

Advances in Social Informatics and its Applications

Research in Computing Science

Series Editorial Board

Editors-in-Chief:

Grigori Sidorov (Mexico)
Gerhard Ritter (USA)
Jean Serra (France)
Ulises Cortés (Spain)

Associate Editors:

Jesús Angulo (France)
Jihad El-Sana (Israel)
Alexander Gelbukh (Mexico)
Ioannis Kakadiaris (USA)
Petros Maragos (Greece)
Julian Padget (UK)
Mateo Valero (Spain)

Editorial Coordination:

Alejandra Ramos Porras

RESEARCH IN COMPUTING SCIENCE, Año 17, Volumen 136, Febrero de 2017, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. <https://www.rcs.cic.ipn.mx>. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2005-121611550100-102. ISSN: en trámite, otorgado por el Instituto Nacional del Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 06 de Febrero de 2017.

RESEARCH IN COMPUTING SCIENCE, Year 17, Volumen 136, February, 2017, is a monthly publication edited by the National Polytechnic Institute through the Center for Computing Research. Av. Juan de Dios Bátiz S/N, Esq. Miguel Othón de Mendizábal, Nueva Industrial Vallejo, C.P. 07738, Mexico City, Tel. 57 29 60 00, ext. 56571. <https://www.rcs.cic.ipn.mx>. Editor in charge: Dr. Grigori Sidorov. Reservation of Exclusive Use Rights of Title No. 04-2005- 121611550100-102. ISSN: pending, granted by the National Copyright Institute. Responsible for the latest update of this issue: the Computer Research Center, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Last modified on February 06, 2017.

Advances in Social Informatics and its Applications

**Francisco Alejandro González Horta
Luis David Huerta Hernández
Juan Gabriel Ruiz Ruiz (eds.)**



Instituto Politécnico Nacional
“La Técnica al Servicio de la Patria”



Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2017

ISSN: 1870-4069

Copyright © Instituto Politécnico Nacional 2017

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional “Adolfo López Mateos”, Zacatenco
07738, México D.F., México

<http://www.rcs.cic.ipn.mx>

<http://www.ipn.mx>

<http://www.cic.ipn.mx>

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

All rights reserved. No part of this publication may be reproduced, stored on a retrieval system or transmitted, in any form or by any means, including electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the Instituto Politécnico Nacional, except for personal or classroom use provided that copies bear the full citation notice provided on the first page of each paper.

Indexed in LATINDEX, DBLP and Periodica

Printing: 500

Printed in Mexico

Editorial

This volume of the journal *Research in Computing Science* contains selected papers on Social Informatics and its Applications. Social informatics is a young movement that emerges by the need to understand and solve social issues through information and communications technologies. Since the 1970s we have experienced a transition from an industrial society to an informational society on a global scale. The information society we live today has transformed the way we live, we work, we play, we learn. The central resource of this Information Society is knowledge. This volume provides valuable knowledge on informatics and its applications to social welfare. The contributions in this volume are represented by eight papers organized in the following areas:

- Biomedical Informatics,
- Imaging Informatics,
- Social Informatics, and
- Software Development Informatics.

The papers in this volume have been carefully chosen by the Scientific Committee of the Second Conference on Informatics and Innovative Technologies (CIIT-2017), organized by the Universidad del Istmo (UNISTMO), Campus Ixtepec, Oaxaca, Mexico. Every paper was evaluated by three reviewers in a double-blind mode and with the common objective of selecting the papers with highest quality. The eight papers out of a total of fifteen received papers were selected and their extended versions were invited for this issue. Submission, reviewing, and selection process was supported free of charge.

This volume is the result of hard work and collaboration of many people. First of all, we thank the authors of the papers included in this volume for their technical excellence, which made possible the high quality of this volume. We also thank the members of the Scientific Committee, the members of the Organizing Committee, and the Editorial Board of the volume for their outstanding work and advice. Finally, we would also like to thank to the National Polytechnic Institute (IPN) and its Center for Computing Research (CIC) for the given support in order to achieve the success of this volume, which we believe, is very important for the academic and scientific life in this country.

October, 2017

Francisco Alejandro González Horta
Luis David Huerta Hernández
Juan Gabriel Ruiz Ruiz
UNISTMO–Campus Ixtepec, Mexico
Guest Editors

Table of Contents

	Page
Diseño de métricas basadas en estresores laborales para desarrolladores de software	9
<i>Ma. Alejandra Zamudio, Ramón R. Palacio, Luis A. Castro</i>	
STUTTY: Aplicación móvil para mermer la tartamudez utilizando la técnica del “tartamudeo-sencillo”	21
<i>Ricardo Solano Monje, Nayeli Joaquinita Meléndez Acosta, Homero Vladimir Ríos Figueroa</i>	
Clasificación de género utilizando medidas antropométricas, análisis de texturas y la arquitectura ANFIS.....	33
<i>Cosijopii García García, Nayeli Joaquinita Meléndez Acosta</i>	
Selection of Best Software Engineering Practices: A Multi-Criteria Decision Making Approach	47
<i>Gil Hernández-Ledesma, Erik G. Ramos, Carlos A. Fernández-y-Fernández, Jorge R. Aguilar-Cisneros, Juan J. Rosas-Sumano, Luis A. Morales-Ignacio</i>	
Behavior Modification Applied to Requirements Engineering for Medical Applications	61
<i>Gil Hernández-Ledesma, Noé Hernández-Ledesma, Erik G. Ramos, Carlos A. Fernández-y-Fernández</i>	
Métricas estructurales para evaluar la similitud de texturas naturales.....	73
<i>Alma Alheli Pedro Pérez, Raúl Pinto Elias, Jasiel Hassan Toscano Martinez</i>	
Selección y clasificación de genes cancerígenos utilizando un método híbrido filtro/wrapper	85
<i>Luis Alberto Hernández Montiel, Carlos Edgardo Cruz Pérez, Juan Gabriel Ruiz Ruiz</i>	
Análisis exploratorio para la caracterización de la adicción a la cocaína a través del aprendizaje computacional.....	99
<i>Arturo Téllez-Velázquez, Eduardo A. Garza-Villareal, Jorge J. González-Olvera, Raúl Cruz-Barbosa</i>	

Diseño de métricas basadas en estresores laborales para desarrolladores de software

Ma. Alejandra Zamudio¹, Ramón R. Palacio², Luis A. Castro³

¹ Instituto Tecnológico de Sonora, Unidad Nainari,
Sonora, México

² Instituto Tecnológico de Sonora, Unidad Navojoa,
Sonora, México

alejandra.aguilasoch@hotmai.com, luis.castro@acm.org ramon.palacio@itson.edu.mx

Resumen. En este trabajo se investigaron los estresores laborales en desarrolladores de software con el objetivo de determinar un modelo de métricas que ayude a predecir y controlar los niveles de estrés. Para ello, se realizaron 10 entrevistas semiestructuradas a ingenieros en software y se analizaron los datos utilizando técnicas de teoría fundamentada. Los resultados muestran cuales son los principales estresores laborales en nuestros informantes. Además, se presenta un modelo constituido por cuatro métricas con sus respectivos ítems de medición: carga de trabajo, fatiga, demanda, y distracción laboral. Los resultados de este trabajo se pueden utilizar como base para la creación de herramientas creativas que permitan identificar patrones de comportamiento del desarrollador de software, de modo que ayuden a mejorar la productividad y disminuir las complicaciones psicológicas y biológicas que trae consigo el estrés durante el proceso de desarrollo de productos de software.

Palabras clave: estrés laboral, métricas de estrés, desarrolladores de software.

Design of Metrics Based on Work-related Stressors for Software Developers

Abstract: This study investigated work-rated stressors in software developers aimed at determining a metric model to help predict and control stress levels. To this end, 10 semi-structured interviews were conducted with software engineers and data were analyzed using Grounded Theory techniques. The results of this work show the main work stressors in our informants. In addition, in this work, we present a model that consists of four metrics with their respective measurement items: workload, fatigue, demand, and work-distraction. The results of this work can be used as a basis for the creation of creative tools that allow the identification of behavioral patterns of the software developer, so they

can help improve productivity and reduce psychological and biological difficulties caused by stress during the software development process.

Keywords: work stress, stress metrics, software developers.

1. Introducción

El estrés laboral o también denominado ocupacional se refiere al nerviosismo, ansiedad y angustia relacionada con el trabajo, lo cual deriva en el sentimiento de un conjunto de reacciones emocionales, cognitivas, de comportamiento y biológicas que afectan a las personas [1]. Un estresor se refiere a “la condición o situación que provoca una respuesta negativa como la ira, frustración, ansiedad o tensión” [2]. Este concepto hace referencia al estímulo perturbador o amenazante que, tras su percepción y evaluación, genera una respuesta de estrés [3]. En este sentido, toda situación o condición presente en el medio externo, puede convertirse en un estresor laboral. En la industria del software, el estrés laboral adquirió mayor relevancia en los últimos años, a partir del incremento exponencial de la demanda ocasionada por los diferentes negocios que requieren adoptar continuas innovaciones tecnológicas para incrementar su ventaja competitiva [4, 5]. Esta exigencia ha generando notables cambios en el patrón de desgaste de los desarrolladores de software, provocando monotonía, aislamiento y colapso, las cuales son representaciones de estrés [6]. Como se menciona, el estrés laboral, es ocasionado por las exigencias que superan las habilidades, conocimientos y capacidades de los trabajadores para afrontar situaciones, el cuál repercute en la afectación de una gran cantidad de variables incluyendo productividad, eficiencia organizacional, salud física y mental [7]. En este sentido, el estrés laboral ha sido identificado como uno de los retos primordiales en el entorno empresarial actual, además de ser uno de los riesgos emergentes más importantes dentro de las organizaciones.

La influencia del estrés laboral en desarrolladores de software es una evidente realidad conocida a nivel mundial por diversas investigaciones que han demostrado los factores que atraen el estrés en los profesionales del software [2, 8, 9]. Además se ha determinado cómo es la relación entre los síntomas depresivos, el estrés ocupacional y las características del personal de TI [4, 10]. Así como los efectos del estrés durante la generación de errores en el proceso de desarrollo de software [11]. A pesar de los múltiples estudios que existen en relación a este tema, ninguno de ellos se ha enfocado en establecer un conjunto de métricas que permitan identificar el grado de estrés entre los profesionales de software.

En México, el estrés ocupacional en desarrolladores de software ha sido poco examinado y de acuerdo a [12] el país representa el primer sitio de estrés ocupacional en el mundo, superando a China y Estados Unidos. Por otra parte, [13] reporta que todos los profesionales manifiestan algún síntoma de estrés laboral, además que el 60% sostiene altos niveles de éste. Por lo tanto, el interés de realizar este estudio surgió a partir de las evidentes dificultades que enfrentan los profesionales del software, las cuales repercuten en el desarrollo personal y profesional. Esto, debido a que día con

día, el ambiente de trabajo resulta ser más dinámico y se trabaja bajo presión, razones por las cuales resulta crítico medir y disminuir el estrés, con el propósito de lograr una mejora en el rendimiento individual de los desarrolladores de software, condición que a su vez permitirá incrementar el rendimiento organizacional.

Por ende, el presente trabajo tiene como objetivo determinar un modelo de métricas que ayude a predecir los niveles de estrés en desarrolladores de software pertenecientes a organizaciones dedicadas a la producción de software. En consonancia con lo anterior, se pretende resolver el siguiente cuestionamiento: ¿Cuáles son las métricas que indican el grado de estrés de los trabajadores del software?

Para realizar esto es importante considerar a las métricas como un tipo de medida cuantificable, las cuales son utilizadas para tomar mejores decisiones [14]. Además sirven como instrumento fundamental para ayudar a los directores de proyectos a seleccionar los indicadores adecuados en torno al monitoreo del estrés de su equipo de trabajo. En referencia a lo anterior se afirma que son un medio esencial para entender, monitorear, controlar, predecir y probar el desarrollo de los equipos de trabajo de software [15].

El resto del documento se encuentra estructurado de la siguiente manera: En la sección II se presentan los trabajos relacionados al estrés laboral entre profesionales del software. La sección III detalla los pasos utilizados para construir y evaluar la solución. En las secciones IV y V se muestran los resultados y conclusiones del estudio.

2. Trabajos relacionados

Con la finalidad de identificar y entender la forma en que se ha abordado el estrés laboral en desarrolladores de software, en este apartado se presentan un conjunto de trabajos que describen algunos indicadores estresores que han sido percibidos en este tipo de trabajadores. Dado que resulta interesante tener una perspectiva general de esto y así categorizar dichos estresores acorde a su impacto en lo físico o psicológico del trabajador.

Diversos estudios se han enfocado en analizar el estrés ocupacional entre los profesionales de software. En [4] realizaron una investigación con el objetivo de examinar las relaciones entre el estrés laboral y los síntomas depresivos en los ingenieros y gerentes de software de una empresa japonesa. El instrumento de medición contempló ocho indicadores estresores subjetivos (sobrecarga de trabajo, falta de control, falta de recompensas, ambigüedad en el desarrollo profesional, conflictos con el equipo de trabajo, dificultad en la comunicación con los usuarios, cambio de TI y equipos de información inadecuados) y dos objetivos (horas de trabajo excesivas y promedio de horas frente al computador). Para los síntomas depresivos utilizaron escalas de depresión del centro de estudios epistemológicos (CES-D). La población de estudio fue de 1,694 ingenieros de software y 296 gerentes, todos de género masculino. Después de efectuar el análisis reportaron que los ingenieros tienen más falta de control de trabajo, menos sobrecarga laboral y cambios de tecnología informática que los gerentes. Además, los gerentes trabajan más horas pero usan menor cantidad de tiempo

frente a una pantalla y tienen niveles de depresión más bajos en comparación con los ingenieros.

Por otro lado, [11] presentan una investigación cuantitativa con el objetivo de determinar los efectos del estrés sobre la generación de fallas durante el desarrollo de software. Como participantes eligieron a dos equipos de desarrollo conformados por dos programadores a quienes se les atribuyó realizar un sistema de gestión de bibliotecas bajo las mismas circunstancias excepto las metodologías de diseño. Uno de los equipos trabajó con diseño funcional mientras que el otro trabajó con diseño estructurado. Para esto utilizaron métricas psicológicas para evaluar el estrés mental ya que son fáciles de implementar. La métrica “presión por la fecha de entrega” fue la más eficaz, al revelar el nivel de estrés mental, mientras que la carga de trabajo tuvo la más alta correlación al número de fallas causadas. Para el estrés físico se usó una métrica llamada “condición física” que calculaba la afectación de dicho estrés. Después de aplicar métricas psicológicas y físicas para la medición de estrés durante el proceso de desarrollo se obtuvieron los siguientes resultados: los fallos de estrés representan una alta proporción de todas las fallas producidas, aunque el equipo que empleó la metodología diseño funcional tuvo mayor defectos que el de diseño estructurado; las fallas de naturaleza humana está correlacionada con los altos niveles de estrés; y por otro lado, el estrés físico lleva a cometer más fallas que el estrés mental.

En otro trabajo similar se examinó la relación entre los agentes estresores y las características del personal de TI de Singapur [10]. El instrumento fue elaborado mediante una extensa revisión bibliográfica y validado por expertos. La recolección de datos fue realizada a los trabajadores de TI pertenecientes a una empresa del ramo industrial mediante 257 encuestas electrónicas y 24 empleados del área de TI que participaron en cinco grupos focales con sesiones dirigidas de aproximadamente una hora y media. Tras hacer el análisis de los datos se dedujo que los principales factores que emergen como agentes estresores laborales son: la exigencia laboral, relaciones con el equipo de trabajo, preocupaciones, mantenimiento de sistemas, y ambigüedad de funciones y tareas administrativas.

Por otra parte, en [2] se desarrolló un instrumento para estudiar el estrés en los profesionales del software. Dicho cuestionario fue creado a partir del análisis de literatura existente e interacciones con académicos y empleados de software. Para este estudio se llevó a cabo el análisis factorial exploratorio a partir de 156 encuestas respuestas. Como resultados se identificaron que los principales factores que atraen estrés en los desarrolladores de software son: temor a la obsolescencia con el 13.1% de la varianza total, interacción con el equipo de trabajo 12.4%, interacción con el cliente 7.2%, interfaz trabajo-familia 6%, sobrecarga de roles 5.5%, cultura de trabajo 4.9%, limitaciones técnicas 4.6%, apoyo familiar hacia la profesión 3.7%, exceso de trabajo 3.1% y riesgo al uso de nueva tecnología 3%.

Otro estudio relacionado exploró los sucesos que generan estrés entre los profesionales de TI que asumen diversos roles en las compañías de Pakistán [8]. Dicha investigación se apoyó de técnicas cuantitativas para la recolección de datos a una población integrada por programadores, administradores de proyectos e ingenieros de calidad a través de un instrumento de medición que contenía los indicadores propuestos por [2]. Tras recabar los datos y efectuar el análisis se infiere que los factores no

contribuyen de la misma forma entre los profesionales, a pesar de que las variables contempladas tienen el mismo peso, éstas impactan de manera diferente. Por lo que se encontró que el miedo a la obsolescencia, interacción con el cliente y carga de trabajo son los que más contribuyen al estrés del trabajador.

Además, [9] utilizó un enfoque cuantitativo con la finalidad de estudiar la relación entre las horas de exposición frente al computador y el estrés en los profesionales de TI. Veintiún ingenieros en software pertenecientes a empresas tecnológicas fue a quienes se les aplicó el cuestionario de ansiedad estado-rasgo (STAI) conformado por 20 indicadores de ansiedad/estado y 20 de ansiedad/rasgo. Al examinar los datos se encontró que estar expuesto más de ocho horas diarias frente a la computadora ocasiona mayor nivel de estrés a los trabajadores. Por lo tanto, existen relaciones positivas entre el nivel de estrés y el uso de la computadora.

Después de haber explorado los diferentes trabajos, es claro observar que aspectos relacionados a la cultura y exceso de trabajo son los que más fueron encontrados en la literatura, mientras que los menos considerados son el cambio de tecnología de información, síntomas depresivos, ansiedad y condición física. Dicho lo anterior, es importante mencionar que el nivel de consideración no es sinónimo del grado de incidencia, debido a que los indicadores impactan de manera diferente.

3. Metodología

Dada la naturaleza de la investigación se optó por utilizar una metodología cualitativa, puesto que el objetivo es entender la perspectiva de la población bajo estudio [16]. Para su elaboración se utilizaron como base técnicas de codificación abierta y codificación axial pertenecientes a la teoría fundamentada. Esta última se define como la teoría explicativa de un fenómeno en particular que emerge de los datos. Su finalidad es describir realidades, predecir sucesos, construir teorías o proposiciones no a priori [17].

3.1. Participantes

Se invitó a participar en entrevistas semiestructuradas a ingenieros en software pertenecientes a empresas localizadas en el municipio de Cajeme Sonora, México, dedicadas al desarrollo de software a la medida, implementación de software aeroespacial así como soporte y mantenimiento de sistemas. Los participantes fueron 10 ingenieros, entre ellos cuatro mujeres y seis hombres de edades entre 23 y 30 años. Nueve de estado civil solteros y uno casado. Con experiencia laboral menor a tres años.

Resulta oportuno destacar que el sector de software en México cuenta con importantes fortalezas que lo ubican como un claro centro de atracción de inversiones en el mercado global. Actualmente ocupa el octavo lugar mundial como mejor destino para proveer servicios de tecnologías de información [18]. En el periodo 2010-2015 esta industria representó una tasa de crecimiento promedio anual de 13% [19]. Además, desde el año 2008 a la fecha han generado más de 7,000 mil empleos que representan un elevado impacto en el desarrollo tecnológico y social del país [20].

3.2. Materiales y métodos

La técnica empleada para recopilar datos fue entrevistas semiestructuradas ya que permiten al entrevistador estar preparado y otorga a los informantes amplia libertad de expresión [21]. Para esto se utilizó un protocolo de entrevista con preguntas abiertas previamente elaboradas que abordaban los siguientes temas:

Incompetencia laboral: Los cuestionamientos de este tema se basan en analizar aspectos relacionados con el estrés que provoca el aprendizaje y la adaptación respecto a las nuevas tecnologías y mejores prácticas de desarrollo.

Interacción con el equipo de trabajo: Incluye interrogantes para conocer el trato que recibe el desarrollador de software, la existencia de rivalidades, discriminación o favoritismo.

Interacción con el cliente: Las interrogantes agrupadas en este apartado tienen la finalidad de investigar cómo es la relación que encara el desarrollador durante la interacción con los clientes en el proceso de desarrollo de software.

Carga laboral: Se encarga de indagar la cantidad de actividades asignadas a los desarrolladores de software, y la percepción del estrés que puede causarle.

Ambiente laboral: Se cuestiona la percepción del desarrollador de software sobre su ambiente de trabajo y los aspectos relacionados que son causantes de estrés.

Las entrevistas fueron grabadas con consentimiento previo, sin interrupciones, las cuales tuvieron duraciones entre 50 y 53 minutos. Después, se procedió a su transcripción textual para su análisis.

3.3. Procedimiento de análisis

Para el análisis se utilizó teoría fundamentada con la finalidad de detectar patrones en los datos. El análisis utilizando teoría fundamentada involucra tres principales etapas:

1. Codificación abierta, donde se analizó cada renglón de las transcripciones buscando conceptos, acciones o enunciados que ayuden a describir los estresores laborales. A la información encontrada se le asignó un código de clasificación, los cuales fueron agregados a una lista que fue depurada al final de hacer la codificación, dejando aquellos que tienen una fuerte relación con el estrés en los desarrolladores de software.
2. Codificación axial, la cual consistió en examinar cada uno de los códigos, buscando similitudes para agruparlos conforme a sus circunstancias y condiciones. A cada agrupación se le asignó una propiedad que los definiera para determinar la categoría. Posteriormente se procedió en determinar los factores críticos con la finalidad de conocer los elementos que intervienen en el estrés de los profesionales del software, a partir de ello se elaboraron el conjunto de métricas.

3. Codificación selectiva, consistió en integrar y elegir las categorías para delimitar la teoría.

4. Resultados y discusiones

En este apartado se presentan los resultados del análisis de las entrevistas, centrados principalmente en la pregunta de investigación: ¿Cuáles son las métricas que indican el grado de estrés de los trabajadores del software?, planteada en la sección I.

A. Estresores laborales

Este análisis inició con la identificación de los principales estresores laborales, los cuales responden parcialmente a la pregunta de investigación. Por lo que a continuación se detallan cada uno de ellos:

Sobrecarga laboral. Este estresor hace hincapié en sobrepasar el límite de requerimientos que un desarrollador de software es capaz de resolver en un determinado tiempo. Por ejemplo, el Participante 6 comenta: “...*me estreso en particular cuando me asignan mucha carga de tareas y muy poco tiempo para realizarlas*” (Participante 6). De manera similar, el Participante 3 explica: “...*es frustrante tener muchos requerimientos por resolver y sentirme incapaz de terminarlos a tiempo*” (Participante 3). En ese sentido, uno de los aspectos que parecen influir en niveles de estrés puede ser el número de tareas, porque pueden ser demasiadas a realizar en un corto plazo. Sin embargo, es interesante cómo el Participante 3 hace referencia a su capacidad para realizarlas, ya que el sentirse abrumado por el número de tareas puede tener un efecto negativo en su percepción de capacidad para realizar su trabajo.

Organización en la fase final del proyecto. Se refiere a la administración inconsistente de las actividades respecto a la fecha de entrega parcial o total del proyecto al cliente. En relación a este estresor el Participante 10 expresa: “...*siempre que estoy en la fase final del proyecto siento que mi rendimiento es bajo, porque tengo el tiempo encima y muchos requerimientos pendientes por resolver*” (Participante 10). Además el Participante 4 argumenta: “...*la mala organización del tiempo en las actividades provoca en mi niveles altos de frustración durante la fase final del proyecto*” (Participante 4). Respecto a lo antes mencionado, es claro observar que al tener inadecuada organización durante el proyecto se podrían traer consecuencias en los resultados u “outputs” del desarrollador durante la fase final. No obstante, puede causar afectaciones psicológicas en el individuo, tal es el caso del Participante 4, quien percibe sentimientos de ira y decepción.

Complejidad de actividades. Representa la dificultad que constituye a un algoritmo de programación. Surge a partir de los inconvenientes que tiene que superar el desarrollador de software para concluir la actividad. Como muestra de ello, el participante 2 manifiesta: “...*en el proyecto pasado cuando no me tocaba hacer*

las tareas difíciles me causaba alegría porque sabía que no batallaría para terminirlas, pero cuando me tocaba realizarlas era preocupante y me estresaba mucho porque no sabía si iba a terminirlas en tiempo y forma” (Participante 2). Esto, nos revela que un parámetro interesante a considerar en la asignación de tareas complejas es el tiempo, ya que desde antes de iniciar la tarea puede agobiar al desarrollador.

Incompetencia laboral. Se refiere a la necesidad de poseer las competencias necesarias para el desempeño exitoso. Los desarrolladores de software relacionan este estresor con la adaptación, aprendizaje e implementación de nuevas tecnologías y buenas prácticas de desarrollo. Para ejemplificar esto, el Participante 9 argumenta: “...es indispensable para mí estar en constante actualización respecto a las nuevas tecnologías y mejores prácticas de desarrollo para poder estar a la vanguardia profesional y brindarles todo lo que la compañía y los clientes requieren” (Participante 9). Con este comentario, el participante percibe la necesidad de aumentar sus conocimientos y habilidades para tener aumentar capacidad productiva.

Actividades no preferidas. Este estresor representa las actividades que un desarrollador de software no disfruta realizar. Por ejemplo, el Participante 1 dice: “...realizar la documentación u otras actividades aburridas, simplemente me ocasiona estrés, distracción, cansancio y frustración porque no disfruto lo que estoy haciendo” (Participante 1). En este sentido, el participante percibe estas actividades como causantes de respuestas psicológicas negativas, al mismo tiempo pueden ocasionarle distracciones.

B. Métricas

A partir de la identificación de estresores, se diseñó un modelo de métricas que tienen como objetivo medir el estrés en desarrolladores pertenecientes a la industria del software. Acorde a lo anterior, la Tabla 1 muestra cuatro métricas propuestas con sus respectivos ítems de medición, los cuales se describen a continuación

Carga de trabajo: consta de seis ítems relacionados con la cantidad, duración y tamaño de las tareas asignadas a un desarrollador. Tiene la finalidad de estimar el número de requerimientos óptimos que un desarrollador puede realizar sin sobrepasar su carga laboral.

Fatiga laboral: su propósito es medir la complejidad de actividades e incompetencia laboral, por lo tanto, ésta métrica se integra de seis ítems que abordan información relacionada al sobreesfuerzo o tensión sostenida por un desarrollador de software.

Demanda laboral: tiene el objetivo de contrarrestar el estrés respecto a la falta de organización en la fase final del proyecto, para ello se incluyen dos ítems relacionados con el tiempo requerido, disponible y adicional demandando por el desarrollador de software.

Distracción laboral: métrica conformada por tres ítems vinculados al estresor actividades no preferidas. Su fin es determinar tareas que ocasionan estrés al desarrollador relacionadas con la apatía.

Tabla 1. Métricas para medir el estrés en desarrolladores de software.

Métrica	Ítems	Definición del ítem	Estresores afectados
Carga de trabajo	Tareas asignadas	Número de tareas impuestas	Sobrecarga laboral
	Tasa de llegada de la tarea	Número de tareas impuestas en una unidad de tiempo	
	Tamaño de la tarea	Cantidad de actividades necesarias para completar la tarea	
	Tiempo destinado a completar la tarea	Duración para completar la tarea	
	Tiempo invertido en completar la tarea	Duración empleada para completar la tarea	
	Prioridad de la tarea	Nivel de importancia Nivel de urgencia	
Fatiga laboral	Complejidad cíclica de la tarea	Número de caminos independientes dentro de un fragmento de código	Complejidad de actividades Incompetencia laboral
	Conocimiento técnico	Porcentaje de conocimientos técnicos relacionados al proyecto	
	Autoaprendizaje técnico	Porcentaje de aprendizaje adicional involucrado durante la ejecución de la tarea	
	Información de la tarea	Porcentaje de claridad en cuanto a metas y objetivos de la tarea	
	Incertidumbre de la tarea	Porcentaje de información necesaria proporcionada para la actividad	
	Número de bugs	Número de defectos en el software	
Demanda laboral	Demanda temporal del proyecto	Tiempo requerido para terminar la tarea y el disponible	Organización en la fase final del proyecto
	Horas extra de trabajo	Porcentaje de horas extra semanales	
Distracción laboral	Interrupciones de la tarea	Número de veces en que se suspende la ejecución de la tarea	Actividades no preferidas
	Duración de la interrupción	Tiempo que transcurre desde el inicio al fin durante la interrupción	
	Frecuencia de interrupciones	Número de veces que sucede la interrupción de la tarea	

La mayoría de los factores estresores identificados ya han sido discutidos en estudios similares con diversas semejanzas. En su trabajo [2] presentaron un listado de factores estresores similares a los identificados en esta investigación. Tal vez la diferencia más marcada tiene que ver con la aparición de un nuevo factor que ellos no contemplaron, el cual está relacionado con las actividades no preferidas por los desarrolladores. Por otro lado [4, 8-11], ha dejado demostrado que la sobrecarga laboral es una de las circunstancias más generadoras de estrés. Respecto al estresor organización en la fase final del proyecto en [2] determinaron que la presión por la fecha de entrega es eficaz al momento de revelar niveles de estrés mental. A pesar que diversos autores han

orientado sus esfuerzos en determinar cuáles son los factores estresores más influyentes en desarrolladores de software, ninguno de ellos se enfocó en diseñar métricas que ayuden a identificar el grado de estrés entre los profesionales.

5. Conclusiones

En este trabajo se presentaron los resultados de un estudio cualitativo sobre el diseño de métricas que permiten controlar los factores estresores en los desarrolladores de software. Los resultados serán utilizados como base para la creación de herramientas creativas que permitan determinar patrones de comportamiento, de modo que ayuden a que los desarrolladores a ser más productivos y disminuir las complicaciones psicológicas y biológicas que trae consigo el estrés durante el proceso de desarrollo de productos de software, a su vez con dichos indicadores se podrá mostrar el nivel de estrés de cada uno de ellos.

Por otro lado, estas métricas pueden ser utilizadas por los directores de proyectos de software para crear mecanismos apropiados relacionados con la selección de tareas adecuadas para tener un mejor rendimiento por parte de su equipo de trabajo. Como trabajo futuro, se pretende evaluar con un caso de estudio práctico el estrés de los desarrolladores de software utilizando las métricas planteadas anteriormente, con la finalidad de determinar patrones de comportamientos y crear estrategias para ayudarlos a que trabajen de una forma más confortable. El estudio se llevó a cabo sólo con ingenieros en software, razón por la cual los resultados deben interpretarse con precaución, además fue un estudio de tipo exploratorio transversal y está limitado en su capacidad de interpretaciones causales.

