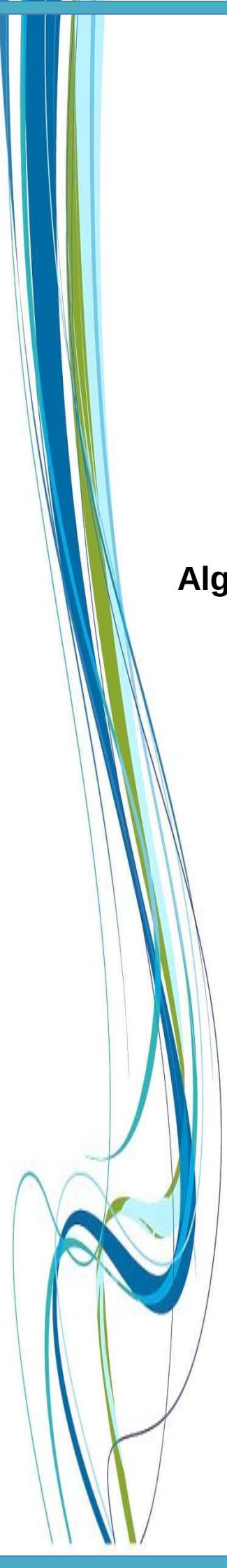

Universidad de las Ciencias Informáticas

Facultad 3

Algoritmo de búsqueda fonética parael Sistema de Gestión Integral de Aduanas

Trabajo final presentado en opción al título de
Ingeniero en Ciencias Informáticas

Autora

Selianne Labañino Urbina

Tutores

MSc. Orlando Grabiel Toledano López
Ing. Camileidis Labañino Gainza

La Habana, septiembre de 2020

Declaración jurada de autoría

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los _____días del mes de _____del año 2020.

Selianne Labañino Urbina

_____

Firma de la autora

MSc. Orlando Grabiel Toledano López

_____

Firma del tutor

Ing. Camileidis Labañino Gainza

_____

Firma del tutor

AGRADECIMIENTOS

Todos estos años de universidad han estado cargados de sacrificio y dedicación con el objetivo de cumplir un sueño: ser universitaria.

A mi madre que ha sido el motor impulsor de todas mis acciones, esa persona que además de ser madre es amiga, confidente, mi guía. Esa que se sacrifica y mueve el mundo para que sus hijos no se sientan solos o necesiten algo. Ha sido el mayor privilegio y orgullo el ser tu hija, eres la mejor madre que me podría haber tocado.

A mi papá por siempre estar presente en cada día de mi vida, a pesar de la distancia que nos separaba en ocasiones. Darme sus consejos, y siempre inculcarme ese ejemplo de perseverancia para lograr nuestros sueños.

A mi hermano, esa personita que ha estado conmigo siempre y me da fuerzas para que siga siendo su ejemplo a seguir como estudiante.

A mi novio, mi compañero de vida, mi cosa adorada, quien llegó en el momento más duro de mi vida y supo darle un giro inmenso con su amor, su compañía, su dedicación para ayudarme siempre en cada obstáculo que se me presenta. Gracias por ser ese brazo amigo, ese hombro donde lloré tantas veces, pero en el que apoyé y seguí adelante. Eres lo mejor que me ha pasado en la vida.

A mis abuelos Milagros y Diosmede, por apoyarme y darme su ayuda cuando lo necesité.

A mis amigas Danet, Patry, Guadalupe, Elianis, por estar conmigo a pesar de tantos años de amistad, aún cuando en ocasiones estábamos separadas.

A mis tutores por saber guiarme en todo este proceso tan difícil, y ayudarme a convertirme en una profesional capacitada e integral.

A mis amigos Eric y Samuelque participaron en mi formación y me brindaron su apoyo y amistad incondicional.

A mis primos Yoel y David por ser parte de mi vida, y esos hermanos de sangre con quienes comparto tantos momentos.

A mis suegros que me han sabido recibir en su familia como otro miembro más, y han estado al pendiente de mi formación profesional.

A mis compañeras de aula Rachel, Bety, Yaraisy, Daniela, Melissa, Arlet, y otros, por compartir juntos todos estos años de estudio, desvelo, y diversión.

A todos los profesores que han sido parte de mi formación en todos estos años.

A los colegas del proyecto GNA,en especial a Yoandris, que siempre me brindaron su apoyo para culminar con éxito este trabajo de diploma.

A la Revolución por darme la oportunidad de estudiar en esta grandiosa universidad y ser una profesional más con la que se pueda contar en cualquier circunstancia.

DEDICATORIA

Agradezco a Dios todo poderoso, mi Virgencita del Cobre, y todos mis ángeles guardianes, por guiarme a lo largo de mi existencia, ser el apoyo y fortaleza en aquellos momentos de dificultad y debilidad.

A mis padres: Yanira y Juan Francisco, por ser los principales promotores de mis sueños, por creer en mis expectativas, por los consejos, valores y principios inculcados.

A mi hermano que con sus palabras me hace sentir orgullosa de lo que soy y de lo que le puedo enseñar.

A mi novio amado, por ser mi guía y acompañarme en tantos momentos, brindándome su paciencia y sabiduría para culminar con éxitos cada una de mis metas.

A mi familia y amigos que aportaron su granito de arena para poder lograr este sueño de todos.

RESUMEN

Actualmentese ha evidenciado una alta tasa de error en palabras o datos escritos idénticos, debido a errores ortográficos y tipográficos, variaciones en el orden de las palabras y el uso de prefijos y sufijos en los procesos de gestión de información en los sistemas. Existen algoritmos que han sido trabajadospara simplificar la búsqueda en las bases de datos cuando sólo se conoce la pronunciación de un nombre propio, pero no su trascripción exacta. En el Sistema de Gestión Integral de Aduanas se está utilizando un algoritmo fonético que no brinda resultadoscorrectos 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 institución.Para darle solución a esta temática, 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 la implementación de un algoritmo fonético, que fue desarrollado mediante la metodología AUP en su versión UCI, basada en el framework Symfony en su versión 1.2 y el lenguaje de programación PHP en su versión 5.6.Además, se cumplió con las disciplinas definidas por la metodología empleada: Modelado de negocio, Requisitos, Análisis y diseño, Implementación y Pruebas. 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 enla Aduana General de la República.

Palabras clave: algoritmos fonéticos, codificación fonética, Caverphone,Double Metaphone,Metaphone, NYSIIS, Soundex.

ABSTRACT

Nowadays a high rate of error has evidenced in words or identical written data, due to spelling and typographical errors, variations in word order and the use of prefixes and suffixes in the information management processes in the systems. There are algorithms that have been worked on to simplify the search in databases when only the pronunciation of a proper name is known, but not its exact transcription. 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 with it errors of an orthographic and typographical nature, in the processed data of the persons who interact with institution. In order to solve this issue, 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 implementation of a phonetic algorithm, which was developed using the AUP methodology in its UCI version, based on the Symfony framework in its 1.2 version and the PHP programming language in its 5.6 version. In addition, the disciplines defined by the methodology employed were met: Business Modeling, Requirements, Analysis and Design, Implementation and Testing. The proposed algorithm will contribute to a better decision making process, helping to eliminate errors that could be committed in the process of information management in the General Customs of the Republic.

Keywords: *Caverphone, Double Metaphone, Metaphone, NYSIIS, phonetic algorithms, phonetic encoding, Soundex.*

ÍNDICE

INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (TIC) hansido consideradas en los últimos 50 años, como elementos esenciales de la Sociedad de la Información, estas habilitan la capacidad universal de acceder y contribuir a la información, las ideas y el conocimiento (Romaní, 2009).Las TIC forman parte de la cultura tecnológica que rodea al hombre y con la cual se debe convivir.La misma provocacontinuas transformaciones en las estructuras económicas, sociales y culturales, e incide en casi todos los aspectos de la vida diaria. Su gran impacto en todos los ámbitos hace cada vez más difícil que se pueda actuar eficientemente prescindiendo de ellas, convirtiéndose en una herramienta esencial para impulsar el desarrollo de una nación(Milanés, 2018).

Tomando en consideración esa realidad mundial, Cuba defiende que las tecnologías deben ser empleadas para lograr el desarrollo económico y social. Pero a su vez, pueden ser utilizadas para promover la paz, el conocimiento, erradicar la pobreza y la exclusión social(Di-Lella, 2019). En este sentido, se promueve la construcción de una sociedad de la información y el conocimiento centrada en la persona, integrada y orientada al desarrollo sostenible en el que todos pueden crear, consultar, utilizar y compartir la información dentro del marco regulatorio, legal y normativo que sustenta el proceso de informatización del país (Ministerio de Comunicaciones, 2017).

Para llevar a cabo esta transformación digital, en el país se han desarrollado múltiples acciones encaminadas a lograr la informatización de la sociedad.En este marco, esas acciones están relacionadas con el proceso de utilización ordenada y masiva de las TIC, para satisfacer las necesidades de información y conocimiento. Lo que constituye una de las metas que tiene Cuba, en el presente y para los próximos años(Collado et al., 2016).

El proceso de inclusión de la informática en las empresas cubanas ha brindado resultadospositivos, pues estas instituciones estatales hanllegado a ser consideradas como actores principales en la industria de aplicaciones informáticas.La Aduana General de la República (AGR), es un órgano de la Administración del Estado que está inmersa en el proceso de informatización. Esta entidad se encarga del control en frontera y de la actividad interna vinculada al comercio exterior,garantizando la seguridad y protección de la sociedad socialista y de la economía nacional. Así como la recaudación fiscal y las estadísticas del comercio exterior, a través del cumplimiento de las políticas estatales de competencia aduanera para el tráfico internacional de viajeros, mercancías y medios de transporte(Aduana General de la República de Cuba, 2019).

La Universidad de las Ciencias Informáticas (UCI) y la AGR, están unidas por un convenio de colaboración, del cual ha surgido un sistema que gestiona los procesos que tienen lugar en la

Aduana, conocido como el Sistema de Gestión Integral de Aduanas (GINA). El uso del sistema, contribuye a que se evite la entrada al país de drogas, armamento o de cualquier otro tipo de sustancias tóxicas(García, 2018). Ayudandoademás,en la alerta tempranade las entidades correspondientes para actuar sobre las personas que incurren en diversos delitos en el país.

En elúltimosemestre del año 2019, Cubafue visitada poralrededor de 5 millones de visitantes internacionales, y más de 50 mil cubanos han viajado al exterior en los últimos cinco años(Digital, 2018). Lo que trae aparejado un aumento acelerado de la cantidad de información disponible, incrementándose el número de personas registradas en la base de datos(BD) del sistema GINA.

En los módulos de GINA, se registran los datos de identificación de las personas naturales, así como información clasificada del individuo. La información que se maneja está propensa a contener errores ortográficos, fonéticos o tipográficos (que darían lugar a omisiones, inserciones o sustituciones de caracteres en los nombres propios),razón por la cual se puede afectar la recuperación o búsqueda de información que se realiza en el sistema. Con el avance acelerado de las TICs, se ha demostrado que la codificación fonética persigue el objetivo de recuperar información que tiene similitud sonora; y cuya presentación a través de la palabra puede diferir de su pronunciación. Estos métodos de codificación permiten realizar búsquedas en un amplio dominio que comprende las diferentes variantes de los nombres, apellidos o localidades, basándose principalmente en la pronunciación de la palabra en lugar de su grafía.

En el sistema se utiliza un algoritmo con el objetivo de lograr el reconocimiento fonético, en el cual, se detectaron varias deficiencias que se presentan a continuación:

- El algoritmo utilizado está implementado bajo los principios de la fonética para el idioma inglés, haciendo uso de las reglas definidas para ese idioma, por lo que cuando se realizan búsquedas de nombres más comunes en el idioma español, no muestra resultados correctos. Además de ser evidente que el codificador fonético implementado, no corresponde a la perfección con el uso de caracteres especiales y pronunciaciones específicas del idioma español.

- Se implementó un algoritmo que mezcla principios de codificación de los algoritmos *Soundex* y *Metaphone* para el idioma oficial en el que fueron desarrollados, donde los caracteres que se definieron para conformar los grupos de similitud fonética, no corresponden a los caracteres definidos para el idioma español. En este algoritmo no se contemplan las consonantes (Ñ) y (H) que forman parte del alfabeto del idioma.

- En el proceso de construcción del código fonético, dada una cadena normalizada, el resultado no cumple con la cantidad de dígitos establecidos. Esto ocurre cuando la cadena

normalizada comienza con una vocal, ya que no se consideró que a las vocales no se les asignaban grupos numéricos, sino que eran eliminadas en la salida del algoritmo.

- Según la agrupación definida para los caracteres, el algoritmo fonético utilizado, muestra errores en cuanto a la codificación de cadenas, presentado en el resultado final similitudes en nombres que no concuerdan fonéticamente, como es el caso de Olga y Alexis, que el algoritmo los codificaba con igual valor.

Todos estos inconvenientes traen 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 en el momento de subsanar esos errores.

Por lo que se hace necesariounamodificación del algoritmo de búsqueda fonética utilizado en GINA, para garantizarla unicidad e integridad de los datos de las personas almacenadas en la base de datos. Además de lograrel reconocimiento de una persona, por la forma en la que se pronuncia su nombre; siendo imprescindible para identificar los hechos, indicios e ilícitos en que incurren personas naturales consideradas de interés aduanal, procedentes del tráfico mercantil, viajero o postal, permitiendo detectar posibles violaciones de las leyes migratorias vigentes en el país y donde es de vital importancia trabajar con datos específicos muy relevantes como es el caso de los nombres propios de las personas.

A partir de lo planteado como situación problémica se formula el siguiente **problema a resolver:**¿Cómo aumentar la eficacia en la identificación de personas registradas en el sistema GINA a partir de la semejanza sonora de sus datos primarios?

El diseño de la investigación permite declarar como **objeto de estudio**los algoritmos de búsqueda fonética, enmarcado en el **campo de acción** de la identificación de personas mediante la aplicación de algoritmos de búsqueda fonética en el sistema GINA.

Se define como **objetivo general**desarrollar una modificación del algoritmo de búsqueda fonética para el sistema GINA,en aras de aumentar la eficacia en la identificación de personas a partir de la semejanza sonora de sus datos primarios.

Para dar cumplimiento al objetivo general se plantean los siguientes **objetivos específicos:**

1. Establecer el marco teórico para el estudio de algoritmos de codificaciónfonética, así como la adecuación a la etimología de los datos primarios en el idioma español paramejorar el algoritmo de búsqueda fonética utilizando las principales tecnologías, herramientas y métodosadecuados.

2. Realizar el Modelado de negocio, Requisitos y Análisis y diseño para la implementación de la propuesta de solución.

3. Implementar el algoritmo de búsqueda fonética para el sistema GINA.

4. Validar la solucióndesarrollada, mediante la aplicación de métodos estadísticos y pruebas de softwareque permitan evaluar su valor potencial.

Para el desarrollo de la investigación se emplearon los siguientes **métodos de la investigación científica:**

Métodos teóricos:

- Analítico-Sintético: el uso del mismo permitió descomponer el problema de investigación en varias partes. Con este se pudo profundizar en los fundamentos de los algoritmos fonéticos, así como las modificaciones a realizar en dependencia de la etimología del idioma español.De esta formase pudo determinar los principios fundamentales que debe cumplir la propuesta de solución.

- Modelación: es utilizado para la representación matemática del problema, partiendo de los fundamentos de los algoritmos fonéticos analizados.

- Histórico-Lógico:se estudió la existencia de los algoritmos fonéticos y su comportamiento en el transcursodel tiempo, permitiendo así profundizar en las particularidades de cada uno de ellos, con el fin de obtener experiencias propias para la futura implementación.

Métodos empíricos:

- Observación: se utilizó para recopilar información de las necesidades existentes en el sistema GINA, con el objetivo de contar con un registro de datos válidos, confiables y tomados de manera sistemática, sobre el uso del algoritmoy los resultados brindados por el mismo en el sistema.

- Entrevista: se empleó para recoger los criterios de los especialistas que interactúan con el sistema GINA, sobre el comportamiento del algoritmo actual.

- Medición: se utilizó para precisar toda la información numérica del algoritmo en relación con los resultados obtenidos cuando se realizan búsquedas en el sistema GINA.

Estructura de la tesis

La tesis está compuesta por tres capítulos que responden a los objetivos específicos anteriormente descritos. Cada uno de ellos contiene los elementos necesarios que conforman la solución final, dividiendo el estudio del problema planteado.

En el capítulo 1 se abordan los aspectos teóricos más relevantes de los algoritmos de búsqueda basados en la fonetización de las cadenas de caracteres, así como el estudio de sus limitaciones y principales características.Se enuncian también los principios de funcionamiento de los algoritmos de normalización y codificación canónica, y la definición de los grupos de similitud fonética para el idioma español.Se describen además las tecnologías, metodología, herramientas y el lenguaje de programación utilizado en el desarrollo de la solución.

En el capítulo 2 se exponen los elementos que permiten describir la propuesta de solución, tales como: requisitos para la implementación del algoritmo fonético, teniendo en cuenta las modificaciones para el idioma español. Se modelan los aspectos del diseño basados en el diagrama de clases de diseño, también se muestra modelo entidad relacióndonde se especifican las clases principales con las que interactúa el algoritmo.

En el capítulo 3 se realiza la implementación de la propuesta de solución, donde se obtiene un algoritmo fonético que cumple con las reglas ortográficas y fonéticas establecidas para el procesamiento de cadenas en el idioma español. Se describen además,un conjunto de pruebas realizadas al algoritmo, una vez que concluye la implementación. Asegurando que este cumple con las especificaciones requeridas.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se abordan los elementos teóricos de los algoritmos de búsqueda basados en la fonetización de las cadenas de caracteres. Así como el estudio de diferentes sistemas, donde se han utilizado estos algoritmos en sus diversas variantes. Se exponensus limitaciones y principales características, partiendo además del principio de funcionamiento de los algoritmos de normalización canónica, basados en los grupos de similitud fonética definidos para el idioma español.También se describen las diferentes herramientas a utilizar durante el proceso de desarrollo de la solución planteada, así como el uso de patrones de arquitectura.

