

# Software Educativo Didáctico para la Oralización de Niños Hipoacúsicos

Miguel Angel Navarrete Sánchez<sup>a</sup>, Osbaldo Vite Chávez<sup>a</sup>, Roberto Olivera Reyna<sup>a</sup>, Reynel Olivera Reyna<sup>a</sup>, Eduardo García Sánchez<sup>a</sup>, Claudia A. Rivera Romero<sup>b</sup>, Elvia Palacios Hernández<sup>b</sup>, Antonio Marín Hernández<sup>c</sup>

<sup>a</sup>Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica Campus Jalpa  
Libramiento Jalpa Km. 156+380, Fraccionamiento Solidaridad, Carretera Guadalajara-Salttillo, Jalpa Zacatecas México, CP 99601.  
<http://campusjalpa.uaz.edu.mx/>

<sup>b</sup>Universidad Autónoma de San Luis Potosí. Facultad de Ciencias. Departamento de Electrónica.  
Av. Salvador Nava Mtz. s/n Zona Universitaria, San Luis Potosí, Mexico, C. P. 78290.  
[galia.fc.uaslp.mx/~epalacios](mailto:galia.fc.uaslp.mx/~epalacios)

<sup>c</sup>Universidad Veracruzana. Departamento de Inteligencia Artificial.  
Sabastián Camacho No. 5, Zona Centro, Xalapa, Veracruz, México. C.P. 91090  
<https://www.uv.mx/dia/>

2016 Published by DIFU<sub>100ci</sub>@ <http://difu100cia.uaz.edu.mx>

---

## Resumen

Actualmente, en la intervención educativa, se clasifican a los niños con dificultades auditivas, en sordos profundos y en hipoacúsicos, los cuales presentan una audición limitada. A pesar de ello, logran desarrollar un nivel de lenguaje y comunicación propio, por debajo de los niños sin estas dificultades. Con la ayuda de diferentes dispositivos y herramientas computacionales, se les puede apoyar para que se integren a la sociedad con mayor facilidad en todos los ámbitos al desarrollar estas habilidades de lenguaje y comunicación. Por lo anterior, en este trabajo, se presenta una herramienta de software educativo, que tiene como objetivo fortalecer el proceso de la enseñanza-aprendizaje del lenguaje y de la oralización para niños hipoacúsicos que cursan el nivel básico. El software fue generado utilizando el modelado de proceso para desarrollar entornos didácticos interactivos computacionales (EDIC). En el presente trabajo se puede apreciar la culminación del desarrollo de software como herramienta para su instalación en instituciones educativas de nivel básico y en los hogares.

*Palabras clave:* Enseñanza-Aprendizaje, Hipoacúsico, Oralización, Software Educativo.

---

## 1. Introducción

Una persona con capacidades diferentes se puede definir como “toda persona que por razón congénita o adquirida presenta una o más deficiencias de carácter físico, mental, intelectual o sensorial permanente o temporal” [1], y que al interactuar con el

entorno social, son restringidos en igualdad de condiciones con respecto a las demás personas. La mayor parte de ellos se comunican con pensamientos, necesidades, sentimientos e ideas a través del habla y estas son percibidas a través de la audición y visión; actualmente en la intervención educativa, se clasifica a los niños con dificultades auditivas en: hipoacúsicos, que

presentan una audición limitada, pero que, con prótesis o sin ella, tienen una audición bastante funcional para la vida diaria permitiendo la adquisición del lenguaje oral por vía auditiva, ver Fig. 1 y sordos profundos, donde su audición no funciona [2].

La Dirección General de Planeación y Estadística Educativa (DGPyEE) de la Secretaría de Educación Pública (SEP) presenta las principales cifras del Sistema Educativo Nacional 2014-2015, sobre la población total atendida en educación especial, que es de 563 mil 976 mexicanos, esto, incluyendo a los alumnos que presentan necesidades educativas especiales temporales o permanentes, que pueden o no estar asociadas a una discapacidad o aptitudes sobresalientes. Para el caso particular de la hipoacusia, se atendió a 8 mil 383 alumnos, teniendo un incremento a comparación del periodo 2013-2014, que fue de 7 mil 964 [3].

