

Tipo de artículo: Artículo de revisión

Análisis y desarrollo de interfaz gráfica de usuario (GUI)

Analysis and development of the graphical user interface (GUI)

Jorge Chicala Arroyave ^{1*} , <https://orcid.org/0000-0001-9630-2377>

Jenny Arízaga Gamboa ² , <https://orcid.org/0000-0002-2098-9077>

Eduardo Alvarado Unamuno ³ , <https://orcid.org/0000-0001-6145-7926>

¹ Universidad de Guayaquil, Ecuador. E-Mail: jorge.chicalaa@ug.edu.ec

² Universidad de Guayaquil, Ecuador. E-Mail: jenny.arizagag@ug.edu.ec

³ Universidad de Guayaquil, Ecuador. E-Mail: eduardo.alvaradou@ug.edu.ec

* Autor para correspondencia: jorge.chicalaa@ug.edu.ec

Resumen

Los diferentes avances tecnológicos han tenido repercusiones en la educación de las personas, en especial en el proceso de aprendizaje de los niños. El objetivo de la presente investigación es la definición de un prototipo y validación de la Interfaz Gráfica de Usuario (Graphical User Interface, GUI, inglés) de una plataforma que tiene como misión el reforzar terapias médicas enfocadas en el problema de aprendizaje en niños y niñas con dislexia a la vez, dicha interfaz también será de gran utilidad para los terapeutas, denominados logopedas, puesto que podrán crear, editar y clonar terapias previamente definidas, adicional a agregar contenido multimedia, con la finalidad de enriquecerlas con material didáctico y lúdico. Como herramienta de desarrollo del componente de modelado de las terapias se utilizó el framework Vaadin, basado en el lenguaje de programación Java, el cual es una plataforma de código abierto para el desarrollo de aplicaciones web. El objetivo del estudio es comprobar el beneficio que las nuevas tecnologías le brindan a los tratamientos psicopedagógicos, en este caso, la dislexia. A lo largo de la investigación se desarrollará los distintos conceptos y plataformas que relacionan a las nuevas tecnologías con el soporte de las terapias de dislexia.

Palabras clave: Interfaz Gráfica de Usuario (GUI), Vaadin, Dislexia, Java.

Abstract

Different technological advances have had repercussions on people's education, especially on the learning process of children. The objective of this research is the definition of a prototype and validation of the Graphical User Interface (GUI, English) of a platform whose mission is to reinforce medical therapies focused on the learning problem in boys and girls With Dyslexia at the same time, this interface will also be very useful for therapists, called speech therapists, since they will be able to create, edit and clone previously defined therapies, in addition to adding multimedia content, in order to enrich them with educational and recreational material. As a development tool for the therapy modeling component, the Vaadin framework was used, based on the Java programming language, which is an open source platform for the development of web applications. The objective of the study is to verify the benefit that new technologies provide to psychopedagogical treatments, in this case, dyslexia. Throughout the investigation, the different concepts and platforms that relate to new technologies with the support of dyslexia therapies will be developed.

Keywords: Graphical User Interface (GUI), Vaadin, Dyslexia.

Recibido: 12/03/2021

Aceptado: 14/07/2021



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

Introducción

Interfaz Gráfica de Usuario, GUI por sus siglas en inglés, es un medio de comunicación informático que permite facilitar la interacción humano computador gracias al uso e intercambio de componentes audiovisuales. No obstante GUI regula la interacción entre ambos elementos del sistema.

Una interfaz tiene como función proporcionar un entorno visual amigable, eficiente y fácil de utilizar, esto facilita la comunicación del usuario con el computador. A la edad de 6 y 7 años aproximadamente la dislexia es más fácil de diagnosticar dado que en las escuelas enseñan el reconocimiento del fonema y grafema, es decir letras y sonidos.