Referencias

1. Singh, J., Pratap, B., Narban, S., Singh, J.: A Conceptual Study on Occupational Stress (Job Stress/Work Stress) and its Impacts. Jaipur (Rajasthan) 2 MSc (CS), pp. 2395–4396 (2016)
2. Rajeswari, K. S., Anantharaman, R. N.: Development of an Instrument to Measure Stress Among Software Professionals: Factor Analytic Study. In: Proceedings of the 2003 SIGMIS Conference on Computer Personnel Research: Freedom in Philadelphia—leveraging Differences and Diversity in the IT Workforce, ACM, New York, NY, USA, pp. 34–43 (2003)
3. Durán, M.M.: Bienestar psicológico: el estrés y la calidad de vida en el contexto laboral. Revista nacional de administración 1, pp. 71–84 (2010)
4. Haratani, T., Fujigaki, Y., Asakura, T.: Job stressors and depressive symptoms in Japanese computer software engineers and managers. Advances in Human Factors (1995)
5. Laudon, K. C., Laudon, J. P.: Management information systems : Managing the Digital Firm. 9th edition, pp. 103–105 (2007)
6. Velázquez, A.: Mujer, trabajo y estrés. Revista latinoamericana de salud en el trabajo (2001)

7. Chilton, M. A., Hardgrave, B. C., Armstrong, D. J.: Person-Job Cognitive Style Fit for Software Developers: The Effect on Strain and Performance. *Journal of Management Information Systems* 22, pp. 193–226 (2005)
8. Rashidi, Z., Jalbani, A. A.: Job Stress among Software Professionals in Pakistan: A Factor analytic Study. *J Independent Stud Res (MSSE)*, 7, pp. 1–12 (2009)
9. Sushma, S. S.: Influence of Hours of Computer Usage on Stress in Software Professionals. *Journal of Evolution of Medical and Dental Sciences*, 3, pp. 9621–9626 (2014)
10. Lim, V., Teo, T.: Occupational stress and IT personnel in Singapore: factorial dimensions and differential effects. *International Journal of Information Management* (1999)
11. Furuyama, T., Arai, Y., Iio, K.: Analysis of fault generation caused by stress during software development. *Journal of Systems and Software* (1997)
12. Villaseñor, N.: Te sientes quemado o desgastado por el trabajo? Es el síndrome de burnout. In: *México Ciencia y Tecnología*. Recuperado de: <http://conacytprensa.mx/index.php/ciencia/salud/6398-sindrome-de-burnout-reportaje-final> (2016)
13. Prado J Estudio confirmatorio de la escala mexicana de desgaste ocupacional (EMEDO): Un instrumento de burnout para mexicanos. *Revista Interamericana de Psicología*. (2016)
14. Pfleeger, S. L.: Assessing measurement. *IEEE Software*, Vol. 14, No. 25 (1997)
15. Serrano, M., Piattini, M., Calero, C.: Un método para la definición de métricas de software. In: *1er Workshop en Métodos de Investigación y Fundamentos filosóficos en Ingeniería del Software y Sistemas de Información*, pp 65–74 (2002)
16. Olabuénaga, J.: *Metodología de la investigación cualitativa* (2012)
17. Strauss, A., Corbin, J.: *Bases de la investigación cualitativa: técnicas y procedimientos para desarrollar la teoría fundamentada* (2002)
18. Kearney, A. T.: A.T. Kearney Releases 2016 Global Services Location Index (2016)
19. PROMÉXICO: *Servicios de TI y software en México* (2016)
20. PROMÉXICO: *ProMéxico e industria de tecnologías de la información y comunicaciones desarrollan plan de trabajo para fortalecer al país* (2017)
21. Longhurst, R.: Semi-structured interviews and focus groups. *Key methods in geography* (2003)

STUTTY: Aplicación móvil para mermar la tartamudez utilizando la técnica del “tartamudeo-sencillo”

Ricardo Solano Monje¹, Nayeli Joaquinita Meléndez Acosta²,
Homero Vladimir Ríos Figueroa³

¹ Centro Veracruzano de Investigación y Posgrado, Xalapa, Veracruz,
México

² Universidad del Istmo, Ciudad Ixtépec, Oaxaca,
México

³ Universidad Veracruzana, Xalapa, Veracruz,
México

rsolanomonje@gmail.com nayelimelendez@gmail.com hrios@uv.mx

Resumen. Este artículo se enfoca en el desarrollo de una aplicación móvil (app) STUTTY, para ayudar en el problema del tartamudeo para niños de entre 6 y 11 años. Con el uso de STUTTY se pretende proporcionar una herramienta de apoyo al paciente en el proceso de tratamiento y prevención de la tartamudez. La aplicación móvil propuesta utiliza la técnica del “tartamudeo-sencillo”, la cual está enfocada en separar la palabra en sílaba, con la diferencia susceptible de un alargamiento del sonido vocal que sigue a la consonante. Esta aplicación está dirigida al tipo de tartamudez clónica, en la cual existe una frecuencia en las repeticiones de sílabas, principalmente en consonantes, más en oclusivas y el momento de la tartamudez está presente al principio de la palabra. Para la realización de STUTTY se utilizó el modelo desarrollo incremental para identificar los servicios con prioridad y fueran los primeros en desarrollarse. Las pruebas muestran una efectividad del 73.6%.

Palabras clave: tartamudez, aplicación móvil, tartamudeo-sencillo, consonantes oclusivas.

STUTTY: Mobile Application for Diminishing Stuttering Using the "Simple Stuttering" Technique

Abstract: This article focuses on the development of a Mobile App named STUTTY made to help with the stuttering problem for children between the ages of 6 and 11. STUTTY is intended to provide a tool to support the patient in the process of treatment and prevention of stuttering. The proposed mobile application uses the "simple-stuttering" technique, which is focused on separating words into syllables, susceptible to an extension of the vocal sound following the consonant. This application is directed to the type of cloned

stuttering, in which there is a frequency in the repetitions of syllables, mainly in consonants, more in occlusive and the moment of stuttering is present at the beginning of the word. For the implementation of STUTTY, the incremental development model was used to identify the services with priority and were the first ones to be developed. The tests show an effectiveness of 73.6%.

Keywords: stuttering, mobile application, simple stuttering, consistent occlusion.

1. Introducción

La tartamudez es una dificultad del habla que muchas veces se presenta cuando se habla en público. Hay más hombres que mujeres con este problema, en una proporción de 4 a 1 y comúnmente inicia en la infancia entre 2 a 5 años. La tartamudez no es una enfermedad, es involuntaria y aparece y desaparece por periodos de tiempo variables [1].

En la infancia los niños empiezan a tartamudear, aunque la mayoría de las veces este problema desaparece sólo, sin la ayuda de un profesional, si la dificultad continúa después de un año desde su comienzo, existe el riesgo de que el problema se mantenga [2]. Por ello es conveniente realizar una intervención temprana que ayude a prevenir el problema con una terapia adecuada, que ayude a los niños a conseguir un habla fluida, para que la dificultad no persista.

Todas las personas tartamudean de diferente forma, algunas repiten sílabas o palabras, otras alargan sonidos y otras cuando hablan o antes de terminar una palabra se bloquean [3], [1].

Esta aplicación tiene como objetivo fundamental proporcionar una herramienta de apoyo al paciente en el proceso de tratamiento y prevención de la tartamudez.

Este artículo presenta el desarrollo de una aplicación para dispositivos móviles, que ayuda en el tratamiento de la tartamudez clónica en niños entre 6 y 11 años, utilizando la técnica “tartamudeo-sencillo” de Ann en [4]. El contenido del artículo se encuentra organizado de la siguiente manera, en la segunda sección se presenta el marco teórico respecto a la tartamudez y se describen algunas aplicaciones que se han implementado para tratar dicho problema, luego en la tercera sección se explica el funcionamiento de la técnica “tartamudeo-sencillo” utilizada para el desarrollo de la aplicación móvil propuesta, posteriormente en la cuarta y quinta sección se presenta la metodología utilizada y la implementación de la aplicación móvil STUTTY, enseguida se muestran las pruebas realizadas y finalmente en la sexta sección se presentan las conclusiones y trabajo futuro.

2. La tartamudez

Tartamudeo es la interrupción en la fluidez normal del habla, que afecta la comunicación y se acompaña de tensión muscular, miedo y estrés [3].

En la tartamudez intervienen muchos factores tales como los neurológicos, psicológicos y sociales, la tartamudez afecta entre 1,5% y 2% de la población adulta

[3]. Esta alteración en la fluidez del habla tiene un gran impacto en la calidad de vida y en el rendimiento escolar, laboral y social [1], [5]. Los niños de preescolar que tartamudean son menos precisos en tareas implícitas de respuesta verbal y exhiben tiempos de reacción más lentos que los niños que no tartamudean [6]. Esto no significa que sean menos inteligentes, pero si afecta su rendimiento escolar.

El tartamudeo se presenta de manera diferente en cada persona. Alguna persona tendrá problemas en un conjunto de palabras mientras que otra los tendrá en otro diferente. Otras personas tartamudean cuando hablan en voz alta, pero no lo hacen cuando nadie los escucha, algunas otras tartamudean cuando nadie los escucha y hablan fluido cuando hablan en voz alta [4].

Otro aspecto importante es que las personas que tartamudean por lo regular hablan muy rápido, quizás porque quieren terminar pronto la lectura, antes de que les afecte el tartamudeo. Pero hablar rápido estorba más que ayudar.

2.1. Tartamudez en niños

La tartamudez inicia con frecuencia alrededor de los 2 años, sin que exista un motivo desencadenante [7], pero con frecuencia desaparece sin la ayuda de un especialista. La tartamudez puede identificarse a tiempo para prevenir consecuencias de su persistencia.

Aproximadamente el 80% de los tartamudos son niños [8]. Además la mayoría de los tartamudos comenzaron a tartamudear en la niñez, es decir, el 90% de casos de tartamudeo iniciaron antes de los 4 años.

Cuando un niño tartamudea es necesario no ponerle atención a su tartamudeo, si se le demuestra ansiedad o se le hace sentir que su lenguaje es inadecuado, se le induce a evitar el tartamudeo [4]. El niño debe saber que es libre de tartamudear, no hay que presionarles a hablar con fluidez.

La tartamudez temprana se presenta principalmente en repeticiones de sílabas y palabras. En la tartamudez existe algo llamado momento del tartamudeo, que es el instante en el que ocurre el tartamudeo durante el habla, algunos de estos pueden ser: repetición de palabra, repetición de sonidos o sílabas, repetición de frases, frases incompletas, palabras rotas y prolongación [5].

2.2. Momentos de la tartamudez

Por lo general los momentos de tartamudez en el lenguaje no suceden al azar, con frecuencia se presenta al principio de una palabra o al iniciar una frase [4], [5].

Los momentos se pueden clasificar en tres principalmente [5]:

La consonante inicial de la palabra. La presencia de tartamudez es más frecuente en consonantes y en un 90% en sonidos iniciales de las palabras

La primera palabra de una frase. Así como ocurren en la primera sílaba de una palabra, también ocurre en la primera palabra de una frase.

Al final de las palabras. Es muy raro, pero en ocasiones también ocurre el tartamudeo al final de las palabras [9].

2.3. División de los sonidos

Los sonidos se dividen en dos vocales y consonantes, estos últimos pueden dividirse a su vez en dos grupos oclusivas y continuas [4].

En las consonantes oclusivas debemos hacer explotar algo antes de oír el sonido, estas son seis: “p”, “b”, “t”, “d”, “k” y “g”.

Las consonantes continuas son aquellas que se pueden decir hasta que se acabe el aliento y son el resto de consonantes.

2.4. El tartamudeo en sonidos determinados

Por lo regular el tartamudeo no ocurre en un solo grupo de sonidos, sino en varios. La mayoría de las veces el problema del tartamudeo se encuentra en las consonantes oclusivas [4]. Los sonidos para el análisis del tartamudeo se pueden agrupar en cinco: vocales, oclusiva, nasales, semivocales y continuas restantes, ver Fig.1.

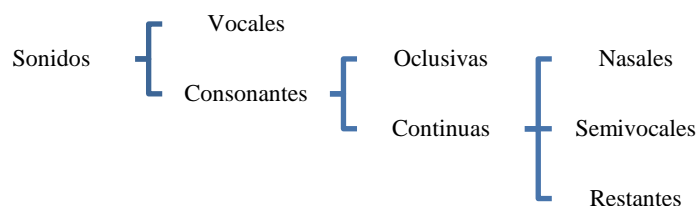


Fig. 1. División de los sonidos en el análisis del tartamudeo [4].

Los sonidos nasales son “m”, “n” y “ñ”, estos sonidos provienen de la nariz en lugar de la boca.

Los sonidos semivocales son dos “y” y “w”, estos sonidos inician con una vocal y se deslizan a otro sonido vocal.

2.5. Clasificación de la tartamudez

La tartamudez se puede clasificar según su origen o su manifestación [1]:

Según su origen puede ser **neurológica** causada por una lesión del cerebro, **psicógena** producida por un trauma difícil y **de desarrollo** que ocurre cuando el niño comienza hablar.

Según su manifestación puede ser **tónica** que se presenta cuando hay interrupciones por espasmos, **clónica** cuando existe una frecuencia en las repeticiones de sílabas principalmente en consonantes, más en oclusivas y está presente al principio de la palabra y **tónico-clónica o mixta** esta es una combinación de las dos anteriores, además la fluidez del habla es lenta y de repente se acelera.

2.6. Técnicas de la tartamudez

Please note that the first paragraph of a section or subsection is not indented. The first paragraphs that follows a table, figure, equation etc. does not have an indent, either. Existen numerosas técnicas para el control de tartamudez, algunas de ellas son [4]:

Seguimiento. Esta técnica funciona de la siguiente forma, se tiene grabada una cinta con un monólogo de otra persona, el paciente con tartamudeo la escucha utilizando audífonos y trata de producir lo que oye con voz suave, de esta forma habla y escucha al mismo tiempo. En algunas personas ha tenido éxito esta técnica.

Retroalimentación Auditiva Demorada (DAF, Delayed Auditory Feedback). El objetivo es que el paciente escuche sus propias palabras a través de audífonos, con un retraso de más o menos un quinto de segundo. Su eficiencia varía, pero en algunos casos el tartamudeo disminuye.

Habla Silábica. El paciente debe usar el mismo tiempo y dar el mismo énfasis a cada sílaba. Es muy fácil de aprender. Tiene mejores resultados cuando se aplica con intensidad.

Metrónomo electrónico. En esta técnica el paciente lleva unos audífonos que le transmiten una señal a sus oídos, el objetivo es hacer coincidir el énfasis de su habla con la señal. El paciente no requiere la asistencia constante de un terapeuta como en las otras terapias, el paciente sólo requiere el metrónomo y las instrucciones y puede continuar la terapia.

Encubrimiento (masker). En esta técnica la voz del paciente es disfrazada, lo que significa que el paciente es incapaz de escuchar sus propias palabras. Ya existe un aparato electrónico llamado Masker, al utilizarlo no es necesario realizar un esfuerzo para mejorar el tartamudeo. En algunas personas sirve en otras no.

Terapia de Johnson. Cuando existe temor al tartamudeo se trata de evitar las palabras, los sonidos y situaciones que se temen. El paciente pretende evitar el tartamudeo, pero para controlarlo es necesario dejar de evitarlo. Está terapia no va dirigida a romper los síntomas del tartamudeo, sino todo lo contrario a ganar el control del mismo, es decir aprender exactamente como se tartamudea y luego se debe tartamudear a propósito.

Algunas de las técnicas más usadas en aplicaciones móviles son retroalimentación auditiva demorada, donde el usuario escucha su voz con un poco de retraso, el metrónomo en esta técnica el usuario debe pronunciar cada sílaba en determinado tiempo y la técnica del espejo, en la cual el usuario observa su conducta mientras practica sílabas.

2.7. Trabajos relacionados

Algunas aplicaciones que se han implementado para tratar el problema de la tartamudez son:

Ernest Stuttering App. Esta aplicación está diseñada para niños de edad escolar y adolescentes que tartamudean. Implementa cuatro juegos llamados “Hablando”, “Identificando”, “Memorizando” y “Dibujando”. Además permite a los terapeutas trabajar con otros trastornos del habla. Pero no es gratuita [10].

Fluxxy. Es una aplicación móvil controlada por medio de la voz, diseñada para niños mayores de 6 años con tartamudez. Ayuda a entrenar el manejo de la voz y la respiración lúdica. Es gratuita, pero por el momento solo se encuentra en alemán [11].

Cure The Stuttering Free. Esta aplicación implementa la técnica de retroalimentación con retraso auditivo, graba la voz del paciente y su voz se reproduce con un retardo. Esta es la versión libre que limita a 60 segundos de uso continuo [12].

Sttuter Help. Esta aplicación móvil permite controlar la fluidez del habla y puede ser utilizado como parte de una terapia del habla, algunas de las técnicas que implementa son el metrónomo, el espejo y la retroalimentación con retraso auditivo. No es gratuita [13].

Stutter Rater. Es un contador de tartamudeo simple, esta aplicación es usada para diagnosticar y muestra el porcentaje de tartamudez por sílaba, así como sílabas por minuto. Es gratuita [14].

En su mayoría las aplicaciones móviles para ayudar a la tartamudez son de uso comercial y ninguna ha implementado la técnica del “tartamudeo-sencillo”.

3. Técnica “tartamudeo-sencillo”

La técnica de tartamudeo utilizada en el desarrollo de la aplicación es “tartamudeo-sencillo”, que consiste en pronunciar una palabra de forma natural, con la diferencia susceptible de colocar un alargamiento del sonido vocal que sigue a la consonante [4].

El funcionamiento de la técnica consiste en realizar una pronunciación adecuada de las palabras: la consonante se debe pronunciar suave, luego se debe deslizar sobre el sonido vocal el cual se debe prolongar (no separar ambos sonidos), y terminar con un sonido normal si es que existen más sílabas. En la Tabla 1 se muestra la técnica utilizando seis palabras con iniciales oclusivas:

Tabla 1. Tartamudeo-sencillo en consonantes oclusivas.

Palabra	Contante oclusiva	Sonido Vocal	Restante
Piedra	(Suave) << p >>	Prolongado << ie >>	Normal << dra >>
Bolsa	(Suave) << b >>	Prolongado << ol >>	Normal << sa >>
Taco	(Suave) << t >>	Prolongado << a >>	Normal << co >>
Dos	(Suave) << d >>	Prolongado << os >>	
Coco	(Suave) << c >>	Prolongado << o >>	Normal << co >>
Ganso	(Suave) << g >>	Prolongado << an >>	Normal << so >>

Ahora vemos las oclusivas seguidas por otra consonante, en estas palabras aparecen dos consonantes seguidas antes del sonido vocal, en este caso no se alarga la segunda consonante porque se distorsiona el sonido de la palabra, en estos casos se consideran las dos primeras consonantes como una sola unidad con un sonido suave, ver Tabla 2.

Tabla 2. Tartamudeo-sencillo en consonantes oclusivas seguidas por otra consonante.

Palabra	Contante oclusiva	Sonido Vocal	Restante
Premio	(Suave) << pr >>	Prolongado << e >>	Normal << mio >>
Blusa	(Suave) << bl >>	Prolongado << u >>	Normal << sa >>
Tractor	(Suave) << tr >>	Prolongado << ac >>	Normal << tor >>
Dragón	(Suave) << dr >>	Prolongado << a >>	Normal << gón >>
Clavo	(Suave) << cl >>	Prolongado << a >>	Normal << vo >>
Globo	(Suave) << gl >>	Prolongado << o >>	Normal << vo >>

La técnica de “tartamudeo-sencillo” se ha seleccionada debido a su simplicidad y porque ha mostrado en la práctica empírica buenos resultados en los pacientes que realmente trabajan para asimilarlo. La aplicación implementada en este trabajo está enfocada en la tartamudez clónica, que es un tipo de tartamudez muy común. Además de que esta técnica no ha sido implementada en ninguna aplicación móvil.

4. Metodología

4.1. Desarrollo incremental

La metodología utilizada para la implementación de esta aplicación móvil es desarrollo incremental. En esta metodología los clientes identifican de manera somera los requerimientos de la aplicación y se les establecen prioridades, de esta forma se entregan primero los de mayor prioridad, además eso significa que el usuario encontrará menos fallas en las tareas más importantes de la aplicación [15].

Una vez definidos los requerimientos se definen los incrementos. En cada incremento esta metodología permite entregar un producto operacional, aunque muchas de las veces los primeros incrementos son versiones incompletas.

Esta metodología ha sido seleccionada porque permite que el usuario experimente con la aplicación ayudando a clarificar sus requerimientos para los incrementos posteriores. Además de que el usuario puede seguir agregando más especificaciones.

4.2. Análisis de requerimientos

Esta subsección se centra en la descripción de los requerimientos funcionales, la identificación de su prioridad y la definición de los incrementos, pasos necesarios en el modelo de desarrollo incremental. La definición de los requerimientos iniciales se realiza utilizando dos técnicas, mediante entrevistas con el cliente, especialista en trastornos del habla y mediante estudios de documentación respecto al funcionamiento de la técnica “tartamudeo sencillo”.

Existen diversos aspectos a tomar en consideración para priorizar los requerimientos, la técnica utilizada para priorizar los requerimientos de la aplicación es ranking, donde

se asigna una posición a cada requerimiento que va de 1 a N, siendo N la cantidad de requerimientos y 1 el requerimiento de mayor importancia.

Tabla 3. Requerimientos iniciales de la aplicación.

No.	Requerimientos	Prioridad	Incremento
1	La aplicación debe permitir al usuario elegir un tipo de consonante oclusiva, nasal, semivocal, continua o vocal a practicar.	6	3
2	Al seleccionar una consonante o vocal, la aplicación debe mostrar una lista de palabras a practicar.	7	3
3	La lista debe contener 10 palabras a practicar, seleccionadas de forma aleatoriamente.	5	2
4	Al seleccionar una palabra se debe escuchar la pronunciación usando la técnica de “tartamudeo-simple”.	1	1
5	La aplicación debe permitir volver a escuchar de nuevo la palabra.	2	1
6	La aplicación debe permitir pasar a la siguiente palabra	3	1
7	Debe permitir regresar a la lista de palabras	4	2

Los requerimientos iniciales, su prioridad y los incrementos utilizados en la elaboración de la aplicación están definidos en la Tabla 3.

En la Tabla 3 se puede observar que se definen siete requerimientos iniciales y 3 incrementos, los cuales se conforman por dos o tres requisitos. Podemos ver que el primer incremento cumple con los requerimientos más importantes.

4.3. Diseño

En esta fase se describe brevemente el diseño. En la Fig. 2, se muestra el caso de uso general de la aplicación, donde se pueden apreciar las acciones más importantes que el usuario podrá realizar al usar la aplicación:

4.4. Desarrollo

Una "Actividad" es el nombre dado a una sola ventana de la aplicación móvil con la que el usuario puede interactuar en un momento dado, a partir de aquí hablaremos de actividad refiriéndonos a una ventana. En esta sección se mostrarán las interfaces realizadas en el entorno de desarrollo Android Studio. La aplicación implementada fue probada en un teléfono inteligente Android versión Lollipop 5.1. La fig. 3 (a) muestra la ventana principal de inicio de la aplicación STUTTY, aquí el usuario debe seleccionar el botón de “INICIO”.

STUTTY cuenta con 5 módulos: 4 de tipo consonante y uno para las vocales (Ver Fig. 3 (b)). Las consonantes oclusivas se colocaron en primer lugar debido a que la

aplicación está enfocada a la tartamudez clónica, donde el problema frecuentemente se presenta en este tipo de consonantes.

La Fig. 4 (a) muestra la lista de palabras a practicar de la letra seleccionada, esta lista es dinámica, cada vez que el usuario entra las palabras cambian de forma aleatoria.

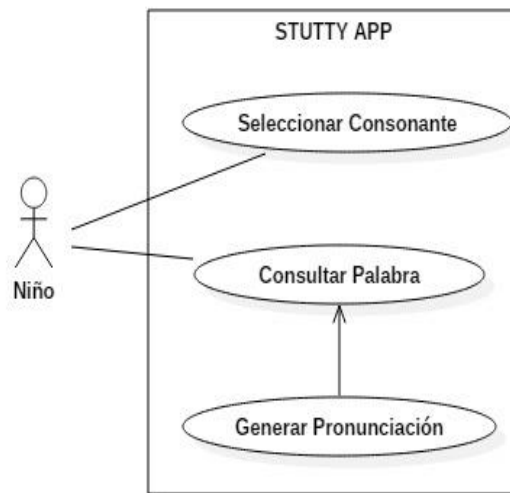


Fig. 2. Diagrama de Casos de uso de la aplicación.

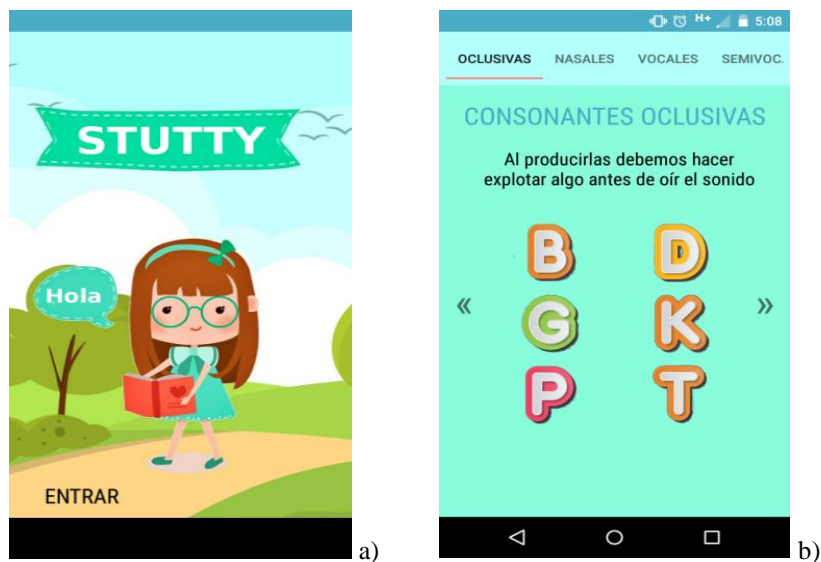


Fig. 3. a) Actividad de inicio de la aplicación STUTTY. b) Actividad para seleccionar el tipo de consonante.

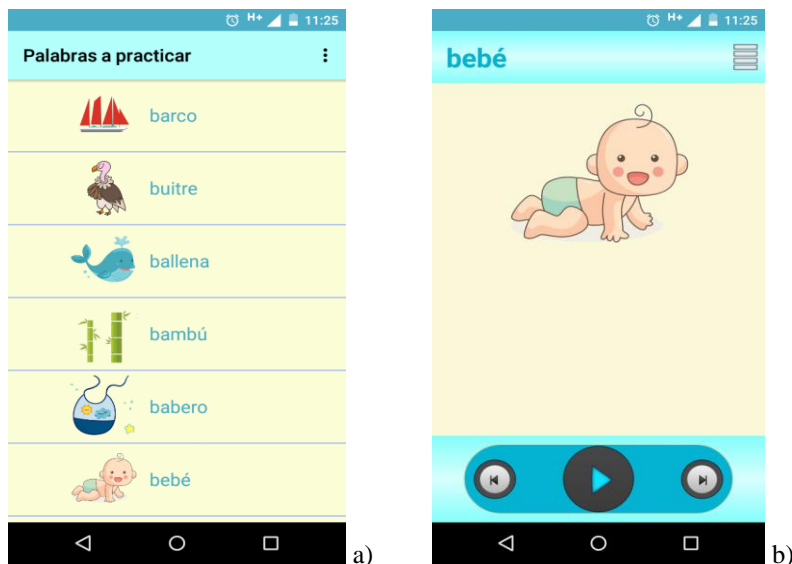


Fig. 4. a) Actividad que muestra la lista de palabras a practicar. b) Actividad para escuchar la pronunciación usando la técnica “tartamudeo-sencillo”.

La aplicación STUTTY permite al usuario tantas veces necesite y quiera escuchar la pronunciación de la palabra, usando la técnica del “tartamudeo-sencillo” (Ver Fig. 4(b)). También podrá ir a la siguiente palabra o regresar a la palabra anterior, además de poder retornar a la lista de palabras para seleccionar otra en particular.

4.5. Pruebas

En las pruebas realizadas para evaluar la fluidez del habla en los usuarios, se mide tomando en cuenta dos variables: frecuencia y cantidad. La variable frecuencia mide cuantas palabras en un texto pronuncia con alteraciones y cantidad mide cuantas veces una palabra, sílaba o sonidos son repetidos.

En la prueba participaron 4 usuarios (niños entre 6 y 11 años), de los cuales uno tiene tartamudez establecida inicial y tres con tartamudez normal. La prueba consistió en leer 10 frases antes de usar la aplicación, un mes después de utilizar la aplicación el usuario volvió a leer otras 10 frases. Los resultados se muestran en la siguiente sección.

5. Resultados

Las pruebas realizadas se describieron en la sección anterior, a continuación se muestran los resultados obtenidos. La Tabla 4 presenta promedios de frecuencia y cantidad al evaluar a cuatro usuarios.

En la Tabla 4 se observa que hubo una pequeña mejoría, está claro que es necesario tiempo y uso contante de la aplicación para tener mejores resultados, como los obtenidos en la práctica empírica al utilizar la técnica “tartamudeo-sencillo” [4].

Tabla 4. Promedios de frecuencia y cantidad sobre 10 frases.

Usuario	Antes		Después	
	Frecuencia	Cantidad	Frecuencia	Cantidad
1	0.13	0.10	0.10	0.09
2	0.12	0.11	0.09	0.09
3	0.12	0.10	0.10	0.10
4	0.20	0.13	0.18	0.12

En las pruebas se ha comprobado que la aplicación ayuda al usuario a realizar su práctica diaria con mayor interés, incluso recuerda que debe practicar.

El resultado de efectividad total del software fue de 73.6%, el cual es un promedio de la efectividad de cada usuario. Durante las pruebas realizadas la utilización de aplicación STUTTY en niños de entre 6 y 11 tiene una aceptación muy favorable, los niños se interesan en seguir practicando, les agrada escuchar la voz y repiten continuamente las palabras como se les indica.

La intervención temprana se realiza al fomentar el uso de la de la aplicación STUTTY, asumiendo que en algunos casos no hace falta.

6. Conclusiones y trabajo futuro

En este trabajo se ha desarrollado una aplicación móvil llamada STUTTY, para ayudar en el tratamiento de tartamudez clónica para niños entre 6 y 11 años, utilizando la técnica “tartamudeo-sencillo”.

La aplicación STUTTY incentiva a que los niños practiquen la técnica “tartamudeo-sencillo”, además esta aplicación puede ser usada en niños aunque no tengan tartamudez, lo cual nos permite realizar una intervención temprana para prevenir futuros problemas.

La aplicación tiene una aceptación muy favorable, los niños se interesan en seguir practicando, les agrada escuchar la pronunciación y repiten la palabra como se les indica.

Como trabajo futuro se pretende agregar tipos de usuario, ya que en estos momentos la aplicación aun no contiene un registro de utilización. También se pretende agregar más actividades para practicar el “tartamudeo-sencillo”.

