

ISSN: 2007 - 5448



Índice

Computación e Informática

Estudio del Proceso Software Personal (PSP) en un entorno académico.....I

Omar S. Gómez, Antonio A. Aguilera,
Gerzon E. Gómez, Raúl A. Aguilar.

Biomédica

Desarrollo de las úlceras en el pie diabético.....II

Aurora Espinoza Valdez, Beatriz Carely Luna Olivera,
Gualberto Solís Perales.

Interfaz humano-computadora basada en señales de
electrooculografía para personas con discapacidad motriz.....III

Daniel Pacheco Bautista, Ignacio Algreto Badillo,
David De la Rosa Mejía, Aurelio Horacio Heredia Jiménez.



Esta obra está bajo una licencia de Creative Commons
Reconocimiento-NoComercial-CompartirIgual 2.5 México

Estudio del Proceso Software Personal (PSP) en un entorno académico

Omar S. Gómez

Facultad de Matemáticas, Universidad Autónoma de Yucatán,
Anillo Periférico Norte, Tablaje Cat. 13615,
Mérida, Yucatán, México
omar.gomez@uady.mx

Antonio A. Aguilera

Facultad de Matemáticas, Universidad Autónoma de Yucatán,
Anillo Periférico Norte, Tablaje Cat. 13615,
Mérida, Yucatán, México
aaguilet@uady.mx

Gerzon E. Gómez

Facultad de Matemáticas, Universidad Autónoma de Yucatán,
Anillo Periférico Norte, Tablaje Cat. 13615,
Mérida, Yucatán, México
gerzon.gomez@uady.mx

Raúl A. Aguilar

Facultad de Matemáticas, Universidad Autónoma de Yucatán,
Anillo Periférico Norte, Tablaje Cat. 13615,
Mérida, Yucatán, México
avera@uady.mx

Resumen: El Proceso Software Personal (En Inglés, Personal Software Process o PSP) es un proceso individual cuyo objetivo es ayudar a los ingenieros en software a medir y mejorar su productividad personal. El PSP suele enseñarse exclusivamente a profesionales, no obstante éste ha comenzado a incluirse como parte de cursos universitarios. En este trabajo se presenta un estudio sobre PSP efectuado en un entorno académico donde se analizan los siguientes

indicadores: precisión en las estimaciones de tamaño y esfuerzo, calidad del producto así como productividad. Los resultados aquí reportados sugieren una mejoría parcial con respecto a la precisión de las estimaciones, una reducción sustantiva en la densidad de defectos de acuerdo a un factor de 6.6 y una productividad media de 53 líneas de código fuente por hora.

Palabras clave: ingeniería en software; proceso software personal; PSP; medición del software; proceso de mejora continua.

Study of the Personal Software Process (PSP) Under an Academic Setting

Abstract: The Personal Software Process (PSP) is an individual process where its main goal is to help software engineers to measure and to improve their own productivity. PSP courses usually are given to practitioners; nevertheless it has started to be taught in bachelor's and master's degree programs too. In this work, we present a study of the PSP under an academic setting; where the following elements are analyzed: Size and effort precision estimates, product quality and productivity. Results shown suggest a partial improvement regarding size and effort precision estimates; an important decrease, by a factor of 6.6, with respect of defect density; and an average productivity of 53 source lines of code (SLOC) per hour.

Keywords: Software Engineering; Personal Software Process; PSP; Software Measurement; Software Process Improvement.

1. Introducción

Las empresas dedicadas a la construcción de software están en la búsqueda continua de la mejora de sus productos. Un elemento imprescindible en estas empresas son los ingenieros en software, donde cada ingeniero produce una serie de módulos o componentes que son integrados hasta conformar un

sistema software funcional. Si estas empresas desean mantener una alta competitividad, es de suma importancia mejorar la eficiencia, productividad y calidad de los productos software producidos por cada uno de sus ingenieros.

Tomando en cuenta la relevancia del trabajo individual del ingeniero en software, Watts Humphrey desarrolló un proceso de mejora personal siguiendo como referente un modelo para evaluar y mejorar la madurez de las capacidades en organizaciones dedicadas a la construcción de software¹ (Humphrey, 1988, 1989). Humphrey desarrolló este proceso en la década de 1980, en el Instituto de Ingeniería en Software (En Inglés, Software Engineering Institute o SEI). Durante algunos años, Humphrey escribió más de 60 programas donde fue refinando este proceso, posteriormente, comenzó a aplicarlo en grupos de estudiantes adscritos a la Maestría en Ingeniería en Software de la Universidad de Carnegie Mellon. Humphrey llamó a este proceso individual como: Proceso Software Personal² (En Inglés, Personal Software Process o PSP) (Humphrey,1995).

Se dice que PSP ayuda a los ingenieros a mejorar en sus estimaciones con respecto al tamaño y esfuerzo requeridos para construir un componente software. PSP también ayuda a los ingenieros a mejorar la calidad de sus productos reduciendo las tasas de inyección de defectos. Diversas compañías quienes han aplicado los principios del PSP han mostrado un aumento de calidad en sus productos software, así como una reducción en los tiempos de desarrollo (Hayes y Over, 1997; Ferguson et al., 1997; Kelly y Culleton, 1999).

Hasta hace poco tiempo el PSP se enseñaba exclusivamente a profesionales, no obstante, éste ha comenzado a impartirse como parte de cursos universitarios en distintas universidades alrededor del Mundo (Ferguson et al., 1997; Prechelt et al., 1997; Carrington et al., 2001; Maletic et al., 2001; Runeson, 2001; Borstler et al., 2002; Abrahamsson y Kautz, 2002a, b; Hou y Tomayko, 1998; Lisack, 2000).

En México, algunas Instituciones de Educación superior del país³ han comenzado a incluir en sus planes de estudio cursos de PSP, sin embargo, no se han encontrado estudios donde se reporten los resultados de aplicar PSP en instituciones de educación superior mexicanas.

En este trabajo se presentan de manera detallada los resultados del estudio del PSP en un entorno académico de acuerdo a los siguientes indicadores: precisión en las estimaciones de tamaño y esfuerzo, calidad del producto así como productividad. Una versión previa de este trabajo donde se discuten las experiencias de aplicación del PSP se encuentra disponible en Gómez et al. (2014).

Aunque puede cuestionarse que los resultados aquí reportados son de valor limitado para la industria, dado que el estudio se realizó en un entorno académico, diversos autores han observado que el uso de estudiantes universitarios para el estudio de diversos aspectos de la ingeniería en software también son representantes válidos de profesionales en la industria (Jørgensen y Sjøberg, 2001; Höst et al., 2000; Carver et al., 2003; Runeson, 2003; Svahnberg et al., 2008).

El resto del artículo se estructura de la siguiente manera: en la sección 2 se presenta un panorama general del PSP. En la sección 3 se describe el contexto del estudio. En las secciones 4, 5, 6 y 7 se describen los resultados del estudio en términos de la precisión en las estimaciones de tamaño y esfuerzo, calidad del producto así como productividad. En la sección 8 se discuten los resultados encontrados. Por último, en la sección 9 se presentan las conclusiones.

2. El Proceso Software Personal

El PSP se desarrolló para ayudar a los ingenieros en software a hacer bien su trabajo. En éste se enseña al ingeniero a aplicar métodos avanzados de ingeniería a sus tareas diarias. Este proceso proporciona métodos detallados de planificación y estimación donde el ingeniero aprende a tener mayor control

sobre su trabajo con respecto a planes que previamente establece así como le ayuda a producir productos de calidad, reduciendo su número de defectos inyectados (Humphrey, 1995).

De acuerdo a los principios de planificación del PSP, para que los ingenieros sean eficaces deben seguir procesos definidos que sean medibles así como planificar su trabajo. Por otra parte, los principios de calidad del PSP promueven que cada ingeniero realice trabajo de calidad. Para alcanzar esta calidad los ingenieros son responsables de la calidad de los productos que producen, previniendo defectos y haciendo su trabajo de manera correcta (Nichols y Salazar, 2009). Estas mejoras se logran a través de la introducción gradual de nuevos elementos o versiones a la línea base del proceso software personal (Abrahamsson y Kautz, 2002b). La progresión de las distintas versiones del PSP se muestra en la Figura 1.

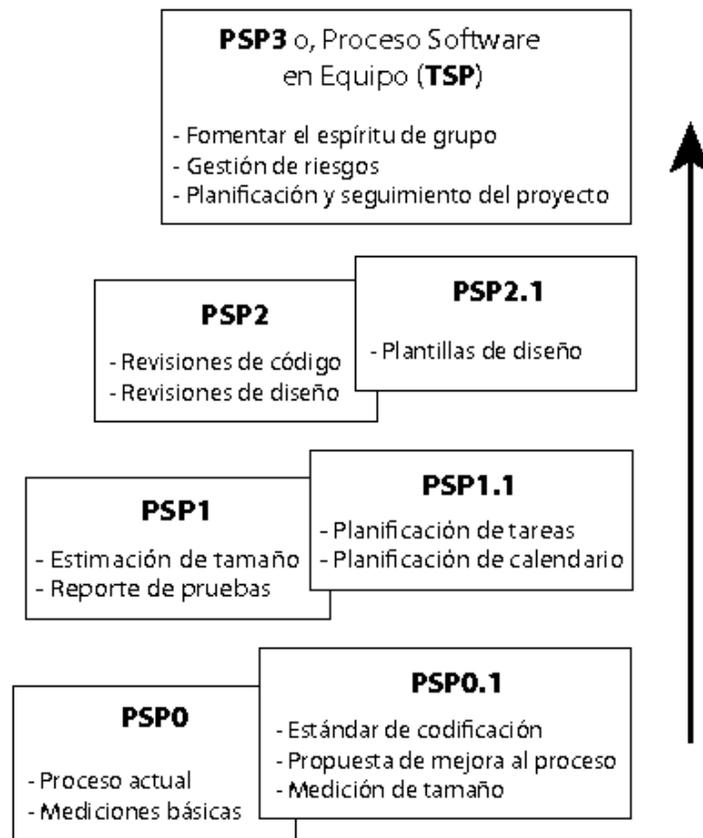


Figura 1. Versiones del proceso PSP (adaptado de Humphrey [2005])

Durante un curso de PSP el ingeniero realiza varios ejercicios de programación así como una serie de informes. Este proceso se introduce en seis versiones ascendentes y compatibles. Los participantes inscritos en este curso escriben uno o dos programas en cada versión del proceso, donde recopilan y analizan distintas mediciones. Posteriormente, utilizan las mediciones recolectadas para realizar diversos análisis con el fin de mejorar su desempeño (Nichols y Salazar, 2009).

El ingeniero inscrito en un curso de PSP comienza con la versión inicial del proceso identificada como PSP0, es decir, usa su proceso actual de desarrollo de software y sólo registra sus tiempos y defectos de los programas que realiza. PSP0 se compone de 3 fases que son: planificación, desarrollo (que incluye el diseño, la codificación y la verificación) así como postmortem. También se establece una línea base para llevar a cabo la medición actual del proceso: tiempo dedicado a la programación, fallos inyectados y eliminados, así como el tamaño del programa medido en líneas de código fuente (LOC). La fase de planificación se usa para documentar el plan del producto a construir (ejercicio de programación). En la fase de desarrollo se realizan varias actividades como son: diseño, codificación, compilación y pruebas. En la fase de postmortem se complementa el plan realizado en la primera fase de acuerdo a las mediciones obtenidas tras construir el producto.

PSP0.1 extiende el proceso mediante la adición de un estándar de codificación y de una propuesta de mejora al proceso (En Inglés, Process Improvement Proposal o PIP) que le facilita al ingeniero identificar acciones de mejora para obtener mayor desempeño en su proceso software personal.

En las versiones PSP1 y PSP1.1 (estimación y planificación), de acuerdo a las mediciones recolectadas en PSP0 y PSP0.1, el ingeniero estima el tamaño del producto a construir así como prepara un informe con casos de prueba (PSP1). Las mediciones acumuladas en los ejercicios anteriores se emplean para estimar el tiempo requerido para construir el siguiente producto. En cada producto que

se construye se registran los tiempos de las distintas fases del proceso (planificación, desarrollo y postmortem) así como el tamaño del producto (medido en LOC). Esta información se utiliza para estimar el tamaño y esfuerzo necesarios para construir el siguiente producto. En PSP1.1 se añade a la anterior base del proceso (PSP1) la planificación y calendarización de las actividades definidas en las fases del proceso.

Por otra parte, en PSP2 y PSP2.1 (gestión de calidad y diseño) se añaden dos actividades a la fase de desarrollo: revisión de diseño y revisión de código. La prevención y eliminación de defectos es la parte central de PSP2. En esta versión del proceso, los ingenieros aprenden a evaluar y mejorar sus estimaciones, así como a mejorar la calidad de sus productos. Los ingenieros elaboran y usan listas de comprobación para revisar tanto sus diseños como sus códigos fuente. Finalmente, PSP2.1 introduce técnicas de especificación de diseño y análisis formales con el fin de reducir los defectos inyectados durante el diseño de sus productos.

3. Contexto del estudio

Una vez descrita de manera general la estructura del PSP, a continuación se detalla el contexto donde se realizó este estudio.

El estudio se realizó en una de las asignaturas del programa de licenciatura en Ingeniería de Software (IS) ofertado en la Facultad de Matemáticas (FMat) de la Universidad Autónoma de Yucatán (UADY).

El año académico en la UADY consta de dos semestres donde cada semestre suele dividirse en 16 semanas. El estudio se llevó a cabo durante el semestre agosto-diciembre del año 2012.

La FMat-UADY oferta el curso PSP en el programa de licenciatura en (IS) como parte de una serie de cursos optativos que complementan la formación del estudiante. Los cursos optativos se organizan en áreas de concentración, donde

en este caso el curso PSP pertenece al área de concentración definida como mejora del proceso software. Los cursos optativos se imparten en los últimos semestres de la carrera, una vez que el estudiante ha cubierto la mayoría de sus cursos obligatorios. En este estudio, el curso PSP se impartió a estudiantes de séptimo semestre de la carrera.

El curso PSP donde se realizó el estudio se dividió en 16 semanas, donde semanalmente se impartieron dos sesiones (lunes y miércoles) con una duración de dos horas por sesión. Durante el curso los estudiantes desarrollaron ocho programas así como elaboraron dos reportes que entregaron a la mitad y al final del curso. En la Tabla 1 se muestra la descripción de los ejercicios de programación empleados, así como se muestra el promedio de las líneas de código fuente codificadas así como el esfuerzo promedio medido en minutos que les llevó a los estudiantes realizar cada programa.

Programa	Versión PSP	Descripción	Tamaño (LOC)	Esfuerzo (Mins)
1	v0	Cálculo de promedio y desviación estándar de un conjunto de números almacenados en una lista enlazada.	140.1	169.9
2	v0.1	Contador de líneas de código fuente.	218.14	339.71
3	v1	Cálculo de los parámetros de regresión lineal β_0 y β_1 así como los coeficientes de correlación r y r^2 dado un conjunto de pares de valores.	249.36	283
4	v1.1	Cálculo de rangos, ya sea medidos en LOCs o páginas de acuerdo con los siguientes tamaños relativos: muy pequeño, pequeño, mediano, grande, muy grande.	231.21	189.43
5	v2	Integración numérica de una función empleando la regla de Simpson.	123.07	219.57
6	v2.1	Análisis semántico y sintáctico de un programa que funciona como línea de comandos.	267.29	327.57
7	v2.1	Uso de operaciones sobre una tabla de símbolos, como son: inserción, asignación, búsqueda e impresión en pantalla de la tabla de símbolos.	285.79	261.71
8	v2.1	Uso de operaciones para gestionar un árbol n-ario.	297	274.93

Tabla 1. Listado de los programas usados en este estudio así como el promedio del tamaño (medido en LOC) y promedio del esfuerzo (medido en minutos).

Como material para la impartición del curso se usaron dos de los libros del autor quien desarrolló este proceso (Humphrey, 1995, 2005). La meta del curso consistió en aprender todo el proceso PSP (hasta la versión 2.1).