La escasez de plataformas web con interfaz gráfica de usuario interactivas, de código abierto (Open Source) y la falta de componentes legibles son uno de los principales problemas que se presentan para los profesionales terapeutas, dado que no cuentan con las herramientas necesarias para lograr crear y editar terapias didácticas y lúdicas que ayuden a sobrellevar los problemas de aprendizaje que padecen los pacientes con déficit cognitivo como la dislexia.

Condiciones como la dislexia, la disgrafía (trastorno en la caligrafía) y la discalculia (trastorno en las operaciones matemáticas) son trastornos psíquicos que presentan una situación desfavorable en el sistema educativo ecuatoriano, lo que repercute en la desmotivación, la baja autoestima y la deserción escolar.

Este proyecto tecnológico tiene como objetivo analizar y desarrollar un prototipo de la sección interfaz gráfica de usuario del módulo framework, el cual va dirigido a los logopedas y niños (as) con dislexia de 9 a 12 años, proporcionando una plataforma web más usable e interactiva que permita crear terapias, las mismas que ayudarán a reforzar el progreso como tal del paciente y otorgar facilidad a la comunidad educativa.

Materiales y métodos

El término dislexia fue acuñado por un oftalmólogo alemán para describir aquellas dificultades en las lecturas de pacientes adultos que habían sufrido daño cerebral. Es la dificultad específica para el aprendizaje de la lectura, que presentan algunos niños, sin ninguna razón aparente que lo justifique, de tipo pedagógico, intelectual o sociocultural.



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

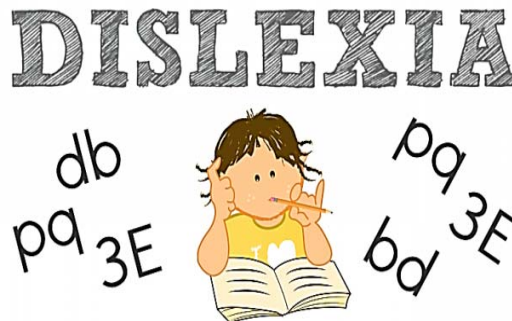


Figura 1. Imagen Referencial de Dislexia
Fuente: Internet

Se define como un desorden que involucra la dificultad de aprender a leer o interpretar palabras, letras, y otros símbolos, pero que no afecta la inteligencia general (Tweeley, 2019). Es independiente del nivel de inteligencia (se rumora que Albert Einstein padecía de esta condición) y tiene cierta vinculación con la audición y el nivel de atención de las personas. Investigadores de la Universidad de Northwestern en Estados Unidos, indican haber descubierto un mecanismo biológico que juega un papel importante en el desencadenamiento de esta condición.

El cerebro de los niños con Dislexia no procesa el sonido del mismo modo que aquellos que no padecen el trastorno. De acuerdo a investigaciones realizadas descubrieron que la forma como el sonido era decodificado por el cerebro se presentaba de forma inestable e irregular. La buena noticia de la investigación es que la alteración puede ser corregida. En lo referente a la escritura, que es un proceso que se desencadena en los niños, casi de forma natural, pero de acuerdo a las estadísticas uno de cada 10 casos se vuelve una lucha.

Se estima que, a nivel mundial, entre 3 y 10% de la población padece algún trastorno psicológico.

Las personas disléxicas no suelen tener problemas con la decodificación de los sonidos que representan las vocales, porque son sonidos relativamente simples y largos. Son los sonidos de las consonantes (más cortos y acústicamente más complejos), lo que resultan difíciles de categorizar.

TIPOS DE DISLEXIA

En la dislexia hay una gran variación individual a pesar de que, en la teoría, se presenta una sintomatología común. En el ámbito educativo, normalmente, se habla de *dislexia del desarrollo* o *dislexia evolutiva*, en contraste con la denominada *dislexia adquirida*, que es aquella provocada tras una lesión cerebral en las áreas encargadas del procesamiento lectoescritor. (Silva, 2011)



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

Dislexia Fonológica o Indirecta: se identifica por daño en la ruta fonológica, es decir la ruta indirecta que utiliza la conversión grafema-fonema con acceso al léxico, los sujetos leen por vía visual, cometiendo errores en palabras desconocidas, palabras poco frecuentes. Palabras polisílabas y pseudopalabras. Presentan errores morfológicos o derivativos, en los cuales mantienen la raíz de las palabras, pero cambian los morfemas.

