

# **Implementación de un experimento psicológico para examinar atención auditiva en PEBL**

Juan Sebastián Gómez Rosas<sup>a</sup>, María Elizabeth Mónica Carlier Torres<sup>b</sup>

<sup>a</sup> Universidad Autónoma de Bucaramanga

<sup>b</sup> Universidad Nacional Autónoma de México

## Resumen

El lenguaje para la construcción de experimentos psicológicos PEBL, (por su sigla en inglés, The Psychology Experiment Building Language) es una herramienta de utilidad para transformar experimentos psicológicos en contenidos digitales de fácil distribución y acceso bajo un marco homogéneo. La plataforma PEBL fue concebida con base en las mecánicas de aplicación de experimentos psicológicos tradicionales, por lo que incluye funciones y herramientas para el desarrollo de nuevas tareas y facilita su aplicación en pacientes y la obtención de información para uso de los profesionales. Los experimentos psicológicos para examinar atención auditiva en plataformas de software libre son escasos pese al importante papel que juega en la adquisición y el procesamiento del lenguaje. En este artículo se presenta la implementación de un experimento psicológico para examinar atención auditiva utilizando la plataforma PEBL.

## Abstract

The Psychology Experiment Building Language, PEBL, is a useful tool for transforming psychology experiments into digital contents with ease of distribution and low cost under a homogeneous framework. This platform was conceived based on the mechanics for applying traditional psychology experiments, therefore it includes functions and toolkits for developing new tasks and facilitates their application on patients as well as obtaining information for professional use. Psychology experiments focused on auditory attention examination based on open source software are scarce despite their important role in acquisition and processing of language. This paper explores the exercise of multidisciplinary production of a psychology experiment using the PEBL platform.

## *Introducción*

El desarrollo de la atención se caracteriza por la habilidad de sincronizar el estado de alerta con la información recibida por los sentidos, además de seleccionar la información que entra al sistema nervioso central facilitando su procesamiento cognitivo posterior, lo que finalmente conlleva a la generación de conocimiento en el niño (Zuluaga, 2005). El estudio de la atención se ha centrado clásicamente en la modalidad visual, sentando las bases para la comprensión del proceso en otras modalidades sensoriales incluida la auditiva, por lo que los estudios de la atención auditiva son escasos (Gomes, et al., 2000). En niños en edad escolar la atención auditiva se puede examinar mediante el desempeño de tareas de discriminación y secuenciación de sonidos. Para discriminar y mantener la secuencia de sonidos el niño tiene que estar alerta, mantener en memoria el sonido anterior y comparar el sonido actual con el almacenado en memoria de trabajo, antes de dar una respuesta correcta.

Los programas de diagnóstico de atención auditiva requieren tecnología para presentar los sonidos y no se encuentran disponibles fácilmente y de manera gratuita; por lo tanto, existen muy pocas propuestas computarizadas de acceso libre que permitan diagnosticar limitaciones en

procesos cognitivos relacionados con la atención auditiva en niños de 5 a 9 años. La detección de posibles problemas de atención entre más temprano se haga mejor, porque se pueden tratar y evitar que dichos problemas se conviertan en un déficit de atención e hiperactividad. El propósito de este proyecto fue implementar en formato digital un experimento psicológico para examinar atención auditiva en niños de 5 a 9 años en PEBL, una plataforma de software libre.

### Características de la plataforma PEBL

PEBL es una aplicación de código abierto, multiplataforma, que provee su propio lenguaje de programación, interfaz de programación (API), motor de gráficos y ejecutables para los principales sistemas operativos. El código fuente está disponible para descarga bajo los términos de la licencia pública general o GNU-GPL versión 2 (Kogut & Metiu, (2001)). PEBL es un lenguaje de programación que facilita implementar experimentos psicológicos en formato digital. Un lenguaje de programación se entiende como un conjunto de símbolos estructurados que permiten escribir instrucciones consecutivas y ordenadas para la ejecución de tareas específicas. Los lenguajes de programación permiten dar a las máquinas instrucciones en forma de código para la ejecución de procesos con objetivos específicos llamados programas o software. Generalmente para facilitar el desarrollo dichos lenguajes ofrecen interfaces de programación (API) y bibliotecas de software.

