

Desarrollo de una solución de software dirigida a niños con deficiencia auditiva

C.H. Aparicio-Alvarez¹, R. L. Lucio-Martínez², E. Moreno-Nuñez³, E. A. Meraz-Salazar⁴.

Resumen— El desenvolvimiento social de niños con deficiencia auditiva inmersos en una cultura oyente, presenta múltiples diferencias en relación con el resto de los miembros de su grupo generando grandes barreras en la inserción escolar. El artículo presenta el desarrollo de una solución tecnológica dirigida a niños con problemas de audición; se redacta la manera en que la solución tecnológica facilita el aprendizaje de los niños acerca de los elementos fundamentales del lenguaje de señas mexicano, en las instituciones públicas de nivel básico. El objetivo de esta solución es que los niños con y sin deficiencia auditiva comprendan los conceptos básicos de dicho lenguaje, por medio de la implementación computacional de un juego. Las interfaces interactivas permiten una retroalimentación constante entre los niños y la plataforma computacional, al mismo tiempo que les facilita el aprendizaje gracias a la dinámica de un juego de memoria. Entre los resultados obtenidos, la solución desarrollada se implementó en una escuela primaria de la Comarca Lagunera, con un grupo de 30 niños, reportando excelentes efectos en el interés de los menores por aprender el lenguaje de señas mexicano para comunicarse con sus compañeros con deficiencia auditiva.

Palabras claves—Deficiencia, desarrollo, desenvolvimiento social, implementación, inmerso, retroalimentación.

Abstract— The social development of children with hearing impairment, immersed in a listening culture, presents multiple differences in relation to the rest of the members of their group, generating great barriers in school insertion. The article presents the development of a technological solution aimed at children with hearing problems; the way in which the technological solution facilitates children's learning about the fundamental elements of Mexican sign language, in public institutions at the basic level, is written. The objective of this solution is that children with and without hearing impairment understand the basic concepts of said language, through the computational implementation of a game. The interactive interfaces allow a constant feedback between the children and the computational platform, at the same time that it facilitates the learning thanks to the dynamics of a memory game. Among the results obtained, the solution developed was implemented in a primary school in the Comarca Lagunera, with a group of 30 children, reporting excellent effects in the interest of the children to learn the Mexican sign language to communicate with their peers with hearing impairment.

Keywords— Deficiency, development, social development, implementation, immersed, feedback.

I. INTRODUCCIÓN

Hoy en día las dificultades que enfrentan los niños con deficiencia auditiva para poder comunicarse en una cultura oyente, merman en su desarrollo educativo, por lo que se ven condicionadas sus oportunidades de inclusión.

Desde hace mucho tiempo en la historia el ser humano ha sido capaz de poder comunicarse de diferente manera, la comunicación empieza por medio de sonidos y gestos que hacían los primeros hombres en la tierra para poder comunicarse sí mismos. Tanto ha evolucionado la manera en que nos expresamos verbal, que se han ido desarrollando nuevas técnicas de comunicación. El LSM (lenguaje de señas mexicano o lengua de señas mexicana) es un lenguaje de expresión visual que se implementó a mediados del siglo xx. Dicho lenguaje derivó del lenguaje de señas francesas y es fundamental en la vida de personas con deficiencia auditiva, esto debido a que les permite adquirir individualidad e independencia, para que ellos puedan establecer una personalidad propia.

La deficiencia auditiva es uno de los problemas más grandes que afectan a la sociedad, pues en el país hay tan sólo 40 intérpretes certificados en lenguaje de señas mexicano y la educación especial para ellos no es obligatoria.

En el país existen 2.4 millones de personas con dicha deficiencia según un sondeo realizado en 2014. En la Comarca Lagunera existe un número considerable de personas que padecen deficiencia auditiva según el Instituto Lagunero de Audición y Lenguaje A.C. Tan solo en ésta región el porcentaje de personas con dicho padecimiento es del 21 % (27,459 personas) según estadísticas del INEGI en 2010.

