

Software Educativo para mejorar el Desarrollo Cognitivo Conductual en Niños y Niñas entre 6 y 9 años de edad, con Trastorno de Déficit de Atención e Hiperactividad (TDAH)

Educational Software to improve Cognitive Behavioral Development in Boys and Girls between 6 and 9 years of age, with Attention Deficit and Hyperactivity Disorder (ADHD)

Jesús D. Fernández-Boscán

Universidad Rafael Urdaneta, Facultad de Ingeniería, Escuela de Ingeniería de Computación. Maracaibo, Venezuela.

 <https://orcid.org/0009-0001-3283-8698> | Correo electrónico: jesusdfer5@gmail.com

Recibido:04-10-2024 Admitido:14-10-2024 Aprobado:11-03-2024

Resumen

El presente trabajo de investigación tiene como finalidad el desarrollo de un software educativo que mejore el desarrollo cognitivo conductual en niños y niñas entre 6 y 9 años de edad con trastorno de déficit de atención e hiperactividad (TDAH). Esta investigación es de tipo proyectiva y descriptiva con modalidad proyecto factible, con un diseño no experimental. La técnica de recolección de datos aplicada fue la observación documental y la entrevista. El software educativo se desarrolló mediante la metodología SCRUM, la cual consta de tres fases: planeación, desarrollo y finalización. Así mismo, para el desarrollo del software educativo se utilizó en el Frontend la librería de React, en el Backend el entorno de ejecución de Node.js con express como framework, y como Gestor de base de datos PostgreSQL.

Palabras clave: Software educativo, TDAH, Aprendizaje.

Abstract

The purpose of this research is to develop educational software that improves cognitive behavioral development in boys and girls between 6 and 9 years of age with attention deficit hyperactivity disorder (ADHD). This research is projective and descriptive with feasible project modality, with a non-experimental design. The data collection technique applied was documentary observation and interview. The educational software was developed using the SCRUM methodology which consists of three phases: planning, development and completion. Likewise, for the development of the educational software, the React library was used in the Frontend, and in the Backend, the Node.js execution environment was used with Express as the framework and PostgreSQL as the Database Manager.

Keywords: Educational software, ADHD, Learning.

Introducción

Los seres humanos tienen múltiples formas de asimilar el conocimiento, lo que ha impulsado a la búsqueda de diferentes estrategias que permitan mejorar la calidad, cobertura e integración de la educación en los estudiantes, sin importar sus condiciones o limitaciones físicas, mentales, emocionales, económicas o de otra índole.

El Trastorno por Déficit de Atención e Hiperactividad (TDAH) o ADHD por sus siglas en inglés, es uno de los trastornos del neuro-desarrollo más prevalentes que afecta a los niños en edad escolar, definido por Scandar [1], como un trastorno neurobiológico en gran medida transmitido genéticamente que suele comenzar en la niñez y puede persistir hasta la edad adulta, reconocido por la Organización Mundial de la Salud (OMS), cuando una persona presenta desatención, hiperactividad e impulsividad comportamientos que interfieren notablemente en el funcionamiento académico y social de quienes lo padecen.

Por ello, es importante desarrollar recursos adaptados a las diferentes necesidades individuales de los niños y niñas que lo padecen. Zenteno, Carhuachín y Rivera [2], definen el software educativo como: una

herramienta de enseñanza, en forma de programa para ordenador, diseñado con el fin de ser usado como medio didáctico, para facilitar los procesos de aprendizaje y de enseñanza, lo cual abarca todos los programas que han sido diseñados con fines didácticos, desde los típicos programas basados en los modelos conductistas, hasta softwares experimentales de Enseñanza Inteligente. En este contexto, el uso de un software educativo especialmente diseñado para niños con TDAH, se presenta como una herramienta pedagógica valiosa en la educación, ya que, a través de actividades interactivas y adaptadas a su ritmo cognitivo, esta tecnología logra mejorar la motivación, imaginación, enfoque y retención de contenidos, brindando retroalimentación inmediata en su aprendizaje.

En la República Bolivariana de Venezuela, se ha avanzado en la incorporación de las Tecnologías de la Información y la Comunicación (TIC) como herramienta general de información. Sin embargo, se ha dado poco a conocer sobre las principales ventajas de los recursos técnicos como un software educativo que puede responder a la diversidad del proceso de aprendizaje, por su capacidad de adaptación a las necesidades del alumno. De esta manera, se reducen las diferencias para todos, facilitando el acceso al aprendizaje en aquellos que tienen más dificultades. Es decir, son herramientas privilegiadas para la igualdad de oportunidades en las diferentes maneras de aprender, incluyendo el déficit de atención.