Las bibliotecas de software son piezas de código que constituye un conglomerado de funciones para una plataforma, lenguaje de programación o tecnología específica. Si se piensa en el programa como una receta de cocina, las bibliotecas que incluye serían ingredientes previamente cocidos y añadidos para facilitar y complementar el producto final (Sebesta, et al., 2002). Una API o interfaz de programación de aplicaciones, consiste en un conjunto de funciones y utilidades que presenta una biblioteca de software para interactuar con sus funcionalidades al momento de crear un software mediante un lenguaje de programación. PEBL permite trabajar experimentos psicológicos desde un modelo que parte de la configuración o preparación del experimento en sí, para seguir con la aplicación de los estímulos, la recolección de los datos originados por la respuesta del participante y por último el registro de estos datos. Ver fig. 1.

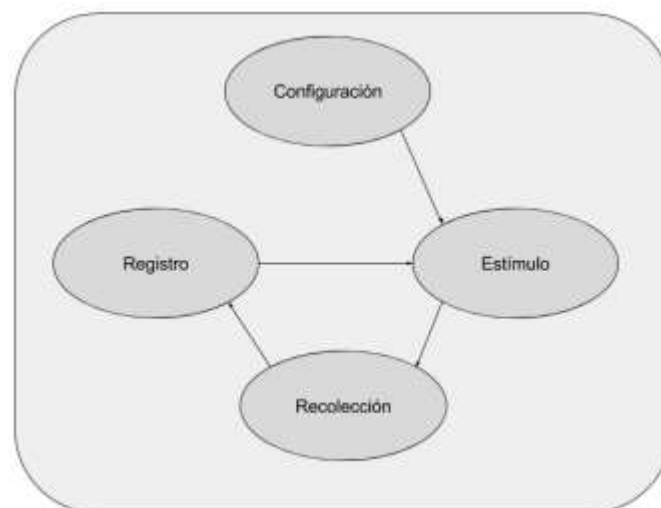


Figura 1. Flujo de un experimento en PEBL.

La plataforma PEBL permite el uso de parámetros ajustables para cada ejercicio según se haya definido. Estos parámetros son definidos en un documento de texto que acompaña al ejercicio y adquieren un valor predefinido en caso de que no sean ajustados manualmente por el usuario. Para ajustar los parámetros se dispone de una interfaz gráfica que presenta una tabla con los parámetros y sus respectivos valores., ver fig. 2.



Figura 2. Interfaz de parámetros en PEBL.

El manual del aplicativo PEBL se encuentra disponible en la página del proyecto, fue escrito por el autor del software, Dr. Shane T. Mueller y representa una referencia completa a las características y al lenguaje de programación así como a las interfaces que componen el aplicativo. Dicho manual detalla cada uno de los aspectos funcionales del lenguaje y la plataforma con respecto a los diferentes casos de uso que componen un ejercicio, relaciona las funcionalidades que provee la plataforma y proporciona ejemplos prácticos para los desarrolladores (Mueller, 2010)

La interfaz de programación de aplicativos API de PEBL provee funciones para la construcción de experimentos psicológicos. Estas funciones se han elaborado de forma que faciliten la abstracción de las funcionalidades presentes en los experimentos psicológicos (Mueller & Piper, 2014). Cuenta con funciones matemáticas que permiten realizar cálculos y aproximaciones con magnitudes escalares, programas de lectura y escritura de archivos, utilidades de control de flujo y tiempo que permiten la orientación del programa al momento de presentar estímulos y recoger información. Adicionalmente dicha API cuenta con auxiliares de dibujo que facilitan las tareas de ubicación, desplazamiento, visibilidad de elementos gráficos como imágenes y texto en pantalla. Un ejemplo de las utilidades anteriormente mencionadas sería la función EasyLabel( ) que Crea una etiqueta de texto visual con sólo proporcionar la ubicación deseada.