El presente artículo describe el proceso de una investigación aplicada que tuvo como objetivo desarrollar una solución de software dirigida a niños con deficiencia auditiva, la cual es capaz de enseñar los conceptos básicos del lenguaje de señas mexicano (LSM) a niños de 8 a 12 años, de modo que les permita comunicarse por medio de este lenguaje.

La aplicación fue desarrollada para ser utilizada en los centros de cómputo de las instituciones públicas de

educación básica en la comarca lagunera. El lenguaje de programación utilizado fue C# en el entorno de desarrollo Visual Studio, con SQL Server 2014, sobre un sistema operativo Windows a partir de la versión 7. En la parte técnica del escrito se detalla la metodología aplicada en el proyecto, así como las herramientas tecnológicas de software que fueron utilizadas en cada una de las etapas de desarrollo.

II. PARTE TÉCNICA DEL ARTÍCULO

A. *Fundamento Teórico.*

Interfaces GUI

Una interfaz gráfica de usuario (GUI, por sus siglas en inglés) es un tipo de interfaz que permite a los usuarios navegar por una computadora o dispositivo y completar acciones a través de indicadores visuales e iconos gráficos. Inicialmente se desarrollaron para su uso con mouse y teclado, pero ahora se usan ampliamente en muchos dispositivos móviles de mano, como teléfonos inteligentes y tabletas, y usan una combinación de tecnologías para proporcionar una plataforma para la interacción.

El principal beneficio de una GUI es que los sistemas que usan alguna, son accesibles para personas de todos los niveles de conocimiento, desde un principiante absoluto hasta un desarrollador avanzado u otras personas conocedoras de la tecnología.

Las GUI también brindan retroalimentación instantánea. Hacer clic en un ícono lo abrirá, por ejemplo, y esto se puede ver en tiempo real. Usando una interfaz de línea de comando, no se sabrá si es una entrada válida hasta que se toque el retorno; si no es válido, no pasará nada.

La GUI es una interfaz de usuario gráfica para una computadora que permite la interacción entre el usuario y la computadora. [1]

Visual Studio.

Visual Studio es un IDE (entorno de desarrollo integrado) que se utiliza para el desarrollo de GUI (interfaces gráficas de usuario) así como también para escribir código administrado compatible con .NET. El entorno de desarrollo Visual Studio es compatible con IntelliSense mientras que el depurador integrado admite la depuración de origen y nivel de máquina.

Visual Studio incluye otras herramientas incorporadas en el entorno de desarrollo, como un diseñador de formularios que se utiliza para desarrollar las interfaces gráficas de usuario; un diseñador web para la creación de páginas web

dinámicas; un diseñador de clases para crear bibliotecas personalizadas.

Los servicios de lenguajes incorporados ofrecen soporte para diferentes lenguajes conocidos de programación (como C #, C ++, VB y F #), dichos servicios también se pueden instalar por separado como Python y Ruby, entre otros lenguajes compatibles.

Algunas versiones de Visual Studio admiten lenguajes individuales, que son una solución rentable para principiantes que solo necesitan Visual Basic, o usuarios avanzados que trabajan en aplicaciones de procesamiento rápido y continuamente necesitan C # visual o C ++ visual, pero no están interesados en la baja velocidad de Visual Basic.

Visual Studio es una plataforma de desarrollo de software creativa que puede usarse para ver, editar depurar, crear y publicar una aplicación. [2]

SQL (Structured Query Language).

Structured Query Language es el medio estándar para manipular y consultar datos en bases de datos relacionales, aunque con extensiones propietarias entre los productos. SQL es un lenguaje de programación que sirve para definir y manipular los datos de una base de datos relacional. Las bases de datos se crean como un conjunto de tablas, sus relaciones se representan mediante los valores en las tablas y los datos se recuperan especificando una tabla de resultados que se deriva de una o más tablas base.

Las sentencias de SQL las ejecuta un gestor de bases de datos. Una de las principales funciones del gestor de bases de datos es transformar la especificación de una tabla resultante en una secuencia de operaciones internas que puedan optimizar la recuperación de los datos. Esta transformación se produce en dos fases las cuales son la preparación y vinculación. Así también, las sentencias SQL deben prepararse antes de ejecutarse.