Hoy en día, el software educativo se visualiza, como uno de los pilares del sistema educativo a distancia, y se perfila como la herramienta principal, utilizarán las próximas generaciones de educandos [4], es por ende, que en este trabajo, se desarrolló un software como complemento didáctico para el proceso de la enseñanza-aprendizaje del lenguaje y oralización de niños Hipoacúsicos, que permita al usuario con esta discapacidad, disponer de su propio diccionario multimedia, esto, con elementos multimedia, como: imágenes, videos, texto y audio, apoyados de sus maestros en la escuela, de maestros de lenguaje y comunicación que los apoyan de las Unidades de Servicio de Apoyo a la Educación Regular (USAER) y de los Centros de Atención Múltiple (CAM), y de sus padres de familia en el hogar, adaptándose de forma pertinente el material, a las necesidades de aprendizaje de contenidos de cada alumno. En la actualidad, existen una gran cantidad de herramientas multimedia que tienen como objeto apoyar el proceso enseñanza-aprendizaje, las cuales, se consiguen a un precio no tan accesible en algunos casos, aunque también pueden ser gratuitas o de libre distribución, pero sin la certeza de que cumplan con las necesidades de los usuarios. En el desarrollo de software educativo, es conveniente la participación de un equipo de expertos multidisciplinarios en diferentes áreas del conocimiento para cumplir con los requerimientos y necesidades que marcan los usuarios [5, 6]. El desarrollo y generación de software multimedia educativo ha avanzado gracias al uso de los lenguajes de programación, lenguajes de autor y sistemas de autor [7], que permiten generar aplicaciones más amigables y funcionales, y que en base a los requerimientos, necesidades y características planteadas, facilitan su desarrollo y generación. Sin embargo, en la mayoría de los casos, el diseñador no



Figura 1. Niño hipoacúsico con aparato, tomada de [8].

cuenta con amplios conocimientos sobre programación y el uso de los lenguajes de programación y de autor, no sabe utilizarlos. Sin embargo, esto no se presenta cuando el software es diseñado por un grupo multidisciplinario, en el que participa un experto de programación. Aunque es importante mencionar, que existen los sistemas de autor, que no utilizan comandos y sentencias propias de los lenguajes de programación y de autor, sino que trabajan a través de una interfaz más amigable, utilizando menús, iconos, plantillas, etc., que permiten generar y desarrollar el software educativo, de una manera más sencilla y práctica, y que no se requiere ser un experto en el área de computación.

## 2. Método

Para poder desarrollar software multimedia de calidad, es necesario, como ya se había mencionado anteriormente, asegurarnos que reúna los requisitos y necesidades del cliente y/o usuarios al que va dirigido, para lo cual, se debe seguir un proceso para su generación, denominado metodología, la cual, puede ser constituida como una combinación de los métodos existentes [9] en ingeniería de software (método de evolutivo, desarrollo en espiral, proceso incremental, ciclo de vida, en cascada, en reutilización, desarrollo concurrente, basada en componentes, proceso unificado, etc.) que permiten

describir las etapas o fases de producción [5, 14].

### 2.1. Metodología del desarrollo

En la literatura existe una gran cantidad de metodologías propuestas para la generación de software multimedia [4, 10, 11, 12, 13, 14]. Sin embargo, en este trabajo, se utilizó la metodología denominada El modelado del proceso para desarrollar Entornos Didácticos Interactivos Computacionales (EDIC) [15] que especifica claramente las actividades a desarrollar en cada una de sus fases o etapas, permitiendo que los sub-productos generados en cada una, por medio de un seguimiento lógico, formen un software educativo de calidad. Esto por la creciente necesidad de elaborar más y mejores recursos digitales que apoyen al proceso de enseñanza-aprendizaje. La metodología EDIC está dividida en seis etapas ver Fig. 2, las cuales se describen más adelante, y permiten enfocarse en generar una descripción detallada de la información que se debe ir generando en cada una de ellas, esto, con el fin de facilitar el proceso de implementación de la metodología. Cabe señalar que todo el proceso es iterativo, por lo que una vez que se concluyen todas las etapas, se puede regresar a la etapa que se considere requiere mayor detalle o presente errores. Así mismo, al final del proceso, se realiza un análisis de mejora continua, que permite detectar áreas de oportunidades y mejoras continuas en el proceso, esto, para refinar las prácticas aprendidas y adoptadas durante el proceso.