## Experimento Psicológico

El experimento psicológico que se implementó mediante software fue el propuesto por Tallal y Piercy (1973), cuyo fin es examinar percepción auditiva en niños con problemas específicos del lenguaje. Este experimento usa tonos altos y bajos y propone estudiar secuenciación y discriminación de tonos presentados como pares. El paradigma propone cinco sesiones. La primera sesión es de entrenamiento en identificar tonos altos y bajos. La segunda es de demostración de cómo responder. Estas dos sesiones se hacen con la orientación del experimentador o psicólogo. La tercera es de entrenamiento con retroalimentación, esto quiere decir que se le indica al niño si

la respuesta fue correcta o incorrecta. La cuarta es entrenamiento sin retroalimentación, no se le indica al niño nada después de la respuesta y la quinta es la prueba.

La prueba consiste en discriminar y secuenciar tonos que ocurren rápidamente o con intervalos entre tonos muy breve como por ejemplo 15 ms. En los ejercicios de entrenamiento se usa un intervalo entre tonos de 428 ms. En la prueba se usan diferentes intervalos, desde muy breves (15 ms) a muy largos (4096 ms) escogidos aleatoriamente. Para mayor detalle sobre el experimento ver Tallal y Piercy, (1973).

Los estímulos auditivos son tonos complejos con frecuencias dentro del rango del habla donde el tono1 tiene frecuencia fundamental de 54 Hz y el Tono frecuencia fundamental de 180 Hz y con rise/fall tiene de 40 ms-1.

### *Método*

El método utilizado consiste en aprovechar conocimientos y experticia en desarrollo de software multimedial en conjunción con directrices en la construcción de experimentos psicológicos y las ventajas del software libre a nivel de desarrollo y distribución para obtener un marco de trabajo desde diferentes disciplinas con el fin de producir una herramienta de software que constituya un experimento para el diagnóstico de retrasos en procesos cognitivos relacionados con retrasos en la atención en niños de 5 a 9 años. (Mockus, et al., 2002.)

### *Procedimiento*

Para el desarrollo del software en que se basa el experimento psicológico se utilizó la plataforma para el diseño y aplicación de experimentos PEBL. (pebl.org). Se partió de la exploración del sistema de software libre PEBL en cuanto a su ecosistema, comunidad y plataforma de desarrollo. (Mueller. et al., 2014). Se evaluó las capacidades de PEBL para permitir la diversificación y enriquecimiento de actividades enfocadas al diagnóstico de retrasos en procesos cognitivos relacionados con retrasos en la atención mediante el diseño e implementación de juegos de atención y memoria para niños entre 5 y 9 años al igual que los lineamientos para elaborar software multimedia que permita la simulación de experimentos psicológicos.

El desarrollo se llevó a cabo utilizando una metodología de desarrollo de software basada en prototipos (Beck et al., 2001), prestando constante atención a la usabilidad desde el área de psicología y enriqueciendo el aplicativo con las observaciones que se generaron.

Luego del desarrollo se procedió a la publicación y liberación del experimento documentado y acompañado de su código fuente en la plataforma de hospedaje de software de código abierto Github. (github.com). Para lograr la comprobación del correcto funcionamiento del aplicativo y utilidad se realizaron pruebas de caja blanca, las cuales permiten validar el comportamiento del software por parte del desarrollador con conocimiento del código y se planteó la futura realización de pruebas de caja negra como parte de ejercicios de diagnóstico del déficit de atención en niños de 5 a 9 años de edad para validar que el programa cumple los requisitos funcionales descritos, esto de acuerdo al ciclo de vida para pruebas de software acompañado de verificaciones realizadas desde la psicología con la documentación respectiva (Klingberg, et al., 2005).

