



“Sistema para la gestión de eventualidades en niños autistas”

**Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

FACULTAD CITEC

Autor

Abdiel Blanco Serrano

Tutores

Ing. Dargel Veloz Morales

Ing. Laritza González Marrero

La Habana, Julio 2023

DECLARACIÓN DE AUTORÍA

Declaramos ser los autores del presente trabajo de diploma titulado: “Sistema para la gestión de eventualidades en niños autistas”. Concedemos la facultad de usar los derechos patrimoniales del mismo a la Universidad de Ciencias Informáticas, autorizado su uso conformarme desde la institución.

Para dejar constancia de la declaración,

firmamos a los _____ días del mes de _____ del año 2023.

Ing. Laritza González Marrero

(Tutor)

Ing. Dargel Veloz Morales

(Tutor)

Abdiel Blanco Serrano

(Autor)

RESUMEN

En Cuba la atención primaria y especializada para pacientes con Trastorno del Espectro Autista se basa en consultas con especialistas en esta área. Durante estas consultas, participan tanto los pacientes como sus padres. La calidad de la consulta depende del seguimiento que los padres hayan realizado a sus hijos durante todo el tiempo antes de la cita. Actualmente existen dificultades con dicho proceso, ya que los padres deben memorizar o anotar las novedades de sus hijos. La investigación propone una aplicación web que a través de un Calendario realice la gestión del seguimiento de eventos de niños autistas. Para ello se establecieron los fundamentos teóricos asociados a sistemas similares que fundamentan las funcionalidades de la misma, además se hizo uso de la metodología SCRUM, y se emplearon herramientas y tecnologías como: JavaScript, Node.js, PostgreSQL y Visual Paradigm. Así mismo, se ejecutó una estrategia de pruebas para validar las funcionalidades y la seguridad del sistema. Con el desarrollo de la aplicación se espera lograr un registro efectivo de las eventualidades de los niños autistas para el correcto aprovechamiento en las consultas.

Palabras clave: Trastorno del Espectro Autista, aplicación web, gestión, registro.

ABSTRACT

In Cuba, primary and specialized care for patients with Autism Spectrum Disorder is based on consultations with specialists in this area. During these consultations, both patients and their parents participate. The quality of the consultation depends on the monitoring that parents have carried out on their children throughout the entire time before the appointment. Currently there are difficulties with this process, since parents must memorize or write down their children's news. The research proposes a web application that, through a Calendar, manages the monitoring of events of autistic children. For this, the theoretical foundations associated with similar systems were established that underlie its functionalities, in addition, the SCRUM methodology was used, and tools and technologies such as: JavaScript, Node.js, PostgreSQL and Visual Paradigm were used. Likewise, a testing strategy was executed to validate the functionalities and security of the system. With the development of the application, it is expected to achieve an effective record of the events of autistic children for correct use in consultations.

Keywords: Autism Spectrum Disorder, web application, management, registry.

AGRADECIMIENTOS

Gracias a Dios por haberme apoyado desde el principio y estar conmigo en cada momento dándome la sabiduría y las fuerzas para continuar y no detenerme hasta el final, sin él verdaderamente no lo hubiera logrado.

Gracias a mis tutores, tampoco lo hubiera logrado sin ellos los cuales me ayudaron mucho por sus consejos, correcciones, por su paciencia siempre oportunos.

Gracias a mis padres por apoyarme desde la distancia aunque hoy no pudieron estar aquí.

Gracias a mi esposa que también estuvo ahí siempre dándome su apoyo y amor en cada cosa que necesitaba, por confiar que lo lograría, que me graduaría.

Gracias a mi amiga Liz también por ayudarme y estar ahí para lo que necesitara.

Gracias a todos los que de una forma u otra pusieron su granito de arena para que esto fuera posible.

¡Muchas gracias!

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO I: Fundamentos teóricos de los sistemas asociados a la gestión del seguimiento del autismo.....	6
1.1 Principales Conceptos.....	6
1.2 Análisis de sistemas homólogos.....	8
1.3 Metodología, tecnologías y herramientas para el desarrollo de la solución.....	11
1.3.1 Metodología SCRUM.....	11
1.3.2 Lenguaje de modelado de software.....	13
1.3.3 Herramienta para el modelado de la solución.....	13
1.3.4 Tecnologías para la implementación.....	13
1.3.5 Entorno de Desarrollo Integrado.....	15
1.3.6 Herramienta para el desarrollo de la base de datos.....	15
1.3.7 Herramienta de evaluación.....	15
Conclusiones parciales.....	16
CAPÍTULO II: Análisis y diseño del sistema para la gestión de eventualidades del seguimiento realizado a niños autistas.....	17
2.1 Modelo de dominio.....	17
2.2 Requisitos del sistema.....	18
2.3 Historias de usuario.....	23
2.4 Plan de iteraciones.....	29
2.5 Diseño de la arquitectura del sistema.....	30
2.5.1 Patrones arquitectónicos.....	30
Conclusiones parciales.....	39
CAPÍTULO III: Implementación y validación del sistema para la gestión de eventualidades del seguimiento realizado a niños autistas.....	41
3.1 Estándares de codificación.....	41
3.2 Verificación y evaluación de la aplicación.....	44
3.2.1 Estrategia de pruebas.....	44
3.2.2 Pruebas unitarias.....	45
3.2.3 Pruebas de integración.....	46
3.2.4 Pruebas de validación.....	49
3.2.5 Pruebas de rendimiento.....	55
3.2.6 Evaluación del objetivo de la investigación.....	56
Conclusiones parciales.....	58
CONCLUSIONES.....	60
RECOMENDACIONES.....	61
BIBLIOGRAFÍA.....	62

ÍNDICE DE TABLAS

Tabla 1. Sistemas homólogos.....	10
Tabla 2. Requisitos Funcionales.	19
Tabla 3. Historia de Usuario No. 8: Crear evento.....	26
Tabla 4. Plan de Iteraciones.....	30
Tabla 6: Diseño de caso de prueba para validación Crear Evento.....	50
Tabla 7: Resultado de las pruebas de validación.....	55
Tabla 8. Pruebas de Cargas y Estrés.....	56
Tabla 9. Cuadro lógico de ladov.....	57
Tabla 10. Resultados de las escalas de satisfacción.....	58

ÍNDICE DE FIGURAS

Figura 1. Modelo de dominio Fuente: Elaboración propia.....	17
Figura 2. Arquitectura Modelo Vista Vista Modelo.....	31
Figura 3. Componente Categoría.....	32
Figura 4. Biblioteca store de Vuex.....	33
Figura 5. Sistema de rutas.....	33
Figura 6. Arquitectura Modelo Vista Controlador.....	34
Figura 7. Patrón singleton sequelize.....	35
Figura 8. Patrón decorador.....	35
Figura 9. Patrón middleware.....	36
Figura 10. Diagrama de clases con Estereotipos Web: Gestionar Evento.....	37
Figura 11. Modelo de datos Entidad-Relación.....	38
Figura 12. Diagrama de despliegue.....	39
Figura 14. Indentación del código.....	42
Figura 15. Comentarios.....	42
Figura 16. Modularidad.....	42
Figura 17. Try-Catch.....	43
Figura 18. Promesas.....	43
Figura 19. Respuesta satisfactoria.....	46
Figura 20. Prueba de integración. Crear evento.....	48
Figura 21. Respuesta recibida a la prueba de integración Crear evento.....	49

INTRODUCCIÓN

En la sociedad actual, las Tecnologías de Información y Comunicación (TIC) se han convertido en un instrumento fundamental para la investigación y la innovación (Dina et al. 2023). Estas han generado cambios significativos en la forma en que la sociedad interactúa y transmite el conocimiento (Pérez Zúñiga et al. 2018). Además, las TIC han permitido la creación de ecosistemas tecnológicos que facilitan la gestión del conocimiento y la colaboración entre individuos y organizaciones (Ávila 2021).

Cada vez es más considerable la influencia que tienen los avances tecnológicos en la vida de las personas, desde ámbitos cotidianos hasta profesionales, cobrando mayor importancia en ambientes laborales y educativos (Quevedo 2022). Además, las TIC se han instalado como una herramienta que bien utilizada contribuye a mejorar las condiciones de vida de los seres humanos, entre ellos los pacientes autistas (Andrade 2022).

También, “gracias a las nuevas tecnologías se consigue un acercamiento al mundo de soledad del autismo comprendiendo mejor sus estados mentales y desarrollando habilidades, que sin la interacción sujeto-medio tecnológico no es posible, e intentan alejar a la persona con autismo de su mundo de soledad” (Stephanidis, Antona 2013).

En el progreso de estas metas con apoyo tecnológico desempeñan un papel determinante los sistemas web, los cuales permiten el acceso rápido a la información y pueden ser utilizadas en diferentes dispositivos y sistemas operativos (Sevilla Robles 2020).

Los sistemas web son escalables y actualizables sin necesidad de una nueva versión de aplicación, lo que las hace una opción ideal para empresas y organizaciones (Telenchana Chimbo 2022). Por otra parte, pueden mejorar la eficiencia de los procesos y facilitar el trabajo colaborativo y a distancia (FLOREZ MENA 2019). Además, los sistemas web provocan menos errores y problemas y los datos son más seguros (Fernández et al. 2020).

Partiendo del ámbito educativo, en los últimos años, se ha generado un interés científico en la digitalización de los procesos de enseñanza y aprendizaje del alumnado con Trastorno del Espectro Autista (TEA) debido a las evidencias recogidas respecto a sus múltiples beneficios. Por esta razón, actualmente existen numerosos recursos y apoyos tecnológicos para las diferentes áreas de intervención educativa

con niños con TEA como las aportaciones de Tseng y Yi-Luen y el uso de Apps educativas (SANROMA-GIMENEZ et al., 2017).

