

PROPUESTA DE UN PROTOTIPO DE SOFTWARE LIBRE COMO APOYO PARA
ESTUDIANTES CON DISCAPACIDAD VISUAL DE EDUCACION DE NIVEL
SUPERIOR

MAYCOL CÁRDENAS ACEVEDO



FACULTAD DE INGENIERÍA
MAESTRÍA EN SOFTWARE LIBRE
UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
BUCARAMANGA
JULIO
2018

PROPUESTA DE UN PROTOTIPO DE SOFTWARE LIBRE COMO APOYO PARA
ESTUDIANTES CON DISCAPACIDAD VISUAL DE NIVEL SUPERIOR

MAYCOL CÁRDENAS ACEVEDO

Proyecto de tesis para optar por el título de:
MAESTRÍA EN SOFTWARE LIBRE

Director:
Julián Santiago Santoyo Díaz

FACULTAD DE INGENIERÍA
MAESTRÍA EN SOFTWARE LIBRE
UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
BUCARAMANGA
JULIO
2018

Nota de aceptación

Jurado

Jurado

DEDICATORIA

Dedico esta tesis a Dios quien con su infinita bondad me ha dado la fuerza para lograr alcanzar mis metas; a mi familia que con su apoyo incondicional han propiciado mi formación personal, en especial mi abuelo, el cual ha sido un ejemplo a seguir, llenando mi vida de amor y alegría.

AGRADECIMIENTOS

Deseo expresar mis agradecimientos a:

- Dios quien me ha dado la fuerza y la dedicación para realizar mis sueños.
- A mi familia por estar presentes en cada instante de mi vida brindándome su apoyo.
- A mi director, el Doctor Julián Santiago Santoyo Díaz, quien me orientó y asesoró incondicionalmente.
- Gracias a cada uno de los maestros que participaron en mi desarrollo profesional durante la maestría, sin su ayuda y conocimientos no estaría en donde me encuentro ahora.
- Gracias a mis profesores Luis Eduardo Sepúlveda rodríguez y Julián Esteban Gutiérrez que aportaron sus conocimientos en toda mi vida académica y profesional.

RESUMEN

La educación ha tenido a lo largo del tiempo muchos cambios en sus modelos pedagógicos, teniendo una gran preocupación por la forma como los estudiantes aprenden, pero se han dejado de lado las personas con discapacidades, en especial a los discapacitados visuales.

El gran potencial de las nuevas tecnologías ha permitido que se puedan adaptar los entornos académicos para un fácil acceso a la información en especial el *software* libre, el cual nos permite hacer adaptaciones a nuestras necesidades, en nuestro caso a los estudiantes con discapacidad visual.

Se realizó una recolección de información referente al *software* libre que permitiera realizar un material educativo con estándares de accesibilidad y también el *software* que sirviera de plataforma para hacer uso de este material, en pocas palabras los sistemas operativos que servirán de plataforma para soportar todo este ecosistema de aplicaciones adaptadas a las necesidades del usuario con discapacidad visual.

Con esto se puede concluir que la adaptación del entorno de *software* y el material educativo accesible permiten un mejor desempeño de los estudiantes con discapacidad, adicional a esto el entorno altamente adaptable permite apoyar a los alumnos con diferentes grados de discapacidad visual.

PALABRAS CLAVES: Sistemas Operativos, *Software* libre, *Vinux*, Discapacidad visual, Accesibilidad, *Software* Educativo.

ABSTRACT

Education has had many changes in pedagogical models over time, having a great concern for the way how our students learn, but people with visual disabilities have been left out.

The great potential of new technologies has allowed academic environments to be adapted for easy access to information, especially free software, that allows us to adapt our needs, in our case to students with visual disabilities.

A collection of information regarding free software that would allow an educational material with accessibility standards and also the software that would serve as a platform to use of this material, in a few words the operating systems that will serve as a platform to support this ecosystem of applications adapted to the needs of the visually impaired user.

With this, it can be concluded that the adaptation of the software environment and the accessible educational material allow a better performance of students with disabilities, in addition to this the highly adaptable environment allows to support students with different degrees of visual disability.

KEYWORDS: *Operating System, Free Software, Vinux, Visual Disability, Accessibility, Education Software.*

TABLA DE CONTENIDO

	Pág.
INTRODUCCIÓN.....	15
1. MARCO TEÓRICO	17
1.1 DISCAPACIDAD	17
1.1.1 DISCAPACIDAD VISUAL.....	17
1.2 UsABILIDAD Y ACCESIBILIDAD.....	18
1.2.1 ACCESO A LA INFORMACIÓN PARA INVIDENTES	19
1.3 SOFTWARE LIBRE.....	20
1.4 SISTEMAS OPERATIVOS	22
1.5 MODELO EDUCATIVO	24
1.5.1 DISEÑO INSTRUCCIONAL.....	24
1.6 TRABAJO RELACIONADO.....	26
2. PROTOTIPO DE SOFTWARE LIBRE COMO APOYO PARA ESTUDIANTES CON DISCAPACIDAD	29
2.1 PROTOTIPO DE HERRAMIENTAS DE SOFTWARE LIBRE COMO APOYO PARA ESTUDIANTES DE EDUCACIÓN DE NIVEL SUPERIOR CON DISCAPACIDAD VISUAL.....	29
2.2 ANÁLISIS BIBLIOGRÁFICO	31
2.3 EXPLORACIÓN LEVANTAMIENTO DE REQUERIMIENTOS	32
2.4 INSTALACIÓN Y CONFIGURACIÓN.....	32
2.5 INSTALACIÓN DE LA HERRAMIENTA DE AUTOR JCLIC	41
2.6 ELABORACIÓN DE LOS MATERIALES EDUCATIVOS	44
2.7 PRUEBAS A USUARIOS FINALES.....	46
3. RESULTADOS DE LA INVESTIGACIÓN	52
3.1 ANÁLISIS DE VENTAJAS Y DESVENTAJAS DEL PROTOTIPO.....	52
3.2 PROTOTIPO.....	52
3.3 PRUEBAS DE FUNCIONAMIENTO	53
4. CONCLUSIONES	66
5. RECOMENDACIONES Y TRABAJOS FUTUROS	68
REFERENCIAS.....	69
ANEXOS	73

LISTA DE TABLAS

Tabla 1. Resumen sobre derechos y restricciones en las licencias de <i>software</i> privado y <i>software</i> libre	23
Tabla 2. Respuestas modelo Bon	30
Tabla 3. Metodología estado del arte.....	31
Tabla 4. Equipos de cómputo utilizados.....	33
Tabla 5. Selección herramientas de <i>software</i>	34
Tabla 6. Entorno para estudiantes de discapacidad nivel 1	35
Tabla 7. Entornos para estudiantes con discapacidades nivel 2	35
Tabla 8. Entorno para estudiantes con nivel 3 de discapacidad visual.....	37
Tabla 9. Estudiantes Discapacitados Visuales.....	48
Tabla 10. Criterios prueba <i>Think Aloud</i>	51

LISTA DE FIGURAS

Ilustración 1. Diseño Instruccional.....	25
Ilustración 2. Modelo de mejora de proceso	29
Ilustración 3. Comando configuración lector de pantalla ORCA	35
Ilustración 4. Ventana de configuración ORCA.....	36
Ilustración 5. Modulo lector de tarjetas NFC	37
Ilustración 6. Asistente de instalación de sistema operativo <i>Raspbian</i>	38
Ilustración 7. Sensor conectado a <i>Raspberry</i>	39
Ilustración 8. Controladores del sensor.	39
Ilustración 9. Script para activar el sensor.	40
Ilustración 10. Contenido del script.	40
Ilustración 11. Código para activar el material educativo	41
Ilustración 12. Herramienta de visualización <i>JClic</i>	42
Ilustración 13. Herramienta de creación de materiales educativos <i>JClic</i>	43
Ilustración 14. Herramienta de informes <i>JClic</i>	43
Ilustración 15. Material educativo <i>JClic</i>	45
Ilustración 16. Interface gráfica <i>JClic</i>	49
Ilustración 17. Secciones material educativo	50
Ilustración 18. Material educativo en ejecución.....	50
Ilustración 19. Ejecución de material educativo con tarjeta <i>NFC</i>	51
Ilustración 20. Equivalente	53
Ilustración 21. Alternativo	54
Ilustración 22. Directo.....	54
Ilustración 23. Compatible	55
Ilustración 24. Principios generales.....	55
Ilustración 25. Principios generales-2	56
Ilustración 26. Principios generales -3	56
Ilustración 27. Teclado-1	57
Ilustración 28. Teclado-2	57
Ilustración 29. Teclado-3	57
Ilustración 30. Teclado-4	58
Ilustración 31. Teclado-5	58
Ilustración 32. Descriptivos apuntadores-1	59
Ilustración 33. Descriptivos apuntadores-2	59
Ilustración 34. Pantalla-1	60
Ilustración 35. Pantalla-2.....	60
Ilustración 36. Pantalla-3.....	61
Ilustración 37. Sonido y multimedia-1	61
Ilustración 38. Notificación al usuario-1.....	62
Ilustración 39. Información de objetivos	62
Ilustración 40. Tiempo-1	63
Ilustración 41. Documentación-1	63

Ilustración 42. Documentación-2 64
Ilustración 43. Documentación-3 64
Ilustración 44. Otros requisitos-salir 65

LISTA DE ANEXOS

ANEXO A. Instrumentos de evaluación.	73
ANEXO B. Material educativo accesible creado en la herramienta de autor seleccionada (<i>JClic</i>).	74
ANEXO C. Guía de construcción de dispositivos lector de tarjetas <i>NFC</i> para alumnos con casos graves de discapacidad visual.....	75
ANEXO D. Guía de adaptación para discapacitados visuales en sistemas operativos tipo <i>Unix Ubuntu 18</i>	76
ANEXO E. Guía de elaboración de materiales educativos con <i>JClic</i>	77
ANEXO F. Artículo	78
ANEXO G. Video de evidencia del funcionamiento del prototipo	79

GLOSARIO

CIDDM: la Clasificación Internacional de Deficiencias, Discapacidades y Minusvalías (*United Nations*, 2003), Beta-2 (CIDDM-2) pertenece a la "familia" de clasificaciones desarrolladas por la Organización Mundial de la Salud para su aplicación a varios aspectos de la salud y proporciona el lenguaje para codificar una amplia gama de información sobre la salud.

FSF: la Fundación para el *Software* Libre *FSF* (*Free Software Foundation*, 2017) es una organización no lucrativa con una misión en todo el mundo, la cual es promover la libertad de los usuarios de *software*.

OMS: la OMS (Organización mundial de la salud) (Naciones Unidas, 2017) es la autoridad directiva y coordinadora de la acción sanitaria en el sistema de las Naciones Unidas. Es la organización responsable de desempeñar una función de liderazgo en los asuntos sanitarios mundiales, configurar la agenda de las investigaciones en salud, establecer normas, articular opciones de política basadas en la evidencia, prestar apoyo técnico a los países y vigilar las tendencias sanitarias mundiales.

OCR: reconocimiento Óptico de Caracteres, o *OCR* (*ABBYY*, 2017), es una tecnología que le permite convertir diferentes tipos de documentos, tales como documentos escaneados, archivos de *PDF* o imágenes captadas por una cámara digital y los convierte en datos editables con opción de búsqueda.

ADDIE: el modelo *ADDIE* (*Belloch*, 2013) es un marco que lista procesos genéricos que utilizan diseñadores instruccionales y desarrolladores. Representa una guía descriptiva para la construcción de herramientas de formación y apoyo al desempeño eficaz en cinco fases.

W3C: es la sigla utilizada por la *World Wide Consortium* (*The Society of Motion Picture and Television Engineers*, 2013), una organización mundial que se encarga de estudiar, desarrollar y perfeccionar los estándares y recomendaciones que sirve como guía para la elaboración de una web de calidad.

Think-aloud: esta técnica llamada pensando en voz alta (*Boren, M. T., & Ramey, J*, 2000) es utilizada como *test* de usabilidad, consiste en que el usuario utilice sistema y piense en voz alta mientras maneja la interfaz, permitiendo al investigador obtener información de este proceso.

Orca: el proyecto *Orca* (*The GNOME project*, 2017) proporciona acceso al entorno gráfico a través de la voz y braille, por medio de un lector de pantalla.

Raspberry: dispositivo *hardware* enfocado al *internet* de las cosas, desarrollado por la Fundación *Raspberry Pi* (*Raspberry Pi Foundation*, 2016).

Raspbian: sistema operativo basado en *debían*, desarrollado para dispositivos *Raspberry Pi* (*Raspberry Pi Foundation*, 2016).

NFC: tecnología de conectividad inalámbrica de corto alcance basada en estándares.

INTRODUCCIÓN

Los entornos académicos están diseñados para funcionar en plenitud con estudiantes que no poseen ningún tipo de discapacidad. Por esta razón se implementa un entorno de herramientas de *software* que permiten realizar la adaptabilidad necesaria para este tipo de estudiantes, permitiéndoles un mejor desempeño, productividad y calidad de vida.

De acuerdo al *ranking* de universidades en *software* libre clasificación Hispanoamérica (Portal Programas, 2016), solo 8 Universidades del país tienen aportes significativos a la comunidad el *software* libre, siendo la Universidad de los Andes Colombia la que mayor representación tiene con un 9.89%, frente al 100% que posee la Universidad Nacional del Callao en Perú, este bajo porcentaje de aporte y número de universidades, muestran la poca probabilidad que tienen los estudiantes con discapacidad visual para encontrar entornos que sean altamente adaptables para sus necesidades, siendo esta una población vulnerable ya que poseen grandes limitaciones a la hora de adquirir conocimiento con las herramientas de *software* convencionales.

El *software* libre ha sido una pasión, tanto por su diversidad de implementaciones, como por su filosofía de compartir el conocimiento, de esta forma lo expresa *Richard Stallman* en el proyecto *GNU* (FSF, 2017). Esto lleva a compartir el conocimiento con un gran número de personas, beneficiando a nuestra población con discapacidad.

Se decidió usar *software* libre por las ventajas adaptativas y académicas que proporciona, si observamos la definición de *software* libre aportada por la FSF (FSF, 2017) nos damos cuenta que sus libertades nos permiten tener un gran aporte, como la posibilidad de ejecutar el *software* como lo desee, estudiar el código, realizar modificaciones, y poder distribuirlas libremente.

Por este motivo el principal objetivo fue el desarrollo de un prototipo de *software* o un entorno de herramientas de *software* de autor libres, que permiten hacer el uso de materiales académicos universitarios de una forma accesible para estudiantes con discapacidad visual.

La mayoría de los contenidos educativos que se proporcionan en las universidades no están diseñados para todo tipo de estudiantes y sus discapacidades. La intención de compartir conocimiento o de impartir una clase es que llegue a todos los estudiantes sin importar su limitación, por lo que se debe mostrar un alto interés en la forma cómo van a ser consumidos nuestros materiales o cursos.

La investigación permitió conocer las diferentes herramientas de *software* libre que serán utilizadas en los entornos académicos, proporcionando de este modo una utilidad tanto para las personas que deseen crear sus materiales accesibles, como para los estudiantes que consuman los contenidos académicos.

Para el desarrollo de esta investigación se utilizaron una serie de fases:

- Fase 1: Inicialmente se hizo un análisis bibliográfico para entender qué proyectos se han realizado anteriormente y las diferentes áreas del conocimiento que intervienen.
- Fase 2: Se realizó una exploración donde se analizó las necesidades mínimas de los estudiantes con discapacidad visual.
- Fase 3: Se efectuó la instalación y la configuración del entorno de *software*.
- Fase 4: Elaboración del material educativo accesible.
- Fase 5: Se realizaron las pruebas a los usuarios finales.
- Fase 6: Análisis estadístico.

En las últimas fases se pone en funcionamiento el proyecto, para determinar la calidad del material educativo y la usabilidad de las herramientas libres usadas. Por último, para la recolección de estos datos se utilizaron dos pruebas, una llamada Pensando en voz alta y una encuesta escrita, las cuales nos permitieron recolectar información a medida que los alumnos desarrollaban o consumían el material.