El principal problema es el actual desconocimiento de los beneficios que aportan herramientas educativas como software de escritorios o aplicaciones para móviles, que se convierten en una estrategia didáctica para la enseñanza. Siendo este el caso de los estudiantes entre 6 y 9 años de edad diagnosticados con TDAH pertenecientes al aula integrada de la U.E.N. “Roberto González” del municipio La Cañada de Urdaneta, estado Zulia, donde los docentes no cuentan con un software educativo especializado que les ayude a trabajar diferentes aspectos cognitivos en estos niños. Es decir, utilizar una plataforma de ejercicios y entornos abiertos que le permitan aprender de manera diferente y divertida, pero con la intencionalidad propia de producir aprendizaje significativo según su plan de estudio.

En consecuencia, según lo antes expuesto se plantea como objetivo general desarrollar un software educativo que mejore el desarrollo cognitivo conductual en niños y niñas entre 6 y 9 años de edad con trastorno de déficit de atención e hiperactividad (TDAH) de la U.E.N. “Roberto González”, y como objetivos específicos: identificar las necesidades de aprendizaje de los niños y niñas entre 6 y 9 años de edad diagnosticados con TDAH de la institución educativa, determinar las estrategias interactivas adecuadas para el desarrollo cognitivo en los niños y niñas, diseñar la interfaz y estructura del software educativo, desarrollar un software educativo como herramienta para el desarrollo de las capacidades de aprendizaje en niños y niñas, y demostrar el funcionamiento del software educativo en niños y niñas con TDAH entre los 6 y 9 años de edad.

La razón que justifica esta investigación es la evidente problemática que presentan los planteles educativos públicos en el municipio La Cañada de Urdaneta, especialmente la U.E.N. “Roberto González”, donde los docentes carecen de alternativas tecnológicas para conseguir una evolución positiva en el desarrollo cognitivo conductual de los estudiantes. Con el uso de este software educativo se pretende la modernización del proceso de enseñanza – aprendizaje de esta población estudiantil, mejorando su prosecución, disminuyendo la probabilidad de abandono escolar, y realizar aportes que contribuyan con futuras investigaciones relacionadas al desarrollo de software educativo.

Fundamentos teóricos

Características del software educativo

De acuerdo a Zenteno, Carhuachín y Rivera, el software educativo, “busca simular la labor tutorial personalizada que realizan los docentes y muestran esquemas de representación del conocimiento en armonía con los procesos cognitivos que desarrollan los estudiantes” [2, Pág. 181].

Según Marqués[3], las cinco características que distinguen a un software educativo son las siguientes:

- Finalidad didáctica: están elaborados con una intención pedagógica y en función de unos objetivos de enseñanza.

- Uso del ordenador: no requiere de mayor explicación. Sabemos que un software es para ser explorado a través del ordenador.
- Interacción: estimulan la participación del estudiante y el intercambio de información entre el estudiante y el ordenador.
- Individualización del trabajo: le permiten al estudiante o usuario trabajar de forma individual, de acuerdo a su propio ritmo de aprendizaje.
- Facilidad de uso: los conocimientos requeridos para el uso de estos programas son mínimos. El usuario o estudiante, sólo debe seguir las instrucciones que el programa le ofrece, tanto para acceder a él, como para navegar en él.

El mismo autor expresa que existen diversos elementos que componen un software educativo, estos son:

a) Componente de comunicación o interfaz: es aquel que permite la iteración satisfactoria entre el usuario y el programa, el cual promueve la comunicación entre los mensajes, que serán entendibles al usuario, de igual modo, con el programa, este captará por medio de los dispositivos de entrada, las acciones coherentes provenientes del usuario y así generar respuesta a través de los dispositivos de salida.

b) Componente pedagógico o instruccional: determina los objetivos de aprendizaje que se lograrán al finalizar el empleo del software, los contenidos a desarrollar con el programa en función a los objetivos educacionales, las secuencias de la instrucción, los tipos de aprendizajes que se quieren lograr, sistemas de evaluación que se deben considerar para determinar los logros y los sistemas de motivación extrínseca e intrínseca que se deben introducir.

c) Componente computacional o técnico: este permite establecer la estructura lógica para la interacción del software, donde cumpla con las acciones requeridas por el usuario, así como ofrecer un ambiente al estudiante, con el cual pueda aprender lo deseado y servir de entorno. A la estructura lógica del programa, se liga íntimamente la estructura de datos, que organiza la información necesaria para que el software pueda cumplir con sus objetivos instruccionales.

Teorías del aprendizaje

García[4] afirma que el concepto básico de las teorías del aprendizaje es que la elaboración de la conducta que se observa en el curso del desarrollo puede explicarse, aparte de ciertas concesiones al crecimiento y la maduración física, como una formación continua de relaciones entre estímulos y respuestas. Existen tres tipos de teorías las cuales permiten una mejor relación y comprensión del aprendizaje, estas son:

Teoría Cognitiva: Ausubel [5] describe el aprendizaje repetitivo, que implica, la memorización de la información a aprender y el aprendizaje significativo, donde la información es comprendida por el alumno y se dice que hay una relación sustancial entre la nueva información y la presente en la estructura cognoscitiva. Plantea dos formas de cómo aprender, una de ella es por recepción, donde la información es proporcionada en su forma final y el alumno es un receptor de ella, y la otra es por descubrimiento, el alumno descubre el conocimiento y sólo se le proporcionan elementos para que llegue a él.