SQL es un lenguaje específico del dominio que da acceso a un sistema de gestión de bases de datos relacionales que permite especificar diversos tipos de operaciones en ellos. [3]

SQL Server

SQL server es un SGBD (sistema gestor de bases de datos) que admite una amplia variedad de aplicaciones de procesamiento de transacciones. Siendo uno de los SGBD líderes en el mercado

Microsoft SQL Server está construido sobre SQL (Structured Query Language), un lenguaje de programación estandarizado que los ADB (administradores de bases de datos) usan para gestionar las bases de datos y realizar consultas con los datos que contienen.

SQL Server está vinculado a Transact-SQL, una implementación de SQL de Microsoft que agrega un

conjunto de extensiones de programación propietarias al lenguaje estándar SQL server Es un sistema para la gestión de bases de datos basado en el modelo relacional el cual permite crear tablas fijas y temporales, procedimientos almacenados, crear vistas y funciones definidas por el usuario. [4]

Ingeniería de Software

La ingeniería de software se conoce como un grupo o conjunto de técnicas y métodos empleados en el desarrollo de sistemas (software).

El ingeniero del proyecto de software es el encargado de toda la gestión del proyecto de software para que pueda ser desarrollado en un determinado tiempo y con el presupuesto establecido.

La ingeniería de software incluye el análisis previo de la situación, el diseño del proyecto, el desarrollo del software, las pruebas de caja negra y caja blanca las cuales son esenciales para su adecuado funcionamiento.

El proceso de desarrollo del software implica un ciclo de vida del mismo el cual está conformado por cuatro etapas: concepción, elaboración, construcción y transición. La idea proporciona el alcance del proyecto y su elaboración sustenta la arquitectura y detalla las características.

La ingeniería de software es la aplicación práctica del conocimiento científico en el desarrollo y construcción de programas de computadoras y la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software de producción. [5]

Lenguaje de programación C#

El lenguaje de programación C# fue creado por el danés Anders Hejlsberg que diseñó también los lenguajes Turbo Pascal y Delphi. C# (pronunciado en inglés "C sharp" o en español "C sostenido") es un lenguaje de programación orientado a objetos (POO). Este lenguaje fue desarrollado para mejorar los lenguajes de los que se deriva como C y C++.

En C# se pretendió incorporar las ventajas o mejoras que tiene el lenguaje JAVA. Dando como resultado que tuviera las ventajas de los lenguajes C y C++ así como la productividad que posee el lenguaje JAVA.

Una de las características notables del lenguaje de programación C# es que su código se puede tratar íntegramente como un objeto. Su sintaxis es muy similar a la del JAVA.

A pesar que C# forma parte de la plataforma .NET, que es una interfaz de programación de aplicaciones, C# es un lenguaje independiente que originariamente se desarrolló para producir programas sobre esta plataforma.

.NET se desarrolló debido a que Basic (visual Basic), que era uno de los lenguajes de programación encargados de

desarrollar dichas aplicaciones, es un lenguaje orientado a objetos limitado debido a que desde su creación se optó por que fuera un lenguaje fácil de aprender, así surgió el C#, para suplir esta deficiencia del Visual Basic.

C# es un lenguaje de programación orientada a objetos general para redes y desarrollo web. C# se especifica como un lenguaje de infraestructura de lenguaje común (CLI, por sus siglas en inglés). [6]

B. Metodología del desarrollo.

Se determinó que para el desarrollo del software se utilizara un método evolutivo e incremental. Los métodos evolutivos son modelos flexibles, en éstos se puede realizar modificaciones del sistema durante desarrollo; los procesos iterativos permiten realizar versiones cada vez más completas. El modelo incremental, espiral y del desarrollo concurrente son algunos modelos evolutivos.

En este caso se implementó el modelo incremental, debido a la ventaja de poder realizar incrementos progresivos de la funcionalidad del software. Por cada iteración que se tenía con el sistema se le iban agregando nuevas y mejores funcionalidades al mismo, teniendo como resultado un software óptimo y listo para su distribución.

