

UNIVERSIDAD NACIONAL DE TUMBES

FACULTAD DE CIENCIAS SOCIALES

ESCUELA PROFESIONAL DE EDUCACIÓN



El software educativo libre en el aprendizaje de la lectoescritura de los
niños de 4 años de edad

Trabajo académico presentado para optar el Título Profesional de
Segunda Especialidad de Educación Inicial

Autora:

Maira Angela Montalbán López

PIURA - PERÚ 2018

UNIVERSIDAD NACIONAL DE TUMBES

FACULTAD DE CIENCIAS SOCIALES

ESCUELA PROFESIONAL DE EDUCACIÓN



El software educativo libre en el aprendizaje de la lectoescritura de los
niños de 4 años de edad

Los suscritos declaramos que la monografía es original en su
contenido y forma.

Maira Angela Montalbán López (Autora)

Oscar Calixto La Rosa Feijoo (Asesor)

PIURA - PERÚ 2018



UNIVERSIDAD NACIONAL DE TUMBES
FACULTAD DE CIENCIAS SOCIALES
ESCUELA PROFESIONAL DE EDUCACIÓN
PROGRAMA DE SEGUNDA ESPECIALIDAD

ACTA DE SUSTENTACION DE TRABAJO ACADEMICO

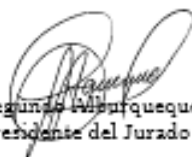
En Piura, a los cuatro días del mes de agosto del dos mil dieciocho, se reunieron en la I.E.P Pontificia los integrantes del Jurado Evaluador, designado según convenio celebrado entre la Universidad Nacional de Tumbes y el Consejo Intersectorial para la Educación Peruana, al Dr. Segundo Alburquerque Silva, coordinador del programa; representantes de la Universidad Nacional de Tumbes (Presidente), Dr. Andy Figueroa Cárdenas (Secretario) y Mg. Ana María Javier Alva (vocal) representantes del Consejo Intersectorial para la Educación Peruana, con el objeto de evaluar el trabajo académico de tipo monográfico denominado: "El software educativo libre en el aprendizaje de la lectoescritura de los niños de 4 años de edad", para optar el título Profesional de Segunda Especialidad en Educación Inicial a la señora MAIRA ANGELA MONTALBÁN LÓPEZ.

A las DIECIOCHO horas CERO minutos y de acuerdo a lo estipulado por el reglamento respectivo, el Presidente del Jurado dio por iniciado el acto.

Luego de la exposición del trabajo, la formulación de preguntas y la deliberación del jurado lo declararon APROBADO por UNANIMIDAD con el calificativo BUENO.

Por tanto, MAIRA ANGELA MONTALBÁN LÓPEZ., queda APTA, para que el Consejo Universitario de la Universidad Nacional de Tumbes, le expida el título profesional de Segunda Especialidad en Educación Inicial.

Siendo las DIECIOCHO horas con VEINTICINCO minutos, el presidente del jurado dio por concluido el presente acto académico, para mayor constancia de lo actuado firmaron en señal de conformidad todos los integrantes del jurado.


Dr. Segundo Alburquerque Silva
Presidente del Jurado


Dr. Andy Figueroa Cárdena
Secretario del Jurado


Mg. Ana María Javier Alva.

DECLARACIÓN JURADA DE AUTENTICIDAD

Yo, MAIRA ANGELA MONTALBÁN LÓPEZ estudiante del Programa Académico de Segunda Especialidad de Educación Inicial la Facultad de Ciencias Sociales de la Universidad de Tumbes.

Declaro bajo juramento que:

1. Soy autor del trabajo académico titulado: INFLUENCIA DEL SOFTWARE EDUCATIVO LIBRE: FACIL EN EL APRENDIZAJE DE LA LECTOESCRITURA DE LOS NIÑOS DE 4 AÑOS DE EDAD, la misma que presento para optar el título profesional de segunda especialidad.
2. El trabajo Académico no ha sido plagiado ni total ni parcialmente, para la cual se han respetado las normas internacionales de citas y referencias para las fuentes consultadas.
3. El trabajo Académico presentado no atenta contra derechos de terceros.
4. El trabajo Académico no ha sido publicado ni presentado anteriormente para obtener grado académico previo o título profesional.
5. Los datos presentados en los resultados son reales, no han sido falsificados, ni duplicados, ni copiados.

Por lo expuesto, mediante la presente asumo frente a la UNTUMBES cualquier responsabilidad académica, administrativa o legal que pudiera derivarse por la autoría, originalidad y veracidad del contenido de El Trabajo Académico, así como por los derechos sobre la obra y/o invención presentada.

Tumbes, _____ de 2018

Firma
MAIRA ANGELA MONTALBÁN LÓPEZ

El presente trabajo está dedicado a Dios quien me guía en cada uno de mis pasos, mis queridos Padres quienes han hecho de mí una persona de bien.

INDICE

RESUMEN

INTRODUCCIÓN

CAPITULO I 1

OBJETIVOS DE LA MONOGRAFIA 11

1.1 Objetivo General 11

1.2 Objetivos Específicos 11

CAPITULO II 12

MARCO TEORICO 12

2.1 Software Libre 12

2.1.1 Conceptos de Software libre 13

2.1.1.1 La definición de la Free Software Foundation (FSF): el software libre 14

2.1.1.2 La definición de la Open Source Initiative (OSI): el software de código
abierto 15

2.1.2 Las licencias en el software privativo y libre 17

2.1.2.1 Licencias privativas 17

2.1.2.2 Licencias libres 19

2.1.3 El origen del software libre 20

2.2 El Software libre en educación 24

2.2.1 Ventajas pragmáticas 19

2.2.2 Razones políticas, éticas y sociales 20

2.2.3 El software libre en la educación 21

2.2.4 El software libre y la innovación en tecnología educativa 24

2.3 Software FACIL 24

2.3.1 Funcionamiento del programa 25

2.3.2 2. Actividades que el programa puede generar 28

2.4 Lectoescritura 29

2.4.1 La importancia de aprender a leer y escribir 29

2.4.2 Cómo enseñar a leer y escribir 32

2.4.2.1 Cuando se debe empezar la enseñanza sistemática de Lectoescritura	32
2.4.2.2 Es necesario el aprestamiento para la lectoescritura	33
2.4.2.3 Qué aprendizajes debe lograr el niño en lectoescritura	34
CONCLUSIONES	37
REFERENCIAS CITADAS	38

RESUMEN

El presente trabajo tiene como objetivos determinar la influencia del software libre: FACIL en el aprendizaje de la lectoescritura de los niños de 4 años de edad, descargar e instalar el software libre: FACIL en las computadoras a usar, capacitar a los alumnos en el manejo del software libre: FACIL, realizar la clase usando el software libre: FACIL, evaluar los resultados. Teniendo como conclusión que: El uso de tecnologías como los softwares educativos libres son una ayuda muy importante para el proceso de enseñanza-aprendizaje de los niños del nivel inicial, ya que nos brindan muchas ventajas sobre las herramientas de enseñanza tradicionales, más aun en un mundo donde los niños son nativos digitales; pero sin embargo no son insustituibles.

Palabras claves: Influencia, lectoescritura, FACIL

INTRODUCCION

El mundo en la actualidad está sumergido en una mega revolución industrial conocida como la era del conocimiento, en donde la tecnología avanza a pasos agigantados por todos los frentes a través de la superproducción de aparatos electrónicos (smartphones, tablets, computadores y laptops muy avanzadas), microelectrónicos (chipset, chips, microprocesadores), la automatización, la educación a larga distancia en entornos virtuales de aprendizajes; etc.

Específicamente en la educación esta llamada mega revolución, nos proporciona un sin número de herramientas que podemos utilizar en beneficio de nuestro proceso de enseñanza como docentes y en nuestro proceso de aprendizaje como alumnos; una de estas herramientas son sin duda alguna los softwares educativos con rango libre, cuyas interfaces son simples y amigables para el usuario final, haciendo que la relación software- usuario se torne sencilla.