Por otro lado otros componentes de la fluidez que serán tomados en cuenta en próximas pruebas serán: el ritmo, la velocidad, la prosodia, la secuencia y la fluencia.

Referencias

1. Gutiérrez-Ornia, J.: Aceptando la tartamudez. Universidad de Valladolid (2015)
2. Yairi, E.: Disfluency characteristics of childhood stuttering. *Nature and Treatment of Stuttering: New Directions*, Needham Height, Allyn & Bacon (1997)
3. Groesman, C. P.: Tartamudez, guía para docentes. España: Fundación española de la tartamudez
4. Ann, I.: Cómo vencer el tartamudeo. Mondadori, Grijalbo (2001)
5. Onslow, M.: Stuttering and its Treatment. Eleven Lectures, Sydney, Australian Stuttering Research Centre The University of Sydney (2017)
6. Anderson, J. D., Wagovich, S. A.: Explicit and Implicit Verbal Response Inhibition in Preschool-Age Children Who Stutter. *Journal of Speech, Language, and Hearing Research*, Vol. 60, pp. 836–852 (2017)
7. Sangorrín, J.: Disfemia o tartamudez. *Revista de Neurología* (2005)
8. Kehoe, T. D.: No Miracle Cures, A Multifactorial Guide to Stuttering Therapy (2008)
9. MacMillan, V., Kokolakis, A., Sheedy, S., Packman, A.: End-word dysfluencies in your children: A clinical report. *Folia Phoniatica et Logopaedica*, pp. 115–125 (2014)
10. Gaiolas, M., Cintra, Z., de Ler, C.: Ernest, Stuttering App. 12 julio 2017, [En línea], Disponible en: <https://play.google.com/store/apps/details?id=toymobi.ernesto&hl=en>
11. Institut der Kasseler Stottertherapie: FLUXXY. 12 julio 2017, [En línea], Disponible en: <http://www.kasseler-stottertherapie.de/stottertherapie/smartphone-app-fluxxy/>
12. Cure The Stuttering Free. 12 julio 2017, [En línea], Disponible en: <https://play.google.com/store/apps/details?id=jp.ne.nama.android.curethestutteringfree&hl=en>
13. Stutter Help. 12 julio 2017, [En línea], Available: <https://play.google.com/store/apps/details?id=com.pcp.stutterhelp&hl=en>
14. Stutter Rater. 12 julio 2017, [En línea], Disponible en: <https://play.google.com/store/apps/details?id=com.oatau.stutter&hl=en>
15. Pressman, R. S.: Ingeniería de Software. Un enfoque práctico, México, McGRAW-HILL (2010)

Clasificación de género utilizando medidas antropométricas, análisis de texturas y la arquitectura ANFIS

Cosijopii García García, Nayeli Joaquinita Meléndez Acosta

Universidad del Istmo Campus Ixtpec, Ciudad Ixtpec, Oaxaca,
México

{cosijopiigarcia, nayelimelendez}@gmail.com

Resumen. La automatización del proceso de clasificación de género está justificado por diversas aplicaciones que juegan un rol muy importante en la interacción entre los humanos y la computadora como: motores de búsqueda, recolección de datos demográficos y como parte de frameworks biométricos y psicológicos. En el presente artículo se diseña un método de clasificación de género utilizando imágenes faciales y la arquitectura ANFIS como clasificador, la cual nos ofrece las ventajas de la lógica difusa y las redes neuronales. La metodología propuesta nos llevará por varias fases, en las cuales se usarán diferentes métodos para la extracción de las características del rostro, como son ojos, frente, nariz, boca, barbilla, etc., así como el algoritmo LBP (Local Binary Pattern) que ayudan en la extracción de texturas y el análisis antropométrico. Se realizaron pruebas utilizando 428 imágenes los cuales fueron introducidos al clasificador, y se obtuvo un porcentaje de clasificación del 86.44%.

Palabras clave: ANFIS, imágenes faciales, clasificación, LBP, clasificación de género, análisis de texturas, lógica difusa, FCM.

Gender Classification Using Anthropometric Measurements, Texture Analysis and ANFIS Architecture

Abstract: The automation of the gender classification process is justified by several applications that play a very important role in the interaction between humans and the computer such as: search engines, demographic data collection and as part of biometric and psychological frameworks. This paper designs a method of gender classification using facial images and the ANFIS architecture as a classifier, which offers us the advantages of diffuse logic and neural networks. The proposed methodology will take us through several phases, in which different methods will be used for the extraction of facial characteristics, such as eyes, forehead, nose, mouth, chin, etc., as well as the LBP algorithm that help in the extraction of textures and anthropometric analysis. Tests were

performed using 428 images which were introduced to the classifier, and a grading percentage of 86.44% was obtained.

Keywords: ANFIS, facial images, classification, LBP, gender classification, texture analysis, fuzzy logic, FCM.

1. Introducción

El rostro humano nos provee de la forma más rápida y directa, la posibilidad de evaluar implícitamente información social crítica, por ejemplo, el rostro humano nos puede dar información acerca de la raza, el género, la edad, expresiones e identidad, los cuales son de utilidad para ayudar al proceso de toma de decisiones [1].

La investigación del comportamiento psicológico muestra, que el encuentro con un nuevo individuo frente a un estímulo del rostro humano activa tres evaluaciones neuronales primitivas, la raza, el género y la edad, estos tienen consecuentes para el perceptor y el percibido [1]. Por lo tanto este problema ha inspirado a investigadores que se han dedicado al análisis facial, con énfasis especial en la identificación del rostro, extracción de características, clasificación de género, edad y expresiones [2].

La identificación del género está justificada por diversas aplicaciones en diferentes áreas como: La interacción humano computadora (HCI) y la seguridad [3].

En la indexación de bases de datos de imágenes, optimizando las búsquedas y mejorando el tiempo de consulta, así como reduciendo los espacios de búsqueda en bases de datos muy grandes, también como el ordenamiento de fotos digitales [4].

También se puede extrapolar a áreas como el marketing digital mostrando publicidad a un tipo de género [4, 5].

Tiene aplicaciones en la autenticación de usuarios y como parte de sistemas de seguridad biométricos [6]. Y en la vigilancia y re identificación de sujetos utilizando cámaras CCTV y *Soft Biometrics* [6, 7].

Este artículo está organizado por secciones, la primera explica el trabajo relacionado en la clasificación de género, así como los materiales y la metodología que se utilizó, la sección dos explica como la imagen inicialmente es normalizada y pre procesada para la extracción de características, la sección tres explica cómo se encontraron los puntos del rostro y como se obtuvieron las medidas antropométricas del rostro, así como el uso de LBP para el análisis de texturas, la sección cuatro nos muestra cómo se prepararon los datos y el uso de ANFIS para la clasificación así como el uso del algoritmo FCM para la creación de un modelo difuso el cual ANFIS optimizó, en las secciones cinco y seis nos muestran los resultado de la clasificación y las validaciones que se realizaron al modelo, por ultimo las conclusiones a las que se llegaron con el modelo realizado y las referencias.

1.1. Trabajo relacionado

Una revisión de los métodos de clasificación de género es dada por [8] y más recientemente también por [6]. Algunos puntos relevantes son los siguientes: uno de

los primeros métodos utilizados para la clasificación de género fueron las redes neuronales, esta red fue entrenada con un conjunto de imágenes faciales frontales pequeñas, este pequeño clasificador fue llamado SEXNET [9].

En [10] se utiliza la combinación de la estructura en 3D del rostro humano extraído por un escáner laser y las intensidades de la imagen, esto para clasificar el género. Los clasificadores SVM fueron utilizados por [11] usando directamente las intensidades de la imagen, algo raro en el uso de SVM's [12].

En [13] usa AdaBoost para la clasificación de género aplicando de igual manera las intensidades de la imagen. En años más recientes autores como [3] han usado *Webers Local texture descriptors* para el reconocimiento del género, utilizando la base de imágenes faciales FERET [14].

Los autores [15] proponen un método usando redes neuronales para la clasificación de grupos de edad y género, el cual utiliza 4 grupos de edad, su clasificación está basada en la geometría del rostro dependiendo del género, y en el análisis de textura de la piel, su metodología propuesta se divide en tres pasos principales : preprocesamiento , extracción de características y clasificación.

En [7] proponen un método utilizando un descriptor *Zernike moments*, combinando estos datos con lógica difusa, construyeron un sistema de inferencia difuso. En [12] proponen un método de clasificación utilizando redes convolucionales, entrenan la CNN (*Convolutional Neural Network*) con imágenes de la base de datos *Adience benchmark for age and gender classification*, estos proponen pasar la imagen por varios filtros a través de la red, cada filtro aplica una operación de convolución y la pasa a la siguiente capa de la red.

La arquitectura ANFIS ha sido también utilizada en la clasificación de género, mediante diferentes tipos de métodos como la utilización de la voz [16] y de imágenes de huellas dactilares [17].

1.2. Materiales y métodos

Para el desarrollo de esta investigación se utilizaron imágenes faciales de dos bases de datos, FERET [14] y PAL [18] ya que contienen imágenes de rostros frontales y alineados. Las bases de datos se unieron para formar un total de 1437 imágenes, siendo 704 imágenes faciales femeninas y 733 imágenes faciales masculinas, la base de datos FERET está compuesta por imágenes de tamaño 640×480 píxeles y la base de datos PAL de 192×128 píxeles. Se utilizó la herramienta matemática MATLAB ya que nos provee de herramientas especializadas para el trabajo con redes neuro-difusas en especial la arquitectura ANFIS.

La metodología de desarrollo que se utilizó está basada en los siguientes autores [15], [5] y [2], los cuales proponen una metodología para el problema de la clasificación de edad y género utilizando comúnmente las redes neuronales, en este caso se usó en la fase de clasificación la arquitectura ANFIS [19], la cual es un tipo de red neuro difusa que utiliza las técnicas de lógica difusa y redes neuronales, por lo tanto de la combinación de estas técnicas se espera una herramienta poderosa [20].

A continuación, se muestran las fases y el diagrama donde se observan las diferentes etapas de la metodología que se utilizaron en esta investigación, la metodología está

dividida en 3 secciones principales: preprocesamiento, extracción de características y clasificación (véase Fig. 1)

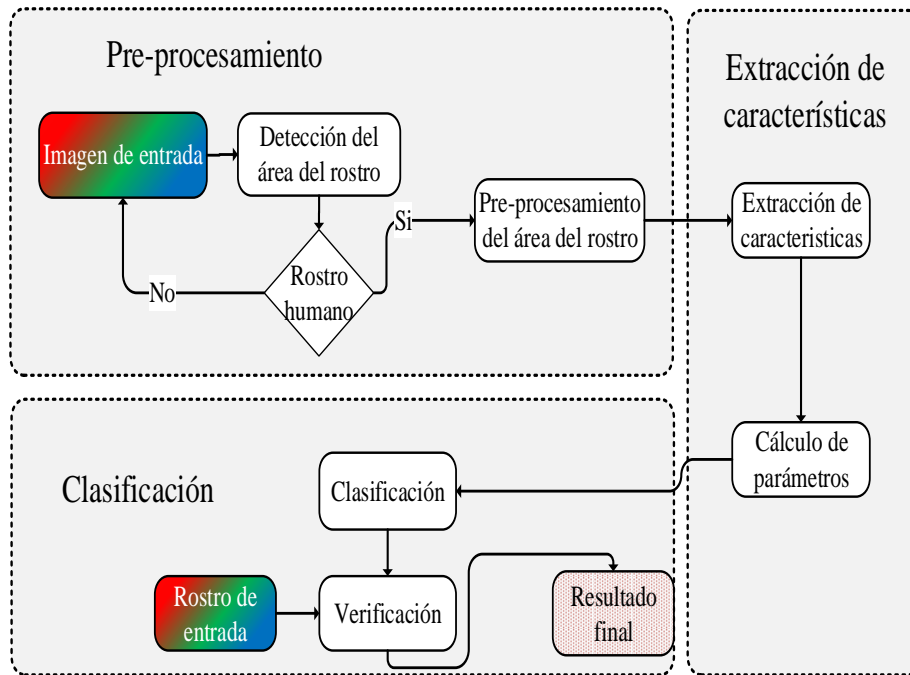


Fig. 1. Etapas de la metodología propuesta para la resolución de la clasificación de género.

2. Preprocesamiento del rostro

La imagen del rostro humano fue normalizada en tamaño, color y aspecto, véase Fig. 2 (a) y (b), esto para que el rostro sea fácilmente encontrado, haciendo que el paso (a) y (b) funcionen con cualquier imagen de tamaño diferente, para esto se utilizó un umbral en el que si el tamaño de la imagen es mayor que 640×480 píxeles, esta se reduce a la mitad y si es menor que esta resolución, se duplique el tamaño de la imagen utilizando un algoritmo de interpolación bilineal, esto para reducir la perdida de información, posteriormente se busca si la imagen contiene una rostro humano utilizando el algoritmo de Viola-Jones [21] véase Fig. 2 (c), para luego realizar el corte del área del rostro de manera automática y posteriormente convertir la imagen a un tamaño predeterminado de 100×100 píxeles, esto con el propósito de normalizar la imagen recortada sin importar el tamaño de la imagen de entrada, esta imagen es la que se utilizará en todos los procesos siguientes, véase Fig. 2 (d).

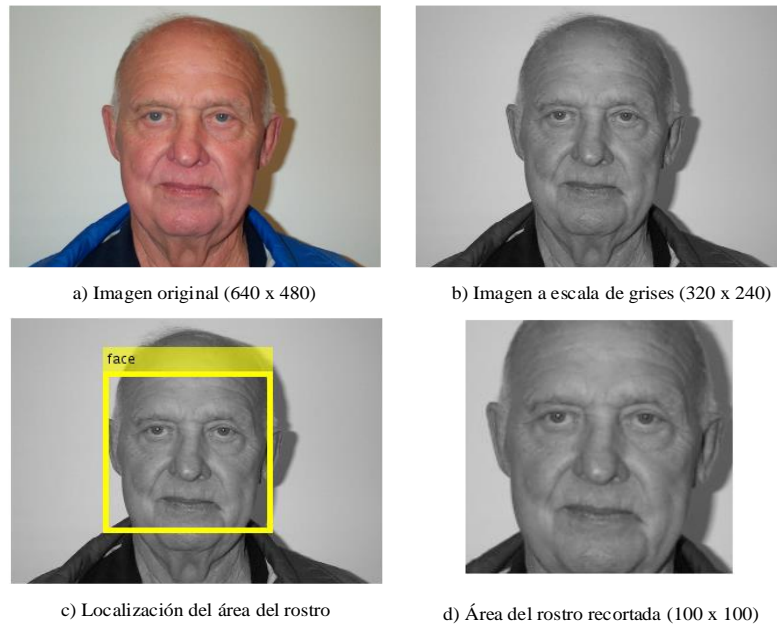


Fig. 2. Preprocesamiento de la imagen facial para encontrar el área del rostro.

3. Extracción de características

La imagen normalizada y convertida a escala de grises es utilizada para ubicar diferentes puntos en el rostro humano, los puntos fueron escogidos siguiendo el estudio de la antropometría del rostro según [22]. A estos le añadimos algunos puntos que son utilizados en la antropometría del rostro de [2], el conjunto completo de puntos se muestra en la Fig. 3.

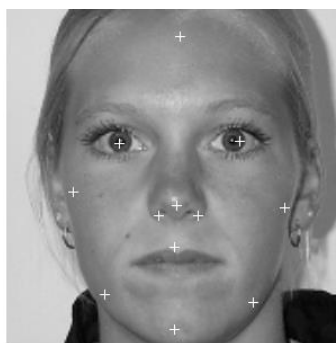


Fig. 3. Puntos del rostro humano seleccionados.

Para ubicar los puntos se usaron dos técnicas diferentes, en la ubicación de los ojos se utilizó un clasificador Viola-Jones, que ya ha sido entrenado con imágenes que contengan a ambos ojos como se ve en la Fig. 4.



Fig. 4. Par de ojos recortados de la imagen facial.

Teniendo la imagen de los ojos se procede a realizar un recorte de cada ojo, estos recortes se hacen de manera automática, siendo el tamaño de la imagen recorta dependiendo del tamaño del área de los ojos, de esta forma se tienen dos imágenes nuevas en las cuales se buscará el iris, siendo este el centro de los ojos. Como primer paso se realizó una ecualización del histograma de cada imagen, posteriormente cada imagen es reducida de todos los lados hacia el centro 35 píxeles, se escogió esta cantidad porque proporciona los mejores resultados, esto se realizó en cada imagen de manera automática, para eliminar el posible ruido alrededor de nuestro objetivo, el iris (véase Fig. 5).



Fig. 5. Ecualización del histograma de ambos ojos, así como el recorte para centrar el iris de cada ojo.

Utilizando estas dos imágenes se procede a encontrar el mínimo local que existe en cada imagen, este siempre tiende a ser el iris por la poca intensidad que contiene los píxeles cercanos [22] (véase en la Fig. 6).

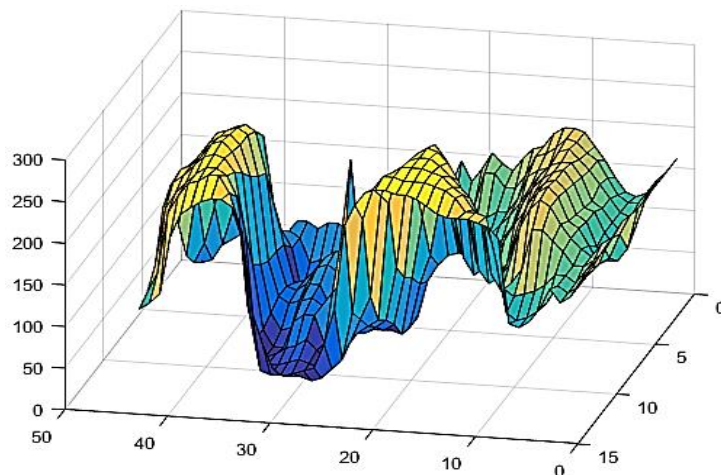


Fig. 6. Representación en 3D de una imagen del ojo derecho se puede observar que las partes más oscuras están ubicadas en el iris del ojo, las partes amarillas corresponden a la esclerótica del ojo.

Teniendo las coordenadas de los ojos se procede a ubicar el centro de la frente, utilizando la distancia que existen entre los ojos mediante la siguiente ecuación:

$$D_{eye} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} . \quad (1)$$

Posteriormente se utilizó una analogía del triángulo equilátero, usando la cúspide del triángulo como el centro de la frente (véase Fig. 7), la altura del triángulo es encontrado utilizando la siguiente ecuación:

$$h = \frac{D_{eye} \sqrt{3}}{2} . \quad (2)$$

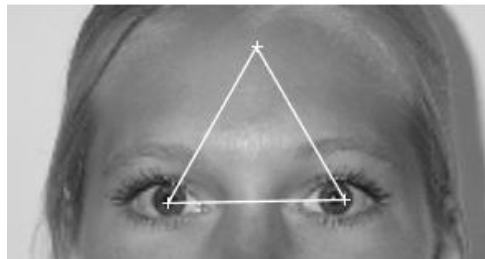


Fig. 7. La distancia que existe entre los ojos, es utilizada como base de un triángulo equilátero y de esta manera se encuentra en la cúspide el centro de la frente.

Para ubicar el resto de los puntos mostrados en la Fig. 3 se utilizó el proyecto OpenFace [23], el cual es un framework que nos proporciona la posibilidad de encontrar en el rostro hasta 68 características, sin embargo, ni la frente, ni el iris de cada ojo son parte de ellas. El proceso por el cual OpenFace encuentra las características es el siguiente: utiliza un preprocesamiento y redes neuronales para obtener una representación de bajo nivel del rostro, el preprocesamiento se divide en dos partes: una transformación para alinear los ojos y la nariz, más la detección de las 68 marcas en el rostro utilizando la librería *dlib's face landmark detector*, posteriormente la imagen es recortada e introducida a una red neuronal la cual fue entrenada con una base de datos privada llamada *FaceNet*, esta base de datos contiene de 100 a 200 millones de imágenes [23].

Tabla 1. Medición entre puntos del rostro.

Característica	Medidas
F1	Distancia de P9 a P11
F2	Distancia de P10 a P12
F3	Distancia de P3 a P7
F4	Distancia de P1 a P12 + Distancia de P12 a P2
F5	Distancia de P8 a P12
F6	Distancia de P4 a P5 + Distancia de P5 a P6
F7	Distancia de P3 a P4+ Distancia de P4 a P5 + Distancia de P5 a P6 + Distancia de P6 a P7

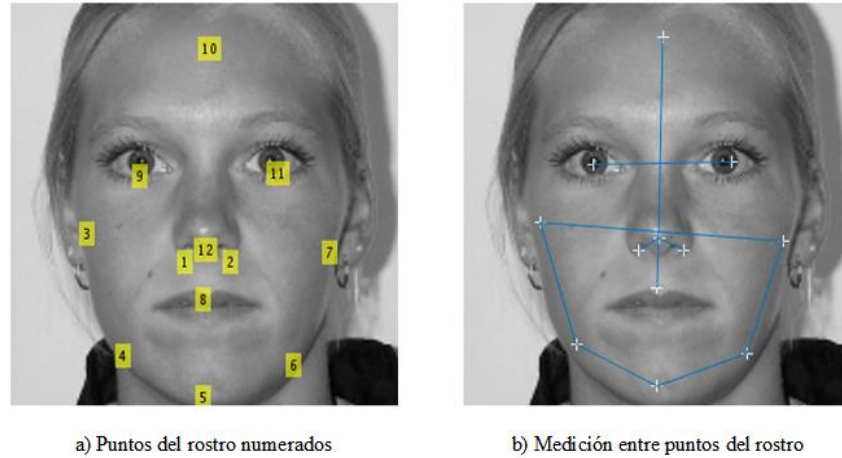


Fig. 8. Puntos del rostro numerados y líneas que definen las distancias entre los puntos.

El ubicar los puntos fue necesario para establecer 7 características las cuales son medidas entre los puntos encontrados en la imagen, las medidas fueron inspiradas en [15], véase Fig. 8 (b), a continuación, en la Tabla 1, se muestran las características y la forma en la que fueron extraídas según la Fig. 8 (a).

Después de extraer las características utilizando las medidas antropométricas, se procede a utilizar LBP, el cual analiza las texturas de la imagen, esto se realiza separando la información de los patrones y del contraste, al hacer esto se obtienen los cambios invariantes en las imágenes a escala de grises [24]. LBP asigna un valor a cada píxel comparándolo con sus píxeles vecinos [25]. El operador LBP es definido como:

$$LPB_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p, \quad s(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0. \end{cases} \quad (3)$$

Donde $g_c = I(x, y)$ es el píxel central y $g_p = I(x_p, y_p)$, $p = 0, \dots, P-1$, es la intensidad de gris de la vecindad P y R del radio alrededor de los puntos (x, y) , los puntos x_p e y_p son calculados de la siguiente manera: $y_p = y - R \sin(2\pi p / P)$ y $x_p = x + R \cos(2\pi p / P)$.

Los patrones producidos por el operador $LPB(p, r)$ describen la textura de la imagen, estos son llamados patrones uniformes, los cuales contienen al menos dos transiciones en el bit 0 o 1 y viceversa, se observó que la mayor parte de la información de textura está contenida en los patrones uniformes. Etiquetar los patrones que tienen más de 2 transiciones con una sola etiqueta produce un operador LBP, denominado $LBP(P, R, u2)$, que produce patrones sin perder demasiada información [26]. LBP dividió la imagen del rostro de 100×100 píxeles en 59 bloques para posteriormente procesar individualmente cada uno, esto nos devuelve el histograma de cada bloque, como resultado final los bloques se concatenan en un solo vector (véase Fig. 9).

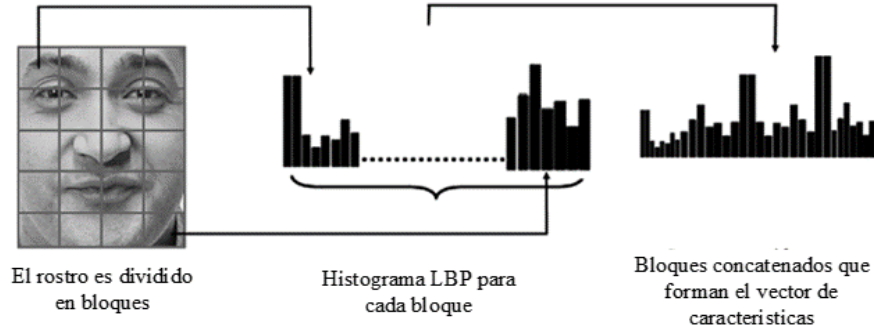


Fig. 9. Proceso por el cual LBP obtiene las características de textura de la imagen facial.

Este vector de 59 características se concatena al primer vector de 7 características teniendo así en total 66 características, el cual será utilizado en la fase de clasificación.

4. Clasificación

En esta fase se utilizó la arquitectura ANFIS (*Adaptive Neuro Fuzzy Inference System*), esta fue propuesta por R. Jang [19] en 1993. ANFIS puede lograr un mapeo altamente no lineal, muy superior a una red neuronal multicapa (MLP, *Multilayer perceptron*) y otros métodos lineales comunes de complejidad similar [27]. ANFIS es una red de cinco capas, véase Fig. 9; ésta requiere como entrada un sistema difuso tipo Sugeno el cual es optimizado mediante la modificación de los parámetros en el proceso de aprendizaje, tal proceso utiliza una combinación de mínimos cuadrados y el algoritmo *Backpropagation*, en otras palabras ANFIS utiliza las ventajas de la lógica difusa y las redes neuronales para ajustar sus parámetros y encontrar la solución óptima [20].

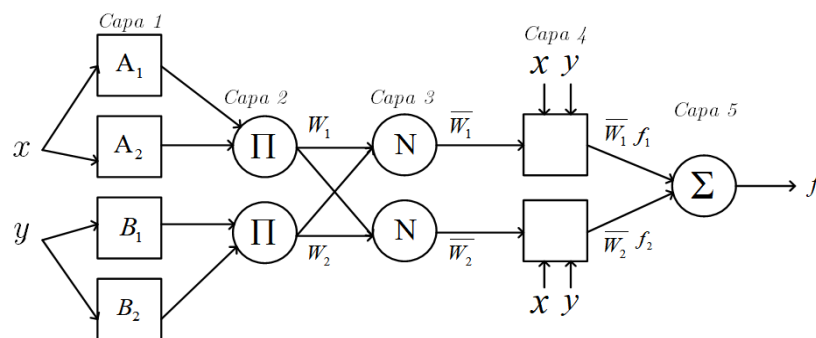


Fig. 10. Arquitectura de ANFIS equivalente a un modelo difuso tipo Sugeno.

Cada capa es especificada de la siguiente manera:

Capa 1. Cada nodo i de esta capa es un nodo adaptativo con una función:

$$O_{1,i} = \mu_{A_i}(x), i = 1, 2, \quad (4)$$

donde x o y es la entrada del nodo i y A_i es una etiqueta lingüística asociada con tal nodo, en otras palabras $O_{1,i}$ es el grado de pertenencia del conjunto difuso

Capa 2. Los nodos de esta capa son nodos fijos representados con el símbolo Π , la salida de los nodos se calcula como el producto de todas las señales de entrada:

$$O_{2,i} = w_i = T(\mu_{A_i}(x)\mu_{B_i}(y)), i = 1, 2, \quad (5)$$

donde la salida es calculada con una norma-T.

Capa 3. En esta capa cada nodo es etiquetado con N , el i -ésimo nodo calcula la relación de la intensidad del disparo de la i -ésima regla con respecto a la suma de las intensidades de disparo de todas las reglas:

$$O_{3,i} = \overline{w_i} = \frac{w_i}{w_1 + w_2}, i = 1, 2. \quad (6)$$

Capa 4. Cada nodo i en esta capa es un nodo adaptado a la función:

$$O_{4,i} = \overline{w_i} f_i = \overline{w_i} (p_i x + q_i y + r_i), \quad (7)$$

donde $\overline{w_i}$ es la salida normalizada de la capa 3 y $\{p_i x + q_i y + r_i\}$ son el conjunto de parámetros que definen a los polinomios Z en este nodo, los parámetros de este nodo son conocidos como parámetros del consecuente.

Capa 5. El único nodo que representa esta capa es denotado por Σ , el cual calcula la salida global como la suma de todas las señales de entrada:

$$O_{5,1} = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}. \quad (8)$$

Y de esta forma se compone una red adaptativa equivalente a un sistema Sugeno de primer orden.

Para crear el sistema Sueno de primer orden que es utilizado por ANFIS, se recurrió al algoritmo Fuzzy C-Means el cual es un algoritmo iterativo que permite que cada dato pertenezca a un clúster con diferentes grados de pertenencia, la función objetivo se define de la siguiente manera:

$$J_m = \sum_{i=1}^D \sum_{j=1}^N \mu_{ij}^m \|x_i - c_j\|^2, \quad (9)$$

donde D es número de datos, N el número de clúster, m es la partición difusa la cual controla el grado de solapamiento difuso $m > 1$. x_i es el i -ésimo dato, c_j es el centro del j -ésimo clúster y por último μ_{ij} es el grado de pertenencia de x_i en el j -ésimo clúster [28]. El algoritmo FCM fue configurado con 8 clústers iniciales y los datos introducidos fueron las 66 características de 1009 imágenes (véase Fig. 11), esto con la finalidad que el algoritmo nos proporcione un FIS (*Fuzzy inference System*) inicial el cual fue optimizado por ANFIS.

67	94	120	25	18	93	191	0.1	0.3	0.9	0.1	0.2	0.5		0.7	0.2	0.4	0.5	0.5	1.3	0.5	0.6	1
61	98	124	28	18	94	192	0.3	0.1	0.2	0.3	0.1	0.7	...	0.2	0.1	0.3	0.3	0.9	0.8	0.2	0.5	0
61	91	116	27	14	99	199	0.9	0.3	0.8	0.8	0.9	0.2		0	1.3	0.4	0.5	0.5	0.6	0.5	0.3	1
⋮													⋮									
63	80	125	24	19	98	208	0.9	0.8	0.3	0.5	0.1	0.6	...	0.1	0.2	0.3	0.5	0.2	0.8	0.9	0.7	0

características extraídas utilizando
medidas antropométricas

Características extraídas utilizando
LBP

Objetivo

Fig. 11. Representación de la matriz de características.