De un total de 19 estudiantes inscritos al curso, sólo 14 lo completaron de manera satisfactoria; se tuvo una tasa de deserción del 26%. Los estudiantes inscritos en el curso iniciaban el cuarto año de la carrera (séptimo semestre).

Como herramienta de soporte para la recolección de mediciones se empleó Process Dashboard (Tuma Solutions, 1998). Se decidió en este curso usar una herramienta con el fin de reducir el esfuerzo que conlleva recolectar manualmente las mediciones. Una vez descrito el contexto del estudio en las

siguientes secciones se describen los resultados obtenidos con respecto a la precisión en las estimaciones de tamaño y esfuerzo, calidad del producto así como productividad.

4. Resultados con respecto a la precisión en las estimaciones de tamaño

En PSP la estimación de tamaño se realiza con el fin de obtener un estimado preciso del esfuerzo requerido para desarrollar un producto. Para medir el tamaño se emplean LOCs, ya que existe cierta evidencia que sugiere una correlación entre las LOCs y el esfuerzo de desarrollo (Humphrey, 1995, 2005).

Con respecto a esta correlación, en la Tabla 2 se muestra el valor obtenido del coeficiente de correlación⁴ r y del coeficiente de determinación⁵ r^2 de cada estudiante que aplicó PSP. Estos coeficientes se calcularon a partir de las mediciones que representan el total de LOCs de los ocho programas, así como el esfuerzo total que le llevó a cada estudiante desarrollar los programas.

Estudiante	Coeficiente de correlación r	Coeficiente de determinación r^2
1	0.851	72.42
2	0.5342	28.54
3	0.4396	19.32
4	0.7671	58.84
5	0.7041	49.58
6	0.3713	13.79
7	0.3196	10.21
8	0.1956	3.82
9	0.7076	50.07
10	0.3507	12.3
11	0.5731	32.85
12	0.5176	26.79
13	0.4702	22.11
14	-0.4122	16.99

Tabla 2. Correlaciones observadas entre el tamaño y el esfuerzo.

Por ejemplo, las mediciones de tamaño y esfuerzo del estudiante 1, indican un coeficiente de determinación $r^2=72.42\%$, es decir, que el 72.42% de la variación total en el esfuerzo de desarrollo (variable dependiente) puede ser explicada por una relación lineal entre el tamaño (variable independiente) y el esfuerzo. El otro 27.58% de variación restante permanece sin explicar. Como se observa en la Tabla 2, sólo las mediciones de los estudiantes 1, 4, 5 y 9 pueden explicar aproximadamente el 50% o más de variación entre estas dos variables.

En la Figura 2 se muestran las correlaciones de los estudiantes 1, 8, 4 y 14. Como se observa en esta figura, las mediciones de los estudiantes 1 y 4 describen una correlación positiva “fuerte”. En el caso de las mediciones del estudiante 8 no se observa una correlación. Respecto al estudiante 14 sus mediciones indican una correlación negativa.

Continuando con las estimaciones referentes al tamaño, cabe señalar que los estudiantes obtuvieron sus estimaciones a partir de las mediciones que ellos previamente recolectaron de los programas que desarrollaron. Al comienzo del proceso (PSP versión 0.1) las estimaciones tienden a variar de manera considerable, no obstante según evidencia encontrada en Hayes (1998), conforme madura el proceso (PSP versiones 2 y 2.1) esta variación tiende a estabilizarse dentro de un 25% de margen de error, es decir la precisión en las estimaciones aumenta. En la Figura 3 se muestra un diagrama de cajas⁶ con la distribución de la precisión de las estimaciones referentes al tamaño de los programas desarrollados (programas 2 a 8 que representan las versiones 0.1 a 2.1 del PSP⁷).

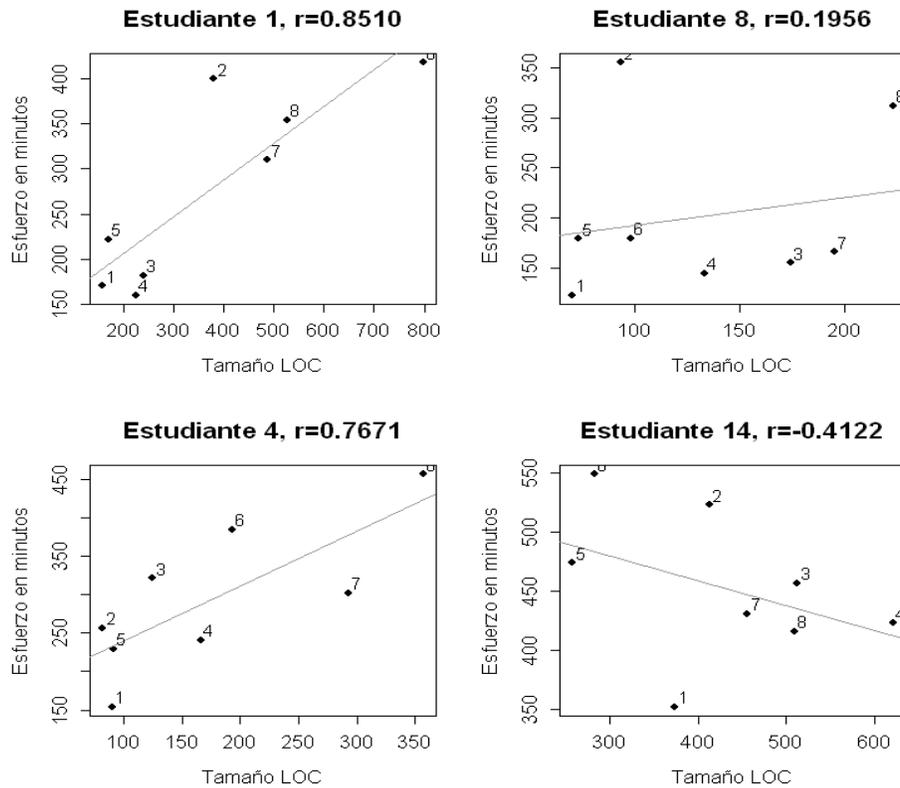


Figura 2. Gráficos de correlaciones entre tamaño y esfuerzo en cuatro estudiantes.

Como se observa en la Figura 3, al inicio del proceso los estudiantes tienden a sobrestimar su trabajo, no obstante conforme madura el proceso, las estimaciones tienden a estabilizarse hacia cierto grado de subestimación. Tomando como valor de referencia el 25% de margen de error en las estimaciones, a excepción del programa 2 y 6 se observa que las medianas del resto de los programas se aproximan o están dentro de este valor de referencia.

Respecto a los programas 6, 7 y 8 (versión 2.1 del proceso), el 36%, 50% y 71% de los estudiantes fueron capaces de estimar dentro del 25% de margen de error mencionado en Hayes (1998).

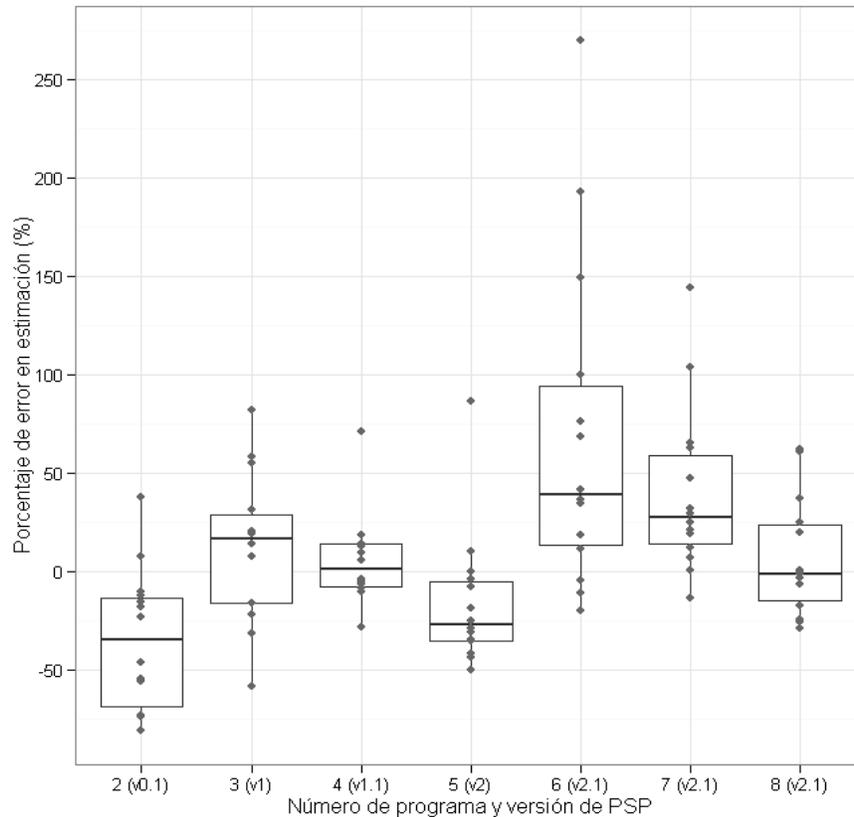


Figura 3. *Precisión en estimaciones con respecto al tamaño.*

Para llevar a cabo el cálculo del tamaño, en PSP se contabilizan diferentes tipos de LOCs como son:

- Líneas de Código Fuente Base. Son aquellas LOCs que no han sufrido modificaciones y que están sujetas a futuros cambios. Por ejemplo, en un programa a desarrollar no habría LOCs base.
- Líneas de Código Añadidas. Son las LOCs nuevas que se añaden a las LOCs base. Por ejemplo, en un programa a desarrollar no habría LOCs base, sólo LOCs añadidas.
- Líneas de Código Modificadas. Es parte de las LOCs base que se han modificado.
- Líneas de Código Borradas. Se refiere a una parte de LOCs base que se han eliminado y que no serán usadas en futuras versiones del programa.

- Líneas de Código Re-utilizadas. Se mantienen sin modificar y son copiadas a un nuevo programa, por ejemplo, funciones, métodos o librerías. Si éstas LOCs sufren alguna modificación por mínima que sea, dejan de contarse como re-utilizadas y se añaden a las LOCs base, mientras que las LOCs cambiadas se cuentan como modificadas.

En la Figura 4 se muestra un diagrama de barras con los distintos tipos de LOCs registrados por los estudiantes tanto en la versión inicial como final del PSP. Al inicio del proceso (PSP versión 0.1) la mayor parte de LOCs fueron añadidas (87%), seguidas de LOCs re-utilizadas (12%), sólo un 1% de las LOCs se emplearon como base. Un patrón similar se observa en la versión 2.1 del proceso, donde se observa un 82% de LOCs añadidas y un 8% de LOCs re-utilizadas. En esta versión del proceso el uso de LOCs base se incrementó en un 7%, mientras que se modificó el 2% de LOCs y se eliminó un 1% de éstas.

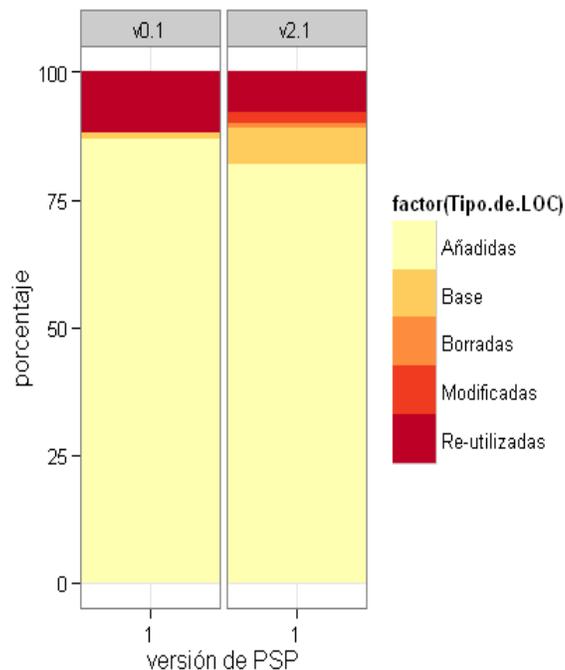


Figura 4. Tipos de LOCs empleadas en la versión inicial y final del PSP.

5. Resultados con respecto a la precisión en las estimaciones del esfuerzo

PSP usa las estimaciones referentes al tamaño para calcular el esfuerzo estimado en completar un producto. En la Figura 5 se muestra un diagrama de cajas con la distribución de la precisión en las estimaciones afines al esfuerzo. De manera similar que en las estimaciones con respecto al tamaño, se observa que al inicio del proceso los estudiantes tienden a sobrestimar el esfuerzo de construcción del producto, no obstante conforme madura el proceso sus estimaciones tienden a mejorar.

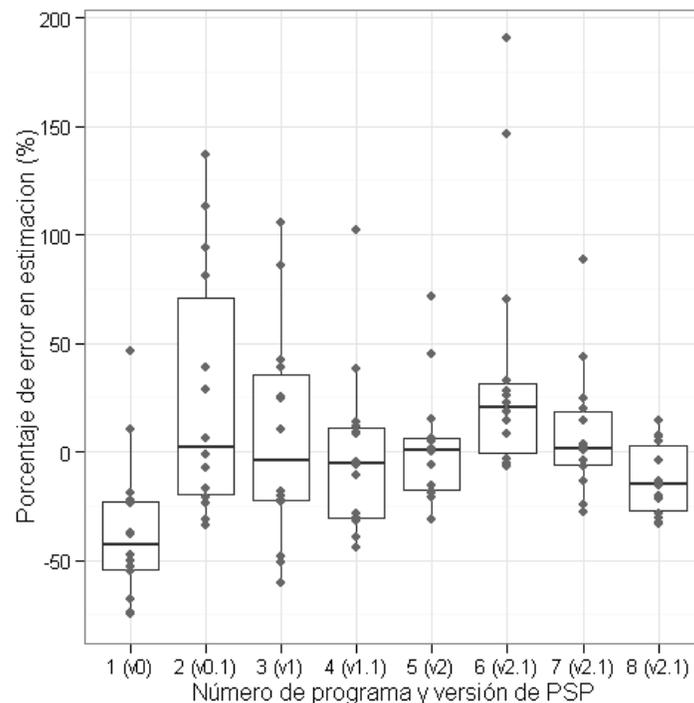


Figura 5. Error en estimación respecto al esfuerzo.

Tomando como valor de referencia el 25% de margen de error en las estimaciones, con excepción del programa 1, se observa que las medianas del resto de los programas se encuentran dentro de este valor de referencia.

Con respecto a la versión 2.1 del proceso, el 64%, 79% y 71% de los estudiantes que desarrollaron los programas 6, 7 y 8 (respectivamente) fueron capaces de estimar dentro un 25% de margen de error.

Como se mencionó en la sección 2, cada versión del PSP se conforma de tres fases que son: planificación, desarrollo y postmortem. La fase de planificación se usa para documentar el plan del producto a realizar. En la fase de postmortem se complementa el plan realizado en la fase de planificación de acuerdo a las mediciones obtenidas tras desarrollar el producto. En la fase de Desarrollo se realizan actividades como: diseño, codificación, compilación y pruebas. A partir de la versión 2 se introducen dos nuevas actividades en la fase de desarrollo que son: revisión de diseño y revisión de código.

En la Figura 6 se muestra un diagrama de barras con la distribución del tiempo empleado en las distintas fases y actividades del PSP, tanto en su versión inicial (PSP 0.1) como final (PSP 2.1). Como se observa en esta figura, con respecto a la fase de desarrollo, aproximadamente la mitad del esfuerzo (48%) se usó en codificación, mientras que un 24% se empleó en verificar el producto, un 16% del esfuerzo se usó en el diseño y sólo un 1% en compilación⁸. En las fases de planificación y postmortem se tuvo un esfuerzo del 6% y 4%, respectivamente.