Teoría Conductista: Para Mergel, el conductismo se remonta a la época de Aristóteles, quien realizó ensayos de “memoria” enfocado en las asociaciones que se hacían entre los eventos como los relámpagos y los truenos [6, Pág. 3]; al respecto indica: “La teoría del Conductismo se concentra en el estudio de conductas”.

Teoría Constructivista: Pérez [7, Pág. 315] señala que “se suele justificar el término constructivismo a partir de la idea fundamental de la teoría: la consideración del hombre como constructor de su propio aprendizaje”. En esta teoría es de gran importancia la identificación del contexto, el conocimiento y la práctica, todo ello para permitir el aprendizaje y la construcción de su propio conocimiento mediante la interacción social entre los sujetos y sus pares.

Trastorno por Déficit de Atención e Hiperactividad

TDAH o ADHD por sus siglas en inglés, es un trastorno caracterizado por una evolución inapropiada en la regulación del nivel de actividad, en el control de los impulsos y en lograr un adecuado perfil atencional, tal como lo define el autor Scandar [1].

Metodología

La presente investigación es de tipo proyecto factible ya que está focalizado en el desarrollo de una propuesta para satisfacer una necesidad, o brindarle solución a una problemática siguiendo una metodología. Al respecto, Palella y Martins, señalan que el proyecto factible “consiste en elaborar una propuesta viable destinada a atender necesidades específicas, determinadas a partir de una base diagnóstica” [8, Pág. 97]. A su vez Hurtado [9, Pág. 114], define la investigación proyectiva como aquella que “intenta proponer soluciones a una situación determinada, implica explorar, describir, explicar y proponer alternativas de cambio, y no necesariamente ejecutar la propuesta.” De acuerdo a la autora esta definición se aplica a todas las investigaciones que conllevan a diseños o creaciones dirigidas a cubrir una necesidad, basadas en conocimientos anteriores.

Asimismo, para Arias la investigación descriptiva “consiste en la caracterización de un hecho, fenómeno, individuo o grupo, con el fin de establecer su estructura o comportamiento” [10, Pág. 24]. Similarmente, Según Hernández, Fernández y Baptista explican que la investigación descriptiva “busca especificar propiedades, características y rasgos importantes de cualquier fenómeno que se analice” [11, Pág. 80].

De acuerdo a lo mencionado, la presente investigación encaja perfectamente con los enfoques expuestos por los diferentes autores citados, se cataloga como proyecto factible dado que el caso de este estudio se basa en el diseño y desarrollo de un software educativo interactivo que busca satisfacer las necesidades de niños con TDAH, es del tipo descriptiva porque se especifican y caracterizan los componentes del software utilizando técnicas creativas e imaginativas para generar ideas abordando un problema práctico y obteniendo respuestas.

En cuanto al diseño de la investigación, tiene un diseño de campo pues se encarga de la recolección de datos directamente de los sujetos investigados, o de la realidad donde ocurren los hechos (datos primarios). Es importante señalar que su finalidad es teórica y no hay manipulación de las variables, por lo que se trata de un estudio no experimental. Al respecto, Palella y Martins [8], definen que la investigación de campo consiste en la recolección de datos directamente de la realidad donde ocurren los hechos, sin manipular o controlar las variables. Estudia los fenómenos sociales en su ambiente natural. El investigador no manipula variables debido a que esto hace perder el ambiente de naturalidad en el cual se manifiesta.

Por otro lado, la unidad de análisis estuvo representada por el aula integrada de la U. E. N “Roberto González” ubicada en el municipio La Cañada de Urdaneta, por ser esta la institución donde se ha tomado la muestra poblacional para desarrollar el presente proyecto. En este proyecto de investigación se inició con la observación documental basado en la lectura y análisis de diferentes fuentes bibliográficas, así como estudios realizados con anterioridad que sirvieron de base para la realización del mismo, de igual manera se formularon preguntas a los diferentes expertos, obteniendo sugerencias y recomendaciones en los diferentes temas seleccionados para trabajar en el contenido del software.

La metodología utilizada para el desarrollo del software educativo fue la metodología SCRUM, definida por Mariño y Alfonso como, “marco de trabajo iterativo e incremental para el desarrollo de proyectos y se estructura en ciclos de trabajo llamados Sprints” [12, Pág. 414]. Esta metodología permite un entorno de desarrollo controlado con un conjunto de actividades con características ágiles, flexibles que se ajustan a las necesidades de los requerimientos y que, en caso de presentarse eventualidades, son resueltas a corto plazo.