1.2 Conceptos fonológicos para la comprensión del problema

Para la comprensión total de la investigación, se debe tener en cuenta una serie de conceptos y definiciones asociadas al dominio del problema.

Fonética

Se ocupa de la realización concreta del sonido, es decir, de su materialidad. Describe el sonido en sus cualidades físicas: **tono** (altura musical), **intensidad** (energía articulatoria), **cantidad** (duración en el tiempo) y **timbre** (depende de la cavidad bucal). Además, la fonética se ocupa de la producción y percepción del sonido. No solo se ocupa de analizar los sonidos aislados, sino también ciertas combinaciones de sonidos que constituyen las sílabas(Hualde, 2014).

A partir de la definiciónantes mencionada se concluye que esta rama otorga una forma precisa, única y estandarizada para representar los sonidos de cualquier lenguaje oral.

Alfabeto Fonético Internacional (AFI en español, e IPA en inglés)

Es un sistema de notación fonética creado por lingüistas. Su propósito es la representación de los sonidos de cualquier lenguaje oral. En su forma básica tiene aproximadamente 107 símbolos básicos y 55 modificadores (Ríos, 1996).

Partiendo de esta definición se concluye que el alfabeto fonético es un sistema de notación para la representación fonética.

Grafema

Unidad mínima e indivisible de la escritura de una lengua. Para inventariar los grafemas que intervienen en la escritura de una lengua, lo que se debe hacer es ir comparando palabras escritas para descubrir diferencias mínimas que van asociadas a un cambio de significado (Bustos, 2007).

A partir de la definiciónantes mencionada se concluye que el grafema es la parte más pequeña e indivisible del lenguaje escrito.

Fonema

Segmento contrastivoenunalengua. Mínimaunidad desonidoconvalor contrastivo (Hualde, 2014).Partiendo de esta definición se concluye que los fonemas son los sonidos del habla, portadores de una intención significativa especial, que resulta de la descripción teórica de los sonidos de la lengua.

Algoritmo fonético

Es un algoritmo de indexación de palabras con su pronunciación, estos algoritmos obedecen a la necesidad de recuperar informacióna partir de palabras o n-gramas[1](Jurafsky & Martin, 2019)mediante su semejanza sonora (Benitez, 2017).Partiendo de esta definición se concluye que los algoritmos fonéticos convierten las palabras en código para simplificar el espacio de búsqueda en los sistemas.

El estudio de estos conceptos permitió considerar a la fonética como una de las disciplinasmás abarcadoras en el área del estudio del lenguaje, además de permitir la construcción de reglas fonéticas para el idioma español, centradas en la unificación de las diferentes combinaciones de grafemas a través de fonemas representativos que unifiquen los sonidos similares resultantes.

1.3Análisis de principales sistemas donde se utilizan los algoritmos fonéticos

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).

Existen actualmente diferentes sistemas informáticos donde se han utilizado algoritmos fonéticos. Para identificar la tendencia actual en cuanto a la utilización de un algoritmo específico es necesario realizar un estudio de sistemas que, por sus características, hayan empleado los algoritmos de codificación fonética para la búsqueda de información.

Los sistemas identificados permiten analizar un espectro mayor de alcance, a diferencia de otros sistemas de corrección ortográfica, pues las palabras recogidas dentro de sus resultados de

[1]N-grama: representa una secuencia conformada por n palabras.

búsqueda, no proceden de una falta de escritura, sino de la equiparación sonora de los términos buscados contra otros ya existentes.

Sistema para búsquedas en las listas negras de la OFAC

En la Ciudad de México en el año 2015, se llevaron a cabo varias actividades de monitoreo para combatir el lavado de dinero y el financiamiento del terrorismo. Estas acciones fueron monitoreadas con herramientas de software, que permitieron seguir el paso en tiempo real de operaciones como transacciones, transferencias u operaciones en ventanilla de alguna entidadfinanciera(Cámara, 2016).Para esto se utilizaron algoritmos de coincidencia fonética, que su objetivo era buscar en las listas negras de la Oficina de Control de Activos Extranjeros(OFAC, por las siglas de *Office of Foreign Assets Control*), brindando resultados certeros y en el menor tiempo posible de todas aquellas personas u organizaciones que hayan incurrido en sanciones económicas.

Instituto Nacional de Defensa de la Competencia y de la Protección de la Propiedad Intelectual(INDECOPI)

En el Instituto Nacional de Defensa de la Competencia y de la Protección de la Propiedad Intelectual, se desarrolló un sistema de búsqueda fonética con el objetivo de proteger todas las formas de propiedad intelectual: desde los signos distintivos y los derechos de autor, hasta las patentes y la biotecnología(*INDECOPI*, 2020).

Sistema de Marcas y Signos Distintivos (*Webmarks*)

El sistema *Webmarks* fue desarrollado con la finalidad de ser un buscador de marcas y signos distintivos de un estudio de abogados, que mantiene información de sus clientes, y requiere un procedimiento de detección de nombres de empresas de pronunciación similar. Los algoritmos fonéticos se utilizaronpara alcanzar un acercamiento a los probables infractores que utilizan la semejanza fonética como competencia desleal, para que el consumidor pueda confundirse con el producto verdadero(Gonzáles-Cam, 2008).

1.3.1 Valoración de los sistemas analizados

Desde los primeros usos de los algoritmos fonéticos en 1930, se ha requerido que la información registrada tuviera confiabilidad en la forma que se tomaron los datos, y por cuestiones de recuperación se necesitaba tener una aproximación exacta del término.

En cada uno de los sistemas antes mencionados,se utilizó el algoritmo más antiguo de todos, el *Soundex,*ya que permitiótener una exactitud en la recuperación de los datos en esos sistemas. A partir del uso de este algoritmo en sistemas vigentes en el mundo, se considera al algoritmo

*Soundex*como parte de la solución propuesta en esta tesis, con modificaciones para ser adaptado a las reglas de normalización del idioma español.

1.4Importancia de la utilización de los algoritmos fonéticos

La codificación fonética busca representar con un mismo código aquellas palabras cuya pronunciación es similar partiendo de su forma escrita, permitiendo descubrir palabras fonéticamente similares. Las técnicas fonéticas para detección de duplicados existentes no están orientadas al idioma español, lo que en ocasiones dificulta la identificación y corrección de problemas como errores ortográficos en textos escritos en este idioma(Amón et al., 2012).

Una manera de ilustrar el proceso de gestión que se lleva en la entidad aduanera, es cuando la persona ingresa al país, su nombre queda archivado en las bases de datos de la Aduana. Dicha información puede presentar errores ortográficos o tipográficos, uso de abreviaturas o simplemente palabras faltantes, transposición, y sustitución de caracteres (Bustamante & Díaz, 2006).Por lo que es recomendable la utilización de algoritmos fonéticos que ayuden a unificar todas esas soluciones encontradas, en un solo resultado, el más certero. 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), *NYSIIS*(Gálvez, 2007). Estas técnicas,cabe destacar, se basan en la pronunciación de las palabras en el idioma inglés.

A partir de lo anterior, se hace necesario un estudio y comparación de algoritmos fonéticos para realizar búsquedas dentro de los sistemas de información, complementando los aspectos teóricos más relevantes de la fonetización de las cadenas de caracteres, el estudio de sus principales características y limitaciones, y el principio de funcionamiento de los algoritmos de normalización y codificación, basados en los grupos de similitud fonética definidos para el idioma español. Para realizar este estudio se utilizará el método de mapeo sistemático en la literatura científica, con el objetivo de alcanzar un conocimiento más profundo sobre este tema.

El mapeo sistemático es un método útil para construir clasificaciones y obtener información sobre el conocimiento existente en una temática específica; por tanto, permite identificar los vacíos y las necesidades en un área determinada, con lo que se acerca a la definición de un nicho de investigación pertinente (Jácome et al., 2018).

En (Gonzáles-Cam, 2008) se revisan algunos elementos de la aplicación de los algoritmos fonéticos en un sistema de marcas y signos distintivos, pero no se aborda sobre las principales diferencias entre ellos. Además, en (Alejandro & Barragán, 2009)los autores llevan a cabo una

revisión sistemática de literatura para capturar el estado del arte de estos algoritmos en los sistemas de información. No obstante, solo se focalizan en las técnicas de recuperación de la información, quedando por analizar el alcance de la codificación fonética. Finalmente en(Manuel & González, 2010) se hace un resumen sobre estos algoritmos encaminados a darle solución a una problemática en particular aunque en este estudio no se abordan los temas de codificación, ni los principios de implementación a tener en cuenta para la aplicación de los algoritmos.De esta forma, se contempla el establecimiento de la línea base del estado del arte sobre la aplicación de los algoritmos fonéticos en los sistemas de información, a partir de la cual, la comunidad científica puede desarrollar propuestas de mejora.

El mapeo se define como un proceso y una estructura de informe que permite categorizar los resultados que han sido publicados hasta el momento en un área específica. A pesar de ser poco exhaustivo permite obtener una visión científica e identificar tendencias sobre el empleo de los algoritmos fonéticos en determinados sistemas(Toledano López, 2019). Para ello el mapeo sistemático determina realizar primero preguntas de investigación para obtener estudios primarios, se definen criterios de análisis y luego se discuten los resultados.

Se plantean varias macropreguntas para resolver a través del mapeo sistemático que serán respondidas en los próximos acápites:

1. ¿Cuáles son los algoritmos fonéticos que se conocen en la actualidad, así como sus principales características?

2. ¿Cómo se realizan las particiones de datos hechas por los algoritmos?

3. ¿Cuáles son las principales diferencias entre los algoritmos analizados?

En esta investigación se seleccionaron las bases de datos*Google Scholar, Web of Science, Springer,*consideradasactualmente como fuentes de búsquedasque pueden contener artículos referidos al tema de la investigación. También se realizaron búsquedas en internet, limitándose a artículos de publicaciones académicas que mostraran texto completo, también entre las referencias de los artículos seleccionados y otros artículos ya identificados. El período de las búsquedas comprendió desde el año 2009, inclusive, a febrero del 2019.

Para conformar la cadena de búsqueda en las bases de datos de publicaciones se tuvo en cuenta el foco central de la investigación que eran los algoritmos fonéticos aplicados a los sistemas de información. En el caso de las palabras clave "algoritmos fonéticos " se seleccionaron, además conceptos normalmente utilizados en el área de investigación, tales como: codificación fonética y similitud fonética.

Para la selección de estudios se consideraron los siguientes criterios de inclusión/exclusión:

- Son elegibles las publicaciones científicas relacionadas con los algoritmos de codificación fonética donde se propusieran, discutieran o evaluaran los principales algoritmos o algunas variaciones de los mismos.
- No hubo restricciones sobre el área de aplicación de las publicaciones, siempre y cuando, el enfoque fuera los algoritmos de codificación fonética.
- Se incluyeron estudios cualitativos y cuantitativos realizados por profesionales en el tema.
- Se excluyeron los artículos que solo tratan el uso específico de los algoritmos fonéticos, sin especificar un proceso real donde estuvieran incluidos.
- Si se encuentran varios artículos del mismo grupo de investigadores que tratan la misma propuesta desde diferentes perspectivas, se considerará el de mayor completitud.
- Se excluyeron todos aquellos estudios en los que su enfoque principal se alejara de los algoritmos fonéticos.

1.4.1Algoritmos de codificación fonética. Principales características

La codificación basada en la similitud fonética de los datos primarios de las personas se aplica principalmente a los nombres y apellidos para reducirlos a una forma común. Entre los procedimientos de codificación fonética más conocidos se encuentraelsistema*Soundex* desarrollado en 1918, *Daitch-MokotoffSoundex* de 1985, *Metaphone* de 1990 y *NYSIIS* en 1970.

Algoritmo fonético*Soundex*

Este algoritmo fue desarrollado en 1918 por *Robert Russell*y *Margaret Odell*, patentado finalmente en 1922. Una variante de este algoritmo fue llamada*American Soundex*, y utilizado en 1930 para el análisis retrospectivo de los censos de Estados Unidos. *Soundex* es un sistema mediante el cual los valores se asignan a los nombres, de tal manera que los nombres que suenan similares obtienen el mismo valor. Estos valores se conocen como codificaciones de *Soundex*. Una aplicación de búsqueda basada en *Soundex* no buscará un nombre directamente, sino que buscará la codificación correspondiente. Al hacerlo, obtendrá todos los nombres que suenan como el que se busca (Bhatti et al., 2014).

Este algoritmo en su versión estadounidense, coincide con palabras del índice numérico como A126. Su principio de funcionamiento se basa en la partición de consonantes en grupos con números ordinales, que luego se compilan al valor resultante(Gonzáles-Cam, 2008).

Tabla1. Particiones del algoritmo Soundex

Soundex Original	Soundex Refinado

Grafemas	Código	Grafemas	Código
B P F V	1	B P	1
C S K G J Q X Z	2	F V	2
D T	3	C K S	3
L	4	G J	4
M N	5	Q X Z	5
R	6	D T	6
		L	7
		M N	8
		R	9

Fuente.(Del Pilar Angeles et al., 2015)

Las letras que no están enumeradas en la tabla (todas las vocales y algunas consonantes) son ignoradas. Letras adyacentes o letras del mismo grupo separadas por H o W están escritas como una letra. El resultado se trunca a 4 caracteres. Las posiciones perdidas se rellenan con ceros.

Ejemplos: Cielo (C400), Giraldo (C643).

Limitaciones

Es claro que,*Soundex* no está orientado al español ya que ni siquiera contempla el juego de caracteres (“ñ”, “ll”). Asimismo, la dependencia de la letra inicial, la agrupación por punto de articulación del idioma inglés y el estar limitado a cuatro caracteres, implica que no es eficiente para detectar errores ortográficos comunes en el idioma español(Amón et al., 2012).

Algoritmo fonético *Metaphone*

En 1990 *Lawrence Phillips* publicó un artículo que describía un sistema *Soundex* más avanzado al que llamó *Metaphone*. El mismo intenta producir su codificación en función de cómo se pronuncia un nombre en lugar de como se deletrea(P.Parmar & K Kumbharana, 2014). Al igual que *Daitch-Mokotoff*, también usa secuencias de letras en lugar de solo letras. A diferencia de todos sus predecesores, basa sus codificaciones en el nombre completo en lugar de truncar, después de considerar solo una parte inicial del nombre. Este algoritmo genera una codificación única para cada nombre, como se hizo en los sistemas anteriores(Beider & Morse, 2010).

El código final es un conjunto de caracteres 0BFHJKLMNPRSTWXY, pero al comienzo de una palabra también pueden aparecer vocales del conjunto AEIOU(Koneru et al., 2016).

El algoritmo de *Metaphone* tiene una eficiencia mayor que el algoritmo anterior. Tiene un enfoque diferente al proceso de codificación: transforma la palabra original usando reglas de pronunciación en inglés, por lo que las reglas de conversión son mucho más complicadas.El algoritmo pierde mucha menos información, ya que las letras no están divididas en grupos.

Tabla 2. Algoritmo de cálculo del código *Metaphone*

Entrada	Sea$S$ una cadena de caracteres
Salida	Sea S' cadena de caracteres que representa el código final
Pasos	

1. Eliminar todas las letras vecinas que se repiten, excepto la letra C.
2. El comienzo de la palabra debe transformarse usando las siguientes reglas:
    a. KN → N
    b. GN → N
    c. PN → N
    d. AE → E
    e. WR → R
3. Eliminar la letra B al final, si está después de la letra M.
4. Reemplazar C usando las reglas a continuación:
    a. Con X: CIA → XIA, SCH → SKH, CH → XH
    b. Con S: CI → SI, CE → SE, CY → SY
    c. Con K: C → K
5. Reemplazar D usando las siguientes reglas:
    a. Con J: DGE → JGE, DGY → JGY, DGI → JGY
    b. Con T: D → T
6. Reemplazar GH → H, excepto si está al final o antes de una vocal.
7. Reemplazar GN → N y GNED → NED, si están al final.
8. Reemplazar G usando las siguientes reglas:
    a. Con J: GI → JI, GE → JE, GY → JY
    b. Con K: G → K
9. Eliminar toda H después de una vocal, pero no antes de una vocal.
10. Realizar las siguientes transformaciones usando las reglas que a continuación se muestran:
    a. CK → K

b. PH → F
c. Q → K
d. V → F
e. Z → S

11. Reemplazar S con X:
    a. SH → XH
    b. SIO → XIO
    c. SIA → XIA
12. Reemplazar T usando las siguientes reglas:
    a. Con X: TIA → XIA, TIO → XIO
    b. Con 0: TH → 0
    c. Eliminar: TCH → CH
13. Transformar WH → W al principio. Eliminar W si no hay vocal después de eso.
14. Si X está al principio, entonces
    a. reemplazar X → S,

    sino

    b. reemplazar X → KS.
15. Eliminar todas las Y que no están delante de una vocal.
16. Eliminar todas las vocales, excepto la vocal al comienzo de la palabra.

Fuente.(Gálvez, 2007)

Algoritmo fonético *Double Metaphone*

El nombre proviene del hecho de que produce dos codificaciones para cada nombre. Por lo tanto, no tiene la solidez de *Daitch-Mokotoff* que puede tener muchas codificaciones, pero sin duda es más robusto que los sistemas anteriores que tienen solo una codificación por nombre. Otra característica nueva del *Double Metaphone* es que incluye pronunciaciones extranjeras, pero agrupa todas las reglas extranjeras juntas y no distingue qué regla corresponde a qué idioma (Gonzáles-Cam, 2008).

Este algoritmo produce una clave secundaria junto con una palabra primaria codificada para identificar el nativo más común en cuanto a la pronunciación. *Double Metaphone* tiene una codificación extensa para letra "C", "G" y "S", ya que tienen mayor diferenciación en pronunciaciones al combinar con otras vocales y consonantes (P.Parmar & K Kumbharana, 2014). El algoritmo retiene solo el primer sonido de vocal carácter "A" mientras que todos los demás sonidos de las vocales son omitidos. Posteriormente se realizan otras transformaciones en las letras restantes basadas en las letras presentes en el índice sucesor y el índice predecesor(Benitez, 2017).