Podemos encontrar softwares educativos libres para todas las áreas de la educación humana, desde la historia hasta la matemática, física y sin lugar a duda el área de comunicación, la cual hoy en día es cada vez más virtual que presencial; pero así la comunicación se vuelva virtual totalmente, su integración en el ser humano se da en los primeros años de su vida, razón por la cual es indispensable que la misma se da al más alto nivel, pues de no ser así, en primer lugar este niño sería un analfabeto más, sin posibilidades de comunicación e interrelación con los demás llevándolo al fracaso como ser humano.

Entonces se asevera que el no alfabetismo lectoescritor en el ser humano, es la base para su alfabetización digital del mundo de hoy, convirtiendo por esta razón a los softwares educativos de carácter abierto en herramientas indispensables, primero para lograr la alfabetización personal y luego la alfabetización digital.

Por todo lo antes mencionado, es imperativo que los docentes conozcan, pero sobre todo usen y sepan utilizar estas herramientas pedagógicas modernas, para ayudar a desarrollar a los niños desde su primera infancia escolar.

En el capítulo II, hago mención a las definiciones de softwares libre, software educativo libre, concepto de lectoescritura, que aprendizajes se esperan, definición y usos del software libre FACIL.

CAPITULO I

OBJETIVOS DE LA MONOGRAFIA

1.1 Objetivo General

Determinar la influencia del software libre: FACIL en el aprendizaje de la lectoescritura de los niños de 4 años de edad.

1.2 Objetivos Específicos

- ✓ Descargar e instalar el software libre: FACIL en las computadoras a usar.
- ✓ Capacitar a los alumnos en el manejo del software libre: FACIL
- ✓ Realizar la clase usando el software libre: FACIL
- ✓ Evaluar los resultados

CAPITULO II

MARCO TEORICO

2.1 Software Libre

“Antes de explicar qué es el software libre necesitamos conocer algunos aspectos clave del software, por ejemplo, cómo se produce y qué es el código fuente de los programas. Por software entenderemos tanto el sistema operativo como las aplicaciones que utilizamos los usuarios. Además del software, son muy importantes los formatos de fichero (la manera de organizar y codificar la información que producen las aplicaciones) y los protocolos de comunicación (la forma de comunicarse entre sí los programas a través de las redes). Pero de ellos hablaremos más adelante.” (Adell & Bernabè, s.f. p. 2)

“Un programa no es más que un conjunto de instrucciones que le dicen al ordenador qué tiene que hacer. Los programas los escriben los seres humanos utilizando lenguajes de programación. Pero antes de que el ordenador pueda ejecutar un programa es necesario traducir dichas instrucciones a su lenguaje, esto es a código máquina: largas series de ceros y unos. A fin de cuentas, un ordenador no es más que una gran cantidad de interruptores eléctricos, que pueden estar únicamente en uno de dos estados: dejar pasar la corriente o no dejarla pasar, uno o cero. Un programa le dice al ordenador qué interruptores poner a cero y cuáles poner a uno en cada momento, dependiendo de las acciones que ejecutemos en el teclado o con el ratón. El proceso de convertir un programa escrito en un lenguaje de programación a instrucciones inteligibles para el ordenador, se denomina compilación y lo hacen otros programas de ordenador especializados: los compiladores. Una vez el programa está

compilado ya es posible ejecutarlo, a cambio, una vez traducido a código máquina, es casi imposible que un ser humano entienda algo de la larga serie de unos y ceros en que se ha convertido. Y este es el meollo del software libre.” (Adell & Bernabè, s.f. p. 2)

2.1.1 Conceptos de Software libre

“El hecho de que, en inglés, el idioma en el que se acuñó y difundió el término software libre (free software), una misma palabra (free) signifique tanto libre como gratuito y que gran parte del software libre sea efectivamente gratuito, ha favorecido las malas interpretaciones: mucha gente considera equivalente los términos software libre y software gratuito. Sin embargo, el rasgo esencial que define el software libre es la libertad, no el precio. Cuando se habla de software libre (free software) debemos pensar en libertad de expresión (free speech), no en cerveza gratis (free beer). El propietario de los derechos sobre el software libre garantiza a los usuarios, mediante una licencia, una serie de libertades que no otorga el propietario del software privativo, que se reserva numerosos derechos en base a la legislación sobre propiedad intelectual (por ejemplo, no permite el acceso al código fuente o no permite ninguna modificación y su subsecuente distribución). El usuario de software privativo en realidad paga por el derecho a usar, con numerosas limitaciones, el software. Pero pagar por él, no lo convierte en algo de su propiedad. Más adelante trataremos este punto con mayor detenimiento.” (Adell & Bernabè, s.f. p. 4)

“Una idea crucial para entender el revuelo que ha creado el software libre es que software libre es mucho más que software. Así, se ha asociado a un modelo de desarrollo del software (Raymond, 1999), como una comunidad de prácticas (Edwards, 2004 & Tuomi, 2005), una escena (Lehman, 2004), una aproximación a cierto tipo de licencias (Perens, 1999), un modelo económico (Khalak, 2000; Lerner & Tirole, 2000), un sistema social y de valores (Lessig, 2004; Stallman, 1992), un movimiento social por una cuestión ética (Wynants & Cornelis, 2005) o un modelo híbrido, público-

privado, de innovación (Lyn, 2005). Todas estas perspectivas, junto a estrategias dirigidas a despolitizar las ideas que subyacen al concepto original de software libre y a hacerlo digerible a la industria, han producido bastante confusión. En la Internet y en la bibliografía se suelen emplear cada vez más acrónimos como FLOSS (Free, Libre, Open Source Software) o FOSS (sin Libre) o FS/OS para no entrar en polémicas sobre si lo importante es la libertad o el código abierto.” (Adell & Bernabè, s.f. p. 4)

2.1.1.1 La definición de la Free Software Foundation (FSF): el software

libre

“La FSF propugna como elemento clave para la definición del software libre la libertad de la comunidad de usuarios para poder ejecutar, copiar, estudiar, mejorar y redistribuir el software. La palabra clave aquí es libertad. Libertad de usar el programa para cualquier propósito, de estudiar cómo funciona y adaptarlo a las diferentes necesidades, de distribuir copias, de poder mejorarlo y de hacer públicas las mejoras. La única restricción es que, si redistribuimos el programa, tenemos que hacerlo reconociendo los mismos derechos en los usuarios de nuestras modificaciones. Como vemos, el acceso al código fuente es un requisito previo y necesario para ejercer la mayoría de estas libertades.” (Adell & Bernabè, s.f. p. 5)

“En concreto, la FSF se refiere a cuatro libertades que deben tener los usuarios del software para que pueda ser calificado como libre (Stallman, 1996)” (Adell & Bernabè, s.f. p. 5):

- “Libertad 0. Es la libertad de usar el programa para cualquier propósito.” (Adell & Bernabè, s.f. p. 5).
- “Libertad 1. La libertad de estudiar cómo funciona el programa y adaptarlo a las propias necesidades. Una condición previa para que se dé esta libertad es el acceso al código fuente” (Adell & Bernabè, s.f. p. 5).
- “Libertad 2. La libertad de redistribuir copias.” (Adell & Bernabè, s.f. p. 5).

- “Libertad 3. La libertad de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie. Esta libertad también requiere el acceso al código fuente.” (Adell & Bernabè, s.f. p. 5).

“Por tanto, un programa puede definirse como software libre solamente si los usuarios tienen todas estas libertades. Las cuatro libertades básicas de la FSF se concretan en la licencia GPL GNU (GNU General Public License)” (Adell & Bernabè, s.f. p. 5).