TEA es un trastorno del desarrollo neurológico condicionante de una neurovariabilidad caracterizada por interacción social disminuida con deficiencia en el desarrollo de la comunicación a través del lenguaje verbal y no verbal e inflexibilidad en el comportamiento al presentar conductas repetitivas e intereses restringidos (Davey 2019). Por esta causa, el impacto de los que lo adolecen no se limita a la educación, sino que involucra las relaciones familiares, sociales y de salud.

En Cuba la atención primaria y especializada a pacientes con TEA está basada en las consultas con especialistas en esta área. En ellas participan los pacientes y sus padres y se evalúa el comportamiento del niño, sus interacciones sociales, sus habilidades de comunicación y su conducta; también se puede administrar pruebas que midan la audición, el habla, el lenguaje, el nivel de desarrollo y los problemas sociales; la calidad de la consulta depende del seguimiento que hayan realizado los padres a sus hijos, durante todo el tiempo previo a la consulta y la comunicación con su especialista.

En cuanto a la educación, la enseñanza especial persigue el reto de transmitir conocimientos a niños que poseen este trastorno, valiéndose de actividades diversas para obtener progresos y lidiando con muchos desafíos, entre ellos analizar y procurar la evolución positiva de cada uno.

Actualmente, la comunicación entre los padres de niños con TEA y otros actores, como los médicos especializados, maestros, familiares y otras personas en general, puede ser complicada debido a que los padres deben memorizar o anotar las novedades y progresos de sus hijos; para obtener de esta forma una guía, patrón o evidencia que le permita explicar mejor su evolución ante amistades, familiares, progresos en casa en ámbitos académicos o poner al tanto al médico que le atiende en una consulta.

Según (Guzmán et al. 2017), se han implementado alternativas que permiten mejorar la comunicación entre los padres y los médicos especializados en otros países, pero en su país esto no ha sido posible.

A partir de lo anteriormente mencionado, se puede identificar la siguiente situación problemática:

Los padres o tutores carecen de una herramienta tecnológica que les permita registrar las eventualidades de los niños autistas, provocando que se pase por alto con el devenir de los días, pequeños avances en el desarrollo de los niños autistas que son relevantes en contextos de salud, educación, familia y sociedad.

Ante el cúmulo de eventualidades de progreso de los niños autistas registrada, no se cuenta con la gestión de las mismas. Esto resulta en la incapacidad de tener disponible en los diferentes contextos las eventualidades de progreso relevantes y evidencias de las mismas.

Partiendo de esto se plantea como **problema a resolver**: ¿Cómo contribuir a la gestión de eventualidades del seguimiento realizado a niños autistas?

Como **objeto de estudio** se tienen las aplicaciones que permiten la gestión del seguimiento de los niños con TEA y como **campo de acción** las aplicaciones web para la gestión de eventualidades del seguimiento realizado a niños autistas.

Por consiguiente, el **objetivo general** es desarrollar una aplicación web que permita la gestión de eventualidades del seguimiento realizado a niños autistas.

Como **objetivos específicos** se tienen:

1. Elaborar el marco teórico de la investigación sobre soluciones informáticas asociadas a la gestión de seguimiento de eventualidades en personas con autismo.
2. Realizar el análisis y diseño de una aplicación web para la gestión de eventualidades del seguimiento realizado a niños autistas.
3. Implementar un sistema que permita la gestión de eventualidades del seguimiento realizado a niños autistas.
4. Evaluar la aplicación desarrollada.

A partir de todo lo antes definido se plantean las siguientes **preguntas científicas**:

1. ¿Cómo se gestionan las eventualidades para el seguimiento a los niños autistas?
2. ¿Cómo se puede diseñar la aplicación web de manera que sea fácil de usar y permita la gestión de eventualidades de manera intuitiva?

3. ¿Cuáles son los aspectos que caracterizan a las aplicaciones para la gestión de eventualidades del seguimiento realizado a niños autistas?
4. ¿Cómo se puede evaluar la efectividad de la aplicación web en la gestión de eventualidades del seguimiento a los niños autistas?

Para dar cumplimiento al objetivo planteado se determinaron las siguientes **tareas de la investigación**:

1. Realización de un estudio sobre los sistemas de información asociados a la gestión de seguimiento de eventualidades de personas autistas para establecer semejanzas con las soluciones existentes.
2. Realización de un estudio para la selección de la tecnología a emplear en la propuesta de solución.
3. Selección de la metodología de desarrollo.
4. Realización del análisis y diseño de la propuesta de solución.
5. Implementación de la propuesta de solución.
6. Documentación de las pruebas y validación de la aplicación.

Para el correcto desarrollo de la investigación, se emplearon los siguientes **métodos científicos**:

Métodos Teóricos:

- Histórico -Lógico: se trata de investigar y estudiar los aspectos generales del objeto de estudio excavando en el funcionamiento y desarrollo, realizando el estudio de estos para definir semejanzas con las soluciones existentes e identificar características o funcionalidades similares a las del sistema a implementar.
- Análisis-Síntesis: para realizar un estudio de la documentación recopilada sobre el tema a investigar, analizando la información en elementos más sencillos y reunir ese conocimiento para formular soluciones creativas para un sistema que permita la gestión de eventualidades del seguimiento a los niños autistas.

Métodos Empíricos:

- Observación: se pone de manifiesto en la investigación al analizar y documentar el comportamiento de niños autistas con sus padres y otros factores de la sociedad que le rodean.
- Entrevista: se realiza a padres de niños con TEA sobre acciones cotidianas relacionadas con sus hijos y la evolución de los mismos.
- Encuesta: emplear los criterios de los expertos para emplearlos en la evaluación del sistema.

Este trabajo de diploma presenta la siguiente estructura capitular:

CAPÍTULO I: Fundamentos Teóricos de los sistemas asociados a la gestión del seguimiento del autismo

Abarca los principales conceptos a tratar en la investigación. Además, el análisis de otros sistemas de información asociados a la gestión del seguimiento del autismo, también se plasma la metodología, las tecnologías y herramientas para desarrollar la solución de la investigación.

CAPÍTULO II: Análisis y diseño del sistema para la gestión de eventualidades del seguimiento realizado a niños autistas

Especifica de manera detallada el proceso de análisis y diseño de la aplicación. Se definen y describen los requisitos a implementar durante el desarrollo del sistema para la gestión de eventualidades del seguimiento realizado a niños autistas.

CAPÍTULO III: Implementación y validación del sistema para la gestión de eventualidades del seguimiento realizado a niños autistas

Análisis y evaluación del sistema para determinar la calidad de su funcionamiento con el fin de que cumpla con los requisitos planteados, para esto se aplican técnicas y herramientas para probar que el producto final cumpla con lo establecido.

CAPÍTULO I: Fundamentos teóricos de los sistemas asociados a la gestión del seguimiento del autismo

En este capítulo se presenta la fundamentación teórica a través de las bases conceptuales para comprender la investigación. Se estudian los sistemas para comprender su diseño y funcionamiento. Además, se plasma la metodología, semejantes tecnologías y herramientas para la implementación de la solución propuesta.

1.1 Principales Conceptos

El **Trastorno del Espectro Autista** (TEA) es un concepto propuesto por la psiquiatra británica Lorna Wing (1928-2014), la cual relata que el autismo es un trastorno de espectro, es decir, que involucra una variedad de síntomas, habilidades y niveles de discapacidad que puede tener quien lo padece. Después de la anterior aportación, la American Psychological Association (APA) incluye en la quinta edición del Manual Diagnóstico y Estadístico de los Trastornos Mentales (DSM-V) al autismo en la clasificación de los trastornos del neurodesarrollo, con el nombre de Trastornos del Espectro Autista (Psychiatry, 2013).

En verdad la mayoría de las investigaciones sobre los aportes al autismo se han centrado en niños, más que en las personas adultas (Hillary, Dalton 2019), debido a que el diagnóstico temprano y la intervención temprana pueden tener un impacto significativo en el desarrollo y la calidad de vida del niño.

La importancia de intervenir tempranamente radica en la posibilidad de minimizar las consecuencias de deterioro en el desarrollo del niño a largo plazo, ayudar a mejorar los resultados del niño en la comunicación, fortalecer a la familia, así como también poder disminuir los costos de atención en salud (Sazo 2022).

Los niños autistas pueden experimentar **desórdenes sensoriales** que afectan su procesamiento de estímulos sensoriales. Estos desórdenes pueden manifestarse de diferentes maneras, como hipersensibilidad o hipersensibilidad a ciertos estímulos sensoriales.

La hipersensibilidad sensorial se refiere a una respuesta intensa y exagerada a los estímulos del ambiente. Los niños hipersensibles pueden reaccionar de manera negativa o evitar ciertos estímulos sensoriales, como ruidos fuertes, luces brillantes, texturas de alimentos o sensaciones táctiles en la piel.

Por otro lado, la hipersensibilidad sensorial implica una menor sensibilidad a ciertos estímulos sensoriales. Los niños con hipersensibilidad pueden no ser conscientes de su entorno o no responder adecuadamente a situaciones peligrosas, como el dolor, el frío o el calor.

Estos desórdenes sensoriales pueden afectar la vida diaria de los niños autistas, ya que pueden experimentar dificultades para adaptarse a ciertos entornos o situaciones. Es importante para padres o cuidadores percibir estas situaciones y llevar un registro de la ocurrencia de estas eventualidades.