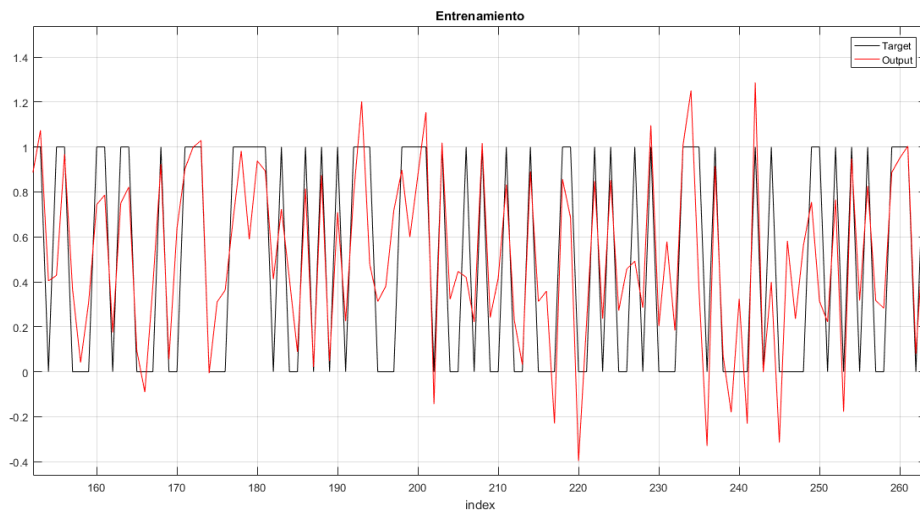


Fig. 12. Aproximación de ANFIS (línea roja) a los datos de entrenamiento (línea negra), se realizó un zoom al gráfico para hacer visible el comportamiento de la aproximación.

5. Resultados

A continuación se muestran los datos del entrenamiento de la arquitectura ANFIS: Se realizaron 100 *epochs* con una matriz de 1009×66 datos de entrenamiento y lo cual dio como resultado un *MSE* (*Median Square Error*) de 0.13551 y *RMSE* (*Root Median Square Error*) de 0.36812. En la Fig. 12 se muestra el comportamiento y la aproximación de los datos de entrenamiento conseguidos por ANFIS.

Para validar el modelo se utilizaron 428 imágenes distintas a las del entrenamiento, de las cuales 204 son de la clase femenina y 224 de la clase masculina. Estos datos fueron introducidos al modelo entrenado previamente. El porcentaje de precisión para

cada clase se puede ver en la Tabla 2 y en la Fig. 13 la matriz de confusión, como se puede observar, la clase masculino demostró una mejor precisión. El promedio de clasificación total fue del 86.44%.

Tabla 2. Rendimiento por clases utilizando la arquitectura ANFIS.

Clase	Porcentaje de clasificación
Femenino	83.33%
Masculino	89.28%

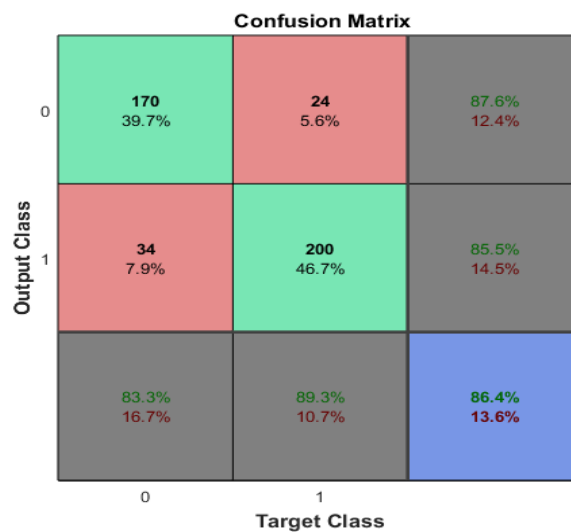


Fig. 13. Matriz de confusión que representa gráficamente los porcentajes de clasificación correcta e incorrecta de cada clase la clase 0 femenino y la clase 1 masculino.

En la tabla 3 se muestra la comparación del método propuesto con otros métodos, en la cual se puede observar un mejor desempeño de nuestro método. Las bases de datos utilizadas por los autores son las siguientes: [7] utilizó 42 imágenes para el entrenamiento, estas imágenes son obtenidas de la base de datos FERET, para realizar las pruebas los autores utilizaron 1199 imágenes de la misma base de datos. [15] utilizó las bases de datos FERET y FGNET, los autores utilizaron 1500 imágenes para realizar el entrenamiento y 200 imágenes para la realización de las pruebas, finalmente [12] utilizó la base de datos facial llamada *Adience benchmark for age and gender classification*, esta base de datos contiene 26000 imágenes, los autores utilizaron la versión que contiene los rostros ya recortados y alineados, el número de imágenes para el entrenamiento fue de 19487 y se utilizaron 6513 imágenes para las pruebas.

Cabe destacar que los autores [7] y [15] utilizaron imágenes en un entorno controlado como las usadas en esta investigación, y en [12] imágenes medianamente controladas, ya que estas son alineadas pero pueden contener ruido como: posturas no frontales y desenfoque en la imagen del rostro.

Tabla 3. Comparación del método propuesto con otros métodos.

Método	Porcentaje de clasificación
<i>P. Moallen y B. Mousavi</i> [7]	85.05%
<i>T. Kalansuriya y A. Dharmaratne</i> , [15]	85.83%
<i>G. Levi y T. Hassner</i> [12]	86.35%
<i>Método propuesto</i>	86.44%

6. Conclusiones

Un nuevo método para la clasificación de género es propuesto en este artículo basado en el análisis antropométrico y de texturas, en el que se consiguió un porcentaje de clasificación del 86.44%. Se realizó el preprocesamiento de la imagen enfocado a la normalización del área del rostro sin importar el tamaño de la imagen, en la sección de extracción de características la combinación de las medidas antropométricas y LBP hasta el momento, han sido muy poco utilizadas específicamente en la clasificación de género. Las imágenes fueron clasificadas en dos grupos, masculino y femenino, la arquitectura ANFIS y el algoritmo FCM fueron utilizados para realizar la clasificación, los resultados experimentales demostraron que usando el algoritmo propuesto la clasificación de género se realiza con mayor o igual precisión que otros métodos.

Referencias

1. Fu, S., He, H., Hou, Z. G.: Learning race from face: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36, pp. 2483–2509 (2014)
2. Colmenares-Guillen, L. E., Velasco-Albores, F. J., Moreno-Ledo, O. G.: Análisis de estudio de casos para estimar la edad basado en detección de rostros en las imágenes digitales [Case study analysis to estimate the age based on face detection digital images]. *Int. J. Innov. Sci. Res.*, 25, pp. 130–140 (2016)
3. Ullah, I., Hussain, M., Muhammad, G., Aboalsamh, H., Bebis, G., Mirza, A. M.: Gender recognition from face images with local WLD descriptor, pp. 11–13 (2012)
4. Grother, P., Ngan, M.: Face Recognition Vendor Test (FRVT) Performance of Face Identification Algorithms. *Natl. Inst. Stand. Technol.* (2014)
5. Saravia, J. A.: Desarrollo de un método de clasificación de edad para imágenes faciales basado en análisis antropométrico y de texturas (2011)
6. Reid, D., Samangoeei, S., Chen, C., Nixon, M., Ross, A.: *Soft Biometrics for Surveillance : An Overview*. Elsevier B.V. (2013)
7. Moallem, P., Mousavi, B. S.: Gender classification by fuzzy inference system. *Int. J. Adv. Robot. Syst.*, 10 (2013)
8. Makinen, E., Raisamo, R.: Evaluation of Gender Classification Methods with Automatically Detected and Aligned Faces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30, pp. 541–547 (2008)
9. Golomb, B. A., Lawrence, D. T., Sejnowski, T. J.: SexNet: A Neural Network Identifies Sex from Human Faces. In: *Proceedings of the 1990 Conference on Advances in Neural Information Processing Systems*, Morgan Kaufmann Publishers Inc., San Francisco, CA,

- USA, 3, pp. 572–577 (1990)
10. Otoole, A. J.: Sex classification is better with three-dimensional head structure than with image intensity information, 26 (1997)
11. Moghaddam, B., Ming-Hsuan, Y.: Learning Gender with Support Faces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24 (2002)
12. Levi, G., Hassner, T.: Age and gender classification using convolutional neural networks. 2015 *IEEE Conf. Comput. Vis. Pattern Recognit. Workshop*, pp. 34–42 (2015)
13. Baluja, S., Rowley, H. A.: Boosting Sex Identification Performance. *Int. J. Comput. Vis.*, 71, pp. 111–119 (2007)
14. NIST: Face Recognition Technology. Disponible en: <https://www.nist.gov/programs-projects/face-recognition-technology-feret>.
15. Kalansuriya, T. R., Dharmaratne, A. T.: Neural Network based Age and Gender Classification for Facial Images. *Int. J. Adv. ICT Emerg. Reg.*, 7, 1 (2015)
16. Lakra, S., Singh, J., Singh, A. K.: Automated pitch-based gender recognition using an adaptive neuro-fuzzy inference system. 2013 *Int. Conf. Intell. Syst. Signal Process*, pp. 82–86 (2013)
17. Sahu, S., Rao, A. P., Mishra, S. T.: Comparison between Neural Network And Adaptive Neuro- Fuzzy Inference System (Anfis) Results In Determination Of Gender Using Fingerprints, pp. 1064–1068 (2015)
18. Minear, M., Park, D. C.: A lifespan database of adult facial stimuli. *Behav. Res. Methods. Instrum. Comput.*, 36, pp. 630–3 (2004)
19. Jang, J. S. R.: ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Trans. Syst. Man Cybern.*, 23, pp. 665–685 (1993)
20. Cortés, E., Algreto-Badillo, I., García, V. H.: Performance Analysis of ANFIS in short term Wind Speed Prediction. *IJCSI Int. J. Comput. Sci. Issues*, 9, pp. 9 (2012)
21. Viola, P., Jones, M. J.: Robust Real-Time Face Detection. *Int. J. Comput. Vis.*, 57, pp. 137–154 (2004)
22. Dehshibi, M. M., Bastanfard, A.: A new algorithm for age recognition from facial images. *Signal Processing*, 90, pp. 2431–2444 (2010)
23. Amos, B., Ludwiczuk, B., Satyanarayanan, M.: OpenFace: A general-purpose face recognition library with mobile applications. (2016)
24. Pietikäinen, M., Hadid, A., Zhao, G., Ahonen, T.: *Computer Vision Using Local Binary Patterns*. Springer-Verlag London, London (2011)
25. Khanmohammadi, S., Ghaemi, S., Samadi, F.: Human Age Group Estimation Based on ANFIS Using HOG LBP Features. *Electr. Electron. Eng. An International J.*, 2, pp. 21–29 (2013)
26. Shan, C.: Learning local binary patterns for gender classification on real-world face images. *Pattern Recognit. Lett.*, 33, pp. 431–437 (2012)
27. Navarro, R. I.: Study of a neural network-based system for stability augmentation of an airplane. (2013)
28. Brain, F. C. M., Segmentation, A., Ibrahim, S., Elaiza, N., Khalid, A., Manaf, M.: Seed-Based Region Growing (SBRG) vs Adaptive Network-Based Inference System (ANFIS) vs Fuzzy, 4, pp. 315–325 (2010)

Selection of Best Software Engineering Practices: A Multi-Criteria Decision Making Approach

Gil Hernández-Ledesma¹, Erik G. Ramos², Carlos A. Fernández-y-Fernández²,
Jorge R. Aguilar-Cisneros³, Juan J. Rosas-Sumano⁴, Luis A. Morales-Ignacio⁴

¹ Universidad Tecnológica de la Mixteca, Ingeniería en Computación, Huajuapán de León,
Oaxaca, México

² Universidad Tecnológica de la Mixteca, Instituto de Computación, Huajuapán de León,
Oaxaca, México

³ Universidad Popular del Estado de Puebla, Decanato de Ingenierías,
Puebla, México

⁴ Clínica de rehabilitación integral física y mental (CRIFYM),
Oaxaca, México

hernandezlg1f@gmail.com, {erik,caff}@mixteco.utm.mx, jorge.aguilar@upaep.mx,
jujorosas@hotmail.com, alfmoring@gmail.com

Abstract. The development of an application based on a set of best software engineering practices involves first their selection. This process may become complex due to the variables involved. This paper presents an Operations Research approach for the selection of best software engineering practices for the development of applications or computational systems considering a series of alternatives and selection criteria. Initially, 31 practices were considered, 19 were selected with 11 criteria using the Multi Criteria Decision Making (or MCDM) method called PAPRIKA (Potentially all pairwise rankings of all possible alternatives). This approach was applied to the development of an application that will help patients with painful hemiplegic shoulder.

Keywords: MCDM, best practices.

1 Introduction

The complexity and diversity of tasks involved in software development have resulted in methodologies that have changed according to the needs of systems and industry, the technologies involved, and scientific advances. Likewise, with the creation of development methodologies, best software engineering practices have emerged, which can be integrated into a methodology or applied jointly, indicating the steps to be taken in development.

In this paper, a MCDM method called PAPRIKA [1] was used for the selection of best software engineering practices for the development of the first component of the Hippocrates project (an application that will help patients with painful hemiplegic shoulder). An evaluation of 31 practices was done and 19 were selected using 11 evaluation criteria: C1) It can be developed efficiently by a single person, C2) Enables agile development, C3) Provides customer interaction, C4) Allows the management of progress indicators, C5) Allows early deployment, C6) Reduces risk, C7) Allows the development of a modular system, C8) Allows work in short times, C9) Allows low cost of fault repair, C10) Allows continuous testing and C11) Development team experience. Each criterion received one of the 3 possible values: bad, regular and good. According to the evaluations, the most important criterion was C1, followed by criteria C9, C5 and C10.

Hippocrates will be a system to help control people's eating habits, emphasizing their physical activity and their rehabilitation. It includes a smart assistant planner with detailed dieting habits for a patient with high cholesterol and triglyceride problems. Hippocrates components will be: A module that works with Microsoft Kinect, which will help patients with painful hemiplegic shoulder. This paper only describes the process of selecting practices for this module. A web module that will help patients keep an accurate record of what they consume and the physical activity they perform thus facilitating and improving the analysis of the nutritionist. This module will be complemented with a component that encourages you to perform physical exercise. A smart assistant which helps create diets for people with cholesterol and triglyceride problems. The remainder of the paper is organized as follows: In section two, three and four important works and concepts related MCDM and some examples of the best practices in software engineering [2] are presented; in section five, important aspects of the selection of the best practices for the development of an application that will help patients with painful hemiplegic shoulder are described; in section six, the proposed practice; in section seven the results of applying a MCDM method to the problem raised in this article. Finally, the section eight report the conclusions and propose some future work.

2 Basic Concepts of Multi-Criteria Decision Making

Multi-Criteria Decision Making (or MCDM) is an area of Operations Research, which is considered a quantitative method (these use numerical information to reach conclusions). MCDM helps to select among several alternatives from the proposed criteria for a given problem. MCDM tries to differentiate between existing alternatives and provides mechanisms to select the solution that best suits your problem. There is a great variety of MCDM methods, which can be classified in deterministic, stochastic, or Fuzzy.

Analytical Hierarchy Process (or AHP), is a method where the decision problems can be modeled with an AHP hierarchy in levels, where the attributes can be objective or subjective. This method compares alternatives through pair-wise comparisons and numerical evaluations [3]. In [4] the authors show the inconsistencies in the AHP

model, so they proposed a new version of the AHP in which the relative value of each alternative was divided by the maximum of these relative values.

Technique for Order Preferences by Similarity to Ideal Solutions (or TOPSIS) is a method that uses the Euclidean distance, has problems with the consistency of the judgments, but has good results if used with the Fuzzy approach [5]. Potentially all pairwise rankings of all possible alternatives (or PAPRIKA) is a patented method and implemented in 1000Minds software. The method obtains the best alternatives through a series of comparisons among them [1].

3 Software Engineering Best Practices

A best practice in software engineering is a tool, language or methodology that represents an improvement in the development of a system or application [2]. Software engineering best practices can be classified according to their context: best practices by type of software; best practices by size of the application; best practices by activity. Some Software Engineering best practices are:

Requirements

- Inspections (requirements): This technique helps to find errors in the requirements [6].
- Product Backlog, Sprint Backlog and User Stories: The Product Backlog is the system requirements, this can change throughout the creation of the product. The Sprint Backlog are the tasks of the workers for the sprint. [7]
- Formal requirements analysis and use cases: The former is based on the creation of requirements documents with a formal notation [8] and “a use case is a unit of functionality expressed as a transaction among actors and the subject” [9].

Analysis and Design

- Unified Modeling Language (or UML) is a language that allows you to specify the elements of a software [9]. There are several UML diagrams, for example: UML class diagrams, UML Object diagrams, UML Interaction diagrams, UML State diagrams and UML Activity diagrams.
- Test-first development: This approach develops first the test cases [10].
- Simple Design: It is a design that lacks unnecessary complex elements [11].
- High-level languages: These kinds of languages have a high abstraction of computer logic.
- Object-oriented (OO) development: The object-oriented paradigm is an abstraction of programming in which everything is seen as objects and methods [2].

Development Organization

- Work organized in Sprints: A Sprint is an iteration in the development of the system or application. [7]
- Work organized in Cascade steps: This is a phase in the software lifecycle.

- Work organized in Spiral loops: A cycle of the Spiral is an iteration in which each of the activities planned for the development of the system are developed over and over again. [12]

Planning

- Planning Poker: This is a technique that uses the numbers of the poker cards to estimate the effort. There is a variation in which the numbers of the Fibonacci sequence are used. [13] [14]
- COCOMO II: This is a mathematical estimation model.

Evaluation and Control

- Gantt and PERT chart: These are charts that model the tasks in a process. [10]
- Burndown chart: This chart shows the estimated time of effort required to complete the project. [7]
- Formal progress reports (weekly): Continuous reports of the development of the system. [2]

Testing

- Formal test plans and templates: This practice uses templates with common test cases for software functions [15].
- Automated unit testing: These are tests that require minimal human intervention. [11]
- Regression test: Check that some improvement did not affect the rest of the program.
- Incremental testing Top-down: Strategy in which, the top components of the application are tested first [16].
- Incremental testing Bottom-up: Strategy in which, the terminal components of the application are tested first. [16]
- Usability test: This practice evaluates the interfaces with real users. [16]

Practices Additionally Considered

- Continuous integration: The code is integrated and tested several times a day.
- On-site customer: The customer is a fundamental part of the work team. [11]
- Prototyping: This consists of the creation of prototypes throughout the development of the system.
- Automated documentation tools: The use of these types of tools saves time and effort for the developers.
- Refactoring: This consists of making improvements to the design of the existing code. [11]

4 Related Work

The MCDM has been used in software engineering for the prioritization of requirements in [17] using the AHP method for this activity. It has also been used for

the selection of a suitable software lifecycle model (SLCM), in this work [18] a fuzzy multi-criteria decision making approach is proposed.

In [19], the authors propose the selection of the best-fit agile software development methodology for small and medium enterprises based on the multi-criteria method SMARTER with a three-point scale. On the other hand, in [20] test techniques are selected using the AHP and TOPSIS methods.

5 Methodology

The application of the PAPRIKA method included in the 1000 minds software was proposed for the selection of best software engineering practices for the creation of projects that do not conform to a software development methodology. The PAPRIKA method was applied for the selection of best software engineering practices for the first component of our Hippocrates project (application that will help patients with painful hemiplegic shoulder). These are the needs that must cover the practices to be selected.

Requirements

- Partial definition of requirements: The requirements do not have to be fully defined before beginning the development of the system.
- Flexibility of requirements: The requirements may change throughout the development of the system.

Control, Evaluation, Effort Estimation and Duration of Tasks

- A method of estimating and controlling tasks is necessary.
- Tools and metrics are needed to visualize the development progress of the application.

Testing

- Test methods are executable by a single person and verify that the changes do not introduce new defects.

Additional Conditions

- No development teams are required: The component to develop is part of an individual project, so a developer will assume all roles.
- Continuous customer feedback: The client must be involved in all the application development because this component will help to provide medical treatment.

In the first place, we hoped that the development would have an agile approach, which explains why some characteristics of this approach were sought and taken. In the Requirements section, a process of requirements such as the Cascade Methodology was considered deficient, since the development process is continuous and new functional requirements can be discovered as tests or prototypes are performed. Consequently,

different practices were chosen that allowed the flexibility of requirements and their partial definition.

For the section control, evaluation, effort estimation and duration of tasks, an effort estimation method was determined necessary, because this would allow the analysis and evaluation of the progress of the development of the system. In the case of testing, it was once again considered the main limitation of the system (this is an individual project), so techniques such as test automation are quite necessary. Hippocrates is a medical system and it was necessary for the clients and users (doctors) to be closely linked with its development, so that they could carry out the necessary medical validation. The practices described in the section "Software Engineering best practices" were evaluated because these fulfilled certain application needs.

The criteria considered were as follows: C1) It can be developed efficiently by a single person, C2) Enables agile development, C3) Provides customer interaction, C4) Allows the management of progress indicators, C5) Allows early deployment, C6) Reduces risk, C7) Allows the development of a modular system, C8) Allows work in short times, C9) Allows low cost of fault repair, C10) Allows continuous testing and C11) Development team experience.

An agile approach to system development was considered, due to its advantages [21]. Criterion C1 was created because this is an individual project and responds to one of the additional conditions already mentioned. Criterion C2 is used for practices with an origin in agile development to obtain a better score. Criterion C3 helped to select practices that encourage constant client intervention in development with the intention of validating the aspects related to medicine in the system. The criteria C4, C6, C7 and C9, respond entirely to the needs of control, evaluation, effort estimation and duration of tasks, since they are criteria that will help to obtain the best practices to diminish risks and to recover from errors in the development.

Table 1. Practice: Selection of best practices for software development using the MCDM method: PAPRIKA.

Alternative	Criteria											Rank	Score
	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10	C 11		
Incremental testing Bottom-up	G	G	G	G	G	G	G	G	G	G	R	1 st	99.8
Incremental testing Top-down	G	G	G	G	G	G	G	G	G	G	R	1 st	99.8
Usability test	R	G	G	G	G	G	G	G	G	G	G	3 rd	94.6
Continuous integration	G	G	-	G	G	G	G	G	G	G	G	4 th	94.4
Planning poker	G	G	-	G	G	G	G	G	G	G	G	4 th	94.4
Planning poker: Fibonacci	G	G	-	G	G	G	G	G	G	G	G	4 th	94.4
Burndown chart	G	G	-	G	G	G	G	G	G	G	G	4 th	94.4
Simple Design	G	G	-	G	G	G	G	G	G	G	R	8 th	94.3
Automated unit testing	G	-	G	G	G	G	G	G	G	G	G	9 th	93.3
Refactoring	G	G	-	G	G	G	G	R	G	G	R	10 th	93.1
Prototyping	R	R	G	G	G	G	G	R	G	G	G	11 th	91.1
PERT chart	G	G	G	G	-	G	G	G	G	G	G	12 th	89.6
Work organized in Sprints	R	G	-	G	G	G	G	G	G	G	G	13 th	89.1
Regression test	R	G	-	G	G	G	G	R	G	G	G	14 th	88.0

Alternative	Criteria											Rank	Score
	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10	C 11		
Product Backlog, Sprint Backlog and User Stories	R	G	G	G	G	G	G	G	G	-	G	15 th	84.3
Gantt chart	G	G	-	G	-	G	G	G	G	G	G	16 th	84.1
Automated documentation tools	G	G	-	R	G	G	G	G	G	-	G	17 th	83.0
UML Activity diagrams	G	R	-	-	G	G	G	G	G	-	G	18 th	72.0
UML class diagrams	G	R	-	-	G	G	G	G	G	-	G	18 th	72.0
UML Interaction diagrams	G	R	-	-	G	G	G	G	G	-	G	18 th	72.0
UML Object diagrams	G	R	-	-	G	G	G	G	G	-	G	18 th	72.0
UML State diagrams	G	R	-	-	G	G	G	G	G	-	G	18 th	72.0
Test-first development	B	R	R	R	G	G	G	R	G	G	B	23 rd	71.9
Formal test plans and templates	B	R	-	G	R	G	G	R	G	G	R	24 th	70.0
On-site customer	-	G	G	-	G	G	-	R	G	G	G	25 th	67.8
Inspections (requirements)	R	R	-	G	B	G	R	R	G	R	B	26 th	67.4
Work organized in Spiral loops	B	B	G	G	G	R	G	R	R	G	B	27 th	62.6
COCOMO II	R	R	-	-	-	G	G	G	G	-	G	28 th	56.3
Formal requirements analysis and Use cases	R	B	G	G	B	R	R	B	G	-	R	29 th	53.9
Formal progress reports (weekly)	B	B	-	G	B	G	-	B	G	B	G	30 th	34.3
Work organized in Cascade steps	B	B	R	G	B	B	G	B	B	B	R	31 st	27.4

Criteria C5 and C8 were thought to encourage the selection of agile practices and the continuous generation of deliverables. Criterion C10 represents the need presented in the testing section. Criterion C11 allowed for the evaluation of the possible learning curve in new practices for the developer. This is important because this factor could cause variation in the development time.

6 Proposed Practice: Selection of Best Practices for Software Development Using the MCDM Method: PAPRIKA

Table 1 defines the methodology as a practice, so that it can be replicated by anyone. The practice was defined following the Essence specification, a Kernel and Language for Software Engineering Methods [22], which is a standard approved by the Object Management Group. In particular, the KUALI-BEH Kernel Extension of Essence was used, which allows for the definition of practices independently of the technology, size and life cycle of the project.

7 Results

In this project, 31 different practices for the development of the Hippocrates application were evaluated. Table 2 shows the alternatives ranked with each of their qualifications for the criteria. Hyphens (-) indicate that the alternative could not be evaluated for this

criterion, the results generated by PAPRIKA were not affected by the above. The values of evaluation of the practices are: Bad (B), Regular (R) and Good (G).

The criterion C1 has the greatest weight according to the preferences detected by the method, the variation of weights for each criterion is shown graphically in Fig. 1a. The best rankings were taken as long as they were not focused on the same objectives or activities in the development of the application. The list of selected practices (grouped by type of activity) is as follows:

- Requirements: Product Backlog, Sprint Backlog and User Stories.
- Analysis and design: UML class diagrams, Simple Design, High-level languages and Object-oriented (OO) development.
- Development organization: Work organized in Sprints.
- Planning: Planning Poker Fibonacci.
- Evaluation and control: Burndown chart.
- Testing: Automated unit testing, Regression test, Incremental testing, Bottom-up and Usability test.
- Practices additionally considered: Continuous integration, On-site customer, Prototyping, Automated documentation tools and Refactoring.

Table 2. Practices evaluation for development organization.

Practice: SelectionBestPracticesForSoftwareDevelopmentUsingPAPRIKA	
Selection of best practices for software development using the MCDM method: PAPRIKA.	
Objective: Select n best practices for software development from a set of practices using the MCDM method called PAPRIKA, considering a set of criteria and needs according to the application.	
Entry. Conditions: The development team will propose a set of m practices that they themselves can develop within which n will be selected.	Result. Conditions: The development team will have a set of best practices that will fit the needs and variables of the project.
Completion Criteria: 75% of the development team members must agree to the selected practices.	
Guide	
Activity 1. Project leader will meet with the customer and discuss the needs of the application which will be listed.	
Input. Conditions: Project leader and customer are in time and place. Output. Work products: Application needs list. Competences: The project leader must have knowledge of requirements engineering. Measures: None.	
Activity 2. Development team will analyze and propose a set of criteria that reflects the application needs and other variables that they consider appropriate, for example: the type and levels of development team experience, the soft and hard skills of the developers, the type of approach to testing, organization of development, customer interaction, type of progress indicators and all those factors relevant to development.	
Input. Work products: Application needs list. Conditions: Development team is in time and place. Output. Work products: Criteria list. Competences. The development team must know or have experience in several methodologies and best practices of software development. Measures: None.	
Activity 3. The team will propose a set of practices that according to their experience could be adjusted to the needs of the project.	

Input. Work products: Application needs list. Conditions: Development team is in time and place. Output. Work products: Practices list. Competences: The development team must know or have experience in several methodologies and best practices of software development. Measures: None.
Activity 4. Give a rating (Good, Regular or Bad) to each of the practices in each of the criteria.
Input. Work products: Practice list. Conditions: Development team is in time and place. Output. Work products: Practices table with grades. Competences: The development team must know or have experience in several methodologies and best practices of software development. Measures: None.
Activity 5. Answer the questions requested by the PAPRIKA method, these will determine the preferences of the development team and reflect the needs of the project.
Input. Work products: Practice table with grades. Output. Work products: Ranking of practices. Competences. None. Measures. None.
Activity 6. Select the best ranked practices.
Input. Work products: Ranking of practices. Output. Work products: List of selected practices. Competences. None. Measures. None.

The practices with the same total score, the practices that originated in agile development (criterion C2) were chosen, due to the advantages of this approach [21]. High-level languages and Object-oriented (OO) development practices were also adopted without evaluation because the development of applications for Kinect V2 uses C#, a high-level language with the OO paradigm.

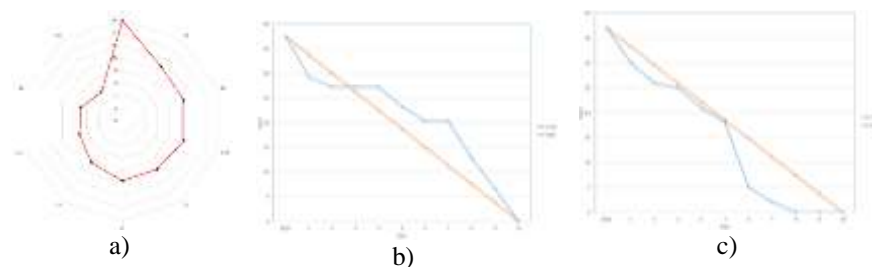


Fig. 1. a) Radar chart of criterion weights. The criterion with greater weight (15.9 %) is C1: The practice can be efficiently developed by a single person. b) Burndown chart sprint 1: It can be observed that for more than half of the sprint, the developer worked slowly but was able to complete in time all the activities planned in the sprint. c) Burndown chart sprint 2: It can be seen that an overestimation of effort was made as the goals of the sprint were fulfilled two days ahead of schedule. The blue line indicates the work to be done and the orange line the remaining ideal hours of effort.

7.1 Case Study: Best Practices Selected and the Development of the Application

The work was organized in sprints; all effort estimates were made using Planning poker Fibonacci. Below is a summary of the activities and practices used in each sprint. For the evaluation of the progress, tables were used in which the effort was documented. From these, the burndown chart of sprint 1 (see Fig. 1b) and sprint 2 (see Fig. 1c) were generated.

Sprint 1: Four meetings were held with the client to create the user stories and the product backlog. The system design was analyzed and later the class diagram and the database were modeled. Three components were detected for the application: one that controls the therapists, another the patients and another the routines through which patients can be given therapy. The technologies with which the system could be implemented were analyzed.

A prototype was developed in C# for usability testing. Fig. 2a shows a screen capture of one of the prototype interfaces. A summary of the results of tests is shown below: Tests were applied to 5 different users: a physiotherapist, a physician, a psychology student with studies in industrial ergonomics, a bachelor's student and a senior citizen. Each user was chosen because their studies or condition would provide good feedback to the system that focused on senior citizens and physicians. The results revealed that: Another method was needed to remove records from patient sections, therapists, and routines; the buttons should provide feedback when the user clicks or positions on them and the silhouette (see Fig. 2b) showing the therapeutic movements to be performed had a low opacity and were difficult to perceive for the senior citizens or patients with some disease related to the hemiplegic shoulder, such as diabetes which can affect the eyes of those who suffer.