Algoritmo fonético *NYSIIS*

Otro código fonético fue el propuesto por *Taft*en 1970 y desarrollado por *The New York State Division of Criminal Justice*. El sistema de codificación presentado se denomina *New York State Identification and Intelligence Systems* (*NYSIIS*) y se basa en la reducción de los nombres a un código de hasta 6 letras(Koneru et al., 2016).

Tabla 3. Reglas utilizadas por el algoritmo *NYSIIS* para la codificación fonética

Entrada	Sea S una cadena de caracteres
Salida	Sea S' cadena de caracteres que representa el código final
Pasos	
1. El primer caracter de la clave fonética corresponde al primer caracter del nombre. 2. Traducir los primeros caracteres del nombre. a. MAC → MCC b. PH → FF c. KN → NN d. K → C e. SCH →SSS 3. Traducir los últimos caracteres del nombre. a. EE →Y b. IE → Y c. DT, RT, RD, NT, ND →D d. Si el último caracter es S, eliminar. e. Si el último caracter es A, eliminar. f. Si los últimos caracteres son AY, sustituir por Y.	

Fuente. (Gálvez, 2007)

En su trabajo, *Taft* compara elalgoritmo *NYSIIS* con el *Soundex* y concluye que *NYSIIS* tiene una precisión del 98.72%, que la del*Soundex* es del 95.99% (Gálvez, 2007).

Limitaciones

Este algoritmosólo trata las variantes de los nombres producidas por errores fonéticos por lo que en 1998 *The New York State Division of Criminal Justice* sustituye el sistema *NYSIIS* por el producto *NameSearch*, por medio del cual no sólo se identifican las variantes fonéticas, sino las

producidas por errores de transcripción, formas abreviadas, o variantes originadas por la distinta ordenación de las secuencias de los componentes que forman los nombres personales(Carmen Gálvez, 2016).

Emparejamiento fonético de Beider-Morse

El sistema Beider-Morse, intenta resolver un problema inherente a todos los sistemas de correspondencia inexactos hasta la fecha. El número de coincidencias encontradas suele ser extremadamente grande y se compone de muchos nombres que no pueden considerarse relevantes en ningún momento de la imaginación(Benítez, 2017). El sistema Beider Morse reduce el número de coincidencias irrelevantes al determinar primero el idioma a partir de la ortografía del nombre y luego aplicar reglas de pronunciación basadas en ese idioma específico. En el mismo se considera el nombre completo en lugar de solo una parte inicialproporcionando múltiples codificaciones (Beider & Morse, 2010).

1.4.2 Aplicaciones de los algoritmos fonéticos

La utilización de estos algoritmos ofrece soporte actualmente para un conjunto de interfaces de búsqueda avanzada, implementadas a partir de las necesidades específicas de diferentes sistemas entre los que se incluyen aplicaciones de búsqueda en bibliotecas virtuales como es el caso de *AquaBrowser Library* e *Hibernate Search*. La codificación fonética ha sido extendida también a otras especialidades, por lo que se emplea a menudo en traductores potentes como el desarrollado por *Google*. Además, proporcionan las herramientas de corrección ortográfica de buscadores reconocidos en Internet como *Yahoo* y *SCIRUS Scientific Information*, para corregir los criterios de búsqueda cuando son ingresados erróneamente (Manuel & González, 2010).

Los algoritmos fonéticos se pueden encontraradem ás (P.Parmar & K Kumbharana, 2014):

- Incorporados en sistemas de reconocimiento de voz para identificar la palabra correcta entre palabras fonéticamente similares.
- En la corrección ortográfica ayuda a producir más de una palabra correcta, con pronombres similares.
- En las aplicaciones de búsqueda pueden proporcionar el conjunto de búsquedas relacionadas cuando se le dan palabras con error ortográfico y encuentra palabras codificadas similarmente.
- Útil para que los niños aumenten su vocabulario incluyendo palabras y homófonos que tienen similitudes y que se pronuncian igual, pero con un significado diferente.

La comparación de todos estos algoritmos implica un análisis de tiempo, cuestiones de rendimiento, precisión e implementación. La complejidad del tiempo no es el principal problema de

desempeño, debido a los avances en la programación, mejoras en el lenguaje y en el hardware. Para tal fonética, la precisión del algoritmo, es el factor crítico para la corrección del mismo, porque debido a que el idioma inglés es demasiado grande con la estructura ortográfica compleja que tiene, existe entonces un significado diferente con pronunciación similar (Moreno & Echeverri, 2012).

Tabla 4. Resumen de los principales elementos de los algoritmos

Algoritmos Fonéticos	**Parámetros a evaluar**			
	Año de surgimiento	**Principios de funcionamiento**	**Longitud del resultado final**	**Consideraciones importantes**
Soundex	1918,1930	Este algoritmo transforma los apellidos ingleses en un código de cuatro caracteres. El primer caracter es una letra mayúscula y los tres restantes son dígitos.	4 caracteres	No es eficiente para detectar errores ortográficos comunes en el idioma español.
NYSIIS	1970	Traduce los primeros y los últimos caracteres del nombre, respondiendo a las reglas de normalización y codificación fonética.	6 caracteres	Solo trata las variantes de los nombres producidas por errores fonéticos.
Metaphone	1990	Basa sus codificaciones en el nombre completo en lugar de truncar después de considerar solo una parte inicial del nombre.	6 caracteres	Las reglas de conversión son mucho más complicadas.
Double Metaphone	*2000*	Retorna una codificación primaria y secundaria para	6 caracteres	Incluye pronunciaciones extranjeras, pero agrupa

		diferentes pronunciaciones de una sola palabra en idioma inglés.		todas las reglas extranjeras juntas y no distingue qué regla corresponde a qué idioma.

El estudio detallado de los principales elementos de los algoritmos fonéticos tratados anteriormentepermitióreunir un conjunto de requisitosque deben ser evaluados en cada algoritmo referenciado, con el objetivo de seleccionar los más idóneospara formar parte de la solución del problema planteado.

Después de haber estudiado las características de los algoritmos, se determinó implementar un algoritmo fonético llamado *Meta_Soundex_Algorithm*. El mismo está compuesto por dos algoritmos que se seleccionaron para ser parte de la propuesta de solución: *Soundex* y *Metaphone*. A cada uno de estos se les realizará modificaciones en las reglas de codificación de acuerdo a la etimología del idioma español. El algoritmo *Metaphone* pasa a ser parte de la soluciónya que el mismoutiliza las reglas de pronunciación nativas de la fonética definida para cada idioma, siendo necesario para que el algoritmo sea fácilmente adaptable a las reglas de normalización fonética del castellano.Además,se utilizará el algoritmo *Soundex* porque permite definir un alfabeto que reúne los diferentes grupos basados en la similitud de los fonemasasociados a los sonidos de cada letra.

1.5 Metodología de desarrollo

Uno de los aspectos fundamentales en la elaboración de un software, es seleccionar las tecnologías que mayores beneficios aporten a la solución informática. En este caso, la metodología, las herramientas,tendencias y tecnologías a utilizar ya se encuentran definidas por la dirección del proyecto y por distintas políticas de la dirección del CentroCEIGE. De esta manera, se favorece en el desarrollo, la soberanía tecnológica, la disminución de los costos de producción, así como el mantenimiento y soporte técnico.

Las metodologías de desarrollo de softwareson un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en los sistemas. En un proyecto de desarrollo de software la metodología ayuda a definir: Quién debe hacer,Qué, Cuándoy Cómodebe hacerlo(Ruiz & Angeles, 2017). La metodología parael desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito.

La propuesta de solución presentada en esta tesis, está basada en el uso de una metodología que fue adaptada a las características propias de la universidad y a su vez, de los ciclos de vida definidos para cada proyecto en la institución. El proyecto GINA, ha sido desarrollado bajo los principios de la metodología AUP_UCI, por lo que para la implementación del algoritmo fonético propuesto se determina el uso de la metodología ya definida por la dirección del centro.Esta metodología de desarrollo de software tieneentre sus objetivos aumentar la calidad del softwareque se produce(Y. Rodríguez et al., 2018).

Proceso Unificado Ágil (AUP):constituye una versión simplificada del ProcesoUnificado Racional (RUP, por las siglas de*Rational Unified Process*),desarrollada por *Scott Ambler*. Esta metodologíacombina procesos propios del concepto unificadotradicional con técnicas ágiles, con el objetivo de mejorar la productividad. Permitiendo así describirde manera simple y fácil de entender cómo desarrollar aplicaciones de software de negocio(T. Rodríguez, 2014).

AUP aplica técnicas ágiles, entre las que se incluyen:

- El desarrollo dirigido por pruebas.
- El modelado ágil.
- La gestión de cambios ágil.
- La refactorización de bases de datos para mejorar la productividad.

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, a la que llamaremos Ejecución y se agrega la fase de Cierre.

Fases AUP variante UCI(T. Rodríguez, 2014)

- **Inicio:** durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
- **Ejecución:** en esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, se obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
- **Cierre:** en esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Disciplinas AUP variante UCI(T. Rodríguez, 2014)

AUP define 7 disciplinas (4 ingenieriles y 3 de gestión de proyectos), las disciplinas son:

- **Modelado de negocio:** el Modelado del Negocio es la disciplina destinada a comprender los procesos de negocio de una organización. Se comprende cómo funciona el negocio que se desea informatizar para tener garantías de que el software desarrollado va a cumplir su propósito.
- **Requisitos:** el esfuerzo principal en la disciplina Requisitos es desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto.
- **Análisis y diseño:** en esta disciplina, si se considera necesario, los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de estos, y una descripción que sea fácil de mantener y ayude a la estructuración del sistema (incluyendo su arquitectura). Además, en esta disciplina se modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales. Los modelos desarrollados son más formales y específicos que el de análisis.
- **Implementación:** en la implementación, a partir de los resultados del Análisis y diseño se construye el sistema.
- **Pruebas internas**: en esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas. Se deben desarrollar artefactos de prueba como: diseños de casos de prueba, listas de chequeo y de ser posible componentes de prueba ejecutables para automatizar las pruebas
- **Pruebas de liberación:** pruebas diseñadas y ejecutadas por una entidad certificadora de la calidad externa, a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.
- **Pruebas de aceptación:** es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.

Escenarios para la disciplina Requisitos

A partir de que el Modelado de negocio propone tres variantes a utilizar en los proyectos: Caso de uso del negocio (CUN), Descripción de proceso de negocio (DPN) o Modelo conceptual (MC) y existen tres formas de encapsular los requisitos: Caso de uso del sistema (CUS), Historia de usuarios (HU), y Descripción de requisitos por procesos (DRP). Además, surgen cuatro escenarios

para modelar el sistema en los proyectos, manteniendo en dos de ellos el MC, quedando de la siguiente forma(T. Rodríguez, 2014):

Escenario No 1: proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.

Escenario No 2: proyectos que modelen el negociocon MC solo pueden modelar el sistema con CUS.

Escenario No 3: proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.

Escenario No 4: proyectos que no modelen negocio solo pueden modelar el sistema con HU.

En este caso, el escenario a utilizar es el **Escenario No 3**de la metodología **AUP_UCI** puesto que aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y su continuidad. Se debe tener presente que en el sistema GINA, se utilizó este escenario ya que se estudió el negocio a informatizar, llegando a ser este un negocio complejo porque se gestionan todos los procesos que se llevan a cabo en la Aduana, contando con varios subsistemas, que están relacionados entre sí, y que con su uso prolongado por parte del cliente en la Aduana General de la República suelen surgir peticiones de cambio continuamente.

1.6 Descripción de herramientas y tecnologías para el desarrollo de la propuesta de solución

- **Gestor de base de datos**

Oracle Database es un sistema de gestión de base de datos de tipo objeto-relacional (ORDBMS, por el acrónimo en inglés de *Object-Relational Data Base Management System*).*Oracle Database*ofrece rendimiento, escalabilidad, seguridad y fiabilidad a líderes del sector en una amplia gama de servidores en clúster que ejecutan *Windows*, *Linux* y UNIX. Este gestor proporciona funciones completas para gestionar fácilmente las aplicaciones más exigentes del mundo actual como transacciones u otras acciones en el ámbito empresarial(Oracle.com, 2019).

Para el desarrollo de la propuesta se utilizará*Oracle*en su versión 11G (ejército de procesadores conectados en red con el propósito de manejar enormes tareas de computación mediante la distribución de trabajo entre los diversos recursos de procesamiento), donde esta versión posibilita la creación de aplicaciones de bases de datos personalizadas, además de proporcionar el uso de

estándares del código abierto para ayudar a que las funciones estén creadas específicamente para el negocio en cuestión(Manriquez, 2009).

- **Lenguaje de programación**

En la actualidad existen para el desarrollo de aplicaciones web un gran número de lenguajes de programación. En el desarrollo de la propuesta se utilizará el lenguaje de programación PHP (PHP, por las siglas de*Hypertext Pre-processor)*, que fue desarrollado puntualmente para diseñar páginas web dinámicas programando *scripts* del lado del servidor(Torres, 2016), en su versión 5.6.

PHP se caracteriza por ser un lenguaje gratuito y multiplataforma. Además de su posibilidad de acceso a muchos tipos de bases de datos. También es importante destacar su gran cúmulo de documentación en todo el mundo, siendo de gran ayuda para los programadores.

Ventajas adicionales de PHP(Torres, 2016)

- PHP es completamente expandible y modificable. Está compuesto de un sistema principal, un conjunto de módulos y una variedad de extensiones de código.
- Permite la interacción con gran cantidad de motores de bases de datos tales como *MySQL*, *MS SQL*, *Oracle*, *Informix*, y *PostgreSQL*.
- PHP es *open source.*
- PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.

- **Framework de desarrollo**

En el mundo del desarrollo de software, el término '*framework*' se refiere a las bibliotecas de archivos que incluyen varias funciones básicas. El objetivo del mismo es darle una base que se pueda utilizar para desarrollar proyectos de forma más eficiente.

La solución propuesta en esta tesis utilizará el *framework Symfony*en su versión 1.2.El cual está diseñado para optimizar el desarrollo de las aplicaciones web, basado en el patrón Modelo_Vista_Controlador. Este automatiza las tareas más comunes, lo que permite al desarrollador, dedicarse a tiempo completo a las funcionalidades específicas de la aplicación web. Además de que posee facilidad de instalación y configuración en plataformas *Unix*y *Windows*(*Symfony*, 2019).

Este marco de trabajo es independiente del Sistema Gestor de Base de Datos (SGBD) que se utilice y permite la internacionalización para la traducción del texto de la interfaz, los datos y el contenido de localización. Una de las bondades de este *framework* es que estápreparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa,

además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo(Potencier & Zaninotto, 2008).

Características principales(Eguiluz, 2012):

- Utiliza un *framework* altamente flexible que permite configurar componentes individuales.
- Ofrece la funcionalidad de prueba incorporada.
- Constacon excelente documentación para usar la plataforma.

Propel integrado al framework Symfony

"Propel, que también es un proyecto de software libre, es una de las mejores capas de abstracción objetos-relacional disponibles en PHP. Propel está completamente integrado en Symfony e incluso es su ORM por defecto" (Potencier & Zaninotto, 2008). Ambos desarrollan una combinación rápida y flexible en cuanto al acceso a la información de una base de datos. En la solución se utiliza Propel como capa de persistencia del modelo físico de datos.

- **Capa de presentación**

Este término no es más que una transición que ha ocurrido en los últimos años hacia el desarrollo de aplicaciones que funcionan a través de la Web, enfocada al usuario final. Su objetivo fundamental es reemplazar a las aplicaciones de escritorio por otras que ofrezcan colaboraciones y servicios igual de eficientes, pero a través de un navegador conectado a un servidor de aplicaciones (Manuel & González, 2010).

El término Web 2.0 que utiliza la vista del sistema GINA, está conformado por un entorno dinámico construido a través de un *framework* llamado ExtJs, el cual define un conjunto de librerías *Java Script* que mejoran notablemente la calidad de las aplicaciones desde el punto de vista del usuario final, logrando un entorno gráfico más amigable a partir de un grupo de componentes desarrollados con este fin. También cuenta con validación de formularios, motivo por el cual será utilizado durante la realización de las pruebas al algoritmopara la posterior implantación de la solución.

- **Entorno de desarrollo**

JetBrains PhpStorm es un IDE (IDE,por las siglas de*Integrated Development Environment*) es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica, comercial y multiplataforma para PHP. *PhpStorm* es una herramienta con enorme potencial. Este IDE ofrece edición en directo con tecnologías como CSS, HTML5, *JavaScript* o *TypeScript.*

Proporciona autocompletamiento de código, prevención de errores durante la marcha, además de su integración a las tecnologías *frontend* como *Sass, Less, Stylus*, entre otras(Jetbrains, 2015).

PhpStorm es reconocido por su depurador visual, proporcionando una visión de lo que sucede en la aplicación en cada paso y puede usarse tanto local como remotamente. Además de la posibilidad de realizar pruebas unitarias con *PHPUnit,* brinda su integración a los sistemas de control de versiones más usados como el *Git* (Y. Rodríguez et al., 2018).

La utilización de la versión 9.0,ofrece algunos cambios con respecto a sus predecesores tales como las mejoras importantes en el autocompletamiento de código y el soporte completo para sugerencias de tipo PHP 7 y tipos de retorno (*PhpStorm*, 2019).

- **Servidores de aplicaciones web**

Los servidores de aplicación son una pieza de software de comunicaciones que intermedia entre el servidor en el que están alojados los datos solicitados, y el computador del cliente, permitiendo conexiones bidireccionales o unidireccionales, con cualquier aplicación del cliente.

En el sistema GINA, se utiliza el servidor*Apache*en su versión2.0, que es un servidor web, o servidor HTTP, muy popular. Es uno de los proyectos de código abierto más usados en la actualidad y el servidor web más extendido de Internet.*Apache* es multiplataforma y está disponible en *Windows, Linux, Unix* y *Mac*, aunque en servidores en producción se instala sobre *Linux* habitualmente(Software, 2019).