## Instrumentos

El diseño de pruebas de software a partir del ciclo de vida requiere de verificación de acuerdo a los comportamientos esperados del aplicativo. Los resultados producidos por el experimento en forma de archivos de salida en texto son un insumo para estas pruebas y para la verificación desde otras disciplinas. Estos resultados determinarán no sólo el correcto funcionamiento sino el producto final del aplicativo luego de la intervención.

Los datos fueron recopilados durante las pruebas del software.

En el capítulo siguiente se presentan los resultados de acuerdo al proceso de desarrollo descrito, comenzando por el análisis de requisitos y documentación de diseño, los manuales de usuario y mantenimiento, así como el aplicativo en sí y las pruebas de caja blanca realizadas al mismo.

## Resultados

### Requisitos Funcionales

El experimento funciona bajo un paradigma de repeticiones de secuencias de pares de tonos y un segundo paradigma de discriminación de pares de tonos iguales o diferentes. En ambos métodos el procedimiento a seguir es el mismo y consiste de 5 módulos. El primero de entrenamiento en tonos altos y bajos, el segundo, entrenamiento sobre cómo funciona el programa, el tercero entrenamiento con retroalimentación y cuarto entrenamiento sin retroalimentación y el quinto es la prueba. El intervalo entre tonos en los módulos del 1 al 4 es de 428 ms. En la prueba. el intervalo entre tonos es variables y se presenta de manera aleatoria.

### *Método de repetición PA-REP.*

- Se entrena el niño para que reconozca cual es el Tono1 y oprima flecha a la izquierda y cuál es el Tono 2 y oprima flecha derecha.
- Se presentan cuatro reactivos para que el experimentador le muestre al niño como operar el programa. Esto en forma de tutorial con cuatro ejemplos.
- Luego se presentan 8 reactivos de entrenamiento con retroalimentación. La retroalimentación debe decir correcto o incorrecto y además mostrar la secuencia de sonidos y de flechas que se debían ser oprimidas como respuesta correcta.
- Luego vienen 24 reactivos sin retroalimentación. Y esto lo debe hacer el sujeto hasta que responda 20 correctos de 24. Entonces el paradigma debe dar la posibilidad al evaluador de que si no cumple el criterio lo vuelva a hacer.
- Por último, viene la prueba. La prueba consiste en 48 trial de secuenciación de dos tonos con diferentes intervalos de separación que son de acuerdo al artículo 8, 15, 30, 60, 150, 305, 947, 1466, 1985, 3023, 3545, 4062 milisegundos presentados en orden aleatorio.

### *Método de igual/diferente PA-ID.*

El paradigma consiste en:

Se sigue el mismo procedimiento anterior solo que la respuesta aquí es para determinar si el par de tonos son iguales o diferentes. Se responde oprimiendo flecha hacia arriba o flecha hacia abajo.

En resumen, son 4, 8, 24 y 48 reactivos para cada método. En ambos casos se genera un archivo del desempeño del niño. Dicho archivo contiene información sobre si los niños fueron capaces de alcanzar el criterio de entrenamiento 20 de 24 y el número respuestas correctas e incorrectas en todos los casos, y además el tiempo de reacción en la respuesta.

Al final, el resultado de la prueba presenta la información de la siguiente forma:

Participante	Repetición	Correcto	Tiempo
54	1	Si	215
54	2	No	112

Adicionalmente a estos dos ejercicios se crearon unos ejercicios de motivación que consisten en memoramas de gráficas y de tonos.

### *Tarea de memoria PEBL-M*

Se trata de un memorama clásico en el que se forman parejas iguales. El participante cuenta con intentos y aciertos.

El participante debe elegir un elemento utilizando el mouse. Este queda descubierto.

Luego debe elegir otro elemento.

Si coinciden se suma un acierto y un intento, y se dejan ambos elementos descubiertos.