Según la Real Academia de la lengua española una **eventualidad** es una circunstancia o hecho de realización incierta, o fundado en el juicio que se forma de algo por indicios u observaciones.

En tanto que un **registro** es un espacio físico o virtual donde se deja constancia de un hecho (Florez Portela 2023). Esto, con el fin de que terceras personas y las autoridades competentes estén informadas al respecto.

Según Rodríguez, Martínez y Fumero, el registro es la acción de anotar un suceso, en el mundo contemporáneo suele ser una base de datos virtual (Rodríguez Hernández, Martínez Vence, Fumero Moreno 2023).

Por su parte, Rosales Vera define un registro como un documento o una entrada que contiene información específica, detallada y organizada sobre un evento, transacción, actividad o entidad (Rosales Vera 2023). Los registros son necesarios para controlar cada proceso, evidenciar que se cumplan, medir el rendimiento de los procesos y evidenciar un trabajo realizado.

En la presente investigación, un registro es la constancia de una eventualidad positiva o negativa fundada en el juicio que se forman los padres sobre la observación de su hijo en un momento dado.

Por otro lado, un **sistema de gestión** es una estructura o modelo de administración que busca mejorar el funcionamiento de una organización, conjunto de reglas o principios relacionados entre sí que contribuyen a la acción y el efecto de administrar o gestionar (Hernández 2020). Incluye un proceso de ideación, planificación, implementación y control (Lizano, Alberto 2015).

Según Mina y Lasso tradicionalmente un sistema es entendido como la interrelación mutua que se establece entre los elementos que componen un todo y que conducen al

logro de objetivos y, Gestión como el conjunto de acciones que permiten interrelacionar cada uno de los elementos con el fin de dirigir las organizaciones; al integrar estos conceptos se obtiene que un Sistema de Gestión es el conjunto de actividades que, interrelacionadas y a través de acciones específicas, permiten definir e implementar los lineamientos generales y de operación de las entidades públicas” (Mina Álvarez, Lasso Mosquera 2019).

Un software de gestión es un sistema informático integrado por múltiples herramientas que permiten planificar, organizar, administrar y automatizar diversas tareas. Su objetivo principal es aumentar la eficiencia y la productividad en los procesos y operaciones de la organización.

1.2 Análisis de sistemas homólogos

En este epígrafe se realiza la investigación y el estudio de las aplicaciones y herramientas tecnológicas nacionales e internacionales que se emplean para el seguimiento de eventualidades de niños autistas.

Choiceworks Calendar

El Calendario de Choiceworks es una herramienta de aprendizaje basada en imágenes que ayuda a los niños a aprender lo que sucede día a día, semana a semana y mes a mes durante cada año. Choiceworks Calendar enseña el concepto abstracto de tiempo en un formato visual estructurado, ayuda a los niños a organizar sus vidas y a comprender la secuencia y el tiempo. Choiceworks Calendar proporciona un calendario con todas las funciones diseñado y pensado tanto en el niño como en el cuidador (Bee Visual® 2023).

Entre las ventajas que posee esta aplicación se encuentran que los cuidadores pueden agregar sus propias imágenes usando su biblioteca de fotos, cámara o usar la biblioteca de imágenes precargada para subir su propio audio, vídeo y editar subtítulos para una personalización ilimitada. Además, pueden agregar fácilmente eventos recurrentes. Presenta funciones de búsqueda y guardado automático. También, específicamente para los cuidadores, proporciona una interfaz optimizada para crear, personalizar, compartir y mantener un calendario de próximos eventos.

Día a Día

Día a Día es una aplicación de diario visual desarrollada por la Fundación Orange y BJ Adaptaciones, especialmente pensada para personas con autismo o dificultades de

comunicación. La aplicación permite guardar y revisar las actividades realizadas durante un día, de una forma visual y estructurada, añadiendo imágenes, fotos, vídeos, etc. También es posible anticipar actividades o eventos recurrentes en el tiempo que ya han sido realizados y que se han anotado en el diario. Además, el usuario puede incluir la actividad en distintos momentos del día (mañana, tarde, noche) y describirla mediante imágenes, vídeos, sonidos, textos. El diario es personalizable y se puede adecuar a las necesidades de cada uno (*Día a Día, diario visual para personas con autismo o dificultades de comunicación*).

Autism Tracker

Autism Tracker es una aplicación de seguimiento para familias con niños autistas. La aplicación permite a los usuarios realizar un seguimiento de lo que importa para el desarrollo de su hijo autista. La aplicación cuenta con un calendario visual y gráficos de varios elementos para ayudar a organizar un día en la vida de un niño autista (Track & Share Apps, LLC).

Autismo iHelp- Clasificar

Autismo iHelp es una herramienta pedagógica para el aprendizaje de vocabulario, desarrollada por los padres de un niño con autismo y un especialista en patologías del habla y del lenguaje. Autism iHelp fue inspirada en la necesidad de contar con herramientas específicas para el tratamiento del lenguaje en niños con Trastornos del Espectro Autista, enfocándose en sus fortalezas y dificultades individuales mediante el uso de vocabulario expresivo.

Autismo iHelp es una aplicación divertida e interactiva que requiere que el niño arrastre y suelte objetos dentro de la casilla correspondiente a la categoría del objeto. Esta actividad ofrece al niño una nueva oportunidad de generalizar el vocabulario previamente aprendido a través de imágenes fotográficas de la vida real, y mejora sus habilidades de clasificación, atención dividida y procesamiento visual. Además, permite evaluar el progreso del niño a través de reportes que promedian los últimos 3 desempeños en cada actividad de aprendizaje (Autism iHelp 2018).

En el ámbito nacional, no se encontraron aplicaciones homólogas para la gestión de eventualidades o diario de niños con Trastorno del Espectro Autista.

A continuación, se presenta una tabla comparativa (ver tabla 1) entre los sistemas Choiceworks Calendar, Día a Día, Autism Tracker y Autismo iHelp- Clasificar, los

mismos se comparan en cuanto a funcionalidades, compatibilidad, costo, enfoque en necesidades especiales y soporte.

Tabla 1. Sistemas homólogos

Característica	Choiceworks Calendar	Día a Día	Autism Tracker	Autismo iHelp-Clasificar
Funcionalidad	Agregar imágenes, audios, videos, eventos recurrentes, que ayuda a los niños a aprender lo que sucede día a día, semana a semana y mes a mes durante cada año	Permite guardar y revisar las actividades realizadas durante un día, de una forma visual y estructurada.	Permite a los usuarios realizar un seguimiento de lo que importa para el desarrollo de su hijo autista	Requiere que el niño arrastre y suelte objetos dentro de la casilla correspondiente a la «categoría» del objeto, entre otras cosas.
Compatibilidad	Disponible en iOS	Disponible en Android e iOS	Disponible en iOS	Disponible en Android e iOS
Costo	Pagada	Gratuita	Ofrece una versión gratuita y una versión Premium	Gratuita
Enfoque en necesidades especiales	Proporciona un calendario con todas las funciones diseñado pensando tanto en el niño como en el cuidador	Se enfoca en mejorar la calidad de vida de las familias afectadas por el autismo	Se enfoca en mejorar la calidad de vida de las familias afectadas por el autismo	Evalúa el progreso del niño a través de reportes que promedian los últimos 3 desempeños en cada actividad de aprendizaje entre otras cosas.
Soporte	Si	No	Si	Si

Ciertamente estas aplicaciones son muy útiles en sus diferentes enfoques, pero también tienen sus limitantes. Por ejemplo, Choiceworks Calendar a pesar de que pueden agregar fácilmente eventos recurrentes, y los cuidadores pueden agregar sus

propias imágenes, existe un detalle, solo es posible acceder a ella si se compra la aplicación denotando una gran desventaja para el público cubano.

Por otra parte, la apk Día a Día al ser un diario no posee un calendario ni etiquetas personalizadas para asociar a las diferentes eventualidades. Además, no cuenta con soporte o mantenimiento.

En cambio, Autism Tracker posee un calendario visual, pero, aunque ofrece una versión gratuita, también tiene una Premium limitando a los usuarios que no pueden pagar poseer todos los privilegios de la versión pro.

Por último, está la aplicación Autism iHelp – Clasificar que a pesar de no tener el mismo objetivo que las anteriores, llama la atención el reporte de los tres últimos desempeños del niño, lo cual se convierte en una característica a tomar en cuenta para el nuevo sistema que se pretende desarrollar.

De manera general, y en adición a lo antes expuesto, estas aplicaciones al ser móviles tienen la desventaja de no funcionar en todos los sistemas operativos. Por tanto, se concluye, que ninguna de las aplicaciones antes mencionadas satisface las necesidades de los padres cubanos de llevar un registro de las eventualidades, avances o retrocesos de sus hijos a corto, mediano y largo plazo.

1.3 Metodología, tecnologías y herramientas para el desarrollo de la solución

La metodología de desarrollo de software facilita un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo de sistemas de información.

1.3.1 Metodología SCRUM

SCRUM es una metodología ágil de gestión de proyectos que se utiliza comúnmente en el desarrollo de software y otros proyectos complejos. Fue desarrollada por Ken Schwaber y Jeff Sutherland y su objetivo principal es ayudar a los equipos a trabajar de manera más eficiente y efectiva, fomentando la colaboración y la comunicación. Entre sus roles se encuentra el Product Owner como el responsable de maximizar el valor del producto y del trabajo del equipo de desarrollo; SCRUM Master como el responsable de asegurar que el equipo de desarrollo comprenda y aplique correctamente el marco de trabajo Scrum; finalmente el SCRUM Team es el grupo de personas que desarrolla el producto. SCRUM entrega el proyecto a tiempo y con un coste mínimo (Mahalakshmi & Sundararajan, 2013).