Este proyecto se realizó en la Institución Universitaria EAM de Armenia Quindío con los estudiantes del programa de Ingeniería de *Software* entre los semestres 1 y 9, ya que en este programa se encuentran alumnos que poseen este tipo de discapacidades, para la realización de las pruebas se hizo uso de un laboratorio de sistemas en el cual se efectuaron las instalaciones correspondientes.

Con esta investigación se proporciona una serie de guías y manuales que permitan a los docentes realizar las adaptaciones a los entornos académicos y elaborar materiales educativos accesibles para los estudiantes con discapacidad visual.

1. MARCO TEÓRICO

Para tener una visión general del proyecto fue necesario definir una serie de conceptos como una forma de entender la aplicación de este proyecto.

1.1 DISCAPACIDAD

En este apartado describiremos el concepto de discapacidad, tomando como referencia el boletín sobre los discapacitados No 2/1999 la clasificación internacional de las deficiencias, discapacidades y minusvalías (CIDDM)(*United Nations*, 2003), siendo este una de las clasificaciones desarrolladas por la organización mundial de la salud con la finalidad de codificar una amplia gama de información sobre la salud para estandarizar la comunicación en todo el mundo, donde se define de acuerdo a un modelo médico la discapacidad visual “como un problema personal directamente causado por una enfermedad, trauma o estado de salud, que requiere de cuidados médicos prestados en forma de tratamiento individual por profesionales” y definida desde el modelo social como “discapacidad considera el fenómeno principalmente como un problema creado socialmente y principalmente como una cuestión de la integración de las personas en la sociedad. La discapacidad no es un atributo de la persona, sino un complicado conjunto de condiciones, muchas de las cuales son creadas por el ambiente social.”

1.1.1 Discapacidad Visual. Para la definición de discapacidad visual tomaremos de referencias diferentes entidades como la CIE-10(Ministerio de Sanidad, 2016) definiéndonos discapacidad visual como la ceguera, la cual es una discapacidad física que consiste en la pérdida total o parcial del sentido de la vista.

Existen varios tipos de ceguera parcial dependiendo del grado y tipo de pérdida de visión, como la visión reducida, el escotoma, la ceguera parcial (de un ojo) o el daltonismo.

Por otro lado, la OMS (Naciones Unidas, 2017) especifica que la función visual se subdivide en cuatro niveles:

- La visión normal,
- Discapacidad visual moderada,
- Discapacidad visual grave y ceguera.

Estas subdivisiones se agrupan de la siguiente manera:

- Baja visión a las discapacidades visuales moderadas.
- Graves.

La baja visión y la ceguera representan el total de casos de discapacidad visual. Con esto podemos entender tal como se expresa en la guía de accesibilidad en Linux elaborada por *Michael De la Rue, Sharon Snider* (*Michael De La Rue & Sharon Snider, 2002*) en la cual expresan que, si hay dos categorías de discapacidad visual, los que poseen una deficiencia visual como por ejemplo visión borrosa de cerca o de lejos, ceguera a algunos colores y los que son totalmente ciegos.

1.2 USABILIDAD Y ACCESIBILIDAD

Las personas con discapacidad visual necesitan facilidades de uso en su entorno, tanto académico como laboral y las tecnologías de la información proporcionan el conjunto de herramientas adecuado para su desarrollo. Pero primero debemos entender el concepto de usabilidad y accesibilidad, por esta razón tomaremos como referencia la definición de usabilidad o facilidad de uso que nos proporciona los estándares ISO en especial la ISO 9241-11 orientación sobre Usabilidad (1998) (*Usability Net, 2006*) donde describen la facilidad de uso como el grado en que un producto puede ser usado por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso. Este estándar nos proporciona una forma de identificar la información que es necesaria para evaluar la usabilidad en términos de medidas de rendimiento y satisfacción del usuario.

Otra definición que podremos encontrar en las normas ISO es la ISO / IEC 9126: Evaluación de productos de *software* - Las características de calidad y directrices para su uso (1991) Que ya nos proporciona una definición más enfocada al diseño de la interfaz de usuario y especifican que la facilidad de uso es un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la evaluación individual de tal uso, por un conjunto explícito o mínimo de usuarios.

Por último, tomaremos en cuenta otra definición aportada en la norma ISO / IECFDIS 9126-1: Ingeniería de *software* –La calidad del producto–Parte 1: Modelo de Calidad (2000). Donde se especifica que la facilidad de uso es la capacidad del producto de *software* para ser comprendido, aprendido, usado y atractivo para el usuario, cuando se utiliza en condiciones especiales.

Estas definiciones nos permiten tener una visión más general de lo que trata la usabilidad y proporcionará una serie de pautas para desarrollar nuestro material educativo accesible el cual lo utilizaremos en sistemas operativos accesibles para personas con discapacidad visual.

De acuerdo con el artículo publicado por el Mc. Guillermo M. Martínez de la Tejada (Martínez De La Teja, 2003) y las normas ISO antes citadas se puede deducir que un producto usable implica los siguientes puntos:

- Puede ser utilizado en forma adecuada, eficiente y satisfactoria por la mayoría de posibles usuarios.
- Sea tan fácil de aprender a usar que no requiera manuales.
- Pueda ser utilizado por personas con diferentes habilidades o discapacidades.
- Cualquier persona, sin importar su edad o cultura, pueda usarlo.
- Evita que el usuario cometa errores.

1.2.1 Acceso a la información para invidentes. Las TIC son una herramienta fundamental para transmitir la información de una forma adecuada a las personas con discapacidad. Con la intención de entender el concepto de TIC podemos tomar como referencia dos definiciones:

“En líneas generales podríamos decir que las nuevas tecnologías de la información y comunicación son las que giran en torno a tres medios básicos: la informática, la microelectrónica y las telecomunicaciones; pero giran, no sólo de forma aislada, sino lo que es más significativo de manera interactiva e interconexionadas, lo que permite conseguir nuevas realidades comunicativas”. (Cabero, 1998: 198)”.

“La tecnología educativa encuentra su papel como una especialización dentro del ámbito de la Didáctica y de otras ciencias aplicadas de la Educación, refiriéndose especialmente al diseño, desarrollo y aplicación de recursos en procesos educativos, no únicamente en los procesos instructivos, sino también en aspectos relacionados con la Educación Social y otros campos educativos. Estos recursos se refieren, en general, especialmente a los recursos de carácter informático, audiovisual, tecnológicos, del tratamiento de la información y los que facilitan la comunicación” (En A. Bautista y C. Alba, 1997:2).

Al tener en cuenta estas definiciones, la que más se adecúa a nuestra investigación es la segunda, ya que el enfoque es más en la educación y es una realidad decir que las tecnologías facilitan la accesibilidad a las personas con discapacidad visual permitiéndonos adaptar estos entornos a sus necesidades en especial el *software* libre.

1.3 SOFTWARE LIBRE

El *software* libre posee una serie de ventajas para las personas discapacitadas visualmente las cuales son resultado del enfoque altruista o la libertad que este tipo de *software* proporciona. Existen diferentes tipos de licencias libre, pero para este caso solo utilizaremos la definición que proporciona la **FSF** (*Free Software Foundation*, 2017).

En la **FSF** encontramos la definición de *software* libre como una serie de criterios que se deben cumplir para que un programa sea considerado libre.

El *software* libre es una expresión de libertad, donde los usuarios tiene la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el *software*, se especifica que se debe entender el *software* por el concepto de libre como de libre expresión y no como gratuidad.

De acuerdo con la cita anterior sobre la *FSF* las cuatro libertades esenciales que debe poseer un *software* para considerarse libre son las siguientes:

- La libertad de ejecutar el programa como se desea, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.
- La libertad de redistribuir copias para ayudar a su prójimo (libertad 2).
- La libertad de distribuir copias de sus versiones modificadas a terceros (libertad 3). Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello.

Si las libertades antes citadas se cumplen de manera adecuada el *software* sería considerado como libre.

Proyectos de *software* libre accesibles. En la comunidad del *software* libre hay una serie de proyectos que en conjunto proporcionan el entorno adecuado para las personas discapacitadas visualmente. Proyectos como *Debian*, *Ubuntu*, *Knoppix*, *Grml*, han incorporado ciertas utilidades en sus sistemas operativos como por el soporte braille y lectores de pantalla, a continuación, enunciaremos algunos *software* enfocados en la accesibilidad:

Proyecto Orca: Orca (*The GNOME project*, 2017) es un proyecto de código abierto que proporciona acceso al entorno gráfico a través de la voz y braille por medio de un lector de pantalla.

Proyecto BRLTTY: Es un demonio o proceso en segundo plano que proporciona acceso a la consola de los sistemas operativos tipo Unix para una persona con discapacidad visual usando una línea braille y se ha incorporado ciertas funcionalidades de voz. Al ser un servicio nos permite establecer su ejecución al inicio del sistema creando así un entorno de trabajo configurado de acuerdo con las necesidades.

Emacspeak: Es una interfaz de voz para entornos en modo texto, permitiendo una facilidad de uso a personas con discapacidad visual a la hora de utilizar entornos Tipo Unix (Dom Nov, 2002).

Speakup: Escrito originalmente por *Kirk Reiser* y *Andy Berdan*, es un paquete de revisión de pantalla para el sistema operativo Linux. *Speakup* le permite interactuar con las aplicaciones y el sistema operativo con retroalimentación audible desde la consola con un dispositivo de voz sintética (*Kirk Reiser & Andy Berdan*, 2015).

Sócrates: Es un sistema de escaneo que permite reconocer caracteres u objetos permitiendo el reconocimiento de imágenes escaneadas o fotografías obteniendo como resultado un documento en una serie de formatos editables o de fácil uso, también soporta lectura basada en texto. El proyecto es mantenido por el antes citado *Kirk Reiser* y *Jim Danley*.

NDVEA: Es un lector de pantalla libre que permite a las personas con discapacidad visual o totalmente ciegas usar una computadora. Posee la posibilidad de convertir texto a braille, siendo una herramienta de gran utilidad para la educación de personas que cumplan con esta discapacidad. A pesar de ser *software* libre solo funciona en sistemas operativos *Windows* (*NV Access*, 2003.).

Navegador web Firefox: Proyecto de *software* libre el cual es un navegador *web* multiplataforma que cumple con estándares de accesibilidad proporcionados por la W3C (*The Society of Motion Picture and Television Engineers*, 2013).

Chromevox: Este *software* es un complemento del navegador google Chrome de libre distribución y de libre uso, no cuenta con una licencia libre ni de código abierto, pero puede ser de gran utilidad en algunos sistemas operativos accesibles.

JClic: Conjunto de herramienta de *software* libre que permite la elaboración de contenido educativo (“zonaClic - JClic,” 2017).

Vinux: Sistema operativo *GNU/Linux* basado en *Ubuntu* diseñado para personas discapacitadas visualmente, consta de una serie de herramientas preconfiguradas para el uso adecuado en estos tipos de discapacidad, de fácil instalación.

Sonar: Sistema operativo *GNU/Linux* diseñado para personas discapacitadas y con un arsenal de herramientas que permiten el uso de este por personas discapacitadas (*Sonar GNU Linux*, 2017).

Lynx: Cliente *web* (web@lynx.browser.org, 2017) en modo texto útil para personas discapacitadas ya que puede ser usado con un lector de pantalla, posee algunos servicios avalados por la *W3C*.

1.4 SISTEMAS OPERATIVOS

Los sistemas operativos son una herramienta fundamental para el manejo de un sistema de cómputo, el uso de ciertos dispositivos necesita de un sistema operativo y de ciertas habilidades del usuario, siendo esto una gran limitante para las personas con algunos tipos de discapacidad, en nuestro caso discapacitados visuales.

Por esta razón los sistemas operativos libres serían el ecosistema adecuado para personalizar los entornos de trabajo de las personas discapacitadas ya que nos permiten la implementación de todas las posibles herramientas hechas para este fin y su licencia nos otorga privilegios para adaptarlas a necesidades muy específicas.

En el campo educativo de pregrado sería un aporte fundamental el uso de estos sistemas, por el motivo del libre acceso y costos, siendo el costo un impedimento fuerte en el entorno académico, ya que las licencias de algún *software* son demasiado costosas y más si estas licencias obligan el uso del *software* con algún hardware específico.

Como alternativa de sistemas operativos libres accesibles vamos a enunciar el sistema *Vinux* y *Sonar*, los cuales nos permitirán hacer la adecuada adaptación del entorno para los usuarios discapacitados visuales y el consumo del material pedagógico, aunque cabe resaltar que cualquier distribución tipo Unix puede ser adaptada para una persona discapacitada.

La siguiente tabla comparativa obtenida de la investigación doctoral realizada por Francisco de Assis da Costa (de Assis, 2010) y el libro Aspectos legales y de explotación del *software* libre (Bain, Gallego Rodríguez, Martínez Ribas, & Rius Sanjuán, 2004) hacen referencia a las restricciones y derechos entre sistemas *Windows* y tipo *Unix* con respecto a sus licencias.

Tabla 1. Resumen sobre derechos y restricciones en las licencias de *software* privado y *software* libre

Derechos y Restricciones	EULA (de Assis, F. 2010)	GNU/GPL (FSF, 2017)
Derechos Otorgados	<p>Puede usarse en un único ordenador con un máximo de 2 procesadores.</p> <p>La licencia sólo puede transferirse una vez a otro usuario. El licenciatarario debe destruir su copia.</p>	<p>Permite el libre uso, copia, modificación y redistribución del <i>software</i>.</p> <p>Pueden concederse sucesivas licencias y se puede cobrar por los servicios sobre el <i>software</i>.</p>
Restricciones Obligaciones	<p>Se prohíbe la copia, la modificación y la redistribución.</p> <p>No puede ser usado como servidor web o de archivos (<i>file server</i>).</p> <p>Impone una limitación sobre la ingeniería inversa (la única modificación posible).</p> <p>Registro necesario a los 30 días.</p>	<p>Las modificaciones y obras derivadas deben distribuirse bajo la misma licencia (<i>copyleft</i>).</p> <p>La distribución tiene que incluir el código fuente.</p>
Derechos Reservados	<p>Puede dejar de funcionar si se efectúan cambios en el <i>hardware</i>.</p> <p>Las actualizaciones del sistema pueden modificar la licencia, si el licenciante lo desea.</p>	<p>En principio, la <i>GPL</i> no ofrece ninguna garantía, aunque permite al licenciante que decida ofrecerla, o se vea obligado a ello por la ley que le sea aplicable.</p>

Tabla 1. (Continuación)

Derechos y Restricciones	EULA (de Assis, F. 2010)	GNU/GPL (FSF, 2017)
	<p>Se reserva el derecho para, en cualquier momento, recoger la información del sistema y su uso, y entregar dicha información a terceros.</p> <p>La garantía es por los primeros 90 días, y limitada al precio del <i>software</i>.</p> <p>Las actualizaciones y los parches no tienen garantía.</p>	

Fuente: Elaboración propia

Al observar la tabla comparativa entre estas licencias podemos comprender el motivo e interés en usar sistemas operativos tipo *Unix* ya que las libertades que otorgan al usuario promueven en gran parte la accesibilidad.

1.5 MODELO EDUCATIVO

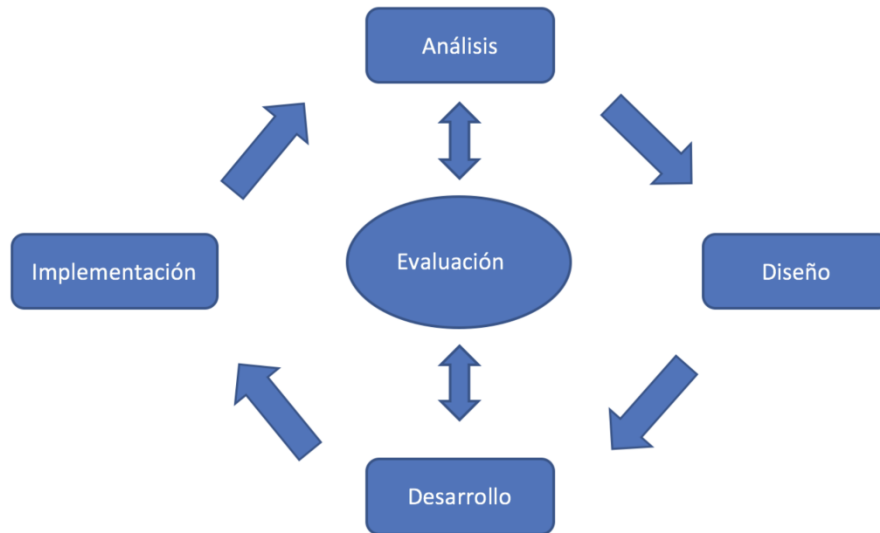
Un modelo educativo es la estructura de enfoques pedagógicos los cuales describen cómo se debe impartir el conocimiento de una forma adecuada, en pocas palabras los pasos que los docentes deben realizar para la elaboración de sus programas de estudio.