2.1.1.2 La definición de la Open Source Initiative (OSI): el software de código abierto

“En 1998, algunos programadores y activistas del software libre, como Eric S. Raymond, Bruce Perens y Tim O'Reilly entre otros, crearon la Open Source Initiative (OSI). Eric Raymond había escrito poco antes el ensayo sobre ingeniería de software *La catedral y el bazar* (Raymond, 1997), en el que describía el proceso de desarrollo de Linux como una manera nueva, diferente y muy eficiente de crear software. La OSI nació con el objetivo de crear y fomentar el uso de programas informáticos de código abierto. La razón por la que la OSI defiende el acceso al código fuente de los programas no es la libertad, sino la posibilidad de crear mejor software, adaptado a necesidades reales de los usuarios. La OSI se centra en destacar las ventajas pragmáticas de la utilización del software de código abierto, basadas en la constante exposición de la estructura y funcionamiento de los programas al escrutinio de la comunidad y en la capacidad de implicación y participación de sus usuarios en su mejora. El resultado final es que, según la OSI, el software de código abierto tiende a ser de mejor calidad que el privativo.” (Adell & Bernabè, s.f. p. 6)

“Para que un programa pueda ser considerado de código abierto, la OSI establece un decálogo de condiciones que ha de cumplir. En su versión 1.9 (OSI, 2006), y tal como las reproduce Mas (2005, p. 33-35), son las siguientes”

(Adell & Bernabè, s.f. p. 6):

- **“Libre distribución.** No se puede impedir la venta o distribución del programa o parte de él. Así mismo, tampoco se puede exigir el pago de un canon o tasa a cambio de su distribución por parte de terceros” (Adell & Bernabè, s.f. p. 6).
- **“Código fuente.** El programa debe incluir su código fuente y no se puede restringir su redistribución.” (Adell & Bernabè, s.f. p. 6).
- **“Trabajos derivados.** No debe impedirse realizar modificaciones o trabajos derivados del programa y debe permitirse que éstos sean distribuidos bajo los mismos términos que el software original.” (Adell & Bernabè, s.f. p. 6).
- **“Integridad del código de fuente original.** Puede exigirse que una versión modificada del programa tenga un nombre y número de versión diferente que el programa original para poder proteger al autor original de la responsabilidad de estas versiones.” (Adell & Bernabè, s.f. p. 6).
- **“No discriminación contra personas o grupos.** Las condiciones de uso del programa no pueden discriminar a una persona o un grupo de personas.” (Adell & Bernabè, s.f. p. 6).
- **“No discriminación contra usos.** No se puede negar a ninguna persona el uso del programa para ningún fin como, por ejemplo, el comercial o el militar.” (Adell & Bernabè, s.f. p. 6).
- **“Distribución de la licencia.** Los derechos del programa deben aplicarse a todos quienes se redistribuye el programa sin ninguna condición adicional.” (Adell & Bernabè,

s.f. p. 6).

- **“La licencia no debe ser específica de un producto.** Los derechos garantizados al usuario del programa no deben depender de que el programa forme parte de una distribución o paquete particular de software.” (Adell & Bernabè, s.f. p. 6).

2.1.2 Las licencias en el software privativo y libre

“Pero, ¿cómo se obtienen y garantizan las libertades en el software? ¿Cómo saber qué derechos tenemos o no tenemos respecto a un programa que hemos conseguido en una tienda o bajado de Internet? ¿Qué es la licencia? Las condiciones de uso del software por parte del usuario las proporciona la licencia, un documento legal que acompaña al software y que poca gente lee”. (Adell & Bernabè, s.f. p. 7)

“Se denomina acuerdo de licencia de software a un contrato entre un productor o propietario y un usuario de un programa de ordenador. Si el usuario es una persona, se le denomina usuario final. Un EULA o End User License Agreement (Acuerdo de Licencia de Usuario Final) es el contrato entre un propietario y un usuario final. El EULA especifica los límites de los permisos garantizados por el propietario a dicho usuario final, es decir, lo que podemos y no podemos hacer con el software adquirido.” (Adell & Bernabè, s.f. p. 7)

2.1.2.1 Licencias privativas

Según (Adell & Bernabé, s.f.).

“La lectura atenta de un EULA de cualquier programa comprado en una tienda es un ejercicio muy conveniente para entender qué nos ofrece el software libre frente al software privativo. Físicamente, los EULA son esos documentos que van dentro de las cajas del software que venden en las tiendas y que nunca nadie suele leer (tienen la letra muy pequeña y un estilo francamente farragoso). También pueden aparecer durante la instalación del programa en el ordenador: es una de esas pantallas en las que sale un fragmento de un texto

más largo, que se” (Adell & Bernabè, s.f. p. 7)

“os dice que leamos atentamente y que cliquemos sobre el botón Acepto si estamos de acuerdo. Veamos resumidamente qué dice un EULA típico.” (Adell & Bernabè, s.f. p. 7).

“Lo primero que sorprende es que no empieza con algo así como La parte contratante de la primera parte será considerada como la parte contratante de la primera parte. Lo que dice en conjunto es que, aunque tengamos esa impresión, en realidad no hemos comprado el software, sino que simplemente hemos adquirido el derecho a usarlo con numerosas restricciones, es decir, que el programa no es nuestro. En segundo lugar, ¡sorpresa! que al desprecintar la caja ya hemos aceptado todas sus condiciones sin haberlas leído siquiera. Lo que viene a continuación es, en esencia, una larga lista de prohibiciones para el usuario: no podemos instalar el software más que en un ordenador o en todo caso, solamente podemos ejecutar el software en un único ordenador a la vez (dependiendo del tipo de software), no podemos hacer más de una copia de seguridad, no podemos descompilarlo o intentar de ninguna manera ver cómo está hecho por dentro, no podemos alquilarlo, venderlo o prestarlo más que una sola vez a otro usuario final y, en tal caso, tenemos que destruir toda la copia de seguridad, la garantía es limitada, etc.” (Adell & Bernabè, s.f. pp. 7-8)

“En resumen, no tenemos ningún otro derecho que el de usar el programa en un ordenador determinado, tal y como es, y nos sometemos a todas las obligaciones, condiciones y prohibiciones imaginables, incluyendo el hecho de que usándolo autorizamos a la empresa propietaria del software a que reúna datos sobre nosotros mismos y nuestro ordenador cada vez que iniciamos o actualizamos el programa, sin preguntar siquiera qué nos parece la idea. Si encontramos algún fallo, no podemos arreglarlo. En el pasado, algunas licencias prohibían incluso publicar cualquier análisis del funcionamiento del software sin permiso previo del propietario (es, Ud. no

puede ni siquiera hablar de algo que es mío, mío y sólo mío). Increíble y seguramente ilegal en cualquier país con una constitución democrática, que reconozca la libertad de expresión de sus ciudadanos.” (Adell & Bernabè, s.f. p. 8)

2.1.2.2 Licencias libres

✓ **Licencias BSD**

“La licencia BSD (Berkeley Software Distribution), creada por la Universidad de Berkeley, es una de las menos restrictivas. Permite la redistribución y las modificaciones del software y no impone restricciones a cómo puede ser utilizado el código (por ejemplo, dentro de productos comerciales). Por eso, hay código licenciado BSD en productos comerciales, como Windows o en la parte no-libre de Mac OS X. Como casi única restricción, incluye la prohibición de usar el nombre del propietario de los derechos para la promoción de productos desarrollados a partir del original.” (Adell & Bernabè, s.f. p. 8)

✓ **La Licencia Pública General de GNU (GNU GPL)**

“La GNU General Public License es la licencia del proyecto GNU, propuesta por la Free Software Foundation y la más utilizada hoy en día en el software libre. Fue ideada por Richard Stallman para impedir que el software que se creara en el proyecto GNU pudiera convertirse en privativo y negar las libertades que reconoce a los usuarios. Se denomina copyleft y, genialmente, utiliza las leyes de copyright para otorgar libertades a los usuarios, no para reservar derechos a los propietarios, y asegurar que éstos lo harán del mismo modo si modifican el código del programa y lo redistribuyen. El propio Stallman cuenta la anécdota de la asignación del nombre a este tipo de licencia all rights reversed, en contraposición al all rights reserved del copyright, al relatar la historia del proyecto GNU (Stallman, 2002)” (Adell & Bernabè, s.f. pp. 8-9)

✓ **La Licencia Pública General Menor de GNU (GNU LGPL)**

“La Lesser GNU Public License o LGPL ha sido también creada para el proyecto GNU de la Free Software Foundation. Esta licencia, menos restrictiva, permite específicamente la integración con cualquier otro software sin prácticamente limitaciones” (Adell & Bernabè, s.f. p. 9)..