Si no coinciden se dejan descubiertos un momento y sólo se suma un intento. Luego se cubren de nuevo.

Y además, el resultado de la prueba presenta la información de la siguiente forma

Participante	Intento	Acierto
54	1	Si
54	2	No

### *Requisitos no funcionales*

En el desarrollo del aplicativo PEBL\_PA se tuvieron en cuenta los siguientes requisitos no funcionales:

- El aplicativo debe operar bajo la plataforma PEBL dada la validez de esta en cuanto al diseño y ejecución de experimentos psicológicos.

- Debe funcionar en cualquier versión de PEBL actual para cualquier sistema operativo soportado por PEBL.
- El aplicativo debe permitir la integración de gráficos y sonidos.
- El aplicativo debe estar en capacidad de recibir interacciones por teclado y con el mouse.
- El aplicativo debe medir tiempos con precisión.
- El aplicativo debe recopilar información del usuario en cada iteración.
- El aplicativo debe escribir archivos de texto.
- El aplicativo debe ser de uso intuitivo para el experimentador y para el usuario final.
- El aplicativo debe informar al usuario el momento de la terminación de la prueba.
- El software debe contar con manual de usuario y documentación de mantenimiento.

### Diagrama de casos de uso

Este diagrama ilustra las funcionalidades que dan lugar a los resultados producidos en el aplicativo. Se planteó el experimentador como un actor y el sujeto como otro debido a que la interacción de ambos es necesaria para el correcto funcionamiento del aplicativo:

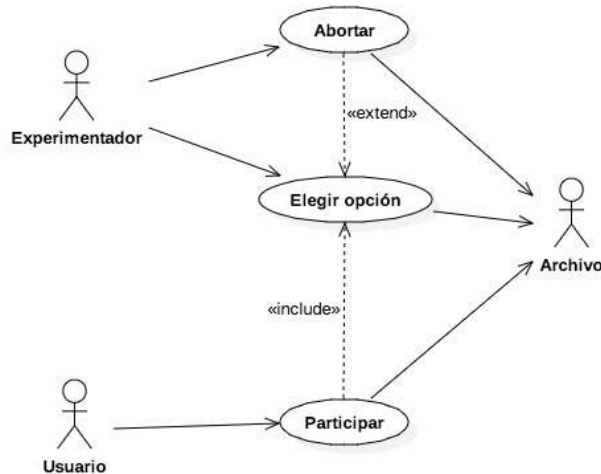


Figura 1. Diagrama de casos de uso.

En este diagrama se evidencian tres actores. Por una parte el experimentador quien decide el punto de arranque de acuerdo a las opciones presentadas en el menú y también está en capacidad de abortar la actividad. En segundo lugar tenemos el usuario que representa al participante de la prueba este únicamente puede participar de la misma mediante el uso de la mecánica de juego establecida en el paradigma. Y por último tenemos el archivo de texto que almacena la información de acuerdo a los resultados obtenidos en la prueba este actor interviene como base de datos por lo que la dirección de interacción en los casos de uso se da hacia y no desde él.

Cabe Resaltar que no se hace diferencia funcional entre el experimentador y el usuario por lo que el mismo usuario está en capacidad de realizar todas las acciones mencionadas. La razón de esto radica en que este aplicativo está pensado para ser utilizado en acompañamiento del experimentador.

## *Pruebas*

Siguiendo el modelo de ciclo de vida del software (Jacobson, et al.), se probó el aplicativo de la siguiente forma:

### Pruebas de caja blanca

El método de caja blanca se considera como la línea base de pruebas y permite conocer si el aplicativo está funcionando como se espera de acuerdo al diseño.

### Plan de pruebas

La estrategia a seguir para las pruebas se basó en el seguimiento a los requisitos funcionales de acuerdo al paradigma descrito por Tallal y Piercy.