- **Lenguaje de modelado**

El “Lenguaje de Modelado Unificado” (UML,por las siglas de*Unified Modeling Language*) es un lenguaje basado en diagramas cuyo objetivo general es su utilización para especificar, visualizar, construir y documentar los artefactos de un sistema de software. Captura decisiones y conocimiento sobre sistemas que deben ser construidos. Está pensado para ser utilizado con todos los métodos de desarrollo, etapas del ciclode vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre las técnicas de modelado e incorporar las mejores prácticas de *software* actuales, en una aproximación estándar. UMLincluye conceptos semánticos, notación y principios generales (Rumbaugh et al., 2007).Para el desarrollo de la propuesta de solución se estará utilizando el lenguaje UML en su versión 2.0.

- **Herramienta CASE**

Para el uso de un lenguaje de modelado existen varias herramientas CASE (Ingeniería de Software Asistida por Computadoras), que son diversas aplicaciones informáticas destinadas a

aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y dinero. Estas herramientas, pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software, en tareas como el proceso de realizar un diseño del proyecto, cálculo del costo, compilación automática, documentación o detección de errores(Y. Rodríguez et al., 2018). Para el desarrollo de la solución se propone utilizar el *Visual Paradigm*en su versión 15.1.

Visual Paradigm: es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: Análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Ayuda a una rápida construcción de aplicaciones con calidad y a un menor costo. Permite crear todos los tipos de diagramas de clases, ingeniería inversa, generar código desde diagramas y generar documentación(*Visual Paradigm*, 2019).

1.7Conclusiones parciales

- La necesidad de identificar las variantes de los nombres personales se ha manifestado en varios aspectos de la vida, evidenciándose en diversas aplicaciones, tales como las bibliotecas digitales, las bases de datos de pacientes en un hospital, o los sistemas de reservas aéreas.
- Los algoritmos fonéticos constituyen uno de los procedimientos más usados para obtener cadenas normalizadas.
- Se evidencióque la principal limitación de estos algoritmos es la dependencia del lenguaje para el que son implementados, lo que hace necesario la adaptación de las reglas de codificación en correspondencia de las normas del idioma en que se van a utilizar.
- Se estudiaron la metodología y herramientas necesarias para desarrollar el algoritmo propuesto, como es el caso de la metodología AUP_UCI, la cual guiará el proceso de desarrollo de software y para su construcción se utilizará el framework*Symfony* v1.2 sobre el entorno de desarrollo *PHPStorm*v9.0. Este trabajará en paralelo con el servidor *Apache versión 2.0*.Como gestor de base de datos se empleará *Oracle* en su versión 11G, y como herramienta de modelado *Visual Paradigm* utilizando el lenguaje de modelado UML en su versión 2.0.

CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

2.1 Introducción

En este capítulo se realiza el modelado de negocio, requisitos, Análisis y diseño del algoritmo propuesto. Para ello se obtienen las descripciones de procesos de negocio y el modelo conceptual. Se identifican y describen los requisitos funcionales y no funcionales. Se realizó el diagrama de clases del diseño con estereotipos web, el diagrama de clases del algoritmo y el modelo entidad relación con las clases principales con las que interactúa el algoritmo.

2.2 Modelado del negocio

Para modelar el negocio se utilizó el escenario #3 de la metodología de desarrollo para la actividad productiva de la universidad, con el objetivo de comprender el proceso fundamental del módulo Control de Personas en el sistema GINA. El modelado de negocio se realiza con el objetivo de comprender cómo funciona el negocio que se desea informatizar para tener garantías de que el software desarrollado va a cumplir su propósito(T. Rodríguez, 2014).

2.2.1 Descripción de procesos de negocio

El sistema GINA, cuenta con un módulo llamado Control de Personas. Este subsistema permite garantizar la protección técnica en frontera donde se registran todas las personas que entran y salen del país. Para realizar dicha función es necesario realizar búsquedas en el sistema sobre los datos de estas personas y su historial aduanero, en caso necesario. A continuación, se describe dicho proceso:

- **Gestión de búsqueda de personas**

Este proceso inicia cuando el viajero arriba al país, se dirige inmediatamente a un funcionario aduanero con todos sus documentos personales.El funcionario ingresa su nombre en el sistema para corroborar si esta persona ya está registrada.En caso de no habersido registrado antes, se ingresan sus datos en el sistema y se le entrega la documentación. Pero si el viajero ya ha sido registrado en otras ocasiones, se verifica su historial aduanero, si no presenta ninguna incidencia registrada, se le entrega la documentación, sino, se le informa a las autoridades pertinentes sobre el individuo.

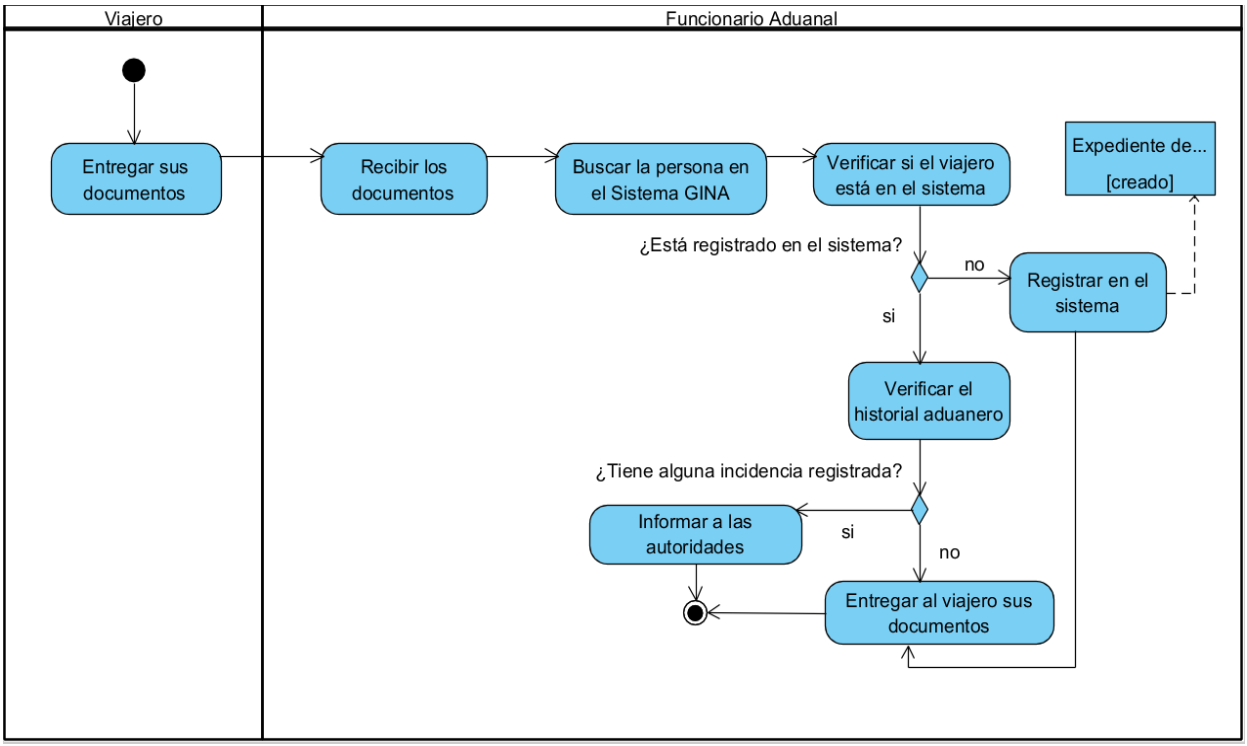

Figura 1. Descripción del proceso Gestión de búsqueda de persona

Para una mayor comprensión de los términos tratados en las descripciones de procesos de negocio, se realizó el modelo conceptual. En este producto de trabajo se detallan cada uno de los conceptos utilizados y sus relaciones.

2.2.2 Modelo conceptual

El modelo conceptual es una representación de conceptos u objetos en el dominio del problema. El objetivo de este modelo es identificar los conceptos obvios expresados en los requerimientos para brindar un conocimiento básico del vocabulario que tienen relación con el problema (Craig, 2003).

La realización del modelo conceptual del proceso de Gestión de búsquedas de personas permitió obtener una representación gráfica de los conceptos que se manejan en dicho proceso, así como las relaciones que existen entre ellos.A continuación, se muestra dicho modelo conceptual:

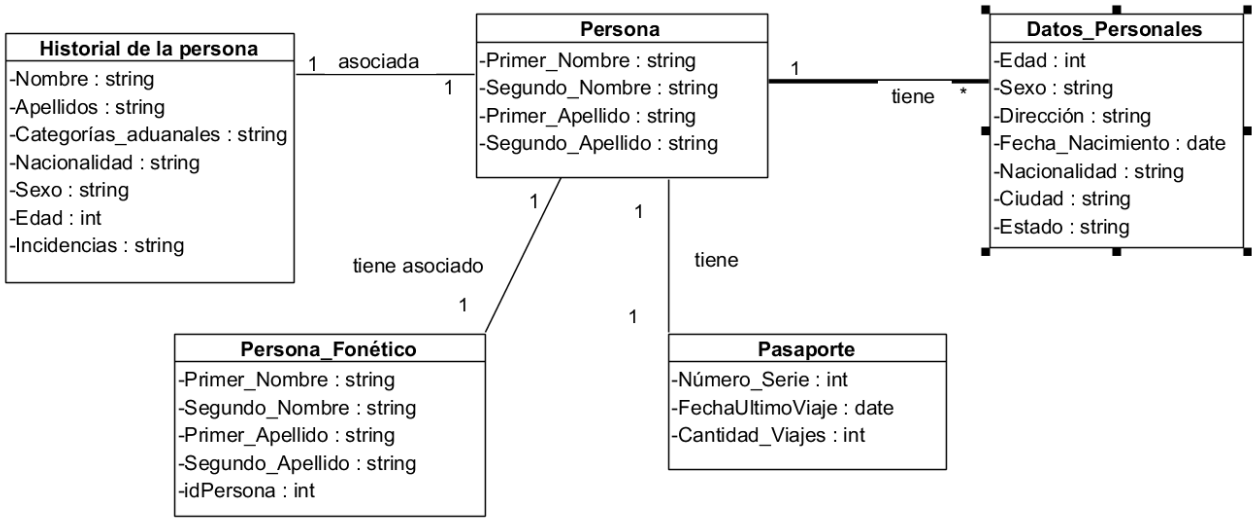

Figura 2. Modelo conceptual

2.3 Requisitos del sistema

Los requerimientos para un sistema son descripciones de lo que el sistema debe hacer: el servicio que ofrece y las restricciones en su operación. Tales requerimientos reflejan las necesidades de los clientes por un sistema que atienda cierto propósito (Sommerville, 2011).Después de haber sido realizadas las descripciones del procesoy el modelo conceptual es posible identificar y describir los requisitos funcionales del sistema.

2.3.1 Técnicas utilizadas para la de obtención de requisitos

Una etapa fundamental en proyectos de ingeniería de software, es la identificación y documentación de los requisitos del futuro sistema al comienzo del proyecto, pues en numerosas ocasiones, se ha demostrado que es cuando pueden prevenirse errores que puedan significar el fracaso del proyecto (Sommerville, 2011). En la ingeniería de requisitos, el levantamiento de los mismo se refiere a la identificación y documentación de los requisitos de un sistema, a partir de los usuarios, clientes o interesados. A la práctica también se le conoce como recopilación de requisitos.

Para realizar el levantamiento de requisitos necesarios para la implementación del algoritmo fonético se utilizaron las siguientes técnicas:

- **Análisis de documentación:**para realizar la modificación del algoritmo se estudiaron los casos de búsqueda donde no se brindaban resultados correctos. Además, se utilizó con el objetivo de reunir detalles de los sistemas ya existentes, incluyendo las reglas de negocio, entidades y atributos que deben ser incluidos en el nuevo algoritmo o que necesitan ser actualizados para el sistema. La ventaja que brinda esta técnica es que no se parte desde

cero y se aprovechan los elementos existentes para descubrir y/ o confirmar los requerimientos. Sin embargo, se corre el riesgo de que la documentación no esté actualizada o no sea válida (Sánchez, 2015).

- **Entrevista:** las entrevistas formales o informales con participantes del sistema son una parte de la mayoría de los procesos de ingeniería de requerimientos (Sommerville, 2011). En las entrevistas, el equipo formuló preguntas a los especialistas que manejan el sistema, para de esta forma identificar los requerimientos que debería cumplir el algoritmo propuesto.

2.3.2 Requisitos funcionales

Los requerimientos del sistema son descripciones de lo que el sistema debe hacer: el servicio que ofrece y las restricciones en su operación.Son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse el sistema en situaciones específicas (Sommerville, 2011).

Luego de aplicar las técnicas para la obtención de requisitos se obtuvieron 12 requisitos funcionales, resumiéndose en 4 agrupaciones de requisitos. A continuación, se enuncian los requisitos funcionales del algoritmo:

1. Generar cadena normalizada mediante el algoritmo *Spanish Metaphone*.
   1.1 Normalizar la cadena original.
   1.2 Filtrar cadena original.
   1.3 Verificar la presencia de vocales en la cadena y eliminarlas.
2. Generar cadena normalizada mediante el algoritmo *Spanish Soundex*.
   2.1 Procesar un arreglo de cadenas.
   2.2 Eliminar las letras repetidas.
   2.3 Filtrar los caracteres especiales de la cadena.
   2.4 Reemplazar caracteres especiales en la cadena.
   2.5 Construir un código fonético para la cadena.
3. Generar código fonético mediante el algoritmo *Meta_Soundex.*
4. Actualizar el código fonético generado.

2.3.3 Requisitos no funcionales

Los requerimientos no funcionales son limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones tanto de temporización y del proceso de desarrollo, como impuestas por los estándares. Los requerimientos no funcionales se suelen aplicar al sistema como un todo, más que a características o a servicios individuales del sistema (Sommerville, 2011).

Para la utilización del sistema GINA se deben tener en cuenta estos requerimientos no funcionales que se enuncian a continuación:

1. Hardware:
    a) **Del lado del servidor:**

    Aplicaciones: Tarjeta de Red:1

    Procesador: 2.6 GHZ,

    RAM: 1GB

    Disco duro: 120 GB

    UPS: 1

    Lector de CD:1

    Base de Datos: Tarjeta de Red: 1

    Procesador: 2.6 GHZ,

    RAM: 1GB

    Disco duro: 120 GB

    UPS: 1

    Lector de CD: 1

    b) **Por el lado cliente:**

    Procesador: 2.5 GHZ,

    RAM: 512Mb (se recomienda 1Gb)

    Tarjeta de Red: 1

2. Software:
    a) Del lado del servidor:
    - **Aplicaciones:** Sistema Operativo *Ubuntu Server*.v10.04 o superior;

      Soporte para PHP 5.3.

      Servidor web *Apache* versión 2.0.

    - **Base de Datos:** Sistema Operativo *Ubuntu Server*.v10.04 o superior;

Oracle versión 11G.

b) Del lado cliente: Navegador *Mozilla Firefox* versión 3.6 o superior

2.3.4 Validación de requisitos

La validación de requisitos es el proceso de verificar que los mismos definan realmente el sistema que en verdad quiere el cliente (Sommerville, 2011).En este proceso se comprueba la validez, la consistencia, la totalidad, el realismo y la verificabilidad de los requisitos.

Durante el proceso de validación de requisitos, se realizó la técnica de generación de casos de prueba, con el fin de que los requisitos sean comprobables.

2.4 Diseño de la propuesta de solución

El diseño arquitectónico se interesa por entender cómo debe organizarse un sistema y cómo tiene que diseñarse la estructura global de ese sistema(Sommerville, 2011). Durante el desarrollo de esta disciplina se modela el algoritmo, teniendo en cuenta la arquitectura, para que haya relación tanto en los requisitos funcionales como los no funcionales definidos con anterioridad, convirtiéndose en precedente para la etapa de implementación

2.4.1Patrón de arquitectura

La arquitectura de *software*, en un sentido estricto, se define como el conjunto de estructuras que componen el sistema, lo que incluye elementos de *software*, las relaciones entre los mismos, y las propiedades tanto de los elementos como de sus relaciones. En otras palabras, la arquitectura de *software* define el conjunto de componentes de un sistema, las interfaces de comunicación de los mismos, y la manera como estos componentes se comunican entre ellos usando estas interfaces(Villegas, 2008).

La arquitectura de un sistema puede basarse en un patrón o en un estilo arquitectónico particular. Un patrón arquitectónico es una descripción de una organización del sistema, tal como una organización cliente-servidor o una arquitectura por capas. Los patrones arquitectónicos captan la esencia de una arquitectura que se usó en diferentes sistemas de *software*(Sommerville, 2011).

La propuesta de solución desarrollada está enmarcada en la arquitectura usada en el Sistema de Gestión Integral de la Aduana, la cual se basa en los principios de la arquitectura Modelo_Vista_Controlador. Esta arquitectura separa la presentación e interacción de los datos del sistema.

Con el objetivo de lograr un mayor entendimiento acerca del actual dominio arquitectónico del sistema GINA, se realizará una breve descripción que comprende los aspectos fundamentales de

cada elemento presente en la arquitectura. El sistema se estructura en tres componentes lógicos que interactúan entre sí:

- El componente **Modelo** maneja los datos del sistema y las operaciones asociadas a esos datos.
- El componente **Vista** define y gestiona cómo se presentan los datos al usuario.
- El componente **Controlador** dirige la interacción del usuario (por ejemplo, teclas oprimidas, clicks del *mouse*, etcétera) y pasa estas interacciones a Vista y Modelo.