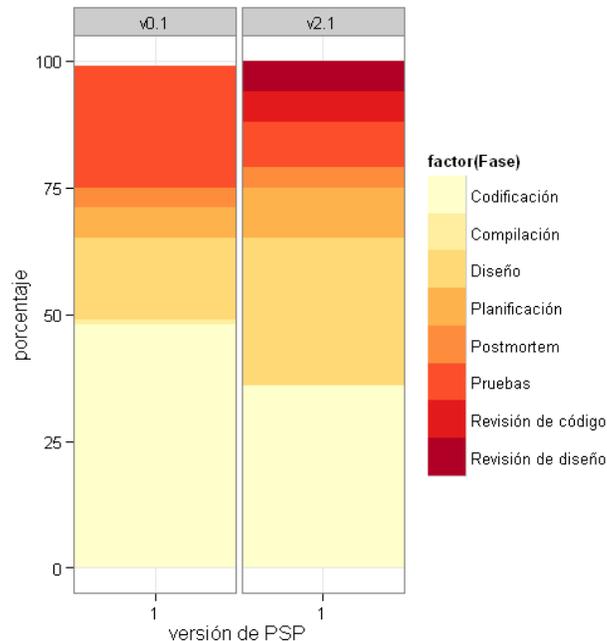


Figura 6. Esfuerzo empleado en las diferentes fases del proceso.

Conforme los estudiantes avanzan en su proceso software se añade mayor énfasis en la calidad del producto, tal como se observa en la Figura 6. A diferencia de la versión PSP 0.1, en la versión PSP 2.1 se observa que es mayor el esfuerzo empleado en planificación (10%) y diseño (29%). Por otra parte, se observa una disminución del esfuerzo en codificación (36%) y pruebas (9%). Respecto a las revisiones de diseño y código se observó un esfuerzo del 6% en ambas actividades.

6. Resultados con respecto a la calidad del producto

El otro componente principal del PSP es el referente a la calidad, donde se promueve la habilidad de encontrar y remover defectos en etapas tempranas del proceso de desarrollo software. Las revisiones de diseño y de código son actividades que se incorporan en el PSP a partir de la versión 2 del proceso. Al incluir estas actividades en etapas tempranas, se espera que los defectos

encontrados en compilación y pruebas disminuyan de manera considerable, aumentando así la calidad del producto.

En la Figura 7 se muestra un diagrama de cajas con la distribución de los defectos removidos por cada mil LOCs (o KLOCs) durante la actividad de verificación o pruebas. Como se observa en esta figura, los defectos tienden a reducirse conforme madura el proceso.

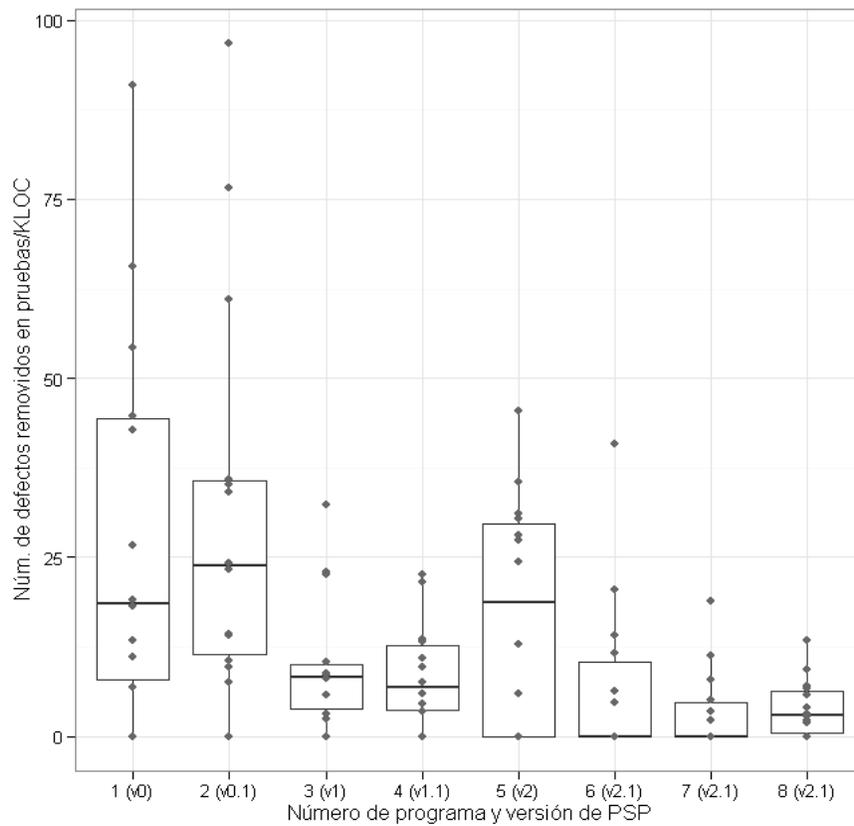


Figura 7. Defectos removidos por cada mil LOCs en la actividad de pruebas.

De acuerdo a Humphrey (1995), los defectos pueden clasificarse en una de las siguientes categorías:

- Asignación. Declaración, nombres duplicados, ámbito, límites.
- Construcción. Librerías, control de versión.
- Comprobación. Mensajes de error, verificaciones inadecuadas.
- Datos. Estructura, contenido.
- Documentación. Comentarios, mensajes.
- Entorno. Diseño, compilación, pruebas y otros problemas que soporta el sistema.
- Función. Lógica, punteros, ciclos o bucles, recursión, computación, defectos de función.
- Interfaz. Llamadas a procedimientos y referencias, E/S, formatos de usuario.
- Sintaxis. Ortografía, puntuación, erratas, formato de las instrucciones
- Sistema. Configuración, temporización, memoria.

En total, los estudiantes removieron 886 defectos. En la Figura 8 se muestra la distribución de los distintos tipos de defectos removidos. Los tipos de defectos que más veces se eliminaron fueron aquellos de tipo entorno (273), función (240), datos (126) y sintaxis (115). Por otra parte, los tipos de defectos que menos veces se removieron fueron los de tipo construcción (2) así como defectos de tipo sistema (2).

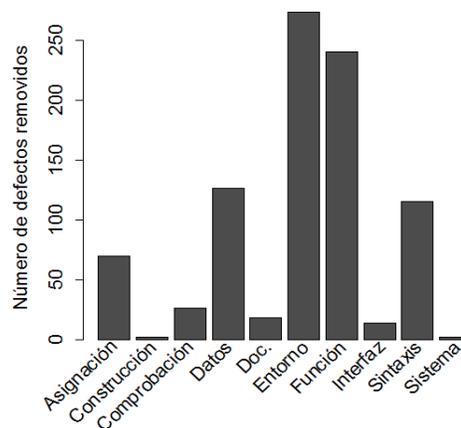


Figura 8. Frecuencia de los tipos de errores removidos.

Una de las mediciones importantes del PSP con respecto a la calidad es el porcentaje de remoción de defectos antes de compilación. Esta medición identificada como *yield* se emplea para examinar la calidad del proceso. El *yield* indica qué tan bien funciona el proceso actual para prevenir que los defectos entren en etapas posteriores del proceso. Lo ideal es contar con un *yield* cercano al 100% lo que indica que los defectos se encuentran y remueven antes de las actividades de compilación y pruebas, aumentando de esta manera la calidad del producto resultante.

Se tiene cierta evidencia que sugiere un aumento del *yield* conforme madura el proceso (Hayes y Over, 1997) debido a que se incluyen otras actividades en la fase de desarrollo como son: revisiones al diseño y al código fuente.

En la Figura 9 se muestra un diagrama de cajas con la distribución de yields en cada uno de los programas. Como se observa en esta figura, a partir de la inclusión de las actividades de revisión en la versión 2 el *yield* aumenta de aproximadamente 55% a un 80%, es decir, se observa un incremento favorable de un 25% al añadir las revisiones de diseño y de código.

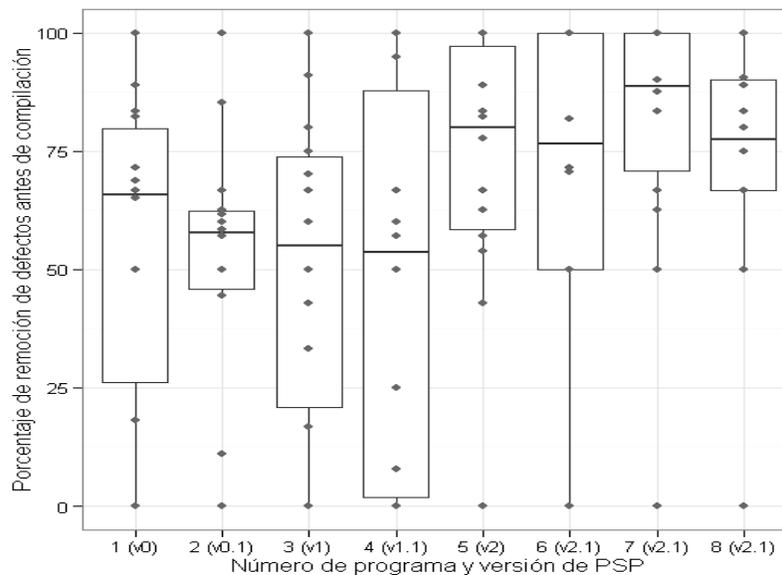


Figura 9. Porcentaje de remoción de defectos antes de compilación (*yield*) en los distintos programas.

7. Resultados con respecto a la productividad

A nivel empresarial, la productividad es un indicador relevante para cuantificar los productos o servicios generados en términos del tiempo requerido para producirlos. En PSP la productividad se mide de acuerdo a las LOCs por hora (LOCs/Hora) que produce un ingeniero, siendo esta medición un indicador de la productividad individual. En la Figura 10 se muestra el diagrama de cajas con la distribución de la productividad registrada de los estudiantes en cada programa.

Como se observa en esta figura, las medianas presentan cierto grado de variabilidad en los ocho programas. En el programa 5 se observó la menor productividad (28 LOCs/Hora) mientras que la mayor productividad se presentó en el programa 4 (73 LOCs/Hora). De manera general, se observó una productividad aproximada de 53 LOCs/Hora.

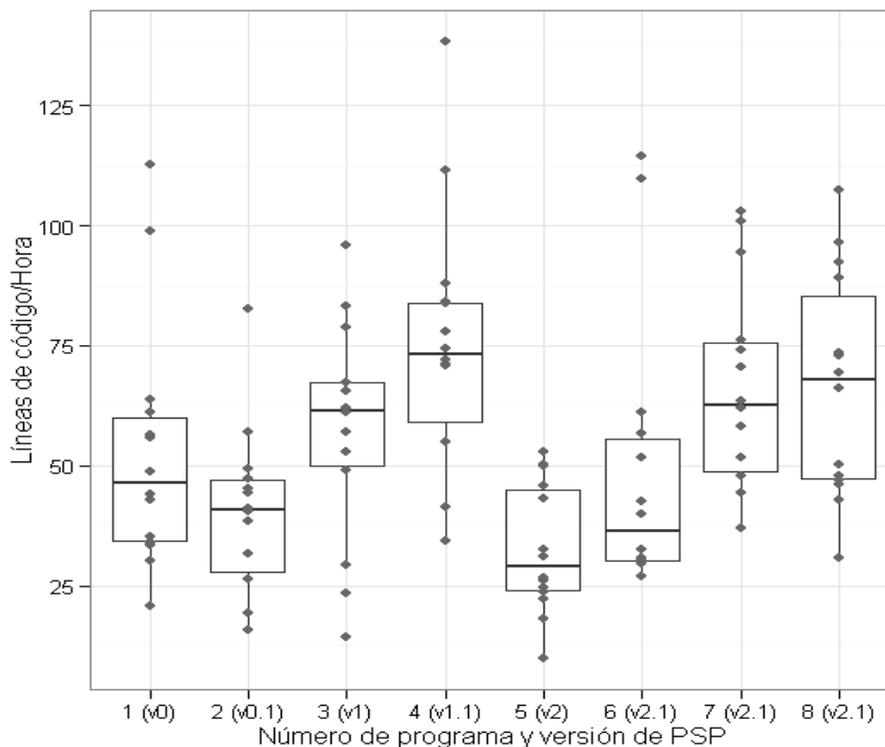


Figura 10. Productividad observada en los distintos programas

8. Discusión

Los resultados observados con respecto a la precisión en las estimaciones de tamaño y esfuerzo sugieren que cierto porcentaje de estudiantes mejora en sus estimaciones durante las distintas versiones del proceso. A diferencia de las estimaciones con respecto al tamaño, las estimaciones afines al esfuerzo presentan menor variación por lo que tienden a incrementar su precisión conforme madura el proceso. Este aumento de precisión con respecto al esfuerzo se corrobora de acuerdo a los resultados reportados en Hayes (1998).

Se dice que las LOCs correlacionan razonablemente bien con el esfuerzo de desarrollo (Humphrey, 1995), no obstante de acuerdo a los resultados obtenidos esta afirmación se corrobora parcialmente en un 28%, sólo en las mediciones de cuatro de los catorce estudiantes se obtuvieron coeficientes de determinación iguales o superiores al 50%.

Con respecto a la calidad, se observa una disminución en la densidad de defectos en la fase de pruebas de aproximadamente 20 defectos por cada mil LOCs (al inicio del proceso) a aproximadamente 3 defectos por cada mil LOCs. Es decir, se observó un factor de mejora de 6.6. Por su parte, en el estudio realizado por Hayes (1998), el autor reporta un factor de mejora de 2.5, mientras que Abrahamsson y Kautz (2002a) reportan un factor de mejora de 4.2.

De acuerdo al porcentaje de remoción de defectos antes de compilación o *yield*, se observa un incremento conforme madura el proceso. Esta tendencia se corrobora de acuerdo con los resultados reportados en Hayes (1998).

En cuanto a la productividad, se observa una variabilidad en las distintas versiones del proceso, no obstante, no se observa una mejora en la productividad en las distintas versiones del proceso, es decir, no se tuvo una ganancia o pérdida sustantiva de la productividad. Este hallazgo se corrobora de acuerdo a los resultados reportados en Hayes (1998).

9. Conclusiones

El PSP es un proceso donde el ingeniero en software aprende a controlar y desarrollar su propio proceso de desarrollo de software. No obstante como se discute en Morisio (2000) y Cannon (1999), el clima organizacional es un factor determinante para que el ingeniero en software continúe aplicando prácticas disciplinadas como las descritas en el PSP.

En este trabajo de investigación se presentaron los resultados obtenidos de estudiar el Proceso Software Personal en un contexto académico. Este estudio se realizó durante el semestre agosto-diciembre 2012 en la Licenciatura en IS de la FMat-UADY.

Los resultados de este estudio indican una ligera mejoría con respecto a las estimaciones de tamaño y esfuerzo. Estos resultados probablemente se deban a que los estudiantes no cuentan con suficiente experiencia codificando programas informáticos. Referente a la calidad del producto, los defectos removidos en la actividad de verificación se redujeron en un factor de 6.6. Se observa que el uso de actividades preventivas como son las revisiones de diseño y de código demuestran ser efectivas.

Para finalizar, a continuación se listan algunas recomendaciones a tener en cuenta en la aplicación del PSP en un entorno académico:

- Se recomienda enseñar PSP en los últimos semestres de la carrera, una vez que el estudiante cuente con cierta experiencia en algún lenguaje de programación.
- Se recomienda que el estudiante cuente con conocimientos de probabilidad e inferencia estadística. Esto con el fin de facilitar el aprendizaje de las diferentes técnicas estadísticas en las que se apoya el PSP.
- Es recomendable que el profesor supervise la fase de planificación en clase para asegurarse que se han registrado de manera correcta las estimaciones.

- Se recomienda que al menos una parte de los ejercicios de programación asignados se trabajen en clase, de tal manera que el profesor pueda supervisar el proceso de desarrollo del estudiante.
- Es recomendable el uso de alguna herramienta que apoye en el proceso de recolección de mediciones. Esto facilita centrarse en el aprendizaje del proceso y no en el cálculo manual de las distintas mediciones del PSP.

Notas

¹ A este modelo se le conoce como **Modelo de Madurez de Capacidades**, en Inglés, **Capability Maturity Model** o **CMM**.

² Otras traducciones en castellano comunes a este término son: **proceso de desarrollo de software personal**, **proceso de software personal** y **proceso personal de software**.

³ Tales como: **Universidad Autónoma de Yucatán**, el **Centro de Investigación en Matemáticas**, **Instituto Tecnológico y de Estudios Superiores de Monterrey**, la **Universidad Autónoma de Zacatecas**, la **Universidad Autónoma de Nuevo León**, entre otras.

