

**Novática**, revista fundada en 1975 y decana de la prensa informática española, es el órgano oficial de expresión y formación continua de **ATI** (Asociación de Técnicos de Informática), organización que edita también la revista **REICIS** (Revista Española de Innovación, Calidad e Ingeniería del Software).

<<http://www.ati.es/novatica/>>  
<<http://www.ati.es/reicis/>>

ATI es miembro fundador de **CEPIS** (Council of European Professional Informatics Societies) y es representante de España en **IFIP** (International Federation for Information Processing); tiene un acuerdo de colaboración con **ACM** (Association for Computing Machinery), así como acuerdos de vinculación o colaboración con **AdaSpain**, **A12**, **ASTIC**, **RITSi** e **HispaLinux**, junto a la que participa en **Prolnnova**.

**Consejo Editorial**  
Ignacio Aguiló Sousa, Guillem Alsina González, María José Escalona Cuaresma, Rafael Fernández Calvo (presidente del Consejo), Jaime Fernández Martínez, Luis Fernández Sanz, Didac López Viñas, Celestino Martín Alonso, José Onofre Montes Andrés, Francesc Noguera Puig, Ignacio Pérez Martínez, Andrés Pérez Payeras, Víticu Pons i Colomer, Juan Carlos Vigo López

**Coordinación Editorial**  
Llorenç Pagés Casas <[lpages@ati.es](mailto:lpages@ati.es)>  
**Composición y autoedición**  
Jorge Llácer Gil de Ramales  
**Traducciones**  
Grupo de Lengua e Informática de ATI <<http://www.ati.es/gt/lengua-informatica/>>  
**Administración**  
Tomás Brunete, María José Fernández, Enric Camarero, Felicidad López

**Secciones Técnicas - Coordinadores**  
**Acceso y recuperación de la información**  
José María Gómez Hidalgo (Optenel) <[jmgomez@mahuel.es](mailto:jmgomez@mahuel.es)>  
Manuel J. María López (Universidad de Huelva) <[manuel.mana@diesia.uhu.es](mailto:manuel.mana@diesia.uhu.es)>  
**Administración Pública electrónica**  
Francisco López Crespo (MAE) <[flc@ati.es](mailto:flc@ati.es)>  
Sebastià Justicia Pérez (Diputación de Barcelona) <[sjusticia@ati.es](mailto:sjusticia@ati.es)>  
**Arquitecturas**  
Enrique F. Torres Moreno (Universidad de Zaragoza) <[enrique.torres@unizar.es](mailto:enrique.torres@unizar.es)>  
José Filichardo (Universidad Politécnica de Valencia) <[jfilich@disca.upv.es](mailto:jfilich@disca.upv.es)>

**Auditoría SITIC**  
Marina Touriño Trolitio <[marinatourino@marinatourino.com](mailto:marinatourino@marinatourino.com)>  
Manuel Palao García-Suelto (ATI) <[manuel@palao.com](mailto:manuel@palao.com)>  
**Derecho y tecnologías**  
Isabel Hernando Collazos (Fac. Derecho de Donostia, UPV) <[isabel.hernando@ehu.es](mailto:isabel.hernando@ehu.es)>  
Elena Davara Fernández de Marcos (Davara & Davara) <[edavara@davara.com](mailto:edavara@davara.com)>  
**Enseñanza Universitaria de la Informática**  
Cristóbal Pareja Flores (DSIP-UCM) <[cpareja@dsip.ucm.es](mailto:cpareja@dsip.ucm.es)>  
J. Ángel Velázquez Irbide (DLSI, URJC) <[angel.velazquez@urjc.es](mailto:angel.velazquez@urjc.es)>

**Entorno digital persona**  
Andrés Martín López (Univ. Carlos III) <[amarin@it.uc3m.es](mailto:amarin@it.uc3m.es)>  
Diego Gachet Pérez (Universidad Europea de Madrid) <[gachet@ueem.es](mailto:gachet@ueem.es)>  
**Estandares Web**  
Encarna Quesada Ruiz (Virati) <[encarna.quesada@virati.com](mailto:encarna.quesada@virati.com)>  
José Carlos del Arco Prieto (TCP Sistemas e Ingeniería) <[jcarco@gmail.com](mailto:jcarco@gmail.com)>  
**Gestión del Conocimiento**  
Joan Baiget Solé (Cap Gemini Ernst & Young) <[joan.baiget@ati.es](mailto:joan.baiget@ati.es)>

**Informática y Filosofía**  
José Ángel Olivares Valls (Escuela Superior de Informática, UCLM) <[josangel.olivares@uclm.es](mailto:josangel.olivares@uclm.es)>  
Roberto Feltrero Oreja (UNED) <[rfeltrero@gmail.com](mailto:rfeltrero@gmail.com)>  
**Informática Gráfica**  
Miguel Chover Selles (Universitat Jaume I de Castellón) <[mchover@lsi.uji.es](mailto:mchover@lsi.uji.es)>  
Roberto Vito Hernández (Eurographics, sección española) <[rvito@dsic.upv.es](mailto:rvito@dsic.upv.es)>  
**Ingeniería del Software**  
Javier Dolado Cosín (DLSI-UPV) <[dolado@si.ehu.es](mailto:dolado@si.ehu.es)>  
Daniel Rodríguez García (Universidad de Alcalá) <[daniel.rodriguez@uah.es](mailto:daniel.rodriguez@uah.es)>