Dislexia Superficial: Conllevan dificultades se producen en la ruta visual, léxica o directa. Los sujetos leen por la ruta fonológica, por lo que no presentan grandes dificultades en la lectura por medio de la conversión grafema-fonema de palabras regulares, pero manifiestan dificultades en la lectura de irregulares (ej. Palabras de la lengua inglesa), tendiendo a regularizarlas, cometiendo errores de omisión, adición o sustitución de letras.

Dislexia Mixta o Profunda: El déficit esta ocasionado por la lesión o disfunción en ambas rutas, visual y fonológica, lo que provoca que se cometan errores semánticos, es decir, que exista confusión entre palabras que no tienen ningún parecido visual, pero si semántico.

Adicional existe una dificultad en las lecturas de pseudopalabras, palabra función, verbos y palabras poco frecuentes en el acceso al significado y la lectura de palabras por la ruta visual (GENTO PALACIOS & HERNÁNDEZ MORENO, 2012)

INTERFAZ GRÁFICA DE USUARIO

Las Interfaces Gráficas de Usuarios (GUI, del inglés GraphicalUserInterface) son un tipo de interfaz que permite a los usuarios de un sistema interactuar con él a través de iconos gráficos e indicadores visuales, en oposición a las interfaces basadas en texto. Las acciones realizadas en una interfaz gráfica de usuario son realizadas normalmente a través de la manipulación directa de elementos gráficos (Colino Barrigòn, 2015).

El concepto de interfaz es un concepto amplio que ha sido definido, según el ámbito de conocimientos, desde varios puntos de vista: desde la biología (interfaz), ha sido definida como la “capa” de un organismo que separa su interior del exterior, desde la electrónica y las telecomunicaciones, se ha definido como “puerto a través del que se envían o reciben señales desde un sistema o subsistemas hacia otros (Gutierrez Miranda, 2014).



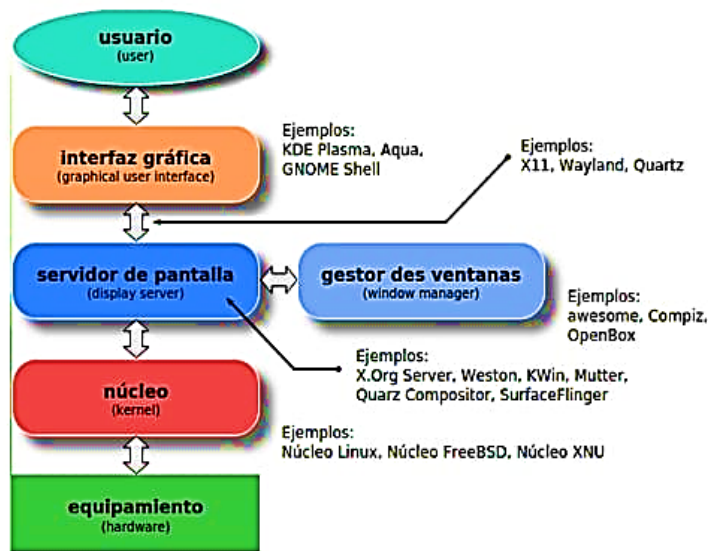


Figura 2. Interfaz Gráfica de Usuario

Fuente: <https://www.linuxadictos.com/wp-content/uploads/entorno-de-escritorio.jpg>

Elaborado por: Linux Adictos

Existen varios lenguajes de programación para la creación de interfaces gráficas de usuarios, uno de los más conocidos y utilizado es Java, adicional existen varios marcos de desarrollo o también denominados frameworks para la creación de aplicaciones web basadas en este lenguaje de programación, entre ellos se encuentra Vaadin de los cuales se hablan a continuación:

Uno de los lenguajes de programación más usados en las aplicaciones es Java ya que es un lenguaje multiplataforma, ya que se ejecuta en diversos dispositivos que poseen el componente denominado Java Virtual Machine (JVM). Posee una amplia documentación de manuales en el cual se muestran las funciones y prestaciones de las diferentes APIs de programación, las cuales permite usarlas desde su primera lección (Wanumen Silva, Mosquera Palacios, & Edwin, 2017).