✓ **Licencias para la documentación del software**

“Además de establecer las condiciones de uso de los programas, también se puede regular la utilización de su documentación. Para ello, Richard Stallman propuso una licencia de documentación libre de GNU. Es la licencia copyleft de la Free Software Foundation para documentos de carácter técnico o pedagógico relacionados con los programas informáticos.” (Adell & Bernabè, s.f. p. 9)

✓ **Licencias Creative Commons**

“Otras licencias muy difundidas son las Creative Commons (Xalabarder, 2006), desarrolladas principalmente para licenciar recursos intelectuales de carácter literario, científico y/o artístico y que permiten a los autores mayor flexibilidad para definir las condiciones de uso de sus materiales que el copyright clásico, reservándose aquellos derechos que deseen (por ejemplo, permitiendo la reproducción, pero prohibiendo el uso comercial o la obra derivada). Las licencias Creative Commons han recibido críticas desde la FSF porque no aseguran la libertad de los usuarios.” (Adell & Bernabè, s.f. p. 9)

2.1.3 El origen del software libre

“Una de las mejores historias del software libre en castellano es, sin duda, la incluida en González Barahona, Seoane y Robles (2003). En inglés, puede consultarse Bretthauer (2001), DiBona et al. (1999) o Wayner (2000). Para una biografía de Richard Stallman, el principal ideólogo del movimiento, véase Williams (2002). En este apartado resumiremos los hechos e ideas esenciales para comprender el origen y la evolución del software libre hasta nuestros días.” (Adell & Bernabè, s.f. p. 10)

“En realidad, los orígenes del software libre son los del propio software: en los años sesenta, cuando los ordenadores eran máquinas enormes y costosas, el software era libre. Era considerado un complemento necesario para que funcionara el ordenador y algo que escribían los programadores de universidades, centros de investigación, oficinas gubernamentales, etc., que compartían entre ellos sin ningún problema y que modificaban tranquilamente. A nadie se le ocurría decir que algo era suyo: lo hacían un poco entre todos. En ningún caso se trataba como un producto con valor de mercado, por el que hubiera que pagar, sino información que libremente compartían sus usuarios y que éstos mejoraban de manera acumulativa, beneficiando a todos. La cultura hacker de los laboratorios de informática universitarios estadounidenses (Stanford, Berkeley, Carnegie Mellon, MIT, etc.), creada en los 60 y los 70 paralelamente del movimiento contracultural californiano, es el caldo de cultivo del software libre.” (Adell & Bernabè, s.f. p. 10)

“No fue hasta principio de los años setenta cuando IBM comenzó a vender separadamente sus máquinas y sus aplicaciones. Para protegerlas de posibles modificaciones, se comenzó a restringir la distribución del código fuente. De este modo, si los usuarios únicamente disponían del código máquina, las modificaciones eran prácticamente imposibles. La política de IBM se generalizó a medida que crecía la industria informática. A pesar de ello siguió existiendo el software libre. El caso más destacado es el desarrollo del sistema operativo Unix, que rápidamente se extendió por universidades y centros de investigación. Aunque la relación entre empresas y universidades en el desarrollo de Unix se vio comprometida por pleitos y querellas cuando las empresas se dieron cuenta que había beneficios a ganar. Este episodio, protagonizado por AT&T y la Universidad de Berkeley, dio lugar a la aparición de distintas versiones de Unix, unas libres, otras de pago, y a una fragmentación del mercado y una inseguridad jurídica de la que todo el mundo salió perdiendo.” (Adell & Bernabè, s.f. p. 10)

“El año 1984, Richard Stallman, un brillante programador de ordenadores del

Laboratorio de Inteligencia Artificial del MIT (Instituto de Tecnología de Massachussets) abandonó su trabajo, harto de la política de las empresas de software. Se cuenta que la gota que colmó el vaso fue la negativa de una empresa a proporcionarle acceso al código fuente de un programa que controlaba una impresora. Stallman quería repararlo porque funcionaba mal. En esa época ocurrieron dos hechos, menos anecdóticos, que condujeron a Stallman a la decisión de abandonar el MIT y crear software libre. El sistema operativo que había contribuido a crear con sus colegas se convirtió en trabajo inútil tras la compra de un nuevo ordenador por parte del laboratorio. La nueva máquina usaría un sistema privativo. Por la misma época, la mayoría de los hackers que trabajaban con Stallman abandonaron la universidad para formar una empresa privada. El paraíso hacker en el que vivía Stallman se estaba desmoronando ante sus ojos (Williams, 2002). A principios de 1984, Stallman dejó su trabajo en el MIT a fin de que la institución no interfiriera en sus planes: crear un sistema operativo completo, tipo UNIX. Lo llamó GNU (un acrónimo recursivo, esto es, que se refiere a sí mismo, algo muy del gusto de los programadores de inteligencia artificial, y que significa GNU No es Unix) y fundó la Free Software Foundation (FSF) para reunir los fondos necesarios” (Adell & Bernabè, s.f. p. 11)

“A principio de la década de los noventa la FSF tenía desarrolladas casi todas las piezas necesarias para que su sistema operativo fuera utilizable, excepto el kernel (Stallman, 1999). El kernel es el núcleo del sistema operativo de un ordenador. Es la pieza que hace posible la comunicación entre el resto del software y el hardware, una pieza esencial. Pero en esa época ocurren dos hechos que cambiarán radicalmente la historia del software libre. El primero es la popularización de la Internet, anteriormente confinada a las universidades y centros de investigación. Los programadores ya disponen de un medio para comunicarse, coordinarse y colaborar de un modo mucho más rápido, barato y eficiente que mandando cintas de ordenador por correo. El segundo es la aparición en escena de un joven estudiante finlandés de 21 años llamado Linus Torvalds, que, para aprender cómo funcionaba, había decidido crear un kernel

para ordenadores personales por sí mismo. La idea, sorprendentemente cuaja y -con la ayuda de numerosos voluntarios, que se comunicaban por Internet- se desarrolla un kernel, llamado Linux. De la unión entre las herramientas creadas por el proyecto GNU y Linux, nació el sistema GNU/Linux con licencia GPL. Era rápido, eficiente y, lo que es más importante, se podía modificar y compartir.” (Adell & Bernabè, s.f. p. 12)

“La aparición de la Open Source Initiative (OSI) en 1998 marca un punto de inflexión en la historia del software libre. Un grupo de programadores, liderados por Eric Raymond, Bruce Perens, Tim O'Reilly (un editor de libros técnicos), hartos del lenguaje mesiánico y revolucionario de Stallman, que asustaba a los empresarios y a los medios de comunicación, de la confusión entre libre y gratuito y, sobre todo, de la personalidad de Stallman, decidieron adoptar el término código abierto para referirse a lo mismo que Stallman llamaba software libre. Raymond diseñó una cuidadosa estrategia de comunicación que triunfó cuando Netscape, una empresa puntera en tecnología web, pero con graves problemas debido a la decisión de Microsoft de regalar preinstalado su navegador web, señaló su ensayo La catedral y el bazar como la inspiración de su decisión de liberar el código de Netscape. Raymond y Perens crearon la OSI para certificar que las licencias del software se ajustaban a la definición de software de código abierto elaborada por Perens en base a la licencia del proyecto Debian, una de las más famosas distribuciones GNU/Linux. A nivel de ideas, la OSI defiende la superioridad técnica, no moral, del software libre, producto del nuevo modelo de desarrollo creado por Linus Torvalds y la comunidad Linux y que Raymond había descrito en la metáfora de “La catedral y el bazar”. La “catedral” es Stallman y el proyecto GNU, incapaz de crear durante una serie de años un kernel utilizable, encerrado en su torre de marfil. El “bazar” es Linux, una comunidad de iguales, gestionada de manera abierta, descentralizada y poco jerárquica, a la que cualquiera puede contribuir con su código, sus propuestas, informes de errores, escribiendo documentación, localizando (traduciendo) software, etc.” (Adell & Bernabè, s.f. p. 12)