Se basa en un enfoque iterativo e incremental, lo que significa que el equipo trabaja en pequeñas entregas de valor en lugar de intentar entregar todo el proyecto al final del proceso. Está diseñada para ser bastante flexible en todo momento, proporcionando mecanismos de control para planificar el lanzamiento de un producto y gestionar las variables a medida que avanza el proyecto. Esto permite a las organizaciones cambiar el proyecto y los entregables en cualquier momento, ofreciendo la versión más adecuada. La metodología SCRUM da libertad a los desarrolladores para idear las soluciones más ingeniosas a lo largo del proyecto, a medida que se aprende y cambia el entorno. Los equipos de desarrolladores son pequeños y pueden compartir conocimientos tácitos sobre los procesos de desarrollo. La estimación sólo sirve para empezar, ya que el calendario y el coste globales se determinan dinámicamente en función de los factores del entorno (Schwaber 1997).

Fases del SCRUM (Pardo et al. 2020):

- Inicio: El objetivo de esta fase es identificar y conocer al cliente, tener una visión clara de sus deseos, justificación del negocio y los requisitos sobre los cuales trabajará el SCT. Se identifican los stakeholders, y se establece el equipo que desarrollará el proyecto, se define un acuerdo de constitución del proyecto y se especifican y garantizan los recursos que sean necesarios y que mejoren la viabilidad y ejecución del proyecto.
- Estimación y Planeación: El objetivo de esta fase es llevar a cabo la planificación de los sprints y la selección del trabajo de la pila del producto para agregarlas a la pila del sprint.
- Implementación: El objetivo es llevar a cabo las actividades que se encuentran en la pila del sprint de acuerdo con los tiempos estimados, para esto, es necesario llevar a cabo la implementación de las funcionalidades, realizar el testeo respectivo y documentar el progreso.
- Revisión y Retrospectiva: En la revisión se verifica si todavía hay tareas pendientes en la pila del producto, si es así, se deben incluir en un nuevo sprint y realizar lo que indica la fase de planificación y estimación; si no hay ítems pendientes, se completa el ciclo y se sigue la siguiente fase. Después de la revisión del sprint, se lleva a cabo la mejora del proceso como objetivo fundamental de la retrospectiva.

- Cierre: En esta fase se entrega el producto y se lleva a cabo el cierre del proyecto.
- Transversal: Esta fase consta de una serie de actividades que se llevan a cabo durante todo el proyecto, que además no tienen un tiempo específico para su realización y se pueden llevar a cabo más de una vez, esto depende del equipo de trabajo. Esta fase concentra sus actividades hacia la cultura organizacional, los valores y pilares de Scrum, por esta razón, aunque algunas de las tareas de la fase transversal son complementarias, no debe considerarse como una fase de menor importancia.

1.3.2 Lenguaje de modelado de software

Lenguaje Unificado de Modelado (UML) 2.5, UML es un lenguaje de modelado estandarizado que consta de un conjunto integrado de diagramas, desarrollado para ayudar a los desarrolladores de sistemas y software a especificar, visualizar, construir y documentar los artefactos de los sistemas de software, así como para el modelado de negocios y otros sistemas que no son de software (García-Holgado et al. 2020).

1.3.3 Herramienta para el modelado de la solución

Visual Paradigm 17.0 es una herramienta UML que está diseñada para una amplia gama de usuarios, incluidos ingenieros de software, analistas de sistemas, analistas de negocios y arquitectos de sistemas, o para cualquiera que esté interesado en construir de manera confiable sistemas de software a gran escala utilizando un enfoque orientado a objetos (Visual Paradigm 8.0. (M.T.I, 2022).

1.3.4 Tecnologías para la implementación

JavaScript consiste en un lenguaje de programación interpretado, que habitualmente se utiliza en sitios web para ejecutar acciones del lado del cliente, estando embebido en el código fuente de la página web (Armada González 2019). Este trabaja sobre un enfoque útil y práctico hace que el proceso de desarrollo de aplicaciones web dinámicas sea eficiente, está disponible en cualquier navegador web, es un lenguaje utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes orientados a objetos mucho más complejos (Santiesteban Catallops 2019).

Node.js 20.6.1 es un entorno de tiempo de ejecución de JavaScript que permite a los desarrolladores crear aplicaciones altamente escalables (Hidalgo Molina & García

Castro, 2021). Node.js se utiliza en la mayoría de los casos para construir servicios Backend, o mejor conocidos como API (Interfaces de programación de aplicaciones), estos son los servicios que dan poder a las aplicaciones web que se ejecutan dentro de un navegador o a una aplicación móvil que se ejecuta dentro de un dispositivo móvil (Hidalgo Molina, García Castro 2021).

HTTPS es un elemento destacado en Node.js, diseñado teniendo en cuenta la transmisión de operaciones con streaming y baja latencia. Esto hace que Node.js sea muy adecuado para la base de una librería o un framework web (*Index | Node.js v20.6.1 Documentation*). Sabiendo las ventajas de Node.js se usará para gestionar el servidor, aprovechando su rapidez y fiabilidad para archivos pesados y aplicaciones de carga de red pesada debido a sus enfoques basados en eventos, no bloqueantes y asíncronos que se utilizarán en el proyecto.

HTML 5 es un lenguaje de marcado hipertexto con tres características básicas: estructura, estilo y funcionalidad; producido por la combinación de HTML, CSS y Javascript para crear páginas web interactivas. Cada parte del documento está diferenciada, declarada y determinada por etiquetas específicas que indican al navegador web cómo mostrar el contenido de la página. Su versión 5 es extremadamente flexible en cuanto a la estructura y los elementos utilizados para construirla; HTML5 es la última especificación del lenguaje HTML y supuso una importante ruptura con las prácticas de marcado anteriores. El propósito de los profundos cambios introducidos en el lenguaje era estandarizar las muchas nuevas formas en que los desarrolladores lo utilizaban, así como fomentar un único conjunto de mejores prácticas en lo que respecta al desarrollo web (Gauchat 2012).

CSS3, acrónimo de Cascading Style Sheets (hojas de estilo en cascada), es un lenguaje estilístico que define cómo se presentan las páginas web. Complementa a HTML, que es el lenguaje utilizado para describir la estructura del contenido de un sitio web. Una de las principales características de CSS es que permite separar por completo la presentación de un sitio web de su contenido lo cual es una ventaja pues reduce la complejidad y repetición asociadas a la inclusión de información estilística en el contenido estructural (Olsson 2019).

Vue 2 es un framework de JavaScript para crear interfaces de usuario. Se construye sobre HTML, CSS y JavaScript estándar y proporciona un modelo de programación declarativo y basado en componentes que ayuda a desarrollar de forma eficiente interfaces de usuario, ya sean sencillas o complejas; ecosistema que cubre la mayoría

de las características comunes necesarias en el desarrollo frontend. Dicho esto, se utiliza Vue como framework para la implementación del frontend basado en su flexibilidad y adaptabilidad.

1.3.5 Entorno de Desarrollo Integrado

Visual Studio Code 1.79 es un editor de código fuente ligero pero potente que se ejecuta en el escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C++, C#, Java, Python, PHP, Go) y entornos de ejecución (como .NET y Unity) (Del Sole 2021).

1.3.6 Herramienta para el desarrollo de la base de datos

DBeaver Community 23.0.5 es una herramienta gratuita de bases de datos multiplataforma, aplicación de software cliente de SQL y una herramienta de administración de bases de datos. Admite bases de datos SQL populares como MySQL, MariaDB, PostgreSQL, SQLite. Garantiza conexiones seguras a las bases de datos. DBeaver Community es una buena opción para aplicaciones web y móviles debido a su capacidad para conectarse a diversas bases de datos, gestionar usuarios, permisos y otras configuraciones de seguridad y se integra con servicios en la nube (Gutierrez, Pablo 2022).

1.3.7 Herramienta de evaluación

Apache JMeterTM 5.5 es un software de código abierto, una aplicación Java 100% pura diseñada para probar la carga del comportamiento funcional y medir el rendimiento. Se diseñó originalmente para probar aplicaciones web, pero desde entonces se ha ampliado a otras funciones de prueba (*Apache JMeter - Apache JMeter TM*, s. f.).

Insomnia v2023.5.8 es una aplicación de escritorio de código abierto que facilita la interacción con el diseño, depuración y prueba de APIs (Interfaces de Programación de Aplicaciones). Equilibra capacidades y complejidad, crea y organiza solicitudes HTTPS. El uso de esta aplicación permitirá la ejecución de pruebas de integración del sistema web (*The Collaborative API Development Platform - Insomnia* 2023).

Conclusiones parciales

En este capítulo se sentaron las bases teóricas de la presente investigación, al estudiar los conceptos fundamentales se alcanza a una mayor comprensión del contexto de los padres, familiares y cuidadores de niños autistas. Además, con el estudio de los sistemas homólogos se evidenció que ninguno de ellos es capaz de satisfacer las necesidades existentes en el contexto cubano y finalmente, con la definición de la metodología, herramientas y tecnologías quedan sentadas las bases para la construcción de un nuevo sistema para la gestión de eventualidades de niños autistas en Cuba.

CAPÍTULO II: Análisis y diseño del sistema para la gestión de eventualidades del seguimiento realizado a niños autistas

En este capítulo se presenta el modelo de dominio y los principales objetos del sistema. Además, se definen las características y el funcionamiento de la propuesta de solución. Se identifican los requisitos funcionales y no funcionales que deben estar presentes en la solución. Conjuntamente se detalla la arquitectura, los estilos arquitectónicos y patrones utilizados y mediante el modelo de despliegue se muestra cómo queda la distribución física del sistema.