La finalidad de la investigación es tener un entorno o ecosistema adecuado para los usuarios discapacitados visualmente, con la intención de proporcionar una serie de materiales educativos de los cuales ellos puedan hacer uso y consumo. Para el desarrollo adecuado de un contenido educativo tomaremos como base el diseño instruccional descrito a continuación.

1.5.1 Diseño instruccional. El diseño instruccional es una guía o serie de pasos para los educadores, donde se especifica el proceso que se debe realizar para desarrollar los contenidos educativos, tomando como base teórica del aprendizaje, dando como resultado una enseñanza de calidad.

También se puede tomar como referencia la definición dada por *Broderick* (*Belloch*, 2013) donde expresa que el diseño instruccional es el arte y ciencia aplicada de crear un ambiente instruccional y los materiales, claros y efectivos, que ayudarán al alumno a desarrollar la capacidad para lograr ciertas tareas.

Ilustración 1. Diseño Instruccional



Fuente: <https://doi.org/978-987-24871-6-4>

En Ilustración 1 obtenida del texto Unidad de Tecnología Educativa (UTE). Universidad de Valencia (*Belloch*, 2013) se muestra el modelo *ADDIE* el cual se utilizó para la elaboración del material pedagógico y está compuesto por las siguientes fases:

- **Análisis:** El paso inicial es analizar el alumnado, el contenido y el entorno cuyo resultado será la descripción de una situación y sus necesidades formativas.
- **Diseño:** Se desarrolla un programa del curso deteniéndose especialmente en el enfoque pedagógico y en el modo de secuenciar y organizar el contenido.
- **Desarrollo:** La creación real (producción) de los contenidos y materiales de aprendizaje basados en la fase de diseño.

- **Implementación:** Ejecución y puesta en práctica de la acción formativa con la participación de los alumnos.
- **Evaluación:** Esta fase consiste en llevar a cabo la evaluación formativa de cada una de las etapas del proceso *ADDIE* y la evaluación sumativa a través de pruebas específicas para analizar los resultados de la acción formativa.

1.6 TRABAJO RELACIONADO

Antes de iniciar con el proceso de toma de decisiones que condujeron el desarrollo del presente trabajo, se analizaron proyectos relacionados que nos puedan aportar una visión más amplia del tema.

Las personas con discapacidad visual tienen grandes dificultades a la hora de desempeñarse en el campo laboral y académico, esto ocasiona cierto grado de discriminación, por suerte las tecnologías de la información han sido una gran herramienta para mejorar sus condiciones y calidad de vida. Algunas compañías desarrolladoras de tecnologías han creado *software* para satisfacer estas necesidades, pero estos desarrollos tienen precios poco accesibles para ciertos grupos sociales, en especial para las personas discapacitadas.

Un ejemplo de un buen desarrollo es la empresa Apple con su sistema operativo *Mac OS* el cual incorpora una serie de herramientas que proporcionan accesibilidad al sistema, pero con una fuerte limitante en cuanto a su licencia (*APPLE INC*, 2018), ya que están sujetas a unas cláusulas privadas que obligan al usuario a utilizar el *software* en conjunto con el *hardware*, limitando la diversidad de entornos en los cuales podemos utilizar estas aplicaciones accesibles.

Es aquí donde el *software* libre citado anteriormente demuestra una importancia significativa para las personas discapacitadas, debido a que este nos proporciona una serie de ventajas frente a alternativas privadas como lo son: su precio, comunidad, seguridad, estabilidad y eficiencia.

El *software* libre posee una serie de libertades las cuales dan una gran utilidad debido a su enfoque social. Se deben tener en cuenta los diferentes tipos de *software* libre con la intención de poder hacer un uso responsable de este, tanto en el campo académico como el laboral.

Desde 1982 (Organización de Naciones Unidas, 1994) ya se hablaba de garantizar de algún modo la accesibilidad en diferentes áreas como la académica y laboral en la formulación del programa de acción mundial para personas discapacitadas y unos años después con la Secretaría de la convención sobre los derechos humanos de las personas discapacitadas, donde se establecieron una

serie de normas para apoyar los derechos y la dignidad de las personas discapacitadas en general, sirviendo como apoyo judicial.

El DAES (“Naciones Unidas Departamento de Asuntos Económicos y Sociales: Crisis Financiera y Económica,” 2018) hace parte de la secretaría de la convención sobre los derechos humanos y tiene como objetivo:

- Apoyar la participación plena y efectiva de las personas con discapacidad en la vida social y el desarrollo.
- Promover los derechos y proteger la dignidad de las personas con discapacidad.
- Fomentar la dignidad de acceso al empleo, educación, información y a todos los bienes y servicios.

También la W3C (*The Society of Motion Picture and Television Engineers*, 2013) nos proporciona mecanismos para garantizar la accesibilidad, esta es una comunidad internacional que desarrolla estándares *web* y es liderada por el inventor de la *web* *Tim Berners – Lee* y el director ejecutivo de CEO. Estos estándares son de gran utilidad para el uso del contenido *web* por las personas discapacitadas en especial las discapacitadas visualmente.

En el campo laboral se usa una serie de normas las cuales se llaman criterios **DALCO** (Deambulaci3n, Aprehensi3n, Localizaci3n, Comunicaci3n) (P3rez, U. R, 2014) que se encargan de la accesibilidad en las distintas condiciones laborales, estos criterios se definen en las normas:

- UNE 170001-1:2007 Accesibilidad global. Criterios para facilitar la accesibilidad al entorno. Parte 1: requisitos DALCO.
- UNE 170001-2:2007 Accesibilidad global. Criterios para facilitar la accesibilidad al entorno. Parte 2: sistema de gesti3n de la accesibilidad global.
- UNE 41500 IN Accesibilidad en la edificaci3n y el urbanismo. Criterios generales de dise1o.
- UNE 41510 Accesibilidad en el urbanismo.
- UNE 41520 Accesibilidad en la edificaci3n. Espacios de comunicaci3n horizontal.
- UNE 41523 Accesibilidad en la edificaci3n. Espacios higi3nico-sanitarios.

Ya que nuestro enfoque es la discapacidad visual es de gran utilidad el criterio de Aprehensión que también integra el alcance visual.

Como se enuncia al comienzo del texto el *software* libre es una herramienta necesaria y adecuada para apoyar a la comunidad con discapacidad visual. Existe un *software* llamado *Vinux* (*Vinux Project* 2017, 2017), el cual es un sistema operativo basado en Ubuntu adecuado para las personas discapacitadas visuales con un gran repertorio de herramientas accesibles como, por ejemplo, *ORCA* y soporte braille. Siendo este sistema operativo excelente entorno tanto para la educación como para su campo laboral.

Otro de los desarrollos que se considera interesante es *SOLCA*(Autoridad nacional para la innovación personal Gobierno de Panamá, 2010) proyecto que nació como iniciativa para fomentar el uso del *software* de código abierto en el gobierno de panamá, proporcionando una serie de herramientas para diferentes tipos de discapacidad, por ejemplo un lector de documentos, un detector de movimiento facial para controlar el mouse, siendo una gran alternativa de *software* libre para sistemas operativos *Microsoft*.

Por otra parte, un proyecto similar es la tesis doctoral de Julián Esteban Gutiérrez Posada (Gutierrez, 2015) el cual desarrollo un dispositivo que se encarga de utilizar interfaces de usuario diferente para personas con discapacidades haciendo uso de herramientas como lectores *RFID* y tarjetas *NFC*.

Este trabajo pretende formalizar la investigación sobre el uso de *software* libre de autor para personas con discapacidad visual de tal forma que se pueda lograr determinar el conjunto de herramientas adecuado y que satisfaga las necesidades en un entorno académico de educación superior.

2. PROTOTIPO DE *SOFTWARE* LIBRE COMO APOYO PARA ESTUDIANTES CON DISCAPACIDAD

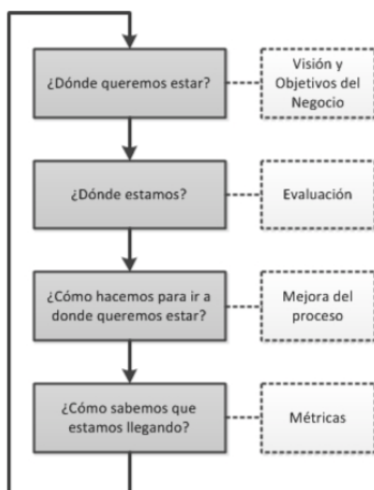
En este capítulo se detallarán los procesos realizados para la elaboración del prototipo de *software* libre como apoyo para estudiantes con discapacidad visual, como también las herramientas de *software* libre utilizadas y las adaptaciones necesarias para los diferentes tipos de estudiantes.

2.1 PROTOTIPO DE HERRAMIENTAS DE *SOFTWARE* LIBRE COMO APOYO PARA ESTUDIANTES DE EDUCACIÓN DE NIVEL SUPERIOR CON DISCAPACIDAD VISUAL

Esta primera etapa del proyecto corresponde a la definición del proceso realizado para la creación del prototipo de *software* libre que servirá de apoyo para estudiantes con discapacidad visual.

Debemos tener presente la definición de proceso (Bon, 2008), como una serie de actividades relacionadas lógicamente. Para empezar a detallar nuestro proceso de elaboración de un prototipo de *software* libre que sirva de apoyo para los estudiantes con discapacidad visual, debemos dar respuesta a las siguientes interrogantes expuestas en el modelo de mejora de Bon antes citado que se muestra en la ilustración 2.

Ilustración 2. Modelo de mejora de proceso



Fuente: Bon, J. V. (2008). Fundamentos de Gestión de Servicios TI basados en ITIL. itSMF Internacional.

A continuación, en la tabla 2 daremos respuesta a cada una de las interrogantes del modelo de mejora mostrado en la ilustración 3.

Tabla 2. Respuestas modelo Bon

¿Dónde queremos estar?	Realizando una adecuada integración del prototipo de <i>software</i> libre para estudiantes discapacitados visuales que aria parte importante de las herramientas educativas de la institución
¿Dónde estamos?	Normal mente en la institución universitaria actual no se poseen herramientas de <i>software</i> libre que permitan acceder adecuadamente a los estudiantes con discapacidad visual a sus materiales educativos, dejando la responsabilidad a cada estudiante
¿Cómo hacemos para ir dónde queremos?	Definimos procesos claros que ayuden a la gestión del entorno y la creación de los contenidos educativos
¿Cómo sabemos que hemos llegado?	Estableciendo métricas en el nivel de satisfacción de los usuarios y el fácil acceso a los entornos.

Fuente: Elaboración propia

Este proyecto es un prototipo o ecosistema para el consumo y creación de materiales educativos para estudiantes con discapacidad visual, por medio de herramientas de autor y configuraciones personalizadas, este ecosistema de herramientas libres se configuro sobre un sistema operativo de licencia libre.

Con este prototipo se facilita el consumo de los contenidos educativos por parte de los alumnos. Para determinar el impacto del proyecto se evaluó la accesibilidad utilizando la población objetivo que son los discapacitados visuales de la institución universitaria EAM, para la evaluación utilizaron dos herramientas, una encuesta verbalizada y la segunda una encuesta escrita, las cuales permiten a los usuarios expresarse de dos formas distintas ante las sensaciones que produce la utilización el prototipo.

Para el logro de los objetivos propuestos, se han realizado las siguientes actividades:

- **Fase 1:** Análisis bibliográfico: Se realizó un análisis detallado de la documentación relacionada y a las diversas áreas del conocimiento que interviene en este estudio.

- **Fase 2:** Exploración: Levantamiento de los requerimientos mínimos que los estudiantes (usuarios) necesitan para el uso del ecosistema.
- **Fase 3:** Instalación y configuración: Se instalaron y configuraron las herramientas de autor.
- **Fase 4:** Elaboración del material educativo: Se realizó el material educativo siguiendo un modelo instruccional.
- **Fase 5:** Pruebas a usuarios finales (PO): Se determinó la calidad del material educativo y la usabilidad de las herramientas de autor libres, aplicando instrumentos tales como encuestas de acuerdo con la muestra de estudio (Población objetivo).
- **Fase 6:** Análisis estadístico: resultados de acuerdo con los datos obtenidos en las encuestas de *think aloud* (Boren, M. T., & Ramey, J, 2000) y la encuesta escrita.

2.2 ANÁLISIS BIBLIOGRÁFICO

Se realizaron las investigaciones de acuerdo con las clasificaciones expuestas en la siguiente tabla.

Tabla 3. Metodología estado del arte

PROCEDIMIENTO METODOLÓGICO	
Contextualización	<p>Ecosistema de herramientas de <i>software</i> libre adaptado para el consumo de materiales educativos por parte de estudiantes discapacitados visualmente.</p> <p>El prototipo debe: Permitir adaptar el conjunto de <i>software</i> necesario para las diferentes necesidades de los usuarios y la realización del material educativo por parte de los docentes</p>
Clasificación	<p>Palabras claves: Sistemas Operativos, <i>Software</i> libre, Discapacidad visual, Accesibilidad, <i>Software</i> Educativo.</p>

Fuente: Elaboración propia

Los proyectos libres por resaltar que se enfocan en la accesibilidad para discapacitados visuales y que nos permitieron la realización del proyecto son *Vinux*, *Sonar* y *JClic*. En el apartado 5.1.3 instalación y configuración se detallará el motivo de la selección de estos proyectos como herramientas fundamentales este proyecto.

2.3 EXPLORACIÓN LEVANTAMIENTO DE REQUERIMIENTOS

Esta actividad consiste en la identificación de las necesidades de los estudiantes con discapacidades visuales para generar la selección y elaboración del prototipo de herramientas de autor libre.

En esta fase realizamos reuniones con cada uno de los estudiantes que poseen la discapacidad visual, para realizar la recolección de los requerimientos necesarios, con la intención de determinar qué tipo necesidades poseen los alumnos

Separamos los estudiantes en 3 niveles de discapacidad los cuales son:

- Nivel 1: Estudiantes que poseen poco grado de discapacidad visual ya que pueden realizar actividades en el sistema solo con un poco de esfuerzo.
- Nivel 2: Estudiantes que necesitan obligatoriamente el uso de gafas o lentes para realizar cualquier tipo de actividad en el sistema.
- Nivel 3: Estudiantes que poseen una ceguera total.

Los estudiantes de nivel 3 de discapacidad deberán hacer uso de un módulo *NFC* que permitirá acceder a los contenidos académicos sin la necesidad de interactuar directamente con el sistema operativo o las herramientas de autor

2.4 INSTALACIÓN Y CONFIGURACIÓN

Preparación de las estaciones de trabajo: Esta actividad consiste en un procedimiento técnico que se enfoca en la preparación adecuada de las estaciones de trabajo las cuales poseerán las herramientas de autor libres para realizar el adecuado uso y consumo de materiales educativos por parte de los alumnos. Adicional a esto se debe realizar con los usuarios una capacitación para el uso básico de las herramientas

Tabla 4. Equipos de cómputo utilizados

Marca	Especificaciones	Sistema operativo
<i>Lenovo</i>	Procesador I5. 8 de memoria RAM 1T de almacenamiento	<i>Vinux</i>
<i>Hp</i>	i3	<i>Ubuntu 18</i>
<i>Macbook</i>	Procesador i7 16 de memoria RAM 500G de almacenamiento	<i>Sonar</i>
<i>Raspberry Pi</i>	Procesador Broadcom 1G de memoria RAM 64G de almacenamiento	<i>Raspbian</i>

Fuente: Elaboración propia

Para la instalación y configuración de los entornos hemos utilizado únicamente *software* libre ya que este nos permite hacer adaptaciones para los diferentes tipos de discapacidades visuales. Adicional en la instalación del prototipo utilizamos computadoras personales con las características expresadas en la tabla 4:

En la tabla 5 se detallarán las herramientas seleccionadas para este proyecto y su motivo.