Resultados

Mediante el uso de la metodología SCRUM, las tareas que deben ordenarse en grupos denominados Sprint, lo que permite abordarlas de forma más rápida y eficiente. En el presente trabajo se han desarrollado cuatro Sprint los cuales se detallan a continuación:

Sprint 1: Planeación

Este primer Sprint da inicio al estudio, análisis y planeación del proyecto, e incluye las primeras tres tareas:

Tarea 1: Revisión de documentos y búsqueda de información. Obtención del contenido para desarrollar un material pedagógico e interactivo, adaptado a las necesidades de los niños con TDAH en edades comprendidas de 6 a 9 años de edad. Se observa la carencia de herramientas tecnológicas y se propone la modernización del proceso de enseñanza – aprendizaje de esta población estudiantil, para mejorar el rendimiento académico.

Tarea 2: Análisis donde se recopila información de requerimientos funcionales y requerimientos no funcionales para desarrollo del software. Entre los requerimientos funcionales se tienen:

- El sistema proporcionará resultados y avances de las actividades, el cual podrá ser consultado por estudiantes o docentes.
- El sistema proveerá de módulos con actividades que permitan el reconocimiento, escritura, lectura de palabras y números. Asimismo, juegos didácticos que desarrollen la memoria operativa de los niños.
- El sistema verificará los aciertos o errores en el proceso de desarrollo de cada una de las actividades.

En cuanto a los requerimientos no funcionales están:

- El tiempo de respuesta del sistema debe ser eficiente, de otra manera los usuarios perderán interés en las actividades y en la motivación hacia su uso.
- La interfaz de usuario debe ser atractiva, amigable y debe ser fácil de usar. Se sugiere la preferencia a escenarios relacionados con la naturaleza (bosques, lagos, desiertos), universo (espacio, estrellas, planetas) y fantasía.

Tarea 3: Selección de plataforma de desarrollo. Estas fueron para el Frontend la librería React, para el Backend entorno de ejecución de Node.js con express como frameworky como Gestor de base de datos se seleccionó PostgreSQL.

Sprint 2: Diseño Arquitectónico

En este Sprint se diseñan los diferentes diagramas UML los cuales permiten comprender la estructura y el comportamiento del sistema, así como las interacciones entre sus componentes. Existen varios tipos de diagramas UML, cada uno con un propósito específico.

Tarea 4: Análisis de caso de uso, muestra las interacciones entre los usuarios y el sistema, identificando los distintos escenarios de uso.