2.1 Modelo de dominio

El Modelo de dominio recoge los conceptos del negocio y sus relaciones. Según García-González y Sánchez-Sánchez (2020), el modelo de dominio es una herramienta fundamental para la identificación de los elementos clave del proceso investigativo a partir del análisis sistémico de la situación problémica y su relación con el objeto de estudio y su campo de acción (García-González et al. 2020).

El modelo de dominio (ver figura 1) muestra las clases conceptuales significativas en un dominio del problema. Se centra en las abstracciones relevantes, vocabulario del dominio e información del dominio (Losavio, Esteves 2015). Es una representación de las clases conceptuales del mundo real, no de componentes software.

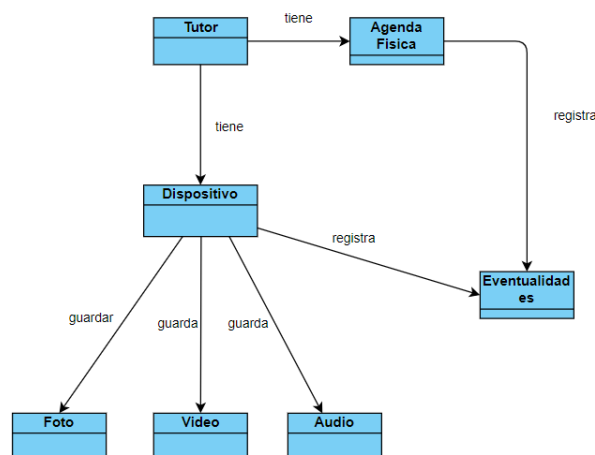


Figura 1. Modelo de dominio Fuente: Elaboración propia

Descripción de conceptos

Para aumentar la comprensión del problema y esclarecer cualquier duda sobre la terminología empleada, se describen los conceptos modelados en el Modelo del dominio mostrado anteriormente (ver figura 1).

- Tutor: persona que ejerce tutela sobre el niño(s) autista(s).
- Agenda física: se refiere a una libreta física de notas donde el tutor registra las novedades y observaciones relacionadas con el paciente autista. En ella se anotan detalles relevantes sobre su comportamiento, avances, retrocesos y cualquier otro aspecto relevante para su desarrollo.
- Eventualidades: es cualquier información relevante en las áreas de desarrollo llevadas a cabo por el niño autista. Puede incluir cambios en su rutina, reacciones inesperadas, logros o dificultades específicas.
- Dispositivo: hace referencia a un teléfono, tableta o computadora que tenga acceso el tutor.
- Foto: se trata de una imagen capturada del evento o actividad que realizó el niño.
- Audio: corresponde a grabaciones sonoras del evento o actividad realizada por el niño o simplemente una descripción por audio del evento.
- Video: se refiere a grabaciones de video del evento o actividad realizada por el niño.

2.2 Requisitos del sistema

Los requisitos describen los servicios que prestará el sistema, así como sus restricciones; reflejando, de un cliente, la necesidad de un sistema que resuelva su problema. Suelen clasificarse en requisitos funcionales (RF) o requisitos no funcionales (RnF) (Uchiha 2015).

Los **Requisitos Funcionales** (ver tabla 2): Enuncian los servicios que el sistema debe proveer, y “la especificación de los mismos debe estar completa y consistente” (Uchiha 2015)

Tabla 2. Requisitos Funcionales.

Requisitos Funcionales			
Id	Nombre	Descripción	Prioridad
1	Registrar usuario (nombre, apellido, usuario, contraseña)	El sistema le permite a un usuario registrarse.	Alta
2	Autenticar usuario (usuario, contraseña)	El sistema le permite al usuario autenticarse.	Alta
3	Listar usuario ()	El sistema le permite al usuario administrador obtener un listado de todos los usuarios.	Alta
4	Buscar usuario (nombre, apellido, usuario, email)	El sistema le permite al usuario administrador buscar otros usuarios por parámetros como: nombre, apellido, usuario y email .	Alta
5	Editar usuario (nombre, apellido, usuario, email, contraseña)	El sistema le permite a un usuario administrador editar a otro usuario.	Alta
6	Eliminar usuario (usuario)	El sistema le permite a un usuario administrador eliminar a otro usuario.	Alta

7	Cambiar Contraseña (contraseña)	El sistema le permite a un usuario cambiar su contraseña.	Baja
8	Crear evento (nombre, descripción, fecha_inicio, fecha_final, etiqueta, categoría, color, imagen, video, audio)	El sistema le permite a un usuario crear un evento.	Alta
9	Modificar evento(nombre, descripción, fecha_inicio, fecha_final, etiqueta, categoría, color, imagen, video, audio)	El sistema le permite a un usuario modificar un evento.	Baja
10	Eliminar evento (id)	El sistema le permite a un usuario eliminar un evento.	Baja
11	Listar eventos ()	El sistema le permite a un usuario listar los eventos.	Baja
12	Buscar evento (id)	El sistema le permite a un usuario buscar un evento.	Baja
13	Ver detalles del evento (id)	El sistema le permite a un usuario ver los detalles de un evento.	Alta
14	Adjuntar video a evento (file)	El sistema permite a un usuario subir un video desde su dispositivo.	Alta
15	Adjuntar audio a evento (file)	El sistema le permite a un usuario subir un audio desde su dispositivo.	Alta

16	Adjuntar foto a evento (file)	El sistema le permite a un usuario subir una imagen desde su dispositivo.	Alta
17	Crear etiqueta (name, color)	El sistema le permite a un usuario crear una etiqueta.	Alta
18	Editar etiqueta (name, color)	El sistema le permite a un usuario editar una etiqueta.	Media
19	Eliminar etiqueta (id)	El sistema permite al usuario eliminar una etiqueta.	Media
20	Listar etiquetas (name)	El sistema le permite a un usuario listar las etiquetas.	Media
21	Buscar etiqueta (name)	El sistema le permite a un usuario buscar una etiqueta.	Media
22	Buscar evento por etiqueta (name_tags)	El sistema le permite a un usuario buscar todos los eventos asignados a una etiqueta.	Media
23	Crear categoría (name, description)	El sistema le permite a un usuario crear una categoría.	Media
24	Editar categoría (nombre, descripcion)	El sistema le permite a un usuario editar una categoría.	Media

25	Eliminar categoría (id)	El sistema le permite a un usuario eliminar una categoría.	Media
26	Buscar categoría (name)	El sistema le permite a un usuario buscar categorías.	Media
27	Listar categoría ()	El sistema le permite a un usuario listar categorías.	Alta
28	Buscar evento por categoría ()	El sistema le permite a un usuario buscar todos los eventos asignados a una categoría.	Media

Los **requisitos no funcionales** son aquellos que no se refieren directamente a funciones específicas del sistema, sino que engloban otras características y/o restricciones. Abarcan mucho más que al sistema de software a desarrollar, incluye desde la política del cliente y el desarrollador hasta factores externos al proceso de desarrollo (Uchiha 2015).

A continuación, se muestran agrupados según el estándar de calidad internacional ISO/IEC 25010:2011. Este modelo de calidad se encuentra compuesto por 8 características que a su vez están divididas en subcaracterísticas, de ellas se han tenido en cuenta: (NORMAS ISO 25000, ISO/IEC 25010:2011)

RnF 1. Eficiencia de desempeño

Capacidad de desempeñar sus funciones de manera oportuna y con el uso eficiente de los recursos bajo determinadas condiciones.

- **RnF 1.1** Comportamiento temporal: El sistema debe responder rápidamente a las acciones del usuario, con un tiempo de respuesta máximo de cinco segundos por transición.

RnF 2. Usabilidad

Capacidad de ser entendido, aprendido, utilizado y atractivo para el usuario. Se tomaron en cuenta subcaracterísticas que ofrece la ISO/IEC 25010:2011.

- **RnF 2.1** El sistema debe proporcionar funciones intuitivas y de fácil comprensión para usuarios sin experiencia previa.
- **RnF 2.2** El sistema debe mostrar todos los mensajes de interacción con el usuario en el idioma español.
- **RnF 2.3** Los botones de navegación deben ser claros, grandes y coherentes.
- **RnF 2.4** Proporcionar retroalimentación para las acciones de los usuarios en la confirmación de acciones correctas y alertar a los usuarios de posibles errores.
- **RnF 2.5** Utilizar términos y nombres que resulten conocidos para todos los usuarios.
- **RnF 2.6** Utilizar textos suficientemente descriptivos y simplificados.
- **RnF 2.7** No utilizar contenidos molestos, que pueden parpadear o brillar

RnF 3. Seguridad

Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos

- **RnF 3.1** Confidencialidad: Se llevará el control del acceso, lectura y escritura a través del uso de roles que se le asignan a los usuarios.

Rnf 4. Portabilidad

Se refiere a la capacidad del software de ser transferido de un entorno a otro.

- **RnF 4.1** Adaptabilidad: El sistema debe ser capaz de adaptarse en diferentes dispositivos, dígame: PC, Tablet, Móvil.