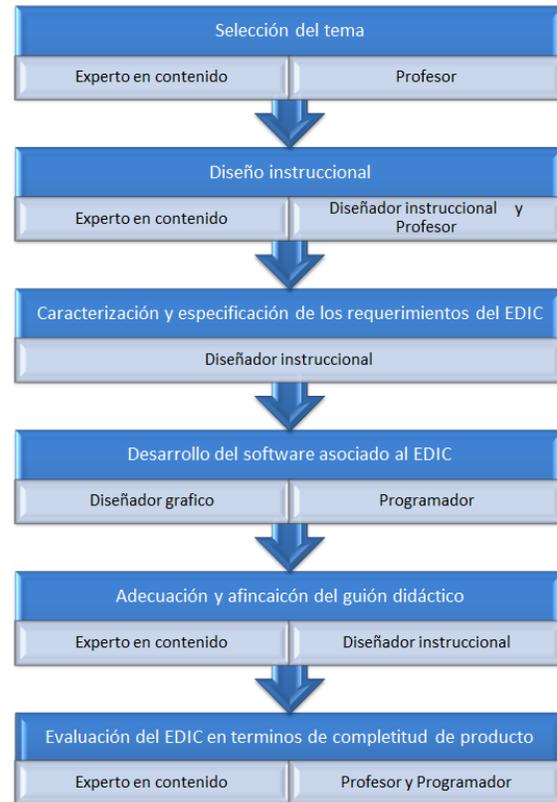


Figura 2. Etapas del modelo de procesos.

- A Etapa 1 Selección del tema. Se basa en el concepto que un profesor prioriza como fundamental, y se apoya en los expertos en contenido para determinar el nivel de profundidad del mismo; esto, para fomentar la comprensión por los estudiantes. La información se contempla por:
  - 1) Instrumento de especificaciones del tema.
  - 2) Evaluación diagnóstica realizada al inicio del curso.
  - 3) Ficha descriptiva del tema a cubrir.
- B Etapa 2 Diseño instruccional. Es el proceso a seguir para obtener el guion didáctico preliminar, participan en dicho proceso de análisis y desarrollo, el experto en contenido, el profesor y el diseñador instruccional.
  - 1) Instrumento del guion didáctico.
  - 2) Ficha descriptiva del tema a cubrir.
  - 3) Instrumento preliminar del guion didáctico.
- C Etapa 3 Caracterización y especificación de los requerimientos del EDIC. Se forman los requerimientos de la actividad académica, y se esquematizan

los diversos escenarios funcionales y adaptables en el software de apoyo por medio del diseñador instruccional.

- 1) Instrumento preliminar del guion didáctico.
  - 2) Documentos de especificación de requerimientos funcionales del software de apoyo.
- D Etapa 4 Desarrollo del software asociado al EDIC. Es considerada la metodología de diseño, implantación e implementación, en la que participan el programador y diseñador gráfico, sus actividades son:
    - 1) Documento de especificación de requerimientos del software de apoyo.
    - 2) Prototipo funcional del software para EDIC.
  - E Etapa 5 Adecuación y afinación del guion didáctico. Se realiza la revisión y prueba piloto, donde se identifican los elementos que requieran adecuación, participan los expertos en contenido y el diseñador instruccional. La información es contemplada por:
    - 1) Prototipo funcional del software para EDIC.
    - 2) Guion didáctico terminado.
    - 3) Software funcional.
  - F Etapa 6 Evaluación del EDIC en términos de completitud de producto. Una vez que se ha probado el



Figura 3. Caso de uso de SORHI.

EDIC en el aula, se realiza una evaluación donde participan los expertos en contenido, el profesor y el programador; se utilizan instrumentos de medición que cuantifican elementos tecnológicos, pedagógicos, de contenido y estéticos. Las actividades de la etapa son:

- 1) Guion didáctico terminado.
- 2) Software funcional.
- 3) Documento para evaluación de completitud.
- 4) Guiones didácticos.
- 5) Software para EDIC.