De acuerdo a los sistemas operativos consultados seleccionamos solo los 4 sistemas expresados en la tabla 5 ya que las distribuciones *Ubuntu 18*, *Vinux*, *Sonar*, poseen el conjunto de herramientas de accesibilidad que nos sirvieron de base para la herramienta de autor *JClic* que permitió realizar los materiales educativos y el sistema operativo *Raspbian* fue incluido ya que en conjunto con su hardware llamado *Raspberry Pi*, nos facilita el ambiente necesario para realizar las configuraciones del lector NFC que automatiza el proceso de acceso a los materiales educativos accesibles.

De acuerdo con los niveles expresados en las fases anteriores los entornos de configuración quedarían de la siguiente manera. En la tabla 6 encontramos la configuración del entorno para los estudiantes de nivel 1, en la tabla 7 se encuentran las configuraciones de los estudiantes de nivel 2 y en la tabla 8 las configuraciones de los estudiantes de nivel 3.

Cabe aclarar que el entorno *Ubuntu 18* posee instaladas por defecto aplicaciones referentes a la accesibilidad como el lector de pantalla *ORCA* y menús de configuración que, para modificar el tamaño del texto, el contraste de colores, uso de lupas y tamaño del cursor.

Tabla 5. Selección herramientas de *software*

Nombre	Descripción	Motivo de selección
<i>Vinux</i>	Sistema Operativo adaptado para personas discapacitadas visuales	Experiencia en el uso de sistemas <i>GNU/Linux</i> , <i>software</i> libre, amplia comunidad, posee gran parte de las herramientas para discapacitados visuales ya configuradas.
<i>Sonar</i>	Sistema Operativo adaptado para personas discapacitadas	Experiencia en el uso de sistemas <i>GNU/Linux</i> , <i>software</i> libre, amplia comunidad, posee gran parte de las herramientas para discapacitados visuales ya configuradas.
<i>Raspbian</i>	Sistema Operativo diseñado para dispositivos <i>Raspberry</i>	Experiencia en el uso de sistemas <i>GNU/Linux</i> , <i>software</i> libre, amplia comunidad, soporte para módulos externos de <i>Arduino</i> .
<i>Ubuntu 18</i>	Sistema Operativo diseñado para usuarios finales	Experiencia en el uso de sistemas <i>GNU/Linux</i> , <i>software</i> libre, amplia comunidad, ambiente adecuado para personalizaciones
<i>Lynx</i>	Navegador web en modo texto	Proporciona navegación en modo texto y permite integración con lectores de pantalla
ORCA	Lector de pantalla	integración en el entorno <i>Gnome</i>
<i>JClic</i>	Herramienta de autor libre, para la creación de materiales educativos accesibles	Herramienta seleccionada para permitir la elaboración de materiales educativos accesibles, compartirlos en red y exportarlos en formato web

Fuente: Elaboración propia

Tabla 6. Entorno para estudiantes de discapacidad nivel 1

Nombre	Descripción
<i>Ubuntu 18</i>	Sistema operativo tipo <i>Unix</i>
<i>JClic</i>	Herramienta de autor libre para la elaboración y consumo de materiales educativos

Fuente: Elaboración propia

Tabla 7. Entornos para estudiantes con discapacidades nivel 2

Nombre	Descripción
<i>Ubuntu 18</i>	Sistema Operativo diseñado para usuarios finales
<i>Lynx</i>	Navegador web en modo texto
<i>ORCA</i>	Lector de pantalla
<i>Jclic</i>	Herramienta de autor libre, para la creación de materiales educativos accesibles

Fuente: Elaboración propia

Para los alumnos de nivel 2 utilizamos la distribución Ubuntu 18 ya que es la versión más actual y con entornos más agradables para el usuario. En este realizamos una configuración más personalizada del *software* ORCA con el comando que se observa en la ilustración 3.

Ilustración 3. Comando configuración lector de pantalla ORCA

```
→ ~ sudo orca -s
```

Fuente: Elaboración propia

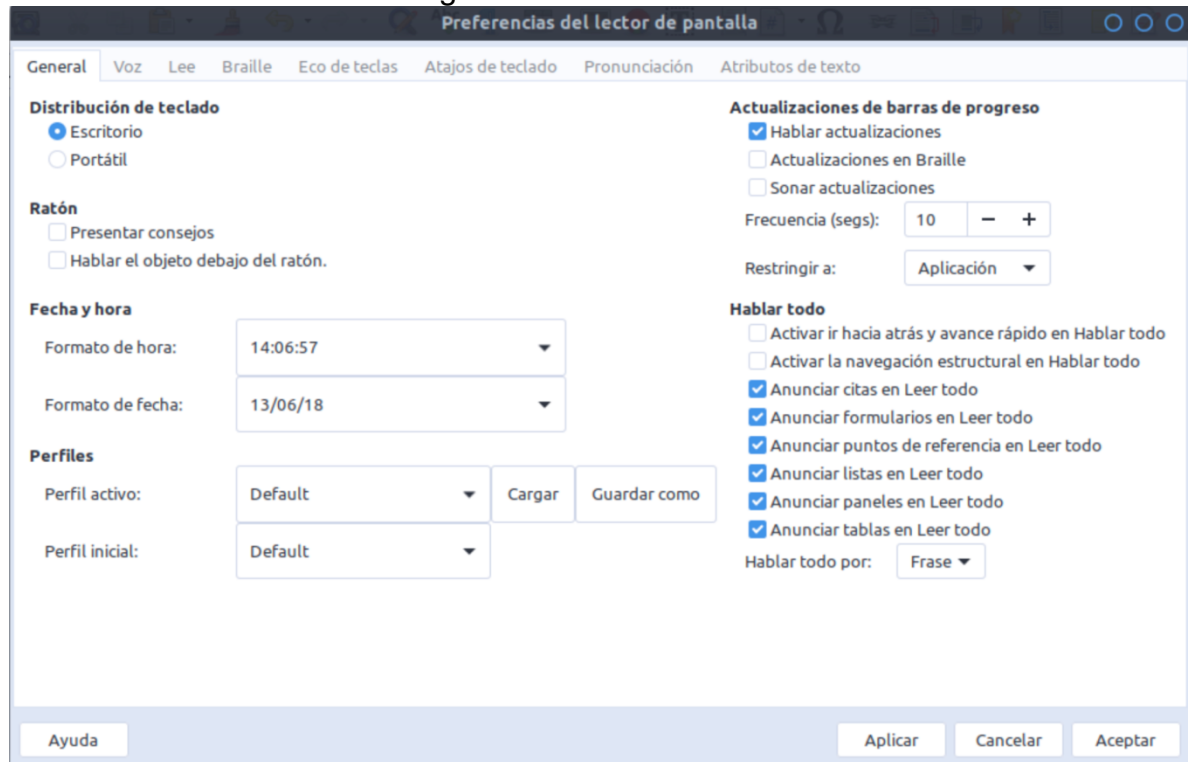
Este comando nos proporciona una ventana de configuración que utilizamos para adaptar el lector *ORCA* a las necesidades de nuestros alumnos como se muestra en la ilustración 4.

Para los usuarios de nivel 3 se construyó un módulo lector de tarjetas NFC que permitirá el acceso asistido a los materiales educativos. Para la construcción de este módulo se proporciona una guía más detallada en el ANEXO C.

El modulo lector NFC podrá ser utilizado en el sistema operativo *Raspbian* ya que este se instala sobre el hardware de una *Raspberry Pi* que permite las condiciones

necesarias para adaptar el lector y hacer uso de este. Sin embargo, el modulo lector también puede ser adaptado por medio de una conexión USB con las configuraciones de los niveles 1 y 2 ya que los sistemas operativos son tipo Unix y permiten la configuración de estas herramientas.

Ilustración 4. Ventana de configuración ORCA



Fuente: Elaboración propia

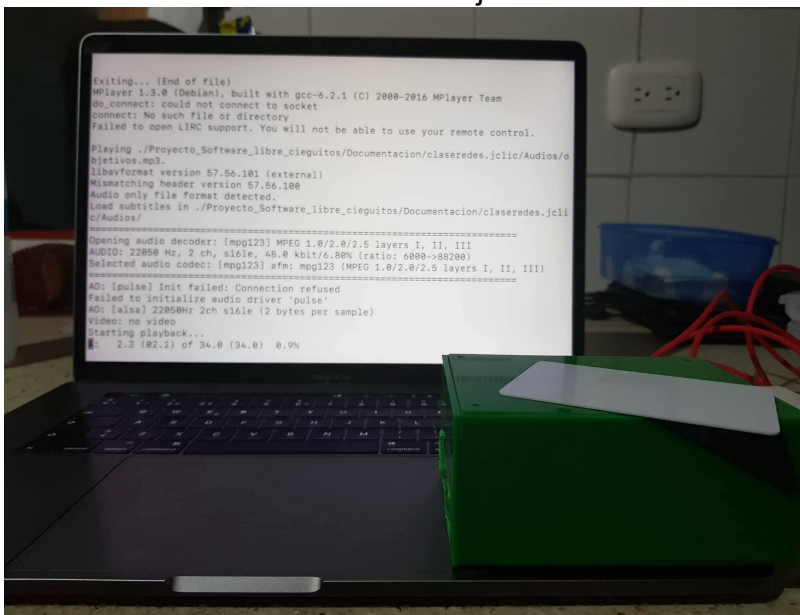
Tabla 8. Entorno para estudiantes con nivel 3 de discapacidad visual.

Nombre	Descripción
Vinux	Sistema Operativo adaptado para personas discapacitadas visuales
Sonar	Sistema Operativo adaptado para personas discapacitadas
Raspbian	Sistema Operativo diseñado para dispositivos <i>raspberry</i>
Lynx	Navegador web en modo texto
ORCA	Lector de pantalla
Jcllic	Herramienta de autor libre, para la creación de materiales educativos accesibles

Fuente: Elaboración propia

En la ilustración 5 se puede observar el módulo NFC de color verde sobre el cual se posan las tarjetas NFC para que efectúen las acciones automáticas.

Ilustración 5. Modulo lector de tarjetas NFC



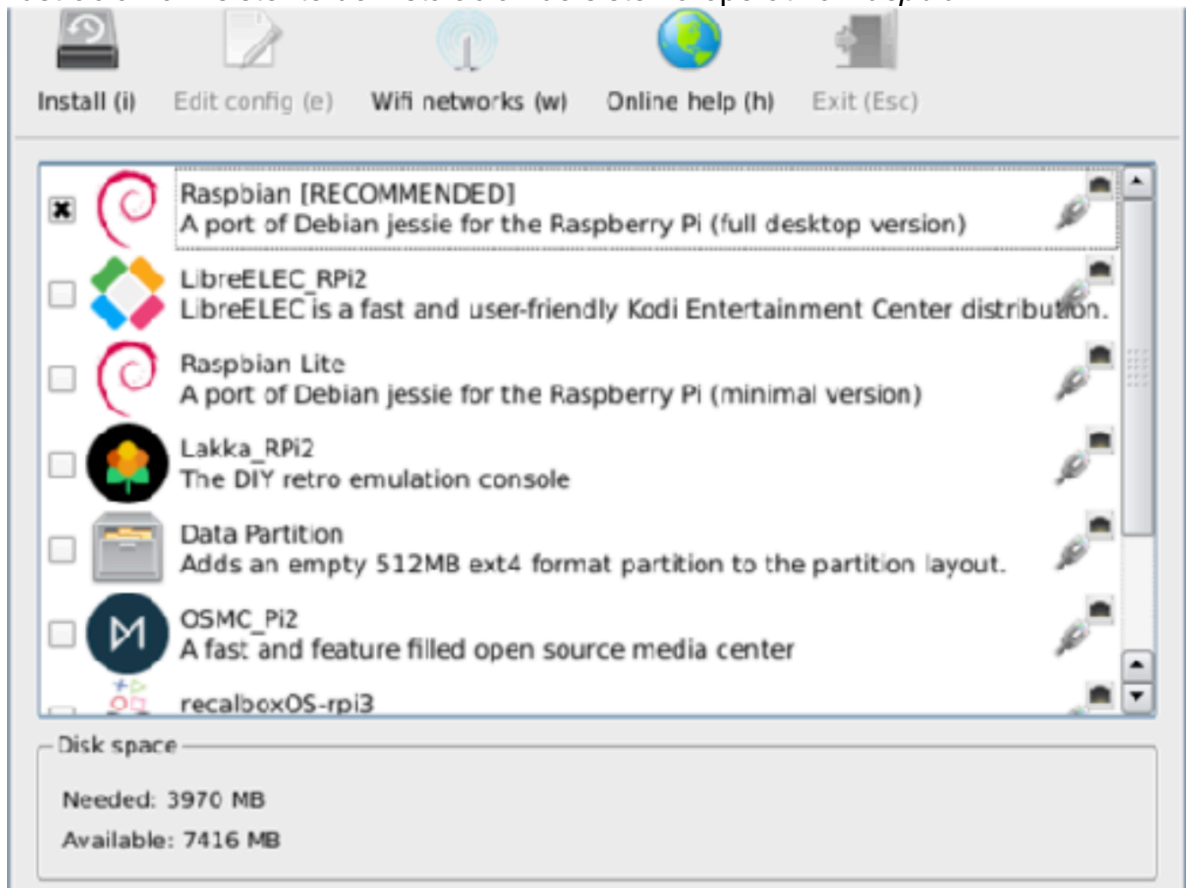
Fuente: Elaboración propia

Para la construcción del Módulo lector NFC se utilizaron los siguientes componentes.

- *Raspberry pi 3.*
- Memoria *SD* mínimo 8G.
- Cable puente hembra a hembra.
- Sensor de tarjetas NFC.

Para que el sensor funcione se instaló el sistema operativo *Raspbian* en nuestra *raspberry pi* tal como se ve en la ilustración 6.

Ilustración 6. Asistente de instalación de sistema operativo *Raspbian*



Fuente: Elaboración propia

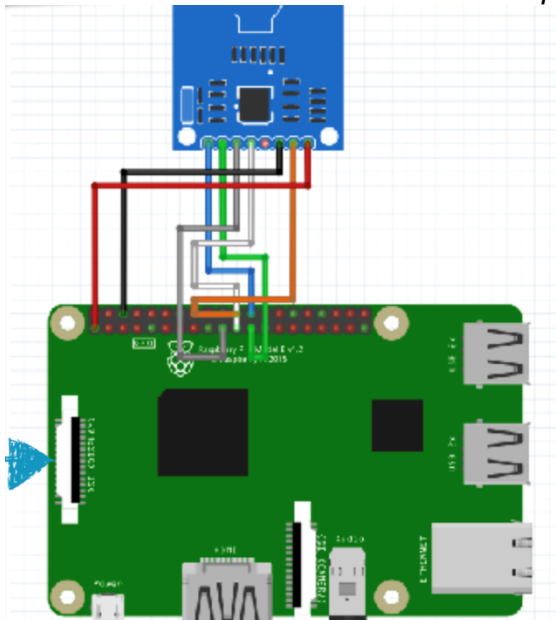
Proseguimos con la adición del sensor a los pines que se ven en la ilustración 7 de la *Raspberry*.

Descargamos los repositorios que contienen los controladores y el *script* desarrollado en el lenguaje de programación *Python* que activará el sensor y permitirá capturar las tarjetas *NFC*.

Los archivos necesarios se descargan con las instrucciones que se observan en las ilustraciones 8 y 9 en la terminal de *Linux*.

Ahora modificamos el *script* que se observa en la ilustración 10 y agregamos el código que se observa en la ilustración 11 para activar el contenido educativo al detectar el id de la tarjeta *NFC*.

Ilustración 7. Sensor conectado a *Raspberry*.