**Inteligencia Artificial**  
Vicente Boti Navarro, Vicente Julián Inglada (DSIC-UPV) <[vboti,vinglada@dsic.upv.es](mailto:(vboti,vinglada)@dsic.upv.es)>  
**Interacción Persona-Computador**  
Pedro M. Latore Andrés (Universidad de Zaragoza, AIP) <[platore@unizar.es](mailto:platore@unizar.es)>  
Francisco L. Gutiérrez Vela (Universidad de Granada, AIP) <[fgutierre@ugr.es](mailto:fgutierre@ugr.es)>  
**Lengua e Informática**  
M. del Carmen Ugarte García (ATI) <[cugarte@ati.es](mailto:cugarte@ati.es)>  
**Lenguajes Informáticos**  
Oscar Belmonte Fernández (Univ. Jaime I de Castellón) <[belferm@lsi.uji.es](mailto:belferm@lsi.uji.es)>  
Inmaculada Coma Tatay (Univ. de Valencia) <[inmaculada.coma@uv.es](mailto:inmaculada.coma@uv.es)>

**Lingüística computacional**  
Xavier Gómez Guinovart (Univ. de Vigo) <[xgg@uvigo.es](mailto:xgg@uvigo.es)>  
Manuel Patomer (Univ. de Alicante) <[mipatomer@dlsi.ua.es](mailto:mipatomer@dlsi.ua.es)>  
**Mundo estudiantil y jóvenes profesionales**  
Federico G. Mon Trotti (RITSi) <[gmon.trote@gmail.com](mailto:gmon.trote@gmail.com)>  
Mikel Salazar Peña (Asociación Jóvenes Profesionales, Junta de ATI Madrid) <[mikelboi\\_uni@yahoo.es](mailto:mikelboi_uni@yahoo.es)>  
**Profesión Informática**  
Rafael Fernández Calvo (ATI) <[rfcalvo@ati.es](mailto:rfcalvo@ati.es)>  
Miguel Sarrías Grijó (ATI) <[miquel@sarries.net](mailto:miquel@sarries.net)>

**Redes y servicios telemáticos**  
José Luis Marco Lázaro (Univ. de Girona) <[joseluis.marco@udg.es](mailto:joseluis.marco@udg.es)>  
Juan Carlos López López (UCLM) <[juancarlos.lopez@uclm.es](mailto:juancarlos.lopez@uclm.es)>  
**Robótica**  
José Cortés Arenas (Sopra Group) <[jccortesa@gmail.com](mailto:jccortesa@gmail.com)>  
Juan González Gómez (Universidad CARLOS III) <[juan@iearobotics.com](mailto:juan@iearobotics.com)>

**Seguridad**  
Javier Arellano Bertollini (Univ. de Deusto) <[jaarellito@deusto.es](mailto:jaarellito@deusto.es)>  
Javier López Muñoz (ETS Informática-UMA) <[jlm@lcc.uma.es](mailto:jlm@lcc.uma.es)>  
**Sistemas de Tiempo Real**  
Alejandro Alonso Muñoz, Juan Antonio de la Fuente Alfaro (DIT-UPM) <[galonso@puente@dit.upm.es](mailto:galonso@puente@dit.upm.es)>  
**Software Libre**  
Jesús M. González Barahona (GSVC - URJC) <[jgib@gsvic.es](mailto:jgib@gsvic.es)>  
Isra Herráiz Taberner (Universidad Politécnica de Madrid) <[isra@herraz.org](mailto:isra@herraz.org)>

**Tecnología de Objetos**  
Jesus Garcia Moine (DIS-UMI) <[jmoine@um.es](mailto:jmoine@um.es)>  
Gustavo Rossi (LIFIA-UNLP Argentina) <[gustavo@sol.info.unlp.edu.ar](mailto:gustavo@sol.info.unlp.edu.ar)>  
**Tecnologías para la Educación**  
Juan Manuel Dodero Beardo (UC3M) <[dodero@inf.uc3m.es](mailto:dodero@inf.uc3m.es)>  
César Pablo Córcoles Briongo (UOC) <[ccorcoles@uoc.edu](mailto:ccorcoles@uoc.edu)>  
**Tecnologías y Empresa**  
Didac López Viñas (Universitat de Girona) <[didac.lopez@ati.es](mailto:didac.lopez@ati.es)>  
Francisco Javier Cantais Sánchez (Indra Sistemas) <[fjcantais@gmail.com](mailto:fjcantais@gmail.com)>

**Tendencias tecnológicas**  
Alonso Álvarez García (TID) <[aad@tid.es](mailto:aad@tid.es)>  
Gabriel Martí Fuentes (Interbits) <[gabim@inet.es](mailto:gabim@inet.es)>  
**TIC y Turismo**  
Andrés Aguiayo Maldonado, Antonio Guevara Plaza (Univ. de Málaga) <[aguiayo, guevara@lcc.uma.es](mailto:(aguiayo, guevara)@lcc.uma.es)>

Las opiniones expresadas por los autores son responsabilidad exclusiva de los mismos.  
**Novática** permite la reproducción, sin ánimo de lucro, de todos los artículos, a menos que lo impida la modalidad de © o copyright elegida por el autor, debiéndose en todo caso citar su procedencia y enviar a **Novática** un ejemplar de la publicación.