Se puede decir que un ciclo de vida incremental es lo opuesto al ciclo de vida en cascada ya que lo que propone este método es ir entregando el software en pequeños módulos. El modelo incremental se compone de 4 fases las cuales son el análisis, el diseño, la codificación y las pruebas correspondientes.

Las ventajas de este modelo son que reduce el tiempo de desarrollo inicial, permite entregar al cliente el software más rápido en comparación con el modelo de cascada.

Otra ventaja es que resulta más fácil realizar cambios ya que provee un impacto favorable al realizar la entrega temprana de las partes operativas

El enfoque incremental de desarrollo es una forma de reducir la repetición del trabajo en el proceso de desarrollo y dar oportunidad de retrasar la toma de decisiones en los requisitos hasta adquirir experiencia con el sistema. [7]

III. RESULTADOS

A. Especificación de requerimientos.

Se realizó un análisis de todos y cada uno de los requisitos del software solicitados, separando los requisitos funcionales de los no funcionales, para obtener los requisitos del sistema fue importante conocer que es lo que el sistema debía hacer (sus funciones) y establecer un conjunto de objetivos que el sistema debía de cumplir.

Ian Sommerville expone que existen tres tipos de requisitos del software los cuales son los requerimientos abstractos del sistema, que son las funciones básicas que el software debe suministrar; propiedades del sistema que son requisitos emergentes no funcionales y por ultimo las características que el software debe mostrar (requisitos no funcionales). En este punto se utilizó el estándar de la IEEE 830-1998 para la documentación de dichos requisitos. Un requisito o requerimiento de software se define según el estándar de la IEEE como una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal. [8]

A continuación, se describen los requerimientos más importantes del software.

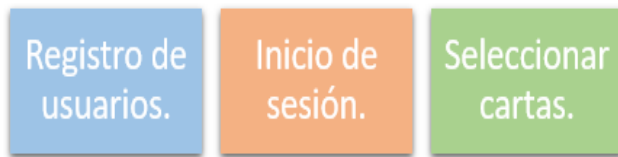


Gráfico 1. Funciones de la aplicación
Fuente: Elaboración propia.

1.- Registro de usuarios.

Los usuarios pueden registrarse en el sistema únicamente proporcionando los siguientes datos personales (nombres, apellidos, usuario, contraseña) excepto los administradores.

2.-Inicio de sesión.

Los usuarios pueden acceder al sistema proporcionando el usuario y contraseña con el cual se registraron. El sistema verifica los datos ingresados, una vez verificados y validados los datos el sistema envía un mensaje de bienvenida y dará acceso al usuario.

3.- Seleccionar cartas.

Los alumnos seleccionan dos cartas del juego al azar (si las cartas coinciden, el sistema las mantiene visibles de lo contrario las vuelve a cubrir).

B. Etapas de diseño de la aplicación.

En la figura 1, se presenta el diagrama de casos de uso que fue creado utilizando la herramienta de software orientada al lenguaje unificado de modelado ArgoUML. En dicho diagrama se observan las siguientes funciones:

- Registrar datos personales.
- Iniciar sesión.
- Seleccionar juego
- Visualizar.
- Denegar acceso
- Seleccionar cartas.

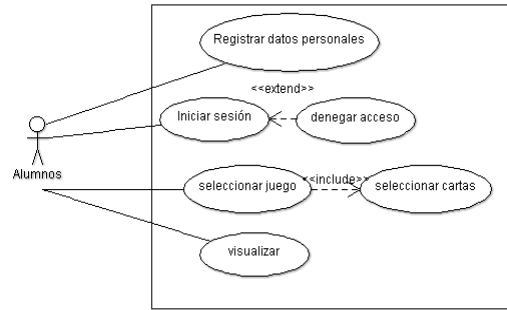


Figura 1. Diagrama de casos de uso de la aplicación.
Fuente: Elaboración propia
Entorno de desarrollo: ArgoUML.