“El sector educativo mira atentamente el software libre como una alternativa real al software privativo, aunque sin decidirse totalmente. Riina Vuorikari, en un Informe Especial para la European Schoolnet (Vuorikari, 2004) ha destacado el escaso partido que las autoridades educativas nacionales europeas han sacado del potencial del software libre. Las razones, afirma, son diversas. Por una parte, la falta de conocimiento sobre el desarrollo del software libre dificulta su extensión en el sector educativo. Por otra, el hecho de que el debate sobre su potencial esté conducido por activistas y lobbistas, que exageran sus ventajas e inconvenientes, ha dejado poco espacio para análisis equilibrados y estudios rigurosos. Incluso algo tan sencillo de evaluar para las autoridades educativas como el coste total de propiedad del software libre y del privativo no ha sido estudiado hasta hace muy poco tiempo (BECTA, 2006), para comprobar que el software libre es más barato considerando todos los conceptos: adquisición, mantenimiento, instalación, soporte, formación, etc. A pesar de la indiferencia denunciada por Vuorikari en el año 2004, el paso ya lo han dado algunas administraciones educativas. Por ejemplo, en España las autoridades educativas de Extremadura, Andalucía, Valencia, Madrid, Castilla-La Mancha, y Galicia han desarrollado sus propias distribuciones, gnuLinEx, Guadalinux, Lliurex, Max, MoLinux y Trisquel, respectivamente, están dotando a todos sus centros educativos de ordenadores con GNU/Linux preinstalado y creando servicios y software libre para cubrir sus necesidades. En países como Brasil (Kim, 2005) o Venezuela ha habido iniciativas legislativas a favor del software libre.” (Adell & Bernabè, s.f. pp. 13-14)

2.2 El Software libre en educación

“Las razones por las que los partidarios del software libre y de código abierto defienden su superioridad sobre el privativo se pueden dividir en dos tipos esenciales, que se corresponden con los discursos de las dos principales corrientes de pensamiento que hemos descrito anteriormente. Mientras la OSI (Open Source Initiative), esto es, Eric Raymond, Tim O'Reilly, Bruce Perens

y otros conocidos desarrolladores como Linus Torvalds defienden el código abierto por razones pragmáticas, como su mayor calidad, su menor coste, mayor seguridad, estabilidad, eficiencia, integración, etc., la FSF (Free Software Foundation), esto es, Richard Stallman y otros conocidos desarrolladores y activistas, defienden su superioridad ética, social y política. En realidad, ambos conjuntos de argumentos son complementarios.” (Adell & Bernabè, s.f. p. 14)

“Algunos autores, conforme se han ido extendiendo las ideas sobre el software libre, han apuntado también argumentos que tienen en cuenta el contexto escolar, es decir, por qué es más adecuado o ventajoso usar software libre en educación, además de las razones genéricas que se aplican a cualquier usuario informático. Entre ellos merecen mención aparte los relacionados con la enseñanza de la informática y la ingeniería del software a nivel universitario, que tiene en el software libre un fenomenal recurso didáctico”. (Adell & Bernabè, s.f. p. 14)

2.2.1 Ventajas pragmáticas

“Las ventajas de índole práctica del software libre derivan de su modo de producción: redes distribuidas de iguales que colaboran voluntariamente por una amplia gama de motivaciones (Hars y Ou, 2001; III-UM and Berlecom Research, 2002). Raymond (2004) lo explica de manera simple: los programadores de código abierto han aprendido que el secreto es enemigo de la calidad. La manera más efectiva de conseguir fiabilidad en el software es publicar el código para que lo revisen otros programadores y no-programadores expertos en el área de aplicación del software. Raymond lo resume en una frase, la Ley de Linus (Torvalds): si suficientes globos oculares miran el código, los errores serán evidentes”. Puede parecer increíble, pero en el desarrollo de Linux han participado, en mayor o menor medida, más de 750.000 programadores de todo el mundo (Raymond, 2004)” (Adell & Bernabè, s.f. pp. 14-15).

“No hay empresa que pueda competir con eso. La Internet y GNU/Linux son dos ejemplos notables de que el modelo de código abierto y libre colaboración entre programadores funciona” (Adell & Bernabè, s.f. p. 15).

“Por tanto, para el movimiento del código abierto, el software desarrollado siguiendo el modo de producción colaborativa entre iguales es de mayor calidad, ofrece mayor seguridad, más estabilidad a lo largo del tiempo (el código no desaparece si una empresa cierra o es comprada por otra), los tiempos de desarrollo son menores y los proyectos interesantes crecen a una enorme velocidad (al ser la programación una tarea altamente paralelizable, especialmente la parte más tediosa: la detección de errores) y tiene un precio sencillamente inigualable. Es más, el concepto de software como servicio y no como producto favorece a la industria local en lugar de contribuir a la creación de monopolios y beneficia a más programadores que el modelo privativo. El software de código abierto es más fácilmente localizable, un tema de la mayor importancia en educación, sobre todo para lenguas minoritarias o minorizadas (Mas, 2003), es más accesible y desarrollar software libre la mejor manera de usar los fondos públicos para potenciar la industria del software local, en lugar de pagar royalties por el software privativo, que van a parar a grandes empresas extranjeras.” (Adell & Bernabè, s.f. p. 15)

2.2.2 Razones políticas, éticas y sociales

“Pero hay más razones, además de las prácticas o técnicas para escoger software libre frente a privativo. La mayor parte de las personas cree que el software vive en el interior de los ordenadores y que su relación con él empieza cuando encienden su ordenador y termina cuando lo apagan. Es una idea tranquilizadora: nos da una falsa sensación de control, de estar al mando.” (Adell & Bernabè, s.f. p. 16)

“Basta apagar el ordenador para que el software se duerma hasta que lo volvamos a necesitar. ¿Qué mal puede hacernos? Lo único que les parece

importante es si funciona bien o no, si cumple su misión. Reflexionemos un momento sobre la importancia del software en nuestra sociedad. Aunque no hayamos usado nunca un ordenador, nuestra vida entera depende del software: ¿cómo se calcula nuestro salario?, ¿cómo lo cobramos?, ¿dónde está nuestro dinero en el banco?, ¿qué pasa con los cajeros automáticos cuando no funciona la red informática que los une al banco?,” (Adell & Bernabè, s.f. p. 16)

“¿y la información sobre nuestra salud?, ¿cómo se gestiona la red eléctrica de nuestra ciudad?, ¿y la de gas o agua?, ¿qué pasa cuando en el supermercado un producto no tiene el código de barras?, ¿quién controla el encendido y los frenos ABS de nuestro coche?, ¿cómo calculan las autoridades los impuestos que debemos pagar?, ¿dónde están anotadas las calificaciones de nuestras asignaturas en la universidad?. (Adell & Bernabè, s.f. p. 16)

“No hace falta seguir: es evidente que las tecnologías de la información y el software que hace que funcionen o no de determinada manera son demasiado importantes en nuestras vidas para que no sepamos qué hacen realmente o para que se comporte de manera que nos aten de por vida a intereses comerciales. El software conforma la estructura de la comunicación y la información en una sociedad post- industrial cuyo mayor factor de producción es el conocimiento, define cómo podemos trabajar, comunicarnos, divertirnos o relacionarnos con nuestros vecinos o parientes. Algunos autores han sugerido que las nuevas tecnologías están conformando nuevos tipos de procesos cognitivos en los jóvenes, una nueva manera de procesar la información determinada por su extensa práctica con nuevos tipos de medios, lenguajes y modelos de comunicación (el hipertexto y el multimedia, la interactividad de los videojuegos, la instantaneidad de los teléfonos móviles y la Internet, etc.). El cambio es de tal magnitud que se habla de nativos e inmigrantes digitales”. (Adell & Bernabè, s.f. p. 17)

“Por otra parte, el software libre promueve la cooperación entre las personas donde el software privativo la convierte en un delito. Y la cooperación es un valor fundamental de nuestra sociedad al que la escuela debe prestar

especial atención.” (Adell & Bernabè, s.f. p. 17).