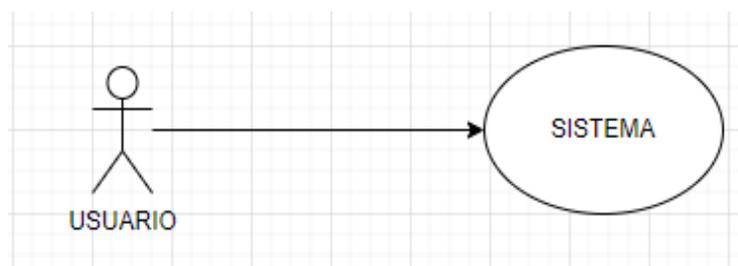


Figura 1. Diagrama general

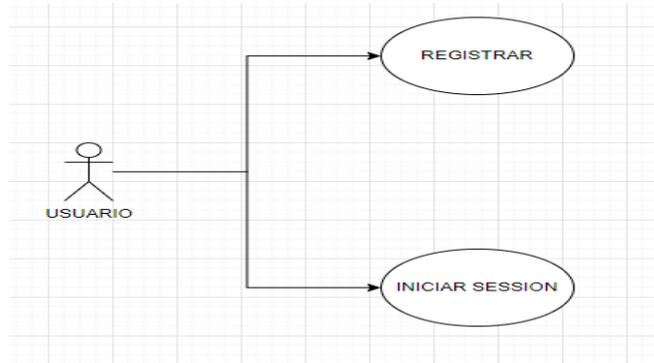


Figura 2. Diagrama principal

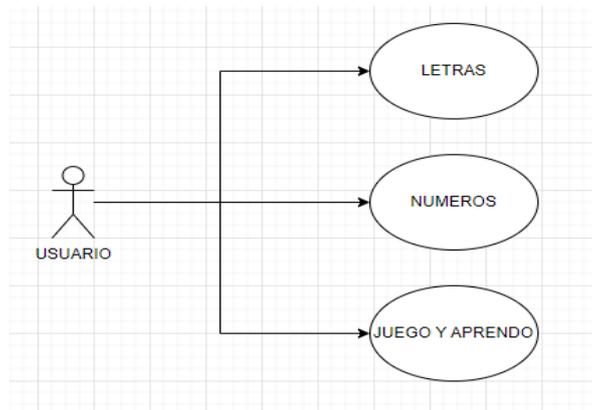


Figura 3. Diagrama del menú principal estudiante

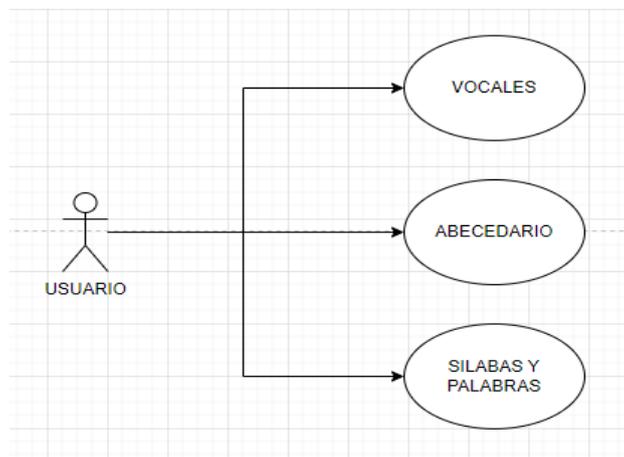


Figura 4. Diagrama del módulo de letras

Tarea 5: Diagrama de componentes de backend, este proporciona una vista de los componentes principales del sistema y las relaciones entre ellos, mostrando cómo se agrupan y se comunican entre sí.

Archivo config.json: Contiene toda la configuración del servidor. Por ejemplo, la configuración del objeto de conexión a la base de datos y el objeto de sesión.

Archivo dbSentences.js: Contiene todas las sentencias SQL del sistema.

Componente de Session: Se encarga del manejo del inicio de sesión, registro, verificación de sesión activa y cerrar sesión.

Componente Dispatcher: Tiene la función de enrutar todas las solicitudes HTTP entrantes dirigidas a la aplicación. Analiza la URL, headers, método HTTP, parámetros de la solicitud para identificar cuál es el recurso solicitado.

Componente DBComponent: Realiza la conexión y acceso a la base de datos.

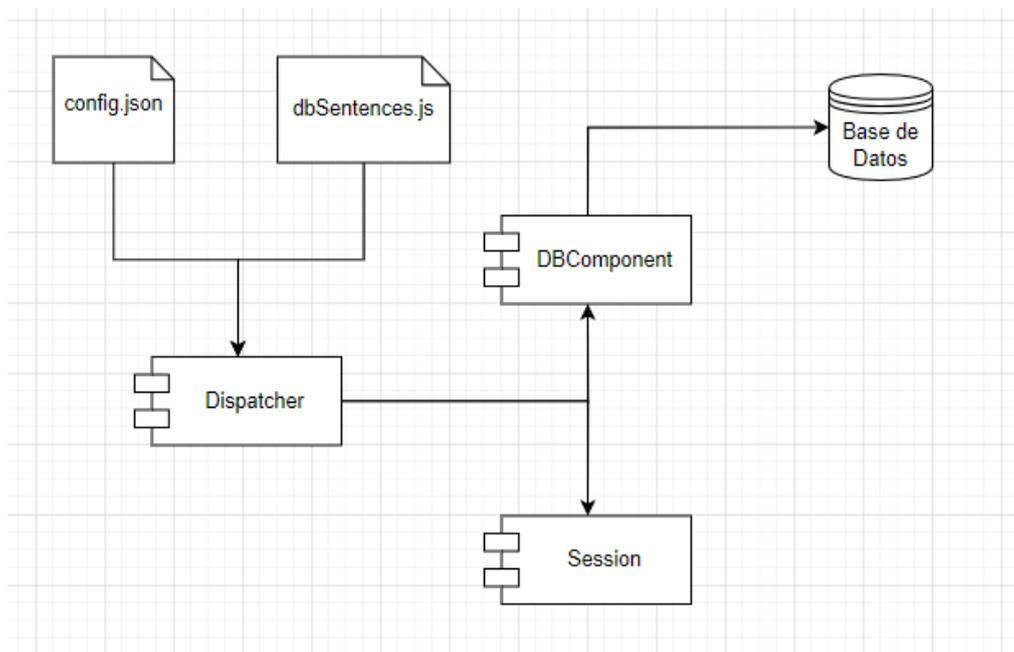


Figura 5. Diagrama de componentes del backend

Tarea 6: Diseño del modelo relacional de la base de datos: El modelo relacional permite organizar los datos de manera estructurada y establecer relaciones lógicas entre ellos.