En seguida, en la figura 2, se presenta el diagrama relacional de la base de datos de la aplicación implementado con el sistema gestor de base de datos SQLServer®.

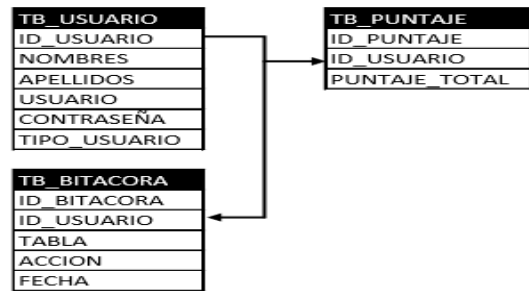


Figura 2. Esquema relacional de la base de datos.
Fuente: Elaboración propia

C. Maquetado de la Aplicación.

La elaboración del prototipo, se realizó con el software libre NinjaMock. A continuación, se muestran las principales pantallas diseñadas para la aplicación.

La figura 3 muestra dos pantallas previas al registro y la interfaz en donde el usuario ingresa los datos de entrada para realizar el registro.



Figura 3. Pantallas de inicio y registro al sistema
Fuente: Elaboración propia
Entorno de Desarrollo: NinjaMock.

D. Implementación.

Para el registro de un usuario, la aplicación solicita al usuario la información correspondiente y la almacena en la base de datos. Los datos del registro son enviados a SQLServer por medio de SqlCommand y ExecuteNonQuery, para después ser almacenadas en la tabla usuario. Para validar que el nombre de usuario no ha sido registrado por otra persona, se ejecuta una consulta (select) en busca de un nombre de usuario igual, si ésta encuentra que ya existe envía un mensaje a través de un mensaje informando que el nombre de usuario ya ha sido registrado por lo que tendrá que ser cambiado para poder realizar el registro.

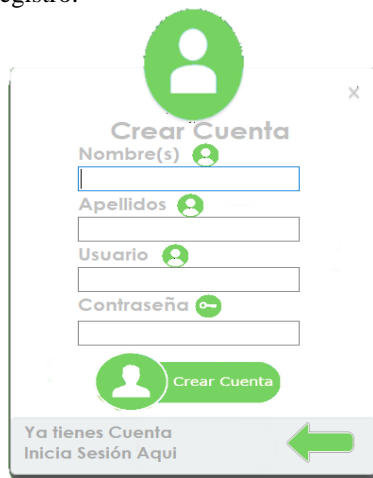


Figura 4. Pantalla de registro.
Fuente: Elaboración propia.
Entorno de Desarrollo: Visual Studio.

Una vez registrado el usuario e iniciado sesión en la aplicación, el usuario puede acceder al menú principal de la aplicación, mismo en el que se muestran las diferentes categorías de juego. En la figura 5 se muestra el menú principal de la aplicación.



Figura 5. Pantalla de menú principal
Fuente: Elaboración propia.
Entorno de Desarrollo: Visual Studio.

La aplicación cuenta con una interfaz interactiva donde se muestran las imágenes del lenguaje de señas mexicano, dicha interfaz aparece una vez seleccionada cualquier categoría del menú principal. En la figura 6 se muestra la interfaz interactiva del juego de memoria junto con las imágenes del LSM.

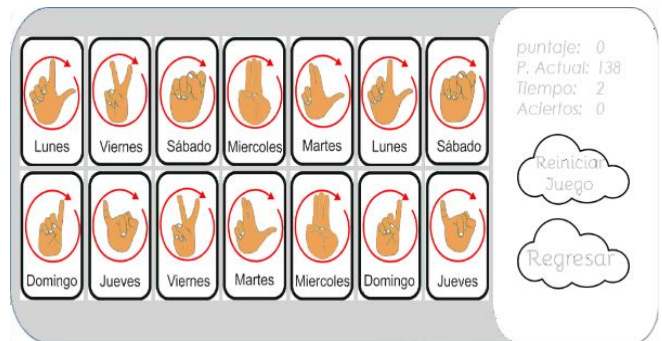


Figura 6. Interfaz del juego interactivo.
Fuente: Elaboración propia.
Entorno de Desarrollo: Visual Studio.