### 3. Resultados

La herramienta didáctica generada, denominada SORHI (Software como complemento didáctico para la oralización de niños hipoacúsicos) conjuga el código fuente con el diseño de interfaces que cumplen con los requerimientos que marcan las etapas del EDIC, obteniendo como resultado, la primera parte del desarrollo del software educativo. Su funcionalidad se visualiza en la Fig. 3, y que se describe de la siguiente manera: Primeramente, al iniciar el software aparece la interfaz de inicio de sesión, solicitando el nombre de usuario y la clave de acceso; se verifica su autenticidad y después muestra la pantalla principal; a continuación se visualiza la información global de SORHI, como se puede



Figura 4. Interfaz principal modo administrador.

observan en la Fig. 4. En este punto existen dos vistas, el modo administrador (profesor) y por default el de usuario (alumno):

1. Modo Administrador (profesor). Al desplegar la pantalla principal, aparecen inhabilitados algunas opciones de botón en el menú, en la parte superior, con el objetivo de que la interfaz inicie en modo alumno, al pulsar sobre el botón administrador, la aplicación solicita la contraseña, para activar el modo administrador, con el cual, se tiene acceso a las opciones; agregar (*cargar un archivo de imagen, audio y escribir un texto*), modificar y eliminar término. Además de todas las funciones con las que cuenta la interfaz. Ésta distinción se efectúa para evitar que los alumnos obtengan acceso a las funciones en las que realizan acciones no deseadas o, que modifiquen la información.
2. Modo Usuario (Alumno). Al desplegar la pantalla principal, como se mencionó anteriormente, se accede al modo Usuario. La funcionalidad de la ventana del menú principal, se lleva a cabo, al dar un clic a cualquiera de los botones, exceptuando aquellos destinados para el administrador. Las opciones de botones activados para el modo usuario son: buscar término, ayuda, flechas de navegación, escuchar audio, visualización de imagen y su texto descriptivo, además, una sección de pequeños botones con texto, que comprenden las letras del abecedario, los cuales permiten acceder a una lista de términos relacionados a la letra correspondiente, y finalmente la opción salir de SORHI, como se observa en la Fig. 5.

El software educativo, tiene características particulares en cuanto a la comunicación con el usuario, las cuales no se logran cuantificar mediante métricas, porque están relacionadas con conductas de aprendizajes [4]. Pero, las reglas en la construcción de un programa son las mismas, ya sea educativo, comercial, de



Figura 5. Interfaz principal modo usuario.

investigación, etc. Una vez generado el código fuente, el software debe probarse para descubrir tantos errores como sea posible antes de ser un producto final. Las pruebas se desarrollaron desde dos perspectivas diferentes:

1. Caja blanca. La lógica de programa interno se verificó con casos de prueba que:
  - Garantizaron que todas las rutas independientes dentro de un módulo se revisarán al menos una vez.
  - Se revisaron todas las decisiones lógicas en sus lados verdadero y falso.
  - Se ejecutaron todos los bucles en sus fronteras y dentro de sus fronteras operativas.
  - Se revisaron estructuras de datos internas para garantizar su validez.
2. Caja negra. Los requerimientos de software permiten derivar un conjunto de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Estas pruebas permitieron encontrar errores en las categorías siguientes:
  - Funciones incorrectas o faltantes.
  - Errores de interfaz.
  - Errores en las estructuras de datos.
  - Errores de comportamiento o rendimiento.
  - Errores de inicialización y terminación [5].

#### 4. Conclusiones

Debido al creciente desarrollo del software educativo durante los últimos años, gran parte del mismo, ha sido realizado en forma desorganizada y poco documentada, y considerando el aumento exponencial que sufrirá en los próximos años, surge la necesidad de lograr metodologías multidisciplinarias para su desarrollo, por esta

razón, en este trabajo, se utilizó la metodología EDIC, siendo cuidadosos en seguir las etapas que comprenden dicha metodología, permitiéndonos llevar un buen diseño estructural e instruccional del contenido de la herramienta incluido dentro del sistema, y de esta manera, tener las bases necesarias para desarrollar un sistema que cumpla con los requerimientos y necesidades del usuario al que va dirigido el software. Esto permite construir programas educativos de calidad, siguiendo las pautas de las teorías educativas y de la comunicación, ya que el software es un elemento clave en la evolución de sistemas y productos basados en computadoras, y una de las tecnologías más importantes en todo el mundo. Esto nos permite concluir que se generó el software de Calidad denominado SORHI, el cual puede ser adoptado para ser utilizado en instituciones USAER y CAM que puedan hacer uso de él. como apoyo didáctico para los niños Hipoacúsicos a los que atienden.

