, el algoritmo *Meta_Soundex* es que tiene más alto valor. Esto indica que su factor de predicción de valores verdaderos positivos en comparación con la tasa de verdaderos positivos es alto, con respecto al algoritmo que se está utilizando.

Conclusiones

Al término de este trabajo se arriban a las siguientes conclusiones:

- Las pruebas unitarias realizadas, contribuyeron a fomentar el cambio y la refactorización para lograr que el algoritmo implementado fuese el esperado, brindando los resultados correctos.
- La aplicación de pruebas estadísticas mediante ANOVA permitieron demostrar que el algoritmo *Meta_Soundex* tiene mayor precisión que el algoritmo que se está utilizando actualmente en GINA, lo cual contribuye en la mejora del proceso de búsqueda de información en el sistema.
- Las pruebas de trayectoria básica efectuadas a la solución permitieron corroborar que cada secuencia del algoritmo se ejecuta al menos una vez. Además de ayudar a corregir errores para mejorar la calidad de la solución y garantizar la aceptación final por parte del cliente.
- Las pruebas realizadas al resultado final permitieron verificar la correcta implementación de cada clase del algoritmo, haciendo énfasis en la reducción de los errores internos. Al igual que ayudaron a comprobar la eficacia del producto en aras de mejorar la precisión del algoritmo en relación al usado actualmente en el sistema.
- Al realizar las pruebas estadísticas a la solución, se pudo corroborar la veracidad de las hipótesis planteadas que permitieron asegurar que el algoritmo diseñado brinda mejores resultados a la hora de realizar alguna búsqueda en el sistema.

Referencias

Amón, I., Moreno, F., & Echeverri, J. (2012). Algoritmo fonético para detección de cadenas de texto duplicadas en el idioma español. *Revista Ingenierías. Universidad de Medellín*, *11*(20), 127–138.

Benitez, A. R. T. (2017). *Motor de búsqueda por datos primarios utilizando algoritmos fonéticos*. Facultad de Ciencias Puras y Naturales.

Bhatti, Z., Waqas, A., Ismaili, I. A., Hakro, D. N., & Soomro, W. J. (2014). Phonetic based SoundEx&ShapeEx algorithm for Sindhi Spell Checker system. *Advances in Environmental Biology*, *8*(4), 1147–1155.

Cámara, J. de J. M. (2016). *Análisis comparativo de algoritmos de búsqueda de coincidencia de nombres por variación fonética contra la lista de OFAC para determinar su eficacia contra una lista de nombres en español*. Universidad Autónoma de Aguascaliente.

Christen, P., & Goiser, K. (2007). *Quality and Complexity Measures for Data Linkage and Deduplication*.

Del Pilar Angeles, M., Espino-Gamez, A., & Gil-Moncada, J. (2015). Comparison of a Modified Spanish Phonetic, Soundex, and Phonex coding functions during data matching process. *2015 4th International Conference on Informatics, Electronics and Vision, ICIEV 2015*, *December*. https://doi.org/10.1109/ICIEV.2015.7334028

Fallas, J. (2012). *Análisis de Varianza.Comparando tres o más medias*.

Gálvez, C. (2007). Identificación de nombres personales por medio de sistemas de codificación fonética. *Departamento de Biblioteconomía Y Documentación*, *11*(22), 105–116. https://doi.org/10.5007/1518-2924.2006v11n22p105

Koneru, K. (2016). *Phonetic Matching Toolkit with State of the art Meta-Soundex-Algorithm(English and Spanish)*. Sam Houston State University.

Koneru, K., Pulla, V. S. V., & Varol, C. (2016). Performance evaluation of phonetic matching algorithms on english words and street names comparison and correlation. *Proceedings of the 5th International Conference on Data Management Technologies and Applications*, *DATA*, 57–64. https://doi.org/10.5220/0005926300570064

P.Parmar, V., & K Kumbharana, C. (2014). Study Existing Various Phonetic Algorithms and Designing and Development of a working model for the New Developed Algorithm and Comparison by implementing it with Existing Algorithm(s). *International Journal of Computer Applications*, *98*(19), 45–49. https://doi.org/10.5120/17295-7795

Pressman, R. S. (2010). *Ingeniería de Software. Un enfoque práctico* (7th ed.).

Sommerville, I. (2011). *Ingeniería de Software* (9th ed.).