Table 3. Equivalence classes for “New Exercise” functionality.

Input Condition	Valid Equivalence Classes	Invalid Equivalence Classes
The repeats are longer than 0 digits and less than 3	length: 1-2 digits (1v)	length < 1 digit(1i), length > 2 digits (2i)
Repetitions are a number	Only has numbers (2v)	Has different characters than numbers (3i)
The degrees have a length longer than 0 digits and less than 4	length: 1-3 digits (3v)	length < 1 digits(4i), length > 3 digits (5i)
Degrees are a number	Only has numbers (4v)	Has different characters than numbers (6i)
The name of the routine is composed of numbers, letters and blanks	only has numbers, letters and blanks (5v)	has characters other than numbers, letters and blanks (7i)
The routine name has a length greater than 1 character and less than 201 characters	length: 1-200 (6v)	length < 1 (8i), length > 200

Sprint 2: In this sprint, the modules of therapists and routines were developed. Incremental testing Bottom-up was used, test cases were developed with Visual Studio software and Coded UI Tests. This tool allowed automated testing and regression tests. The strategy used includes Logic-Coverage Testing, Equivalence Partitioning and Error Guessing; Table 3 shows an example of the equivalence classes for the “New Exercise” functionality. These classes represent subsets of possible input values. The following notation was adopted: the valid classes are indicated as <Number> v, for example: 1v or 3v; while invalid classes are denoted by the letter “i”, <Number> i, for example: 2i or 7i. All the code is in a Bitbucket repository with Git and code was integrated into

this repository daily once it passed the tests. The documentation was developed through Visual Studio when the methods passed the tests. Refactoring was applied to the middle of the sprint and at the end. At this time, the application is still in development. The burndown chart was an effective tool which showed the progress of the sprints and the project. This progress was in accordance with the estimates of Planning poker. Because this practice, the product and sprint backlog, user stories and planning, belong to Scrum, problems of compatibility did not arise.

The creation of test cases with the Incremental Testing Bottom-up and the regression test consumed a large part of the development time despite the use of automated unit testing. The usability tests with the prototype, provide feedback to the development of the project. In this way, errors and peculiarities that could not be described in the product backlog were detected.

With the case study, it was detected that with this approach a “methodology” or framework can be created for the development of an application, selecting the practices that best fit the characteristics of the system to be developed. The application developed in the case study is a medical system and, therefore, a greater validation by the users (doctors and patients) is necessary. This is why it is considered convenient to add a practice like the acceptance testing.

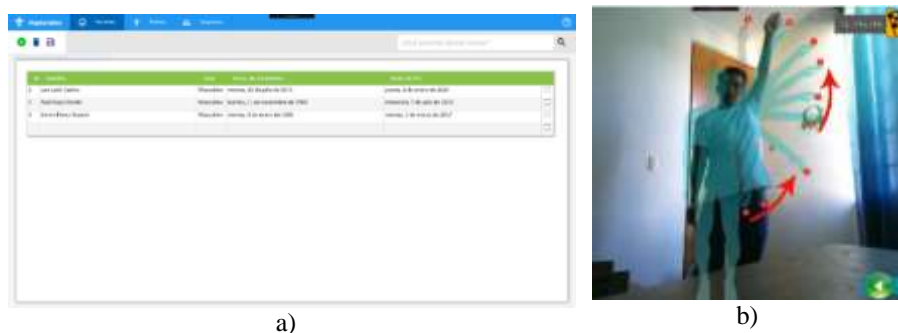


Fig. 2. a) Screenshot of the Patient module of the prototype for usability testing. b) Screenshot of the routine module (physical rehabilitation) in which it is observed that the silhouette indicating the movement is not clearly distinguishable.

8 Conclusion and Future Work

A software development can be affected by factors such as the complexity of the system and the experience and skills of those involved in the project. When using the best software engineering approach, the development may be affected by the number of selected practices and the selection criteria. Particularly if the PAPRIKA method is used, increasing the number of criteria increases the number of decisions that must be entered in the method to generate a result. In the same way, it was determined that the need for the practices be compatible. If this point is not covered, inconsistencies and possible errors will be generated.

This work differs from those already published in that in this case only one person can intervene in the project. Furthermore, different types of practices are considered to be oriented to different activities in the development of a software, not only to the testing as in [20].

If this method is not used, the developer could choose a methodology like PSP. However, works like [23] show that when this methodology is no longer mandatory, the developer leaves it. On the other hand, the developer cannot work with an agile methodology, such as Scrum or eXtreme Programming, which are based on team activities such as: Daily Scrum Meeting or pair programming. However, a methodology such as Lean UX or User-Centered Design, require multidisciplinary teams that collaborate continuously.

This is a low-medium complexity project (took 2 Sprints of two weeks each to develop a quarter of the system). The creation of this system cannot be guided solely by the intuition of the developer, but rather requires a methodology or framework because in its absence, there will be no organization for approximately 4 months of work.

This methodology allows you to select software engineering practices and reduce the possibility of the developer falling into bad practices such as: not documenting code or not fully testing all the features. Furthermore, by using this approach one can adapt or "create" a methodology for unusual projects or constraints like time, money, the experience of developers, as there will not always be ideal projects and clients who are willing to invest money or the time of your staff to perform, for example, continuous iterations of prototypes or usability tests.

The selection of practices from agile methodologies was a success, since these were compatible and adaptable to the main limitation of the system (a developer will assume all the roles), although some practices were not considered for this reason, for example, in a pair programming or Scrum daily meeting. The development times coincided with the estimates, the objectives were reached and the expected products were generated. In general, the practices helped positively in the development. The only drawback was the testing strategy, since the developer did not have any experience in this area, which led to this task consuming more time than expected.

If a Project Leader made this selection of practices, the human factor would be involved. This would be an empirical process in which errors might occur because each project has different needs, different technologies, different work teams with a certain type and levels of experience and hard and soft skills. All of the above factors can result in a large number of variables to be considered, which can be easily represented in a MCDM method.

This article uses an approach to create an application development framework based on the selection of the best practices that adapt to the circumstances of the teamwork, delivery times, the developer's experience, the tools available and all those criteria that the project manager, technical leader or developers consider important. Future work would involve using this approach with a greater amount of initial software engineering practices, developing an analysis of the compatibility of practices and using other methods of MCDM for selection.

References

1. Hansen, P., Ombler, F.: A new method for scoring additive multi-attribute value models using pairwise rankings of alternatives. *Journal of Multi-Criteria Decision Analysis*, 15(3-4), pp. 87–107 (2008)
2. Jones, C.: *Software Engineering Best Practices*. 1st edn., McGraw-Hill, Inc., New York, NY, USA (2010)
3. Majumder, M.: *Impact of urbanization on water shortage in face of climatic aberrations*. 1st edn., Springer Singapore (2015)
4. Belton, V., Gear, T.: On a short-coming of Saaty's method of analytic hierarchies. *Omega* 11(3), pp. 228–230 (1983)
5. Velasquez, M., Hester, P. T.: An analysis of multi-criteria decision making methods. *International Journal of Operations Research*, 10(2), pp. 56–66 (2013)
6. Shull, F., Rus, I., Basili, V.: How perspective-based reading can improve requirements inspections. *Computer*, 33(7), pp. 73–79 (2000)
7. Schwaber, K.: *Agile project management with Scrum*. Microsoft press (2004)
8. Ciancarini, P., Cimato, S., Mascolo, C.: Engineering formal requirements: An analysis and testing method for z documents. *Annals of Software Engineering*, 3(1), pp. 189–219 (1997)
9. Rumbaugh, J., Jacobson, I., Booch, G.: *Unified modeling language reference manual*. 2nd edn., Pearson Higher Education (2004)
10. Lethbridge, T., Laganier, R.: *Object-Oriented Software Engineering: Practical Software Development Using UML and Java*. McGraw-Hill, Inc., 1st edn., New York, NY, USA (2002)
11. Beck, K., Andres, C.: *Extreme Programming Explained: Embrace Change*. 2nd edn., Addison-Wesley Professional (2004)
12. Sommerville, I.: *Software Engineering*. 8th edn., Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (2006)
13. Mahnic, V., Hovelja, T.: On using planning poker for estimating user stories. *Journal of Systems and Software*, 85(9), pp. 2086–2095 (2012)
14. Tamrakar, R., Jorgensen, M.: Does the use of Fibonacci numbers in planning poker affect effort estimates? In: *16th International Conference on Evaluation & Assessment in Software Engineering, EASE 2012, Ciudad Real, Spain, May 14-15, 2012*, pp. 228–232 (2012)
15. Nguyen, H. Q.: *Testing applications on the Web: Test planning for Internet-based systems*. John Wiley & Sons (2001)
16. Myers, G. J., Sandler, C.: *The Art of Software Testing*. John Wiley & Sons (2004)
17. Karlsson, J.: Software requirements prioritizing. In: *Proceedings of the Second International Conference on Requirements Engineering*, pp. 110–116 (1996)
18. Hicdurmaz, M.: A fuzzy multi criteria decision making approach to software life cycle model selection. In: *38th Euromicro Conference on Software Engineering and Advanced Applications*, pp. 384–391 (2012)
19. Silva, V. B. S., Schramm, F., Damasceno, A. C.: A multicriteria approach for selection of agile methodologies in software development projects. In: *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2056–2060 (2016)
20. Victor, M., Upadhyay, N.: *Selection of Software Testing Technique: A Multi Criteria Decision Making Approach*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 453–462 (2011)

21. Petersen, K., Wohlin, C.: A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *Journal of systems and software* 82(9), 1479–1490 (2009)
22. OMG: Essence - kernel and language for software engineering methods. Available at : <http://www.omg.org/spec/Essence> (2015)
23. Johnson, P. M., Kou, H., Agustin, J., Chan, C., Moore, C., Miglani, J., Doane, W. E.: Beyond the personal software process: Metrics collection and analysis for the differently disciplined. In: *Proceedings of the 25th international Conference on Software Engineering*, IEEE Computer Society, pp. 641–646 (2003)

Behavior Modification Applied to Requirements Engineering for Medical Applications

Gil Hernández-Ledesma¹, Noé Hernández-Ledesma², Erik G. Ramos³,
Carlos A. Fernández-y-Fernández³

¹ Universidad Tecnológica de la Mixteca, Ingeniería en Computación, Huajuapán de León,
Oaxaca, Mexico

² Universidad Regional del Sur Este, Licenciatura en Psicología, Oaxaca de Juárez,
Oaxaca, Mexico

³ Universidad Tecnológica de la Mixteca, Instituto de Computación, Huajuapán de León,
Oaxaca, Mexico

hernandezlg1f@gmail.com, herled23@gmail.com, {erik,caff}@mixteco.utm.mx

Abstract. This paper shows a development practice for the requirements process of a medical application. In this process, a psychology specialist was included in order to improve the application from the specification of requirements, considering subjects such as: support networks, behavior modification, increased intrinsic and extrinsic motivation and the creation of tools to facilitate patient-physician communication. This resulted in 11 new requirements. When a psychology specialist is involved in the requirements process, a new perspective on the patient is obtained, one that is broader than the requirements engineer usually has. It considers human needs that can serve to stimulate the user to avoid certain negative behaviors or to encourage positive behaviors. One of the main objectives of an application that helps a patient in their medical treatment is that the user does not abandon it, and the psychological perspective created requirements may help in this.

Keywords: requirements engineering, support networks, intrinsic and extrinsic motivation, behavior modification techniques.

1 Introduction

This paper presents a practice in which a psychological approach was used to develop the requirements process. It was defined with the Essence specification [1], a standard approved by the Object Management Group. This practice was applied to obtain the requirements of a web application (the second component of the Hippocrates system).

The World Health Organization (WHO) indicates that Mexico holds first place in obese and overweight children and second place in adults [2].

According to data from INEGI (*Instituto Nacional de Estadística y Geografía*) during 2014, the primary cause of death in Mexico was heart disease, with a total of 121,427 deaths; diabetes mellitus is the second greatest cause of death with 94,029 deaths, while in sixth place is stroke with 33,166 deaths [3].

Hippocrates will be a system to help control people's feeding habits, and will emphasize their physical activity and their rehabilitation, including a smart assistant planner with detailed dieting habits for a patient with high cholesterol and triglyceride problems.

Hippocrates components will be:

1st component: A module that works with Microsoft Kinect, which will help patients with a painful hemiplegic shoulder.

2nd component: A web module that will help patients keep an accurate record of what they eat and the physical activity they perform thus facilitating and improving the analysis of the nutritionist.

3rd component: The previous module will be complemented with a component that encourages physical exercise.

4th component: A smart assistant that helps create diets for people with cholesterol and triglyceride problems.

To obtain the user stories and product backlog [4], interviews and brainstorms were used with the users and client. According to [5], a first product backlog was obtained, and later a psychology specialist analyzed and suggested changes in the user stories. A meeting was held to analyze and complete the first product backlog with the requirements engineer, the users and the client.

The psychology specialist suggested improvements in the user stories to encourage and increase the patients' affinity to their treatment process, such as:

- **Support networks:** A set of emotionally meaningful people with a sense of community, not necessarily linked to a blood bond that provides social support [6], to facilitate the patient in the therapy process in order to avoid relapses or alienation in treatment, adding meaning to treatment and a feeling of commitment to their health.
- **Tools to facilitate patient-physician communication:** The aim is to improve communication through the implementation of codes shared between patient and physician for the transmission of messages, highlighting the information processed by the patient, helping to understand the treatment and converting the process into something lighter and personalized [7].
- **Behavior modification techniques:** These techniques will be used in order to encourage, reinforce or eliminate certain behaviors in order to reach the target behavior that will have as a consequence adherence to treatment. Among the behavioral principles are: positive reinforcement, generalization and self-control [8] [9].
- **Increased intrinsic and extrinsic motivation:** Based on support networks and behavior modification, it will modify the levels of motivation in the patient, in order to increase the treatment adherence even with a long-term treatment [10].

The consideration and implementation of the elements reviewed above results in an increase in adherence to treatment through the knowledge of their health status, their identification as a patient and the responsibility that this has with itself.

The remainder of this paper is organized as follows: In section two, important works related to the application of psychology in the requirements engineering are presented; Section three presents the development of the requirements process for the second module of the Hippocrates system which will help patients keep an accurate record of what they eat and the physical activity they perform; Section four discusses the proposed practice; In section five, the proposed new requirements are presented by the analysis of the psychology specialist. The final section reports the conclusions and propose some future work.

2 Related Work

2.1 User-Centered Design (UCD)

The UCD is a methodology that considers human factors and ergonomics for the design of computer systems, and has been used in a wide variety of projects: in geo-information technologies [11]; the redesign of an electronic health record with drug-drug interactivity alerts (the system redesigned with UCD improved effectiveness and efficiency) [12]; for the development of an effective tool for rehabilitation using serious games [13]; development of assistive technology for disable and aged with an emphasis on the visually impaired [14]; a prototype for help deaf youths learning English [15].

The latter examples show that the UCD has been used in medical systems as systems like these need to be developed focusing on the needs of the patient or physician (users).

2.2 Requirements Engineering, Human Factors and the Psychology of People

In [16], review the work of programming psychology in five paradigms: 1) individual differences, 2) group behavior, 3) organizational behavior, 4) human factors, and 5) cognitive science, to determine how the programmer behaves when influenced by certain factors. This is done in such a way that seeks, for example, the characteristics necessary for a good programmer.

Ergonomics and human factors is an area closely related to psychology. Authors such as [17] emphasize the importance of human factors in software development because they define this as a technical and human process as development teams generate technological solutions, interact, discuss and work together. A recurrent theme is the design of systems that contemplate phenomena studied intimately by psychology: anxiety and assertiveness [18].

Other works [16] present the importance of the relationship between the development of computational systems and human cognition, generating interdisciplinary branches such as cognitive engineering that considers the modification of environmental demands by modifications in the interaction with technology presented in work environments.

There are papers that evaluate the relationships that are established between the development teams of an application and the users or clients. In [19] sentimental, social and political conditions are analyzed in the requirements process, so that the requirements engineers can handle and interact with the psychology of the people.

In [20] a cognitive engineering technique is used: concept mapping, to convert information provided by users into software requirements. This technique is applied to a case study, namely, through assisting development of a system for the Air Force's Alaskan Air Command.

Similarly, in [21], the relationship between personal values and the requirements of two medical systems through psychology are analyzed.

PERE (Process Engineering in Requirements Engineering) [22] is a tool that analyzes interconnected process components to adapt techniques to the requirements process while conducting an analysis of the human factors of those involved in development.

2.3 Calorie Counter Apps

There are similar systems to the second component of Hippocrates, most of which focus on the calorie counts of users when consuming food and doing exercise. For example, the website Fatsecret [23] in addition to having the two aforementioned functions, it has a catalog of healthy recipes.

This site has some functionalities with the same objective of our application and this paper. For example, Fatsecret has a community in which you can share what you eat. The system also sends you an email to remind you to verify your weight, as aversive reinforcement. Other applications of this type are: MyFitnessPal [24], MyPlate [25] and EasyFit [26].

On the other hand, there are systems completely focused on monitoring your physical activity. For example: Runtastic [27] monitors and measures your training with the GPS of your smart phone when performing activities like: running, cycling or walking. You get statistics such as distance traveled and calories burned. Some other examples of this type of applications are Google Fit [28] and EasyPedometer [29].

3 Methodology

The work presented uses a different approach from that of related work, because in the papers cited, the focus is on the human factors and the psychological background of the requirements engineer and the stakeholders to take advantage of them in the development of the application. Our work considers a part of psychology to create an application that helps modify the behavior of the patient in order to reduce the factors that hinder treatment adherence.

The methodology used is divided into two stages: in the first stage the product backlog was generated after interviewing the users. In this way, the system's functionalities were obtained. In the second stage, the psychology specialist analyzed and began to improve the product backlog with the aforementioned objective.

3.1 First Stage

To obtain the first version of the product backlog, five users were interviewed:

- User #1: The user has used applications similar to the second component of the Hippocrates system, undergone weight reduction treatments with the help of nutritionists and diets and performs daily exercise. Age: 25 years.
- User #2: The user has undergone weight reduction treatments with nutritionists and diets. Age: 42 years.
- User #3: The user has had diabetes for 32 years, and follows a diet with a nutritionist, performs daily exercise and uses insulin. Age: senior citizen.
- User #4: The user has triglyceride problems; he does not have constant advice from a nutritionist. Age: 46 years.
- User #5: The user has reduced his weight only with physical activity. Age: 20 years.

Two techniques were used to obtain the user stories: interviews and brainstorming [5] in three meetings, then the requirements engineer generated the first version of the product backlog. Poker planning was used to calculate the effort of the product backlog. Poker planning was used to calculate the effort to create the application.

3.2 Second Stage: Psychology Intervention in the Requirements Process

The specialist in psychology departed from the general objective of the application specified by the requirements engineer and the product backlog, and began to identify needs and characteristics of the patients, which, in turn, were used to propose requirements that facilitate the development of the skills and behaviors involved in treatment.

The following topics were considered: support networks, motivation and emotions, framing effect (experiences, beliefs, myths, education, sociocultural context), treatment adherence and the human communication process. The specialist was free to suggest new requirements.

Another factor considered was the balance of personal needs involving environmental conditions, family development and social circles in order to optimize and take advantage of these elements to accompany the patient in treatment.

The psychology specialist performs an Applied Behavior Analysis (ABA) considering the Antecedent-Behavior-Consequence (ABC) of the Skinner paradigm [9] to identify possible environmental elements of users, behaviors produced (adaptive or maladaptive) and positive and aversive stimulus.

The effort required for the creation of the application was calculated again considering the new user stories. The engineer, the client, a user and the psychology specialist reviewed this new version of the product backlog. The following section defines the proposed practice.

4 Proposed Practice: Behavior Modification Applied to Requirements Engineering for Medical Applications

Table 1, defines the practice so that it can be replicated. The practice was defined following the Essence specification [1], a standard approved by the Object Management Group. In particular, the KUALI-BEH Kernel Extension of Essence was used, which allows to define practices independently of the technology, size and life cycle of the project to be applied by other practitioners.

5 Results

5.1 Initial Requirements Considering Psychological Factors

The assessment of the psychology specialist found that some of the user stories created by the requirements engineer considered some of the psychological factors mentioned above.

Communication difficulties: patient-physician. Constant and prolonged out-of-office communication.

Advantages. It will increase the feeling of patient safety by continuous communication compared to a non-immediate communication.

Requirements.

- A daily log of the patient's feed will be displayed.
- A daily record of the exercises performed by the patient will be kept, so that they can be added, deleted or edited.
- A daily log of the exercises performed by the patient will be displayed.
- Information such as waist size, weight, cholesterol or glucose of the patient can be added, deleted or edited and statistics will be generated with this information.

Table 1. Practice: Behavior modification applied to requirements engineering for medical applications.

BehaviorModificationToRequirementProcess	Practice
Behavior modification applied to requirements engineering for medical applications.	
Objective. Use behavior modification to create user stories that help decrease the factors that encourage patients to abandon their medical treatment and create a product backlog.	
Entry: Conditions. Users and a psychology specialist should be ready to attend meetings. Result: Work Products. Product backlog. Conditions. The team will agree on possible variations in the effort to develop the application.	
Completion Criteria: The users, the client and the requirements engineer must approve the possible variation in the effort to develop the application.	
Guide	
Activity 1. The requirements engineer meets with users to create user stories and methods such as interviews and brainstorming are used to obtain an initial version of the product backlog.	
Input: Conditions. Requirements engineer and users are in time and place. Output: Work products. First product backlog.	

Competences: The person who creates the product backlog must have knowledge of requirements engineering. Measures: None
Activity 2. Poker planning is used to calculate the effort of the user stories of the first product backlog.
Input: Work products: First product backlog. Output: Work products: First product backlog with effort.
Competences: The person who creates the product backlog must have knowledge of poker planning. Measures: None
Activity 3. Contact psychology specialist with training in general psychology and specific knowledge of behaviorism and theory of motivation.
Input: None. Output: None
Competences: Psychology specialist must have knowledge of behaviorism and theory of motivation. Measures: None
Activity 4. The psychology specialist reviews the first version of the product backlog considering the objective of the system, expected behavioral modification (increase of target behaviors and decrease of maladaptive behaviors) and factors involved in the expected response.
Input: Work products: First product backlog. Output: None
Competences: None. Measures: None
Activity 5. The psychology specialist performs an Applied Behavior Analysis (ABA) considering the Antecedent-Behavior-Consequence (ABC) of Skinner paradigm to identify possible environmental elements of users, behaviors produced (adaptive or maladaptive) and positive and aversive stimulus, from this, new requirements are obtained to try improve the system.
Input: Work products: First product backlog. Output: Work products: Second product backlog.
Competences: The psychology specialist must have knowledge of user stories. Measures: None
Activity 6. Poker planning is used again to calculate the effort of the new user stories of this version of the product backlog.
Input: Work products: Second product backlog. Output: Work products: Second product backlog with effort.
Competences: The person who creates the product backlog must have knowledge of poker planning. Measures: None
Activity 7. The users, the client and the requirements engineer meet to analyze (approve and discard) the user stories proposed by the psychology specialist according to the increase of the time of development.
Input: Work products: Second product backlog with effort. Output: Work products: Second product backlog with effort and reviewed.
Competences: The person who creates the product backlog must have knowledge of requirements engineering. Measures: None

5.2 Requirements Suggested by the Psychology Specialist

Low or Middle Levels of Education Attained

Background. In studies related to treatment adherence, low or middle levels of education attained has been identified as a factor related to the difficulty of understanding the physician's instructions, awareness of their health status, general consequences of treatment failure, communication problems and difficulty gaining insight (awareness) [30].

- Suggestion #1. Educate patients through multimedia (videos, texts, books, magazines, forums) of topics related to their illness.

- Advantages: The system develops awareness of the benefits and consequences of patient treatment and illness.
- New requirements:
 - Show videos and informative articles on the patient's home page related to the user's treatments and diseases.

Lack of Awareness of the Disease

Background. Another factor reviewed in [30] is ignorance on issues related to illness and health status, showing that concern about the repercussions on their daily lives is a factor that increases the index of treatment adherence.

- Suggestion #2. Elimination of myths and false beliefs held about the disease.
 - Advantages: Awareness is generated about the etiology of the disease and the mechanisms to address it in the right way.
 - New requirements:
 - Show videos and informative articles on the patient's home page related to the user's treatments and diseases.
 - The physician can add information to a FAQ section with general patient illness information.
- Suggestion #3. Understanding treatment and prognosis: Explanation of the function of medications, dose ratio and times of progress or forecasts.
 - Advantages: Awareness is generated about the treatment of the patient and their influence on their future.
 - New requirements:
 - The physician may add information related to the medicines commonly provided, such as drug reactions, contraindications and therapeutic use, this information will be shown to patients.

Difficulty understanding the physician

Background. Theory of human communication as a process where it is considered a symmetry and an agreement about the language, code, content and auxiliaries used to improve communication.

- Suggestion #4. Charts for the interpretation of progress.
 - Advantages. Improvement in the understanding of the patient through taking advantage of the ease of interpreting a figure versus a number.
 - New requirements:
 - Charts of calories consumed and weight forecasts.
 - Charts of patient conflation and morphology.

Support from Other Patients

Background. According to the thanatology (set of knowledge based on the significant losses of a person, object or status and the process of mourning subsequent to this), the support networks are identified as a determining factor for the continuation of daily activities, in this case, the treatment.

- Suggestion #5. Linkage with social networks among patients with the same disease to eliminate the low expectation of improvement.
 - Advantages. Creation of support networks, generation of company feeling and commitment to treat the disease.
 - New requirements:
 - Integration with the Facebook API.

Professional Support

Background. In this section, the same background of the sections was used: lack of awareness of the disease and low or middle levels of education attained [30].

- Suggestion #6. Schedule of medical appointments and treatment by videoconference.
 - Advantages. The patient can receive care from the physician anywhere with an internet connection.
 - New requirements.
 - Functionality that allows scheduling appointments with the physician through the system.
- Suggestion #7. Network of experts: Creation of a network of medical specialists for the treatment of the patient.
 - Advantages. Physicians will improve patient care, as treatment will be complemented by the care of more specialists.
 - New requirements.
 - Specialists from different areas can join to the system.

Other Support Features

- Suggestion #8. A calendar.
 - Advantages. This is a tool that will help in the constancy of the treatment.
 - New requirements.
 - Calendar that helps the patient to remember upcoming events related to their treatment.
- Suggestion #9. A reward system.
 - Advantages. Helps maintain patient behavior because rewards are sequential and similar to activities.
 - New requirements.
 - Sequential achievement system, in which the patient is rewarded with badges. Messages with positive content (For example: Well done, you have been working for three days!) or maladaptive content (For example: You have not exercised for a week, right?).

Time estimates were obtained through Scrum planning poker to calculate the effort of the product backlog before and after the intervention of the psychology specialist. In the initial version of the product backlog, the effort was 208 hours, while in the new version the effort was 233 hours, which represents a 25-hour increase (12.01% of initial hours of effort). Because the new functionalities required little time and the gain in

terms of functionality is significant, it was decided to include all the new requirements in the system.

The requirements defined in this paper are not definitive and will be adapted when the application is developed (agile approach).

6 Conclusions and Future Work

This work shows a new practice that unifies two areas: the psychology and engineering requirements for use in medical applications. When an expert in psychology was involved in the process of requirements, a product backlog was generated and considered aspects of users ignored by the requirements engineer in relation to support networks, behavior modification techniques, and increases of intrinsic and extrinsic motivation.

With a psychology specialist, the user is seen from a broader perspective in terms of the understanding of human needs. In this way, this approach considers factors involved in activating the cognitive processes necessary for any interactive activity. The specialist involved not only provides knowledge regarding behavioral and observable issues, he also provides the opportunity for the prediction of possible responses to certain stimuli offered by the requirements engineers before designing such requirements, an alternative to saving effort.

The requirements can be universalized or adapted for a specific population or culture giving a variety in the breadth or reach in the system. From this new specification, an application will be designed that will offer functionalities that will meet the operational needs of the users, while taking into consideration psychological factors involved in the treatment, which can result in health benefits.

Likewise, user stories were obtained to establish channels of communication and understanding between the patient and the physician. The goal of an application is to solve a problem or automate a process. In the case of a medical application that will help patients follow a medical treatment, one of the main objectives is that the user does not leave this treatment. According to the above, the application must be designed to take this into consideration. The development time added in this case is insignificant when considering the user stories added.

Future work would involve the design and development of this component of the Hippocrates system using the requirements expressed in this paper, the implementation of this approach for the design and development of software for other areas, the inclusion and design of systems considering patients with psychological illnesses, such as depression and a more complete validation.