**Coordinación Editorial, Redacción Central y Redacción ATI Madrid**  
Padilla 66, 3º, dcha., 28006 Madrid  
Tlf. 91 402 93 91; fax 91 309 36 85 <[novatica@ati.es](mailto:novatica@ati.es)>  
**Composición, Edición y Redacción ATI Valencia**  
Av. del Reino de Valencia 23, 46005 Valencia  
Tlf. 96 37 40 17 3 <[novatica\\_valencia@ati.es](mailto:novatica_valencia@ati.es)>  
**Administración y Redacción ATI Cataluña**  
Via Laletana 46, ppal. 1º, 08003 Barcelona  
Tlf. 93 41 25 23 5; fax 93 41 27 7 1 3 <[secregen@ati.es](mailto:secregen@ati.es)>  
**Redacción ATI Aragón**  
Lagasca 3, 5-8, 50006 Zaragoza  
Tlf. / fax 91 82 3 81 <[secreara@ati.es](mailto:secreara@ati.es)>  
**Redacción ATI Andalucía** <[secreand@ati.es](mailto:secreand@ati.es)>  
**Redacción ATI Galicia** <[secregal@ati.es](mailto:secregal@ati.es)>

**Subscripción y Ventas** <[novatica.subscriptions@atinet.es](mailto:novatica.subscriptions@atinet.es)>  
**Publicidad** Padilla 66, 3º, dcha., 28006 Madrid.  
Tlf. 91 402 93 91; fax 91 309 36 85 <[novatica@ati.es](mailto:novatica@ati.es)>  
**Imprenta:** Derra S.A., Juan de Austria 66, 08005 Barcelona.  
**Depósito legal:** B 15.154-1975 -- ISSN: 0211-2124; CODEN NOVAEC  
**Portada:** Escalera de color - Concha Arias Pérez / © ATI  
**Diseño:** Fernando Agresta / © ATI 2003

**editorial**

**Ingeniería del Software en un momento de cambios y crisis en resumen** > 02

**Ingeniería del Software y sistemas de engranaje múltiple** > 02

*Llorenç Pagés Casas*  
**noticias de IFIP**

**Reunión del Board de IFIP** > 03  
*Ramon Puigjaner Trepat*

**IFIP TC3 en el congreso WSIS+ 10 de UNESCO** > 04  
*Carlos Delgado Kloos*

**Ramon López de Mántaras obtiene el Premio Nacional de Informática 2012** > 05

**monografía**

**Técnicas avanzadas de desarrollo modular**  
*Editores invitados: Mercedes Amor Pinilla, Lidia Fuentes Fernández, Mónica Pinto Alarcón*

**Presentación. Enfoques actuales para el desarrollo de software modular** > 06  
*Mercedes Amor Pinilla, Lidia Fuentes Fernández, Mónica Pinto Alarcón*

**Análisis de la modularidad en sistemas software mediante un proceso de minería de aspectos** > 09  
*José María Conejero Manzano, Juan Hernández Núñez*

**La Programación Orientada a Aspectos como mecanismo para aumentar la modularidad en la implementación de aplicaciones** > 19  
*Francisco Ortín Soler*

**Enfoque basado en MDA para apoyar evoluciones seguras en sistemas orientados a aspectos** > 25  
*Paulo F. Pires, Flávia C. Delicato, Jesús Martín Talavera Portocarrero*

**Modularidad en transformaciones de modelos** > 34  
*Jesús J. García Molina, Jesús Sánchez Cuadrado*

**Separación avanzada de conceptos en el desarrollo de aplicaciones web** > 42  
*Antonia Mª Reina Quintero, Rafael Corchuelo Gil, Miguel Toro Bonilla*

**Variabilidad en Ingeniería de Software Empotrado** > 51  
*Salvador Trujillo González, David Benavides Cuevas*

**secciones técnicas**

**Enseñanza Universitaria de la Informática**

**Los conocimientos que deberán incluir los planes y programas de estudio de informática los próximos años en México (Modelo Curricular)** > 54  
*Lourdes Sánchez-Guerrero, Rafaela Blanca Silva, José Raymundo Lira-Cortés*

**Mundo estudiantil y jóvenes profesionales**

**Nela: Aprende a escribir usando Braille** > 57  
*Enrique Matías Sánchez, Inmaculada Plaza García, Nuria Tregón Martín*

**GeoTask: Servicios basados en localización para sistemas Android** > 62  
*Francisco Javier Martín Otero*

**daf-collage: Un proyecto innovador en la enseñanza del idioma alemán** > 66  
*Francisco Javier Rodríguez López, Simeón Ruiz Romero*

**Cormoran: Un framework de persistencia para Python** > 69  
*Jaime Gil de Sagredo Luna*

**Referencias autorizadas** > 71

**Sociedad de la Información**

**Programar es crear**

**El problema del supermercado (Competencia UTN-FRC 2011, problema E, solución)** > 77  
*Julio Javier Castillo, Diego Javier Serrano, Marina Elizabeth Cárdenas*

**asuntos interiores**

**Coordinación editorial / Programación de Novática / Socios Institucionales** > 79

Enrique Matías Sánchez<sup>1</sup>,  
Inmaculada Plaza García<sup>2</sup>,  
Nuria Tregón Martín<sup>3</sup>

<sup>1,2,3</sup>Universidad de Zaragoza; <sup>1</sup>Ganador del VI Concurso Universitario de Software Libre en la categoría "Mejor proyecto de accesibilidad" y del Premio Especial del mismo Concurso

<{quique,inmap,niritre}@unizar.es>

## 1. Introducción

Nela, el programa informático que se presenta en este artículo, tiene como principal objetivo servir de herramienta de apoyo en el aprendizaje de la escritura usando el código Braille.