Una de las características más importantes de los lenguajes de programación modernos es la portabilidad. Como se ha comentado antes, un programa es portable cuando es independiente de la plataforma y puede ejecutarse en cualquier sistema operativo y dispositivo físico. Los programas Java son portables porque se ejecutan en cualquier plataforma. La portabilidad de Java ha contribuido a que muchas empresas hayan desarrollado sus sistemas de comercio electrónico y sus sistemas de información en Internet con Java. La evolución de este lenguaje de programación ha



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

sido muy rápida. La plataforma de desarrollo de Java, denominada Java Development Kit (JDK), se ha ido ampliando y cada vez incorpora a un número mayor de programadores en todo el mundo (Martinez Ladrón de Guevara, 2018).

MARCO DE TRABAJO VAADIN (VAADIN FRAMEWORK)

Vaadin es un marco de trabajo basado en Java que hace que sea fácil construir ricas aplicaciones web. La biblioteca incluye componentes de interfaz de usuario que están diseñados para funcionar bien tanto en dispositivos móviles como en dispositivos de escritorio. Los usuarios finales prestarán atención al detalle de las interfaces mientras que el equipo de desarrollo puede concentrarse en la funcionalidad (Vaadin Ltd, 2019).



Figura 3. Vaadin

Fuente: <https://www.predictiveanalyticstoday.com/vaadin/>

Elaborado por: Vaadin

Vaadin soporta dos modelos de programación diferentes: del lado del servidor y del lado del cliente. El modelo de programación dirigido por el servidor es el más potente. Permite un nivel de abstracción de la programación web y que se programe la interfaz de usuario de la misma manera que programarías una aplicación de escritorio con las herramientas convencionales como AWT (AbstractWindowToolkit), Swing o SWT (Standard Widget Toolkit). Pero más fácilmente. Mientras que el modelo de programación web tradicional es una manera divertida de emplear tiempo aprendiendo nuevas tecnologías, (Vaadin, 2016).

Resultados y discusión

COMPARATIVA MARCOS DE TRABAJOS

Una de las formas más naturales de aprender es comparando nuevos conceptos contra aquellos previamente conocidos. Usando algo que ya es conocido, como una referencia, se puede seleccionar y acelerar la adopción de nuevos



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

conocimientos, ya que se hace un re-uso de marcos mentales para dar un contexto y significado a la nueva información, quizás aún, profundizando en el entendimiento de cosas que ya se conocen.

Quizás la pregunta más común en todo proceso de desarrollo es ¿cuál lenguaje de programación o marco de trabajo utilizar para el desarrollo de la aplicación?, Vaadin es un marco de trabajo que está cobrando relevancia en el ámbito del desarrollo de aplicaciones web, ya sea por su simplicidad de uso, por la librería que posee varios componentes visuales o su facilidad de implementación:

Tabla 1. Comparación de Framework de Desarrollo con Vaadin

Fuente: <https://vaadin.com/comparison>

Elaborado por: Vaadin

	Vaadin	React	Angular
Visión General	<p>Es un marco web full-stack que se ejecuta en java: Posee:</p> <ul style="list-style-type: none"> • Una API de componentes basados en Java. • Comunicación automática servidor-cliente a través de XHR o WebSockets. • Sistema de diseño personalizado con más de 40componentens de interfaces. • Navegación (Routing). • Formularios (Form). • Internacionalización. • Inyección de Dependencia (Spring y CDI). <p>Marco único que permite a los desarrolladores crear aplicaciones de una sola página en Java. Puede usar HTML, Javascript y CSS para personalización.</p>	<p>Es una biblioteca de JavaScript para crear interfaces de usuario. Incluye:</p> <ul style="list-style-type: none"> • Un modelo de componentes. • Plantillas declarativas reactivas. • Representación DOM eficiente. • A menudo se considera un marco, pero de hecho es un modelo de componentes. <p>El modelo de componente des flexible y se puede combinar con bibliotecas de terceros para funciones, como enrutamiento y administración de estados.</p>	<p>Es un marco frontend de aplicaciones web y móviles frontend. Tiene:</p> <ul style="list-style-type: none"> • Un modelo de componente basado en TypeScript. • Plantillas declarativas reactivas. • Inyección de dependencia. • Internacionalización. • Animaciones. • Formularios. • Módulos. • Componentes de la interfaz de usuario de diseño de materiales. <p>Es un marco de interfaz con muchas de las características a las que están acostumbrados los desarrolladores empresariales, como módulos e inyección de dependencias. Angular se aleja de las tecnologías web subyacentes e introduce varios conceptos específicos del marco.</p>
Modelo de Componente	<p>Los componentes pueden extender otros componentes, elementos HTML (Div) o diseños de alto nivel como VerticalLayout. Aplicaciones y las vistas son construidas componiendo uno más componentes juntos.</p>	<p>Los componentes se pueden definir como clases o como funciones. Los componentes funcionales son el enfoque recomendado para nuevas aplicaciones.</p>	<p>Los componentes constan de una definición de TypeScript y una plantilla HTML.</p>
Plantillas	<p>No requiere plantillas. En su lugar, puede crear vistas</p>	<p>Las plantillas se declaran utilizando JSX, una extensión</p>	<p>Tiene una sintaxis de plantilla completa y específica del marco que se basa en HTML.</p>



Esta obra está bajo una licencia **Creative Commons de tipo Atribución 4.0 Internacional (CC BY 4.0)**

	mediante programación con Java.	de sintaxis de JavaScript que permite a los desarrolladores escribir una sintaxis similar a HTML en JavaScript.	
Plantillas: Enlaces de Datos	Los valores de texto se establecen mediante el constructor o mediante establecedores (setters). Es fácil encontrar las posibles propiedades usando autocompletar en el IDE utilizado. Debido a que las vistas se construyen en Java, todas las API se escriben. Los componentes, como select y data grid, utilizan genéricos para especificar el tipo de datos utilizados.	Puede vincular valores en la plantilla utilizando corchetes {}. Puede vincular valores, como variables o funciones. La encuadración funciona tanto para el contenido del texto como para las propiedades. A diferencia de HTML, JSX no diferencia entre atributos y propiedades. Puede vincular tanto valores primitivos como valores complejos, como objetos o matrices.	La sintaxis de enlace de datos depende del tipo de enlace: <ul style="list-style-type: none"> • <code>{{variable}}</code>: los corchetes dobles se utilizan para interpolar valores dinámicos en el contenido del texto. La interpolación admite operaciones simples como la suma, siempre que el valor se pueda convertir en una cadena. No puede usar JavaScript arbitrario. <code>[property] = "valor"</code> : se utiliza para vincular un valor a una propiedad. Los valores pueden ser primitivos o tipos de datos complejos.
Plantillas: Eventos	Cualquier componente con el que interactúa un usuario dispara eventos. Puede suscribirse a estos eventos a través de la API <code>addListener</code> .	Puede escuchar eventos enlazándose a una función de controlador. A diferencia de HTML, los nombres de eventos en JSX están en formato camel.	Puede vincular a eventos colocando el nombre del evento entre paréntesis. Puede usar el token <code>\$ event</code> para pasar el evento al controlador

INTERFAZ GRAFICA DEL SOFTWARE

Un software educativo para la reeducación de la dislexia debe estar fundamentado por la integración de áreas del conocimiento tales como la pedagogía, la psicología, la tecnología, entre otros; considerando las áreas que se desean estimular o reeducar en el niño disléxico. En el proceso de desarrollo del sistema se debe tomar en cuenta diversos factores como las áreas que se deben reeducar en un niño disléxico, las teorías cognitivas del aprendizaje, el equipo multidisciplinario involucrado en la reeducación, las técnicas de reeducación, el diseño de sistemas interactivos y lúdicos, los modelos y métodos de desarrollo de software centrado en el usuario (Díaz, 2016).