⁴ El coeficiente de correlación r mide el grado de asociación y dirección entre dos conjuntos de datos (que generalmente representan dos tipos de variable: dependiente $[y]$ e independiente $[x]$). El valor de este coeficiente puede variar entre -1.0 a $+1.0$, donde valores cercanos a ± 1.0 indican una correlación lineal (positiva o negativa). Por el contrario valores cercanos a 0 indican la ausencia de correlación entre los dos conjuntos de datos. Se dice que una correlación es “fuerte” cuando su valor es mayor o igual que ± 0.8 . Por el contrario, ésta se describe como “débil” cuando su valor es menor que 0.5 (en el caso de la correlación lineal positiva, o mayor a -0.5 cuando ésta es negativa).

⁵ El coeficiente de determinación r^2 indica la fluctuación de los valores de la variable dependiente que se pueden explicar en una relación lineal con los valores de la variable independiente. Los valores de este coeficiente oscilan entre 0 y 1 (o en caso de representarse como porcentaje, 0% y 100%).

⁶ Un diagrama de caja es un tipo de gráfico donde se representan los cuartiles de un conjunto ordenado de datos. Este diagrama se compone por un rectángulo o “caja” y dos brazos o “bigotes”. La caja representa el 50% de los datos, la línea que divide la caja representa la mediana, mientras que cada

brazo representa el 25% de los datos. Este diagrama es útil para identificar valores atípicos así como evaluar la simetría de la distribución de los datos.

⁷ El primer programa se ha omitido ya que de acuerdo al proceso en éste no se realiza una estimación del tamaño.

⁸ El esfuerzo empleado en compilación es mínimo ya que los estudiantes trabajaron con un entorno de desarrollo integrado (en Inglés, IDE) que les informa sobre errores de sintaxis que cometen.

Referencias

Abrahamsson, P. and Kautz, K. (2002a). Personal Software Process: Classroom Experiences from Finland. In Kontio, J. and Conradi, R., editors, *Software Quality -- ECSQ 2002*, volume 2349 of *Lecture Notes in Computer Science*, pp. 175--185. Springer Berlin Heidelberg.

Abrahamsson, P. and Kautz, K. (2002b). The Personal Software Process: Experiences from Denmark. In *Euromicro Conference, 2002. Proceedings. 28th*, pp. 367--374.

Borstler, J., Carrington, D., Hislop, G., Lisack, S., Olson, K., and Williams, L. (2002). Teaching PSP: challenges and lessons learned. *Software, IEEE*, 19(5):42--48.

Cannon, R. L. (1999). Putting the Personal Software Process into practice. *Conference on Software Engineering Education and Training*, 0:34.

Carrington, D., McEniery, B., and Johnston, D. (2001). PSPsm in the large class. In *Software Engineering Education and Training, 2001. Proceedings. 14th Conference on*, pp. 81--88.

Carver, J., Jaccheri, L., Morasca, S., and Shull, F. (2003). Issues in using students in empirical studies in software engineering education. In *METRICS '03: Proceedings of the 9th International Symposium on Software Metrics*, page 239, Washington, DC, USA. IEEE Computer Society.

Ferguson, P., Humphrey, W., Khajenoori, S., Macke, S., and Matvya, A. (1997). Results of applying the Personal Software Process. *Computer*, 30(5):24--31.

Gómez, O. S., Gómez G. E., Aguilera, A. A. and Aguilar, R. A. (2014). Proceso de Software Personal en la Academia: Experiencias de Aplicación en México. In Zapata, C. M. and González, G., editors, *Ingeniería de Software e Ingeniería del Conocimiento: Dos Disciplinas Interrelacionadas*, pp. 197—213. Universidad de Medellín.

Hayes, W. (1998). Using a Personal Software Process to improve performance. In *Software Metrics Symposium, 1998. Metrics 1998. Proceedings. Fifth International*, pp. 61--71.

Hayes, W. and Over, J. (1997). Personal Software Process (PSP): An empirical study of the impact of PSP on individual engineers. Technical Report CMU/SEI-97-TR-001, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

Hou, L. and Tomayko, J. (1998). Applying the Personal Software Process in CS1: An Experiment. In Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education, SIGCSE '98, pp. 322--325, New York, NY, USA. ACM.

Humphrey, W. (2005). PSP(sm): A Self-improvement process for software engineers. Addison-Wesley Professional, First edition.

Humphrey, W. S. (1988). Characterizing the software process: A maturity framework. IEEE Software, 5(2):73--79.

Humphrey, W. S. (1989). Managing the Software Process. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Humphrey, W. S. (1995). A Discipline for Software Engineering. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Höst, M., Regnell, B., and Wohlin, C. (2000). Using students as subjects -- a comparative study of students and professionals in lead-time impact assessment. Empirical Softw. Engg., 5(3):201--214.

Jørgensen, M. and Sjøberg, D. I. K. (2001). Software process improvement and human judgement heuristics. Scand. J. Inf. Syst., 13:99--121.

Kelly, D. and Culleton, B. (1999). Process improvement for small organizations. Computer, 32(10):41--47.

Lisack, S. (2000). The Personal Software Process in the Classroom: student reactions (an experience report). In Software Engineering Education and Training, 2000. Proceedings. 13th Conference on, pp. 169--175.

Malefic, J. I., Marcus, A., and Howald, A. (2001). Incorporating PSP into a traditional software engineering course: An experience report. In Proceedings of the 14th Conference on Software Engineering Education and Training, CSEET '01, pp. 89--97, Washington, DC, USA. IEEE Computer Society.

Morisio, M. (2000). Applying the PSP in industry. Software, IEEE, 17(6):90--95.

Nichols, W. and Salazar, R. (2009). Deploying TSP on a national scale: An experience report from pilot projects in Mexico. Technical Report CMU/SEI- 2009-TR-011, Software Engineering Institute, Carnegie Mellon, Pittsburgh, PA.

Prechelt, L., Unger, B., and Gramberg, O. (1997). Experience report: Teaching and using the personal software process (PSP). Submission to ESEC'1997.

Runeson, P. (2001). Experiences from teaching PSP for freshmen. In Software Engineering Education and Training, 2001. Proceedings. 14th Conference on, pp. 98--107.

Runeson, P. (2003). Using students as experiment subjects--an analysis on graduate and freshmen student data. In Proceedings of the 7th International Conference on Empirical Assessment in Software Engineering, pp. 95--102, Keele University, UK.

Svahnberg, M., Aurum, A., and Wohlin, C. (2008). Using students as subjects -- an empirical evaluation. In ESEM '08: Proceedings of the Second ACM-IEEE international symposium on

Empirical software engineering and measurement, pp. 288--290, New York, NY, USA. ACM.

Tuma Solutions, L. (1998-2012). Process dashboard. version 1.15.0.1.

Notas biográficas:

Omar S. Gómez es Ingeniero en Computación por la Universidad de Guadalajara (México, 2001), Maestro en Ingeniería de Software por el Centro de Investigación en Matemáticas A.C. (México, 2005) y Doctor en Software y Sistemas por la Universidad Politécnica de Madrid (España, 2012). Actualmente es profesor en la Facultad de Matemáticas de la Universidad Autónoma de Yucatán (México). Su área de investigación se desarrolla en la experimentación en Ingeniería en Software, diseño y verificación de software así como en temas de procesos para la mejora continua en el desarrollo de software.

Antonio A. Aguilera es Licenciado en Ciencias de la Computación, egresado de la Facultad de Matemáticas (FMat) de la Universidad Autónoma de Yucatán (UADY), Maestro en Ciencias Computacionales por el Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), campus Monterrey. Actualmente es coordinador del Sistema de Información y Control Escolar Institucional (SICEI) de la UADY y Profesor de Tiempo Completo en la FMat. El área de investigación de su interés gira en torno a la calidad en la Ingeniería de Software.

Gerzon E. Gómez es Ingeniero en Sistemas Computacionales por la Universidad Autónoma de Tamaulipas. Actualmente es profesor en el área de Ingeniería de Software en la Facultad de Matemáticas de la Universidad Autónoma de Yucatán (UADY). Es doctor candidato en Informática por la Universidad Politécnica de Madrid. Colaborador en el grupo de investigación SEMEPRO (Seguridad y Mejora de Proceso) y en el proyecto Cátedra de Mejora de Procesos Software en el Espacio Iberoamericano (MPSEI). Autor de Publicaciones en Revistas Internacionales.

Raúl A. Aguilar es Licenciado en Ciencias de la Computación por la Universidad Autónoma de Yucatán, Doctor en Informática por la Universidad Politécnica de Madrid, España. Actualmente es profesor de tiempo completo en la Facultad de Matemáticas de la Universidad Autónoma de Yucatán. Su trabajo de investigación incluye las siguientes áreas: Ingeniería de Software e Informática Educativa.



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.

Desarrollo de las úlceras en el pie diabético

Aurora Espinoza Valdez

Departamento de Ciencias Computacionales, CUCEI,
Universidad de Guadalajara.
aurora.espinoza@cucei.udg.mx

Beatríz Carely Luna Olivera

Cómputo Científico y Matemáticas Aplicadas
de la Universidad de Papaloapan.
bcluna@unpa.edu.mx

Gualberto Solís Perales

Departamento de Electrónica, CUCEI,
Universidad de Guadalajara.
gualberto.solis@cucei.udg.mx

Resumen: El pie diabético constituye un problema de salud pública por su alta prevalencia y sus enormes costos. Debido a esto, es necesario investigar nuevos enfoques de prevención y tratamiento de las complicaciones vasculares del pie diabético, para entender el comportamiento del padecimiento y obtener información relevante. Por tal motivo, se requiere un acercamiento multidisciplinario que ayude a evitar complicaciones vasculares en las extremidades inferiores. En esta contribución describimos como el uso de herramientas matemáticas y en particular la teoría de grafos sirven para estudiar y describir el comportamiento del desarrollo de las úlceras en el pie diabético. Los resultados muestran que es posible describir la dinámica de las úlceras en el grafo incorporando información fisiológica de las úlceras.

Palabras clave: Pie diabético, grafo, úlcera.

Development of diabetic foot ulcers

Abstract: The diabetic foot is a public health problem due to its high prevalence and enormous costs. Because of this, it is necessary to investigate new approaches of prevention and treatment of the vascular complications of diabetic foot, in order to understand the behavior of the disease and to obtain relevant information. Therefore, a multidisciplinary approach is required to help to prevent vascular complications in the lower extremities. In this contribution we describe how the use of mathematical tools and in particular graph theory serve to study and describe the behavior of the development of diabetic foot ulcers. The results show that it is possible to describe the dynamics of the ulcers in the graph incorporating physiological information of them.

Keywords: Diabetic foot, graph, ulcer.

1. Introducción

La diabetes mellitus es una enfermedad crónica que aparece cuando el páncreas no produce insulina suficiente o cuando el organismo no utiliza eficazmente la insulina que produce (Organización Mundial de la Salud, OMS). Actualmente, la diabetes afecta a más de 371 millones de personas en el mundo (OMS). Se considera que en nuestro país actualmente existen aproximadamente 10,603,220 diabéticos (Federación Mexicana de Diabetes, FMD) con una prevalencia comparativa de diabetes 15.59 %, y con 3,452,410 personas con diabetes no diagnosticadas. Los gastos médicos de una persona con diabetes son de dos a cinco veces más altos que los de una persona sin esta enfermedad. En los pacientes con diabetes el riesgo de muerte es al menos dos veces mayor que en las personas sin diabetes.

Uno de los efectos de la diabetes no controlada es la hiperglucemia (aumento de glucosa en la sangre), que con el tiempo daña gravemente muchos órganos y sistemas, especialmente los vasos sanguíneos (Kolluru 2012). Un 50% de los

pacientes mueren de enfermedad cardiovascular (OMS). Un problema frecuente en los pacientes diabéticos es el pie diabético, esto debido a las alteraciones vasculares observadas que muestran características especiales. El pie diabético, se define como infección, ulceración y destrucción de los tejidos profundos, asociadas con anormalidades neurológicas de diversa gravedad en las extremidades inferiores (OMS). La neuropatía de los pies (lesión microvascular de vasos sanguíneos que irrigan los nervios) combinada con la reducción del flujo sanguíneo incrementan el riesgo de úlceras de los pies, y en última instancia, amputación. La neuropatía diabética se debe a lesión de los nervios a consecuencia de la diabetes, y puede llegar a afectar a un 50% de los pacientes.

El control de la glucemia parece prevenir el desarrollo de microangiopatías (afectación de la función capilar) de la retina y el riñón, y probablemente retarda la evolución de neuropatías. Muchas lesiones del pie diabético se inician con síntomas de neuropatía. La neuropatía diabética incluye síntomas sensoriales menores, reducción de la velocidad de conducción nerviosa y resistencia a la falla de conducción isquémica, y la hipoxia nerviosa parece tener un papel significativo en su origen.

De todas las amputaciones relacionadas con diabetes, 70-80% son precedidas por úlceras crónicas y hasta dos tercios experimentarán una segunda amputación (OMS). Se calcula que el tratamiento y atención básica de la diabetes permitirían prevenir hasta el 80% de las amputaciones de pies diabéticos (OMS). Esto es, las amputaciones de pie diabético se pueden disminuir mediante la modificación de factores como la neuropatía y el control glucémico, siendo este último el principal factor que causa un desorden en el proceso de angiogénesis para la generación de nuevos vasos sanguíneos (Lepäntalo et al. 2011).

2. Representación de las úlceras en un grafo

El pie es particularmente vulnerable a daños circulatorios y neurológicos, y el menor trauma puede causar úlceras o infecciones (Figura 1).

La Figura 2 indica las principales zonas del pie en las que aparecen con mayor frecuencia las úlceras. Las úlceras localizadas en la zona plantar tienen forma ovalada, son profundas, con bordes callosos y base granulada. Por lo contrario, las que aparecen en los dedos tienen bordes planos e irregulares y forma redondeada y es posible la afectación ósea. Existe un sistema de clasificación desarrollado por Wagner para la estadificación de las úlceras de pie diabético, que ha sido ampliamente aceptado, ver Figura 3 (Wagner 1979, Gordois et al. 2003):



Figura 1. Úlcera neuropática en una posición típica bajo el metatarso y rodeada de callosidad.



Figura 2. Puntos más susceptibles a la formación de úlceras en el pie diabético.

A fin de modelar un grafo para visualizar la dinámica de las úlceras en el cual se pueda incluir la información fisiológica, definimos un grafo.

Definición 1. Un grafo G es una terna ordenada $(V(G), E(G), \psi_G)$ que consiste de un conjunto no vacío $V(G)$ de vértices, de un conjunto $E(G)$ de aristas y de una función de incidencia ψ_G , que para cada arista se cumple:

- ψ_G asocia la arista e , a un par de elementos de $V(G)$, $\psi_G(e) = \{u, v\}$.

El grafo G tiene vértices etiquetados con todas sus aristas orientadas, la orientación simboliza el cambio de estado de la úlcera. Cada vértice v es etiquetado con la información fisiológica del estado de la úlcera, por la función $V(G) \rightarrow (\mathbf{R}^+ \cup \{0\})^p$, donde p es el número de parámetros fisiológicos considerados para el desarrollo de úlceras. Representación de las úlceras en un grafo, ver Figura 4.