Aunque en un inicio se diseñó pensando en su utilización por parte de niños con discapacidad visual (ceguera o visión reducida), su utilización no se limita a este público objetivo. Nela puede ser utilizado igualmente con niños sin discapacidad visual a los que se desee introducir en la enseñanza del código Braille, pero también por jóvenes o adultos que hayan perdido la visión (o tengan visión reducida) debido a algún accidente o enfermedad.

La importancia de utilizar las Tecnologías de la Información y Comunicación (TIC) en el ámbito de la discapacidad visual ya ha sido tratada previamente por diversos autores, como F. Lidner, P. Toledo y C. Hervás (1992) [1], o el grupo ACCEDO (*Accesibilidad a Contenidos Educativos ONCE*), siendo este último apoyo fundamental para la creación e integración de nuevas tecnologías para personas con discapacidad y específicamente para personas con discapacidad visual.

Aunque ya es ampliamente reconocido el papel que desempeñan las nuevas tecnologías en el desarrollo cognitivo, concretamente por su gran motivación en el discente, hemos de remarcar la importancia de las TIC en el proceso de enseñanza-aprendizaje de los alumnos con discapacidad visual al ser una herramienta, imprescindible y a su vez complementaria, tanto para el acceso a la información como para la comunicación.

Para presentar Nela, el artículo se ha estructurado en los siguientes apartados: en la **sección 2** se describe el contexto en el que surge este proyecto, que ha dado lugar a la realización de un trabajo fin de carrera cuyo resultado es el programa informático presentado. Tras esta contextualización, se pasará a describir con mayor detalle diferentes aspectos de Nela: su funcionamiento (**sección 3**), algunos aspectos a considerar en su utilización en la enseñanza (**sección 4**), aspectos técnicos (**sección 5**) y su licencia (**sección 6**). Para finalizar comentaremos brevemente

# Nela: Aprende a escribir usando Braille

**Resumen:** Nela es un programa informático que ayuda en el aprendizaje de la escritura usando el código Braille, simulando una máquina Perkins con el teclado del ordenador. Está dirigido a niños con discapacidad visual (visión reducida o ceguera). Programado en C++ usando las bibliotecas Qt, es multiplataforma (funciona tanto en Microsoft Windows como en Gnu/Linux) y está internacionalizado. Ha sido valorado positivamente por varios educadores, y obtuvo el primer premio en la fase estatal del VI Concurso Universitario de Software Libre, así como el premio especial de accesibilidad.

**Palabras clave:** Aprendizaje, Braille, ceguera, discapacidad visual, enseñanza, escritura, escuela, formación, software educativo.

## Autores

**Enrique Matías Sánchez** es estudiante de Ingeniería Técnica en Informática de Gestión en la Escuela Universitaria Politécnica de Teruel. Nela es su trabajo de fin de carrera. Decidido defensor del Software Libre, en 2003 organizó una multitudinaria conferencia de Richard Stallman en Zaragoza. Ha sido traductor de KDE, colaborador de la revista Mundo Linux y miembro de diversas organizaciones de carácter social. Actualmente trabaja en el Servicio de Informática de la Universidad de Zaragoza.

**Inmaculada Plaza García** es Profesora Titular de Universidad en la Universidad de Zaragoza e Investigadora Principal del grupo EduQTech (*Education-Quality-Technology*), primer grupo de investigación con sede en Teruel reconocido por el Gobierno de Aragón en el área Tecnológica. Dirige la "Cátedra en innovación y calidad tecnológica" de la Universidad de Zaragoza. Presidenta de la Sociedad de Educación del IEEE en España (2010-2012), actualmente es *Past-Chair* de la misma. Su investigación se centra en tres áreas: calidad de vida/salud, especialmente en el ámbito de las neurociencias, calidad en educación y calidad e innovación empresarial; siempre desde un enfoque de desarrollo en el medio rural. Promotora del software y hardware libre, intenta difundir su utilización en todos los proyectos y redes en los que participa.

**Nuria Tregón Martín** actualmente ejerce como maestra de apoyo del equipo ONCE de Aragón. Paralelamente, es Profesora Asociada de la Universidad de Zaragoza, en el Departamento de Teoría e Historia de la Educación de la Facultad de Ciencias Sociales y Humanas en el Campus de Teruel. Ha finalizado el Máster de Cooperación al Desarrollo y su línea de investigación va encaminada al estudio de la discapacidad en países empobrecidos.

la primera prueba piloto (**sección 7**), concluyendo con un apartado de líneas de trabajo futuro (**sección 8**).

## 2. Contexto y antecedentes

La idea de desarrollar Nela surgió como parte de un proyecto mucho mayor, iniciado en el año 2007, cuando dos profesionales que trabajaban con niños con discapacidad visual en la provincia de Teruel (Lucía Azara e Inés Benedicto) se pusieron en contacto con el grupo de investigación EduQTech (Universidad de Zaragoza) planteando la necesidad de desarrollar herramientas que facilitasen la enseñanza y el aprendizaje de los caracteres del código Braille.