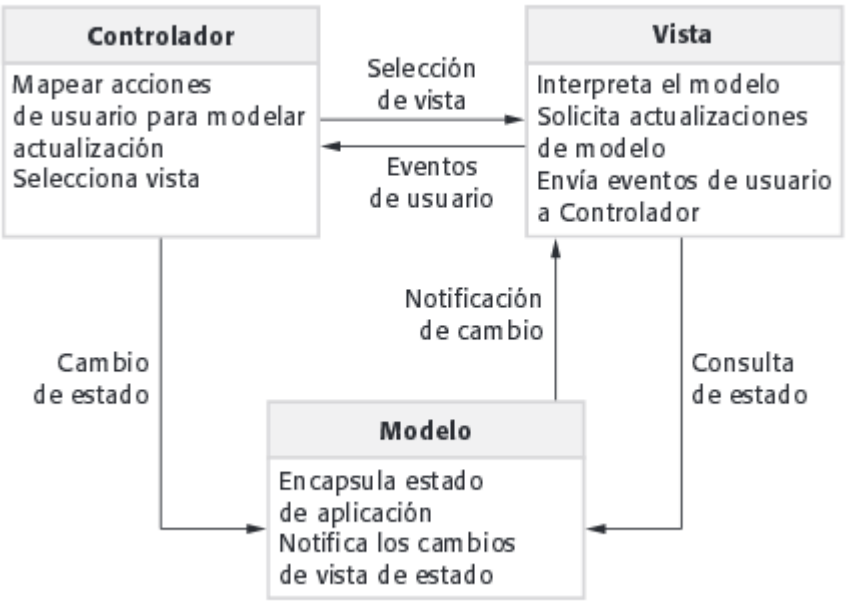

Figura 3. Organización del Modelo Vista Controlador

Fuente.(Sommerville, 2011)

2.4.2 Diagrama de clases de diseño con estereotipos web

El Diagrama de clases del diseño (DCD) es el diagrama principal de la etapa de Análisis y diseño para un sistema. En él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia, además de describir gráficamente las especificaciones de las clases de software y de las interfaces (Craig, 2003).

Se presenta a continuación el diagrama de clases del diseño con estereotipos web del algoritmo implementado, donde se observan las clases participantes y sus respectivas relaciones, basado en los principios del lenguaje de modelado UML, en su versión 2.0.

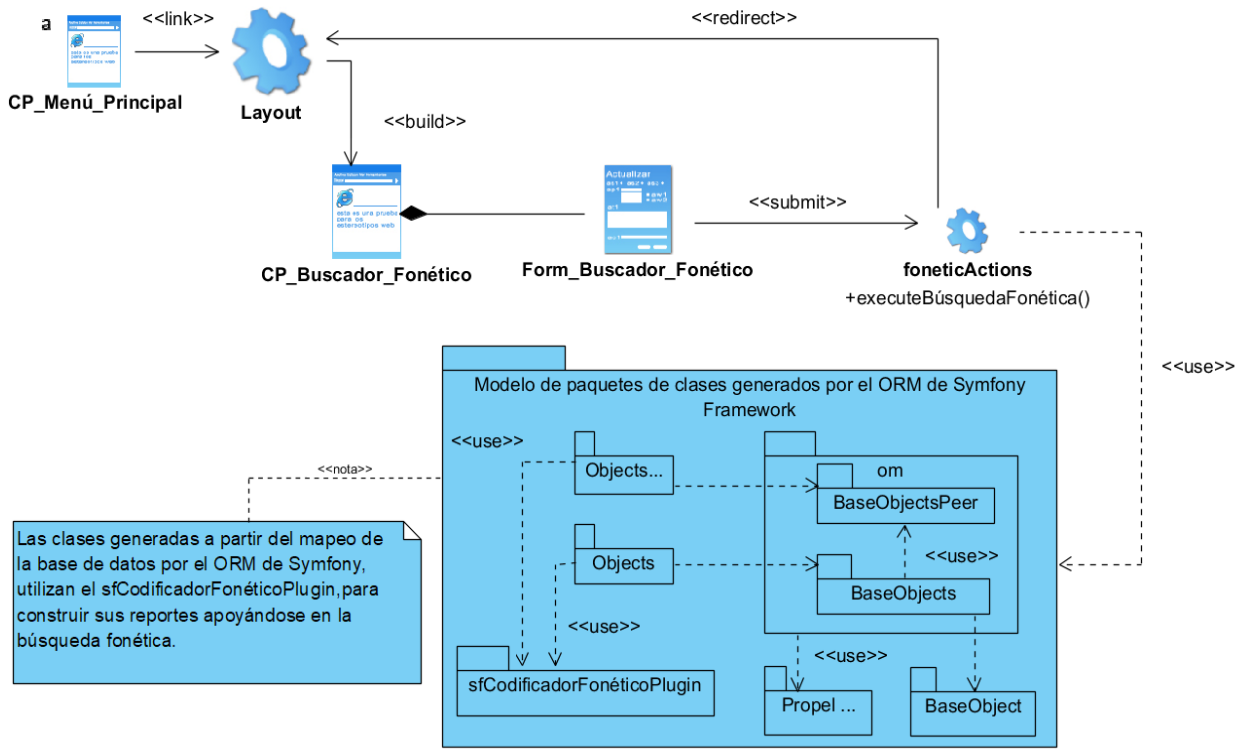

Figura 4. Diagrama de clases de diseño del algoritmo propuesto

2.4.3Diseño de clases del algoritmo desarrollado

Con vista a poder utilizar las bondades del *frameworkSymfony* y el lenguaje PHP, se hace posible separar las reglas de codificación independientes de cada idioma definido para el algoritmo, favoreciendo la reutilización del código para futuras mejoras.

A continuación, se muestran las clases que conforman el algoritmo implementado.

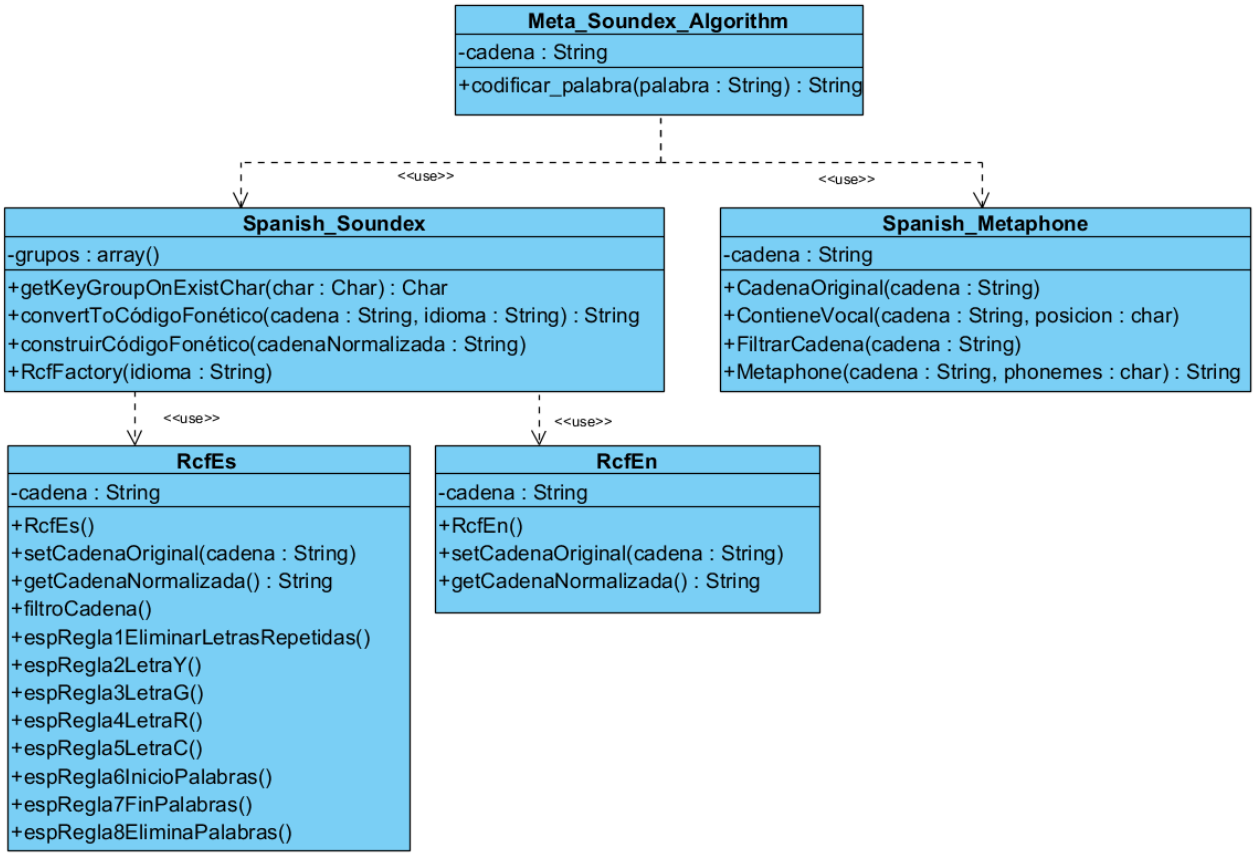

Figura 5. Diagrama de clases

2.4.4Representación mediante el modelo entidad relación

A continuación, se muestra el diagrama entidad relación el cual permite representar algunas de las entidades de la base de datos del sistema GINA, con las que se estará trabajando para desarrollar la propuesta de solución.Este diagrama brinda la posibilidad de almacenar en una entidad independiente, los códigos fonéticos generados a partir del procesamiento de datos realizado por el algoritmo implementado, además de relacionar cada uno con los nombres personales correspondientes.

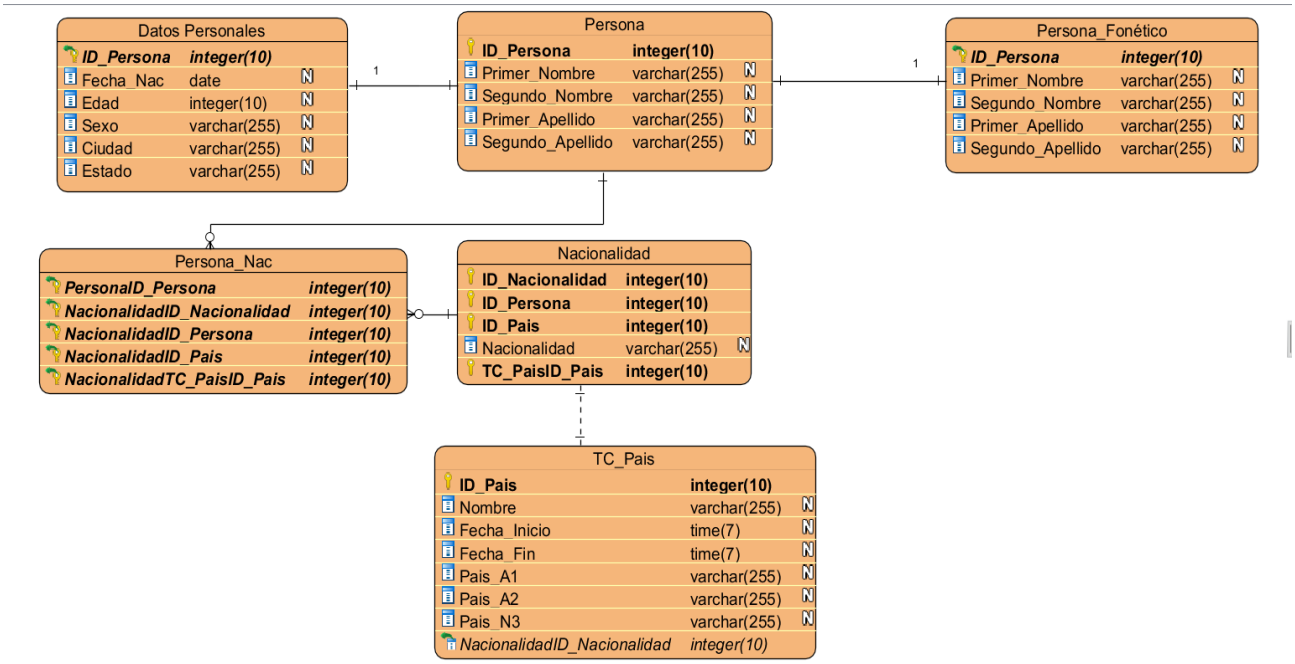

Figura 6. Diagrama de entidad relación

En el diagrama anterior se muestran las entidades con las que el algoritmo estará interactuando en el sistema. Las entidades guardan la información confidencial de los individuos que se han registrado en la Aduana, además de la entidad Persona_Fonético que brinda la posibilidad de almacenar en unaentidad independiente, los códigos fonéticos generados a partir del procesamiento de datos realizado por el algoritmo relacionándolos con los nombres personales correspondientes.

2.5 Patrones de diseño de software

Los patrones de diseño facilitan la reutilización de diseños y arquitecturas exitosas. Un patrón de diseño sistemáticamente nombra, motiva y explica un diseño general que aborda un problema recurrente de diseño en los sistemas orientados a objetos. Describe el problema, la solución, cuándo aplicar la solución y sus consecuencias. También brinda consejos de aplicación y ejemplos. La solución es una disposición general de objetos y clases que resuelven el problema. La misma se adapta y se aplica para resolver el problema en un contexto particular (Gamma et al., 2009).

2.5.1 Patrones GRASP

Los Patrones Generales de Software para Asignar Responsabilidades(GRASP, por las siglas de*General Responsibility Asignment Software Patterns*) describen los principiosfundamentales de la asignación de responsabilidades aobjetos(Craig, 2003). Para la propuesta de solución se pretenden utilizar los siguientes.

- **Bajo acoplamiento:** debe haber pocas dependencias entre las clases. El propósito de este patrón es aumentar la reutilización y eliminar las dependencias entre las clases para propiciar un fácil mantenimiento y entendimiento (Craig, 2003). En el algoritmo se utilizó una biblioteca independiente para cada idioma definido, propiciando que cada una de estas sea independiente, como es el caso de la clase ***RcfEn*** y ***RcfEs***.
- **Alta cohesión:** cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable (Craig, 2003). Este patrón permitirá al equipo de desarrollo que la implementación de las clases encargadas de codificar la información sea fácil de comprender, reutilizar, mantener y poco susceptibles a cambios. En el algoritmo implementado se agruparon las clases según la funcionalidad de cada una, tales como: ***Spanish_Metaphone*** y ***Spanish_Soundex***.
- **Creador:** se emplea cuando se le aplica la responsabilidad a una clase determinada de crear una o más instancias de otra. Esto sucede en caso de que la clase creadora contenga, agregue, registre, utilice o posea datos de inicialización de objetos de alguna clase determinada(Craig, 2003). La utilización de este patrón es muy importante para las clases encargadas de registrar datos de usuariosen el sistema, como lo hace la clase ***Spanish_Soundex***que registra un código fonético, para cada nombre registrado en la base de datos.
- **Controlador:**asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión(Craig, 2003). En el algoritmo diseñado se delegan las responsabilidades principales a su clase controladora: ***Meta_Soundex_Algorithm***, la cual contiene el algoritmo de codificación fonética destinado al procesamiento de cadenas, conjuntamente con los grupos de similitud de grafemas del idioma español.

2.5.2 Patrones GOF

Los patrones *The Gang of Four* (GOF) describen soluciones simples y elegantes a problemas específicos en el diseño de *software* orientado a objetos (Gamma et al., 2009). En el desarrollo de la propuesta de solución se utilizó el siguiente patrón:

- **Singleton:**este patrón define a una clase de la que tan sólo puede haber una única instancia. Este patrón se utilizó cuandola clase principal que contiene el algoritmo de codificaciónfonética, la clase ***Meta_Soundex_Algorithm***, proporciona un punto de acceso global mediante la llamada a la función encargada de la acciónde codificación dada una cadena de entrada.

2.6 Conclusiones parciales

- Mediante el análisis documental y la entrevista se obtuvieron los requisitos a tener en cuenta para la implementación de la propuesta de solución, permitiendo desarrollar una descripción completa del comportamiento del algoritmo en el sistema GINA.
- La utilización de los patrones de diseño GRASP y GOFpermitieron tener una guía para la fase posterior, contribuyendo al uso de buenas prácticas en el proceso de modelado e implementación del software.
- Se generaronlosartefactos ingenieriles correspondientes con la etapa del modelado de negocio, requisitos, Análisis y diseño, propuestos en cada disciplina de la metodología de desarrollo utilizada, que permitieron en gran medida realizar el diseño de la propuesta de solución.

CAPÍTULO 3.IMPLEMENTACIÓN Y VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

3.1 Introducción

En el presente capítulo se realizarán las disciplinas de implementación y pruebas del algoritmo propuesto. En la implementación se plantea la modificación del algoritmo que se utiliza actualmente en GINA,donde se consideran las características propias de la lengua española.Además,se generael diagrama de despliegue que representa de forma general la ubicación del algoritmo propuesto en el sistema GINA. También se muestran los resultados de un conjunto de pruebas realizadas durante las diferentes fases para verificar el correcto funcionamiento del algoritmo. Entre las pruebas recogidas en este capítulo se encuentran las pruebas unitarias apoyadas de la herramienta*PHPUnit,* también se emplea la técnica de camino básico, y posteriormente, se evidencian un conjunto de pruebas estadísticas donde se evalúan varias métricas como la precisión y la *F_Measure* del algoritmo propuesto.

3.2 Implementación

En la etapa de implementación se crea una versión ejecutable del software. La implementación quizá requiera el desarrollo de programas en lenguajes de programación de alto o bajo niveles, o bien, la personalización y adaptación de sistemas comerciales genéricos para cubrir los requerimientos específicos de una organización(Sommerville, 2011).

El algoritmo propuesto, basa su funcionamiento en dos algoritmos: el algoritmo *Spanish_Soundex* y el *Spanish_Metaphone*, encaminados a cubrir las reglas de pronunciación del idioma español.

3.2.1 Algoritmo fonético *Meta_Soundex*

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.

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

1. Convertir todas las letras en mayúscula.
2. Normalizar usando *SpanishMetaphone* 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.