2.3 Historias de usuario

Las Historias de Usuario (HU) son una herramienta creada por metodologías ágiles como Scrum y XP para definir los requisitos del software. Las historias de usuario desempeñan un papel clave en la redacción de los requisitos del cliente (HU end to end); y a través de tablas, se proporciona la información sencilla, clara y breve (User stories applied); necesaria para que los desarrolladores puedan estimar el esfuerzo requerido para implementarlas, estructurando de manera que puedan dividirse en tareas más pequeñas para facilitar su desarrollo en caso de ser necesario (módulo

para el registro) (Builes, Bedoya, Bedoya 2019). A continuación, la definición de los parámetros utilizados en las HU:

- **Número:** número asignado a la HU.
- **Nombre:** nombre de la HU.
- **Programador:** nombre del programador responsable de la HU.
- **Iteración asignada:** iteración a la que pertenece la HU en el plan de iteraciones.
- **Prioridad:** nivel de prioridad de la HU señalado por el cliente (Alta, Media, Baja).
- **Puntos estimados:** estimación que hace el equipo de desarrollo del tiempo necesario para implementar la HU. El 1 equivale a una semana ideal de trabajo o sea 5 días hábiles por 8 horas diarias. Por lo que 0.5 equivale a 2 días y medio, para un total de 20 horas.
- **Puntos reales:** el tiempo real en el que se realizó la HU.
- **Descripción:** breve descripción de la HU.
- **Precondiciones:** condiciones que deben ocurrir para que se desarrolle la HU.
- **Flujo básico:** camino esperado que debe seguir el usuario durante el desarrollo de la HU.
- **Flujo alternativo:** caminos alternativos que puede seguir el usuario durante el desarrollo de la HU.
- **Postcondiciones:** resultado obtenido del desarrollo de la HU.
- **Validaciones:** caracterización de los campos a validar durante el desarrollo de la HU.
- **Conceptos:** descripción de conceptos asociados a la HU.
- **Botones e hipervínculos:** botones e hipervínculos presentes en el prototipo de interfaz de la HU.

- **Requisitos especiales:** requisitos que escapan de la capacidad de respuesta de la HU.
- **Observaciones:** aspectos importantes de interés para el cliente.
- **Prototipo de interfaz:** prototipo visual del diseño esperado

Se tienen un total de 29 HU, de ellas, es presentada a continuación la HU No.8. Entre las historias de usuarios existe la historia “Crear evento” (ver tabla 3), esta refleja la creación de un evento.

Tabla 3. Historia de Usuario No. 8: Crear evento.

Historia de Usuario	
No. 8	Nombre: Crear evento
Programador: Abdiel Blanco Serrano	Iteración asignada: 4
Prioridad: Alta	Puntos estimados: 0.8
Complejidad: Media	Puntos reales: 0.9
Descripción: El sistema le permite a un usuario crear un evento.	
Precondiciones: El usuario debe de estar autenticado.	
Flujo básico:	
1. Hacer clic dentro de una franja de tiempo en el calendario. 2. Se muestra un modal que contiene todos los campos necesarios para crear un evento donde los campos que tienen un asterisco son obligatorios. 3. Llenar los campos mostrados. 4. El usuario Selecciona el botón “Crear” para finalizar la operación.	
Post-condiciones: El usuario debe rellenar todos los campos obligatorios.	
Flujo alternativo:	
Flujo alternativo 8a. 1. En caso de cancelar se cierra el modal. Post-condiciones: N/A.	
Conceptos:	
Nombre: Define el nombre dado al evento. Descripción: Define una descripción del evento para auxiliar al usuario de ser necesario.	

Fecha de inicio: Día, mes, año y hora en la que se generó el evento.

Fecha de fin: Día, mes, año y hora en la que concluyó el evento.

Etiquetas: Selecciona una o varias etiquetas del evento para auxiliar al usuario en las búsquedas y mantener un orden de los eventos.

Categoría: Selecciona una categoría a la cual pertenece el evento.

Color: Define un color para el evento.

Audio: Subir un audio del evento en caso de ser necesario.

Subir imagen: Subir una imagen del evento en caso de ser necesario.

Subir un video: Subir un video del evento en caso de ser necesario.

Validaciones:

Nombre:

- Campo obligatorio
- Campo de texto
- Cantidad de caracteres máximo es 20

Descripción:

- Campo opcional
- Campo de texto
- Cantidad de caracteres máximo es 250

Fecha inicio:

- Campo obligatorio
- Campo de fecha

Fecha fin:

- Campo obligatorio

- Campo de fecha

Etiquetas:

- Campo obligatorio

- Campo de selección múltiple

- Seleccionar una o varias etiquetas

Categoría:

- Campo obligatorio

- Campo de selección única

- Seleccionar una categoría

Audio:

- Campo opcional

- Formato del audio: .wav .mp3 .ogg

Imagen:

- Campo opcional

- Formato de la imagen: .svg, .jpg, .png

Video:

- Campo opcional

- Formato del video: .mp4, .avi, .mov

1. Si se deja un campo de los obligatorios vacío, se mostrará un mensaje de error de color rojo “Campo obligatorio” en el campo que corresponda.

2. Si se introducen más caracteres de los permitidos en un campo, el sistema no permitirá la entrada de más caracteres.

3. Si se crea un evento con igual nombre y/o fecha de inicio y fin de uno ya existente, se mostrará el mensaje “Evento ya existente”.

4. Si se introducen los datos correctamente, se mostrará el mensaje “Nuevo Evento creado satisfactoriamente”.

Botones:

- Botón Aceptar: Permite crear el evento.

- Botón Cancelar: Cancela la acción de crear el evento.

Observaciones:

Prototipo de interfaz:

El prototipo de interfaz muestra un formulario de creación de eventos con los siguientes elementos:

- Nombre ***: Campo de texto obligatorio.
- Descripción**: Área de texto grande para la descripción del evento.
- Fecha de inicio ***: Selector de fecha obligatorio.
- Fecha de fin ***: Selector de fecha obligatorio.
- Seleccionar etiquetas ***: Selector de etiquetas obligatorio.
- Seleccionar categoría ***: Selector de categorías obligatorio.
- Subir una imagen**: Botón con ícono de nube y signo más para subir una imagen.
- Subir un video**: Botón con ícono de nube y signo más para subir un video.
- #FF00FF**: Campo de color con ícono de lápiz para editar.
- Grabar audio**: Botón con ícono de micrófono para grabar audio.
- CANCELAR**: Botón gris para cancelar la acción.
- CREAR**: Botón rojo para crear el evento.

2.4 Plan de iteraciones

El plan de iteraciones en Scrum es una parte fundamental de la metodología ágil. Consiste en la planificación y organización de las iteraciones o sprints, que son períodos de tiempo fijos y cortos en los que se desarrolla y entrega un conjunto de funcionalidades o tareas.

Durante la planificación de la iteración (ver tabla 4), el equipo Scrum se reúne para definir los objetivos y el alcance de la iteración. Se revisa el backlog del producto y se seleccionan las historias de usuario o tareas que se abordarán en la iteración. También se estima la cantidad de trabajo que se puede completar en el sprint y se asignan las tareas a los miembros del equipo.

Tabla 4. Plan de Iteraciones

No.	Nombre de Historia de Usuario	Prioridad	Esfuerzo Estimado (hrs)
1	Gestionar etiqueta	Alta	8
2	Gestionar categoría	Alta	8
3	Gestionar usuario	Alta	8
4	Gestionar evento	Alta	15

2.5 Diseño de la arquitectura del sistema

El diseño de la arquitectura del sistema es la especificación del estilo arquitectónico y el patrón arquitectónico a utilizar. Según García-González y Sánchez-Sánchez (2020), el diseño de la arquitectura del sistema es fundamental para garantizar la calidad y mantenibilidad del sistema. En este sentido, se recomienda utilizar patrones arquitectónicos reconocidos para abordar problemas recurrentes (Domínguez Pérez et al. 2018).

2.5.1 Patrones arquitectónicos

Front-end framework: **VueJS** (*Documentación de Vue.js*)

El patrón **MVVM** (Modelo-Vista-Vista-Modelo) es un patrón arquitectónico utilizado en el desarrollo de aplicaciones de software. En el contexto de **Vue.js**, MVVM se refiere a la forma en que se estructuran y comunican los componentes de una aplicación.

En el patrón MVVM (ver figura 2), el **Modelo** representa los datos y la lógica subyacente de una aplicación. El **Vista** es la interfaz de usuario que muestra los datos al usuario y recibe las interacciones del usuario. El **Vista-Modelo** actúa como un intermediario entre el Modelo y la Vista, gestionando la lógica de presentación y proporcionando los datos necesarios para que la Vista los muestre.