En general, los recursos existentes dirigidos a

los discapacitados visuales no son numerosos y debido a su limitado interés comercial, caros. En concreto, los dispositivos o aplicaciones para la adquisición de la lectoescritura para niños en las primeras etapas de escolarización son muy reducidos, por lo que las profesionales indicaron la necesidad de ampliar el abanico de posibilidades desarrollando herramientas personalizables y que favorecieran el aprendizaje de un modo progresivo.

Para conocer el estado del arte se analizaron diferentes antecedentes. Sin ánimo de ser exhaustivos se pueden mencionar:

- La propuesta de Cétares et al. [2] de un sistema de enseñanza del código Braille para niños con limitaciones visuales.

“ Nela solicitará al niño que escriba una palabra, y le comunicará, tanto de forma visual como sonora, si la introducción ha sido correcta o no, mostrando en la parte inferior de la pantalla la palabra escrita en Braille ”

■ *Automated Braille Writing Tutor*, dispositivo desarrollado en la Universidad de Carnegie Mellon y que desde el primer momento se probó en la India.

■ La patente estadounidense US5902112 (Sally S. Mangold), con una matriz rectangular de zonas sensibles al tacto y un procesador acoplado a un sintetizador que pronuncia el carácter Braille.

■ La patente alemana DE19547742 (René LEMOINE), en la que se describe un dispositivo y un método para practicar idiomas por escrito u oralmente sin necesidad de profesor. En concreto, reinventa un método para la introducción y muestra de signos visuales y sonoros a elección del usuario.

■ La solicitud internacional de patente WO2008120303 con unidades de entrada de caracteres Braille para introducir un carácter Braille por medio de una pluralidad de unidades de presión y de unidades de confirmación para la salida de los caracteres Braille introducidos.

■ La patente estadounidense US576963, especialmente destinada a la enseñanza de operaciones de cálculo básicas.

Así mismo, y a medida que se avanzó en el tiempo, se fueron analizando algunos programas informáticos como por ejemplo "Salón Braille Virtual" o "Cantaletas" y, como complemento, aplicaciones informáticas des-

tinadas a personas no ciegas (familiares, profesionales) como el "curso básico de autoaprendizaje del Braille" de la ONCE (Organización Nacional de Ciegos Españoles), "Braille Virtual" de la Universidad de São Paulo (Brasil), "Detrás de cada punto" del INCI (Instituto Nacional de Ciegos de Colombia), "Demo Braille" y "Alfabeto Braille en línea" de la FBU (Fundación Braille del Uruguay).

El análisis del estado del arte permitió, y ha seguido permitiendo a lo largo de estos años, detectar distintas necesidades no cubiertas. Como su descripción excedería el objetivo de esta publicación, se puede consultar la patente ES2366509 "Dispositivo electrónico para la enseñanza y el aprendizaje de la escritura de caracteres en código Braille" de la Universidad de Zaragoza, en la que se presenta un análisis más detallado y las conclusiones obtenidas.

De esta forma, surgió el proyecto "TerBraille" que se definió pensando en dos etapas:

1) Una primera etapa en la que los usuarios pudieran aprender los caracteres alfanuméricos en código Braille a través de un dispositivo electrónico.

2) Una segunda etapa, en la que se introduciría a los usuarios en el manejo del ordenador, reforzando el aprendizaje de los caracteres alfanuméricos a la vez que extendiendo el aprendizaje a sílabas y palabras. Para ello se

definiría un programa informático basado en el uso de la máquina Perkins<sup>1</sup>.

El grupo EduQTech utiliza y fomenta el uso de software/hardware libre, por lo que los diferentes desarrollos se realizaron bajo esta filosofía.

La primera versión del dispositivo electrónico fue desarrollada, a través de sendos trabajos fin de carrera, por Óscar Díez Sáez [3] y David Murciano [4], dirigidos conjuntamente por los doctores Carlos Medrano e Inmaculada Plaza. El prototipo, basado en la plataforma Arduino fue probado durante un año en un colegio de Teruel, y posteriormente por profesionales de Zaragoza.

En una segunda versión, realizada por Jorge Izquierdo [5], se añadió la comparación del dispositivo con una plataforma conocida basada en PIC. Finalmente, y como cuarto trabajo fin de carrera, se propuso la realización del software objeto de esta publicación. De esta forma, Nela constituye la segunda etapa del proyecto. Junto con el dispositivo electrónico puede considerarse un kit de enseñanza/aprendizaje de la escritura, en el que cada elemento se puede utilizar de forma individual o complementaria.

### 3. Funcionamiento del programa

Como hemos indicado en la introducción, aunque el programa Nela puede ser utilizado por jóvenes y adultos, se desarrolló inicialmente para niños con discapacidad visual, que hayan alcanzado la madurez necesaria para abordar la capacidad lectoescritora.

Nela simula una máquina Perkins usando el teclado normal del ordenador. Una máquina Perkins viene a ser una especie de máquina de escribir para ciegos, que sólo tiene 6 teclas, una para cada uno de los seis puntos del cajetín Braille (más el espaciador, etc).

El niño colocará las manos sobre la fila central del teclado, con los dedos índices sobre las teclas F y J, que tienen unas marcas en relieve. Si el niño tiene dificultades para notar ese relieve, se puede pegar un trozo de fieltro sobre las teclas. Las teclas S-D-F y J-K-L corresponden respectivamente a los puntos 3-2-1 y 4-5-6 del cajetín Braille. Para escribir una letra se pulsán simultáneamente las teclas correspondientes a los puntos que la componen en el código Braille.