Fuente: Elaboración propia

Ilustración 8. Controladores del sensor.

```
→ ~ sudo git clone https://github.com/lthiery/SPI-Py.git
Cloning into 'SPI-Py'...
remote: Counting objects: 85, done.
remote: Total 85 (delta 0), reused 0 (delta 0), pack-reused 85
Unpacking objects: 100% (85/85), done. MFRC522-pyth
```

Fuente: Elaboración propia

Ilustración 9. Script para activar el sensor.

```
sudo: gatt: command not found
→ MFRC522-python-master git clone https://github.com/mxgwxw/MFRC522-python.git
Cloning into 'MFRC522-python'...
remote: Counting objects: 109, done.
remote: Total 109 (delta 0), reused 0 (delta 0), pack-reused 109
Receiving objects: 100% (109/109), 36.08 KiB | 9.02 MiB/s, done.
Resolving deltas: 100% (47/47), done.
warning: unable to access '/Users/mayler/.config/git/attributes': Permission denied
```

Fuente: Elaboración propia

Ilustración 10. Contenido del script.

```
#!/usr/bin/env python
# -*- coding: utf8 -*-
import RPi.GPIO as GPIO
import MFRC522
import signal

continue_reading = True

# Capture SIGINT for cleanup when the script is aborted
def end_read(signal,frame):
    global continue_reading
    print "Ctrl+C captured, ending read."
    continue_reading = False
    GPIO.cleanup()

# Hook the SIGINT
signal.signal(signal.SIGINT, end_read)

# Create an object of the class MFRC522
MIFAREReader = MFRC522.MFRC522()

# Welcome message
print "Welcome to the MFRC522 data read example"
print "Press Ctrl-C to stop."

# This loop keeps checking for chips. If one is near it will get the UID and authenticate
while continue_reading:
    # Scan for cards
    (status,TagType) = MIFAREReader.MFRC522_Request(MIFAREReader.PICC_REQIDL)

    # If a card is found
    if status == MIFAREReader.MI_OK:
        print "Card detected"

    # Get the UID of the card
    (status,uid) = MIFAREReader.MFRC522_Anticoll()

    # If we have the UID, continue
    if status == MIFAREReader.MI_OK:

        # Print UID
        print "Card read UID: %s,%s,%s,%s" % (uid[0], uid[1], uid[2], uid[3])

        # This is the default key for authentication
        key = [0xFF,0xFF,0xFF,0xFF,0xFF,0xFF]

        # Select the scanned tag
        MIFAREReader.MFRC522_SelectTag(uid)

        # Authenticate
        status = MIFAREReader.MFRC522_Auth(MIFAREReader.PICC_AUTHENT1A, 8, key, uid)

        # Check if authenticated
        if status == MIFAREReader.MI_OK:
            MIFAREReader.MFRC522_Read(8)
            MIFAREReader.MFRC522_StopCrypto1()
        else:
            print "Authentication error"
```

Fuente: Elaboración propia

Ilustración 11. Código para activar el material educativo

```
import MFRC522
import signal
import smtplib
import commands
import subprocess
continue_reading = True

# Capture SIGINT for cleanup when the script is aborted
def end_read(signal, frame):
    global continue_reading
    print "Ctrl+C captured, ending read."
    continue_reading = False
    GPIO.cleanup()

# Hook the SIGINT
signal.signal(signal.SIGINT, end_read)

# Create an object of the class MFRC522
MIFAREReader = MFRC522.MFRC522()

# Welcome message
print "Welcome to the MFRC522 data read example"
print "Press Ctrl-C to stop."

# This loop keeps checking for chips. If one is near it will get the UID and authenticate
while continue_reading:

    # Scan for cards
    (status, TagType) = MIFAREReader.MFRC522_Request(MIFAREReader.PICC_REQIDL)

    # If a card is found
    if status == MIFAREReader.MI_OK:
        print "Card detected"

    # Get the UID of the card
    (status, uid) = MIFAREReader.MFRC522_Anticoll()

    # If we have the UID, continue
    if status == MIFAREReader.MI_OK:

        # Print UID
        print "Card read UID: %s,%s,%s,%s" % (uid[0], uid[1], uid[2], uid[3])
        print uid

        if uid == [176, 24, 31, 57, 142]:
            print "hola mundo"
            subprocess.call("mp3layer ./Proyecto_Software_libre_cieguitos/Documentacion/claseredes.jclic/Audios/introduccion.mp3".split())
            subprocess.call("mp3layer ./Proyecto_Software_libre_cieguitos/Documentacion/claseredes.jclic/Audios/objetivos.mp3".split())
            subprocess.call("mp3layer ./Proyecto_Software_libre_cieguitos/Documentacion/claseredes.jclic/Audios/Terminologia/Terminologia_Voz/terminolog-a-10.mp3".split())
            subprocess.call("mp3layer ./Proyecto_Software_libre_cieguitos/Documentacion/claseredes.jclic/Audios/Terminologia/Terminologia_Voz/terminolog-a-2.mp3".split())
            subprocess.call("mp3layer ./Proyecto_Software_libre_cieguitos/Documentacion/claseredes.jclic/Audios/Terminologia/Terminologia_Voz/terminolog-a-3.mp3".split())
            subprocess.call("mp3layer ./Proyecto_Software_libre_cieguitos/Documentacion/claseredes.jclic/Audios/Terminologia/Terminologia_Voz/terminolog-a-4.mp3".split())
            subprocess.call("mp3layer ./Proyecto_Software_libre_cieguitos/Documentacion/claseredes.jclic/Audios/Terminologia/Terminologia_Voz/terminolog-a-5.mp3".split())
            subprocess.call("mp3layer ./Proyecto_Software_libre_cieguitos/Documentacion/claseredes.jclic/Audios/Terminologia/Terminologia_Voz/terminolog-a-6.mp3".split())
            subprocess.call("mp3layer ./Proyecto_Software_libre_cieguitos/Documentacion/claseredes.jclic/Audios/Terminologia/Terminologia_Voz/terminolog-a-7.mp3".split())
            subprocess.call("mp3layer ./Proyecto_Software_libre_cieguitos/Documentacion/claseredes.jclic/Audios/Terminologia/Terminologia_Voz/terminolog-a-8.mp3".split())
            subprocess.call("mp3layer ./Proyecto_Software_libre_cieguitos/Documentacion/claseredes.jclic/Audios/Terminologia/Terminologia_Voz/terminolog-a-9.mp3".split())
            subprocess.call("mp3layer ./Proyecto_Software_libre_cieguitos/Documentacion/claseredes.jclic/Audios/Terminologia/Terminologia_Voz/terminolog-a-1.mp3".split())

        # This is the default key for authentication
        key = [0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF]

        # Select the scanned tag
        MIFAREReader.MFRC522_SelectTag(uid)

        # Authenticate
        status = MIFAREReader.MFRC522_Auth(MIFAREReader.PICC_AUTHENT1A, 8, key, uid)

        # Check if authenticated
        if status == MIFAREReader.MI_OK:
            MIFAREReader.MFRC522_Read(8)
            MIFAREReader.MFRC522_StopCrypto1()
        else:
            print "Authentication error"
```

Fuente: Elaboración propia

2.5 INSTALACIÓN DE LA HERRAMIENTA DE AUTOR JCLIC

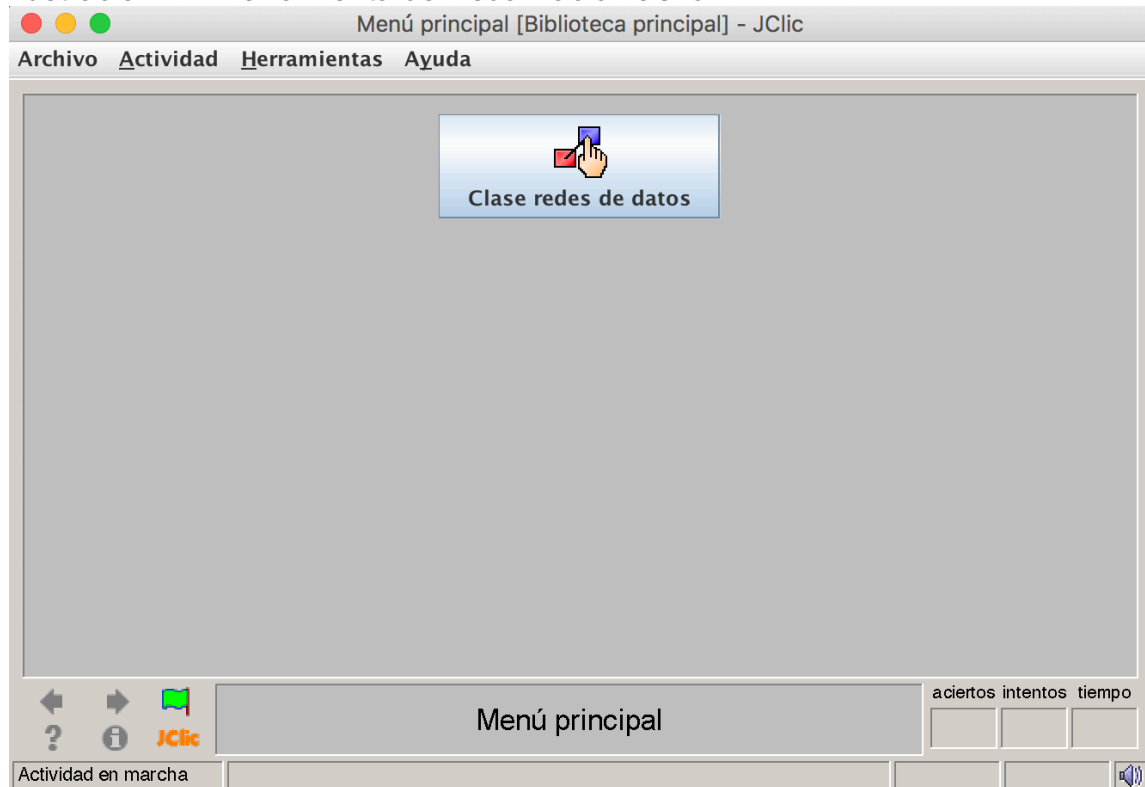
Este apartado indicara el proceso de instalación de la herramienta *JClíc* que permitió realizar los materiales educativos accesibles.

Como punto inicial se descargó de la siguiente dirección *web* <http://clíc.xtec.cat/es/jclíc/download.htm> 3 archivos. El primer archivo se llama *JClíc*

y se puede observar en la ilustración 12, el segundo posee el nombre de *JClic author* que se muestra en la ilustración 13, por último, la aplicación de reportes posee el nombre de *JClic reports* mostrado en la ilustración 14.

JClic: Es la herramienta para visualizar y ejecutar las actividades.

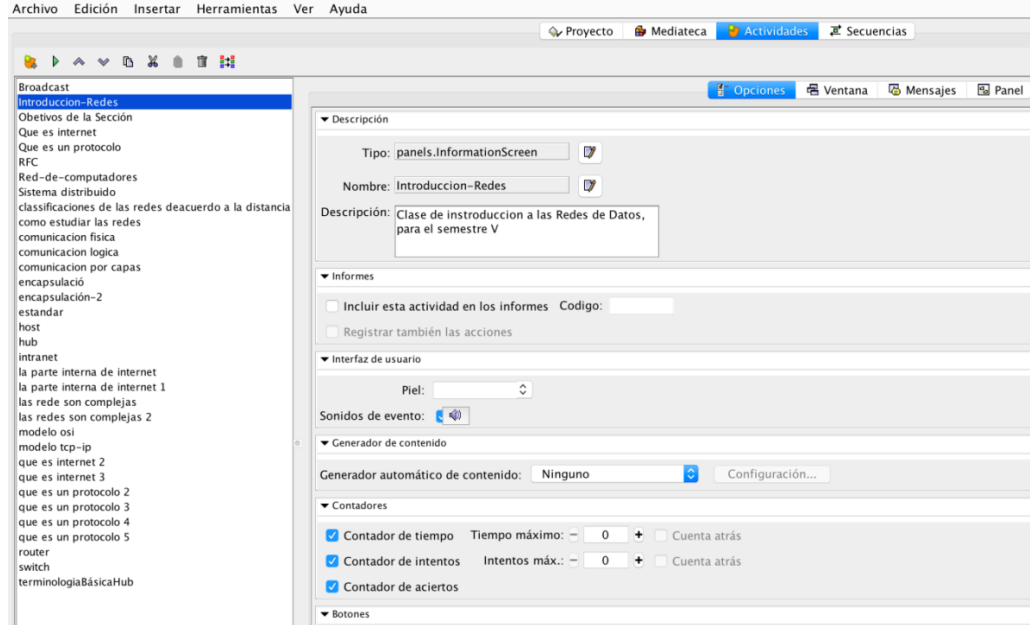
Ilustración 12. Herramienta de visualización *JClic*



Fuente: Elaboración propia

JClic author: Esta herramienta crea y modifica las actividades, adicional a esto exportar en diferentes formatos el contenido ya sea en *web* o en instaladores que posteriormente se ejecutaran con la herramienta *JClic* básica.

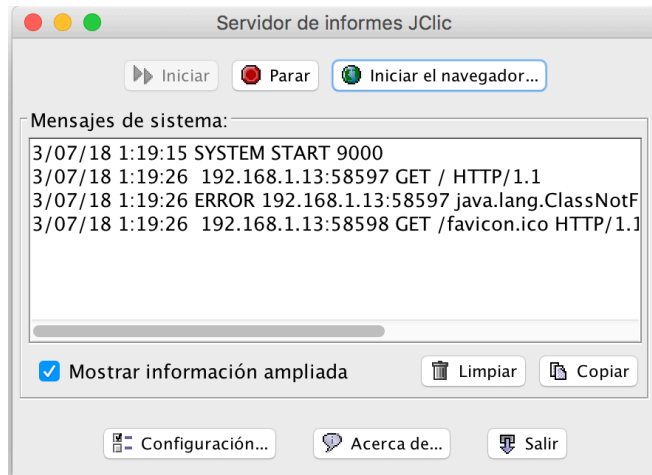
Ilustración 13. Herramienta de creación de materiales educativos *JClic*



Fuente: Elaboración propia

JClic reports: Esta herramienta genera un base de datos en la que se recopilan los resultados de las actividades y se conecta en red con los quipos que posean la herramienta *JClic* en el mismo segmento de red.

Ilustración 14. Herramienta de informes *JClic*



Fuente: Elaboración propia

De acuerdo con las herramientas seleccionadas para la instalación el prototipo se debe cumplir con las siguientes funcionalidades:

- Herramienta libre para la elaboración de material educativo: los usuarios ecosistema pueden ser estudiantes o docentes, la herramienta *jcllic* permite crear el contenido educativo accesible y efectuar el consumo de este. Esta herramienta de autor libre permitirá proporcionar los materiales en formato *web* o un instalador propio de la aplicación.
- lector de pantalla: el ecosistema de *software* dependiendo de la versión de sistema operativo tipo *Unix* que se desee usar traerá por defecto el lector de pantalla *ORCA* o se configurara manualmente.
- lupas e incrementadores de texto: El ecosistema de *software* permitirá hacer uso de lupas para mejor visualización de los iconos y del texto.
- Contrastes de colores: Permitirá realizar cambios de contraste entre los colores para mejorar la visualización de los objetos en pantalla.
- modulo lector de tarjetas *NFC*: Este módulo es un programa desarrollado en *Python* que permita capturar los códigos de tarjetas *NFC* y ejecutar automáticamente el material educativo para discapacitados visuales que necesiten un grado elevado de asistencia.
- navegadores en modo texto: Permite navegar en la web por medio de la consola de comando.
- Soporte braille: el ecosistema soportara dispositivos braille.

2.6 ELABORACIÓN DE LOS MATERIALES EDUCATIVOS

Para la elaboración de los materiales educativos, se utilizó una metodología instruccional, la cual tiene 5 fases.

Fase 1- Análisis. En esta fase analizamos nuestra población objetivo en nuestro caso, los estudiantes con discapacidad visual del programa de ingeniería de *software* de la institución universitaria EAM. Por consiguiente, identificamos cuáles son sus necesidades formativas.

Fase 2 – Diseño. En esta fase elaboramos el contenido del material educativo de una forma se secuencial. Para esta oportunidad diseñamos el material de una clase de Redes de Datos del semestre 5 de ingeniería de *software*.

Los objetivos de esta clase son los siguientes:

- Describir el funcionamiento general de una red de computadores.