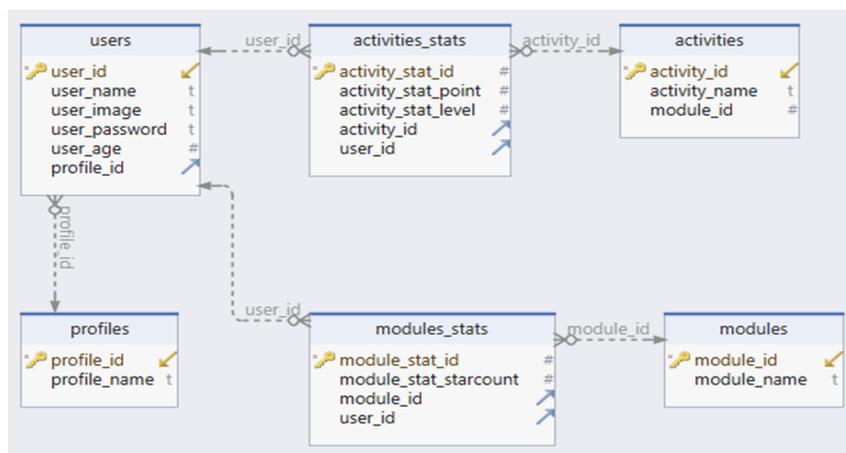


Figura 6. Modelo relacional del software educativo

Diccionario de datos:

Tabla 1. Users (Tabla de usuarios)

	Columna	Tipo de dato	Longitud	Descripción
PK	user_id	integer	32	Número de identificación para el usuario
	user_image	charactervarying	100	Imagen de perfil del usuario

	user_name	charactervarying	50	Nombre completo del usuario
FK	profile_id	integer	32	Número de identificación para el tipo de perfil
	user_password	charactervarying	100	Contraseña del usuario
	user_age	integer	32	Edad del usuario

Tabla 2. Profiles (Tabla de Perfiles)

	Columna	Tipo de dato	Longitud	Descripción
PK	profile_id	integer	32	Número de identificación del perfil
	profile_name	charactervarying	50	Nombre del perfil

Tabla 3. Activities (Tabla de Actividades)

	Columna	Tipo de dato	Longitud	Descripción
PK	activity_id	integer	32	Número de identificación de la actividad
	activity_name	charactervarying	50	Nombre de la actividad
FK	module_id	integer	32	Número de identificación para el modulo

Tabla 4. Modules (Tabla de Módulos)

	Columna	Tipo de dato	Longitud	Descripción
PK	module_id	integer	32	Número de identificación del modulo
	module_name	charactervarying	50	Nombre del módulo

Tabla 5. Activities_stats (Tabla de estadísticas de las actividades)

	Columna	Tipo de dato	Longitud	Descripción
PK	activity_stat_id	integer	32	Número de identificación para la estadísticas de la actividad
	activity_stat_point	real	24	Puntos de la actividad
	activity_stat_level	integer	32	Nivel de la actividad
FK	activity_id	integer	32	Número de identificación para la actividad
FK	user_id	integer	32	Número de identificación para el usuario

Tabla 6. Modules_stats(Tabla de estadísticas de los módulos)

	Columna	Tipo de dato	Longitud	Descripción
PK	module_stat_id	integer	32	Número de identificación para la estadísticas del módulo
	module_stat_starcoun	integer	32	Estrellas de cada módulo
FK	module_id	integer	32	Número de identificación para el módulo
FK	user_id	integer	32	Número de identificación para el usuario

Sprint 3: Desarrollo

Tarea 7: Pantalla principal del software educativo. Este diseño se muestra al ingresar al software. Aquí se presentan dos botones con diferentes opciones:

1. Registrarse: Para ser utilizado por un nuevo estudiante o docente, esta opción envía a la pantalla de registro de la aplicación.



Figura 7. Pantalla principal del software educativo

2. Iniciar sesión: Esta opción es para ingresar como un estudiante o docente ya registrado, la misma envía al usuario a la pantalla de ingreso.

Tarea 8: Pantallas de inicio de sesión del software educativo. Esta incluye el diseño de las interfaces de inicio de sesión del docente y del estudiante. Ambas pantallas muestran un botón tipo switch para la selección del tipo de usuario a ingresar en el sistema, de la misma forma un botón en la parte inferior para volver a la pantalla de inicio. La pantalla de ingreso de los estudiantes presenta dos campos para ser llenados, el primero en cual debe escribir su nombre, esto permitirá filtrar la lista de los estudiantes inscritos en el sistema, mostrando en el segundo campo las imágenes correspondientes al filtrado, donde deberá seleccionar su imagen de perfil para su respectiva validación y posterior ingreso a la aplicación.