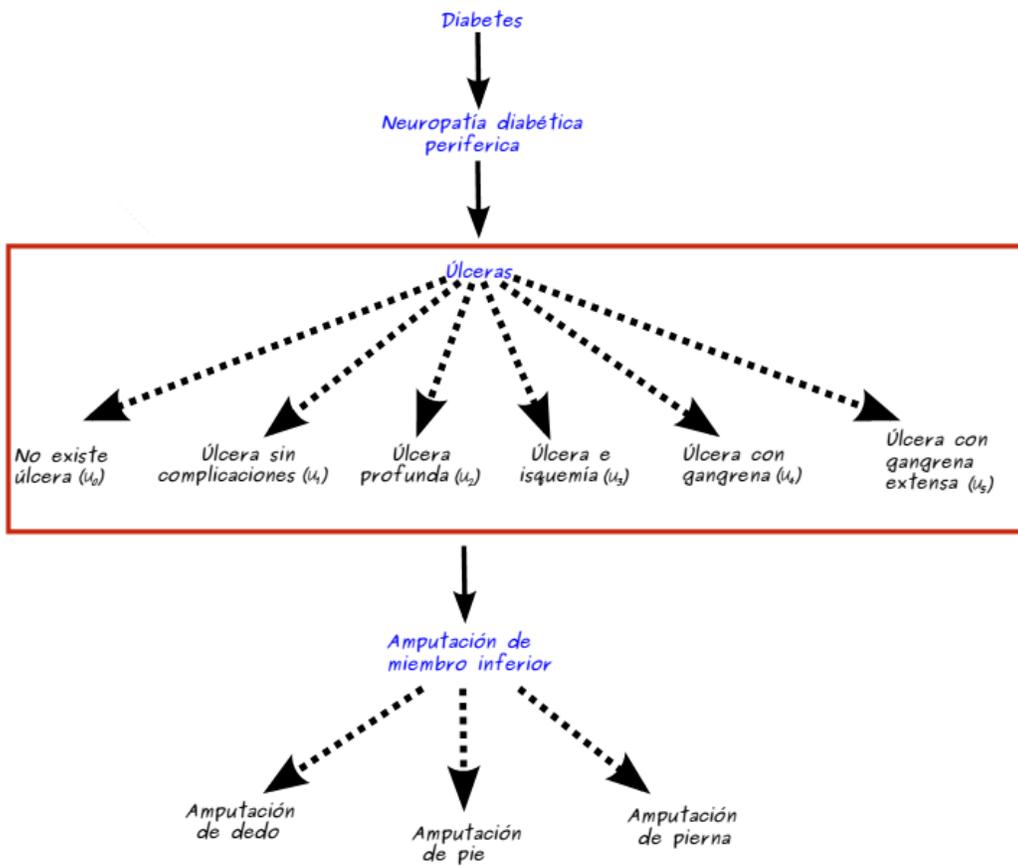


Figura 3. Clasificación de úlceras.

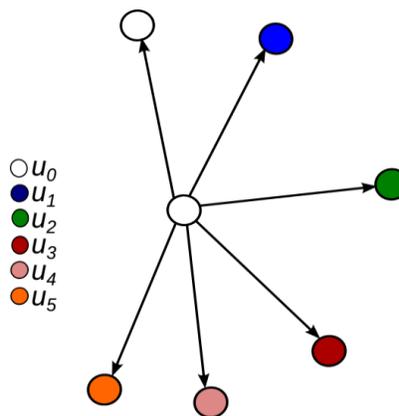


Figura 4. Grafo de esquema de las úlceras.

3. Resultados

El cambio de estado de la úlcera depende de la concentración de glucosa C_{gluc} y el consumo de oxígeno R_{oxy} en sangre (Buchwald 2011). Esto es, la concentración de oxígeno determina el estado de la úlcera (Lepántalo et al 2011). Por otra parte, el tiempo en el que se puede dar el cambio de estado depende de varios factores que se tienen que considerar, pero no se cuenta con la suficiente información experimental al respecto para considerar el factor tiempo.

En base a datos experimentales de monitoreo durante 365 días en pacientes con úlceras, se determinaron los intervalos de concentración de oxígeno (Lepántalo et al. 2011, Beckert et al. 2006):

$$u_0 \in (0.114, 0.143) \text{ M m}^{-3}, u_1 \in (0.084, 0.114) \text{ M m}^{-3}, u_2 \in (0.057, 0.084) \text{ M m}^{-3}, \\ u_3 \in (0.043, 0.057) \text{ M m}^{-3}, u_4 \in (0.03, 0.043) \text{ M m}^{-3} \text{ y } u_5 \in (0.026, 0.03) \text{ M m}^{-3}.$$

En la Figura 5 se describe la dinámica de la evolución de los estados u_i ($i = 0,1,2,3,4,5$) que se definen como sigue: Estado u_0 , se puede cambiar a cualquiera de los 5 estados u_1, u_2, u_3, u_4, u_5 ; Estado u_1 , se puede cambiar a los estados u_2, u_3, u_4, u_5 o bien regresar al estado u_0 ; Estado u_2 , se puede cambiar al estado u_3, u_4, u_5 o bien regresar al estado u_0 o u_1 ; Estado u_3 se puede cambiar al estado u_4, u_5 o bien regresar al estado u_1 o u_2 ; Estado u_4 , se puede cambiar al estado u_5 y no se regresa a ningún otro estado.

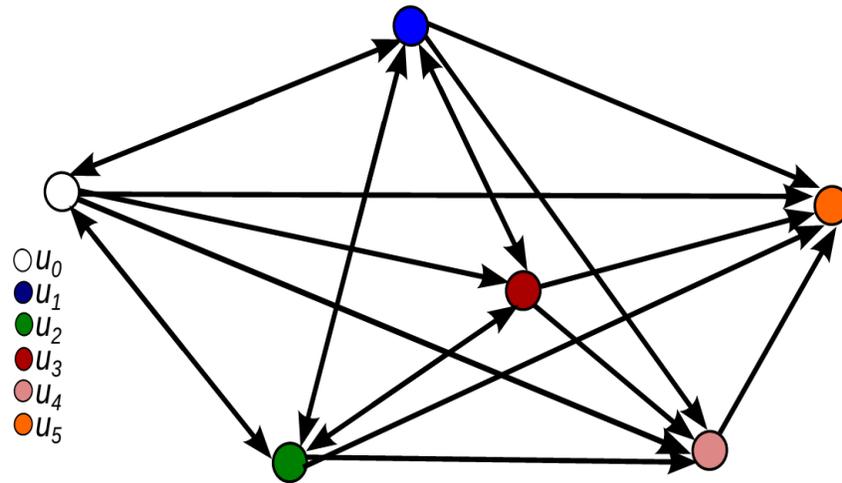


Figura 5. Dinámica de las úlceras.

El estado crítico de las úlceras es el u_3 , ya que además de la úlcera existe isquemia y por lo tanto no se puede regresar a una piel sana, *i.e.* no es reversible hasta u_0 . Existe un conjunto de posibles caminos en función de los parámetros fisiológicos, entonces de acuerdo a la dinámica que existe se determina el siguiente árbol que describe algunas de todas las posibles combinaciones o escenarios que puede tener un paciente con pie diabético (Figura 6).

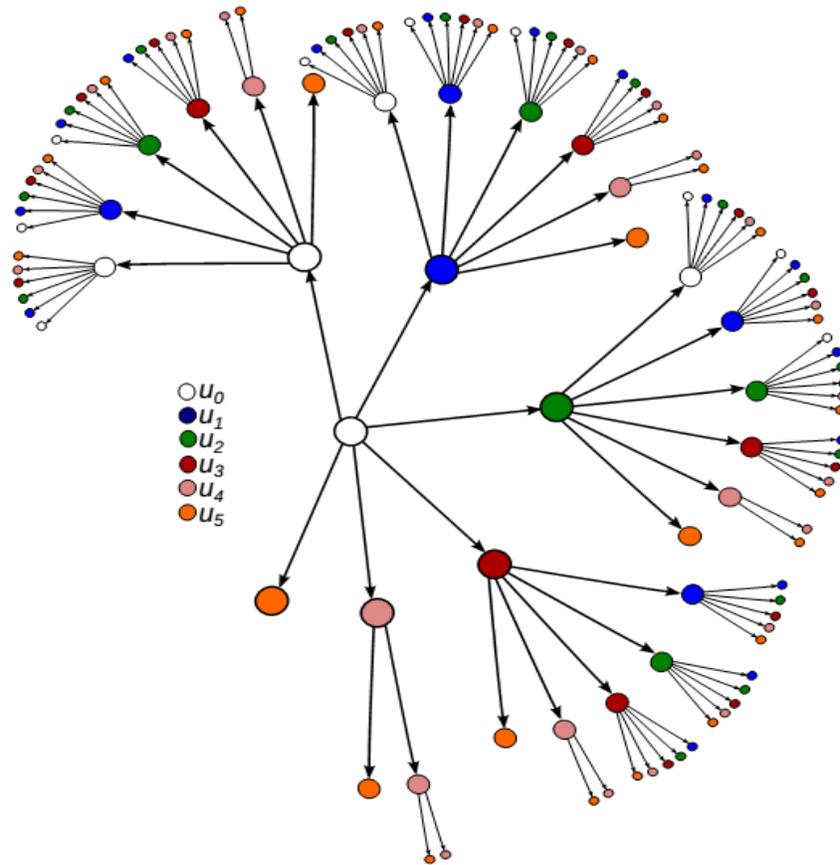


Figura 6. Posibles caminos.

4. Conclusiones

En este trabajo se muestra que el desarrollo de las úlceras en el pie diabético, se puede representar mediante un grafo G con vértices etiquetados y aristas orientadas. El vértice incluyendo información fisiológica del principal parámetro de concentración de oxígeno. En base a esto, se analizó cada uno de los estados de las úlceras para determinar su reversibilidad, obteniendo que el estado crítico es el u_3 de las úlceras. En conclusión, la teoría de grafos permite modelar la dinámica de las úlceras, *i. e.*, describe una red de como interactúan.

Referencias

Beckert S., Witte M., Wicke C., Königsrainer A. and Coerper S. A new wound-based severity score for diabetic foot ulcers. A prospective analysis of 1,000 patients. *Diabetes Care*, Vol. 29 number 5, pp 988-992. 2006.

Buchwald P. A local glucose-and oxygen concentration-based insulin secretion model for pancreatic islets. *Theoretical biology and medical modelling*. Vol. 8. 2011.

Gordois A, Scuffham P, Schearer A, Oglesby A, Ash Tobian J. The health care cost of diabetic peripheral neuropathy in the US. *Diabetes Care*, Vol. 26, No. 6, pp. 1790-1795. 2003.

Kolluru GK, Bir SC, Kevil CG. Endothelial dysfunction and diabetes: effects on angiogenesis, vascular remodeling, and wound healing. *International Journal of vascular medicine*. DOI:10.1155/2012/918267. 2012.

Lepäntalo M., Apelqvist J., Setacci C., Ricco J.B., de Donato G., Becker F., Robert-Ebadi H., Cao P., Eckstein H.H., De Rango P., Diehm N., Schmidli J., Teraa M., Moll F.L., Dick F., and Davies A.H. Chapter V: Diabetic Foot. *European Journal of Vascular and Endovascular Surgery*, Vol. 42(s2), pp s60- s74. 2011.

Tomake R.J. Editor. *Assemble of the vasculature and its regulation*. Berlin Birkhäuser. 2001.

Wagner FW Jr. A classification and treatment program for diabetic, neuropathic, and dysvascular foot problems. In *Instruccion Course Lecturs*. Sr. Luis, CV Mosby, American Academy of Orthopaedic Surgeons, pp 143-165. 1979.

Notas biográficas:

Aurora Espinoza Valdez recibió el grado de Licenciada en Matemáticas Aplicadas en la Universidad Autónoma de Baja California (UABC), México, a finales de 2000. El grado de Maestra en Ciencias Aplicadas en la Universidad Autónoma de San Luis Potosí (UASLP), México, en 2005. El grado de Doctora en Ciencias en el Instituto Potosino de Investigación Científica y Tecnológica (IPICYT), México, a finales de 2010. Es miembro del Sistema Nacional de Investigadores (SNI), México. Desde el 02/2012 es Profesor Investigador del Depto. de Ciencias Computacionales en el Centro Universitario de Ciencias Exactas e Ingenierías de la Universidad de Guadalajara. Su interés de investigación es el modelamiento matemático de sistemas biológicos y sistemas dinámicos.

Beatriz Carely Luna Olivera estudió la Licenciatura en Matemáticas en la UDLA Puebla, realizó su maestría y doctorado en Ciencias con especialidad en Matemáticas Aplicadas en la Universidad Autónoma de San Luis Potosí. Actualmente trabaja en la Universidad del Papaloapan, en donde es profesora investigadora, es miembro del SNI, nivel I y tiene perfil PROMEP. Sus principales intereses son los estudios de sistemas dinámicos discretos, topología y dinámica de redes, y el análisis de redes que aparecen en diversos fenómenos químicos, médicos, biológicos, sociales, etc., así como también la divulgación de las matemáticas en todos los niveles educativos.

Gualberto Solís Perales es Ingeniero Electrónico egresado de la Facultad de Ciencias de la Universidad Autónoma de San Luis Potosí en 1995, Maestro y Doctor en Ingeniería Eléctrica en 1999 y 2002 respectivamente, ambos grados por la Facultad de Ingeniería de la Universidad Autónoma de San Luis Potosí. Es miembro del Sistema Nacional de Investigadores desde el año 2002. Sus áreas de interés son el control no lineal de sistemas dinámicos, y particularmente en la sincronización de sistemas y redes de sistemas dinámicos y sus aplicaciones en sistemas biológicos, eléctricos y electrónicos. Actualmente es Profesor Investigador en el Centro Universitario de Ciencias Exactas e Ingenierías de la Universidad de Guadalajara.



Esta obra está bajo una licencia de Creative Commons
Reconocimiento-NoComercial-CompartirIgual 2.5 México

Interfaz humano-computadora basada en señales de electrooculografía para personas con discapacidad motriz

Daniel Pacheco Bautista

Posgrado en Ingeniería Biomédica
Universidad Popular Autónoma del Estado de Puebla (UPAEP).
daniel.pacheco01@upaep.edu.mx

Ignacio Algreto Badillo

Departamento de Ingeniería en computación
Universidad del Istmo
algreodobadillo@sandunga.unistmo.edu.mx

David De la Rosa Mejía

Posgrado en Ingeniería Biomédica
Universidad Popular Autónoma del Estado de Puebla (UPAEP).
david.delarosa01@upaep.edu.mx

Aurelio Horacio Heredia Jiménez

Posgrado en Ingeniería Biomédica
Universidad Popular Autónoma del Estado de Puebla (UPAEP).
aureliohoracio.heredia@upaep.mx

Resumen: En este trabajo se presenta el desarrollo de un prototipo que asiste, a personas con cierta discapacidad motriz, en la interacción con la computadora de una forma simple y económica, mediante señales de electrooculografía. Esta técnica permite detectar los movimientos oculares basada en el registro de la diferencia de potencial existente entre la córnea y la retina, tal propiedad es aprovechada en este proyecto para controlar el desplazamiento del cursor

del ratón de una forma precisa sobre la pantalla de la computadora. El prototipo es un diseño compacto alimentado de una fuente única de 5V proveniente del puerto USB y utiliza la circuitería ya implementada en cualquier ratón electromecánico convencional con mínimas modificaciones. El uso de tales dispositivos así como de electrodos convencionales hace el producto de un costo relativamente bajo en relación a las propuestas en otros trabajos.

Palabras clave: Electrooculografía, discapacidad motriz, Interfaz humano-computadora, Ingeniería biomédica.

Human-computer interface based on signals of electrooculography for physical disabled people

Abstract: In this paper, the development of a prototype is presented, which allows people with certain mobility disabilities interact with the computer. The design of the prototype is composed of different elements on a simple and economical way by mainly using electrooculographic signals. It is important to highlight that the proposed technique enables the detection of the eye movements based on the records of the potential difference between cornea and retina. In this project, this last property is used to control the movement of the mouse cursor on the computer screen in a precise way. The prototype is powered from a single 5V supply from USB port, uses compact circuitry and is implemented in a conventional electromechanical mouse with minimal modifications. The use of such devices as well as common electrodes make possible that the final product presents a low-cost proposal.

Keywords: Electrooculography, mobility disabilities, human-computer interface, biomedical engineering.

1. Introducción

La discapacidad motriz es un padecimiento que en la actualidad se ha incrementado considerablemente, solo en México, según fuentes del Instituto Nacional de Estadística y Geografía (INEGI) alrededor del 5% de la población presenta discapacidad de algún tipo, del cual el 58% corresponde al tipo motriz. El origen y naturaleza de tales padecimientos son múltiples sin embargo en la mayoría de los casos la actividad mental y la funcionalidad de los ojos del individuo permanecen intactas, siendo este último un excelente recurso que puede permitir a la persona discapacitada seguir interactuando con su entorno realizando actividades de manera prácticamente normal.