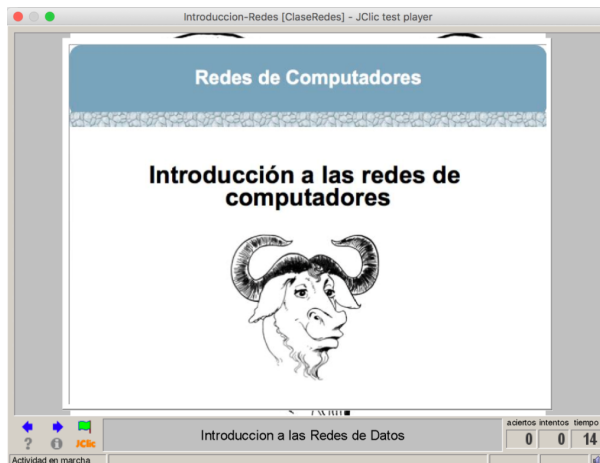
- Entender la necesidad de dividir por capas la problemática de la transmisión de datos.
- Conocer la terminología básica.
- Conocer los medios de transmisión de datos más comunes comúnmente actualmente.

A continuación, observaremos el contenido de la clase introducción a las redes:

- Terminología básica.
- Modelo por capas.
- Medios físicos de transmisión de datos.
- Acceso a internet

Fase 3 – Desarrollo. Elaboramos el material en la herramienta de *software* libre que seleccionamos la cual fue *JClic*, en la ilustración 15 podemos observar cómo se vería la clase en ejecución.

Ilustración 15. Material educativo *JClic*



Fuente: Elaboración propia

Fase 4 – Implementación. Realizamos la puesta en marcha de los materiales educativos.

Fase 5 – Evaluación. Evaluamos los materiales educativos, tanto su calidad como el resultado de los estudiantes.

2.7 PRUEBAS A USUARIOS FINALES

Para este apartado se utilizaron dos tipos de prueba:

Para lograr el objetivo de estudio se utilizaron dos herramientas, la técnica de *Thinking Aloud* y una encuesta realizada en base a la tesis de maestría de Francisco Inesto (Carrasco. F. I, 2008) en la cual enuncia una serie de conceptos fundamentales a tener en cuenta.

Equivalente: Se proporciona a todos los estudiantes la misma actividad educativa, pero por diferentes medios.

Alternativo: Se proporcionan distintas actividades educativas pero que cumplan con el mismo objetivo. Habilitar entornos en los cuales se consiga el mismo objetivo educacional, aunque de manera distinta.

Directo: Hay posibilidad de que los alumnos puedan acceder a los contenidos independientemente de que tengan o no una discapacidad.

Compatible: Los contenidos de una actividad educativa podrán ser accedidos y manejados mediante el uso de una ayuda técnica a la actividad diseñada teniendo en cuenta dichas ayudas

También las pautas globales de accesibilidad a herramientas de autor.

Principios generales:

- El *software* debe minimizar el número de acciones del usuario,
- Permitir que el usuario elija dispositivos de entrada y salida y poder ser utilizado sólo con uno de ellos.
- Se deben usar los servicios de accesibilidad del sistema operativo.
- Es recomendable incorporar la función de deshacer, facilitar la definición de preferencias de usuario y utilizar los servicios de entrada / salida estándar del entorno.

Teclado:

- El usuario debe poder utilizar el programa sólo con teclado.

- La navegación por teclado no debe activar objetos de la interfaz y se deben ofrecer alternativas a la pulsación simultánea de varias teclas.
- Se deben ofrecer atajos de teclado para las funciones más importantes y permitir que el usuario modifique la asignación de funciones a las teclas.
- Hay que ofrecer una secuencia de navegación por teclado consistente con la distribución de elementos en pantalla y respetar las convenciones de uso del teclado del sistema operativo.

Descriptivos apuntadores:

- Deben permitir cambiar la asignación de funciones de los botones del ratón y hacer doble clic y el arrastre mediante pulsaciones simples de teclas o botones.

Pantalla:

- No usar el color como única fuente de información.
- Utilizar las funciones estándar del entorno para mostrar texto.
- Permitir que el usuario cambie aspectos visuales y evitar frecuencias de refresco que puedan provocar ataques epilépticos.

Sonidos y multimedia:

- Deben ofrecerse alternativas a las salidas de sonido y permitir que el usuario modifique parámetros del sonido, principalmente su volumen

Notificaciones al Usuario:

- Los mensajes emitidos al usuario deben ser redactados para facilitar su identificación y comprensión por personas sin conocimientos técnicos

Información de objetivos:

- Deben proporcionar información textual asociada a todos los objetos de su interfaz, para que tanto las ayudas técnicas como los revisores de pantalla puedan presentársela al usuario de la forma más adecuada

Tiempo:

- Se debe ofrecer al usuario el control sobre aspectos dependientes del tiempo, como contenidos dinámicos o mensajes con tiempo límite de respuesta.

Documentación:

- Debe ser clara, sencilla y proporcionarse en formatos alternativos bajo petición del usuario, debe ofrecerse ayuda en pantalla.

Otros requisitos:

- La aplicación debe dar la opción de salir.

La técnica *Thinking Aloud* (pensado en voz alta) (Boren, M. T., &Ramey, J, 2000) con la cual determinamos la usabilidad de las herramientas. Para llevarlo a cabo se siguieron tres pasos o etapas:

Paso 1 Reclutamiento de usuarios: Se reclutaron los usuarios que se muestran en la tabla 9 los cuales cumplen con ciertos criterios relacionados con discapacidades visuales.

Tabla 9. Estudiantes Discapacitados Visuales

Foto	Nombre	edad	Dirección	Semestre	Programa
	Diego Alejandro Lugo	22	Circasia Vereda Villa Ligia C#20	6	Ingeniería de Software
	Santiago Luna Marín	25	Armenia, Quindío, Calle 31 # 21-74	5	ingeniería de software
	Carlos Alberto Bernal Giraldo	21	Cra22a #207 armenia	Egresado	Economía

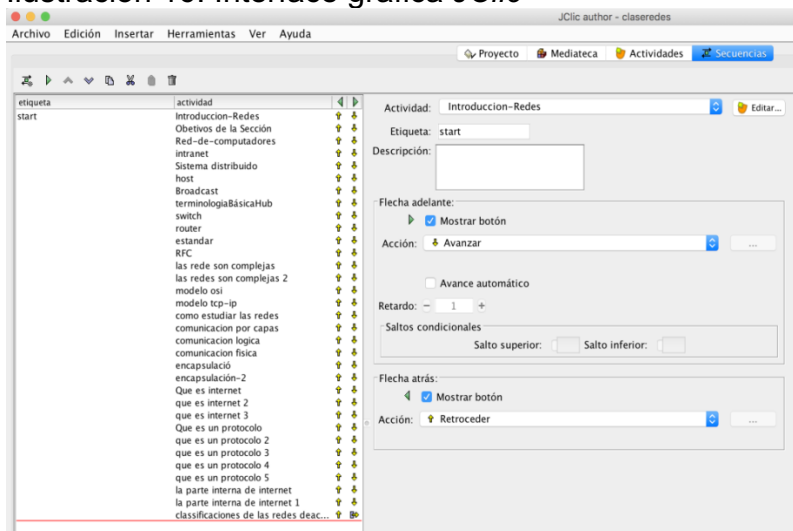
Fuente: Elaboración propia

Paso 2 Tareas por realizar:

- A los alumnos se les proporcionará una serie de materiales educativos y tareas o actividades a realizar.
- los estudiantes o usuarios se les proporcionara un equipo con las clases ya configuradas o en su defecto se les administrara los archivos necesarios para su instalación.
- A los participantes con discapacidad visual se les otorgará una tarjeta *NFC* para hacer uso del dispositivo que automatizará el proceso de acceso al material.

En la ilustración 16 se puede observar la interfaz gráfica de la herramienta de autor llamada *JClic* con la que se desarrolló el material educativo.

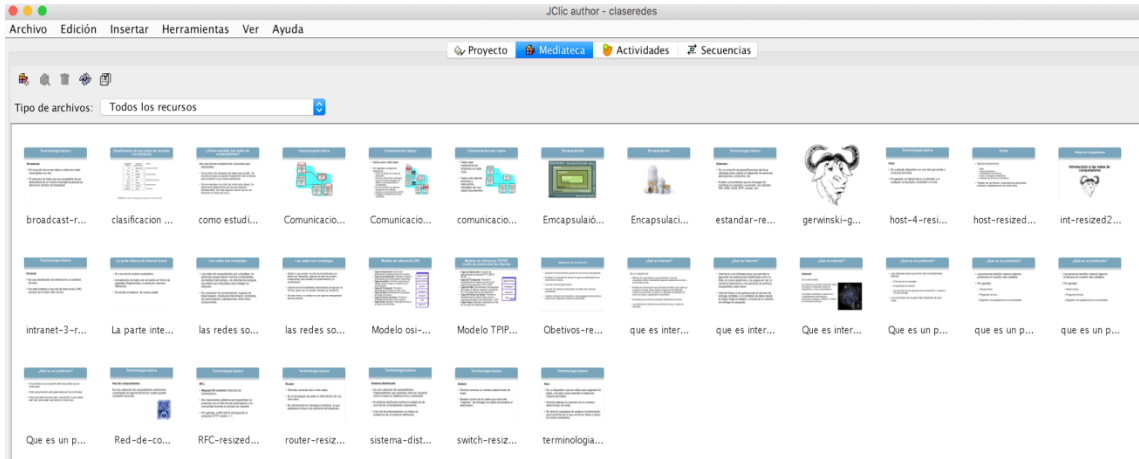
Ilustración 16. Interface gráfica *JClic*



Fuente: Elaboración propia

La ilustración 17 muestra las diferentes secciones del material educativo.

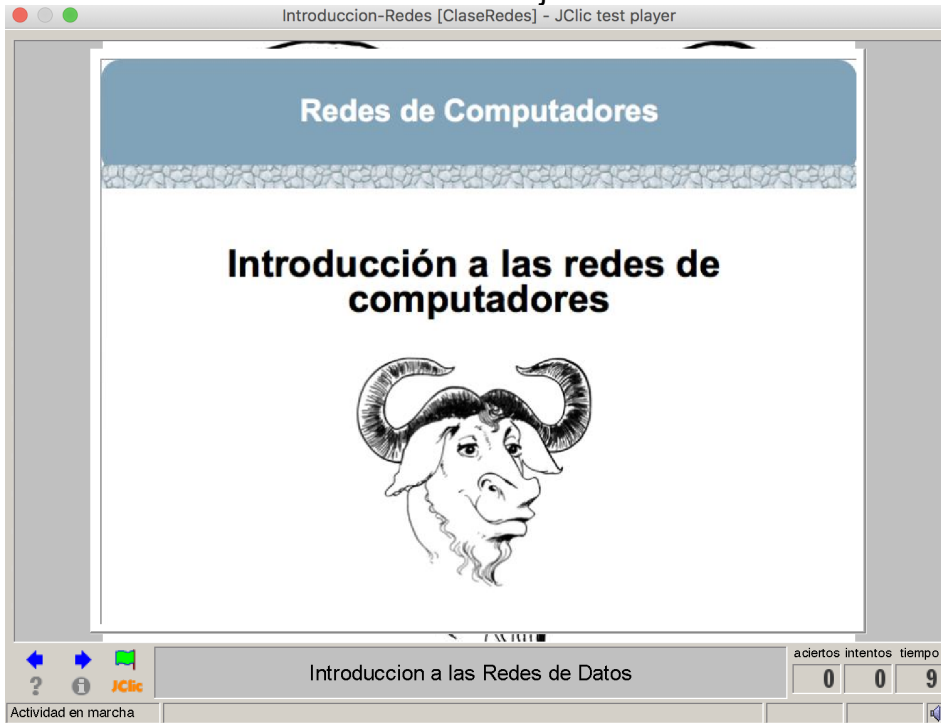
Ilustración 17. Secciones material educativo



Fuente: Elaboración propia

En la ilustración 18 se puede observar la ejecución del material educativo.

Ilustración 18. Material educativo en ejecución



Fuente: Elaboración propia

En la ilustración 19 se puede observar como los participantes hacen uso del lector de tarjetas NFC para ejecutar el material educativo.

Ilustración 19. Ejecución de material educativo con tarjeta *NFC*



Fuente: Elaboración propia

Paso 3 Escuchar los usuarios: mientras los usuarios realizan las actividades asignadas se les pedirá que hablen de cómo se sienten y lo que están experimentando, con la intención de escuchar y tomar notas para contrastar con los criterios mostrados en la tabla 10. Para observar los videos realizados en esta prueba ver el ANEXO G.

Tabla 10. Criterios prueba *Think Aloud*

Encuesta <i>Think Aloud</i>			
PROCESO: Uso de herramientas de autor libre y material educativo			
ASPECTOS	Funcionalidad	Estructura	Proceso
	Satisfacción del usuario	del	Confiabilidad, eficiencia, mantenimiento

Fuente: Elaboración propia

En el anexo A se pueden observar las encuestas realizadas en Formularios de Google y pensando en voz alta.

3. RESULTADOS DE LA INVESTIGACIÓN

En este capítulo se detallarán los resultados obtenidos en la investigación tanto las ventajas y desventajas que encontramos, como los porcentajes obtenidos de las pruebas de accesibilidad.

3.1 ANÁLISIS DE VENTAJAS Y DESVENTAJAS DEL PROTOTIPO

Con respecto a la investigación realizada se puede determinar que existe una amplia gama de herramientas de autor que se pueden adaptar para los distintos tipos de discapacidad, en especial la visual, pero también nos encontramos una serie de dificultades ya que algunos de estos entornos y herramientas son difíciles de configurar por ejemplo los sistemas operativos libres.

La investigación nos permite determinar que el uso del *software* libre en los entornos académicos nos proporciona un alto nivel de adaptabilidad a nuestras necesidades. Además, las libertades que ofrece su licencia proporcionan un valioso objeto de estudio.

Por otra parte, también se debe mencionar que otra ventaja importante de las herramientas de autor libres o del *software* libre en general, es su numerosa y creciente comunidad, que nos permite adquirir y compartir conocimiento libremente.

3.2 PROTOTIPO

Se proporcionará a la UNAB los siguientes productos:

- Material educativo accesible creado en la herramienta de autor seleccionada (*JClic*)-Anexo B.
- Guía de construcción de dispositivo lector de tarjetas *NFC* para alumnos con casos graves de discapacidad visual-Anexo C.
- Guía de adaptación para discapacitados visuales en sistemas operativos tipo *Unix Ubuntu 18*-Anexo D.
- Guía de elaboración de materiales educativos con *jclic*-Anexo E.
- Artículo-Anexo F.
- Video evidencias-Anexo G.

3.3 PRUEBAS DE FUNCIONAMIENTO

Para la evaluación del acceso a los materiales se tomaron en cuenta los siguientes aspectos, EQUIVALENTE, ALTERNATIVO, DIRECTO, COMPATIBLE.

Para las herramientas de autor, PRINCIPIOS GENERALES, TECLADO, DESCRIPTIVOS APUNTAORES, PANTALLA, SONIDO Y MULTIMEDIA, NOTIFICACIÓN AL USUARIO, INFORMACIÓN DEL OBJETIVO, TIEMPO, DOCUMENTACIÓN, OTROS REQUISITOS.

Los criterios utilizados para asignar la puntuación son los siguientes:

Si: Cumple con los requerimientos - a este criterio se le da soporte por medio de los videos elaborados con la técnica de pensar en voz alta.

No: No cumple con los requerimientos – a este criterio se le da soporte por medio de los videos elaborados con la técnica de pensar en voz alta.

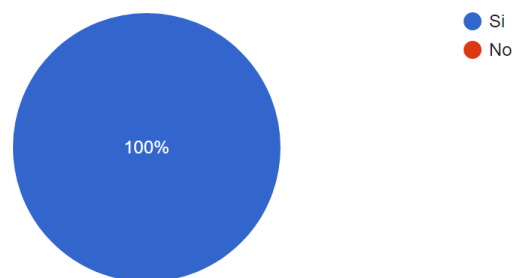
En las siguientes imágenes observaremos los porcentajes de la encuesta.