#### 4.1. Perspectivas

Al implementar el software SORHI, se podrían efectuar casos de estudio con niños hipoacúsicos, apoyados por sus maestros del nivel básico, personal de las USAER y CAM que atienden a estos niños y sus respectivos familiares, que permitan validar el software con respecto a los objetivos planteados en el Diseño Instruccional, al efectuar una evaluación de conocimientos y aprendizajes esperados, con respecto a los aprendidos.

De acuerdo con las recomendaciones del estudio principal de la problemática y necesidad, los resultados como aplicación en las pruebas de caja blanca y negra fueron viables, no obstante aún existen necesidades que complementan en gran medida el desarrollo del software y se han considerado como perspectivas:

- Agregar módulos que permitan la visualización de animaciones y elementos multimedia.
- Insertar un módulo que evalúe el aprendizaje y aprovechamiento de los alumnos.

#### Referencias

- [1] SEP., *Centro Digital de Recursos de Educación Especial.*, Disponible: <http://educacionespecial.sepdf.gob.mx/institucional-que-es-educacion-especial.php>, mayodel2012.
- [2] José del Río, María., *Lenguaje y comunicación en personas con necesidades especiales.*, Barcelona: Martínez Roca, 1997.
- [3] DGPYEE., *Principales Cifras del Sistema Educativo Nacional 2014 - 2015.* Disponible: [http://planeacion.sep.gob.mx/assets/images/principales\\_cifras/2014\\_2015\\_PRINCIPALES\\_CIFRAS\\_BOLSILLO.pdf](http://planeacion.sep.gob.mx/assets/images/principales_cifras/2014_2015_PRINCIPALES_CIFRAS_BOLSILLO.pdf).

- [4] Galvis, A. (2000)., *Ingeniería de software educativo.*, Colombia: Universidad de los Andes.
- [5] Presmman, R. S. , *Ingeniería del Software. Un enfoque práctico.*, USA: Mc Graw Hill.
- [6] Marcano, I. & Benigni, G., *Análisis de alternativas metodológicas para el desarrollo de software educativo.*, Saber, 26(3): 297-304.
- [7] Razquin, P. , *Los sistemas de autor multimedia.*, Revista General de Información y Documentación, 8 (2), 127-139.
- [8] Infogen, *Defectos al nacimiento.* Disponible: <http://infogen.org.mx/sordera-congenita/>.
- [9] Braude, E., *Ingeniería de Software: Una perspectiva orientada a objetos.*, México, D.F.: Alfaomega.
- [10] Benigni, G. (2004)., *Una metodología orientada a objetos para la producción de software multimedia.*, Saber, 16(1): 26-32.
- [11] García Á. J. L. (2011)., *Análisis de los aspectos y factores que se consideran en el diseño de software educativo de calidad.*, Journal La Sociología en sus escenarios, (24).
- [12] Marqués, P. (2005)., *Modelos de Desarrollo de MDCS. La metodología de Pere Marqués.*, Recuperado de <http://www.peremarques.net/disdesa.htm>.
- [13] Polo, M. (2003)., *Aproximación a un modelo de Diseño: ADITE.*, Docencia Universitaria, 1(4): 67-83. Recuperado de [http://www.ucv.ve/fileadmin/user\\_upload/sadpro/Documentos/docencia\\_vol4\\_nl\\_2003/7\\_art.\\_4Marina\\_Polo.pdf](http://www.ucv.ve/fileadmin/user_upload/sadpro/Documentos/docencia_vol4_nl_2003/7_art._4Marina_Polo.pdf).
- [14] Sommerville, I., *Ingeniería de Software.*, España: Pearson y Addison Wesley. 2007.
- [15] Cruz Flores, René., *Modelado del proceso para desarrollar entornos didácticos interactivos.*, Guadalajara Jalisco: s.n., 2012. Congreso Internacional de Investigación e Innovación en Ingeniería de Software 2012.