Figura 1. Interfaz gráfica de Nela.

Nela solicitará al niño que escriba una palabra, y le comunicará, tanto de forma visual como sonora, si la introducción ha sido correcta o no, mostrando en la parte inferior de la pantalla la palabra escrita en Braille.

El programa introduce automáticamente una nueva palabra una vez que el niño escribe habitualmente de forma correcta las palabras presentadas hasta el momento. Las palabras ya vistas seguirán presentándose ocasionalmente para repararlas. Cuando se cometa un error en una palabra, ésta será presentada con mayor frecuencia, para reforzarla. Del mismo modo, las palabras que no presenten dudas aparecerán más raramente.

## 4. Aspectos a considerar en la enseñanza utilizando Nela

El objetivo de Nela es aprender a escribir, usando Braille como código. Por ello, antes de empezar a utilizarlo, es recomendable que el niño haya alcanzado cierta madurez lectoescritora: conceptos espaciales básicos (arriba/abajo, izquierda/derecha), nociones de cantidad (primeros números), coordinación dígito-manual, etc. Con este fin se puede usar el *Método Alameda* [6].

En la enseñanza de la lectoescritura Braille, debería evitar una presentación ordenada de las letras del alfabeto, tal y como aparece en algunos métodos alfabéticos en tinta. Es muy importante no enseñar simultáneamente, desde un principio, letras muy parecidas, para evitar posibles confusiones que dificultarían el proceso de enseñanza, y que en muchas ocasiones son difíciles de corregir si se han adquirido. Teniendo en cuenta la secuencia sugerida por Susana E. Crespo [7], y de acuerdo con su propia experiencia, Begoña Espejo [8] presenta un orden por el que introducir las letras. Para más detalles se puede consultar la documentación de Nela o la bibliografía.

Es esencial fomentar el interés del niño por adquirir la habilidad de leer y escribir, de igual manera que se hace con el niño vidente. Por ello, Nela hace uso desde el primer momento de un vocabulario con significado afectivo y vivencial (boca, cama, paloma...) seleccionado de un corpus de vocabulario infantil.

Paulatinamente, se introducen nuevas palabras con nuevas sílabas y nuevas letras, con dificultad progresiva (sílabas directas, inversas, mixtas y trabadas). Si bien este método tiene una sólida base pedagógica, Nela no lo impone, y el docente que lo desee puede no obstante personalizar el programa para seguir otra metodología.

La aplicación se adapta a las necesidades específicas de cada usuario, no introduciendo

nuevas palabras hasta que no se hayan asimilado las anteriores, y reforzando las que le resulten más complicadas al niño.

## 5. Aspectos técnicos

Nela está dirigido a niños tanto a niños con visión reducida como con ceguera. Para permitir su uso en niños que presenten ceguera total, Nela no precisa del uso del ratón: todas las características del programa están accesibles vía teclado.

Para facilitar su uso en niños con baja visión, Nela emplea letras de gran tamaño. El tipo de letra elegido (Tiresias PCfont<sup>2</sup>) está diseñado para una legibilidad óptima por la Unidad de Investigación Científica del *Royal National Institute of Blind People*<sup>3</sup> (Real Instituto Nacional para los Ciegos) de Londres.

### 5.1. Multiplataforma

Uno de los requisitos de diseño fue que el software debía ser multiplataforma, y funcionase tanto en Microsoft Windows como en GNU/Linux. Se pueden distinguir tres maneras diferentes de escribir software multiplataforma:

- 1) Haciendo uso de lenguajes interpretados, como Python.
- 2) Mediante compilación a *bytecode* intermedio, que será ejecutado en una máquina virtual, como Java.
- 3) Escribiendo el código fuente en un lenguaje estandarizado que se compilará a código nativo en cada plataforma.

Cada método tiene sus ventajas y desventajas, como son la rapidez de ejecución y comodidad para el usuario o el desarrollador. En este caso se optó por la tercera opción, y desarrollaron en C++ usando las bibliotecas Qt.

Aunque mucha gente piensa en las Qt como un *toolkit* gráfico (como pueda ser Swing), sería más acertado entenderlas como una alternativa, más amplia, a la *Standard Template Library de C++*. De hecho, gracias a su arquitectura modular, se pueden emplear para escribir aplicaciones no gráficas.

Además de unas clases para interfaces gráficas con una buena integración con el entorno del usuario, las Qt proporcionan soporte para Unicode, *signals* y *slots* (una implementación del patrón de diseño *Observer o publish-subscribe*) y facilitan la recolección de basura.

Se ha comprobado el correcto funcionamiento del programa sobre GNU/Linux y Microsoft Windows 7. Debería ser bastante sencillo adaptarlo también a Mac OS X.

Además del sistema operativo, otros aspectos a considerar son el tamaño de palabra y la *endianness*<sup>4</sup>. Nela se ha probado sobre 32 y 64 bits. Aunque el autor sólo disponía de siste-

mas *Little Endian*, no se hacen operaciones que puedan suponer un problema en sistemas *Big Endian*.

Si bien el código fuente de Nela se puede compilar en diversos sistemas, éste no es un método accesible a los usuarios finales. Desde la página web se puede descargar un instalador para Windows, que proporciona una instalación guiada que no ofrece ninguna dificultad. Hay diversos programas para crear instaladores para Windows (InstallShield, NSIS, WiX, etc). Se eligió Inno Setup<sup>5</sup>, que es software libre, tiene todas las funcionalidades requeridas (como la capacidad de instalar tipos de letra) y está razonablemente documentado.