Sección 1 – Preguntas Material Educativo Equivalente:

Ilustración 20. Equivalente

Se le proporciono el material educativo por diferentes medios?

2 respuestas



Fuente: Elaboración propia

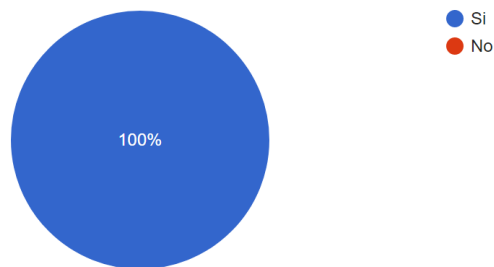
Como podemos observar en la gráfica el material educativo cumple en un 100% con el concepto **Equivalente**

Alternativo:

Ilustración 21. Alternativo

Se proporcionan diferentes entornos para acceder a los materiales educativos? Alternativo

2 respuestas



Fuente: Elaboración propia

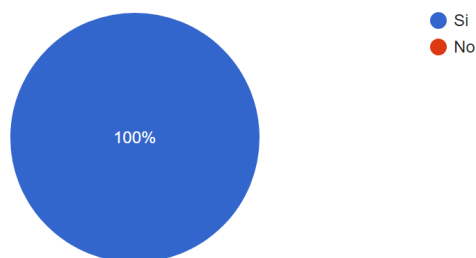
Como podemos observar en la gráfica se cumple en un 100% el concepto de Alternativo evidenciando que se proporcionaron los entornos necesarios para satisfacer las necesidades de los alumnos.

Directo:

Ilustración 22. Directo

El material educativo permite el acceso a todo tipo de estudiantes sin importar si poseen discapacidad o no?

2 respuestas



Fuente: Elaboración propia

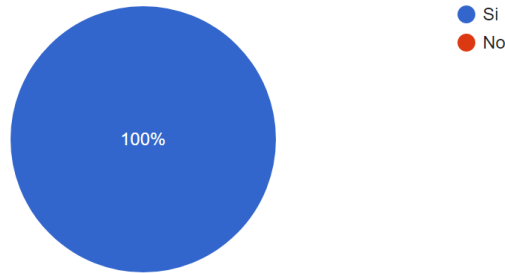
Se cumple en un 100% con el concepto de Directo al permitir el acceso a material educativo a cualquier tipo de usuario.

Compatible:

Ilustración 23. Compatible

Para acceder al material educativo se proporcionaron ayudas que faciliten el acceso?Compatible

2 respuestas



Fuente: Elaboración propia

Con lo visto en la gráfica deducimos que se cumple en un 100% el concepto de Compatible ya que se proporcionaron ayudas por medio de manuales y dispositivos hardware para facilitar el acceso al material educativo.

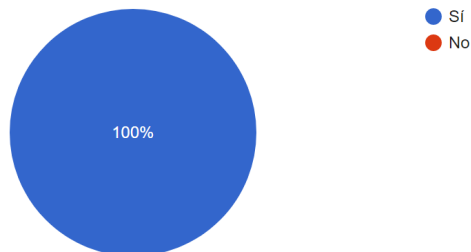
Sección 2 – Preguntas Herramientas de Autor

Principios Generales:

Ilustración 24. Principios generales

El entorno de software minimiza el número de acciones para acceder al material educativo?

2 respuestas



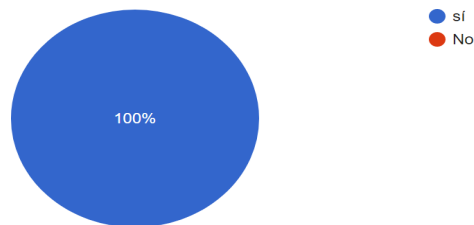
Fuente: Elaboración propia

El dispositivo configurado en la Raspberry pi cumple su función, permitiendo que los usuarios hagan un número mínimo de acciones, en este caso solo una acción que es pasar su tarjeta por el lector.

Ilustración 25. Principios generales-2

El usuario puede navegar por el material educativo por medio de sus dispositivos de entrada y seleccionar hacer uso de solo uno de ellos?

2 respuestas

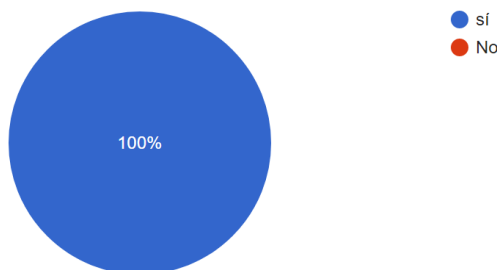


Fuente: Elaboración propia

Ilustración 26. Principios generales -3

Se usan los servicios accesibles del sistema operativo(lupa, lector de pantalla, contrastes, etc)?

2 respuestas



Fuente: Elaboración propia

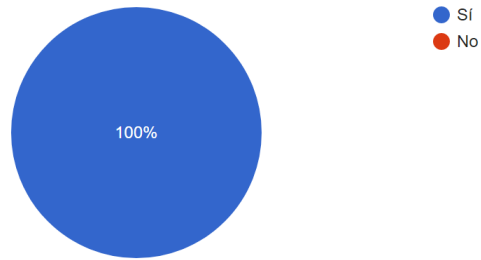
De acuerdo con las respuestas seleccionadas de los usuarios se observa que los **principios generales** se cumplen en un 100%.

Teclado:

Ilustración 27. Teclado-1

los estudiantes pueden hacer uso del material educativo solo con teclado?

2 respuestas

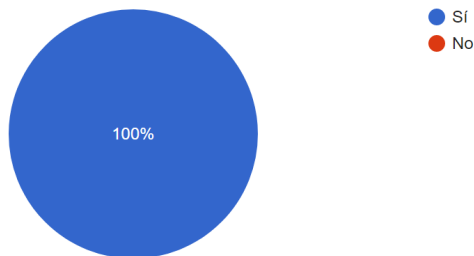


Fuente: Elaboración propia

Ilustración 28. Teclado-2

la navegación por teclado ofrece alternativas a la pulsación simultánea de varias teclas?

2 respuestas

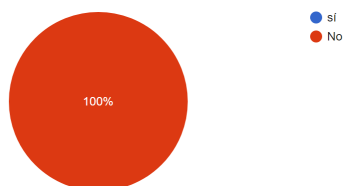


Fuente: Elaboración propia

Ilustración 29. Teclado-3

La navegación por teclado cumple con no permitir la activación de objetos de la interface?

2 respuestas

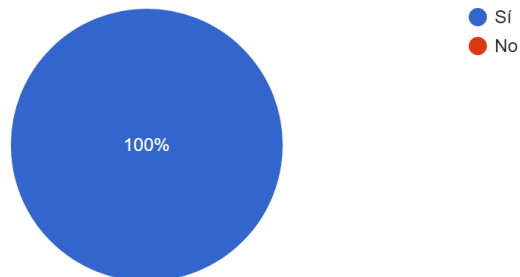


Fuente: Elaboración propia

Ilustración 30. Teclado-4

Se ofrecen atajos de teclado para las funciones mas importantes?

2 respuestas

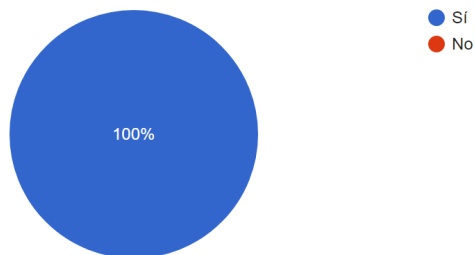


Fuente: Elaboración propia

Ilustración 31. Teclado-5

El usuario puede modificar las funciones asignadas a los atajos de teclado?

2 respuestas



Fuente: Elaboración propia

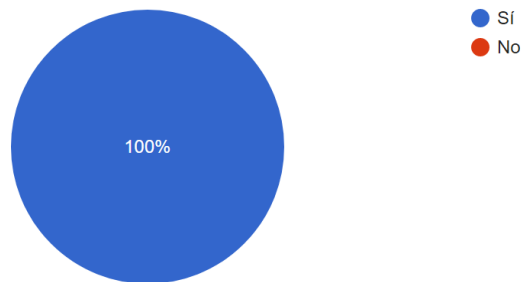
Con respecto a las gráficas referentes a los aspectos del **teclado** tenemos un porcentaje de 80%, siendo este un buen porcentaje frente al problema de permitir la no activación de objetos de la interface.

Descriptivos Apuntadores:

Ilustración 32. Descriptivos apuntadores-1

El entorno permite cambiar las asignaciones de funciones de los botones del ratón y hacer doble clic?

2 respuestas

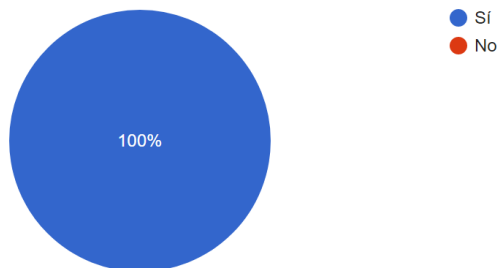


Fuente: Elaboración propia

Ilustración 33. Descriptivos apuntadores-2

El entorno permite arrastre mediante pulsaciones simples de teclas o botones?

2 respuestas



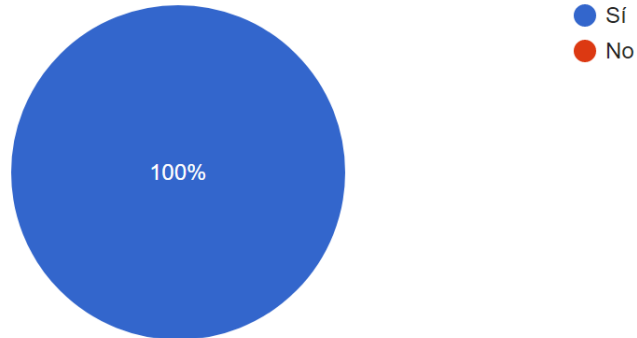
Fuente: Elaboración propia

De acuerdo con el porcentaje obtenido de 100% se expresa que es un éxito el aspecto de **descriptivos apuntadores** permitiéndose una alta configuración de las funciones asignadas a nuestros teclados y ratón.

Pantalla:

Ilustración 34. Pantalla-1
Se permite Cambiar el color del entorno?

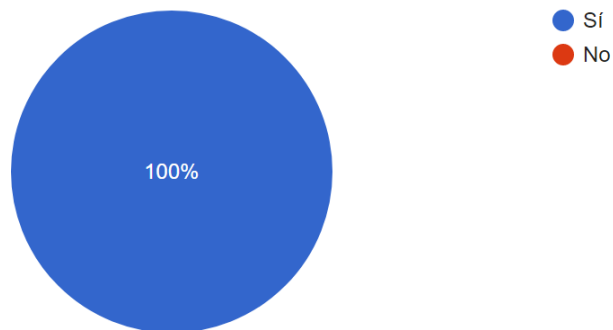
2 respuestas



Fuente: Elaboración propia

Ilustración 35. Pantalla-2
El entorno permite utilizar funciones para mostrar el texto?

2 respuestas

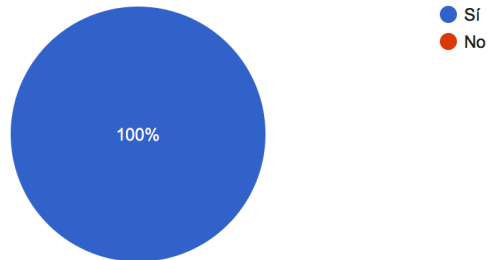


Fuente: Elaboración propia

Ilustración 36. Pantalla-3

El usuario puede cambiar aspectos visuales en el entorno?

2 respuestas



Fuente: Elaboración propia

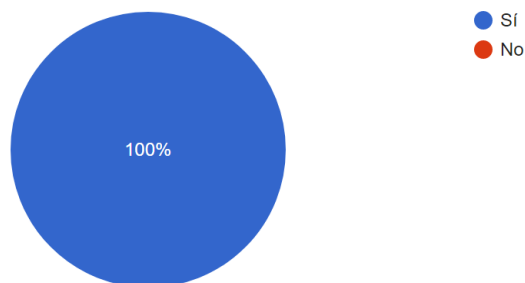
Observando el porcentaje del aspecto **Pantalla** se puede hablar de un éxito del 100%.

Sonido y Multimedia:

Ilustración 37. Sonido y multimedia-1

Se permite que el usuario modifique parámetros de sonido?

2 respuestas



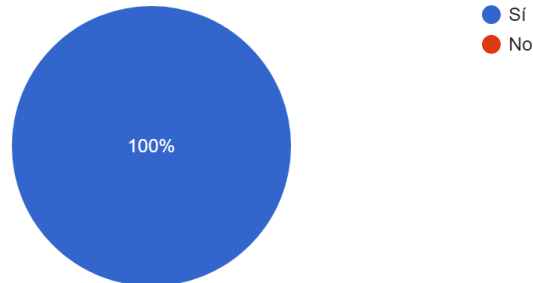
Fuente: Elaboración propia

El aspecto **Sonido y Multimedia** posee un porcentaje de 100% ya que los encuestados determinaron que se pueden modificar los parámetros de sonido.

Notificación al Usuario:

Ilustración 38. Notificación al usuario-1
los mensajes emitidos al usuario son redactados para facilitar su identificación y comprensión?

2 respuestas



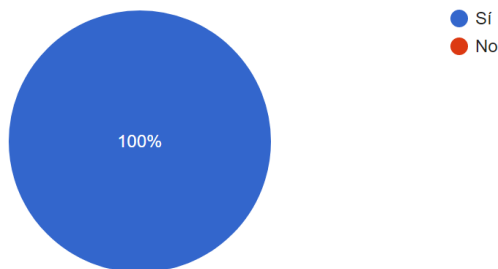
Fuente: Elaboración propia

El aspecto **notificación al usuario** posee un porcentaje de 100% de acuerdo a la calificación de los usuarios

Información de Objetivos:

Ilustración 39. Información de objetivos
Se ofrece información textual asociada a todos los objetivos de la interfaz?

2 respuestas



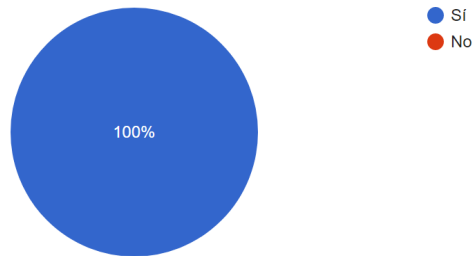
Fuente: Elaboración propia
Aspecto **Información de Objetivos** cumple en un 100%.

Tiempo:

Ilustración 40. Tiempo-1

Se le ofrecen a los usuarios el control sobre aspectos dependientes del tiempo?

2 respuestas



Fuente: Elaboración propia

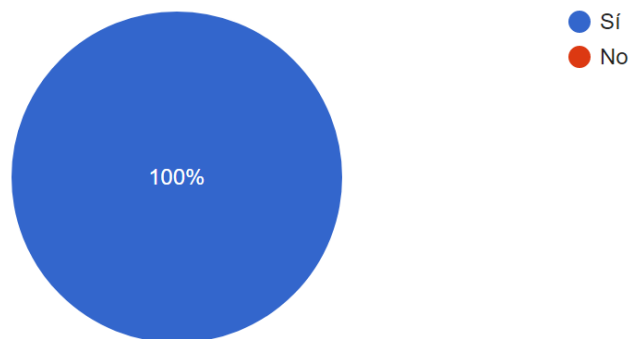
De acuerdo con la gráfica el aspecto **Tiempo** se cumple en un 100%.

Documentación:

Ilustración 41. Documentación-1

Se ofrece documentación clara y sencilla?

2 respuestas

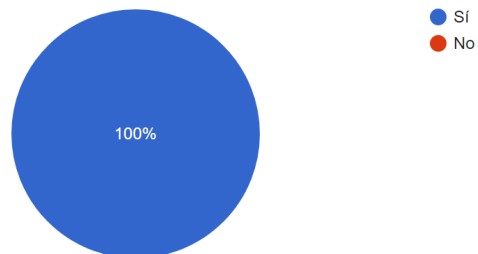


Fuente: Elaboración propia

Ilustración 42. Documentación-2

Se ofrece la documentación en formatos alternativos bajo petición del usuario?