Se probó el aplicativo teniendo en cuenta el punto de vista tanto del autor como de la directora de esta tesis, la Doctora Mónica Carlier, quien complementó las pruebas desde su experiencia como experimentadora e intervencionista.

Lo anterior se realizó de forma iterativa con el fin de refinar cada vez el aplicativo mediante prototipos.

Finalmente se tomaron las medidas correctivas pertinentes hasta lograr un conjunto de prototipos que simulan y cumplen con los paradigmas.

### Pruebas de caja negra

Debido al alcance del presente proyecto, en el cual se plantea el desarrollo de una prueba a aplicar posteriormente con niños, las pruebas de caja negra se sugieren con el fin de validar el uso del software por parte del usuario final, en una etapa posterior a su terminación y liberación.

Lo anterior se hace posible gracias a la publicación del software por medio de una licencia libre. De este modo luego de las pruebas unitarias de caja blanca, el software es publicado permitiendo su obtención para posteriormente ser probado en aceptación y otros aspectos por los usuarios finales ya sea por medio de Feedback, o por medio de contribuciones al código (Mockus, et al).

### *Aplicativo liberado*

El aplicativo resultado de la presente investigación fue denominado con la sigla PEBL-PA por Percepción Auditiva, y en los cinco módulos que lo componen la denominación fue:

Tarea de Atención Auditiva, método de iguales o diferentes: PA-ID.

Tarea de Atención Auditiva, método de repeticiones: PA-REP.

Tarea de Atención Auditiva, discriminación de tonos: PA-DT.

Tarea de memoria, memorama Auditivo: MA.

Tarea de memoria, memorama Gráfico: MG.

El aplicativo terminado y documentado fue publicado para descarga con su código fuente en la plataforma Github bajo los términos de la licencia GNU-GPL versión 3 (Stallman). Esta licencia se encuentra dentro de las licencias tradicionales de software libre y es ampliamente aceptada y



reconocida, por esta razón se utilizó para licenciar el aplicativo desarrollado, la licencia GNU-GPL versión 3 establece que no debe haber restricción alguna para el uso del software de acuerdo a las cuatro libertades fundamentales:

- La libertad de utilizar el software para cualquier propósito.
- La libertad de cambiarlo y ajustarlo a las necesidades.
- La libertad de compartir el software original.
- La libertad de compartir el software modificado.

Para garantizar lo anterior, la licencia se complementa con cláusulas que establecen un contrato de la siguiente forma:

Permitido	No permitido	Obligaciones
Uso comercial. Modificación. Distribución. Garantía. Reclamos de patentes.	Licenciar una derivación con otra licencia. Ofrecer devoluciones o indemnizaciones por garantía.	Incluir el ejecutable original. Mantener registro de cambios. Incluir el código fuente. Distribuir la licencia acompañando el software. Dar crédito a todos los autores anteriores. Incluir manual de instalación y uso.

Así el poseedor de la licencia restringe la capacidad de quienes elaboren obras derivadas para restringir a otros usuarios y contribuyentes. Lo anterior supone la forma más estricta de licencia de tipo copyleft y a su vez la más aceptada (Kumar).

El aplicativo PEBL-PA, acompañado de su código fuente y documentación, se puede encontrar en este enlace que apunta a un repositorio público de Github:

[https://github.com/muddokon/PEBL\\_PA-M](https://github.com/muddokon/PEBL_PA-M)

### *Contribuciones*

El ecosistema de PEBL está orientado a proveer una herramienta útil y sostenible para el desarrollo y aplicación de experimentos psicológicos, por tanto, motiva a sus participantes no sólo al uso del software, sino además a realizar contribuciones al mismo. Los autores plantean en esencia cuatro vías de contribución (Mueller, 2014):

- Mejoramiento de la documentación mediante la edición de las Wiki del proyecto.
- Aporte de scripts de análisis elaborados en herramientas para este fin como SPSS, Excel o R.
- Traducción de la interfaz de PEBL y de los experimentos incluidos.
- Donación de dinero al proyecto.