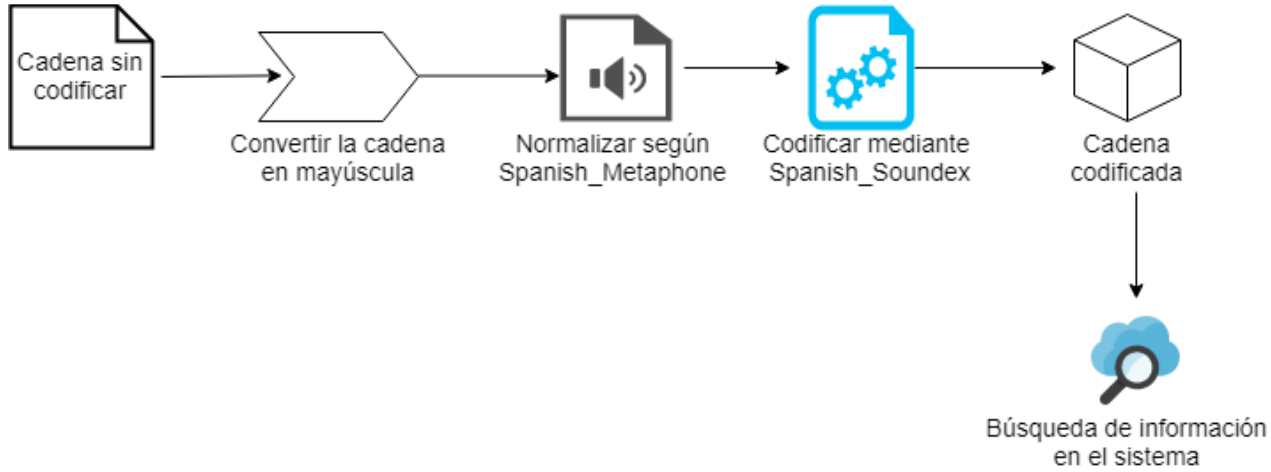

Figura 7. Secuencia de pasos del algoritmo propuesto

Partiendo del conocimiento de los pasos a seguir definidos en el algoritmo propuesto cabe destacar que es importante el transcurso por cada uno de los algoritmos seleccionados ya que el *Spanish_Metaphone*permite normalizar la cadena, y a eliminar aquellos caracteres que podrían considerarse como casos excepcionales en la codificación de palabras. Además de que se podrá obtener un código fonético para dicha palabra a través del algoritmo *Spanish_Soundex*, beneficiando la búsqueda de información en el sistema GINA, y contribuyendo a la toma de decisiones en la entidad aduanera.

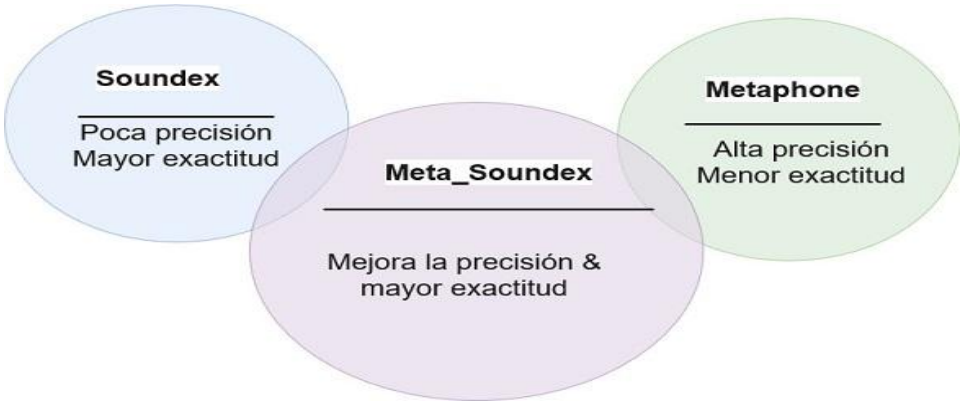

Figura 8. Diseño del algoritmo propuesto

Fuente. (Koneru, 2016)

3.2.2 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.

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 5. Transformaciones del algoritmo *Spanish Soundex*

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

Fuente.(Koneru, 2016)

En el algoritmo propuestoes notable la inclusión de todos los caracteres definidos en el idioma español en los grupos de codificación definidos. Los grupos se arman con base en los problemas ortográficos frecuentes en el idioma español y contemplando caracteres de este idioma como la

"ñ" y la "ll". La "y", fuente de múltiples errores ortográficos en español al confundirla con la "ll", no se descarta, sino que incluye en el mismo grupo.

3.2.3 Algoritmo fonético *Spanish Metaphone*

Este algoritmo fue desarrollado 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 las transformaciones realizadas al algoritmo:

Tabla 6. Transformaciones del algoritmo *Spanish Metaphone*

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

Fuente.(Koneru, 2016)

Tabla 7. Secuencia de pasos del algoritmo *Spanish Metaphone*

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.

Fuente.(Koneru, 2016)

3.2.4 Casos especiales en la codificación de las palabras

El mayor número de casos de debilitamiento consonántico que se encuentran en el mundo, radica en los países de América Latina. El español caribeño, que incluye las variedades habladas en Puerto Rico, la República Dominicana, Cuba, Panamá y casi toda Venezuela y las zonas costeras de Colombia, es generalmente considerado como el más "innovador" o "radical" de los dialectos geográficos del español de Latinoamérica (Hualde, 2014). Los fenómenos de debilitamiento consonántico en comparación con las variedades andaluzas y canarias, alcanzan una mayor intensidad en esta región que en otras partes de Latinoamérica.

A consecuencia de esto, se identifican varios casos de palabras que en su pronunciación son alterados sus caracteres, considerándolos casos especiales de codificación tales como:

- Repetición innecesaria de letras: comúnmente se hacen con la intensión de reforzar un significado que se sobreentiende (Edith Ordóñez García, 2015). Cuando ocurren esas repeticiones, se hacen las transformaciones pertinentes y luego se codifica partiendo de los principios establecidos por la Real Academia Española (RAE).
- Uso incorrecto de caracteres: un ejemplo de estos casos puede ser cuando se está en presencia de palabras como 'cafZ', que inmediatamente se infiere que el término correcto es 'café' (Fuentes et al., 2017). En situaciones como esta las palabras con caracteres incorrectos, son transformadas por términos definidos por la RAE.
- Presencia de palabras ambiguas: ocurre cuando una palabra utilizada pueda tener dos o más significados, o sea, tienen varias interpretaciones, dependiendo del contexto en el que sean utilizadas (Fuentes et al., 2017). En estos casos se pueden introducir varios errores en la precisión del algoritmo que se esté utilizando.
- Uso de abreviaturas:la necesidad de escribir con más rapidez y de encerrar en poco espacio la mayor cantidad de información, son las razones para abreviar ciertas palabras, representándolas con solo una o algunas de sus letras, de las que se deduce con facilidad el vocablo o vocablos aludidos (Edith Ordóñez García, 2015).En la mayoría de estos casos son inferidos los términos correctos en dependencia del contexto de la palabra, desde la primera aparición.
- Caracteres especiales: son considerados como caracteres especiales los siguientes: (@, #, -, =) ya que no son incluidos estos términos en el alfabeto definido por la RAE. Por lo que estos caracteres son ignorados en la codificación (Fuentes et al., 2017).
- *Hashtags*: son consideradas como palabras claves. Técnicamente, es una cadena de caracteres conformada por una o varias palabras concatenadas y precedidas por una

almohadilla o numeral (Fuentes et al., 2017). Al no ser definida por la RAE, estos términos son ignorados en la codificación.

Todos estos casos antes mencionados describen mecanismos de control para ser usados en orden de prevenir la presencia de errores considerados como sistemáticos, en la codificación de caracteres, al no cumplir con las reglas ortográficas independientemente del idioma empleado.

3.2.5 Diagrama de componentes

Un modelo de componentes define un conjunto de estándares para componentes, incluidos estándares de interfaz, estándares de uso y estándares de implementación. La implementación del modelo de componentes brinda un conjunto de servicios comunes que pueden usarse por parte de todos los componentes (Craig, 2003).

A continuación, se muestra el diagrama de componentes del algoritmo implementado en donde se evidencia la agrupación definida para el algoritmo propuesto, el cual cuenta con dos componentes que empaquetan el código de los algoritmos implementados para la normalización de cadenas y para la construcción del código fonético de los nombres de las personas almacenados en la base de datos.

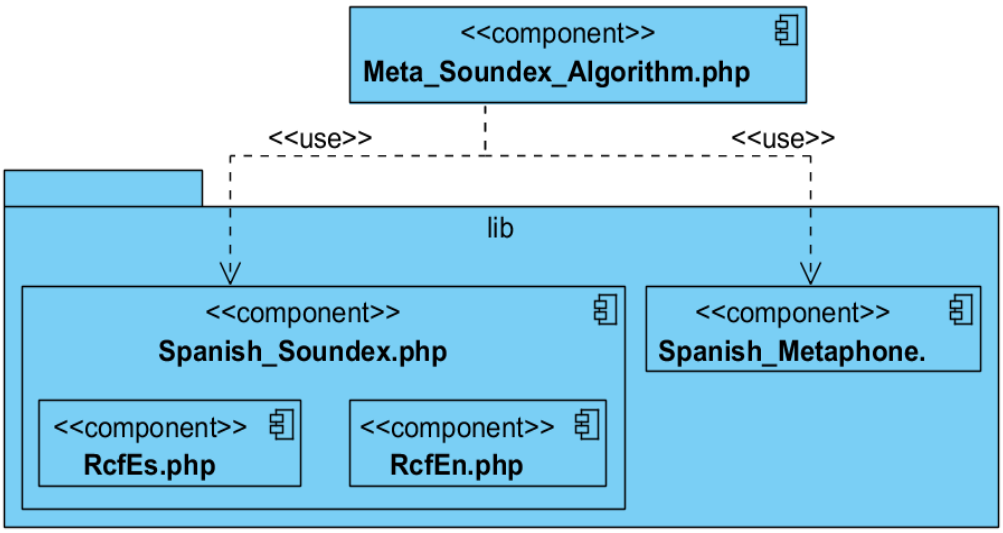

Figura 9. Diagrama de componentes del algoritmo propuesto

3.2.6Diagrama del modelo de despliegue

La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos. Esta vista permite determinar la secuencia de distribución y asignación de recursos. Para su representación se emplea el artefacto diagrama de despliegue.El mismomuestra a los procesadores las distribuciones de los procesos y los componentes(Pressman, 2010).

Para el desarrollo de la solución fueron tomadas en cuenta las especificaciones generales del modelo de despliegue utilizado para el sistema GINA, el cual esta implementado siguiendo la arquitectura Cliente_ Servidor. Tomando en cuenta que el cliente no depende de esta arquitectura para la realización de las operaciones o servicios que brinda el sistema, si se hace necesario que exista una total integración entre el motor fonético y las aplicaciones que lo utilizan, de modo que estas puedan acceder directa y globalmente a cada una de las funciones que este brinda.

A continuación, se muestra el diagrama de despliegue general del sistema GINA:

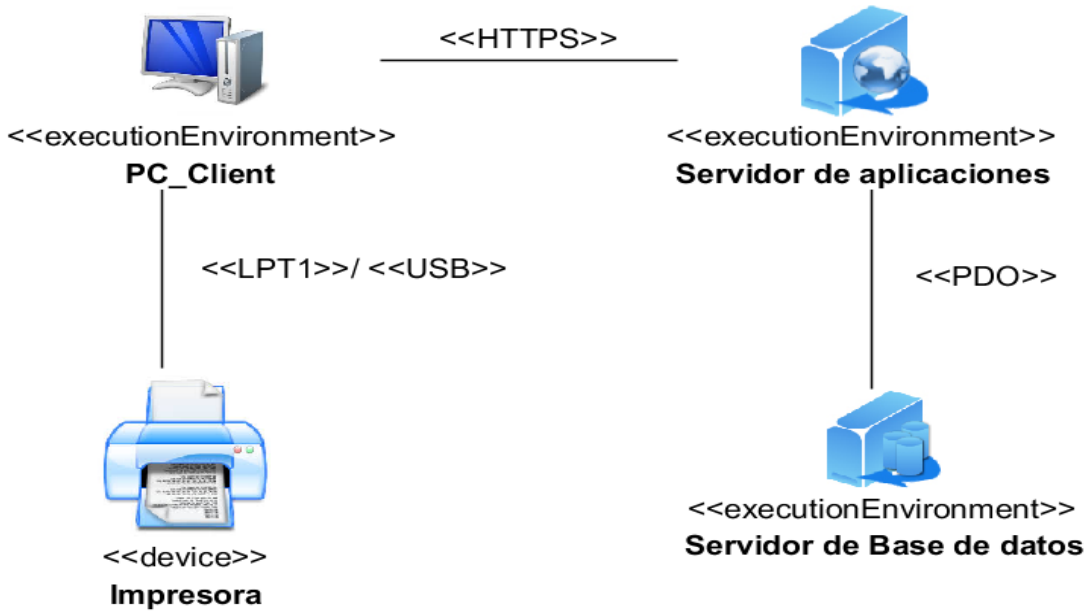

Figura 10. Diagrama de modelo de despliegue

Los servicios que brinda el sistema GINA están centralizados, lo cual significa que un único servidor responde a todas las operaciones realizadas por los analistas de la AGR. De igual forma, también se centralizan las funcionalidades generales del algoritmo fonético, logrando una mayor eficiencia y rendimiento a la hora de retornar los resultados de las búsquedas.

3.3Pruebas de software

Las pruebas de *software* se consideran parte de un proceso más amplio de verificación y validación (V&V) del *software*(Sommerville, 2011).Las pruebas que se realicen deben mostrar un conjunto de características con el objetivo de encontrar la mayor cantidad de errores con el mínimo esfuerzo(Pressman, 2010).

Al ejecutarse esta actividad se persigue descubrir defectos en el sistema, que hagan que este tenga un comportamiento incorrecto o no deseable, y para verificar que cumple con los requerimientos del cliente, con el fin de suplir sus necesidades. Con esto no se obtiene un sistema totalmente libre de errores, pero si apto para ser usado por el usuario final.

3.3.1 Niveles de pruebas

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 crearcomponentes compuestos. La prueba de componentes debe enfocarse en probarinterfaces 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 desarrolladorde *software* con el propósito de detectar errores de interfaces y relaciones entrecomponentes. Consiste fundamentalmente en validar las conexiones e integración entredos o más componentes de *software* haciendo uso de la técnica de caja blanca.
- **Prueba de sistema:** el objetivo principal de esta prueba es ejercitar completamente el sistema de cómputo abarcando diferentes tipos de prueba (Sommerville, 2011).En este nivel se comprueban los requisitos no funcionales, así como utilidades, unidades físicas y entornos operativos después que el *hardware* y el *software* han sido integrados.

En la ejecución de las actividades de prueba,serán aplicadas las pruebas unitarias para poder validar las funcionalidades con las que cuenta el algoritmo propuesto mediante el uso de*PHPUnit,*que es un entorno para realizar este tipo de prueba en el lenguaje de programación PHP.

3.3.2 Métodos de prueba

En pruebas de software se pueden comprobar cualquier producto de ingeniería de dos formas: conociendo la funcionalidad específica para la cual fue diseñado, demostrando mediante pruebas que cada funcionalidad es operativa o conociendo el funcionamiento del producto, donde cada componente interno trabaja de la forma adecuada. Al primer enfoque se le denomina prueba de caja negra y al segundo prueba de caja blanca (Pressman, 2010).

CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

Para la validación de la propuesta de solución será aplicado el método de caja blanca, específicamente la técnica de ruta básica, 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.

- **Pruebas de caja blanca:** Las pruebas de caja blanca o pruebas estructurales son un enfoque sistemático a laspruebas donde se usa el conocimiento del código fuente del programa para diseñar pruebas de defecto(Sommerville, 2011). El conjunto de pruebas debe garantizar que se ejecute toda ruta lógica a través del programa. Además de enfatizar la reducción de errores internos (Darlene et al., 2013).

Técnica de ruta básica: Esta técnica se basa en obtener una medida de la complejidad del diseño procedimental de un programa (o de la lógica del programa).Esta medida es la complejidad ciclomática de *McCabe*, y representa un límite superior para el número de casos de prueba que se deben realizar para asegurar que se ejecuta cada camino del programa(Juristo et al., 2006). Para diseñar pruebas utilizando el principio del camino básico, se deben seguir los siguientes pasos(Pressman, 2010):

1)Obtener el gráfico de flujo del diseño o módulo de código.
2)Obtener la complejidad ciclomática del gráfico de flujo.
3)Definir el conjunto básico de caminos independientes.
4)Determinar los casos de prueba que permiten la ejecución de cada uno de los componentes mencionados anteriormente.
5)Ejecutar cada caso de prueba y verificar que los resultados obtenidos son losesperados.

3.4 Resultados obtenidos del método de caja blanca. Técnica de ruta básica

Para la realización de las pruebas se aplicó el método de caja blanca a través de la técnica de ruta básica sobre las clases principales del algoritmo propuesto, con la ejecución de las mismas, se pudo derivar un conjunto de pruebas con el fin de descubrir errores en el software.