Algunas de las terapias creadas con el software tienen por objetivo ayudan a mejorar ciertos errores presentados en los disléxicos como:

- ✓ Confusión de letras, sílabas o palabras con diferencias sutiles de grafía: a-o, c-ch, c-o, e-c, f-t, i-j, l-ll, m-n, n-ñ, v-u, v-y, etc.



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

- ✓ Confusión de letras, sílabas o palabras con grafía similar, pero con distinta orientación en el espacio: b-d, b-p, d-b, d-p, d-q, n-u, w-m, a-e. o Confusión de letras que poseen un punto de articulación común y cuyos sonidos son acústicamente próximos: d-t, ch-ll, g-j, m-b-p, v-f.
- ✓ Inversiones parciales o totales de sílabas o palabras: la-al, le-el, las-sal, los-sol, loma-malo, etc.



Figura 4. Sistema Temonet
Elaborado por: Grupo Temonet

- ✓ Sustituciones e inversiones de palabras por otras de estructura más o menos similar, pero con diferente significado: araucano-iracundo.
- ✓ Contaminaciones de sonidos.



Figura 5. Sistema Temonet
Elaborado por: Grupo Temonet

- ✓ Adiciones u omisiones de sonidos, sílabas o palabras: famoso por fama, casa por casaca.
- ✓ Repeticiones de sílabas, palabras o frases.



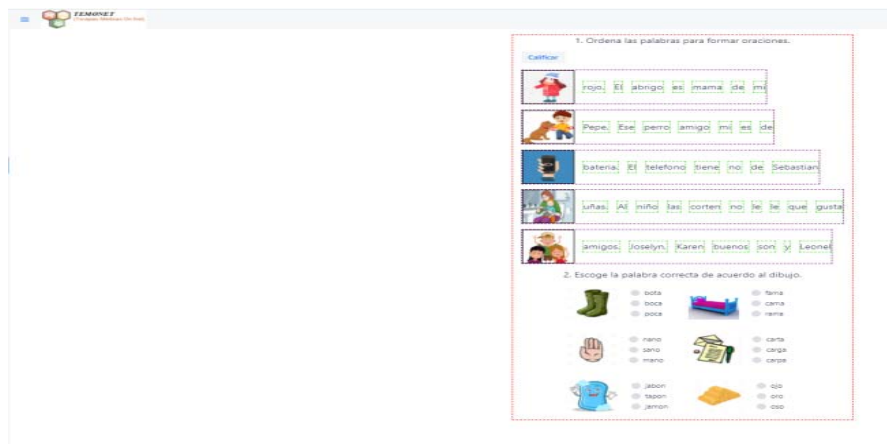


Figura 6. Sistema Temonet
Elaborado por: Grupo Temonet

Conclusiones

Conforme al desarrollo del presente proyecto del prototipo de la sección interfaz gráfica de usuario utilizando Framework Vaadinse puede concluir que al ofrecer esta herramienta que permite gestionar dinámicamente contenidos será de gran utilidad para las necesidades de los logopedas al momento de administrar una terapia, de igual manera será de gran ayuda para sobrellevar los problemas que presentan los niños y niñas con déficit cognitivo como la dislexia.

Se puede concluir que el uso de herramientas tecnológicas con interfaz gráfica de usuario es de gran utilidad para mejorar la calidad de vida de las personas que padezcan estos tipos de trastorno como la dislexia evolutiva; de la misma forma darles la oportunidad de integrarse a la sociedad sin mayores dificultades.

Conflictos de intereses

Los autores expresan que no poseen conflictos de intereses.