Publicación de un estudio utilizando y citando PEBL.

Las contribuciones permiten al usuario de PEBL contar con una contraseña para el acceso al conjunto completo de experimentos incluidos, sin esta contraseña sólo se cuenta con el conjunto básico. Sin embargo, todas las funcionalidades de PEBL se encuentran en operación en cualquier caso.

Para el presente trabajo se realizó la publicación y liberación de la tarea PEBL:PA (Percepción Auditiva) acompañada de la investigación en desarrollo multimedia para experimentos psicológicos en PEBL a manera de tesis. Con esto se realizó una contribución

### *Conclusiones y Recomendaciones*

PEBL constituye una alternativa de software libre para la presentación y el desarrollo de experimentos psicológicos. Las funciones e interfaces incluidas en la plataforma son adecuadas para el manejo de los diferentes paradigmas en torno a los experimentos psicológicos.

Es posible el seguimiento de una guía única para el desarrollo y la aplicación de experimentos psicológicos utilizando la plataforma PEBL.

Se recomienda la consulta del manual y documentación oficial de PEBL al momento de su utilización en el desarrollo experimentos psicológicos nuevos.

Se recomienda a su vez el trabajo conjunto entre profesionales en psicología o áreas afines con profesionales en desarrollo de software tanto para el diseño y Concepción como para el desarrollo y pruebas de un aplicativo que simula un experimento psicológico.

### *Reconocimiento*

El autor desea expresar sus agradecimientos al cuerpo docente del programa de Maestría en Software Libre por su apoyo y acompañamiento en el proceso de formación que culminó en la presente producción de software y en especial a la Doctora María Elizabeth Mónica Carlier Torres, Coordinadora del Área de Lenguaje de la Unidad de Investigación en Neurodesarrollo de la Universidad Autónoma de México, quien con su experiencia en diseño y aplicación de experimentos psicológicos orientó la adopción de las tecnologías utilizadas en la implementación del ejercicio denominado PA-Test (Prueba de percepción auditiva) y los ejercicios complementarios.

### *Referencias*

Sebesta, R. W., & Mukherjee, S. (2002). Concepts of programming languages (Vol. 281). Reading: Addison-Wesley.

Kogut, B., & Metiu, A. (2001). Open-source software development and distributed innovation. Oxford Review of Economic Policy, 17(2), 248-264.

Mueller, S. T. (2010). The PEBL manual. Lulu. com.

Mueller, S. T., & Piper, B. J. (2014). The psychology experiment building language (PEBL) and PEBL test battery. *Journal of neuroscience methods*, 222, 250-259.

Tallal, P., & Piercy, M. (1973). Defects of non-verbal auditory perception in children with developmental aphasia. *Nature*, 241(5390), 468-469.

Beizer, B. (1995). *Black-box testing: techniques for functional testing of software and systems*. John Wiley & Sons, Inc.

Stallman, R. (2007, June 29). GNU-GPL V 3.0. Retrieved June 12, 2017, from <https://www.gnu.org/licenses/gpl-3.0.txt>

Lakhani, K. R., & Von Hippel, E. (2003). How open source software works: “free” user-to-user assistance. *Research policy*, 32(6), 923-943.

Mueller, D. S. (2014, June 2). PEBL: How can I Contribute. Consultado en Junio 6, 2017, en el sitio: [http://pebl.sourceforge.net/wiki/index.php/How\\_can\\_I\\_contribute](http://pebl.sourceforge.net/wiki/index.php/How_can_I_contribute)

Zuluaga, JA. (2005). *Neurodesarrollo y estimulación*. Ed. Panamericana. Colombia. 3ra. Reimpresión. 29

Gomes, H., Wolfson, V., Halperin, J. (2006). Is there a selective relationship between language functioning and auditory attention in children? *J Clin Exp Neuropsychol*. 29:6:660-668.