Existen en la actualidad, diversos trabajos dedicados al desarrollo de prototipos que utilizan las señales eléctricas generadas en los ojos, obtenidas mediante electrooculografía (EOG), para controlar diferentes objetos tales como sillas de ruedas (Yathunanthan, Chandrasena, Umakanthan, Vasuki & Munasinghe, 2008), (Arai & Mardiyanto, 2011), brazos robotizados (Hao et al., 2013), interfaces gráficas de asistencia a discapacitados (Ding, Tong & Li, 2005), cursor y teclado de la computadora (Rosas-Cholula, Ramírez Cortés, Escamilla Ambrosio & Alarcón Aquino, 2011), (Caballero Gaviria, Castro Miller, Ordoñez Medina & Rojas González, 2011), (Merino, Rivera, Gómez, Molina & Dorronzoro, 2010), entre otros. En particular el problema del control del cursor mediante señales EOG ha sido tratado de diversas maneras, en (Rosas-Cholula, et al., 2011) los autores presentan un prototipo usando señales EOG obtenidas indirectamente mediante un set comercial inalámbrico de 16 electrodos, originalmente usado para adquisición de señales de electroencefalografía (EEG), la posición del cursor se obtiene a partir de la información de un giroscopio incluido en el set, mientras que los clics son generados a través del parpadeo voluntario del usuario mediante las señales EOG, para tal propósito ellos proponen la técnica de descomposición en modo empírico (EMD) como una herramienta computacional rápida y simple, para distinguir entre el parpadeo natural y los voluntarios para el control. Otro diseño importante se encuentra en

(Caballero Gaviria, et al., 2011), en este los autores proponen una interface humano computadora basada en EOG consistente de cuatro etapas: adquisición de señales, filtros, amplificadores y un sistema de reconocimiento de patrones y generación de señales de control, el prototipo utiliza circuitos de señal mixta (PSoC), lo cual reduce el número de componentes del mismo, sin embargo solo está enfocado en los movimientos verticales y horizontales del cursor, sin atender a la funcionalidad del clic. En (Merino et al., 2010) se propone un algoritmo para el procesamiento de señales de EOG, el cual utiliza como entradas la derivada y el nivel de amplitud de la señal, la derivada es utilizada para detectar las orillas de las señales, y el nivel de amplitud para filtrar ruido, aun cuando el algoritmo muestra altas expectativas, el artículo no concluye con una implementación final.

En este trabajo se presenta el diseño de un dispositivo basado en señales de EOG para el control de las funcionalidades del mouse, siguiendo como objetivo además de la efectividad y robustez, el bajo costo del mismo. La implementación utiliza dispositivos de fácil adquisición y el algoritmo está basado en una simple máquina de estado implementada en un microcontrolador de gama media de Microchip. El resto de este artículo está organizado de la siguiente manera: La sección II describe el principio básico de la electrooculografía, la amplitud y ancho de banda de la señal, y las bases del funcionamiento de un ratón electromecánico. La sección III muestra el diseño del dispositivo, en la sección IV se presentan las pruebas y resultados obtenidos y finalmente en la sección V se mencionan las conclusiones y el trabajo a futuro.

2. Fundamento teórico

La electrooculografía

El Electrooculograma (EOG) es un método de registro de los movimientos oculares basado en la detección de la diferencia de potencial existente entre la córnea y la retina. El origen de esta diferencia se encuentra en el epitelio pigmentario de la retina y permite considerar la presencia de un dipolo, donde la córnea corresponde al extremo positivo y la retina al extremo negativo, ver figura 1. Si el ojo se mueve desde la posición central directo a la periferia la retina se aproxima a uno de los electrodos, mientras que la córnea al electrodo del lado opuesto. Este cambio en la orientación del dipolo se refleja en un cambio en la amplitud y la polaridad de la señal EOG. Por consiguiente, resulta inmediato que el análisis de estos cambios permite determinar el movimiento de los ojos. La señal EOG puede obtenerse fácilmente mediante un par de electrodos conectados en la vecindad de los ojos más un electrodo de referencia como se muestra en la figura 2. De esta manera se tienen dos canales correspondiendo al movimiento horizontal y al movimiento vertical, respectivamente. De acuerdo a estudios previos, las señales EOG muestran amplitudes de señales que van de 5 μV a 20 μV por grado con un ancho de banda de 0 Hz a 30 Hz (Singh & Singh, 2012), (Brown, Marmor, Vaegan, Brigell & Bach, 2006). Una forma de onda idealizada EOG puede observarse en la figura 3.

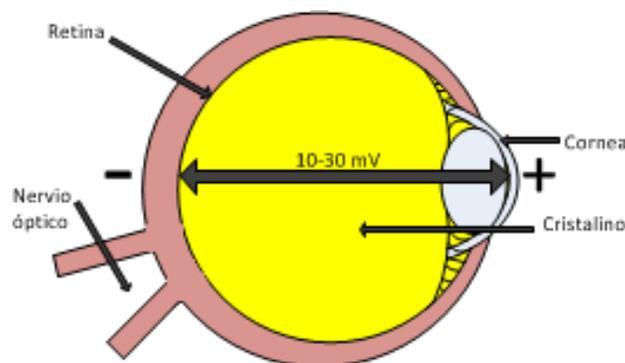


Figura 1. El ojo modelado como un dipolo.

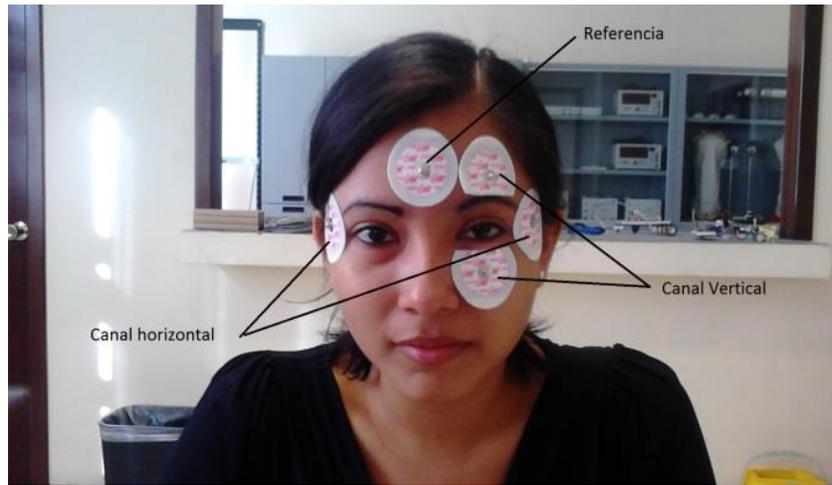


Figura 2. Ubicación de los electrodos.

Es posible detectar diversos movimientos del ojo a través del EOG, no obstante, en este trabajo se utilizan solo los tres principales: sacádicos, fijaciones y parpadeos (Bulling, Ward, Gellersen & Troster, 2010), donde los sacádicos corresponden a los movimientos del ojo al cambiar rápidamente la mirada de un punto a otro (generalmente mayor a los 30 grados), las fijaciones resultan de mantener la mirada en un mismo punto y los parpadeos son los movimientos involuntarios y periódicos del ojo para humedecer la córnea y la conjuntiva.

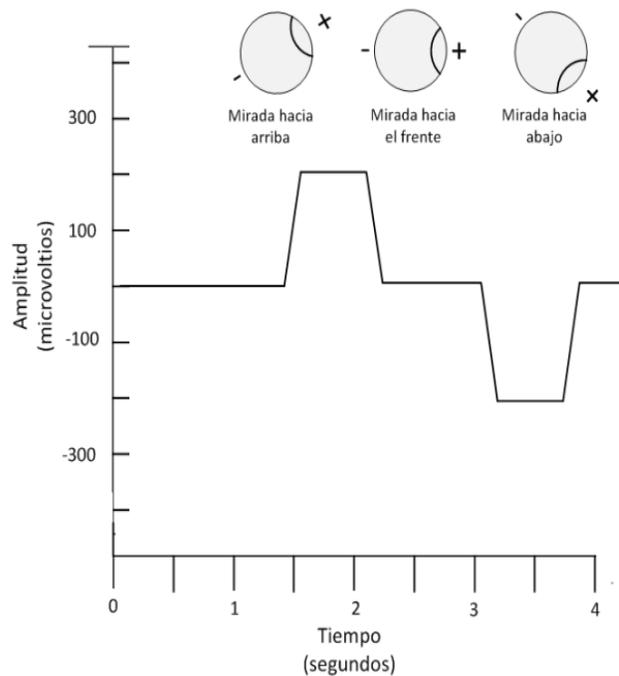


Figura 3. Forma de onda EOG ideales cuando la persona mira hacia arriba, al frente y hacia abajo.

El ratón electromecánico USB

El ratón electromecánico funciona a partir de una bola de caucho acoplada a tres rodillos giratorios: dos de arrastre y uno de apoyo como se muestra en la figura 4. El cilindro de apoyo o estabilizador se encarga de mantener la bola presionada contra los de arrastre, mientras que éstos están colocados perpendicularmente entre sí y giran cuando la bola se mueve, descomponiendo el desplazamiento en las componentes cartesianas x e y (horizontal y vertical).

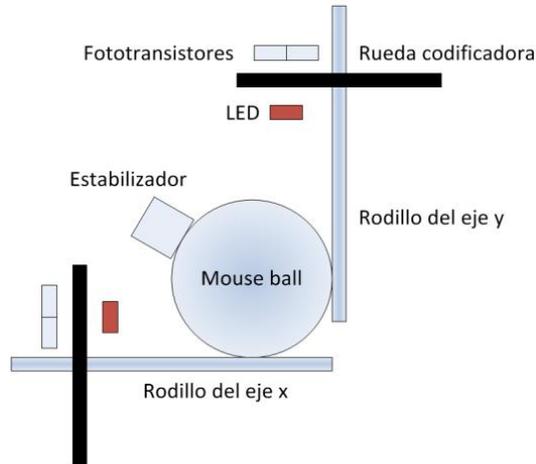


Figura 4. Sección mecánica del ratón.

Al extremo de cada cilindro de arrastre se encuentra un codificador, el cual consiste de una rueda con orificios, un emisor de luz en uno de sus lados y un detector de luz en el opuesto, ver figura 5. Cuando la rueda gira obligado por la bola, se producen interrupciones del haz de luz de manera intermitente. Por ejemplo, movimientos a la izquierda y a la derecha pueden verse como en la figura 6a y 6b, respectivamente. La circuitería del ratón transforma esa información luminosa en movimientos, sumando el número de interrupciones de luz para evaluar cuánto ha de desplazarse y en qué dirección. Esta información descompuesta en x e y se lleva al ordenador para indicarle al cursor cómo desplazarse.

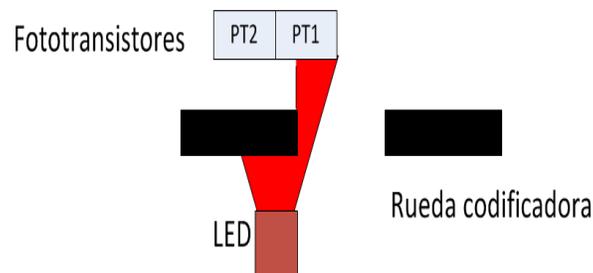


Figura 5. Sección optomecánica del ratón.

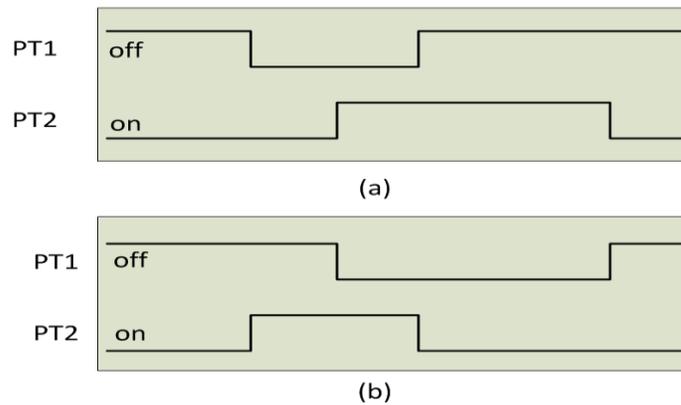


Figura 6. Salida del codificador en un a) movimiento a la izquierda y un b) movimiento a la derecha.

3. Diseño del prototipo

En la figura 7 se muestra el diagrama a bloques del dispositivo propuesto, el cual está formado de los siguientes bloques principales: 1) amplificación y filtrado, 2) conversión analógica digital (CAD) y procesamiento digital, 3) comunicación con la computadora, y fuente de alimentación. En esta sección se describen brevemente cada uno de estos bloques.

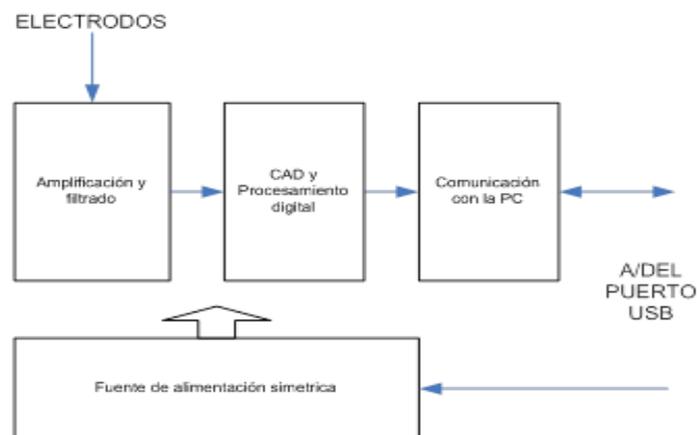


Figura 7. Diagrama a bloques del prototipo.

Amplificación y filtrado

Las señales de EOG presentan amplitudes muy pequeñas del orden de 10 μV a 300 μV con un ancho de banda de 0 a 30 Hz (Singh & Singh, 2012) (Brown, et al., 2006), por consiguiente un amplificador de EOG además de presentar alta ganancia debe tener una excelente respuesta en bajas frecuencias. En particular en este diseño, se eligió una ganancia total de 4725 (proporcionada por dos amplificadores en cascada), cuya motivación principal es obtener en la salida señales con amplitudes que van de 47mV a 1.5V, las cuales resultan apropiadas para su conversión al formato digital en la siguiente etapa (CAD y procesamiento digital). Por otra parte, el ancho de banda de los filtros fue diseñado de 0.15 Hz a 30 Hz, resaltando que la discriminación de algunas frecuencias del espectro no resultan críticas debido a la naturaleza de la aplicación y fue necesario para disminuir el efecto de línea base a la deriva, como se explica posteriormente. Adicionalmente, existe una gran cantidad de ruido e interferencia que corrompe la señal deseada y la hace difícil de distinguir, tales como el ruido de la línea de distribución de energía, la interferencia electromagnética, el ruido ocasionado por otros dispositivos electrónicos, los artefactos de movimiento que cambian la interfaz piel-electrodo, la contracción de los músculos o electromiografía, la respiración, entre otros, muchas de estas señales suelen manifestarse en modo común por lo que una elevada razón de rechazo en modo común (CMRR) es otra característica deseable del diseño (Choudhury, Venkataramanan, Nemade & Sahambi, 2005), (Acharya , 2011).