También se facilita un paquete .deb para sistemas de tipo Debian como Ubuntu. Previamente habrá que tener instalados los paquetes de los que depende (libqt4-svg, libphonon4, etc). Conviene comprobar que no se esté usando el motor de Phonon basado en xine (que ha sido abandonado y ya no recibe mantenimiento), sino los backends actuales basados en VLC o *gststreamer*. El programa está disponible para la distribución Chakra GNU/Linux en el repositorio CCR (*Chakra Community Repository*).

### 5.2. Internacionalización

Otro requisito de diseño fue que el programa estuviese internacionalizado (preparado para traducirse a otros idiomas). En el caso de Nela, esto implica cuatro actividades diferenciadas:

- Traducción de la interfaz del programa.
- Elaboración de la secuencia de palabras.
- Creación de la tabla Braille del idioma en cuestión.
- Grabación de los sonidos.

Todos los textos que aparecen en el programa (menús, etc) se encuentran en un fichero en formato XML y extensión .t.s. La traducción se puede realizar usando el programa Qt Linguist, o un simple editor de texto.

La secuencia de palabras que se presentan al niño debería seguir un orden definido con un criterio científico como el presentado anteriormente, adecuado a las características del idioma en cuestión. Esta secuencia se encuentra en un fichero en formato csv (valores separados por tabuladores). La primera columna es la palabra a presentar, la segunda columna es la cadena en ASCII de 7 bits que se usará para los nombres de los ficheros de las grabaciones correspondientes, y la tercera columna es la imagen que ilustra dicha palabra. Este fichero se puede editar con una hoja de cálculo o un editor de texto.

Inicialmente había una cuarta columna con la palabra escrita en Braille (caracteres Unicode 0x2800 - 0x28FF), para poder comparar con

“ El código Braille es diferente en cada idioma, pues ha de adaptarse a los caracteres que emplee (en el caso del castellano, vocales acentuadas, letra ñe... ). Por ello, ha de crearse un fichero de correspondencias letra-puntos para cada idioma ”

la respuesta introducida por el niño, y mostrar la respuesta correcta en pantalla. Sin embargo esto dificultaba la elaboración de este fichero, porque no hay ninguna manera sencilla de introducir esos caracteres desde el teclado. Finalmente se optó por generar la cadena en Braille a partir de la primera columna.

La primera aproximación fue crear una tabla *hash* que asociaba cada letra con el número del carácter Unicode que le corresponde en Braille. Esta solución tampoco es óptima, porque los educadores o traductores que deseen editar este fichero probablemente no estén familiarizados con las tablas Unicode.

La solución final fue añadir un nuevo fichero de texto con dos columnas: la primera con cada letra, y la segunda con los puntos del carácter Braille que le corresponden (por ejemplo: g 1245). A partir de dichos puntos, se calcula el número del carácter Unicode (10267 en el caso de la letra g) y se devuelve el carácter en cuestión. El código de la función puede verse en la **figura 2**.

El código Braille es diferente en cada idioma, pues ha de adaptarse a los caracteres que emplee (en el caso del castellano, vocales acentuadas, letra ñe e interrogante y admiración inicial). Por ello, ha de crearse un fichero de correspondencias letra-puntos para cada idioma.

Un interesante problema técnico fue la simulación de la máquina Perkins, pues obviamente los sistemas no están pensados para detectar la pulsación simultánea de hasta 6 teclas. Además, los dedos de la mano no se mueven exactamente a la vez.

El sistema emite un evento cada vez que se pulsa o libera una tecla, y lo que hace Nela es recoger en una lista los números del cajetín Braille correspondientes a las teclas soltadas en los 400 milisegundos siguientes a la primera liberación (valor escogido empíricamente). A continuación ordena esa lista, elimina duplicados si los hay, y se la pasa como una cadena de texto a la función mostrada anteriormente, que devuelve el carácter Braille correspondiente a las pulsaciones del usuario. Finalmente, se compara ese carácter con el esperado y se actúa en consecuencia. Se ha observado que algunos teclados, particularmente en *netbooks*, no emiten correctamente los eventos de liberación de teclas cuando hay varias teclas pulsadas a la vez. En caso de encontrarnos ante un hardware con esta limitación, la solución es conectar un teclado externo.

Las grabaciones sonoras se encuentran en formato MP3, aunque en el futuro se podría investigar la posibilidad de usar Speex<sup>6</sup>, un formato de compresión de audio diseñado para la voz humana. Se pueden realizar con un micrófono doméstico y un programa como Audacity<sup>7</sup>, que incluye una funcionalidad de reducción de ruido.

Las imágenes empleadas no se encuentran en formatos *rasterizados* como JPEG o PNG, sino en el formato vectorial SVG que, además de ocupar menos espacio en disco, se pueden escalar a cualquier tamaño de pantalla.

### 6. Licencia

Nela es Software Libre [9], con licencia GNU GPL 3 o posterior: se puede usar, copiar a

otras personas, modificar y distribuir versiones modificadas de forma gratuita y legal.

El desarrollo está abierto a cualquier persona interesada.

### 7. Primeras pruebas piloto

Como prueba piloto inicial se contó con la colaboración desinteresada de cuatro docentes de distintos puntos de España que aportaron ideas y sugerencias de mejora para próximas versiones.