2.2.3 El software libre en la educación

“Richard Stallman (2003) ha escrito un texto sobre las razones por las que las escuelas deberían utilizar exclusivamente software libre. El software libre, recuerda Stallman, permite que los usuarios controlen lo que hacen sus ordenadores y cooperen entre ellos. Las dos razones son también válidas para la educación. Pero hay razones netamente “educativas” (Adell & Bernabè, s.f. p. 17):

- ✓ “La primera es que el software libre se puede copiar y redistribuir a precio de coste. La Administración educativa puede dotar de software a todos sus centros docentes a muy bajo precio y dedicar los recursos ahorrados a otros temas necesarios para la educación: más ordenadores, formación del profesorado, desarrollo de software libre educativo, etc. En los países menos desarrollados, el software libre puede ayudar a dotar de infraestructura tecnológica a sus escuelas y a paliar la brecha digital con el mundo” (Adell & Bernabè, s.f. p. 17)

“desarrollado. Los vendedores de software privativo, que saben de la importancia de la educación para sus futuras ventas, pueden ofrecer software a muy bajo coste o gratuito a las escuelas. Pero se trata en realidad de una estrategia comercial para captar futuros clientes y para formarlos en sus productos a costa del erario público.” (Adell & Bernabè, s.f. p. 18)

- ✓ “La escuela ha de enseñar a los estudiantes valores y estilos de vida que beneficien a toda la sociedad. La escuela ha de promover el uso de software libre por la misma razón que promueve el reciclaje: porque nos beneficia a todos. Si los estudiantes usan el software libre y aprenden que es mejor que el privativo, cuando sean adultos seguirán usando el software libre. Eso permitirá a la sociedad liberarse de los abusos y del

control de las multinacionales que controlan el software privativo.”
(Adell & Bernabè, s.f. p. 18)

- ✓ “El software libre favorece que los estudiantes aprendan cómo funcionan los ordenadores y el propio software. Los futuros programadores se inician en la programación durante la adolescencia. Es una etapa clave en la que necesitan buenos modelos y ejemplos para modificar, copiar y jugar con ellos. Necesitan desafíos. El software libre, al permitir el acceso al código fuente del programa, les facilita enormemente el aprendizaje. El software privativo es una caja negra que no aporta nada para satisfacer su curiosidad y sus ansias de saber. El mensaje que les envía el software privativo es el conocimiento es una mercancía, lo que quieres saber es un secreto comercial, aprender está prohibido por la ley. El software privativo mantiene a la gente alejada del conocimiento, sacraliza la tecnología y contribuye interesadamente a la ignorancia tecnológica que tan buenos resultados económicos les proporciona a las empresas que lo comercializan.” (Adell & Bernabè, s.f. p. 18)

- ✓ “Pero, aunque muchos adolescentes no sientan curiosidad por cómo están hechos los programas de ordenador, hay valores generales que persigue la educación que están en claro conflicto con el mensaje que transmite el software privativo. Las escuelas deben enseñar hechos, conceptos, principios y procedimientos, pero también valores. La misión de la escuela es enseñar a las personas a ser buenos ciudadanos, a cooperar con los demás, a ser solidarios. Esta es la base de la sociedad. En informática, cooperar significa, entre otras cosas, compartir software, poder hacer copias a todos los compañeros de clase, llevarse a casa el software que se usa en la escuela. Y todo eso, con el software privativo es un delito.” (Adell & Bernabè, s.f. p. 18)

- ✓ . “Finalmente, enseñar a los estudiantes a usar software libre y a

participar en la comunidad de usuarios/desarrolladores de software libre es una lección cívica llevada a la práctica. También enseña a los estudiantes que el ideal es el modelo de servicio público y la solidaridad, no el modelo del beneficio a cualquier precio de los magnates. Todos los niveles pueden y deben usar software libre” (Adell & Bernabè, s.f. p. 18).

2.2.4 El software libre y la innovación en tecnología educativa

“Graham Atwell (2005) ha puesto de manifiesto un hecho diferencial del software libre en la educación que no podemos dejar de señalar: su maridaje con la innovación educativa. Las razones son diversas. En primer lugar, en los proyectos de software libre el coste inicial es muy bajo: suelen ser personales o de un pequeño grupo de entusiastas. En segundo lugar, se puede construir sobre el trabajo de otros proyectos y explorar sus aplicaciones educativas (por ejemplo, integrando herramientas que originalmente no fueron diseñadas con propósito educativo, como blogs y wikis). Si el proyecto cuaja, porque la gente lo encuentra de interés, es fácil abrirlo a la colaboración. Un ejemplo de este proceso es Moodle, una plataforma de enseñanza basada en presupuestos socio-constructivistas del aprendizaje que ha sobrepasado en funcionalidades e implantación a sus alternativas privativas y que se ha hecho tremendamente popular en el último año. Iniciado por una sola persona, Martin Dougiamas, que, descontento por cómo estaba diseñado y funcionaba el software privativo equivalente de su universidad, se hizo una plataforma (realmente modesta en sus inicios) para sus clases. Hoy, la comunidad Moodle está formada por decenas de desarrolladores, miles de usuarios, sus instalaciones se cuentan por millares y varios millones de estudiantes y profesores utilizan Moodle en sus clases presenciales, semi-presenciales o a distancia.” (Adell & Bernabè, s.f. pp. 20-21)

2.3 Software FACIL

Según Tàrraga & Fernández, (s.f.) refieren que:

“La Factoría de Actividades Informática y Logopedia (FACIL), es un programa informático de autor de uso gratuito elaborado y distribuido por el portal de la Generalitat de Catalunya xtec.” (Tarraga, s.f. pàrr. 3)

“El programa está dirigido a logopedas y maestros de educación especial, aunque dado su carácter abierto, que permite al usuario ser quien seleccione los contenidos a trabajar, puede ser empleado por maestros generalistas en educación infantil y en los primeros cursos de educación primaria. En el presente artículo se expone el contenido de este programa, y se proporcionan algunas ideas para su uso en el aula.” (Tarraga & Inmaculada, s.f. p. 1)

2.3.1 Funcionamiento del programa

“La Factoría de Actividades Combinadas de Informática y Logopedia (FACIL), es una base de datos de uso y distribución gratuita, preparada para la elaboración de actividades educativas encaminadas fundamentalmente al trabajo logopédico y de educación especial, descargable gratuitamente en la dirección” (Tarraga & Inmaculada, s.f. p. 2)

“El programa tiene dos versiones, una en castellano y otra en catalán, aunque no es recomendable instalar ambas en el mismo equipo informático. FACIL es, en esencia, una base de datos creada con el programa Microsoft Acces (por lo que se requerirá tener instalado en el equipo este programa). En esta base de datos hay 3 elementos fundamentales: imágenes similares a los iconos de SPC (objetos, animales, acciones, lugares), archivos de audio en los que se pronuncia el nombre de las imágenes mencionadas anteriormente, y finalmente la transcripción fonética y gráfica de las palabras correspondientes a las imágenes y a los archivos de audio. Los tres tipos de elementos están relacionados en la base de datos, de modo que se combinan adecuadamente para crear los diversos tipos de actividades que el programa permite generar,

sin necesidad de que el usuario intervenga en la creación de relaciones entre los elementos.” (Tarraga & Inmaculada, s.f. p. 2)

“El programa base posee más de 800 elementos (más de 800 imágenes con sus correspondientes audios y transcripciones), aunque una de sus potencialidades es que permite al usuario introducir, sin demasiada dificultad, nuevos elementos a la base de datos, de modo que a las imágenes existentes se pueden añadir otras nuevas, como fotografías de profesores, alumnos, elementos de la clase o del centro, etc., lo que eleva exponencialmente las aplicaciones del programa, y permite el diseño de actividades realmente individualizadas.” (Tarraga & Inmaculada, s.f. p. 2)

“Para diseñar actividades, el procedimiento a seguir es siempre el mismo: primer lugar es necesario seleccionar las palabras a trabajar. Para ello, las palabras están ordenadas por diferentes criterios: búsqueda según categorías semánticas, juegos de posición de letras (una opción muy útil para el trabajo logopédico), número de letras, número de sílabas, sílaba inicial, sílaba final, y transcripción fonética de la palabra. El menú para seleccionar palabras es similar al siguiente” (Tarraga & Inmaculada, s.f. p. 2)

“Una vez se han seleccionado los elementos con los que se desea trabajar, el único paso que queda por dar es pasar a seleccionar el tipo de actividad que se desea crear. Para ello, en él se volverá al menú inicial, en el que aparecen todos los tipos de actividades que el programa permite crear, se pulsará sobre el icono deseado, y el programa creará automáticamente las actividades correspondientes” (Tarraga & Inmaculada, s.f. p. 3)

“Algunas de estas actividades se ejecutarán automáticamente, por lo que simplemente habrá que lanzarse a realizarlas con el alumno, mientras que otras actividades se almacenarán automáticamente en el directorio: C/PRG_EDUC/fácil (PRG_EDUC es una carpeta que el programa crea

automáticamente en el directorio C cuando se instala el programa).” (Tarraga & Inmaculada, s.f. p. 4)

2.3.2 2. Actividades que el programa puede generar

“FACIL genera automáticamente actividades educativas dirigidas fundamentalmente a trabajar aspectos relacionados con adquisición de vocabulario, inicio de la lectoescritura, y trabajo logopédico” (Tarraga & Inmaculada, s.f. p. 4).