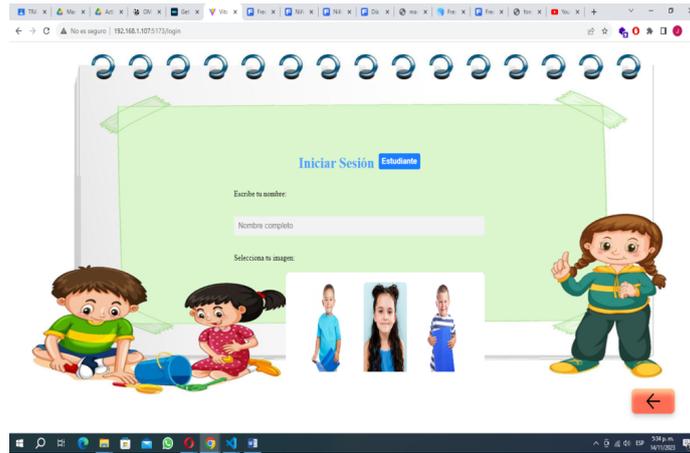


Figura 8. Pantalla de ingreso de los estudiantes al software educativo

Tarea 9: Pantallas de registro del software educativo. Esta incluye el diseño de las interfaces de registro del docente y del estudiante. Ambas pantallas muestran un botón tipo switch para la selección del tipo de usuario a registrar en el sistema, el botón registrar el cual permite guardar la información suministrada por el usuario, además de un botón en la parte inferior para volver a la pantalla de inicio. La pantalla de registro de los estudiantes presenta tres campos para ser llenados, el primero es su nombre, el segundo su edad y el tercero su imagen de perfil.

Tarea 10: Pantalla de menú principal del estudiante. En esta se puede interactuar con cada uno de los módulos de la aplicación. Se presenta el progreso del estudiante, su imagen de perfil en la parte superior izquierda y un botón para cerrar sesión en la parte superior derecha.

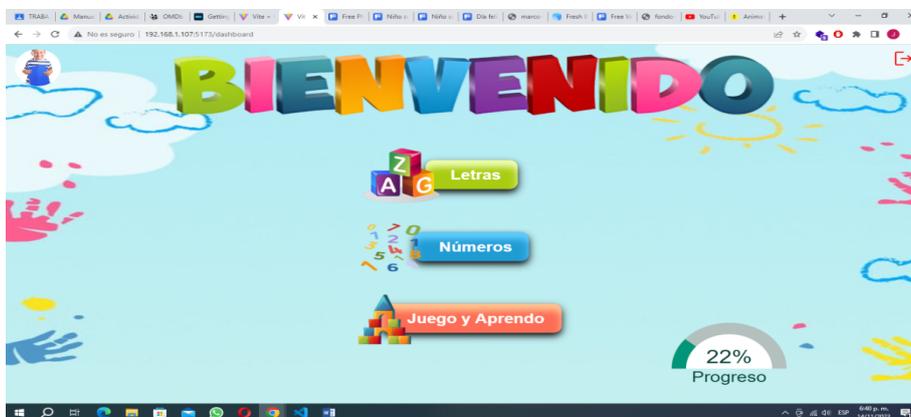


Figura 9. Pantalla de menú principal del estudiante

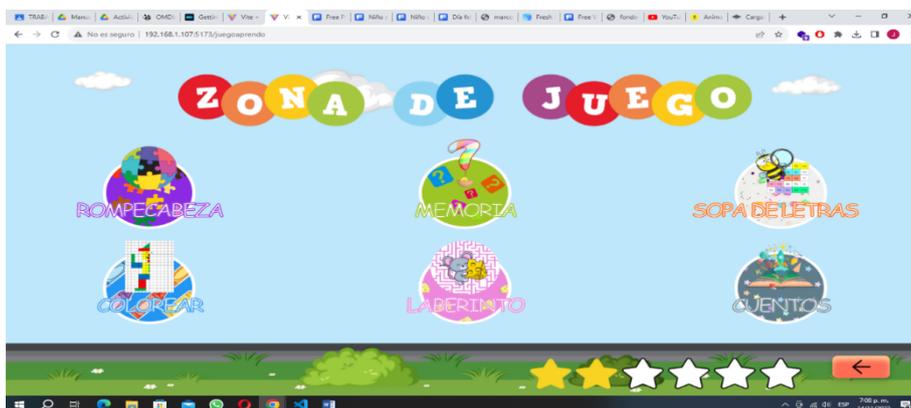


Figura 10. Pantalla de menú del módulo de juego y aprendo

Tarea 11: Pantallas de los módulos. Estas interfaces permiten acceder a las diferentes actividades de los módulos, mostrando los logros acumulados del estudiante representado por estrellas y un botón en la parte inferior para volver a la pantalla del menú principal.

Tarea 12: Pantalla de la actividad de ordenar números. Esta interfaz muestra un juego donde el usuario puede interactuar con el software educativo, el niño debe ordenar los números con ayuda del mouse o de su dedo dependiendo de su dispositivo, e intercambiar los números hasta ordenarlos de manera correcta.

Sprint 4: Finalización

Durante esta etapa, el código se prueba minuciosamente mediante pruebas unitarias que validen el comportamiento de unidades/módulos aislados de la aplicación, como funciones y clases. Una vez que se superan dichas pruebas, se aplican las pruebas de integración donde se verifican que las diferentes unidades del software se integran y comunican correctamente entre sí.