El problema más claro es la baja calidad del audio. Los sonidos fueron grabados inicialmente con el micrófono interno de un portátil. Es necesario volver a grabarlos en un estudio insonorizado, con un micrófono de calidad y a un volumen adecuado.

Otra propuesta es utilizar una mayor variedad de sonidos para el *feedback*, y hacerlos más breves, para evitar que puedan llegar a hacerse monótonos.

También han sugerido algunas mejoras de accesibilidad, por ejemplo para usuarios con problemas motores y que sólo puedan usar una mano. Por último, se nos ha planteado la posibilidad de crear una versión para Mac OS X.

### 8. Líneas de trabajo futuro

Como hemos indicado en la **sección 6**, el desarrollo está abierto a cualquier persona interesada: educadores que propongan mejoras, desarrolladores que deseen implementarlas, traductores que adapten el programa a otros idiomas, ilustradores que faciliten dibujos libres...

Se pretende, de este modo, que la presente versión de Nela sea un punto de partida y un lugar de encuentro y colaboración a través del blog<sup>8</sup> y de la forja del proyecto en RedIris<sup>9</sup>.

### 9. Agradecimientos

Los autores agradecen su colaboración a todos aquellos educadores que han intervenido en la prueba piloto de evaluación, enriqueciendo el trabajo con sus comentarios y sugerencias. Así mismo, agradecen las aportaciones de D. Hermes Ojeda Ruiz quien ha colaborado mediante parches, y empaquetamiento para Chakra GNU/Linux.

```
QChar PerkinsModel::pointsToUnicode(QString points)
{
    int code = 10240;
    foreach (const QString &digit, points) {
        code += pow(2, digit.toInt() - 1);
    }
    return QChar(code);
}
```

**Figura 2.** Función que devuelve el carácter Unicode a partir de los puntos del carácter Braille.



## Referencias

- [1] **M<sup>a</sup> Rosa Villalba Simón.** *Aspectos evolutivos y educativos de la deficiencia visual.* Volumen II. ONCE, Madrid, 2000.
- [2] **A. Cétares, C.A Cortés, L.F. Silva.** *Sistema de enseñanza del código Braille para niños con limitaciones visuales.* Trabajo de Grado. Facultad de Ingeniería, Pontificia Universidad Javeriana, Bogotá (Colombia), 2005.
- [3] **Óscar Díaz Saez.** *Entrenador para niños con discapacidad visual.* Trabajo de Fin de Carrera, EUPT, Universidad de Zaragoza, 2009.
- [4] **David Murciano Ibáñez.** *Evaluación y mejora de un dispositivo electrónico para la enseñanza-aprendizaje del código Braille.* Trabajo de Fin de Carrera, EUPT, Universidad de Zaragoza, 2010.
- [5] **Jorge Izquierdo Najas.** *Dispositivo de aprendizaje de código Braille basado en un microcontrolador PIC: comparación con el desarrollo basado en Arduino.* Trabajo de Fin de Carrera, EUPT, Universidad de Zaragoza, 2012.
- [6] **Julia Fuentes Hernández.** *Método Alameda.* ONCE, Madrid, 1995.
- [7] **Susana E. Crespo.** *La escuela y el niño ciego.* Manual práctico. Graficart, Córdoba (Argentina), 1980.
- [8] **Begoña Espejo de la Fuente.** *El Braille en la escuela.* Una guía práctica para la enseñanza del Braille. ONCE, Madrid, 1993.
- [9] **Enrique Matías Sánchez.** "Breve introducción al software libre". Pueblos, diciembre 2004, nº 14 pp. 49-51. <[http://cusl6-nela.forja.rediris.es/Breve\\_introduccion\\_al\\_software\\_libre.pdf](http://cusl6-nela.forja.rediris.es/Breve_introduccion_al_software_libre.pdf)>.

## Notas

- <sup>1</sup> <[http://en.wikipedia.org/wiki/Perkins\\_Braille](http://en.wikipedia.org/wiki/Perkins_Braille)>.
- <sup>2</sup> <[http://www.tiresias.org/fonts/pcfont/about\\_pc.htm](http://www.tiresias.org/fonts/pcfont/about_pc.htm)>.
- <sup>3</sup> <<http://www.rnib.org.uk/>>.
- <sup>4</sup> El término inglés *endianness* ("extremidad") designa el formato en el que se almacenan los datos de más de un byte en un ordenador. El problema es similar a los idiomas en los que se escriben de derecha a izquierda, como el árabe, o el hebreo, frente a los que se escriben de izquierda a derecha, pero trasladado de la escritura al almacenamiento en memoria de los bytes <<https://es.wikipedia.org/wiki/Endianness>>.
- <sup>5</sup> <<http://www.jrsoftware.org/isinfo.php>>.
- <sup>6</sup> <<http://www.speex.org/>>.
- <sup>7</sup> <<http://audacity.sourceforge.net/>>.
- <sup>8</sup> <<http://nelaproject.blogspot.com.es/>>.
- <sup>9</sup> <<https://forja.rediris.es/projects/cusl6-nela/>>.



Ganadores del VI Concurso Universitario de Software Libre (edición de 2012). De izquierda a derecha: Enrique Matías Sánchez, Jaime Gil de Sagredo Luna, Francisco Javier Rodríguez López, Simeón Ruiz Romero y Francisco Javier Martín Otero.