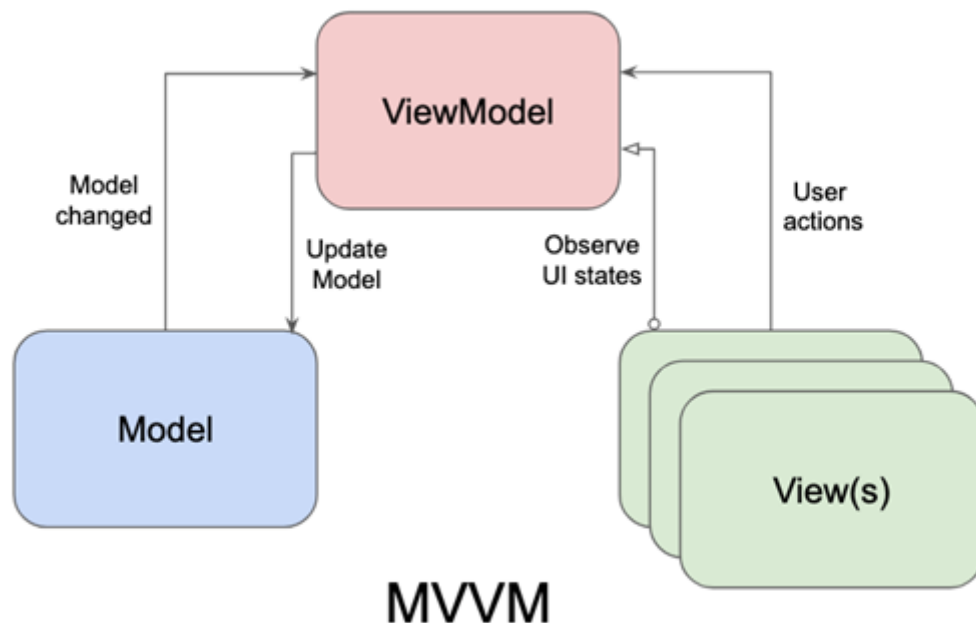


Figura 2. Arquitectura Modelo Vista Vista Modelo

Patrón de Componentes: Vue 2 se basa en el patrón de componentes, donde se construyen aplicaciones dividiéndolas en componentes reutilizables y autónomos. Cada componente tiene su propia lógica y vista (ver figura 3), lo que facilita la modularidad y el mantenimiento del código.

```

Category.vue X
src > components > category > Category.vue > {} script > default
1 <template>
2   <div class="q-pa-lg">
3     <q-timeline layout="dense" side="right" color="secondary">
4       <q-timeline-entry side="left" v-for="(category, index) in categorys" :key="index">
5         <q-card flat style="border-radius: 15px; max-width: 30%;">
6           <q-card-section class="bg-warning text-white" >
7             <div class="text-h6">{{category.name}}</div>
8             <div class="text-subtitle2">Contiene: {{ 12 }} eventos</div>
9           </q-card-section>
10          </q-card>
11        </q-timeline-entry>
12      </q-timeline>
13    </div>
14  </template>
15  <script>
16    import { mapActions, mapMutations } from "vuex";
17    export default {
18      data() {
19        return {
20          categorys: []
21        };
22      },
23      async created() {
24        await this.init()
25      },
26      methods: {
27        ...mapActions("category", ["get_category"]),
28
29        async init(){
30          const res = await this.get_category()
31          if (res.length > 0) {
32            this.categorys = res.category
33          }
34        }
35      },
36    }
37  </script>

```

Figura 3. Componente Categoría

Patrón de Estado Centralizado: Vue 2 utiliza Vuex, una biblioteca de administración de estado, que implementa el patrón de estado centralizado (ver figura 4). Este patrón permite gestionar el estado de la aplicación de manera global, lo que facilita la comunicación entre componentes y el mantenimiento del estado de la aplicación de manera consistente.

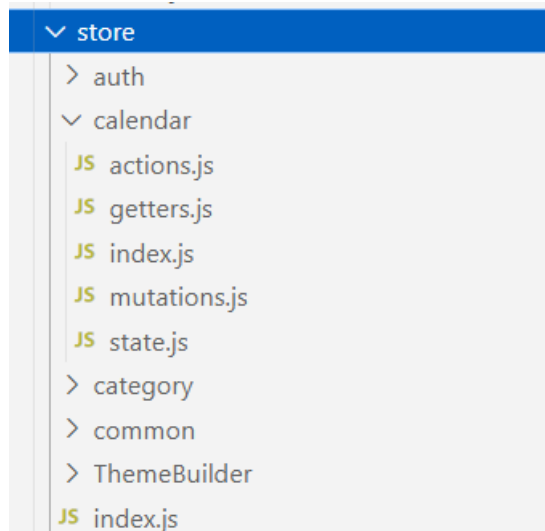


Figura 4. Biblioteca store de Vuex

Patrón de Enrutamiento: Vue 2 utiliza Vue Router para implementar el enrutamiento en las aplicaciones (ver figura 5). Este patrón permite navegar entre diferentes vistas y componentes de manera eficiente y mantener una estructura de URL coherente.

```
{
  path: '/calendar',
  component: () => import('layouts/CalendarLayout.vue'),
  children: [
    { path: '', component: () => import('pages/calendar/PageCalendar.vue') },
    { path: '/tags', component: () => import('pages/tags/PageTags.vue') },
    { path: '/category', component: () => import('pages/category/PageCategory.vue') },
  ]
},
```

Figura 5. Sistema de rutas

Back-end framework: NodeJS (Gavilanez Vaque, Ochoa Morocho 2022)

Node.js es un entorno en tiempo de ejecución multiplataforma para la capa del servidor basado en JavaScript. Está diseñado para crear aplicaciones escalables y gestionar múltiples conexiones al mismo tiempo. Node.js utiliza un bucle de eventos como una construcción en tiempo de ejecución, lo que permite una comunicación simplificada y sin bloqueos.

Modelo-Vista-Controlador (MVC) es un patrón de diseño arquitectónico utilizado para desarrollar aplicaciones web y de software (ver figura 6). Node Js utiliza este patrón, a continuación, la explicación de sus partes:

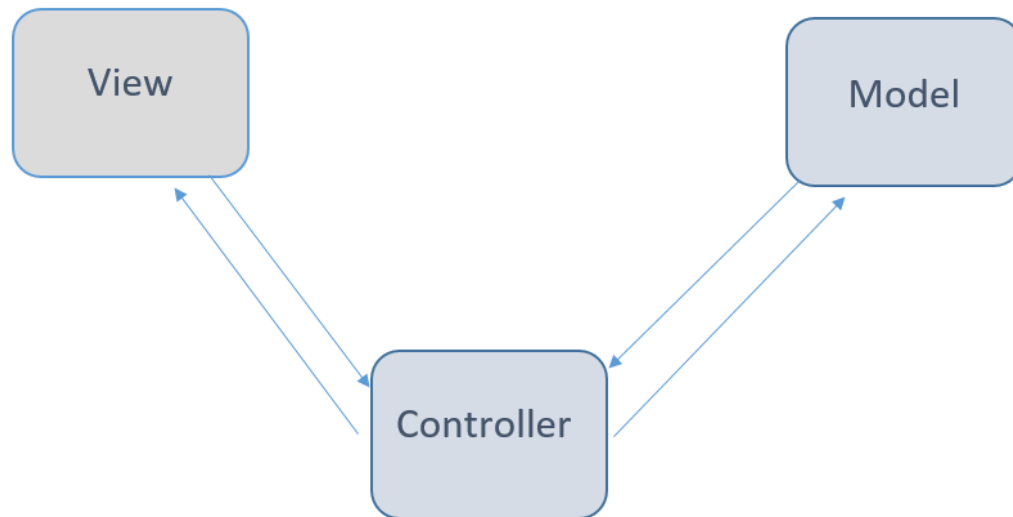


Figura 6. Arquitectura Modelo Vista Controlador

Modelo: Es el componente del backend que contiene toda la lógica de datos. Se encarga de administrar los datos y puede provenir de una base de datos, una API o un objeto JSON. En una aplicación web, el modelo es responsable de administrar los datos necesarios para la aplicación.

Vista: Es el componente del frontend o interfaz gráfica de usuario (GUI). La vista es lo que el usuario ve y con lo que interactúa. En una aplicación web, la vista se encarga de mostrar los datos al usuario.

Controlador: Es el componente que controla cómo se muestran los datos en la vista. Actúa como el cerebro de la aplicación y maneja las interacciones entre el modelo y la vista. El controlador también puede contener la lógica adicional necesaria para procesar las solicitudes del usuario y actualizar el modelo o la vista según sea necesario.

CQRS (Command Query Responsibility Segregation): Este patrón separa las operaciones de lectura (queries) de las operaciones de escritura (commands) en la aplicación. Esto permite optimizar cada tipo de operación de manera independiente y mejora la escalabilidad y el rendimiento del sistema. Es importante tener en cuenta que estos son solo algunos ejemplos de patrones arquitectónicos utilizados en Node.js y que la elección del patrón depende de las necesidades y requisitos específicos de cada proyecto.

2.5.2 Patrones de diseño GOF (Gang of Four)

Los patrones de diseño reconocidos son la descripción de los patrones utilizados para abordar problemas recurrentes y mejorar la calidad y mantenibilidad del sistema. Según The Gang of Four -GOF- los patrones de diseño son soluciones probadas para problemas recurrentes en el desarrollo de software (Guerrero, Suárez, Gutiérrez 2013).

Singleton: es un patrón de diseño creacional que permite asegurar que una clase tenga una única instancia (ver figura 7), a la vez proporciona un punto de acceso global a dicha instancia.

```
const sequelize = new Sequelize(process.env.DATABASE_URL, {  
  dialect: 'postgres',  
  protocol: 'postgres',  
  dialectOptions: {  
    ssl: {  
      require: true,  
      rejectUnauthorized: false  
    }  
  }  
});
```

Figura 7. Patrón singleton sequelize

Decorator: es un patrón de diseño estructural que te permite añadir funcionalidades a objetos (ver figura 8) colocando estos objetos dentro de objetos encapsuladores especiales que contienen estas funcionalidades.

```
@param {*} req |  
@param {*} res
```

Figura 8. Patrón decorador

Middleware: es un patrón de diseño que se utiliza para facilitar la comunicación entre componentes de un sistema (ver figura 9). Actúa como un intermediario entre los componentes, permitiendo que estos se comuniquen de manera más eficiente y que sean más fáciles de mantener y actualizar. El Middleware se puede utilizar para integrar diferentes componentes de un sistema que utilizan diferentes lenguajes de programación o plataformas tecnológicas.

```
var express = require("express");  
var cors = require("cors");
```

Figura 9. Patrón middleware

2.6 Diagrama de clases del diseño

Los diagramas de clases del diseño permiten modelar la vista de diseño estática de un sistema, estos describen gráficamente las especificaciones de las clases además de visualizar las relaciones entre estas. Estos diagramas son una herramienta fundamental en el proceso de diseño de software, ya que permiten visualizar la estructura del sistema y sus componentes, lo que facilita la identificación de problemas y errores en el diseño (Mar Cornelio et al. 2020).