References

1. OMG: Essence kernel and language for software engineering methods. <http://www.omg.org/spec/Essence> (2015)
2. Gutiérrez-Pulido, H., Mariscal-González, M., Almanzor-García, P., Ayala-Dávila, M., Gamba-Hernández, V., Lara-Garzón, G.: Diez problemas de la población de Jalisco: Una

- perspectiva socio demográfica. 1st edn., Gobierno de Jalisco, Secretaría General de Gobierno, Guadalajara Jalisco México (2011)
3. INEGI: Principales causas de mortalidad por residencia habitual, grupos de edad y sexo del fallecido. Available at: <http://www.inegi.org.mx/est/contenidos/proyectos/registros/vitales/mortalidad/tabulados/ConsultaMortalidad.asp> (2014)
 4. Schwaber, K.: Agile project management with Scrum. 1st edn., Microsoft press (2004)
 5. Lethbridge, T., Laganier, R.: Object-Oriented Software Engineering: Practical Software Development Using UML and Java. 1st edn., McGraw-Hill, Inc., New York, NY, USA (2002)
 6. Anaya, N. C.: Diccionario de psicología. Ecoe Ediciones (2004)
 7. Watzlawick, P., Beavin, H., Jackson, D. D.: Teoría de la comunicación humana: interacciones, patologías y paradojas. 1st edn., Herder, Barcelona (1974)
 8. Martín, G., Pear, J.: Modificación de conducta ¿Qué es y cómo aplicarla? 5th edn., Prentice Hall, España (1996)
 9. Stumphauzer, J. S.: Terapia conductual: manual de entrenamiento para psicólogos, médicos, enfermeras, trabajadores sociales, pedagogos, maestros. Trillas (1983)
 10. Marshall, J.: Motivación y emoción. 5th edn., McGraw-hill Interamericana (2010)
 11. Fuhrmann, S.: User-centered design for geoinformation technologies. Comprehensive Geographic Information Systems, Elsevier, Oxford (2017)
 12. Luna, D. R., Lede, D. A. R., Otero, C. M., Risk, M. R., de Quirós, F. G. B.: User-centered design improves the usability of drug-drug interaction alerts: Experimental comparison of interfaces. *Journal of Biomedical Informatics*, 66, pp. 204–213 (2017)
 13. Elnaggar, A., Reichardt, D.: Digitizing the Hand Rehabilitation Using Serious Games Methodology with User-Centered Design Approach. In: 2016 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 13–22 (2016)
 14. Nayak, S. K., Chavan, N. S., Srinath, N.: User centered inclusive design for assistive technology. In: 2016 IEEE Annual India Conference (INDICON), pp. 1–6 (2016)
 15. Anindhita, V., Lestari, D. P.: Designing interaction for deaf youths by using user-centered design approach. In: 2016 International Conference on Advanced Informatics: Concepts, Theory and Application (ICAICTA), pp. 1–6 (2016)
 16. Helander, M. G.: Handbook of human-computer interaction. Elsevier (2014)
 17. Capretz, L. F.: Bringing the human factor to software engineering. *IEEE software*, 31(2), pp. 104–104 (2014)
 18. Gannon, J. D.: Human factors in software engineering. *Computer*, 12, pp. 6–7 (1979)
 19. Sutcliffe, A., Thew, S.: Analysing "people" problems in requirements engineering. In: 2010 ACM/IEEE 32nd International Conference on Software Engineering, 2, pp. 469–470 (2010)
 20. Lind, S., Marshak, W.: Cognitive engineering computer interfaces. I. Knowledge acquisition in the design process. In: Aerospace and Electronics Conference, NAECON, Proceedings of the IEEE 1994 National, 2, pp. 753–755 (1994)
 21. Proynova, R., Paech, B., Koch, S. H., Wicht, A., Wetter, T.: Investigating the influence of personal values on requirements for health care information systems. In: Proceedings of the 3rd Workshop on Software Engineering in Health Care, SEHC '11, ACM, New York, NY, USA, pp. 48–55 (2011)
 22. Viller, S. A.: Human factors in requirements engineering: a method for improving requirements processes for the development of dependable systems. Ph.D. thesis, Lancaster University, UK, available at: <http://eprints.lancs.ac.uk/11686> (1999)
 23. FatSecret: Fatsecret. Available at: <https://www.fatsecret.com.mx> (2017)
 24. MyFitnessPal: Myfitnesspal. Available at: <https://www.myfitnesspal.es> (2017)

25. Leaf-Group-Ltd: Myplate. Available at: <http://www.livestrong.com/myplate> (2017)
26. Hanna, M.: Easy fit. Available at: <http://easyfit-caloriecounter.de/index.html> (2017)
27. Runtastic GmbH: Runtastic.com tu comunidad de salud y fitness. Available at: <https://www.runtastic.com/es/> (2017)
28. Alphabet-Inc.: Google fit. Available at: <https://www.google.com/fit/> (2017)
29. Hanna, M.: Contador de pasos - podómetro. Available at: <https://play.google.com/store/apps/details?id=easypedometer.herzberg.com.stepcounter> (2017)
30. Pfizer: La adherencia al tratamiento: Cumplimiento y constancia para mejorar la calidad de vida (2009)

Métricas estructurales para evaluar la similitud de texturas naturales

Alma Alheli Pedro Pérez¹, Raúl Pinto Elias², Jasiel Hassan Toscano Martinez¹

¹ Universidad de la Sierra Juárez, Oaxaca,
México

² Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca Morelos,
México

almaalheli@unsij.edu.mx, toscano@unsij.edu.mx, rpinto@cenidet.edu.mx

Resumen. En este artículo se evalúa la similitud de imágenes de texturas naturales con cambios de escala y rotación, mediante la aplicación de métricas estructurales. Las métricas aplicadas son: *Complex-Wavelet Structural Similarity Metric* (CW-SSIM) y *Structural Texture Similarity Metric* (STSIM). Primero se realizó una descomposición de las imágenes en sub-bandas. Con la información generada se extrajeron los descriptores y se aplicaron las fórmulas de comparación y las estrategias de unión para obtener un valor de similitud total. En los experimentos con imágenes de texturas con cambios por rotación, las métricas CW-SSIM y STSIM reportan valores promedio de desempeño de: 0.4857 y 0.9735 respectivamente. Con respecto a las pruebas con imágenes de texturas con cambios de escala, las métricas CW-SSIM y STSIM reportan valores promedio de: 0.7002 y 0.9840. Con base en los resultados se determinó que la métrica STSIM es la más recomendable para valorar la similitud de imágenes de texturas.

Palabras clave: Similitud de imágenes, métrica de similitud, texturas naturales, escala, rotación.

Structural Metrics for Evaluating Natural Textures Similarity

Abstract: This article evaluates the similarity of images of natural textures with changes of scale and rotation, through the application of structural metrics. The applied metrics are: *Complex-Wavelet Structural Similarity Metric* (CW-SSIM) and *Structural Texture Similarity Metric* (STSIM). First a decomposition of the images in sub-bands was realized. With the generated information the descriptors were extracted and the comparison formulas were applied and the strategies of union to obtain a value of total similarity. In experiments with texture images with rotation changes, the CW-SSIM and STSIM metrics showed average values of: 0.4857 and 0.9735 respectively. With respect to texture image tests with scale changes, the CW-SSIM and STSIM metrics showed average values of: 0.7002

and 0.9840. Based on the results it was determined that the STSIM metric is the most advisable to evaluate the similarity of texture images.

Keywords: Images similarity, similarity metric, natural textures, scale, rotation.

1. Introducción

En el campo de la visión artificial existen aplicaciones que requieren evaluar la similitud entre imágenes de texturas para realizar tareas de clasificación, recuperación de imágenes con base en el contenido, compresión de imágenes, entre otras aplicaciones [1]. Estas aplicaciones usan métricas o funciones que indican en forma numérica qué tan similar o disímil es una imagen en comparación con otras. En el caso de la clasificación y recuperación, la comparación se realiza entre imágenes contenidas en una misma base de datos [2]. Por ejemplo, existen sistemas médicos que realizan la recuperación de imágenes tomográficas y las clasifican de acuerdo a su modelo de enfermedad [3].

En cuanto a la compresión, se compara la imagen original contra la imagen comprimida [2]. Un ejemplo claro, se observa con el aumento de la cantidad de imágenes tomográficas en un hospital. El aumento de los datos genera nuevas formas de archivar la información, por esta razón, se aplica la compresión de las imágenes para reducir la cantidad de datos. Sin embargo, al comprimir la imagen, se corre el riesgo de deteriorar la calidad. Es por ello, que se requieren métricas de similitud que evalúen el grado de pérdida de información de una imagen a diferentes grados de compresión, para asegurar que la imagen mantenga la calidad necesaria [4].

En la literatura existen investigaciones que abordan la temática de las métricas de semejanza para texturas [5], [6], éstas demuestran un alto nivel de eficacia, en el tema de las métricas desde el punto de vista objetivo y subjetivo, analizan diversas características en función de la estructura de la imagen. Cabe mencionar que estos trabajos se concretan en la clasificación y recuperación de imágenes [7], tareas en las cuales es de suma importancia describir objetos o regiones presentes en las imágenes empleando la textura [8].

Las texturas naturales suelen compartir características similares lo cual provoca que una imagen sea catalogada como de otra clase y no a la que pertenece, es el caso de la textura “árbol” que es semejante a la del “pasto”. Además de estos inconvenientes inherentes a las texturas naturales, existen otros aspectos no controlados, como cambios de escala, rotación, traslación, cambios de contraste e intensidad, entre otros [8]. Estos aspectos se presentan de forma prominente en las texturas naturales porque están constituidas por patrones espaciales cuasi-aleatorios difíciles de preservar ante diversos cambios. Para mejorar algunas de las tareas en el campo de la visión artificial tales como la clasificación de imágenes, es importante contar con métricas de semejanza para la valoración de texturas naturales que consideren las invariancias a diversos cambios que se presentan en las imágenes.

Este trabajo está estructurado de la siguiente manera: en la sección 2 se mencionan algunos trabajos relacionados al tema propuesto, en la sección 3 se describen de forma breve las métricas de similitud utilizadas, en la sección 4 se describe la metodología de la propuesta de solución, en la sección 5 se presentan los resultados obtenidos y finalmente en la sección 6 se presentan las conclusiones derivadas de este trabajo.

2. Trabajos relacionados

Las métricas de similitud de imágenes se clasifican de acuerdo al enfoque de evaluación en tradicionales y estructurales. Las métricas tradicionales evalúan la similitud entre dos imágenes en una base punto a punto. Estas usan modelos explícitos de bajo nivel de la percepción humana y se aplican en el dominio espacial o en el dominio de la frecuencia. Las métricas estructurales utilizan información estructural para realizar comparaciones de la imágenes [6].

En [9] Zhou implementa la métrica tradicional MSE mostrando sus ventajas y desventajas. Esta métrica ha sido criticada por graves deficiencias cuando se emplea en la evaluación de imágenes. El mal rendimiento de la métrica MSE se debe a que emplea mediciones punto a punto y no considera los aspectos que causan distorsiones en las imágenes. Debido al mal rendimiento de la métrica, surgen alternativas interesantes como las métricas SSIM (Structural Similarity Metric) y CW-SSIM (Complex-Wavelet Structural Similarity Metric).

Zujovic [5] realiza una evaluación del rendimiento de las métricas de similitud de texturas aplicadas a la compresión de imágenes sin pérdida estructural. Para realizar la evaluación, se generaron distorsiones en las texturas, estas modelaron las variaciones que ocurren en las texturas naturales. Se evaluaron métricas tradicionales como PSNR, así como métricas con un enfoque estructural como: SSIM, CW-SSIM, STSIM (Structural Texture Similarity Metric) y STSIM-2 Structural Texture Similarity Metric-2). De acuerdo a la experimentación se observó que las métricas estructurales son las más adecuadas para evaluar la similitud de imágenes de texturas sobre todo cuando se trabaja con imágenes de texturas con diversas rotaciones y deformaciones.

Finalmente en [1], Pappas presenta algunas métricas de semejanza de texturas y sus aplicaciones haciendo énfasis en la compresión de imágenes y en la recuperación basada en el contenido, en particular de imágenes de texturas naturales [10]. Expone el uso de métricas tradicionales como PSNR (Peak Signal-to-Noise Ratio) y métricas que incorporan el estudio de la estructura de la imagen, por ejemplo la métrica SSIM. Esta sirve como base para desarrollar la métrica STSIM. En la experimentación las métricas estructurales presentaron un mejor rendimiento en la recuperación de imágenes.

3. Métricas de similitud estructural

Una métrica de similitud estructural incorpora información estructural en las comparaciones de la imagen. Dicha información se basa en estadísticas locales extraídas de las imágenes o de las sub-bandas de estas imágenes [6]. En el estudio de la semejanza de imágenes, las métricas de similitud por lo regular se aplican al análisis de la textura.

Una de las cualidades más importantes de las métricas de similitud estructural, es su invariancia a cambios no estructurales, como el contraste y la intensidad. Así mismo, invariantes a pequeñas traslaciones, rotaciones y cambios de escala que son detectables pero que no afectan la calidad percibida de la imagen. Estas métricas se aplican en el dominio espacial o de la frecuencia. Cuando se implementan en el dominio espacial se obtiene invariancia a cambios de luminancia y contraste. Cuando se implementan en el dominio de la frecuencia son tolerantes a pequeños desplazamientos espaciales,

rotaciones o cambios de escala [6]. A continuación se describen cada uno de los elementos que componen a las métricas CW-SSIM y STSIM.

3.1. Métrica de semejanza estructural (CW-SSIM)

La métrica CW-SSIM (Complex-Wavelet Structural Similarity Metric) requiere que la imagen X y Y a comparar sean descompuestas en sub-bandas. Esto se realiza aplicando una pirámide orientable con determinado número de orientaciones y escalas [11]. Los descriptores locales de esta métrica se extraen de los coeficientes de las sub-bandas (ecuación 1,2 y 3) y después se calculan los términos de luminancia, contraste y estructura (ecuación 4,5 y 6). Finalmente los valores de estas fórmulas se combinan y se obtiene un valor de semejanza total [5] mediante la ecuación 7.

Media de los coeficientes de sub-banda:

$$\mu_x^m = \frac{1}{W} \sum_{(i,j) \in W} X^m(i,j). \quad (1)$$

Desviación estándar de los coeficientes de sub-banda:

$$\sigma_x^m = \sqrt{\frac{1}{W-1} \sum_{(i,j) \in W} (X^m(i,j) - \mu_x^m)^2}. \quad (2)$$

Covarianza de los coeficientes de sub-banda:

$$\sigma_{xy}^m = \frac{1}{W-1} \sum_{(i,j) \in W} (X^m(i,j) - \mu_x^m)(Y^m(i,j) - \mu_y^m), \quad (3)$$

donde x, y son las dos imágenes a comparar. i, j son los índices de los coeficientes en el dominio transformado. m es el índice de sub-bandas. W es el tamaño de la sub-banda. μ_x^m y μ_y^m son las medias de las dos sub-bandas. σ_x^m y σ_y^m son desviaciones estándar de las dos sub-bandas, estas se elevan al cuadrado para obtener las varianzas.

Luminancia de los coeficientes de sub-banda:

$$l_{x,y}^m = \frac{2\mu_x^m \mu_y^m + C_0}{(\mu_x^m)^2 + (\mu_y^m)^2 + C_0}. \quad (4)$$

Contraste de los coeficientes de sub-banda:

$$c_{x,y}^m = \frac{2\sigma_x^m \sigma_y^m + C_1}{(\sigma_x^m)^2 + (\sigma_y^m)^2 + C_1}. \quad (5)$$

Estructura de los coeficientes de sub-banda:

$$s_{x,y}^m = \frac{\sigma_{xy}^m + C_2}{\sigma_x^m \sigma_y^m + C_2}. \quad (6)$$

Estrategía de unión:

$$q_{CW-SSIM}^m(x,y) = (l_{x,y}^m)^\alpha (c_{x,y}^m)^\beta (s_{x,y}^m)^\gamma. \quad (7)$$

Las constantes α , β y γ toman el valor de 1. Los valores de $C_0=6.5025$, $C_1=58.5225$ y $C_2=C_1/2$ se definen con base en los experimentos realizados por [12]. Los descriptores y fórmulas de comparación toman valores de 0 a 1, excepto la estructura, ésta toma valores de -1 a 1. Los valores que se obtienen de aplicar la estrategia de unión van de 0 a 1. Cuando existe mayor similitud entre las imágenes, este valor es próximo a 1.

3.2. Métrica de semejanza estructural de textura (STSIM)

La métrica STSIM (Structural Texture Similarity Metric) es una medida que elimina por completo las comparaciones punto a punto que dependen de los descriptores locales de la imagen. Esto se refiere a reemplazar el término estructura empleado en la métrica CW-SSIM por autocorrelaciones horizontal y vertical. Las imágenes se descomponen en sub-bandas utilizando una pirámide orientable. Los descriptores (ecuación 1,2) y los términos para comparar (ecuación 4,5, 8, 9) se calculan en toda la sub-banda (ventana global) o a través de una pequeña ventana deslizante que recorre la sub-banda. Al final se genera un promedio de la misma [6]. Una vez calculados los términos de comparación, se aplica la ecuación 10 y 11 para obtener las autocorrelaciones horizontal y vertical. Finalmente se emplea la ecuación 12 para obtener un valor de similitud total.

Coeficiente de autocorrelación horizontal:

$$p_x^m(0,1) = \frac{\frac{1}{W} \sum_{(i,j) \in W} (X^m(i,j) - \mu_x^m)(X^m(i,j+1) - \mu_x^m)}{(\sigma_x^m)^2} \quad (8)$$

Coeficiente de autocorrelación vertical:

$$p_x^m(1,0) = \frac{\frac{1}{W} \sum_{(i,j) \in W} (X^m(i,j) - \mu_x^m)(X^m(i+1,j) - \mu_x^m)}{(\sigma_x^m)^2}, \quad (9)$$

donde x , y y son las dos imágenes a comparar. i, j son los índices de los coeficientes en el dominio transformado. m es el índice de sub-bandas. W es el tamaño de las dimensiones de la sub-banda. μ_x^m y μ_y^m son las medias de las dos sub-bandas. σ_x^m y σ_y^m son desviaciones estándar de las dos sub-bandas, estas se elevan al cuadrado para obtener las varianzas.

Autocorrelación horizontal:

$$c_{x,y}^m(0,1) = 1 - 0.5 |\rho_x^m(0,1) - \rho_y^m(0,1)|^p. \quad (10)$$

Autocorrelación vertical:

$$c_{x,y}^m(1,0) = 1 - 0.5 |\rho_x^m(1,0) - \rho_y^m(1,0)|^p. \quad (11)$$

Los valores (0,1) y (1,0) representan el recorrido en la dirección horizontal y vertical de la sub-banda respectivamente. Los valores que se obtienen a partir de la estrategia de unión están en un rango de 0 a 1. Un valor de similitud alto estará más cercano a 1.

Estrategia de unión:

$$Q_{STSIM,x,y}^m = (I_{x,y}^m)^{1/4} (c_{x,y}^m)^{1/4} (c_{x,y}^m(0,1))^{1/4} (c_{x,y}^m(1,0))^{1/4}. \quad (12)$$

4. Metodología

La metodología propuesta involucra los siguientes pasos: selección de un conjunto de imágenes de una base de datos, una descomposición de las imágenes mediante la aplicación de una pirámide orientable compleja, de esta descomposición se procede a extraer la información de cada una de las sub-bandas generadas para posteriormente aplicar las métricas y obtener los valores de similitud. En la Fig. 1 se detalla el esquema general con las etapas y los elementos que intervienen.

1) Base de datos de imágenes. En el paso uno, se eligen de una base de datos las imágenes a comparar. La dimensión de las imágenes debe ser de 128 x 128 píxeles con extensión .JPG. No existe restricción en el tamaño de las imágenes. Sin embargo, éstas deben cumplir con la repetitividad de los patrones. Por convención se determinó usar la dimensión empleada por [2].

2) Descomposición en sub-bandas. En este paso, cada una de las imágenes se descompone en un conjunto de sub-bandas, empleando una pirámide orientable compleja. Primero se realiza una división de la imagen en sub-bandas paso alto y paso bajo, aplicando filtros de paso alto y paso bajo, después la sub-banda paso bajo se divide en un conjunto de sub-bandas de paso de banda orientadas y una sub-banda de paso bajo la cual es sub-muestreada por un factor de dos en las direcciones X y Y. Este proceso se sigue de forma recursiva hasta completar los cuatro niveles de escala y las cuatro orientaciones como se muestra en la Fig. 2.

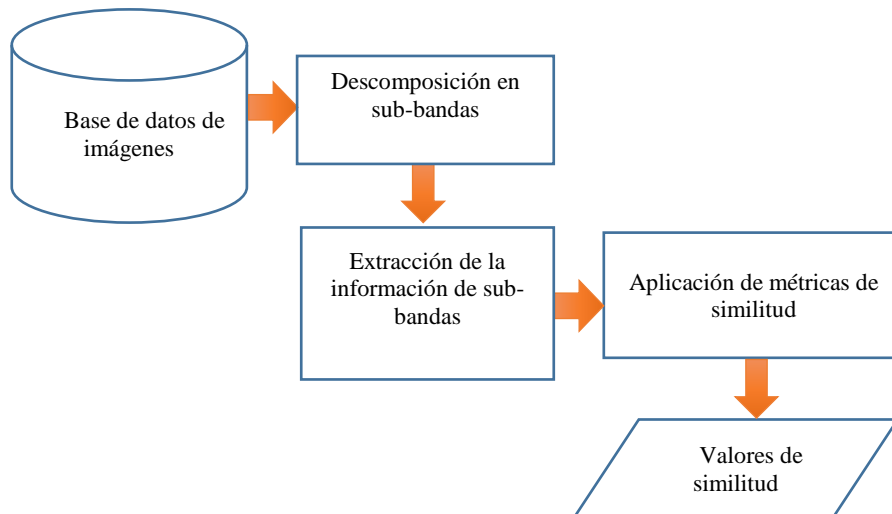


Fig. 1. Arquitectura general del sistema, describe las etapas y los elementos que intervienen.

3) Extracción de la información de sub-bandas. En este paso, cada una de las sub-bandas extraídas es almacenada de forma individual en archivos de texto, por cada imagen se genera un archivo que contiene el nombre de los 18 archivos de texto correspondiente a las 18 sub-bandas en las que se descompuso la imagen. Los archivos

que contienen la información de sub-bandas se almacenan en matrices para la aplicación del siguiente paso.



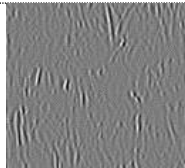
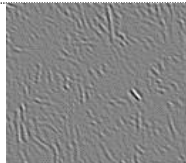
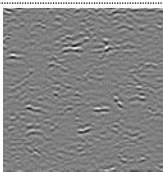
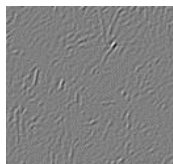
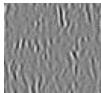
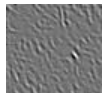
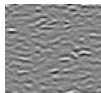
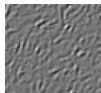









Imagen de referencia		Sub-banda paso alto con dimensión de 128 x 128	
			
Sub-bandas paso banda orientadas			
Orientación de 0 grados	Orientación de 45 grados	Orientación de 90 grados	Orientación de 135 grados
Sub-bandas con dimensión de 128 x 128			
			
Sub-bandas con dimensión de 64 x 64			
			
Sub-bandas con dimensión de 32 x 32			
			
Sub-bandas con dimensión de 16 x 16			
			
Sub-banda paso bajo con dimensión de 8 x 8			
			

Fig. 2. Descomposición de una imagen en sub-bandas.

4) Aplicación de métricas de similitud. Primero se realiza la extracción de descriptores de las matrices, éstos dependerán de la métrica implementada. Una vez extraídos los descriptores, se aplican fórmulas de comparación de descriptores entre matrices de sub-bandas, de la imagen de referencia y la imagen a comparar.

Por último, se aplica una estrategia de unión para obtener un valor de similitud total, este valor se obtiene al multiplicar los resultados de las fórmulas de comparación de descriptores. En la sección 3.1 y 3.2 descritas anteriormente, se describen las métricas implementadas.

5. Experimentos y resultados

5.1. Descripción del conjunto de datos

La base de imágenes se integró de imágenes de texturas naturales de las categorías: flores, montañas (descargadas del sitio web Corbis corporation) y césped (obtenidas mediante captura personal). El número de imágenes por cada categoría es de 50, a estas imágenes se les denomina imágenes fuente, estas imágenes se procesaron para obtener distintas representaciones de cada una con cambios de escala y rotación.

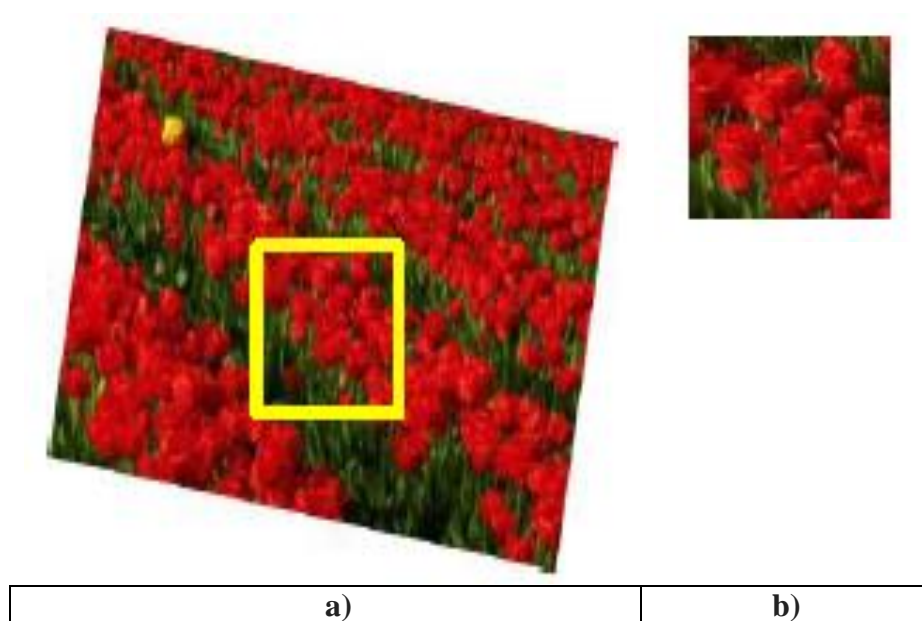


Fig. 3. Proceso de rotación y extracción de un segmento de la imagen fuente. a) Imagen fuente con cambio por rotación y b) Segmento extraído de la parte central de la imagen Fuente.

En los cambios de rotación por cada imagen fuente se tienen 21 representaciones con diferentes grados de rotación, generando así 1050 imágenes por cada categoría y haciendo un total de 3150 imágenes. Para obtener estas imágenes, primero se extrae un segmento de la imagen fuente de 128 x 128 píxeles (este segmento se denomina imagen de referencia), después se rota la imagen y se extrae un segmento de 128 x 128 píxeles como se muestra en la Fig. 3. Los segmentos extraídos forman un conjunto de 21 imágenes (Fig. 4).

En los cambios de escala, por cada imagen fuente, se tienen 11 representaciones con diferentes niveles de escalado, generando así 550 imágenes por cada categoría y haciendo un total de 1650 imágenes. Para obtener estas imágenes, primero se realiza un proceso de extracción de un segmento de la imagen fuente de 128 x 128 píxeles (imagen de referencia), después se escala la imagen fuente y se extrae un segmento de 128 x 128 píxeles como se muestra en la Fig. 5, los segmentos extraídos forman un conjunto de

11 imágenes. En la Fig. 6 se muestra un ejemplo del conjunto de imágenes generado de la categoría flores.

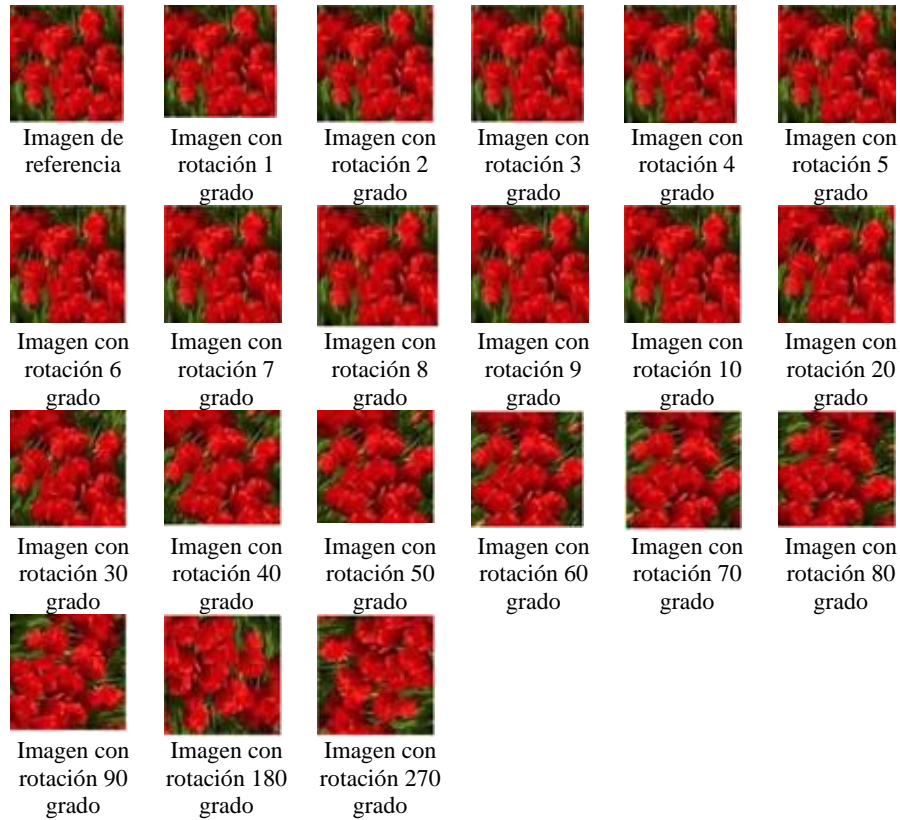


Fig. 4. Imágenes de ejemplo correspondientes a cada magnitud de rotación de la categoría flores.

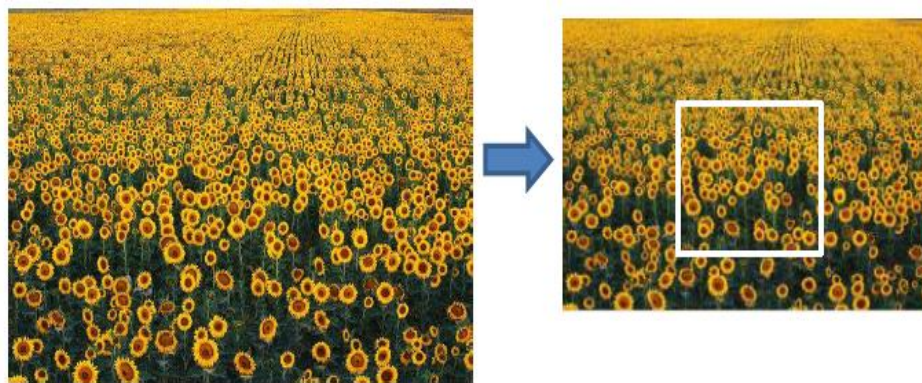


Fig. 5. Proceso de escalado y extracción de un segmento de la imagen fuente.

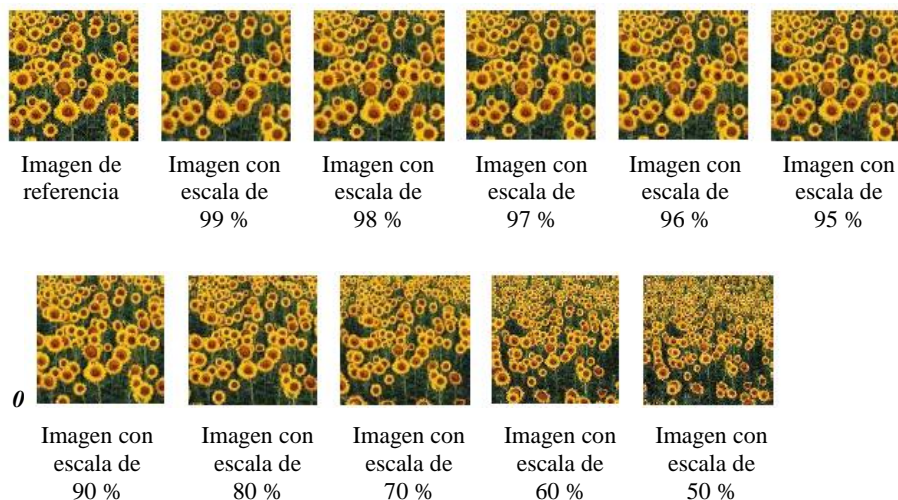


Fig. 6. Imágenes de ejemplo correspondientes a cada escala de la categoría flores.

5.2. Pruebas y resultados

En la parte experimental se realizaron cuatro pruebas. En la primera se utilizaron imágenes de las tres categorías con cambios de rotación aplicando la métrica CW-SSIM. En la segunda prueba, se emplearon las mismas imágenes de la prueba anterior aplicando la métrica STSIM. En la tercera prueba se aplicó la métrica CW-SSIM, para ello se emplearon imágenes de las tres categorías con cambios de escala. Finalmente en la última prueba se utilizaron imágenes de la prueba tres aplicando la métrica STSIM. A partir de los datos obtenidos se realizó un promedio por cada magnitud de rotación y de escala, con la finalidad de obtener un valor de rendimiento total de cada una de las métricas como se muestra en la Tabla 1.