Contribución de los autores

1. Conceptualización: Jorge Chicala Arroyave.
2. Curación de datos: Jenny Arízaga Gamboa.
3. Análisis formal: Eduardo Alvarado Unamuno.
4. Investigación: Eduardo Alvarado Unamuno.



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

5. Metodología: Jenny Arízaga Gamboa.
6. Administración del proyecto: Jorge Chicala Arroyave.
7. Software: Jenny Arízaga Gamboa.
8. Validación: Jorge Chicala Arroyave.
9. Visualización: Jorge Chicala Arroyave.
10. Redacción – borrador original: Jorge Chicala Arroyave, Jenny Arízaga Gamboa, Eduardo Alvarado Unamuno.
11. Redacción – revisión y edición: Jorge Chicala Arroyave, Jenny Arízaga Gamboa, Eduardo Alvarado Unamuno.

Financiamiento

La investigación ha sido financiada por los autores.

Referencias

- Casanoba, E., & Ceci, R. (2016). *Sitios multiplataforma con HTML5 + CSS3*. RedUsers.
- Colino Barrigòn, J. (2015). *Desarrollo de una interfaz gràfica para un sistema electrònico de caracterizaciòn òptica de diodos emisores de luz*. Obtenido de <http://uvadoc.uva.es/bitstream/handle/10324/15207/TFG-G1677.pdf?sequence=1&isAllowed=y>
- Díaz, M. d. (2016). Proceso de Diseño de la Interfaz de un Sistema Interactivo Educativo Orientado a la Reeducción de las Dificultades en el Aprendizaje que presentan los Niños con Dislexia en Panamá. *Memorias De Congresos UTP, 1(1)*, 9-16.
- Gallardo, D. (26 de Noviembre de 2012). *IBM*. Obtenido de Eclipse: <https://www.ibm.com/developerworks/ssa/library/os-ecov/index.html>
- GENTO PALACIOS, S., & HERNÁNDEZ MORENO, M. J. (2012). *Tratamiento Educativo de la Diversidad en Audición Y Lenguaje*. Madrid: Editorial UNED.
- Gutierrez Miranda, M. (Agosto de 2014). *Propuesta de diseño modular para la configuración de un entorno virtual de enseñanza-aprendizaje con tutoría inteligente Prototipo DECANO*. Obtenido de <https://pdfs.semanticscholar.org/d526/1ef54c316b399b767658274a98380c5a37df.pdf>
- Luzardo Alliey, A. M. (Agosto de 2009). *DISEÑO DE LA INTERFAZ GRAFICA WEB EN FUNCIÓN DE LOS DISPOSITIVOS MÒVILES*. Buenos Aires.



Esta obra está bajo una licencia **Creative Commons de tipo Atribución 4.0 Internacional (CC BY 4.0)**

- Martinez Ladrón de Guevara, J. M. (16 de Julio de 2018). *Fundamentos de Programación en Java*. Obtenido de <https://www.tesuva.edu.co/phocadownloadpap/Fundamentos%20de%20programacion%20en%20Java.pdf>
- Robledo, D. (2017). *Desarrollo de aplicaciones para Android I. Ministerio de Educación, Cultura y Deporte*. Colección Aula Mentor.
- Silva, C. (2011). *Tipos de dislexia*. Obtenido de Portal de información sobre dislexia Ladislexia.net.: <http://www.ladislexia.net/tipos-clasificacion/>
- Tweeley, A. (2019). *Dislexia: Guía para Reconocer y Superar la Dislexia [Dyslexia: A Guide to Recognize and Overcome Dyslexia]*. Vincent Noot.
- Vaadin. (2016). *Vaadin 7 Edición - 3º Revisión*. Vaadin Ltd.
- Vaadin Ltd. (25 de Noviembre de 2019). *Vaadin - User Interface Components for busines apps*. Obtenido de <https://vaadin.com/>
- Wanumen Silva, L. F., Mosquera Palacios, D. J., & Edwin, T. (2017). *Java básico*. Bogotá: Ecoe Ediciones.