A continuación, se presenta el resultado de las pruebas realizadasa uno de los métodos principales del algoritmo diseñado:

Tabla 8. 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
{
publicfunctionMetaphone($string,$phonemes=0)

{

$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

Para obtener los casos de prueba a partir de la técnica seleccionada se debe construir el grafo de flujo correspondiente al código de la función como se muestra en la figura:

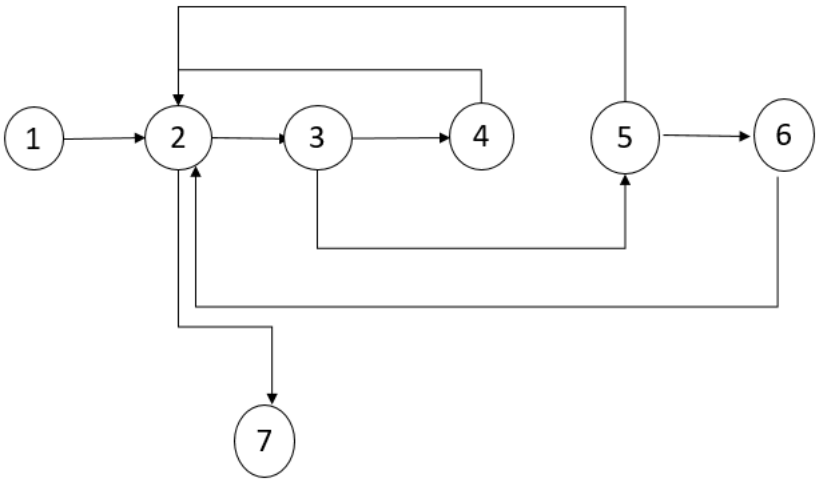

Figura 11. Grafo resultante de aplicar la técnica Camino Básico

Luego se determina la complejidad ciclomática V (G) del grafo resultante, la cual es un indicador del número de caminos independientes que existen en un grafo. Puede ser calculada de tres formas:

Cálculo de la Complejidad Ciclomática

1. V (G) = A – N + 2, siendo A la cantidad de aristas o arcos del grafo y N la cantidad de nodos del grafo.

2. V (G) = P+1, siendo P los nodos predicados, es decir los que tienen más de una arista de salida.

3. V (G) = R, siendo R el número de regiones cerradas del grafo.

Al realizar los cálculos correspondientes se obtiene por cualquiera de las variantes el siguiente resultado:

1. V (G) = A – N + 2 **2.** V (G) = P+1 **3.** V (G) = R

V (G) = 9– 7 + 2 V (G) = 3+ 1 V (G) = 4

V (G) = 4 V (G) = 4

El cálculo arrojó que V (G) = 4, definiendo como posibles caminos básicos:

Camino básico # 1: 1, 2, 7

Camino básico # 2: 1, 2, 3, 5, 2, 7

Camino básico # 3: 1, 2, 3, 4, 2, 7

Camino básico # 4: 1, 2, 3, 5, 6, 2, 7

Luego se definen los casos de prueba para cada uno de los caminos básicos obtenidos. A continuación, se muestra dicho resultado:

Tabla 9. Caso de prueba para el camino 1

Descripción	Codificar el nombre procesado a través del algoritmo Metaphone.
Condición de ejecución	Quela longitud de la cadena a normalizar no sobrepase la cantidad de fonemas establecidos para la cadena.
Entrada	Cadena a normalizar .
Resultado	Se obtiene una cadena normalizadaque cumple con las reglas de normalización definidas por el algoritmo.

Tabla 10. Caso de prueba para el camino 2

Descripción	Codificar el nombre procesado a través del algoritmo Metaphone.
Condición de ejecución	Quela longitud de la cadena a normalizar no sobrepase la cantidad de fonemas establecidos para la cadena .
Entrada	Cadena a normalizar .
Resultado	La cadena no cumple con las reglas establecidas para la normalización.

Tabla 11. Caso de prueba para el camino 3

Descripción	Codificar el nombre procesado a través del algoritmo Metaphone.
Condición de ejecución	Quela longitud de la cadena a normalizar no sobrepase la cantidad de fonemas establecidos para la cadena .
Entrada	Cadena a normalizar .
Resultado	Ningún resultado.

Tabla 12. Caso de prueba para el camino 4

Descripción	Codificar el nombre procesado a través del algoritmo Metaphone.
Condición de ejecución	Quela longitud de la cadena a normalizar no sobrepase la cantidad de fonemas establecidos para la cadena .
Entrada	Cadena a normalizar .
Resultado	Se obtiene una cadena normalizadaque cumple con las reglas de normalización definidas por el algoritmo.

3.4.1Resultados de las pruebas unitarias utilizando la herramienta PHPUnit

Para la realización de las pruebas al código se hicieronpruebas unitarias que permitieron ir comprobando la correcta implementación de las funcionalidades que brindael algoritmo. Para llevar a cabo estas pruebas se empleó el método de caja blanca, aunque también fueron realizadas por el equipo de desarrollo que aprovechó las ventajas que brinda el framework de pruebas *PHPUnit* en su versión 7.5.20.

A continuación, se muestranlos resultados de las pruebas unitarias:

El método**testNormalizacion**se le aplica a la clase controladora ***Spanish_Metaphone***con el objetivo de obtener de forma satisfactoria los nombres normalizados según este algoritmo. El resultado se muestra en la Figura 12de forma satisfactoria.

```
class TestAlgorithmsTest extends TestCase
{
publicfunction testNormalizacion()
{
$rcf=new RcfES ();
$rcf->setCadenaOriginal ("Selianne");
$this->assertRegExp ('/SLNN/', $rcf->getCadenaNormalizada ());
$rcf->setCadenaOriginal ("dayme");
$this->assertRegExp ('/DM/', $rcf->getCadenaNormalizada ());
$rcf->setCadenaOriginal ("rubiano");
$this->assertRegExp ('/RBN/', $rcf->getCadenaNormalizada ());
$rcf->setCadenaOriginal ("alexis");
$this->assertRegExp ('/ALXS/', $rcf->getCadenaNormalizada ());
$rcf->setCadenaOriginal ("OLGA");
$this->assertRegExp ('/OLG/', $rcf->getCadenaNormalizada ());
$rcf->setCadenaOriginal ("CAMILEIDIS");
$this->assertRegExp ('/KMLDS/', $rcf->getCadenaNormalizada ());
$rcf->setCadenaOriginal ("PSIQUIATRA");
$this->assertRegExp ('/jkm/', $rcf->getCadenaNormalizada ());
}//…}
```

Figura 12. Resultado de las pruebas unitarias utilizando PHPUnita la clase *Spanish_Metaphone*

El método **testCódigoFonético**se le aplica a la clase controladora***Spanish_Soundex*** con el objetivo de formar los códigos fonéticos de cada nombre de manerasatisfactoria. El resultado se muestra en la Figura 13.

```
publicfunction testCódigoFonético ()
{
$rcfes=new RcfES ();
$codificador=new CodificadorFoneticoGina ();
$rcfes->setCadenaOriginal ("alexis, norma, dayme, rubiano");
echo"cadena convertida “. $rcfes->getCadenaNormalizada()."<br>";
echo"codificador “. $codificador->convertToCodigoFonetico('alexis','ES')."<br>";
$this->assertEquals ('6’, strlen($codificador->convertToCodigoFonetico('alexis','ES')),");
$this->assertEquals ('544000’, $codificador->convertToCodigoFonetico('alexis','ES'),");
$this->assertEquals ('696000’, $codificador->convertToCodigoFonetico('norma','ES'),");
$this->assertEquals ('360000’, $codificador->convertToCodigoFonetico('dayme','ES'),");
$this->assertEquals('916000',$codificador->convertToCodigoFonetico('rubiano','ES'),");
}
```

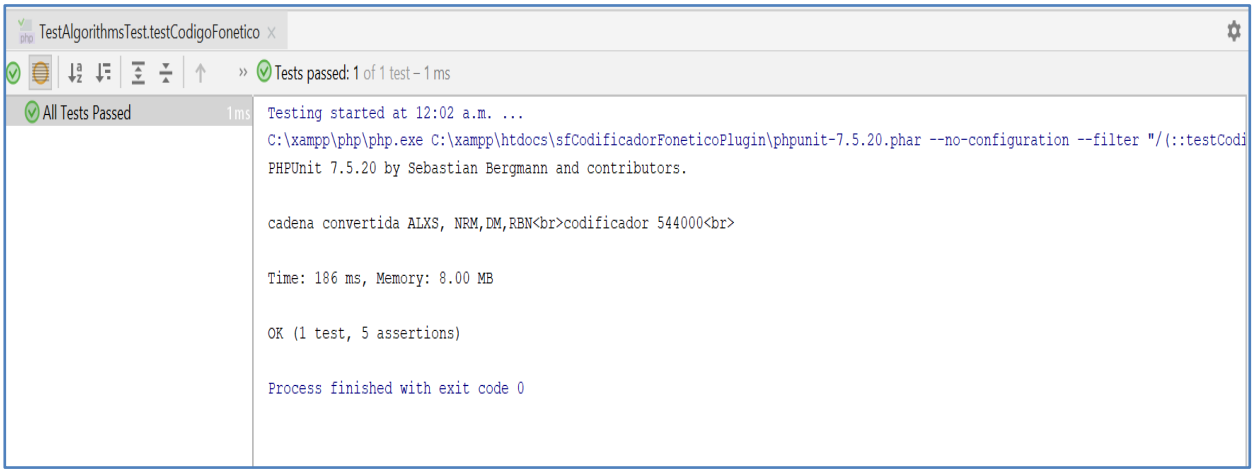

Figura 13. Resultado de las pruebas unitarias utilizando PHPUnit a la clase*Spanish_Soundex*

El método**testMetaSoundex** se le aplica a la clase principal***Meta_Soundex_Algorithm*** con el objetivo cargarcada nombre guardado en la base de datos, con su código fonético asociado de formasatisfactoria. El resultado se muestra en la Figura 14.

```
publicfunctiontestMetaSoundex()
{
$rcfes=newRcfES();
$codificadorMetaphone=newPhoneticAlgorithmsES();
$codificadorSoundex=newCodificadorFoneticoGina();
```

```
echo$codificadorSoundex->convertToCodigoFonetico(
            $codificadorMetaphone->Metaphone('olga,rubiano,guerra',6),'ES');

$this->assertEquals(916000,$codificadorSoundex->convertToCodigoFonetico(
$codificadorMetaphone->Metaphone('rubiano',6),'ES'));

$this->assertEquals(890000,$codificadorSoundex->convertToCodigoFonetico(
$codificadorMetaphone->Metaphone('guerra',6),'ES'));
}
```

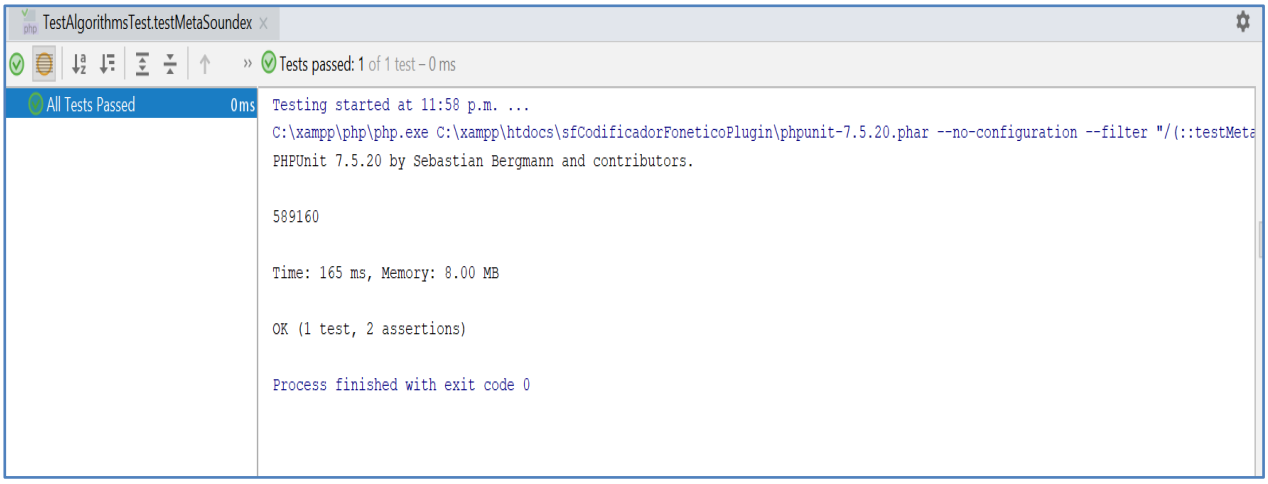

Figura 14. Resultado de las pruebas unitarias utilizando PHPUnit a la clase *Meta_Soundex_Algorithm*

3.5Pruebas estadísticas

Las pruebas estadísticas se emplean con la finalidad de establecer la probabilidad de que una conclusión obtenida a partir de una muestra sea aplicable a la población de la cual se obtuvo (Flores-ruiz et al., 2017).

Para realizar un análisis estadístico es necesario evaluar 3 aspectos importantes, para poder elegir la forma de seleccionar la prueba estadística acorde con los objetivos del estudio: diseño de investigación, número de mediciones y escala de medición de variables. Partiendo de esta secuencia se define un estudio comparativo ya que se estará evaluando los resultados en cuatro grupos de datos, sometidos a 2 mediciones realizadas en diferentes momentos. El tercer aspecto trascendente cuando se planea este tipo de estudio es la escala de medición de las variables, siendo necesario definir la naturaleza de cada uno de los datos o las mediciones que se realizan durante el desarrollo de la investigación; considerándose el estudio de variables cuantitativas continuas.

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.

3.5.1 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:

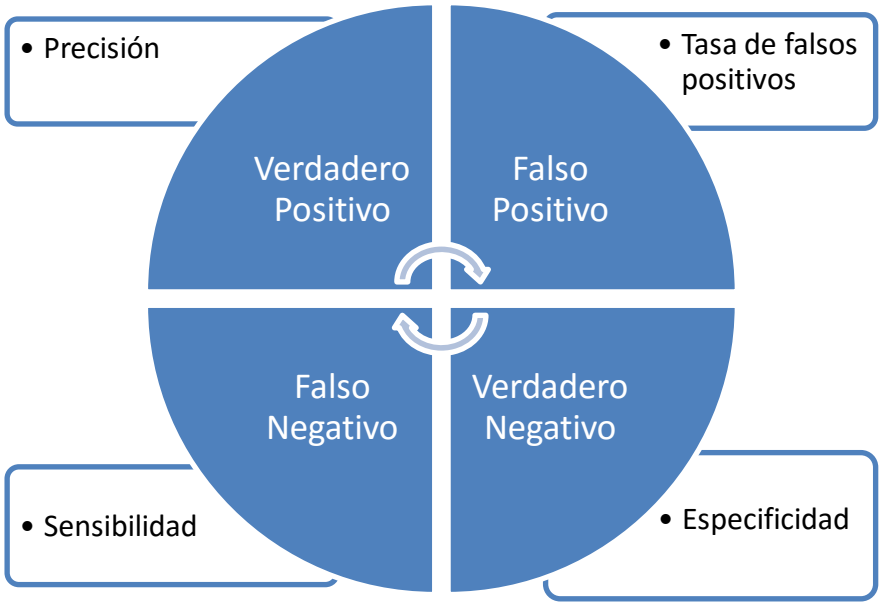

Figura 15. Matriz de confusión de registros de clasificación de datos

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 (Figura15), 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:

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

- 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:

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

- 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:

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

- 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:

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

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:

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

3.5.2 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[2], 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 $\boldsymbol{H_0}$ es la definición de la hipótesis nula y $\boldsymbol{H_a}$ es la hipótesis alternativa, y $\boldsymbol{\mu}$ es el número de aciertos por grupos de algoritmos comparados, y $\boldsymbol{k}$ es el número de grupos evaluados, que en este caso son 2.

3.5.3 Resultados de las pruebas estadísticas

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 personasy 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.

[2]Método estadístico para el análisis de varianza de un factor considerados como valores de evaluación de diferentes variables nominales.

Tabla 13. Resultado de la prueba estadística para el dataset[3]conformadopor 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 algoritmo utilizado en el sistemay 294 FP. Utilizando el algoritmo *Meta_Soundex_Algorithm*, se encontraron 1.471 personas y32 clasificadas como FP.

Tabla 14. Resultado de la prueba estadística para el dataset conformado por 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 3.475TP 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 15. Resultado de la prueba estadística para el dataset conformado por 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

[3] Conjunto de datos.

como FP. Además, se utilizó el algoritmo *Meta_Soundex_Algorithm*, donde se clasificaron 33.771 y528 como FP.

Tabla 16. Resultado de la prueba estadística para el dataset conformado por 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

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.

3.6 Valoración de los resultados estadísticos

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 la hipótesis nula 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 el que presentaun valor más alto con respecto al algoritmo de GINA. Esto indica que su factor de predicción de valores verdaderos positivos en comparación con la tasa de verdaderos positivos es alto.

3.7 Prueba de integración

Existe un enfoque para la integración llamado prueba basada en el uso. Esta prueba comienza la construcción del sistema integrando y probando aquellas clases (llamadas clases independientes) que usan muy pocas de las clases. Después de probar las clases independientes, se comprueba la próxima capa de clases, llamadas clases dependientes, que

usan las clases independientes. Esta secuencia de capas de pruebas de clases dependientes continúa hasta construir el sistema por completo(Juristo et al., 2006).

Durante la implementación de las funcionalidades se realizaron las pruebas de integración donde para evaluar la solución desarrollada se concibieronvarias iteraciones para probar el algoritmo de forma íntegra. Estas pruebas validaron la integración del algoritmo en el sistema GINA, resolviendo las inconformidades presentadas. Las pruebas corroboraron que las funcionalidades establecidas y demás elementos del sistema operan satisfactoriamente, ajustándose a los requisitos especificados.

3.8 Conclusiones parciales

- 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 ruta 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.

CONCLUSIONES GENERALES

Al término de esta investigación se pudo arribar a las siguientes conclusiones:

- El estudio de los principales elementos teóricos de los algoritmos fonéticos permitió identificar las características fundamentales para la implementación de la propuesta de solución, contribuyendo a su correcta adecuación con el idioma español ylas reglas de la fonética del idioma. Además de que permitió que el algoritmo brindara los resultados correctos, garantizando la unicidad e integridad de la información con la que se trabaja en esa institución.
- El uso de una metodología ágil, tecnologías y herramientas libres para la implementaciónde la solución, permitió asegurar la independencia tecnológica del algoritmo desarrollado, evitando la elaboración de documentación innecesaria y disminuyendo considerablemente eltiempo de desarrollo de la solución.
- 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.

RECOMENDACIONES

Teniendo en cuenta los resultados alcanzados con esta investigación se proponen las siguientes recomendaciones:

- Implementar otros algoritmos paragenerar códigos fonéticos que puedan ser combinados con la propuesta actual, basados en las pronunciaciones, el uso de abreviaturas, caracteres extraños y casos especiales en la codificación de cadenas.
- Incluir las reglas de normalización para otros idiomas además del español e inglés.
- Implementar algoritmos de similitud de cadenas para determinar umbrales que mejor separen las cadenas relevantes de las irrelevantes a partir de la cadena codificada.