IV. DISCUSIÓN, CONCLUSIÓN Y RECOMENDACIONES

El software desarrollado se implementó en un laboratorio de cómputo de una escuela primaria ubicada en la ciudad de Lerdo, Dgo. Gracias a un convenio realizado entre la dirección de la escuela y el ITSL, fue posible su utilización por un periodo específico de tiempo por un grupo de 30 niños, algunos de ellos con deficiencia auditiva. Entre los resultados observados, se reportó un aumento en el interés de los menores por aprender el lenguaje de señas mexicano para comunicarse con sus compañeros con deficiencia auditiva.

El software desarrollado cumplió con el objetivo de que los niños con y sin deficiencia auditiva de la institución educativa vinculada con el proyecto, comprendieran los conceptos básicos del lenguaje de señas mexicano, por medio de la implementación computacional del juego de memoria. Al ser un software dirigido a menores de edad, el diseño de las interfaces gráficas constituyó un reto pues éstas debían ser bastante atractivas e interactivas para retener su atención, de tal modo que fueron sometidas a un refinamiento, dando como resultado las que se incluyeron en el presente artículo, que fueron de total aceptación por parte de los niños.

Se recomienda a las instituciones de educación superior fomentar la participación de los estudiantes en los concursos de innovación como el evento nacional estudiantil de innovación tecnológica (ENEIT), enfocándose no solo en las categorías de producto o servicio, sino también en aquellas dirigidas a brindar apoyo

a las personas en situación de exclusión social, como la categoría de innovación social, de la cual se derivó este proyecto. Las innovaciones en esta categoría tienen un impacto importante en los miembros de la comunidad en estado vulnerable. Este tipo de eventos abre muchas posibilidades para los jóvenes emprendedores e innovadores que quieren desarrollar nuevas y mejores herramientas tecnológicas para el bienestar de las personas.

V. AGRADECIMIENTOS

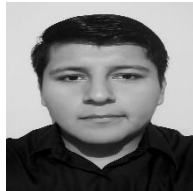
Los autores agradecen al Instituto Tecnológico Superior de Lerdo en especial a la Subdirección de Investigación y Desarrollo Tecnológico por las facilidades otorgadas para la realización de este proyecto.

VI. REFERENCIAS

- [1] Galitz, W. O. "la guía esencial para el diseño de la interfaz de usuario: una introducción a los principios y técnicas de diseño de la GUT", 3ra edición. new jersey.: Wiley.
- [2] Ceballos, F. J. (2010). Enciclopedia Microsoft Visual c#. 3ª Edición. México DF.: Ra-Ma. pp. 112 - 118.
- [3] Allison, C. L. (2004). Aprender a leer y escribir lenguaje de consulta estructurado. Indiana Estados Unidos: AuthorHouse. pp. 102 - 116
- [4] Gabillaud, J. (2015). SQL server 2014 Administracion de una base de datos transaccional. Cornellá de Llobregat Barcelona: ENI. pp 85 - 90.
- [5] Pressman, R (2010). Ingeniería del Software: Un Enfoque Práctico. 7 Ed. McGraw Hill.
- [6] Ferguson, J. (2003). La Biblia de C#. Madrid, España: Anaya. pp 78 - 95
- [7] Sommerville, Ian. (2010) Ingeniería de Software. Octava edición. Addison Wesley. Capítulos 23 y 26.
- [8] IEEE STD 830-1998. "Especificaciones de los requisitos del Software", pp. 3.

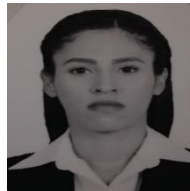
VII. BIOGRAFÍA

Aparicio Álvarez Carlos Humberto. Nació el 25 de marzo de 1996 en la ciudad de Gómez Palacio, Durango, México. Actualmente está cursando el último semestre de la carrera de ingeniería en Sistemas Computacionales con especialización en base de datos y desarrollo multiplataforma por parte del Instituto Tecnológico Superior de Lerdo en la ciudad de Lerdo, Durango, México. Hoy en día se encuentra realizando su residencia profesional con el proyecto titulado "Desarrollo de una solución tecnológica utilizando C# y SQL server, dirigida a niños con problemas auditivos" en la Subdirección de Investigación y Desarrollo Tecnológico del Instituto Tecnológico Superior de Lerdo.