Tarea 13: Pruebas unitarias

```
describe("METHOD hasSession", () => {
  test('calls next when token is valid', () => {
    const next = jest.fn();
    Session.hasSession(req, res, next);
    expect(next).toHaveBeenCalled();
  });
  test('returns 401 when no token', () => {
    req.session.user = false;
    Session.hasSession(req, res);
    expect(res.status).toHaveBeenCalledWith(401);
    expect(res.json).toHaveBeenCalledWith({
      success: false,
      message: 'No estás autorizado'
    });
  });
});
```

Figura 11. Método hasSession de la clase Session

Tarea 14: Pruebas de integración

```
describe("GET /users", () => {
  test("should respond with a 200 status code", async () => {
    const response = await request(app).get("/users").send();
    expect(response.status).toBe(200);
  });
  test("should respond an array", async () => {
    const response = await request(app).get("/users").send();
    expect(response.body).toBeInstanceOf(Array);
  });
});
```

Figura 12. Extracto de prueba de integración del endpoint users

Conclusiones

El trabajo pudo cumplir de manera exitosa el principal propósito el cual fue desarrollar un software Educativo que mejore el desarrollo cognitivo conductual en niños y niñas entre 6 y 9 años de edad con trastorno de déficit de atención e hiperactividad (TDAH) de la U.E.N. "Roberto González", que es una herramienta valiosa para motivar el aprendizaje y reforzar conceptos clave en distintas áreas del conocimiento.

Este software fue diseñado de manera atractiva e interactiva, para ayudar a los niños a aprender de una forma divertida y estimulante, ofreciéndole al docente una alternativa innovadora. Sin embargo, es importante señalar que este tipo de herramientas tecnológicas no pueden reemplazar por completo a los métodos de

enseñanza tradicionales en el aula, sino que sirven de complemento para lograr una mejor integración entre lo presencial y lo virtual.

Del mismo modo, es necesario asegurar que el contenido educativo incluido esté debidamente revisado y adecuado por expertos en las distintas disciplinas, de manera que realmente contribuya al desarrollo cognitivo y académico infantil. Queda claro con esta investigación que el software educativo, cuando está bien diseñado e implementado, constituye una alternativa innovadora y enriquecedora para la educación de los niños.

Una vez finalizada la investigación y alcanzados los objetivos planteados, se encontraron puntos en los cuales el software puede crecer y fortalecerse como una mejor herramienta de aprendizaje. Por esto, se tienen las siguientes recomendaciones:

- Crear un módulo de idiomas que incluyan diferentes actividades.
- Permitir al docente crear contenido personalizado para los estudiantes.
- Implementar el software educativo en la institución educativa U.E.N. “Roberto González” para niños diagnosticados con TDAH.

Referencias Bibliográficas

- [1] R. O. Scandar, *Inquietos, distraídos ¿diferentes? Orientación y consejos para padres y docentes de niños con déficit de atención e hiperactividad*. Editorial EDIBA, 2006.
- [2] F. Zenteno Ruiz, M. Carhuachín, y T. Rivera Espinoza, “Uso de software educativo interactivo para la enseñanza y aprendizaje de la matemática en educación básica, Región Pasco”, *Horizonte de La Ciencia*, vol. 10, no. 19, pp. 178-190, 2020. [En Línea]. Disponible en: <https://dialnet.unirioja.es/servlet/articulo?codigo=7762164>
- [3] Per. R. Marqués Graells, “El software educativo. Una metodología para su diseño y desarrollo”, *Comunicación educativa y Nuevas Tecnologías*, pp. 119-144, 1996.
- [4] F. García, *Teorías de aprendizaje*, 2001. [En Línea]. Disponible en: <http://reescolar.ilce.edu.mx/redescolar/biblioteca/articulos/pdf/Lectura%20%20Teor%EDas.pdf>.
- [5] D. P. Ausubel, *Psicología educativa: Un punto de vista cognoscitivo*. 2a ed. Editorial Trillas, 1983.
- [6] B. Mergel, “Diseño instruccional y Teoría del aprendizaje”, Univ. de Saskatchewan, 1998. [En Línea]. Disponible en: <https://disenoinstruccion.wordpress.com/wp-content/uploads/2007/09/diseno.pdf>
- [7] P. Pérez, *Una teoría educativa Vol. II*. Ediciones UDEP, 2015.
- [8] S. Palella, y F. Martins, *Metodología de la investigación cuantitativa*. 6a ed. FEDUPEL Editores, 2017.
- [9] J. Hurtado de Barrera, *Metodología de la investigación holística*. Fundación Sypal, Caracas, 2000.
- [10] F. Arias, *El Proyecto de Investigación. Introducción a la metodología científica*. 6a ed. Editorial Episteme, Caracas, 2012.
- [11] R. Hernández, C. Fernández, y M. Baptista, *Metodología de la investigación*. McGraw-Hill, México D.F, México, 2010.
- [12] S. I. Mariño, y P. L. Alfonzo, “Implementación de SCRUM en el diseño del proyecto del Trabajo Final de Aplicación”. *Scientia Et Technica*, vol. 19, no. 4, pp. 413-418, 2014.