2 respuestas

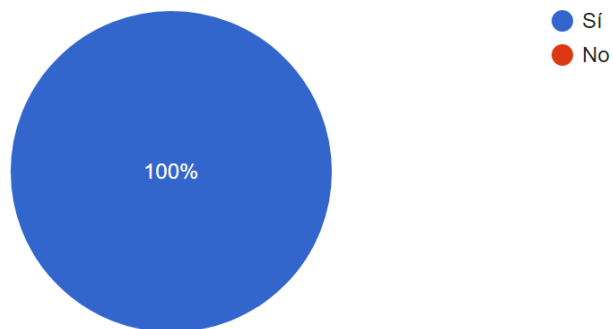


Fuente: Elaboración propia

Ilustración 43. Documentación-3

Se ofrece ayuda en pantalla?

2 respuestas



Fuente: Elaboración propia

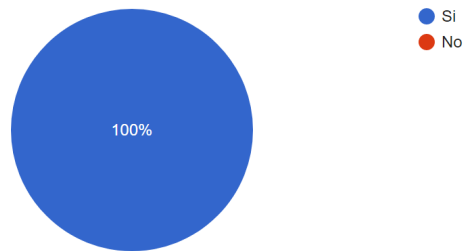
Con respecto a la calificación de los usuarios representada en las gráficas el Aspecto de documentación se cumple en un 100%.

Otros requisitos:

Ilustración 44. Otros requisitos-salir

Las aplicaciones ofrecen la función de salir del programa?

2 respuestas



Fuente: Elaboración propia

Con respecto a la gráfica los otros requisitos en este caso la función de salir del programa se cumple en un 100.

4. CONCLUSIONES

Gracias a nuestro estado del arte pudimos obtener conocimiento de una gran gama de herramientas de autor con los cuales seleccionamos una serie de *software* de acuerdo con su licencia y fácil capacidad de adaptación a las necesidades requeridas.

Nuestra selección de las herramientas de autor se sujetó a los siguientes criterios:

- Licencias libres. Que permitan hacer uso del *software* sin la limitante del costo, ni uso obligatorio de algún hardware.
- Multiplataforma que permita la ejecución de las herramientas de autor encargadas del desarrollo de los materiales educativos en diferentes entornos o ecosistemas de *software*.
- Portable que permita llevar los productos de *software* e interpretarlos en otros ambientes.

Esto nos permitió desarrollar el prototipo utilizando tecnologías libres como: Sistemas Operativos *GNU/Linux (Ubuntu, Vinux, Sonar, Raspbian)*, lenguaje de programación *Python*, herramienta de autor para crear materiales educativos *JClic*, lector de pantalla orca, Navegador *web* en modo texto *Lynx*.

Este dispositivo adaptado de herramientas de *software* libre permite la navegación accesible por medio de este y el uso de las herramientas de autor. Otra de sus funcionalidades es la reducción de las acciones que tienen que hacer los usuarios por medio de su módulo *Raspberry pi* permitiendo hacer uso de los contenidos académicos sin necesidad de interactuar con el sistema operativo o las herramientas, lo único que el usuario debe hacer es acceder con su tarjeta *NFC* y el entorno por medio de unos scripts en *Python* el cual se encargará del consumo de los materiales.

Las pruebas de accesibilidad se comprobaron por medio de dos tipos de encuesta, las cuales arrojaron los siguientes resultados:

Material educativo:

Equivalente: 100%

Alternativo: 100%

Directo: 100%

Compatible: 100%

Herramientas de autor

Principios generales: 100%

Teclado: 80%

Descriptivos apuntadores: 100%

Pantalla: 100%

Sonido y multimedia: 100%

Notificaciones al usuario: 100%

Información del objetivo: 100%

Tiempo: 100%

Documentación: 100%

Otros requisitos: 100%

Con los resultados vistos anteriormente se puede afirmar que los usuarios quedaron satisfechos con el ecosistema configurado y automatizado para ellos.

Recomendaciones finales: Se recomienda continuar con el proyecto y realizar distribuciones estables de estos entornos configurados, para pasar de un prototipo a una distribución oficial.

Adicional a esto se recomienda continuar con el modulo lector de tarjetas *NFC* ya que este permitirá cambiar la forma como los usuarios interactúan con el sistema y permitirá automatizar los procesos que en ocasiones distraen y ocasionan pérdidas de tiempo a los estudiantes.

5. RECOMENDACIONES Y TRABAJOS FUTUROS

El prototipo para la creación de materiales educativos haciendo uso de herramientas de autor libre tiene como su nombre lo dice el nivel de un prototipo, con unas adaptaciones y configuraciones básicas, este ecosistema de *software* adaptado se puede llegar a ser una distribución de sistema operativo *GNU/Linux* estable y funcional, adicionándole a esto la ventaja de interactuar con los materiales educativos por medio del lector *NFC* cambiando la forma en cómo se comunica con el sistema operativo.

Esto permite a que futuros investigadores se interesen en el proyecto y generen nuevas ideas añadiendo funcionalidades al prototipo como lo pueden ser:

- Modulo lector *NFC* que permita automatizar otras funciones del sistema, como, por ejemplo, el envío de correos por comandos de voz.
- Reconocimiento facial de los alumnos por medio de un módulo añadido a la *Raspberry* que permita acceder a los materiales educativos con su rostro.
- Servidor de *JClic* en red

Nota: Actualmente nos encontramos desarrollando el modulo que permita activar por medio de una tarjeta *NFC* un *software* diseñado en *Python* que captura la voz del estudiante convirtiéndola en texto y enviándola por correo al destinatario seleccionado.

La intención de proyecto futuro es elaborar un prototipo que permita a los alumnos hacer la mayoría de las funciones cotidianas en sus entornos académicos con un sistema operativo por medio de tarjetas *NFC*.

REFERENCIAS

- ABBYY. (2017). OCR. Retrieved June 8, 2017, from <https://www.abbyy.com/es-la/finereader/what-is-ocr/>
- Alfonso Cuba, Ileana María. Universidad 2012. Curso corto 17: Usabilidad en la Educación: Garantía de la calidad de la Educación Virtual. La Habana, CU: Editorial Universitaria, 2012. ProQuestebruary. Web. 25 March 2017.
- Alba, C., Bautista, A. y Nafría, E. (1997). Tecnología Educativa: evolución de una concepción, en Alonso, C. (Coord.). *La Tecnología Educativa a finales del siglo XX: concepciones, conexiones y límites con otras disciplinas*. Barcelona: Centre Telemàtic Editorial, SRL
- APPLE INC. (2018). ENGLISH APPLE INC. SOFTWARE LICENSE AGREEMENT FOR macOS High Sierra For use on Apple-branded Systems PLEASE READ THIS SOFTWARE LICENSE AGREEMENT (" LICENSE ") CAREFULLY BEFORE USING THE APPLE SOFTWARE. BY USING THE APPLE SOFTWARE, YOU ARE AGREE. Retrieved from <https://www.apple.com/legal/sla/docs/macOS1013.pdf>
- Aquino Zúñiga, S. P., García Martínez, V., & Izquierdo, J. (2012). La inclusión educativa de ciegos y baja visión en el nivel superior: Un estudio de caso. *Sinéctica*, (39), 01–21. Retrieved from http://www.scielo.org.mx/scielo.php?pid=S1665-109X2012000200007&script=sci_arttext&lng=en
- Autoridad nacional para la innovación personal Gobierno de Panamá. 2010. "Software Para Personas Con Alguna Discapacidad - SOLCA." <http://solca.aig.gob.pa/home/para-discapacitados> (April 23, 2017).
- Ávila-Cabrera, J. J., Talaván, N., & Costal, T. (2016). Traducción y accesibilidad audiovisual. Barcelona, ESPAÑA: Editorial UOC. Retrieved from <http://www.ebrary.com>
- Bain, M., Gallego Rodriguez, M., Martínez Ribas, M., & Rius Sanjuán, J. (2004). *Software libre. Aspectos legales y de explotación del software libre. Parte 1. Doctor*. Retrieved from <http://softlibre.unizar.es/manuales/legal/909.pdf>
- Belloch, C. (2013). Diseño Instruccional. Material Didáctico Web de La Unidad de Tecnología Educativa (UTE) de La Universidad de Valencia, 21, 2–4. <https://doi.org/978-987-24871-6-4>

Boren, M. T., & Ramey, J. (2000). Thinking aloud: Reconciling theory and practice. *IEEE Transaction on Professional Communication*, 43(3), 261–278. <https://doi.org/10.1109/47.867942>

Carrasco, F. I. (2008). Accesibilidad y estandarización en el ciclo de creación de material educativo.

de Assis, F. (2010). Software libre y educación, un estudio de casos en la enseñanza obligatoria en Cataluña., 72. Retrieved from http://www.tdx.cat/bitstream/handle/10803/35696/Tesis_FACS.pdf?sequence=

Dom Nov. (2002). Emacspeak --El escritorio de audio completa. Retrieved May 25, 2017, from <http://www.cs.cornell.edu/home/raman/emacspeak/>

Free Software Foundation. (2017). Free Software Foundation. Retrieved June 8, 2017, from <https://www.fsf.org/about/>

FSF. (2017). gnu.org. Retrieved from <https://www.gnu.org/licenses/license-list.es.html>

Gutierrez, E. (2015). an Esteban Guti ´ errez Posada Interfaces Tang ´ ives e o Design de Ambientes Educacionais para Co-constru , c ~ ao de Narrativas, 3, 104–107. Retrieved from <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=2ahUKEwj0h8TMpazcAhWHuVkkHRrrDhIQFjABegQIAhAC&url=http%3A%2F%2Fwww.nied.unicamp.br%2Fojs%2Findex.php%2Ftsc%2Farticle%2Fdownload%2F153%2F140&usg=AOvVaw13TCkSnBqxA3IP8zSWvN8r>

Kirk Reiser, & Andy Berdan. (2015). Speakup Project. Retrieved May 25, 2017, from <http://www.linux-speakup.org/projects.html>

Martínez De La Teja, G. M. (2003). USABILIDAD Y ACCESIBILIDAD EN WEB. Retrieved from <http://www.semec.org.mx/archivos/6-11.pdf>

Michael De La Rue, & Sharon Snider. (2002). CÓMO Linux Accesibilidad. Retrieved May 25, 2017, from <http://tldp.org/HOWTO/Accessibility-HOWTO/>

Ministerio de Sanidad, S. S. E. I. (2016). Manual de codificación CIE-10-ES Diagnósticos. Clasificación Internacional de Enfermedades, 1, 303.

Retrieved from https://www.msssi.gob.es/estadEstudios/estadisticas/normalizacion/CIE10/UT_MANUAL_DIAG_2016_prov1.pdf

Naciones Unidas. (2017). OMS: Organización Mundial de la Salud - Oficina del Secretario General para la Juventud. Retrieved June 8, 2017, from <http://www.un.org/youthenvoy/es/2013/09/oms-organizacion-mundial-de-la-salud/>

NV Access. (2017). Acceso NV. Retrieved May 25, 2017, from <https://www.nvaccess.org/>

Organización de Naciones Unidas. (1994). El programa de las Naciones Unidas sobre la discapacidad. Retrieved from <http://www.un.org/spanish/disabilities/default.asp?id=498>.

Pérez, U. R. (2014). Entrenamiento de habilidades laborales: UF0801. Madrid, ES: Editorial Tutor Formación. Retrieved from <http://www.ebrary.com>

PortalProgramas. (2016). Ranking de universidades en software libre (RuSL), 1-2. Retrieved from <https://www.portalprogramas.com/software-libre/ranking-universidades/>

Raspberry Pi Foundation. (2016). Raspberry Pi 3 Model B, 2837. Retrieved from <http://www.instructables.com/id/Raspberry-Pi-3-Model-B-MIFARE-RC522-RFID-Tag-Readi/>

Pérez, U. R. (2014). Entrenamiento de habilidades laborales: UF0801. Madrid, ES: Editorial Tutor Formación. Retrieved from <http://www.ebrary.com>

Sánchez Caballero, M. (2010). *No sólo usabilidad: revista multidisciplinar sobre diseño, personas y tecnología*. Yusuf Hassan Montero. Retrieved from http://www.nosolousabilidad.com/articulos/software_libre.htm?utm_source=feedburner

Sonar GNU Linux. (2017). Sonar GNU Linux – An Accessible OS For People With A Disability. Retrieved March 26, 2017, from <http://sonargnulinux.com/>

The GNOME project. (2017). Proyectos / Orca - GNOME Wiki! Retrieved May 22, 2017, from <https://wiki.gnome.org/Projects/Orca>

The Society of Motion Picture and Television Engineers. (2013, June 5). W3C. <https://doi.org/10.5594/S9781614827597>

United Nations. (2003). UN Enable - Boletín sobre los discapacitados No 2/1999 Página 4/6. Retrieved May 10, 2017, from <http://www.un.org/esa/socdev/enable/dpbe19992c.htm>

UsabilityNet. (2006). UsabilityNet: Normas Internacionales. Retrieved May 17, 2017, from http://www.usabilitynet.org/tools/r_international.htm#9241-1x

Valverde Berrocoso, Jesús. El software libre en la innovación educativa. Salamanca, ES: Ediciones Universidad de Salamanca, 2014. ProQuestebruary. Web. 25 March 2017.

Vinux Project 2017. (2017). Downloads | Vinux Project. Retrieved March 26, 2017, from <http://vinuxproject.org/downloads>

Web@lynx.browser.org. (2017). Lynx Information. Retrieved November 27, 2017, from <http://lynx.browser.org/>

zonaClic - JClic. (2017). Retrieved March 27, 2017, from <http://clic.xtec.cat/es/jclic/>

ANEXOS

ANEXO A. Instrumentos de evaluación.

Instrumentos de evaluación	
Descripción	Instrumentos de evaluación para la accesibilidad del prototipo
Enlace	https://drive.google.com/drive/folders/1EWzM9LZ0K7CwzB83iASDNa2QBtgdmPib?usp=sharing

ANEXO B. Material educativo accesible creado en la herramienta de autor seleccionada (*JClic*).

Material educativo creado en <i>JClic</i>	
Descripción	Conjunto de archivos que componen el material educativo elaborado en la herramienta de autor <i>JClic</i>
Enlace	https://drive.google.com/drive/folders/1eJhr3vahnRR0gYlalQ5l-RljhPZzOEcm?usp=sharing

ANEXO C. Guía de construcción de dispositivos lector de tarjetas *NFC* para alumnos con casos graves de discapacidad visual.

Guía construcción de modulo lector <i>NFC</i>	
Descripción	Manual de elaboración de dispositivo lector de tarjetas <i>NFC</i> que automatizara el proceso de acceso a los materiales educativos.
Enlace	https://drive.google.com/drive/folders/1WwecB7c1a35-CUyR9e_37ePodG1oDTqH?usp=sharing

ANEXO D. Guía de adaptación para discapacitados visuales en sistemas operativos tipo *Unix Ubuntu 18*.

Guía de adaptación del sistema operativo <i>Ubuntu 18</i>	
Descripción	Manual de configuración de las herramientas de accesibilidad que posee el sistema operativo <i>Ubuntu 18</i>
Enlace	https://drive.google.com/drive/folders/1dluqSRjtViYsfCNMXtNIHrK_mdFbd0GC?usp=sharing

ANEXO E. Guía de elaboración de materiales educativos con *JClic*.

Guía de elaboración de materiales educativos	
Descripción	Manual básico de elaboración de actividad expositiva en <i>JClic</i>
Enlace	https://drive.google.com/drive/folders/11JsQIUQJQ08XnVVXy6pXaj_X3hwU3SUs?usp=sharing

ANEXO F. Artículo

Artículo	
Descripción	Artículo proyecto de grado
Enlace	https://drive.google.com/drive/folders/1eDLR8Nvf_iam5qFUUBQhp_7WLjXso2D?usp=sharing

ANEXO G. Video de evidencia del funcionamiento del prototipo

Videos	
Descripción	Videos de evidencia del funcionamiento del prototipo y la técnica de pensar en voz alta
Enlace	https://drive.google.com/drive/folders/1MeRnIM_shlqITafTp2pTBQ5vUm8UMfk7?usp=sharing