Obtuvo el grado de Técnico Bachiller en Programación en el Centro de Estudios Tecnológicos Industriales y de Servicios N° 88 en la ciudad de Gómez Palacio, Durango, México (2014). Sus áreas de interés son el análisis y diseño de software, administración de bases de datos y la administración de redes.

Lucio Martínez Rosa Lea. Nació el 13 de mayo de 1994 en la ciudad de Gómez Palacio, Durango, México. Es egresada de la carrera de ingeniería Industrial con especialización en manufactura automatizada por parte del Instituto Tecnológico Superior de Lerdo en la ciudad de Lerdo, Durango, México. (2018). Elaboró una tesis titulada "Metodología de calidad para el diseño y prototipo del proyecto de innovación EnseñApp" en la Subdirección de Investigación y Desarrollo Tecnológico del Instituto Tecnológico Superior de Lerdo.



Obtuvo el grado de Técnico Bachiller en Asistente Directivo en el Colegio Nacional de Educación Profesional Técnica en la ciudad de Gómez Palacio, Durango, México (2012). Sus áreas de interés son el control de calidad y la seguridad e higiene.

Moreno Núñez Elda. Nació en Gómez Palacio, Durango, México. Es ingeniera en Sistemas Computacionales por el Instituto Tecnológico de la Laguna, en Torreón, Coah., México (1996). Obtuvo el grado de Maestría en Administración en la Universidad Autónoma de Coahuila (2012) y el de Doctorado en Desarrollo Educativo en la Universidad Autónoma de la Laguna (2018), ambas instituciones de Torreón, Coah., México.



Actualmente es Profesora Asociada 'C' en el Instituto Tecnológico Superior de Lerdo, en la ciudad de Lerdo, Durango, México, institución a la que pertenece desde 2005. Imparte cátedra en la carrera de Ingeniería en Sistemas Computacionales. Colaboró ocho años en la Gerencia Regional Cuencas Centrales del Norte de la Comisión Nacional del Agua. Sus áreas de interés son la educación, la ingeniería de software y los sistemas de información geográfica.

Meraz Salazar, Erika Alejandra. Nació en la Ciudad de Torreón, Coahuila el 14 de Febrero de 1976. Ingeniera Industrial por el Instituto Tecnológico de la Laguna en Torreón Coahuila en 1999, Maestra en Administración en Tecnologías de Información por el ITESM Campus Monterrey en el 2003, obtuvo el grado de Doctorado en Desarrollo Educativo por la Universidad Autónoma de la Laguna. Ella actualmente labora en el Instituto Tecnológico Superior de Lerdo, en la ciudad de Lerdo, Durango. Cuenta con vasta experiencia como consultora en la Industria en áreas referentes a mejora de proceso, innovación, Desarrollo Organización, Coaching y Desarrollo de Habilidades directivas, así como asesora en proyectos de innovación y desarrollo de nuevos productos. Ha impartido conferencias sobre diversos temas entre las que destacan: Imagen y Personalidad del directivo, Contratación por Competencias, PNL en la educación, Liderazgo Proactivo, Reinventándose. Las áreas de interés de ella son: Innovación, Mejora de Proceso, Coaching, Desarrollo Organización así como Inteligencias Múltiples, Programación Neurolingüística e Imagen personal. MATI. Meraz es miembro de la Asociación de Exatex, así como Evaluadora por parte de CACEL. Ha recibido diversos reconocimientos entre los que destacan: asesora en proyectos del Evento Nacional Estudiantil de Innovación Tecnológica en el 2015, 2016 y 2017 en su etapa local y regional; reconocimiento como Jurado del Evento Nacional Estudiantil de Innovación Tecnológica en su Etapa Regional y Nacional en el 2015 y 2016.