El bloque de amplificación y filtrado es idéntica para ambos canales horizontal y vertical, por lo que en lo que sigue se hace referencia a sólo uno de estos. El diseño y desarrollo de este bloque fue dividido en tres secciones, donde la primera se muestra en la figura 8. El núcleo de esta primera sección es el amplificador de instrumentación INA114, el cual reporta una elevada CMRR de 115 dB, el cual es un valor apropiado para este tipo de aplicaciones (Choudhury, et al., 2005) y fue configurado para proporcionar una ganancia G de 225, a partir de la ecuación 1, donde R_g es la resistencia equivalente de R_1 , R_4 y R_{V2} .

$$G = 1 + \frac{50k}{R_g} \quad [1]$$

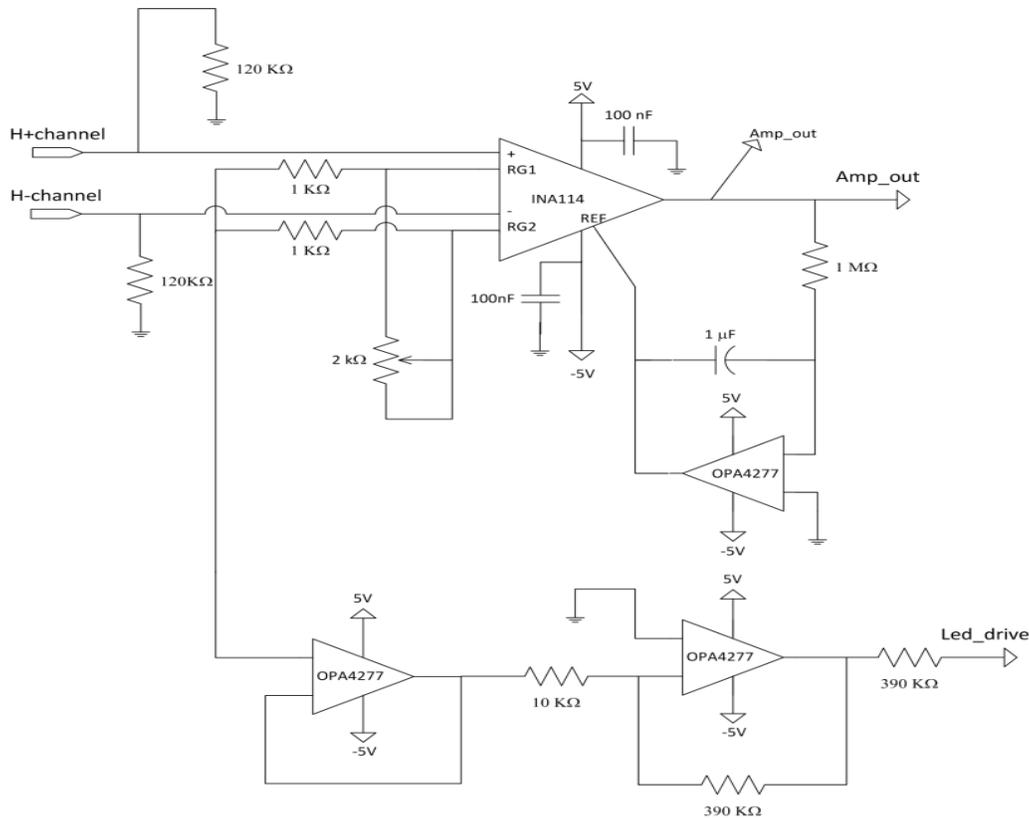


Figura 8. Primera sección de la etapa de amplificación y filtrado: Amplificación, Cancelación de offset e incremento de la CMRR.

Algunas señales de interferencia y la polarización de los electrodos suelen combinarse creando un efecto conocido como línea base a la deriva, es decir la línea de referencia de la señal de salida se mueve lentamente hacia una de las fuentes de polarización sin razón aparente, lo cual provoca la saturación del amplificador en un periodo corto de tiempo. Para evitar este efecto, un amplificador operacional (OPA4277 mostrado en la parte media de la figura 8) integra la componente en DC de la salida, la invierte y la realimenta al punto de referencia. Tal restaurador de DC cambia el amplificador original acoplado en DC a un amplificador con acoplo en AC, donde la frecuencia de corte del mismo es de 0.15Hz, calculado a partir de la ecuación 2.

$$f_{-3dB} = \frac{1}{2\pi R_3 C_1} \quad [2]$$

Adicionalmente para mejorar el rechazo al ruido en modo común, dos amplificadores operacionales (OPA4277 mostrados en la parte inferior de la figura 8) invierten y realimentan el voltaje en modo común de la entrada al cuerpo del paciente, además, en caso de una condición donde el paciente accidentalmente sea conectado a la tierra del sistema, éste provee la trayectoria para la corriente desde el nodo Vref (Acharya , 2011).

La segunda sección, mostrada en la figura 9, es un amplificador inversor convencional, el cual proporciona una ganancia adicional de 21, al mismo tiempo que desplaza la línea base a 2 volts. Lo último es necesario para procesar la señal mediante el convertidor analógico digital del microcontrolador, lo cual se detalla posteriormente.

Finalmente la tercera sección es un filtro pasabajas activo, el cual está sintonizado a una frecuencia de corte de 30Hz, utilizando la configuración Sallen Key, ver figura 10. En este circuito la frecuencia de corte es calculada a partir de la ecuación 3.

$$f_{-3dB} = \frac{1}{2\pi R_{20} \sqrt{(C_8 + C_9) C_{10}}} \quad [3]$$

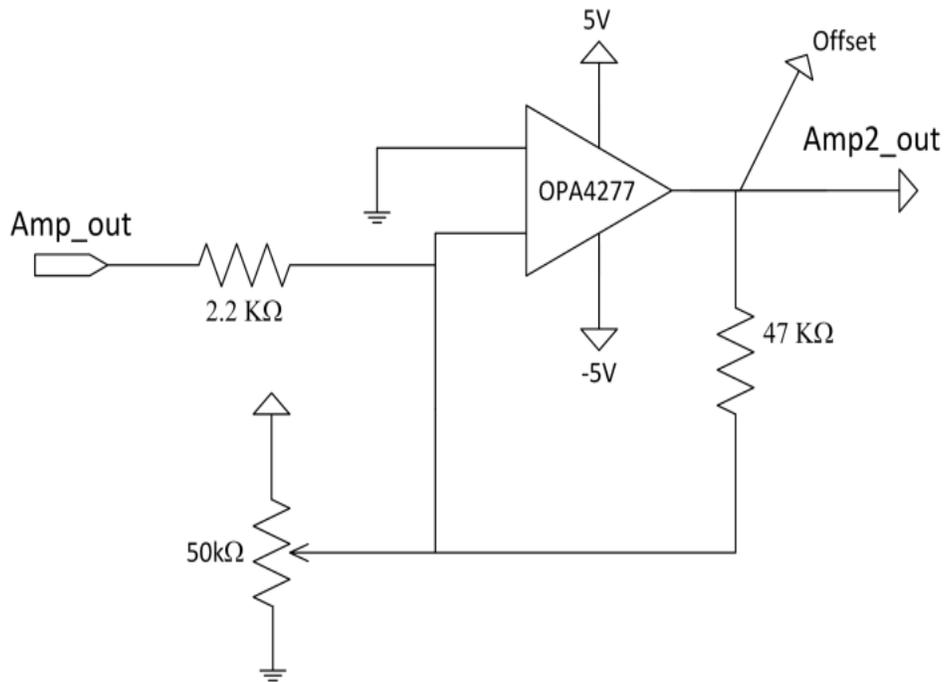


Figura 9. Segunda sección de la etapa de amplificación y filtrado: Amplificación y acondicionamiento de la señal.

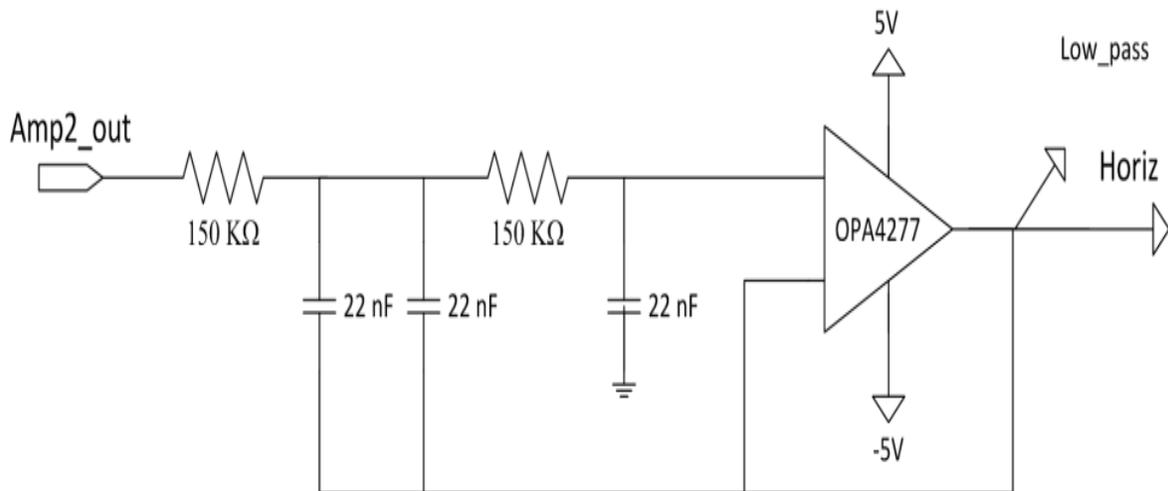


Figura 10. Tercera sección de la etapa de amplificación y filtrado: Filtro pasabajas.

CAD y Procesamiento digital

Este bloque comprende la conversión analógica-digital y el procesamiento digital, ambas funciones fueron implementadas en el microcontrolador PIC16F618. El uso de tal dispositivo y, en particular, de su convertidor analógico-digital, es de manera directa, por lo que en esta sección únicamente se explicará la lógica usada para generar las señales a enviar a la etapa de comunicación a partir de las señales previamente amplificadas y filtradas. Inicialmente, se desarrolló un simple protocolo para el uso del dispositivo, el cual consiste de los siguientes cuatro pasos:

1. El usuario se ubicará a una distancia de 30 a 40 cm de la pantalla, observando al punto central de la misma.
2. Un movimiento sacádico a la izquierda iniciará el recorrido del cursor hacia la izquierda. En este momento el usuario puede regresar la mirada a cualquier otro punto de la pantalla.
3. Un movimiento sacádico a la derecha detendrá automáticamente el cursor.
4. En ese instante, si el usuario así lo desea, un guiño voluntario del ojo seleccionará el objeto señalado.

Situaciones duales ocurren cuando se desea mover el cursor en dirección contraria, hacia arriba o hacia abajo de la pantalla. Es importante resaltar que los movimientos lentos del ojo y con ángulos menores a 30 grados (es decir, movimientos no sacádicos) no afectan el protocolo, lo que evita que el usuario genere movimientos indeseables del cursor al observar algún icono o menú en la pantalla. De igual forma los guiños involuntarios para humectar el ojo resultan despreciables debido a su pequeña magnitud y duración.

Se seleccionó este protocolo buscando que el usuario realice la mínima cantidad de movimientos en los ojos, evitando de esta manera su cansancio. Al mismo tiempo, un protocolo sencillo proporciona robustez y simplicidad al sistema. Es importante notar que para el uso de este esquema es suficiente con detectar la existencia de un movimiento sacádico o la ausencia de éste, sin la preocupación en la duración o la magnitud exacta del mismo. A partir de las formas de onda

de la figura 11, el protocolo fue implementado mediante la máquina de estados mostrada en la figura 12, donde un pulso positivo corresponde a un movimiento sacádico a la izquierda (o hacia arriba), mientras que un pulso negativo es generado por un movimiento sacádico a la derecha (o hacia abajo). Observe como en esta grafica los pulsos ya no son perfectamente rectangulares como en la Figura 3, lo que modela el acoplo en AC resultante de agregar el integrador en la salida del amplificador diferencial. Las etiquetas N1, P1, N2 y P2 son niveles de umbral, necesarios para la operación de la máquina de estados de la Figura 12.

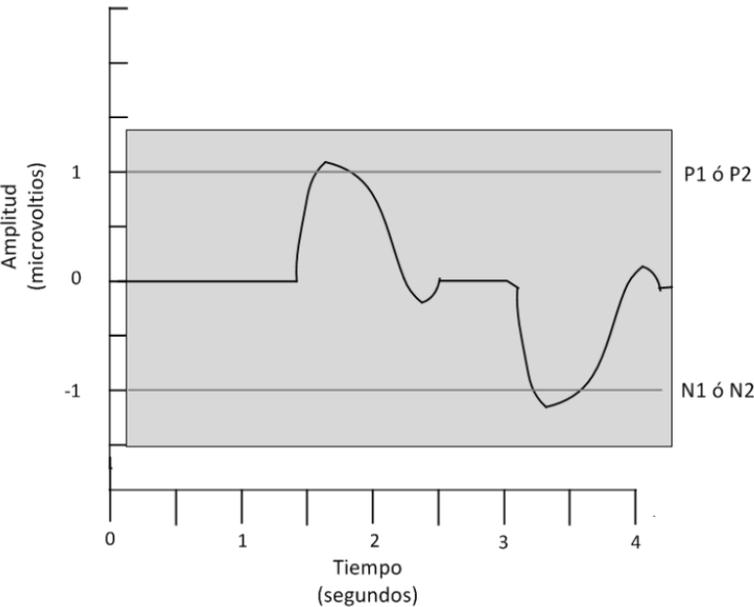


Figura 11. Definición de los niveles de umbral.

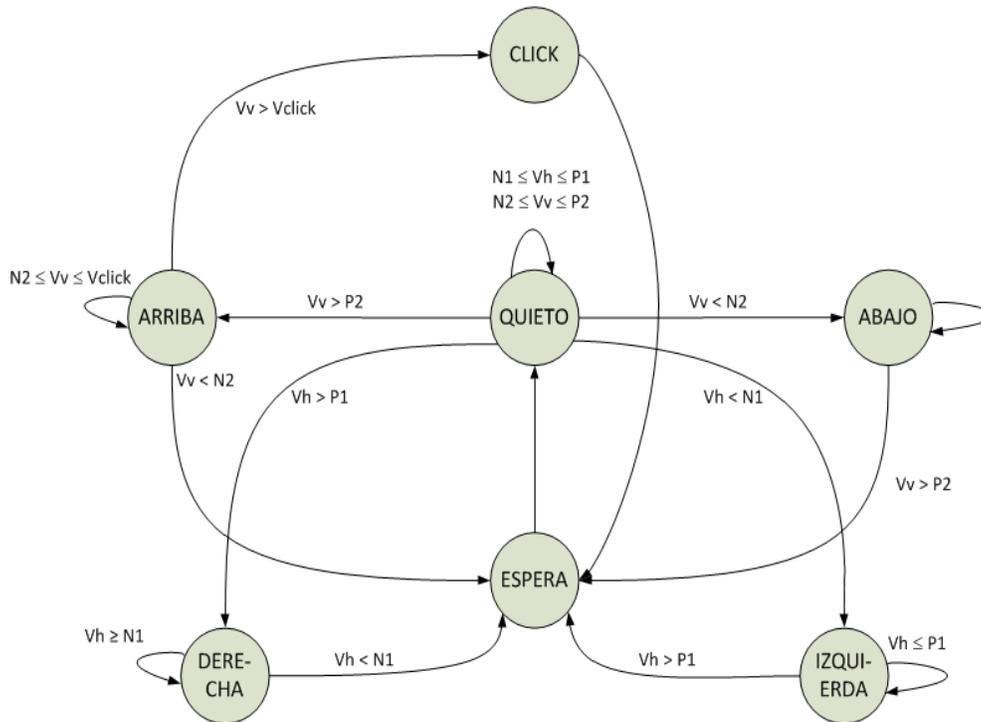


Figura 12. Máquina de estados para implementar el protocolo.

En la figura 11, las variables V_v y V_h representan la salida del convertidor analógico-digital de los canales vertical y horizontal, respectivamente. Los estados derecha e izquierda, generan fundamentalmente las secuencias mostrada en la figura 6a y 6b para el canal horizontal, mientras que los estados arriba y abajo generan esas mismas secuencias para el canal vertical. El estado click genera un pulso cuadrado en el Pin RB7 del microcontrolador, mientras que el estado delay únicamente es un ciclo de retardo de 300ms.