Tabla 1. Resultados de la aplicación aplicación de las métricas CW-SSIM y STSIM a imágenes de texturas naturales de las categorías: flores, césped y montañas.

Cambios en las imágenes	Métrica	Categoría Flores	Categoría Césped	Categoría Montañas	Promedio
Rotación	CW-SSIM	0.5126	0.3948	0.5499	0.4857
Rotación	STSIM	0.9843	0.9854	0.9508	0.9735
Escala	CW-SSIM	0.7008	0.6541	0.7458	0.7002
Escala	STSIM	0.9881	0.9893	0.9748	0.9840

En los experimentos con imágenes de texturas naturales con cambios por rotación, las métricas CW-SSIM y STSIM reportan valores promedio de desempeño de: 0.4857 y 0.9735 respectivamente. En los experimentos con imágenes de texturas naturales con cambios de escala, las métricas CW-SSIM y STSIM reportan valores promedio de desempeño de: 0.7002 y 0.9840 respectivamente.

6. Conclusiones

Con base en los resultados obtenidos, se determinó que la métrica STSIM es la más recomendable para valorar la similitud de imágenes de texturas naturales con cambios de escala y rotación, porque presenta valores promedio de similitud más altos en todas las categorías de imágenes de texturas naturales empleadas.

La métrica STSIM se basa en una descomposición en sub-bandas. Éstas se emplean para la medición de los elementos de luminancia y contraste utilizados en la métrica CW-SSIM y se reemplaza el término estructura por autocorrelaciones horizontal y vertical. Si bien, la luminancia y el contraste siguen comparando las medias y las varianzas, las autocorrelaciones horizontal y vertical comparan las autocovarianzas. Éstas proporcionan información de la direccionalidad horizontal y vertical para mejorar la comparación de texturas.

En este trabajo, la aplicación de las métricas de similitud se sitúa en un nivel de descripción ya que los valores obtenidos describen la semejanza entre las imágenes. Únicamente se analiza el desempeño de las métricas con base en los valores de semejanza obtenidos. Sin embargo, si se requiere realizar tareas de reconocimiento en el área de visión artificial, los valores de semejanza pueden ser empleados por algoritmos de clasificación y segmentación, entre otros.

Las métricas de similitud se aplican en diversas áreas científicas, entre ellas destaca la medicina, en donde se requieren sistemas para clasificar, recuperar y comprimir imágenes.

Referencias

1. Pappas, T. N., Neuhoﬀ, D. L., de Ridder, H., Zujovic, J.: Image Analysis: Focus on Texture Similarity. *Proceedings of the IEEE*, Vol. 101, No. 9, pp. 2044–2057 (2013)
2. Zujovic, J.: Perceptual Texture Similarity Metrics. Northwestern University (2011)
3. La Serna-Palomino, N., Contreras, W., Ruiz, M. E.: Procesamiento Digital de Textura: Técnicas utilizadas en Aplicaciones actuales de CBIR. *Revista de Investigación de Sistemas e Informática*, Vol. 7, No. 1, pp. 57–64 (2010)
4. Kim, K. J., Kim, B., Mantiuk, R., Ritcher, T., Lee, H., Kang, H. S., Seo, J., Lee, K. H.: A Comparison of Three Image Fidelity Metrics of Different Computational Principles for JPEG2000 Compressed Abdomen CT Images. Vol. 29, No. 8, pp. 1496–1503 (2010)
5. Zujovic, J., Pappas, T. N., Neuhoﬀ, D. L., van Egmond, R., de Ridder, H.: Subjective and objective texture similarity for image compression. *Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1369–1372 (2012)
6. Zujovic, J., Pappas, T. N., Neuhoﬀ, D. L.: Structural Texture Similarity Metrics for Image Analysis and Retrieval. *IEEE Transactions on Image Processing*, Vol. 22, No. 7, pp. 2545–2558 (2013)
7. Patil, S., Talbar, S.: Content Based Image Retrieval Using Various Distance Metrics. *Lecture Notes in Computer Science*, Vol. 6411, Springer-Verlag, Berlin Heidelberg (2012)
8. Avellaneda-Avellaneda, D.: Caracterización de Texturas Naturales. Centro Nacional de Investigación y Desarrollo Tecnológico (2009)
9. Zhou, W., Bovik, A.C.: Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures. *Signal Processing Magazine*, Vol. 26, No. 1, pp. 98–117 (2009)

10. Troncoso-Rey, P. A.: Indexado y Recuperación de Imágenes por Contenido. Centro Nacional de Investigación y Desarrollo Tecnológico (2007)
11. An, J., Zhang, X.: Robust Image Matching Method Based on Complex Wavelet Structural Similarity. *Advances in Computer Science*, Springer-Verlag, Berlin Heidelberg, Vol. 215 (2011)
12. Zhou, W., Bovik, A. C., Sheikh, H. R., Simoncelli, E. P.: Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, Vol. 13, No. 4, pp. 600–612 (2004)

Selección y clasificación de genes cancerígenos utilizando un método híbrido filtro/wrapper

Luis Alberto Hernández Montiel, Carlos Edgardo Cruz Pérez, Juan Gabriel Ruiz Ruiz

Universidad del Istmo Campus Ixtotec, Ciudad Ixtotec, Oaxaca,
México

luisahm@itamail.itapizaco.edu.mx, carloscruz@bianni.unistmo.edu.mx,
jugaruizr@gmail.com

Resumen. En este artículo, se presenta un método híbrido aplicado en la selección y clasificación de genes obtenidos de microarreglos de ADN. El método realiza una limpieza del microarreglo utilizando un preprocesamiento basado en técnicas de filtrado de datos, después, se realiza una selección de genes dentro del subconjunto obtenido por el filtro, utilizando una búsqueda gravitacional combinada con un clasificador KNN. El método se ha construido para obtener un subconjunto de genes relevantes de alto desempeño, los resultados obtenidos se comparan con diferentes métodos reportados en la literatura, este método es implementado en cinco microarreglos de dominio público.

Palabras clave: microarreglos de ADN, filtrado de datos, selección, clasificación, búsqueda gravitacional.

Cancerous Genes Selection and Classification Using a Hybrid Filter/Wrapper Method

Abstract: In this paper, a hybrid method applied in selection and classification of genes obtained from DNA-microarray is presented. The method performs a microarray cleaning using a preprocessing based on data filtering techniques, then, a gene selection is performed within subset obtained from a filter, using a gravitational search combined a SVM-classifier. This method has been built for obtain a relevant genes subset with high performance, the results obtained are compared with other methods reported in literature. This method is tested using five public domain microarrays.

Keywords: DNA-microarray, data filtering, selection, classification, gravitational search.

1. Introducción

Actualmente existen tecnologías que apoyan la búsqueda de enfermedades degenerativas (como el cáncer) [1]. Una de las tecnologías más utilizadas son los

microarreglos de ADN, esta tecnología mide los niveles de expresión genética de miles de genes simultáneamente [2]. Diferentes estudios indican que los perfiles de expresión genética proporcionan información para distinguir un tipo de cáncer dentro de tejido morfológicamente similar y generar un mejor diagnóstico y proponer terapias para contraatacar la enfermedad [3]. La clasificación de tumores se asocia con el problema de selección de genes, el objetivo es extraer un conjunto de genes relevantes de un microarreglo. Sin embargo, esta no es una tarea fácil, el microarreglo contiene genes informativos y ruidosos [4]. Por ello es necesaria la identificación de biomarcadores para la elaboración de pruebas de diagnóstico. Para solucionar el problema de la selección y clasificación de genes se propone un método basado en filtros y wrapper. El método genera un pre-procesamiento basado en la puntuación que otorga una estrategia de filtro para hacer una limpieza del microarreglo, después se implementa una búsqueda gravitacional combinada con clasificador KNN para seleccionar los genes con mejor tasa de desempeño dentro del subconjunto obtenido por el filtro. Con este método, se buscan los genes más relevantes para el diagnóstico de cáncer dentro de cinco microarreglos de ADN de dominio público.

2. Estado del arte

La tecnología de microarreglos de ADN manipula grandes volúmenes de información genética perteneciente a varios tipos de enfermedades [2]. Esta información contiene un gran número de datos que requieren un largo tiempo de procesamiento. La alta dimensión del microarreglo genera un problema de precisión de análisis y complejidad computacional [5]. Los microarreglos no solo cuentan con una gran cantidad de atributos (genes) y un número limitado de muestras, también contienen dos o más número de clases (categorías) a las que pertenece cada uno de los atributos, además miles de los genes son redundantes o ruidosos [6].

Para solucionar este problema, diferentes autores han propuesto métodos basados en técnicas de minería de datos (como selección y extracción de características) y de aprendizaje máquina [7] que ayudan a obtener información relevante a través de la exploración del microarreglo.

El método más utilizado es la clasificación de características [8] con esta técnica se logra distinguir entre varias clases de muestras de tejido que contienen alguna enfermedad.

El aprendizaje evolutivo [9] utiliza las propiedades de los algoritmos bioinspirados [10, 11] combinados con algún método de clasificación y así exploran el microarreglo buscando información relevante para el diagnóstico de alguna enfermedad.

Los algoritmos de aprendizaje máquina [12] basados en métodos de filtrado y/o wrapper, eliminan la información menos relevante dentro del microarreglo utilizando una puntuación que sirve como valor de pertinencia del gen, seleccionando solo información confiable para su análisis.

Las técnicas de clusters [13] agrupan los genes similares, después se utiliza algún método de puntuación discriminante que ayude a eliminar la información no relevante del microarreglo.

Actualmente se implementan algoritmos híbridos [14, 15] combinando diferentes técnicas de filtrado de datos y algoritmos wrapper que trabajan de forma paralela,

aplicados en la selección y extracción de genes relevantes para el diagnóstico de una enfermedad. A pesar del número de técnicas implementadas para abordar el problema de selección de genes, no se ha llegado a una solución concreta, cada modelo o método presentado selecciona genes que no se han reportado y abre el panorama a nuevos estudios, debido a esto surgen más trabajos con nuevas propuestas dando un estudio más confiable de los genes seleccionados.

3. Materiales y métodos

Los microarreglos de ADN contienen información relevante mezclada con información ruidosa y redundante [7] esto genera tiempo de procesamiento largo y dificulta para extraer información valiosa de ellos con resultados poco confiables. Para abordar este problema se propone un método híbrido combinando técnicas de filtrado de datos como primera etapa de selección. En la segunda etapa se implementa un modelo de selección y clasificación utilizando una búsqueda gravitacional combinada con un clasificador KNN, los materiales y métodos utilizados se describen a continuación. La figura 1 muestra la estructura general del método propuesto.

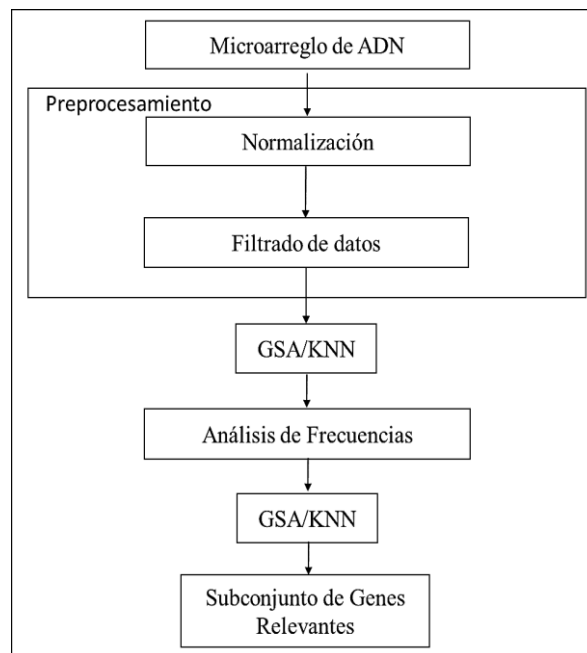


Fig. 1. Método general de selección y clasificación.

3.1. Microarreglos de ADN

Los datos obtenidos de un microarreglo de ADN se obtienen mediante una matriz donde las filas representan los genes y las columnas representan las muestras. Cada

celda dentro de la matriz es un valor de expresión genética que representa la intensidad del gen correspondiente a cada muestra [16]. Lo anterior se observa en la figura 2 donde x representa los datos genómicos, ng (número de gen, filas) los genes dentro de la matriz y nm (número de muestras, columnas) las muestras dentro de la matriz.

$$x_{ij} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & \dots & \dots & x_{1nm} \\ x_{21} & x_{22} & x_{23} & \dots & \dots & \dots & x_{2nm} \\ x_{31} & x_{32} & x_{33} & \dots & \dots & \dots & x_{3nm} \\ \vdots & \vdots & \vdots & & & & \vdots \\ \vdots & \vdots & \vdots & & & & \vdots \\ \vdots & \vdots & \vdots & & & & \vdots \\ x_{ng1} & \cdot & \cdot & & & & x_{ngnm} \end{bmatrix}$$

Fig. 2. Matriz de datos de expresión genética.

En éste trabajo, se utilizan cinco bases obtenidas de la tecnología de microarreglos de ADN, descritas en la tabla 1.

Tabla 1. Características de los microarreglos de ADN.

Microarreglo de ADN	Genes	Muestras	Casos	Controles	Referencia
Leucemia	7128	72	25 ALL	47 AML	[8]
Colon	2000	62	22 Tumor	44 Normal	[3]
Pulmón	12533	181	31 MPM	150 ADCA	[17]
CNS	7129	60	21 survivors	39 failures	[18]
DLBCL	4026	47	23 B-like Activado	24 B-like germinal	[19]

3.2. Pre-procesamiento

La información original de un microarreglo se encuentra en diferentes escalas numéricas o diferentes distribuciones de probabilidad. Para solucionar este problema se genera una transformación de los datos a un rango entre cero y uno para facilitar su estudio. En este trabajo como primer paso del pre-procesamiento se realiza una normalización basada en una técnica min-máx. [20]:

$$X' = \frac{X - \text{Min}(X)}{\text{Max}(X) - \text{Min}(X)}, \quad (1)$$

donde X es la base de datos original. $\text{Min}(X)$ y $\text{Max}(X)$ es el dato mínimo y máximo existente dentro de la base de datos. X' es la nueva base de datos normalizada.

El segundo paso de la etapa de pre-procesamiento es realizar una primera selección de genes utilizando técnicas de filtrado. La idea es que cada método genere un ranking de los genes del microarreglo asignándole un valor de pertinencia que ayude a discriminar los genes relevantes de los no relevantes. Los métodos utilizados en este experimento se describen a continuación.

-BSS/WSS: El método selección genes basado en la razón de la suma de cuadrados entre clases (BSS) y dentro de las clases (WSS). Para el gen j , la razón está dada por [21]:

$$\frac{BSS(j)}{WSS(j)} = \frac{\sum_i \sum_k I(y_i = k) (\bar{x}_{kj} - \bar{x}_j)^2}{\sum_i \sum_k I(y_i = k) (x_{ij} - \bar{x}_{kj})^2}, \quad (2)$$

donde \bar{x}_j denota el nivel medio de la expresión del gen j a través de todas las muestras y \bar{x}_{kj} denota el nivel medio de la expresión del gen j en todas las muestras para la pertenencia de la clase k .

Relación señal a ruido (SNR) En este método se identifican los patrones de expresión genética con diferencia máxima en la expresión media entre dos grupos y la variación mínima de expresión dentro de cada grupo, los genes se clasifican de acuerdo a sus niveles de expresión [22].

$$SNR = |(\mu_1 - \mu_2)/(\sigma_1 + \sigma_2)| \quad (3)$$

Donde μ_1 y μ_2 denotan los valores medios de expresión de la clase 1 y clase 2, respectivamente, σ_1 y σ_2 son las desviaciones estándar de las muestras en cada clase.

-Información mutua: Este método toma dos genes (A y B) de forma aleatoria con distribuciones de probabilidad diferentes y una distribución de probabilidad conjunta. La información mutua entre ambos genes $I(A;B)$ se define como la entropía relativa entre la probabilidad conjunta y el producto de probabilidades [23].

$$I(A;B) = \sum_{a_i} \sum_{b_j} P(a_i, b_j) \log \frac{P(a_i, b_j)}{P(a_i)P(b_j)}, \quad (4)$$

donde $P(a_i, b_j)$ es la probabilidad conjunta de los genes, $P(a_i)$ es la probabilidad del gen A y $P(b_j)$ es la probabilidad del gen B.

Para la segunda etapa del método propuesto, se ha implementado un algoritmo de selección de genes basado en una búsqueda gravitacional combinada con un clasificador KNN, el método se entrena con los conjuntos obtenidos por los filtros y se describe a continuación.

3.3. Búsqueda gravitacional

Es un algoritmo de búsqueda propuesto por Rashedi en 2009 [15, 24]. El GSA está basado en la ley de gravedad de Newton, donde la fuerza de gravedad entre dos cuerpos es directamente proporcional al producto de sus masas e inversamente proporcional al cuadrado de su distancia. Las soluciones en la población del GSA se llaman agentes que interactúan entre sí a través de la fuerza de gravedad. La calidad de cada agente se mide por su masa. Cada agente se considera como objeto y todos los objetos se mueven hacia otros objetos con masa más pesada debido a la fuerza de gravedad que generan, este paso representa un movimiento global (paso de exploración) del objeto, mientras que el agente con una masa pesada se mueve lentamente, lo que representa la etapa de explotación del algoritmo. La mejor solución encontrada por el algoritmo es el agente con la masa más pesada [25].

La constante de gravedad G en la iteración t que utiliza el algoritmo se calcula por [15]:

$$G(t) = G_0 e^{-\alpha t/T}, \quad (5)$$

donde G_0 y α son inicializados al comienzo de la búsqueda y sus valores se irán reduciendo durante cada iteración. T es el número total de iteraciones.

Dentro del algoritmo existen tres tipos de masas: Masa gravitacional activa M_a , masa gravitacional Pasiva M_p y masa inercial M_i . la fuerza de gravedad F_{ij} que actúa sobre las masas i y j obedeciendo la ley de gravedad de Newton definida por:

$$F = G \frac{M_{aj} \times M_{pi}}{R^2}, \quad (6)$$

donde M_{aj} , M_{pi} son la masa activa y pasiva del objeto j , i , respectivamente.

De acuerdo con de la segunda ley de Newton, cuando se aplica una fuerza F a un objeto, el objeto se mueve con aceleración a dependiendo de la fuerza aplicada y la masa del objeto M , la aceleración del objeto i (agente) se calcula como sigue:

$$a_i = \frac{F_{ij}}{M_{ii}}, \quad (7)$$

donde M_{ii} es masa de inercial del agente i .

Los agentes actualizan sus velocidades y posiciones, como se muestra en las ecuaciones 8 y 9 respectivamente:

$$V_i(t+1) = rand_i * V_i(t) + a_i(t), \quad (8)$$

$$X_i(t+1) = rand_i * V_i(t) + a_i(t). \quad (9)$$

3.4. Clasificador KNN

El clasificador k-vecino más cercano (KNN por k-Nearest Neighbor) es un algoritmo de clasificación que basa su criterio de aprendizaje en la hipótesis de que los miembros de una población suelen compartir propiedades y características con los individuos que los rodean [25] de modo que es posible obtener información descriptiva de un individuo mediante la observación de sus vecinos más cercanos.

La regla de clasificación por KNN se describe a continuación: sea x^1, x^2, \dots, x^n una muestra con una función $f(x)$ de densidad desconocida. Se estima $f(x)$ a partir de un elemento central de la muestra x que crece hasta contener k elementos con una distancia euclidiana similar, donde el valor de k se define arbitrariamente. Estas observaciones son los k vecinos más cercanos a x . Se tiene entonces la siguiente condición [15, 26]:

$$\hat{f}(x) = \frac{\frac{k}{n}}{V_k(x)}, \quad (10)$$

donde $V_k(x)$ es el volumen de un elipsoide centrado en x , y de radio la distancia euclidiana de x al k -ésimo vecino más cercano.

3.5. Implementación de algoritmo híbrido GSA/KNN

En nuestro caso, el algoritmo híbrido se implementa de la siguiente forma [15].

Paso 1. Se establecen los valores iniciales, la constante de gravedad G_0 , α , ε y el número de iteraciones t .

Paso 2. La población inicial se genera aleatoriamente siguiendo una distribución uniforme, se compone de N agentes asociados con cada gen dentro del microarreglo.

Paso 3. El clasificador KNN se introduce en la función de costo de algoritmo para evaluar los agentes de la población, para verificar el error del clasificador se utiliza un k-fold cross validation.

Paso 4. La constante de gravedad se actualiza como se muestra en la ecuación 5.

Paso 5. Cuando el agente j actúa sobre el agente i con fuerza, en un tiempo específico (t), la fuerza es calculada por:

$$F_{ij}^d(t) = G(t) \frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \varepsilon} (x_j^d(t) - x_i^d(t)), \quad (11)$$

donde M_{aj} es la masa gravitacional activa del agente j , M_{pi} es la masa gravitacional pasiva del agente i , $G(t)$ es constante de gravedad en el tiempo t .

Paso 6. En cada iteración t , la fuerza total que actúa sobre el agente i se calcula por:

$$F_i^d(t) = \sum_{j \in Kbest, j \neq i} rand_j F_{ij}^d(t), \quad (12)$$

donde $Kbest$ es el conjunto de los primeros K agentes con la masa más grande.

Paso 7. La masa inercial es calculada por:

$$m_i(t) = \frac{fit_i - worst(t)}{best(t) - worst(t)}, \quad (13)$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)}. \quad (14)$$

Paso 8 La aceleración de agente i se calcula por la ecuación 7, la velocidad y la posición de agente i se calculan con las ecuaciones 8 y 9. Se incrementa el número de iteraciones hasta cumplir con los criterios de paro, se produce el mejor subconjunto de agentes (genes) con la mejor aptitud.

4. Protocolo y resultados experimentales

El método propuesto se entrena con cinco microarreglos de ADN descritos en la tabla 1. Se observa que el método es capaz de seleccionar genes con información que

ayuda con el diagnóstico de un tipo de cáncer. En esta sección se muestran los parámetros utilizados y los resultados obtenidos por el método propuesto.

4.1. Parámetros y resultados

El protocolo experimental se realizó en un pc DELL vostro con procesador i5 y memoria RAM de 4gb. El algoritmo fue implementado en Matlab versión 7.12. Los parámetros más confiables se muestran en la tabla 2.

Tabla 2. Parámetros utilizados por el algoritmo híbrido.

PARAMETROS	
Número de agentes	50
Dimensión	300
Número de iteraciones	500

El protocolo experimental se dividió en dos etapas, en la primera se utilizan tres métodos de filtrado de datos que funcionan como una fase de preselección descartando genes ruidosos o redundantes y obtenido como resultado subconjuntos con información relevante, de este proceso se han obtenido un subconjunto de genes. En la siguiente etapa se presenta un algoritmo basado en una búsqueda gravitacional, este algoritmo logra explorar y explotar el espacio de búsqueda dentro del subconjunto obtenido en la etapa de filtrado seleccionando genes para entrenar un clasificador basado en un vecino más cercano (KNN). El clasificador KNN se ha introducido en la función de costo del algoritmo con el objetivo de saber si el gen seleccionado es relevante para distinguir algún tipo de cáncer. La combinación de estas dos técnicas permite seleccionar genes utilizando la tasa de clasificación como método de discriminación y eliminando los genes que no logren entrenar al clasificador, seleccionando genes que obtienen una tasa de clasificación alta. De esta forma se logra reducir el tamaño del microarreglo y se genera un subconjunto de genes que contiene información relevante para el diagnóstico de una enfermedad.

Dentro de cada microarreglo el algoritmo GSA/KNN ha seleccionado un pequeño conjunto de genes informativos, para saber si estos genes son relevantes para el diagnóstico de cáncer, se revisa si han sido reportados en la literatura obteniendo una interpretación biológica confiable de cada gen.

Tabla 3. Tasa de clasificación obtenida por el método propuesto.

Microarreglos	Etiqueta	Nombre del Gen	Reportado
Leucemia	X95735	Zyxin	[27], [8]

En el microarreglo de leucemia el algoritmo selecciona solo un gen que debido a su nivel de expresión logra identificar dos tipos de leucemia aguda y así ser clasificado en la clase Leucemia Mieloide Aguda o Leucemia Linfoblastica Aguda. La tabla 3 muestra la descripción del gen seleccionado por el algoritmo GSA/KNN.

Tabla 4. Tasa de clasificación obtenida por el método propuesto.

Microarreglo	Etiqueta	Nombre del Gen	Reportado
Cáncer de Colon	R87126	<i>MYOSIN HEAVY CHAIN, ONMUSCLE (Gallus gallus)</i>	[28], [29].
	M76378	<i>Human cysteine-rich protein (CRP) gene, exons 5 and 6</i>	

La tabla 5 muestra la descripción del gen seleccionado por el algoritmo para el microarreglo de cáncer de pulmón, logrando separar la clase Malignant Pleural Mesothelioma (MPM) de la clase Adenocarcinoma (ADCA).

Tabla 5. Descripción de los genes seleccionados para Cáncer de pulmón.

Microarreglo	Etiqueta	Nombre del Gen	Reportado
Cáncer de Pulmón	X99270	three prime repair exonuclease 2	[--]

La tabla 6 y la tabla 7 muestran la descripción de los genes seleccionados por el algoritmo para el microarreglo DLBCL y el microarreglo CNS.

Tabla 6. Descripción de los genes seleccionados para el microarreglo DLBCL.

Microarreglo	Etiqueta	Nombre del Gen	Reportado
DLBCL	GENE3327X	Unknown UG Hs.169565 ESTs, Moderately similar to ALU SUBFAMILY SB WARNING ENTRY [H.sapiens]; Clone=825217	[30]
	GENE3261X	*Unknown; Clone=1353015	
	GENE3330X	*Unknown; Clone=825199	
	GENE3329X	Unknown UG Hs.224323 ESTs, Moderately similar to alternatively spliced product using exon 13A [H.sapiens]; Clone=1338448	
	GENE477X	*Putative oncogene protein similar to C. elegans ZC395.7 gene product; Clone=590942	

Tabla 7. Descripción de los genes seleccionados para el microarreglo CNS.

Microarreglo	Etiqueta	Nombre del Gen	Reportado
CNS	L17131	<i>High mobility group protein (HMG-I(Y)) gene exons 1-8</i>	[19]
	U69126	<i>FUSE binding protein 2 (FBP2) mRNA, partial cds</i>	
	X03689	<i>mRNA fragment for elongation factor TU (N-terminus)</i>	
	M64347	<i>FGFR3 Fibroblast growth factor receptor 3 (achondroplasia, thanatophoric dwarfism)</i>	
	X93510	<i>37 kDa LIM domain protein</i>	

En cáncer de colon, el algoritmo selecciona dos genes con información relevante para identificar células con cáncer de colon, separando las células de tejidos cancerosos de los tejidos normales. La tabla 4 muestra la descripción de los genes seleccionados.

Una forma de verificar si el algoritmo propuesto es competente, es comparando las tasas de clasificación obtenidas por el algoritmo GSA/KNN con diferentes tasas de clasificación reportadas por diferentes autores que han utilizado un modelo similar al propuesto en este trabajo. La tabla 8 muestra el estudio de comparación de las tasas de clasificación obtenidas con otros métodos reportados en la literatura. Al comparar los resultados, se verifica que en algunos casos el algoritmo ha superado las tasas de clasificación y/o reduciendo el número de genes a utilizar en comparación con los métodos reportados en la literatura.

Tabla 8. Estudio de comparación.

AUTOR	Leucemia %(G)	Colon %(G)	DLBCL %(G)	Pulmón %(G)	CNS %(G)
Hernández et al [31]	92.52% (6)	87.00%(8)	--	--	95.44% (12)
Filippone et al [32]	94.7% (13)	80.6(21)	--	--	--
Gunavathi [33]	100% (3)	95.00%(7)	--	100%(5)	87.5%(5)
Bonilla-Huerta [34]	97.5%(3)	90.5% (3)	96.00% (3)	93.8% (3)	94.30% (4)
Li et al [35]	95.1%(21)	88.7%(16)	--	--	--
Yu et al. [36]	--	--	--	92.86% (71)	--
Pang et al. [37]	94.1(35)	83.8(23)	--	91.2(34)	--
GSA/KNN	98% (1)	94.77% (2)	97.72% (1)	86.43% (5)	86.33% (5)

Tabla 9. Tiempo de ejecución de los algoritmos implementados.

Algoritmo	Leucemia	Colon	DLBCL	Pulmón	CNS
AG/KNN	2452.189090 segundos.	2343.115777 segundos.	2243.731505 segundos.	2627.635081 segundos.	2449.328465 segundos.
BT/KNN	2580.345980 segundos.	6255.776846 segundos.	10696.475925 segundos.	4021.763980 segundos.	12450.260821 segundos.
BC/KNN	837.955481 segundos.	801.900155 segundos.	795.473768 segundos.	885.674457 segundos.	816.134201 segundos.
GSA/KNN	430.297611 segundos.	438.097286 segundos.	437.329687 segundos.	458.215342 segundos.	438.984319 segundos.

Cada algoritmo propuesto para solucionar el problema de selección y clasificación de genes genera un costo computacional que se puede medir en tiempo-máquina por segundos, en este trabajo se hace un estudio de comparación del tiempo de respuesta del método propuesto y diferentes algoritmos que se han utilizado para solucionar este problema. Los algoritmos con los que se ha comparado son un Algoritmo Genético (AG), una Búsqueda Tabú (BT), una Búsqueda Cuckoo (BC) combinados con un clasificador KNN, la tabla 9 muestra el tiempo de ejecución en segundos de cada algoritmo implementado trabajando bajo las mismas condiciones presentadas en la sección 4.1. Se observa que el algoritmo propuesto GSA/KNN tiene el menor tiempo de ejecución en comparación de los algoritmos comparados.

5. Conclusiones y trabajos futuros

En este trabajo se presentó un método híbrido basado en técnicas de filtro y un algoritmo wrapper implementado en la selección y clasificación de un conjunto de

genes relevantes, explorando dentro de cinco microarreglos de ADN de dominio público (Leucemia, Cáncer de Colon, Cáncer de Pulmón, DLBCL y CNS). El método propuesto tiene una etapa de preselección de genes, utilizando tres técnicas de filtrado de datos, estos filtros generan una puntuación que sirve para eliminar los genes no relevantes (genes ruidosos o redundantes) y selecciona subconjuntos de genes con información pertinente.

Para realizar la selección dentro del subconjunto obtenido por el proceso de filtrado, se ha implementado un método wrapper basado en una búsqueda gravitacional como método de selección combinada con clasificador KNN como método de clasificación. Utilizando las propiedades de la búsqueda se ha logrado implementar un algoritmo que explora el microarreglo exhaustivamente, colocando agentes (genes) que son atraídos por la fuerza de gravedad de agentes con masas más grandes (mejores soluciones) logrando recorrer cada lugar en un espacio dimensional (espacio de soluciones), de esta manera se utiliza la mayoría de genes propuestos para el estudio (p-value) eliminando agentes (genes) que tienen una masa pequeña (malas soluciones). Combinando esta característica con el método de clasificación se seleccionan genes que han obtenido una tasa de clasificación alta.