REFERENCIAS BIBLIOGRÁFICAS

Aduana General de la República de Cuba. (2019). Aduana General de la República de Cuba. In *¿Quienes somos?* http://www.aduana.gob.cu/quienes-somos.html

Alejandro, M., & Barragán, R. (2009). *Similitud fonética entre palabras para mejorar la Recuperación de Información en Documentos Orales*. Instituto Nacional de Astrofísica, Optica y Electrónica.

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.

Beider, A., & Morse, S. P. (2010). Phonetic Matching: A Better Soundex. *Association of Professional Genealogists Quarterly*, *March*.

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.

Bustamante, F. R., & Díaz, E. L. (2006). Spelling error patterns in Spanish for word processing applications. *Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC 2006*, 93–98.

Bustos, A. (2007). *Blog de Lengua*. *2529–8461*. https://blog.lengua-e.com/contacto/

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.

Carmen Gálvez, P. (2016). *Personal name identification through Phonetic Codification Systems*. http://encontros-bibli-blog.blogspot.com/

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

Collado, M. E. G., Orozco, L. C., & Linares, D. G. (2016). *El impacto de las tecnologías de la información y la comunicación en estudiantes de ciencias sociales: un estudio comparativo de dos universidades públicas*. *16*. http://www.scielo.org.mx/scielo.php?script=sci%7B_%7Darttext%7B&%7Dpid=S1665-26732016000200061

Craig, L. (2003). *UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos*.

Darlene, G., Dalila, J., & Delgado Martha. (2013). Unit Tests of Software in a University Environment. *Computación Y Sistemas*, *17*(1), 69–77. https://doi.org/10.13053/cys-17-1-1483

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

Di-Lella, J. L. P. (2019, July 12). TIC para el desarrollo y no como instrumento para la subversión o la desestabilización. *CubaDebate*. http://www.cubadebate.cu/opinion/2019/07/12/cuba-defiende-uso-de-las-tic-en-pos-de-desarrollo-economico-y-social/

Digital, R. (2018, November 22). Nuevos retos para el turismo cubano en 2019. *Granma*. internet@granma.cu

Edith Ordóñez García. (2015). *Uso correcto de los elementos de la oración.Vicios del lenguaje.*

Eguiluz, J. (2012). *Desarrollo web ágil con Symfony*.

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

Flores-ruiz, E., Miranda-novales, M. G., & Villasís-keever, M. Á. (2017). *The research protocol VI : How to choose the appropriate statistical test . Inferential statistics El protocolo de investigación VI : cómo elegir la prueba estadística adecuada . Estadística inferencial*. *64*(3), 364–370.

Fuentes, A. A. G., Parra, I. P., Quevedo-Torrero, J. U., & Perez, R. D. (2017). Comparative Analysis of Phonetic Algorithms Applied to Spanish. *Proceedings - 2016 International Conference on Computational Science and Computational Intelligence, CSCI 2016*, 1180–1185. https://doi.org/10.1109/CSCI.2016.0223

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

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (2009). *Elements of Reusable Object-Oriented Software* (37th ed.).

García, T. T. (2018). Cara a cara con GINA. *Universidad de Las Ciencias Informáticas*. https://www.uci.cu/universidad/noticias/cara-cara-con-gina

Gonzáles-Cam, C. (2008). Algoritmos fonéticos en el desarrollo de un sistema de información de marcas y signos distintivos. *Biblios: Revista Electrónica de Bibliotecología, Archivología Y Museología*, *32*, 8.

Hualde, J. I. (2014). 8. Los sonidos del español. In *Manual de lingüística española*. https://doi.org/10.1515/9783110362084-009

INDECOPI. (2020). https://www.indecopi.gob.pe/indecopi

Jácome, S. J., Ordóñez, A., Cerón, G. M., & Villaquirán, A. F. (2018). *Mapeo sistemático del uso de las tecnologías de la información y la comunicación en la diabetes tipo 2*. scielo.sld.cu/scielo.php?script=sci%7B_%7Darttext%7B&%7Dpid=S2307-21132018000400005

Jetbrains. (2015). https://www.jetbrains.com/phpstorm.

Jurafsky, D., & Martin, J. H. (2019). N-gram Language Models. In *Speech and Language*

Processing (p. 28).

Juristo, N., Moreno, A. M., & Vegas, S. (2006). *Técnicas de Evaluación de Software*. 131. http://www.grise.upm.es/sites/extras/12/pdf/Documentacion_Evaluacion_7.pdf

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

Manriquez, F. (2009). *¿Qué significa la "i" y la "g" en las versiones de Oracle Database?* 23-8. https://dbaproup.cl/que-significa-la-i-y-la-g-en-las-versiones-de-oracle-database/

Manuel, C., & González, R. (2010). *Implementación de un Motor de Codificación y Búsqueda Fonética para el Sistema de Gestión Integral de la Aduana*. Universidad de las Ciencias Informáticas.

Milanés, L. (2018). *Informatización de la sociedad cubana en cifras*. cubahora.cu

Ministerio de Comunicaciones. (2017). *Política Integral para el perfeccionamiento de la informatización de la sociedad en Cuba*.

Moreno, F., & Echeverri, J. (2012). *Algoritmo Fonético Para Detección*. *11*(20), 127–138.

Oracle.com. (2019). *Oracle*. https://www.oracle.com/technetwork/es/database/enterprise-edition/learnmore/index.html

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

PhpStorm. (2019). https://www.jetbrains.com/es-es/phpstorm/

Potencier, F., & Zaninotto, F. (2008). *Symfony:la guía definitiva*.

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

Ríos, A. (1996). Un alfabeto fonético del español para usos informáticos. *Asociación de Lingüística Y Filología de La América Latina*, *1139–8736*.

Rodríguez, T. (2014). *Metodología de desarrollo para la Actividad productiva de la UCI.* 1–16.

Rodríguez, Y., Becerra, S., & Piña, J. L. (2018). Un acercamiento al sistema de Gestión de Ingreso para la Educación Superior de Cuba. *"V Taller Internacional Las TIC En La Gestión de Las Organizaciones*.

Romaní, J. C. C. (2009). *The Information Technologies Concept, Benchmarking of ICT Definitions in the Knowledge Society* (No. 27). https://doi.org/1137-1102

Ruiz, J. L. E., & Angeles, A. Z. (2017). *Metodología de desarrollo de software*.

Rumbaugh, J., Jacobson, I., & Booch, G. (2007). *El Lenguaje Unificado de Modelado*.

Sánchez, F. B. (2015). *Técnicas para el levantamiento de requerimientos en el desarrollo de un*

sistema de información. *114*, 200–217.

Software, F. A. (2019). *APACHE*. https://www.apache.org/

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

Symfony. (2019). https://symfony.com

Toledano López, O. G. (2019). *Revisión sobre el impacto de las bases de datos NoSQL en el contexto de los Datos Educacionales Masivos*.

Torres, Y. O. U. (2016). *Implementación de un sistema web para el control de inventario de medios computacionales .* (Issue 58). Universidad Central “Marta Abreu” de Las Villas.

Villegas, N. M. (2008). *Importancia de la arquitectura de software en las organizaciones*.

Visual Paradigm. (2019). https://www.visual-paradigm.com/

ANEXOS

Anexo 1 Entrevista para la obtención de los requisitos

Preguntas realizadas con el fin de obtener los requisitos necesarios para la implementación del algoritmo:

1. Al realizar ejecuciones de ambos algoritmos con la misma entrada de datos. ¿Existe pérdidas de información en el conjunto de resultados positivos obtenidos?
2. Dentro de los resultados obtenidos de aplicar el algoritmo, ¿existen valores que difieren de la respuesta esperada?¿Cuánto influye esto en la calidad de la información a brindar?
3. ¿Qué comportamiento tienen los recursos del servidor al ejecutar el algoritmo? ¿Influyen en ellos el número de datos a procesar?
4. ¿Existen parámetros de confianza y certeza que permitan evaluar el nivel de incertidumbre en los resultados obtenidos? ¿Qué tan eficientes pueden llegar a ser estos?

Anexo 2 Conjunto de datos obtenidos al realizar búsquedas en el sistema para 25 nombres

#	Nombres a buscar	Encontrados x GINA	Enconntrados x Meta_Soundex	Diferencias
1	ANNA	116	104	"[""EMINE"",""EAMON"",""HAEMOON", ""AMINE"",""AMANA"", ""HAEMIN"",""HAOMIN"",""AIMIN""]"
2	GUY	92	92	[]
3	JULIA	41	41	[]
4	SEBASTIEN	14	14	[]
5	MARIE	237	237	[]
6	MAXIME	11	1	"[""MEIJUN"",""NAIJIAN"",""MEGHAN", ""MAGNA"","MEGAN"",""NAIGEN"", ""MEIJUAN"",""NAIJUN"",""MEIJIAN""]"
7	JACQUES	6	6	[]
8	MARINA	46	46	[]
9	HUGUES	12	3	"[""IAW"",""HAIWU"",""UWE"", ""HOW"",""AWA""]"
10	MILAGROS	1	1	[]
11	BELKYS	1	1	[]
12	MARIE	237	237	[]
13	RICHARD	18	13	"[""RICARDO"",""RICARDA""]"
14	MINAS	88	88	[]
15	NORA	237	237	[]
16	ALESSANDRA	4	4	[]
17	ALESSANDRO	4	4	[]
18	RODOLFO	8	8	[]
19	DEBBIE	2	2	[]
20	JERRY	0	0	[]
21	CARMEN	14	7	"[""CORINNE"",""CORINNA""]"
22	LUCIANO	8	8	[]
23	HANS	128	128	[]

24	YARDEN	0	0	[]
25	ODED	0	0	[]

Anexo 3 Conjunto de datos obtenidos al realizar búsquedas en el sistema para 50 nombres

#	Nombres a buscar	Encontrados x GINA	Enconntrados x Meta_Soundex	Diferencias
1	LAN	43	43	[]
2	STEFONE	33	33	[]
3	RODLIN	0	0	[]
4	ADRIAN	16	16	[]
5	BENJAMIN	11	11	[]
6	ERLINDA	5	5	[]
7	YASMANY	3	3	[]
8	GERMAN	5	2	["GRAINNE"]
9	BENJAMIN	11	11	[]
10	MATTHIAS	16	16	[]
11	JANY	276	276	[]
12	GENE	276	276	[]
13	GEORGES	38	38	[]
14	SARA	55	55	[]
15	MICHAEL	137	106	["MICAELA","NICOLE","NICOLA", "NICOLAS","MIKAEL","MAIKEL", "NICLAS","NIKOLAS"]
16	LARRY	1	1	[]
17	CHRISTOPHER	17	17	[]
18	MARITZA	1	1	[]
19	RAIMUNDO	8	8	[]
20	MERCEDES	4	4	[]
21	YANIRYS	5	5	[]
22	BRIANA	35	35	[]
23	JAVIER	4	4	[]
24	JAVIER	4	4	[]
25	WILLIAM	27	24	["GUILLAUME","WILHELM"]
26	CARMEN	14	7	["CORINNE","CORINNA"]
27	WILFREDO	4	4	[]
28	LOURDES	1	1	[]
29	JORGE	38	38	[]
30	ALEJANDRO	41	4	["ALEXANDER","ALEXANDRA", "ALEXANDRE","ALEXANDRIA"]
31	VANESSA	9	9	[]
32	ANTONIA	31	31	[]
33	BRIANNA	2	1	["VERNON"]
34	LYUBER	0	0	[]
35	ZOILA	23	23	[]
36	JOSUE	31	31	[]
37	JAIRELY	2	2	[]
38	MELISSA	5	5	[]
39	VLADIMIR	2	2	[]

40	YORDANKA	0	0	[]
41	LAZARO	1	1	[]
42	VIVIAN	11	11	[]
43	MAYKOL	137	31	["MICHAEL","MICHEL","MICHEAL", "MICHELE","NICHOLAS","MICHAL" ,"MICHAELA"]
44	MAYKEL	137	31	["MICHAEL","MICHEL","MICHEAL", "MICHELE","NICHOLAS","MICHAL", "MICHAELA"]
45	ORLANDO	5	5	[]
46	ELIER	5	5	[]
47	TOMAS	99	99	[]
48	SARA	55	55	[]
49	ILEANA	81	81	[]
50	RICHARD	18	13	["RICARDO","RICARDA"]

Anexo 4Resultado de la prueba de ruta básica en el método Spanish_Soundex

```
privatefunction espRegla2LetraY()
    {
        $i=0*1;                                                    (1)
        $tmp=$this->cadena;
        while($i!=strlen($tmp))                                    (2)
        {
            if((strpbrk(substr($tmp,$i,1),'Y')!=FALSE)&&($i<=strlen($tmp)))   (3)
            {
                if(strpbrk(substr($tmp,$i+1,1),'AEIOU')!=FALSE)    (4)
                {
                    $tmp=$this->caden=substr($tmp,0,$i)."LL".substr($tmp,$i+1);   (5)
                    $i=$i+2;
                }
                else$tmp=$this->cadena =substr($tmp,0,$i).substr($tmp,$i+1);   (6)
            }
            else$i++;                                              (7)
        }
        unset($tmp);                                               (8)
    }
```

Para obtener los casos de prueba a partir de la técnica seleccionada se debe construir el grafo de flujo correspondiente al código de la función como se muestra en la figura:

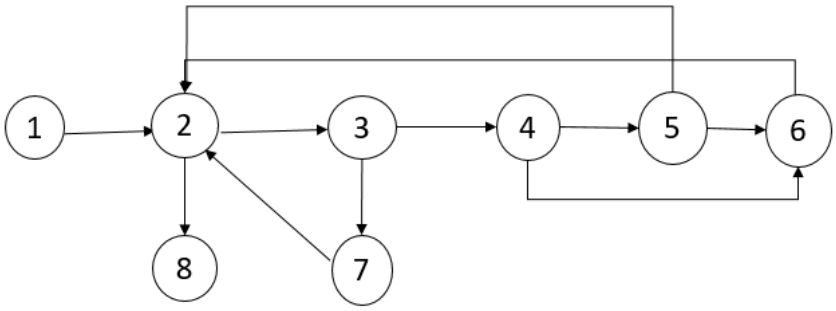

Figura 16. Grafo resultante de aplicar la técnica Camino Básico

Luego se determina la complejidad ciclomática V (G) del grafo resultante, la cual es un indicador del número de caminos independientes que existen en un grafo. Puede ser calculada de tres formas:

Cálculo de la Complejidad Ciclomática

1. V (G) = A – N + 2, siendo A la cantidad de aristas o arcos del grafo y N la cantidad de nodos del grafo.

2. V (G) = P+1, siendo P los nodos predicados, es decir los que tienen más de una arista de salida.

3. V (G) = R, siendo R el número de regiones cerradas del grafo.

Al realizar los cálculos correspondientes se obtiene por cualquiera de las variantes el siguiente resultado:

1. V (G) = A – N + 2 **2.** V (G) = P+1 **3.** V (G) = R

V (G) = 11 – 8 + 2 V (G) = 4+ 1 V (G) = 5

V (G) = 5 V (G) = 5

El cálculo arrojó que V (G) = 4, definiendo como posibles caminos básicos:

Camino básico # 1: 1,2,8

Camino básico # 2: 1, 2, 3, 7, 2, 8

Camino básico # 3: 1, 2, 3, 4, 5, 2, 8

Camino básico # 4: 1, 2, 3, 4, 5, 6, 2, 8

Camino básico # 5: 1, 2, 3, 4, 6, 2, 8

Luego se definen los casos de prueba para cada uno de los caminos básicos obtenidos. A continuación, se muestra dicho resultado:

Tabla 17. Caso de prueba para el camino 1

Descripción	Codificar el nombre procesado a través del algoritmo Spanish_Soundex, y la regla definida para la letra Y.
Condición de ejecución	Quela longitud de la cadena a normalizar no sobrepase la cantidad de fonemas establecidos para la cadena .
Entrada	Cadena a normalizar .
Resultado	No se obtiene una cadena normalizada.

Tabla 18. Caso de prueba para el camino 2

Descripción	Codificar el nombre procesado a través del algoritmo Spanish_Soundex, y la regla definida para la letra Y.
Condición de ejecución	Quela longitud de la cadena a normalizar no sobrepase la cantidad de fonemas establecidos para la cadena .
Entrada	Cadena a normalizar .
Resultado	Ningún resultado.

Tabla 19. Caso de prueba para el camino 3

Descripción	Codificar el nombre procesado a través del algoritmo Spanish_Soundex, y la regla definida para la letra Y.
Condición de ejecución	Quela longitud de la cadena a normalizar no sobrepase la cantidad de fonemas establecidos para la cadena .
Entrada	Cadena a normalizar .
Resultado	No se obtiene una cadena normalizada.

Tabla 20. Caso de prueba para el camino 4

Descripción	Codificar el nombre procesado a través del algoritmo Spanish_Soundex, y la regla definida para la letra Y.
Condición de ejecución	Quela longitud de la cadena a normalizar no sobrepase la cantidad de fonemas establecidos para la cadena .
Entrada	Cadena a normalizar .
Resultado	Se obtiene una cadena normalizadaque cumple con las reglas de normalización definidas por el algoritmo.

Tabla 21. Caso de prueba para el camino 5

Descripción	Codificar el nombre procesado a través del algoritmo Spanish_Soundex, y la regla definida para la letra Y.
Condición de ejecución	Quela longitud de la cadena a normalizar no sobrepase la cantidad de fonemas establecidos para la cadena .
Entrada	Cadena a normalizar .
Resultado	Se obtiene una cadena normalizada.

Anexo 5Resultado de una prueba al algoritmo Meta_Soundex.

```
Paso 0 Cadena Original-> olga
Paso 1: Cadena convertida con el SpanishMetaphone-> OLG
Paso 2 Cadena convertida Soundex-> OLG
Paso 3 Cadena Codificada> 58000
.................
Paso 0 Cadena Original-> alexi
Paso 1: Cadena convertida con el SpanishMetaphone-> ALX
Paso 2 Cadena convertida Soundex-> ALX
Paso 3 Cadena Codificada> 54000
.................
```