Partiendo del estado quieto, si el usuario no genera ningún movimiento sacádico, V_v y V_h no superarán en magnitud los umbrales P_2 y N_2 y P_1 y N_1 respectivamente, permaneciendo en ese mismo estado. Si por ejemplo, estando en esta condición el usuario mira rápidamente hacia la derecha, V_h superara a P_1 y la máquina adquiere el estado derecha, generando las señales que iniciaran el recorrido del cursor hacia esa misma dirección. En estos momentos un movimiento de la mirada hacia la izquierda provocará que V_h sea menor a N_1 ,

pasando al estado espera, el cual básicamente es un retardo utilizado para ignorar la parte complementario del segundo pulso. La secuencia es similar para los otros movimientos del cursor. El clic fue modelado por un guiño fuerte del ojo el cual es un pulso positivo muy similar al de un movimiento sacádico hacia arriba, pero de mayor magnitud y menor duración, al ocurrir tal movimiento, la máquina de estados adquiere el estado click, en el cual se genera básicamente un pulso de 300ms en el Pin RB7.

Comunicación con la PC

Los pines del microcontrolador correspondientes a las secuencias generadas por la etapa previa para el movimiento horizontal y vertical, así como el pulso que emula el clic, son conectados al circuito de un ratón electromecánico, permitiendo que este realice la comunicación con el PC. Para tal efecto los fototransistores y el interruptor del clic en el ratón original fueron retirados del circuito, conectando las señales de movimiento en los puntos correspondientes a los emisores de dichos fototransistores y el pulso que emula el clic en uno de los puntos donde fue retirado el interruptor.

Fuente de alimentación simétrica

La alimentación del circuito es a través del puerto USB de la misma computadora, no obstante la mayoría de los integrados operan con fuentes simétricas, por consiguiente fue necesario incluir una fuente de voltajes simétricos de $\pm 5V$ a partir del integrado TC7660CPA, el cual basa su operación en la conmutación de capacitores. El diagrama de conexión es mostrado en la figura 13.

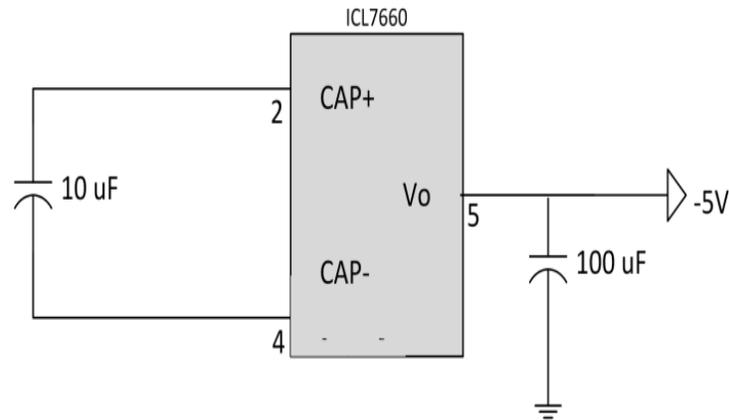


Figura 13. Fuente de alimentación simétrica.

4. Resultados obtenidos

El circuito fue simulado en el software de desarrollo Proteus v7.6 y posteriormente implementado completamente, teniendo un consumo de potencia de aproximadamente 450mW, sin considerar la etapa de comunicación con la computadora. La respuesta en frecuencia de la etapa de amplificación y filtrado simulada se muestra en la figura 14 y se muestra que el ancho de banda y la amplitud de la señal son acorde con lo calculado (0.15Hz-30Hz y 4725, respectivamente). En la figura 15 puede apreciarse la forma de onda obtenida al realizar el usuario un movimiento sacádico a la derecha seguido de uno a la izquierda, en tal imagen se puede observar una señal estable, de suficiente amplitud (800 mV aproximadamente) y con el problema de línea base eliminado. Tales mediciones fueron realizadas con el osciloscopio digital Tektronix TDS2022.

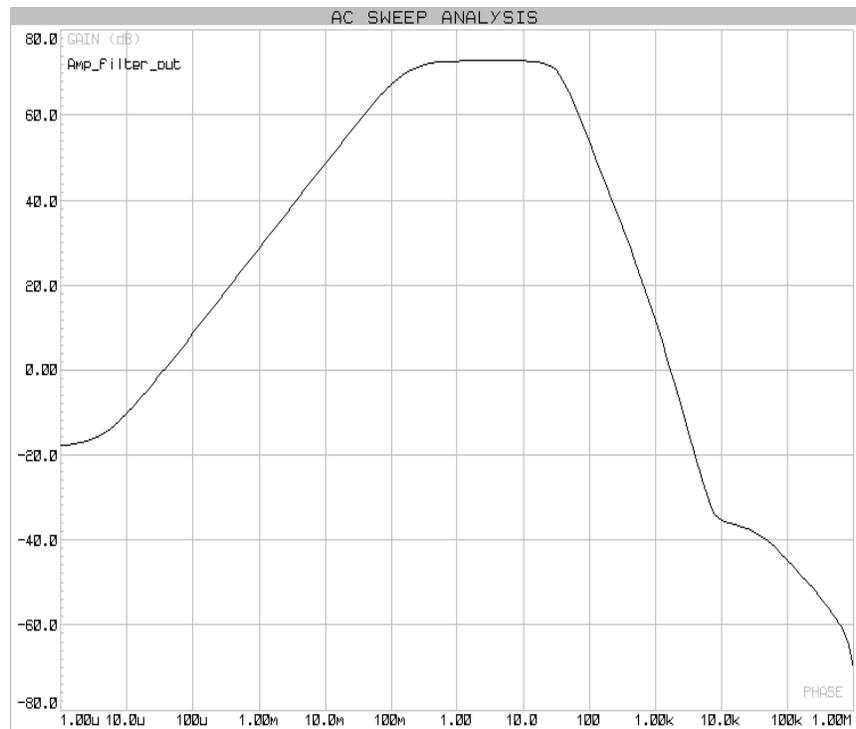


Figura 14. Respuesta en frecuencia de la etapa de amplificación y filtrado.

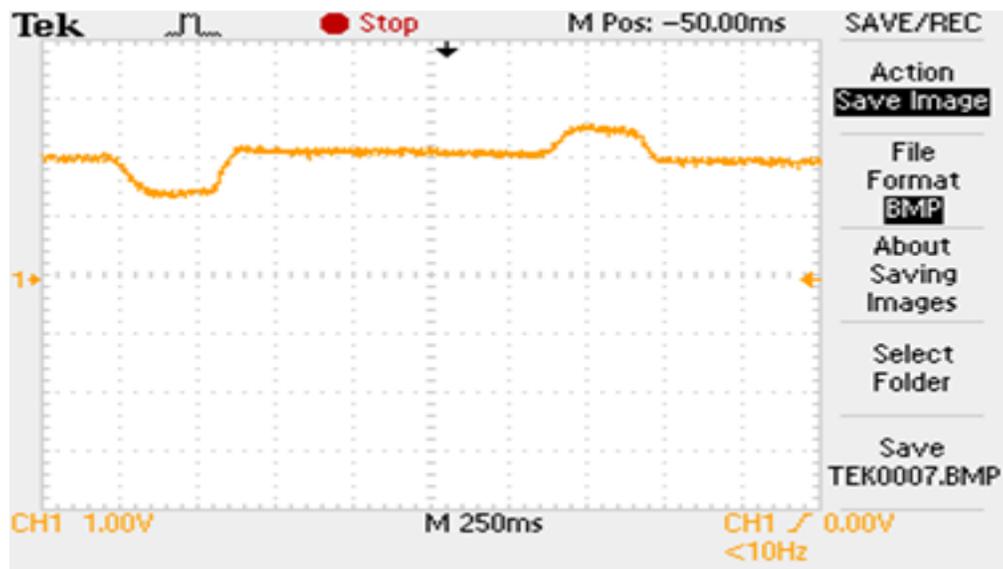


Figura 15. Forma de onda obtenida al realizar un movimiento sacádico a la derecha y luego uno a la izquierda.

Finalmente, la funcionalidad del dispositivo fue evaluado en 20 personas. Las pruebas se dividieron en dos grupos como muestra la tabla 1, donde el primer grupo consiste en inicializar un movimiento del cursor (en cada una de las 4 direcciones) para posteriormente detener su recorrido, comenzando en posiciones aleatorias de la pantalla. En el segundo grupo y partiendo del centro de la pantalla, el usuario tenía como objetivo ubicar el cursor sobre un icono localizado en la parte superior derecha de la misma y, posteriormente, seleccionarlo mediante un clic, en donde la trayectoria que el cursor debería seguir fue libre. Como puede observarse en la tabla 1, el dispositivo opera apropiadamente en el 100% de los casos de movimiento y detención sin embargo aún muestra problemas en la generación del clic, lo cual se atribuye a diferencias en la amplitud y duración del pulso entre los usuarios.

CLASIFICACION	PRUEBAS	PERSONAS CON EXITO
Grupo 1	Iniciar y detener movimiento a la izquierda	20
	Iniciar y detener movimiento a la derecha	20
	Iniciar y detener movimiento hacia arriba	20
	Iniciar y detener movimiento hacia abajo	20
Grupo 2	Partir del centro de la pantalla y posicionarse en el icono	20
	Seleccionar posteriormente el icono	12

Tabla 1. Resultados de las pruebas realizadas en 20 usuarios.



Figura 16. Prototipo siendo probado por un usuario.

5. Conclusiones

Se ha realizado un dispositivo de interfaz humano computadora para personas con discapacidad motriz, a través de señales de electrooculografía, cuyo diseño es compacto, eficiente, robusto y fácil de utilizar. Siendo además de bajo consumo de potencia y económico en relación a otras propuestas en el mercado. En lo sucesivo se pretende optimizar la velocidad de la interfaz y eliminar la limitante encontrada al realizar el clic sobre la pantalla. También se pretende incluir toda la lógica de comunicación, actualmente en la circuitería del ratón electromecánico, en el mismo microcontrolador. El proyecto muestra cómo la electrooculografía es una alternativa muy eficiente y económica para el control de dispositivos. Este mismo proyecto podría utilizarse en lo sucesivo en el control de sillas de ruedas o robots de asistencia, sólo por mencionar algunos ejemplos.

Reconocimientos

Este proyecto fue apoyado por el Programa para el mejoramiento del profesorado (PROMEP) y la Universidad del Istmo (UNISTMO), a través de la beca para estudios de doctorado con folio UNISTMO-003.

Referencias

- Acharya V. (2011). Improving Common-Mode Rejection Using the Right-Leg Drive Amplifier. (Reporte de aplicación SBAA188), Texas Instruments.
- Arai, K. & Mardiyanto, R. (2011). Eyes based electric wheel chair control system. *International journal of advanced computer science and applications*, Vol 2 (12), pp. 98-105.
- Brown, M., Marmor, M., Vaegan, Zrenner, E., Brigell, M. & Bach, M. (2006). ISCEV Standard for Clinical Electro-oculography EOG. *Documenta Ophthalmologica*, Vol. 113 (3), pp. 205-212.
- Bulling, A., Ward, J. A., Gellersen, H. & Troster, G. (2010). Eye movement analysis for activity recognition using electrooculography. *IEEE transactions on pattern analysis and machine intelligence*. Vol. 33(4). pp. 741-743.
- Caballero Gaviria, A.D., Castro Miller, I.D., Ordoñez Medina, S & Rojas González, D.M (2011) Implementación de una interfaz hombre-computador basada en registros EOG mediante circuitos de señal mixta PSoC. *Proceeding of the V Latin American Congress on biomedical engineering, CLAIB 2011, La Habana, Cuba*, pp. 16-21.
- Choudhury, S. R., Venkataramanan, S., Nemade, H. B. & Sahambi, J. S. (2005). Design and Development of a Novel EOG Biopotential Amplifier. *International Journal of Business and Emerging Markets*, Vol. 7(5), pp. 271-274.
- Ding, Q., Tong, K. & Li, G. (2005). Development of an EOG (Electro-Oculography) based human-computer interface, *Proceedings of the 2005 IEEE Engineering in medicine and biology 27th annual conference, Shanghai, China*, pp. 6829-6831.
- Hao, Y., Controzzi, M., Cipriani, C., Popovic, D. B., Yang, X., Chen, W., Zheng, X. & Chiara Carrozza M. (2013). Controlling Hand-Assistive Devices: Utilizing Electrooculography as a Substitute for Vision. *Robotics & Automation Magazine, IEEE*. Vol. 20(1), pp. 40-52.
- Merino, M., Rivera, O., Gómez, I., Molina, A. & Dorrnzoro, E. (2010). A Method of EOG Signal Processing to Detect the Direction of Eye Movements. *First International Conference on Sensor Device Technologies and Applications, Venice/Mestre, Italy*, pp. 100-105.
- Rosas-Cholula, G., Ramírez Cortés, J.M., Escamilla Ambrosio, J. & Alarcón Aquino V. (2011). On the development of a simple EOG-based mouse with BCI technology applying empirical mode decomposition and DWT. *Proceeding of the 15th International graphonomics society Conference, IGS2011, Cancún, México*. pp. 70-73.
- Singh, H. & Singh, J. (2012). A review on electrooculography. *International Journal of Advanced Engineering Technology*, Vol. 3(4), pp.112-122.
- Yathunanthan, S., Chandrasena, L.U.R., Umakanthan, A., Vasuki, V. & Munasinghe, S.R. (2008). Controlling a Wheelchair by Use of EOG Signal. *Proceeding of 4th International Conference on Information and Automation for Sustainability, ICIAFS 2008. Colombo, Sri Lanka*.

Notas biográficas:



Daniel Pacheco Bautista. Recibió el grado de ingeniero en electrónica del Instituto Tecnológico de Oaxaca (ITO), en el 2000, y el grado de Maestro en ciencias con especialidad en diseño de circuitos integrados, del Instituto nacional de Astrofísica, Óptica y Electrónica (INAOE), en el 2003. Actualmente es estudiante del programa de doctorado en Ingeniería Biomédica, en la Universidad Popular Autónoma del Estado de Puebla (UPAEP), y profesor investigador en la Universidad del Istmo (UNISTMO). Sus líneas de investigación incluyen: Arquitectura de computadoras y Lógica reconfigurable, Osciladores controlados por voltaje y Circuito de amarre de fase en VLSI, y recientemente plataformas alternativas para el ensamble de secuencias cortas de ADN.



Ignacio Algreto Badillo. recibió el grado en Ingeniería Electrónica del Instituto Tecnológico de Puebla (ITP) en el 2002 y los grados de Maestría y Doctorado en Ciencias Computacionales del Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) en el 2004 y 2008, respectivamente. Desde 2009, se desempeña como profesor-investigador de tiempo completo adscrito a la carrera de Ingeniería en Computación en la Universidad del Istmo. Él es miembro del SNI desde el 2011 y sus líneas de investigación son enfocadas en el diseño y desarrollo de sistemas digitales, arquitecturas reconfigurables, plataformas basadas en el concepto radio software, seguridad, prototipos didácticos, sistemas criptográficos, implementaciones en FPGA, sistemas basados en microcontroladores y microprocesadores y aceleración en hardware para aplicaciones específicas.



David De La Rosa Mejía. Nació en Tapachula Chiapas, el 18 de Septiembre de 1988, se graduó como Ingeniero biónico y actualmente estudiante de la Maestría en Ingeniería Biomédica en la Universidad Popular Autónoma del Estado de Puebla. Se ha desempeñado en el área Biomédica en mantenimiento preventivo y correctivo de instrumental Médico, actualmente trabaja en diseño y construcción de equipo Médico.



Aurelio Horacio Heredia Jiménez. Recibió el grado de ingeniero eléctrico de la Universidad Juárez Autónoma de Tabasco (UJAT), y los grados de Maestría y Doctorado con especial en óptica, del Instituto nacional de Astrofísica, Óptica y Electrónica (INAOE). Actualmente es profesor investigador en la Universidad Popular Autónoma del Estado de Puebla, adscrito al departamento de posgrado en Ingeniería Biomédica, siendo sus principales líneas de investigación: Diseño y construcción de equipo biomédico, diseño de microsistemas (MEMS/NEMS), Microcomponentes ópticos, y diseño de sistemas utilizando fibras ópticas y guías de onda.



Esta obra está bajo una licencia de Creative Commons
Reconocimiento-NoComercial-CompartirIgual 2.5 México