“Estas actividades pueden dividirse en dos grandes grupos: actividades en soporte de papel (creadas automáticamente con el procesador de textos Microsoft Word, por lo que este programa también será necesario si se desea usar estas actividades), y actividades en soporte informático, creadas en diferentes programas que FACIL instala automáticamente en el ordenador” (Tarraga & Inmaculada, s.f. p. 4)

“Las actividades en papel pueden ser juegos de lengua como lotos, dominós, juegos de la oca, juegos de cartas, o dados, en los que se puede combinar texto (con letra ligada o de palo), con imágenes, según la utilidad que quiera darse a la actividad.” (Tarraga & Inmaculada, s.f. p. 4)

“Otra posibilidad es crear actividades más directamente ligadas a la enseñanza de la lectoescritura. En este caso, FACIL realiza actividades de identificación de letras o sílabas en palabras, dictados, emparejamiento de palabras con dibujos, creación de dibujos a partir de palabras, escritura a partir de imágenes, etc.” (Tarraga & Inmaculada, s.f. p. 5)

2.4 Lectoescritura

2.4.1 La importancia de aprender a leer y escribir

Según Romero, L. (s.f.) refiere que:

“Nos hemos acostumbrado a pensar que la lectura y escritura son aprendizajes mecánicos y puramente instrumentales, pero en realidad son aprendizajes fundamentales cuya transferencia cognitiva y afectiva va mucho más allá de lo que podríamos imaginar; por algo, a nivel universal, se consideran tres aprendizajes esenciales para la vida: la lectura, la escritura y el pensamiento lógico-matemático.” (Villacastin, s.f. p. 2)

“Estas habilidades son herramientas importantes para desarrollar niveles cada vez más elaborados de pensamiento, comunicación e interacción positiva con los demás y con el medio, además de que son instrumentos muy valiosos para aprender, seguir estudiando y seguir aprendiendo. En el mundo actual la capacidad para aprender a lo largo de toda la vida es no sólo una necesidad, por la velocidad con que avanza la ciencia y la tecnología, sino que es también un derecho de todos, que tiene que ver con la equidad que tanto necesitamos para acortar las brechas que existen en nuestro país. Garantizar estos aprendizajes en todos los alumnos de nuestras escuelas se convierte en un compromiso sociopolítico y ético primordial, que estamos en obligación de cumplir.” (Villacastin, s.f. p. 1)

“LA LECTURA, entre otras cosas, es importante para” (Villacastin, s.f. p. 2):

- “Desarrollar la atención y concentración” (Villacastin, s.f. p. 2).
- “Desarrollar la capacidad de decodificación y comprensión de mensajes” (Villacastin, s.f. p. 2).
- “Ayudar al descentramiento, es decir, salir de sí para ponerse en el lugar del otro” (Villacastin, s.f. p. 2).
- “actitud dialógica para escuchar y aprender de los demás.” (Villacastin, s.f. p. 2).
- “Ponernos en contacto con nuestra propia interioridad ya que es, por lo general, una actividad personal, silenciosa, que confronta

permanentemente otras ideas y sentimientos con las de uno mismo” (Villacastin, s.f. p. 2).

- “Ser, además, un vehículo indispensable para ponerse en contacto con el mundo, con el conocimiento, otras realidades, con el pasado, diversas opiniones, etc. No todo lo podemos ver y comprobar por nosotros mismos, gran parte de información nos llega por escrito (periódicos, revistas, libros, internet)” (Villacastin, s.f. p. 2).
- “Ponerse en contacto con todo lo anterior, permitiendo al lector ampliar su panorama y desarrollar criterios para comprender e interpretar su propia realidad” (Villacastin, s.f. p. 3).
- “Estar actualizados, en esta época en que los avances científicos y tecnológicos avanzan rápidamente, quien no lee queda desfasado” (Villacastin, s.f. p. 3).

LA REDACCIÓN favorece:

- “La organización y estructuración del pensamiento” (Villacastin, s.f. p. 3).
- “La actitud dialógica. Escribimos principalmente para ser leídos por otros, para comunicar algo” (Villacastin, s.f. p. 3).
- “El sentido lógico” (Villacastin, s.f. p. 3).
- “La capacidad de argumentación” (Villacastin, s.f. p. 3).
- “Niveles de expresión más elaborados” (Villacastin, s.f. p. 3).

2.4.2 Cómo enseñar a leer y escribir

2.4.2.1 Cuando se debe empezar la enseñanza sistemática de

Lectoescritura

Según Romero, L. (s.f.) refiere que:

“Todos sabemos que los niños aprenden desde muy pequeños a “leer” signos y símbolos que van adquiriendo cierta significación, como: logos comerciales, señales de tránsito, símbolos convencionales y otros. Estas son

primeras experiencias de lectura, sin embargo, el aprendizaje sistemático requiere del desarrollo previo de diversas habilidades que deben conjugarse.” (Centro de recursos alboan, s.f. p. 21)

“El aprendizaje de la lecto-escritura es un proceso de carácter complejo ya que su dominio no se agota en la tarea mecánica de codificación y decodificación, tal como ya se ha señalado. El proceso requiere que el niño haya alcanzado determinados niveles de maduración con respecto a tres factores que intervienen, estos son: desarrollo de la psicomotricidad, de la función simbólica y de la afectividad. La primera se refiere a la maduración general del sistema nervioso, expresada por la capacidad de desplegar un conjunto de actividades motrices; la segunda, a la maduración del pensamiento en su función simbólica, como para comprender, o al menos sentir, que la escritura conlleva un sentido y transmite un mensaje, lo que requiere también de un determinado nivel de desarrollo del lenguaje; y la tercera se refiere a la madurez emocional que le permita no desalentarse ni frustrarse ante el esfuerzo desplegado para lograr los automatismos correspondientes a esas primeras etapas.” (Centro de recursos alboan, s.f. p. 21)

2.4.2.2 Es necesario el aprestamiento para la lectoescritura

Según Romero, L. (s.f.):

“Se entiende por “aprestamiento” a los ejercicios preparatorios para la lecto- escritura. Este período es fundamental y necesario para aprender a leer y escribir en un proceso fluido y exitoso. Ya que hemos dedicado todo un documento al tratamiento del tema aquí mencionaremos rápidamente los aspectos más importantes” (Como se citò en Centro de recursos alboan, s.f. p. 22):

a) “Desarrollo del movimiento, que debe llevar al niño desde la exploración hasta el control del ambiente” (Centro de recursos

alboan, s.f. p. 22).

- b) “Psicomotricidad integral progresiva, empezando por los segmentos más gruesos que son aquellos que se encuentran más cerca del tronco (brazos) hasta llegar a los extremos (mano, dedos)” (Centro de recursos alboan, s.f. p. 22)”.
- c) “Estimular las funciones básicas: esquema corporal, orientación espacial y temporal, percepción táctil, visual y auditiva, memoria y lenguaje (expresivo y comprensivo)” (Centro de recursos alboan, s.f. p. 22).
- d) “Contenidos relativos a la escritura (formas básicas, vocales)” (Centro de recursos alboan, s.f. p. 22).
- e) “Lograr trazos finos organizados en sistemas coordinados y precisos” (Centro de recursos alboan, s.f. p. 22).

2.4.2.3 Qué aprendizajes debe lograr el niño en lectoescritura

Según Romero, L. (s.f.) refiere que:

“En primer lugar, debemos tener en cuenta que el aprendizaje de la lecto- escritura supone dos etapas: aprender a escribir y escribir en forma significativa, o su equivalente para la lectura, aprender a decodificar y leer comprensivamente.” (Centro de recursos alboan, s.f. p. 25).