Referencias

1. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, pp. 1157–1182 (2003)
2. Moreno, V., Solé, X.: *Uso de Chips de ADN (Microarrays) en Medicina: Fundamentos Técnicos y Procedimientos Básicos para el Análisis Estadístico de Resultados*. Unidad de Bioestadística y Bioinformática, Instituto Catalán de Oncología, Barcelona España (2000)
3. Alon, U., Barkai, N., Notterman, D.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Nat. Acad. Sci. USA.*, 96, pp. 6745–6750 (1999)
4. Ben-Dor, A., Bruhn, L., Friedman, N.: Tissue classification with gene expression profiles. *Journal of Computational Biology*, 7(3-4), pp. 559–583 (2000)
5. Rubido, R. P.: Una revisión a algoritmos de selección de atributos que tratan la redundancia en datos de microarreglos. *Revista Cubana de Ciencias Informáticas*, pp. 16–30 (2013)
6. Hwang, T., Sun, C. H., Yun, T., Yi, G. S.: Figs: A Filter-Based Gene Selection Workbench for Microarray Data. *BMC Bioinformatics* (2010)
7. Zhang, Y., Ding, C., Li, T.: Gene Selection Algorithm by Combining ReliefF and mRMR. *IEEE 7th International Conference on Bioinformatics and Bioengineering*, MA, USA (2008)
8. Golub, T., Slonim, D., Tamayo, P.: Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, pp. 531–537 (1999)
9. Kulkarni, A., Kumar, B. S. C.: Colon cancer prediction with genetics profiles using evolutionary techniques. *Expert Systems with Applications*, pp. 2752–2757 (2011)
10. Rakkiannan, T., Palanisamy, B.: Hybridization of Genetic Algorithm with Parallel Implementation of Simulated Annealing for Job Shop Scheduling. *American Journal of Applied Sciences*, pp. 1694–1705 (2012)
11. Li, S., Wu, X., Tan, M.: Gene Selection using Hybrid Particle Swarm Optimization and Genetic Algorithm. *Soft Comput*, pp. 1039–1048 (2008)

12. Yu, G., Feng, Y., Miller, D. J., Xuan, J.: Matched Gene Selection and Committee Classifier for Molecular Classification of Heterogeneous Diseases. *Journal of Machine Learning Research*, pp. 2141–2167 (2010)
13. Do, K.: Applications of gene shaving and mixture models to cluster microarray gene expression data. *Cancer Informatics*, 2, pp. 25–43 (2007)
14. Hernandez-Montiel, L. A.: Hybrid Algorithm Applied on Gene Selection and Classification from Different Diseases. *IEEE Latin America Transactions*, pp. 930–935 (2016)
15. Xiang, J.: A novel hybrid system for feature selection based on an improved gravitational search algorithm and k-NN method. *Applied Soft Computing*, pp. 293–307 (2015)
16. Huang, Q., Tao, D., Li, X., Liew, W. C.: Parallelized Evolutionary Learning for Detection of Biclusters in Gene Expression Data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (2012)
17. Gordon, G. J.: Translation of Microarray Data into Clinically Relevant Cancer Diagnostic Tests Using Gene Expression Ratios in Lung Cancer and Mesothelioma. *Cancer Res.* (2002)
18. Pomeroy, S. L., Tamayo, P.: Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, pp. 436–442 (2002)
19. Alizadeh, A. A., Eisen, B. M., Davis, R. E.: Distinct Types of Diffuse Large (B)–Cell Lymphoma Identified by Gene Expression Profiling. *Nature*, pp. 503–511 (2000)
20. Martínez, W. L., Martínez, A. R.: *Exploratory Data Analysis with MATLAB®*. A CRC Press Company. Boca Ratón London New York Washington, D.C (2005)
21. Dudoit, S., Fridlyand, J., Speed, T.: Comparison of Discrimination Methods for the Classification of Tumors using Gene Expression Data. *J. American Statistical Association*, pp. 77–87 (2002)
22. Mishra, D., Sahu, B.: Feature Selection for Cancer Classification: A Signal-to-noise Ratio Approach. *International Journal of Scientific & Engineering Research* (2011)
23. Zaffalon, M., Hutter, M.: Robust Feature Selection by Mutual Information Distributions. *18th International Conference on Uncertainty in Artificial Intelligence*, pp. 577–584 (2002)
24. Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA: A Gravitational Search Algorithm. *Information Sciences*, Vol. 179, No. 13, pp. 2232–2248 (2009)
25. Sajedi, H., Razavi, S. F.: DGSA: discrete gravitational search algorithm for solving knapsack problem. *Oper Res Int J*, pp. 1–29 (2016)
26. Sugunal, N., Thanushkodi, K.: An Improved k-Nearest Neighbor Classification Using Genetic Algorithm. *IJCSI International Journal of Computer Science Issues*, pp. 18–21 (2010)
27. Wang, X., Gotoh, O.: Cancer classification using single genes. *G. Inf.*, pp. 176–188 (2009)
28. Zhang, H., Song, X., Wang, H., Zhang, X.: Miclique: An Algorithm to Identify Differentially Co-expressed Disease Gene Subset from Microarray Data. *Journal of Biomedicine and Biotechnology* (2009)
29. Li, S., Wu, X., Hu, X.: Gene selection using genetic algorithm and support vectors machines. *Soft Comput*, pp. 693–698 (2008)
30. Aguilar-Ruiz, J. S., Azuaje, F., Riquelme, J. C.: *Data Mining Approaches to Diffuse Large B–Cell Lymphoma Gene Expression Data Interpretation*. DaWaK, Springer-Verlag Berlin Heidelberg, LNCS, 3181, pp. 279–288 (2004)
31. Hernández-Hernández, J. C., Duval, B. J., Hao, K.: SVM-based local search for gene selection and classification of microarray data. *Comunicativos in Computer and Information Science*, Vol. 13, pp. 499–508 (2008)
32. Filippone, M., Masulli, F., Rovetta, S.: Simulated Annealing for Supervised Gene Selection. *Soft Computing*, pp. 1471–1482 (2011)

33. Gunavathi, C., Premalatha, K.: Cuckoo search optimization for feature selection in cancer classification: a new approach. *Int. J. Data Mining and Bioinformatics*, pp. 248–265 (2015)
34. Bonilla-Huerta, E.: Hybrid Framework using Multiple-Filters and an Embedded Approach for an Efficient Selection and Classification of Microarray Data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (2015)
35. Li, S., Wu, X., Tan, M.: Gene Selection using Hybrid Particle Swarm Optimization and Genetic Algorithm. *Soft Comput.*, pp. 1039–1048 (2008)
36. Yu, G.: Matched Gene Selection and Committee Classifier for Molecular Classification of Heterogeneous Diseases. *Journal of Machine Learning Research*, pp. 2141–2167 (2010)
37. Pang, S., Havukkala, L., Hu, Y., Kasabov, N.: Classification consistency analysis for bootstrapping gene selection. In *Neural Computing and Applications*, 16, pp. 527–539 (2007)

Análisis exploratorio para la caracterización de la adicción a la cocaína a través del aprendizaje computacional

Arturo Téllez-Velázquez ¹, Eduardo A. Garza-Villareal ², Jorge J. González-Olvera ²,
Raúl Cruz-Barbosa ³

¹ CONACyT – Universidad Tecnológica de la Mixteca, Instituto de Computación,
Huaquapán de León, Oaxaca, México

² Instituto Nacional de Psiquiatría Ramón de la Fuente Muñiz, Subdirección de Investigaciones
Clínicas, Ciudad de México, México

³ Universidad Tecnológica de la Mixteca, Instituto de Computación,
Huaquapán de León, Oaxaca, México

atellezv@mixtureco.utm.mx, {egarza, jjgonz}@imp.edu.mx, rcruz@mixtureco.utm.mx

Resumen. Las adicciones son trastornos neuro-psiquiátricos con serias repercusiones en la salud. En México, la cocaína es la segunda droga más usada después de la marihuana. Sin embargo, su uso conlleva a mayor adicción, síndrome de abstinencia y psicopatología. Este estudio tiene como objetivo seleccionar, usando aprendizaje computacional, las variables más representativas de entre las variables demográficas, cognitivas y de impulsividad para predecir la adicción a la cocaína con nuevas muestras. Para lograrlo, se obtuvieron datos de 39 pacientes con adicción a la cocaína y 23 controles sanos. A partir de esto, se obtuvo un 88.24% de exactitud de clasificación, usando el método de selección Relief con tan sólo 28 predictores; mientras que usando todos los descriptores iniciales (61 características) el rendimiento obtenido es bajo. Nuestros resultados sugieren que la selección de un subconjunto características es muy importante, no sólo para reducir el cómputo durante el entrenamiento de los métodos, sino también para obtener resultados de clasificación de adicción a cocaína mejores que los obtenidos al utilizar el conjunto completo de características.

Palabras clave: aprendizaje computacional, adicción, cocaína, impulsividad, selección de características.

Exploratory Analysis for the Characterization of Cocaine Addiction through Machine Learning

Abstract. Addictions are neuro-psychiatric disorders with serious repercussions on health. In Mexico, cocaine is the second most commonly used drug after marijuana. However, its use leads to increased addiction, withdrawal syndrome and psychopathology. This study aims to select, using computational learning, the most representative variables among demographic, cognitive and impulsivity

to predict cocaine addiction with new samples. To achieve this, data were obtained from 39 patients with cocaine addiction and 23 healthy controls. From this, an accuracy of 88.24% of classification was obtained, using the Relief selection method with only 28 predictors; while using all the initial descriptors (61 characteristics) the accuracy obtained is low. Our results suggest that selecting a subset of characteristics is very important, not only to reduce computation during training methods, but also to obtain better results from cocaine addiction classification than those obtained when using the complete set of features.

Keywords: machine learning, feature selection, addiction classification, cocaine, impulsivity.

1. Introducción

En México, los trastornos mentales causados por las adicciones se han venido incrementando en los últimos años, por lo que el gobierno ha destinado más de 6 mil 600 millones de pesos (2007-2012) de los recursos públicos para atender a la gente con este tipo de problemas [16]. La necesidad de caracterizar las enfermedades asociadas a las adicciones es una tarea primordial para entender el problema y darle el seguimiento con terapias e intervenciones adecuadas.

En este sentido, existen diversos estudios que ayudan a los médicos a determinar qué variables intervienen en la caracterización de las adicciones. Inclusive existen problemas psiquiátricos que están correlacionados y que son indicadores que permiten entender de mejor forma el problema. Por ejemplo, la impulsividad es un problema psiquiátrico en la mayoría los trastornos del uso de sustancias (SUD, por sus siglas en inglés *Substance Use Disorder*) y sobre todo en la adicción a la cocaína [9].

El presente artículo presenta una perspectiva desde el punto de vista de aprendizaje computacional, donde se trata el problema de clasificación binaria de la adicción a la cocaína a través de métodos de selección de características.

Los selectores de características, frecuentemente usados en aprendizaje computacional, ayudan a jerarquizar las variables más importantes para resolver problemas de clasificación. Para lograr este objetivo en nuestro estudio, se recolectó información demográfica, cognitiva y de impulsividad de pacientes con adicción a la cocaína y controles sanos en las instalaciones del *Instituto Nacional de Psiquiatría Ramón de la Fuente Muñiz* (INPRF). Esta información (que no está disponible públicamente) fue analizada mediante el uso de selectores de características para obtener un conjunto de características reducido y representativo, con el cual es posible caracterizar las adicciones desde el punto de vista de clasificación de patrones [5]. Con esto, también es posible reducir el costo computacional que conlleva el entrenamiento de los modelos de clasificación.

Los resultados obtenidos muestran que los selectores de características ayudan a incrementar el rendimiento de clasificación de adicción a la cocaína de manera sustancial y ayudan a eliminar información redundante contenida en las variables del registro de cada paciente.

Este artículo está organizado en las siguientes secciones. La sección 2 presenta un breve estado del arte del problema de clasificación y caracterización del problema de

adicción a la cocaína. La sección 3 presenta la metodología utilizada para afrontar el problema de clasificación mediante el uso de selectores de características. La sección 4 presenta los resultados obtenidos con base en la metodología implementada y finalmente la sección 5 destaca las conclusiones y el trabajo futuro.

2. Trabajo relacionado

Existen varios trabajos relacionados donde se utiliza el aprendizaje computacional para afrontar el problema de clasificación a las adicciones. Dada la complejidad de este problema debido a que existen problemas psiquiátricos relacionados, la caracterización de las adicciones se ha tenido que analizar de forma indirecta.

Una característica notable en los adictos a la cocaína es la impulsividad. La impulsividad se ha utilizado como una medida indirecta de la adicción a la cocaína (CA) [1]. Para analizar la impulsividad, en [11] se propuso la *Escala de Impulsividad de Barrat* (BIS) como un instrumento que mide el comportamiento impulsivo de las personas (bajo ciertas circunstancias) para entender su relación con otras comorbilidades clínicas. La escala más utilizada es la BIS-11 [7, 13].

Otra prueba relacionada que ha sido usada para caracterizar la adicción a la cocaína es la *Prueba Eriksen-Flanker* (FT), la cual es usada para probar la habilidad de las personas para suprimir respuestas inapropiadas en un contexto particular [6]. Principalmente, la prueba FT estimula al paciente con el objetivo de medir la cantidad de respuestas incongruentes. También es posible con esta prueba medir el tiempo entre el estímulo y la respuesta. Trabajos recientes, que usan aprendizaje computacional [12], utilizan los resultados de la prueba FT como predictores para determinar la relación entre los SUD y las respuestas incongruentes.

Por otro lado, también existe la *Prueba de Iowa-Gambling* (IGT) [4], que es un instrumento frecuentemente usado para determinar el daño que puede existir en la corteza prefrontal en pacientes con SUD, lo cual dificulta la toma de decisiones debido a la insensibilidad a las consecuencias futuras en sus acciones [3]. Los resultados de esta prueba también han sido usados como predictores en trabajos recientes; por ejemplo, en [2] se ha utilizado para caracterizar los trastornos debidos al consumo de heroína y anfetaminas usando técnicas de aprendizaje computacional.

Otro factor que se toma en cuenta, desde el campo de aprendizaje computacional, es que la selección de características, hasta cierto punto, ayuda a mejorar las tareas de clasificación al facilitar la explicación del problema, al usar un número inferior de variables [14]. Con lo que respecta a la selección de características en el ámbito de la caracterización de las adicciones, investigaciones destacadas [8, 10, 15, 17] obtienen una mejora en rendimiento de clasificación cuando se eliminan variables que no aportan un valor significativo a las predicciones.

3. Metodología

A diferencia de los trabajos relacionados descritos en la sección anterior, nuestra investigación realiza una selección de características para la clasificación de la adicción a la cocaína, misma que sirve para identificar las variables que son más representativas.

Para lograr esto, se recolectó información demográfica, de pruebas cognitivas y de impulsividad de 62 pacientes en las instalaciones del INPRF, donde 39 pacientes son adictos a la cocaína y 23 son controles. En total se utilizaron 61 descriptores / características: 8 demográficas, 4 de impulsividad y 49 cognitivas. Por otro lado, se utilizó una muestra representativa de 17 pacientes para prueba y 45 pacientes para el entrenamiento de los algoritmos de clasificación.

Diversas técnicas de clasificación fueron utilizadas en la configuración de los experimentos y los selectores de características utilizados (implementados en MATLAB) fueron los siguientes:

- Prueba Xi-cuadrada (CHI),
- Selección secuencial hacia adelante (SFS),
- Algoritmo Relief (RLFF) y
- Análisis de componentes principales (PCA).

Para fines de comparación, se utilizan todas las características (ALL) como entrada a los algoritmos de clasificación, con la finalidad de obtener un rendimiento de referencia, con respecto al rendimiento obtenido a partir de un subconjunto de estas.

Tabla 1. Rendimientos promedio obtenidos de los clasificadores usando todas las variables demográficas, cognitivas y de impulsividad.

# caract.	NB			DT			SVM		
	ACC.	SENS	SPEC	ACC.	SENS	SPEC	ACC.	SENS	SPEC
61	0.52941	0.45455	0.66667	0.41176	0.54545	0.16667	0.64706	0.63636	0.66667

Tabla 2. Matrices de confusión correspondientes a la Tabla 1 usando todas las características.

NB			DT			SVM		
C	0	1	C	0	1	C	0	1
0	4	2	0	1	5	0	4	2
1	6	5	1	5	6	1	4	7

Los tres métodos de selección enunciados arriba, y el método de extracción PCA permiten obtener un número finito de características menor que el número de variables originales. Para obtener el mínimo número adecuado de características se ha utilizado un algoritmo de clasificación para encontrar el desempeño máximo de cada algoritmo. La idea es simplemente obtener el mejor rendimiento con el menor número de variables posible.

Una vez obtenido el número adecuado de características, se procede a la evaluación del algoritmo de clasificación adecuado. En este experimento se utilizaron los siguientes algoritmos:

- Bayes ingenuo (NB),
- Árboles de decisión (DT) y
- Máquinas de vectores de soporte lineal (SVM).

Debido a la cantidad limitada de datos de entrenamiento, los tres modelos mencionados arriba utilizan validación cruzada *leave-one-out* para estimar su rendimiento.

Por otro lado, las medidas de rendimiento utilizadas para comparar los clasificadores usados en este estudio son los siguientes:

- *Exactitud* (ACC derivado de la abreviación en inglés Accuracy). Esta medida de rendimiento nos permite evaluar de manera general el desempeño del clasificador y es utilizado para determinar la capacidad de discriminación entre adictos y controles correctamente.
- *Sensibilidad* (SENS derivado de la abreviación en inglés Sensitivity). Esta medida permite conocer el desempeño del clasificador para clasificar correctamente a las personas adictas.
- *Especificidad* (SPEC derivado de la abreviación en inglés Specificity). Esta otra medida se usa en este estudio para conocer el desempeño del clasificador para clasificar correctamente a los controles.

Por último y para una mejor visualización del rendimiento de los algoritmos de clasificación, también se muestra la *matriz de confusión* de los resultados de cada método. Esta matriz permite observar claramente los errores de clasificación y confirmar los resultados de las medidas anteriores (exactitud, sensibilidad y especificidad).

Tabla 3. Rendimiento promedio de los clasificadores usando selección y extracción de características para la clasificación de adicción a la cocaína.

Mét.	# car.	NB			DT			SVM		
		ACC.	SENS	SPEC	ACC.	SENS	SPEC	ACC.	SENS	SPEC
CHI	29	0.52941	0.54545	0.50000	0.58824	0.72727	0.33333	0.88235	0.90909	0.83333
SFS	18	0.58824	0.72727	0.33333	0.58824	0.72727	0.33333	0.64706	0.81818	0.33333
RLFF	28	0.70588	0.72727	0.66667	0.47059	0.63636	0.16667	0.88235	0.81818	1.00000
PCA	54	0.41176	0.54545	0.16667	0.35294	0.45455	0.16667	0.82353	0.81818	0.83333

Tabla 4. Matrices de confusión usando las variables seleccionadas de la Tabla 3.

Método	NB			DT			SVM		
CHI	C	0	1	C	0	1	C	0	1
	0	3	3	0	2	4	0	5	1
	1	5	6	1	3	8	1	1	10
SFS	C	0	1	C	0	1	C	0	1
	0	2	4	0	2	4	0	2	4
	1	3	8	1	3	8	1	2	9
RLFF	C	0	1	C	0	1	C	0	1
	0	4	2	0	1	5	0	6	0
	1	3	8	1	4	7	1	2	9
PCA	C	0	1	C	0	1	C	0	1
	0	1	5	0	1	5	0	5	1
	1	5	6	1	6	5	1	2	9

4. Resultados

De acuerdo con la configuración de los experimentos, primero se utilizaron las 61 características (de las que consta nuestra base de datos) con los clasificadores NB, DT y SVM. Esto con el objetivo de obtener una referencia de rendimiento inicial.

El porcentaje promedio de exactitud de clasificación obtenido fue precario en los tres clasificadores (ver Tabla 1) siendo el mejor el obtenido con SVM, alcanzando un porcentaje de 64.71%. Como es de esperarse, en la misma tabla también se observa que se obtuvo un porcentaje bajo tanto para predecir los adictos (sensibilidad) como los controles (especificidad). Para mayor detalle de los resultados de la Tabla 1, se presenta la matriz de confusión correspondiente en la Tabla 2, donde se observa que el desempeño de SVM es similar al de NB, pero este último tiene mayor número de falsos negativos y por ello su sensibilidad es menor que la sensibilidad obtenida con SVM. Como notación, en las Tablas 2 y 4, los encabezados y primera columna de cada matriz de confusión referidos como 0 y 1 representan a las clases *control* y *adicto*, respectivamente.

Con el objetivo de mejorar el desempeño de los clasificadores y reducir el número de características utilizadas durante el entrenamiento, se utilizaron tres selectores y un extractor de características (todos ellos mencionados en la sección anterior).

Tabla 5. Características seleccionadas con RLFF, SFS y CHI, separados por categoría.

RLFF		SFS		CHI	
DEMOGRÁFICA					
SEX DEGREE LATERALITY		SEX AGE		SEX SCORE DEGREE	
IMPULSIVIDAD					
BIS.INoPI BIS.Total BIS.IMo BIS.ICog		BIS.INoPI BIS.ICog		BIS.INoPI BIS.Total BIS.IMo BIS.ICog	
COGNITIVAS					
F-AI4	IGT-B1	F-TREC	IGT-B5	F-AI4	IGT-B1
F-TREC	IGT-B3	F-TREI		F-DTRC1	IGT-B3
F-TREI	IGT-B2	F-EI		F-AC4	IGT-B2
F-EI4	IGT-TRD	F-AC4		F-AI1	IGT-DTRD
F-EI	IGT-T	F-TRI4		F-TRC	IGT-T
F-DTRI4	IGT-I	F-AC		F-DTRI	IGT-I
F-DTRC1	IGT-RV	F-NRC		F-EC	IGT-RV
F-NRI1	IGT-RD	F-EC4		F-ANG4	IGT-RD
F-AC4	IGT-TRV	F-AI1		F-NRC1	IGT-B5
	IGT-B5	F-NRI		F-ANG	IGT-B4
	IGT-B4	F-DTREI		F-NCI4	IGT-DTRV
	IGT-DTRD	F-EI1			
		F-TRC			

En las filas de la Tabla 3 se presentan los resultados de los cuatro algoritmos de selección y extracción utilizados. La segunda columna de esta tabla muestra el número mínimo de características obtenidas con cada método y las demás columnas muestran los resultados de rendimiento máximo (de exactitud, sensibilidad y especificidad) obtenidos con los tres clasificadores utilizados.

De la misma manera que en la Tabla 1, el desempeño de NB y DT fue muy pobre con cualquiera de los métodos de selección. Como puede observarse también, casi todos los algoritmos de selección y extracción mejoraron el desempeño de clasificación con SVM, en comparación con el uso de las 61 variables de la Tabla 1. Los métodos CHI y RLFF, mismos que han sido usados exitosamente para selección de características en otros trabajos [17, 18], maximizaron el desempeño utilizando solamente 29 y 28 características, respectivamente.

Las matrices de confusión de los resultados de todos los métodos de selección y extracción, usando el clasificador SVM, se presentan en la Tabla 4. A pesar de que los métodos CHI y RLFF obtienen la misma exactitud de clasificación (ver Tabla 3), el método CHI tiene la capacidad de reconocer mejor a los adictos (concretamente para el conjunto de datos analizado), lo cual puede ser decisivo al momento de escoger entre estos dos métodos de selección.

Por otro lado, el extractor PCA, que genera 54 variables nuevas a partir de las 61, obtiene resultados modestos y porcentajes de sensibilidad y especificidad similares. Sin embargo, su desempeño no supera a RLFF o CHI y las nuevas variables no son interpretables directamente, ya que son generadas a partir de la combinación lineal de las originales.

Por último, la Tabla 5 presenta las variables que fueron seleccionadas por los tres métodos de selección de características analizados. Observe que tanto RLFF como CHI toman en cuenta la mayoría de las variables obtenidas con las pruebas de impulsividad y cognitivas, así como aquellas que miden la capacidad de toma de decisión de las personas (BIS e IGT, respectivamente).

En todos los métodos de selección analizados, las variables demográficas resultan ser poco relevantes para la caracterización de la adicción a la cocaína. De acuerdo con los resultados del método de selección de ranking RLFF, las variables de impulsividad resultaron ser las más importantes. De esto se desprende que, los pacientes adictos a la cocaína presentan un déficit en la inhibición de las acciones motivadas por las emociones del momento (impulsividad motora, BIS.IMo); en la inhibición al pensamiento apresurado y sin planificación (impulsividad no planeada, BIS.NoPl) y; en la inhibición de la realización de múltiples conjeturas en corto tiempo (impulsividad cognitiva, BIS.ICog). Estos resultados confirman también que algunos factores de impulsividad primarios y secundarios de la escala de Barrat (BIS-11) son predominantes en pacientes adictos a la cocaína.

5. Conclusiones

Los resultados obtenidos de los experimentos en este trabajo muestran que la selección de características es particularmente útil para obtener aquellas variables psiquiátricas que están fuertemente relacionadas con la adicción a la cocaína y para mejorar el rendimiento en las tareas de clasificación.

Las variables seleccionadas indican que los trastornos que se manifiestan con el incremento de la impulsividad y la incapacidad de la toma de decisiones son relevantes para la caracterización de la adicción a la cocaína.

De manera general, los métodos de selección o extracción de características ayudan a reducir el número de características del registro de cada paciente. Como consecuencia, se puede decrementar el tiempo de cómputo y el uso de memoria asociados al entrenamiento de los modelos de clasificación usados en aprendizaje computacional.

Como trabajo futuro, se buscará enriquecer esta investigación utilizando datos adicionales obtenidos a partir de imágenes de resonancia magnética.

Agradecimientos. Parte de este proyecto fue financiado por CONACYT-Cátedras proyecto No. 1170 y No. 2358948, por CONACYT-FOSISS proyecto No. 0201493 y por la Srita. Ana Teresa Martínez Alanís. Por otra parte, agradecemos a las personas que hicieron posible la parte de adicción de este estudio: Francisco J. Pellicer-Graham, Margarita López-Titla, Aline Leduc, Erik Morelos-Santana, Diego Ángeles, Alely Valencia, Daniela Casillas, Sarael Alcauter, Luis Concha y Bernd Foerster. Agradecemos también a Rocío Estrada-Ordoñez e Isabel Lizarindari Espinosa-Luna de la Unidad de Atención Toxicológica Xochimilco por su apoyo. Finalmente, agradecemos a los participantes por su cooperación y paciencia.

Referencias

1. Ahn, W. Y., Ramesh, D., Moeller, F. G., Vassileva, J.: Utility of Machine-Learning Approaches to Identify Behavioral Markers for Substance Use Disorders: Impulsivity Dimensions as Predictors of Current Cocaine Dependence. *Front. Psychiatry*, 7(34) (2016)
2. Ahn, W. Y., Vassileva, J.: Machine-learning identifies substance-specific behavioral markers for opiate and stimulant dependence. *Drug Alcohol Depend*, 161, pp. 247–257 (2016)
3. Bechara, A., Damasio, A. R., Damasio, H., Anderson, S. W.: Insensitivity to future consequences following damage to human prefrontal cortex. *Cognition*, 50(7), pp. 289–202 (1994)
4. Bechara, A., Tranel, D., Damasio, H.: Characterization of the decision-making deficit of patients with ventromedial prefrontal cortex lesions. *Brain*, 132(7), pp. 289–202 (2000)
5. Duda, R. O., Hart, P. E., Stork, D. G.: *Pattern Classification*. Wiley Interscience Publication, U.S., 2nd edn. (2001)
6. Eriksen, B. A., Eriksen, C. W.: Effects of noise letters upon the identification of a target letter in a nonsearch task. *Perception and Psychophysics*, 16(1), pp. 143–149 (1974)
7. Fehrman, E., Muhammad, A. K., Mirkes, E. M., Egan, V., Gorban, A. N.: *The Five Factor Model of personality and evaluation of drug consumption risk*. Tech. rep., Cornell University Library (2017)
8. Mete, M., Sakoglu, U., Spence, J. S., Devous, M. D., Harris, T. S., Adino, B.: Successful classification of cocaine dependence using brain imaging: a generalizable machine learning approach. In: 13th Annual MCBIOS conference, Vol. 17, BioMed Central (2016)
9. Mitchell, M. R., Potenza, M. N.: Addictions and Personality Traits: Impulsivity and Related Constructs. *Curr. Behav. Neurosci. Rep.*, 1(1), pp. 1–12 (2014)
10. Pariyadath, V., Stein, E. A., Ross, T. J.: Machine learning classification of resting state functional connectivity predicts smoking status. *Front. Hum. Neurosci.*, 8(425) (2014)

11. Patton, J. H., Stanford, M. S., Barrat, E. S.: Factor structure of the Barratt impulsiveness scale. *J. Clin. Psychol.*, 51(6), pp. 768–774 (1995)
12. Plevan, T., Wascher, E., Falkenstein, M., Hoffmann, S.: Classifying Response Correctness across Different Task Sets: A Machine Learning Approach. *PLoS One*, 11(3) (2016)
13. Reise, S. P., Moore, T. M., Sabb, F. W., Brown, A. K., London, E. D.: The Barratt Impulsiveness Scale - 11: Reassessment of its Structure in a Community Sample. *Psychol. Assess*, 25(2), pp. 631–642 (2013)
14. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3, pp. 1157–1182 (2003)
15. Rish, I., Bashivan, P., Cecchi, G. A., Goldstein, R. Z.: Evaluating effects of methylphenidate on brain activity in cocaine addiction: a machine-learning approach. In: *Medical Imaging 2016: Biomedical Applications in Molecular, Structural, and Functional Imaging*, SPIE (2016)
16. Villatoro-Velázquez, J., Medina-Mora, M., Fleiz-Bautista, C., Téllez-Rojo, M. M., Mendoza-Alvarado, L. R., Romero-Martínez, M., Gutiérrez-Reyes, J. P., Castro-Tinoco, M., Hernández-Ávila, M., Tena-Tamayo, C., Alvear-Sevilla, C., Guisa-Cruz, V.: *Encuesta Nacional de Adicciones 2011: Reporte de Drogas*. INPRF, Tlalpan, Ciudad de México (2011)
17. Jin, X., Xu, A., Bie, R., Guo, P.: Machine Learning Techniques and Chi-Square Feature Selection for Cancer Classification Using SAGE Gene Expression Profiles. In: *Proceedings of Data Mining for Biomedical Applications: PAKDD 2006 Workshop, BioDM 2006*, Singapore, pp. 106–115 (2006)
18. Durgabai, R.: Feature Selection using ReliefF Algorithm. *International Journal of Advanced Research in Computer and Communication Engineering*, 3(10), pp. 8215–8218 (2014)

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
octubre de 2017
Printing 500 / Edición 500 ejemplares