“En el APRENDER A ESCRIBIR el niño debe aprender: las grafías y su trazo una por una, el mecanismo del ligado y obtener regularidad en la escritura. En el APRENDER A DECODIFICAR LA LECTURA, la correspondencia grafía-fonema, aprender el mecanismo de articulación de fonemas y palabras y obtener fluidez en la lectura. Estos elementos y sus desagregados (que presentamos a continuación), son aspectos a los que con frecuencia no prestamos mucha atención en el proceso de enseñanza, esperamos que los niños lo logren, pero no los ejercitamos con intencionalidad. Cada uno de ellos debe ser motivo de aprendizaje, hay que diseñar actividades para mostrarlos a los niños y que los practiquen; además sirven para evaluar el proceso de aprendizaje y orientarlo mejor.” (Centro de recursos alboan, s.f. p.

25)

A. Para la escritura

✓ En el aprendizaje a escribir

- Conocimiento de grafías y dominio de su trazo:

- “Reconoce cada grafía y la relaciona con su correspondiente fonema.” (Centro de recursos alboan, p. 25).
- “Sigue la dirección correcta.” (Centro de recursos alboan, s.f. p. 25).
- “Identifica y traza con claridad sobre la línea de base, en la zona media, superior e inferior (hace elevaciones y depreciaciones correctas).” (Centro de recursos alboan, s.f. p. 25).
- “Hay secuencia en el trazo (toda la letra la traza de una vez, sin hacer cortes; posteriormente toda la palabra)” (Centro de recursos alboan, s.f. p. 25).

- Aprender el mecanismo del ligado

- “Soltura y flexibilidad del movimiento izquierda-derecha” (Centro de recursos alboan, s.f. p. 25).
- “Continuidad en la articulación de las letras de una palabra.” (Centro de recursos alboan, s.f. p. 25).
- “Rapidez y dinamismo en la escritura” (Centro de recursos alboan, s.f. p. 25).

- Regularidad de la escritura:

- “Alineación, grafías alineadas unas con otras” (Centro de recursos alboan, s.f. p. 25).
- “Proporción y tamaño, de todas las letras y de los trazos

- que se hacen en las zonas superior e inferior.” (Centro de recursos alboan, s.f. p. 25).
- “Inclinación, debe existir regularidad en la inclinación” (Centro de recursos alboan, s.f. p. 25).
 - “Espaciado, espacios regulares para separar las unidades: letra y letra, palabra y palabra.” (Centro de recursos alboan, s.f. p. 25).

B. Para la lectura

✓ **En el aprender a codificar**

- Identificar la correspondencia fonema-grafía

- “Mecanismo de articulación de fonemas y palabras” (Centro de recursos alboan, s.f. p. 26).

- “Pronunciación clara de todos los fonemas, respetando los puntos de articulación (en el aparato fonológico)” (Centro de recursos alboan, s.f. p. 26).
- “Modulación del tono de voz (al contexto y al sentido del texto)” (Centro de recursos alboan, s.f. p. 26).
- “Respeto de los signos de puntuación (de estructuración y de expresividad)” (Centro de recursos alboan, s.f. p. 26).

CONCLUSIONES

PRIMERA. – El uso de tecnologías como los softwares educativos libres son una ayuda muy importante para el proceso de enseñanza-aprendizaje de los niños del nivel inicial, ya que nos brindan muchas ventajas sobre las herramientas de enseñanza tradicionales, mas aun en un mundo donde los niños son nativos digitales; pero sin embargo no son insustituibles, sino mas bien son un complemento.

SEGUNDA. – Al ser los niños de hoy nativos digitales, el uso del computador como herramienta para su aprendizaje, es la mejor motivación que pueden tener, pero al mismo tiempo el docente debe de tener la suficiente capacidad para q esta herramienta no se vuelva distorsionador de su proceso de aprendizaje.

REFERENCIAS CITADAS

Adell, J., & Bernabè, I. (s.f.). *Software libre educacion*. Castellon: universidad jaume i castellon.

Centro de recursos alboan. (s.f.). *El aprendizaje de la lecto-escritura*. Centro de recursos alboan.

Tarraga, M. (s.f.). *Sindrome del savant*. Obtenido de quadernsdigitals:
http://www.quadernsdigitals.net/index.php?accionMenu=autores.VisualizaAutorIU.visualiza&autor_id=12885&PHPSESSID=93e8f54e2de61beface5961021e84c2f

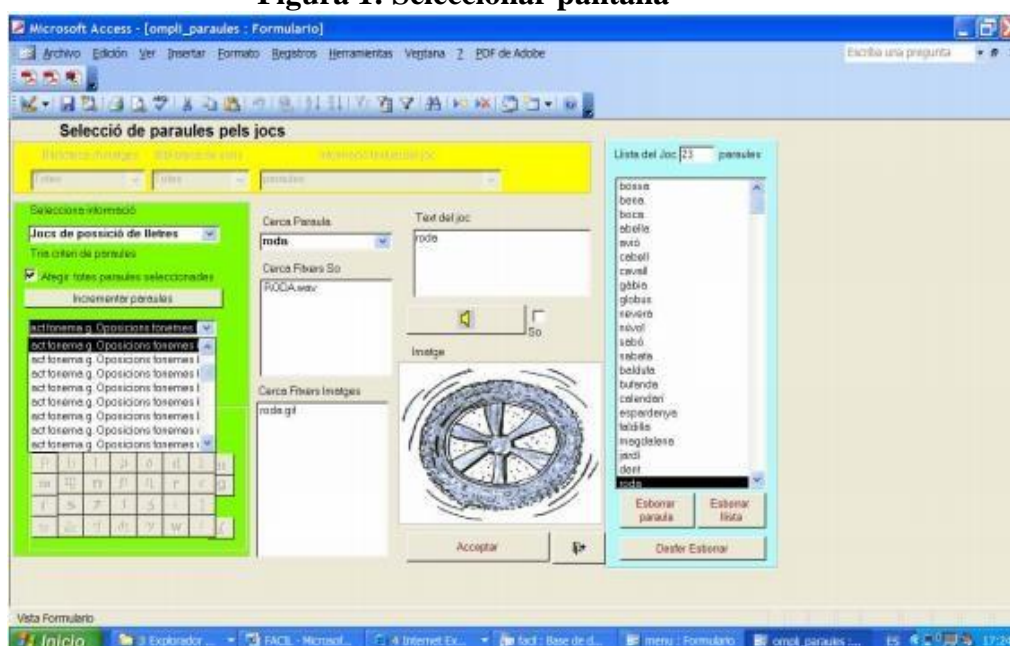
Tarraga, R., & Inmaculada, A. (s.f.). *Facil - quaderns digitals*. Obtenido de Study lib: <https://studylib.es/doc/4798220/facil---quaderns-digitals>

Villacastin, A. (s.f.). *El aprendizaje de la lectura y escritura en la escuela infantil*. Madrid: Universidad complutense de madrid.

ANEXOS

ANEXO 1



















Figura 1: Seleccionar pantalla



Fuente: file:///C:/Users/ACER/Downloads/archivoPDF.pdf

ANEZO 2

Figura 2: Selección de actividad

Relacions		Relacions		Tipus i formes			
escola		escola		obella		agulla	
escola		escola		agulla		ampolla	
escola		escola		aranya		armari	
escola		escola					

Fuente: file:///C:/Users/ACER/Downloads/archivoPDF.pdf

El software educativo libre en el aprendizaje de la lectoescritura de los niños de 4 años de edad

INFORME DE ORIGINALIDAD

10%	10%	0%	8%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	www.researchgate.net Fuente de Internet	3%
2	es.scribd.com Fuente de Internet	2%
3	www.scribd.com Fuente de Internet	1%
4	repositorio.upp.edu.pe Fuente de Internet	1%
5	Submitted to Universidad Alas Peruanas Trabajo del estudiante	1%
6	es.cyclopaedia.net Fuente de Internet	<1%
7	repositorio.ucv.edu.pe Fuente de Internet	<1%
8	repositorio.undac.edu.pe Fuente de Internet	<1%

9 **es.slideshare.net** <1%
Fuente de Internet

10 **Submitted to UNIV DE LAS AMERICAS** <1%
Trabajo del estudiante

11 **tesisenxarxa.net** <1%
Fuente de Internet

Excluir citas Activo

Excluir bibliografía Activo

Excluir coincidencias < 15 words