En un diagrama de clases de diseño (ver figura 10), se pueden representar diferentes aspectos del sistema, como las relaciones entre las clases, los atributos y métodos de cada clase, las interfaces que utiliza cada clase, etc. Estos diagramas emplean los estereotipos web, que son herramientas utilizadas para representar de manera más específica los elementos de una aplicación web.

Estos estereotipos ayudan a identificar y diferenciar el código que se ejecuta en el servidor, el código que se ejecuta en el cliente y los formularios utilizados en la aplicación web.

- **Client Page:** página mostrada al cliente, a través de la cual se interactúa con el sistema.
- **Form:** formulario donde el cliente introducirá la información necesaria al sistema.
- **Server Page:** encargada de manejar las peticiones de la página cliente y de construirle al usuario la página cliente con la que este desea interactuar.
- **Event:** representa la entidad Event y contiene sus datos.
- **EventController:** contiene los métodos crear, listar, editar y eliminar.

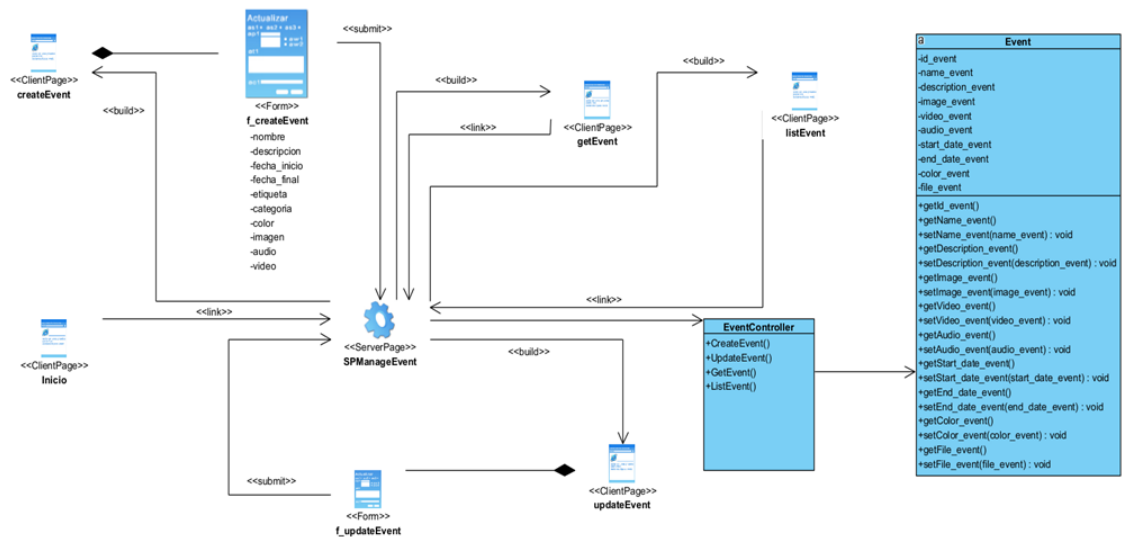


Figura 10. Diagrama de clases con Estereotipos Web: Gestionar Evento.

Fuente: Elaboración propia.

2.7 Modelo de datos

Un modelo de datos es una colección de conceptos que se emplean para describir la estructura lógica de una base de datos. Los modelos de datos definen con claridad cómo se modela la estructura lógica de una base de datos. El modelo de datos a utilizar es el Modelo Entidad-Relación (MER).

El MER es un modelo de datos de alto nivel, fue propuesto en 1976. Está basado en una percepción de un mundo real que consiste en una colección de objetos básicos, denominados entidades, y de relaciones entre estos objetos, así como las características de estos objetos llamados atributos (Gómez et al. 2017).

El modelo Entidad-Relación (ver figura 11) es una herramienta útil para diseñar bases de datos relacionales de este tipo, ya que permite representar de manera simplificada cómo personas, objetos o conceptos se relacionan entre sí.

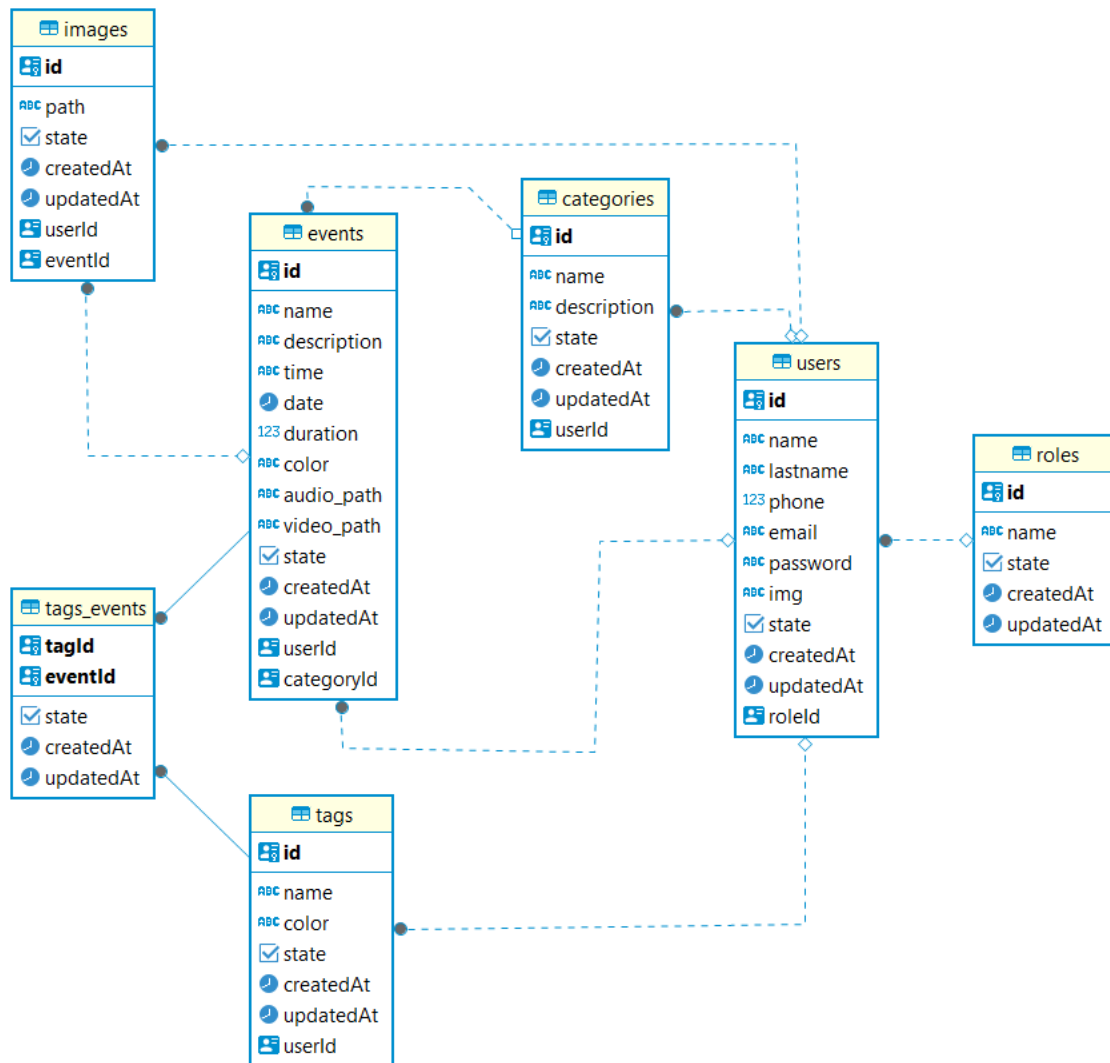


Figura 11. Modelo de datos Entidad-Relación.

Fuente: Elaboración propia.

2.8 Diagrama de despliegue

Un diagrama de despliegue es un tipo de diagrama UML que se utiliza para visualizar los procesadores, nodos o dispositivos de hardware de un sistema, los enlaces de comunicación entre ellos y la colocación de los archivos de software en ese hardware. En otras palabras, el diagrama de despliegue muestra cómo se despliega un sistema en el hardware.

En un diagrama de despliegue (ver figura 12), los nodos representan los recursos físicos, como servidores, computadoras y dispositivos móviles, mientras que los componentes representan las partes del software que se ejecutan en esos nodos. Los artefactos son elementos concretos que son causados por un proceso de desarrollo.

Ejemplos de artefactos son las bibliotecas, archivos, archivos de configuración, archivos ejecutables, etc. (Sommerville 2005).

- PC cliente: Los resultados muestran que la usabilidad de la propuesta de solución es superior en cada criterio evaluado, reconociendo su aprobación por parte el cliente.
- Servidor de aplicación web: Representa el servidor donde se encuentra instalada la aplicación web. Este accede al servidor de Base de Datos para el manejo de la información mediante el protocolo TCP/IP.
- Servidor BD: Representa el servidor donde se encuentra instalada la aplicación web. Este accede al servidor de Base de Datos para el manejo de la información mediante el protocolo TCP/IP.
- HTTPS: Protocolo de transferencia utilizado para conectar la computadora del cliente con el servidor donde está el sistema.
- TCP/IP: Protocolo de transferencia utilizado para conectar la computadora del cliente con el servidor donde está el sistema.

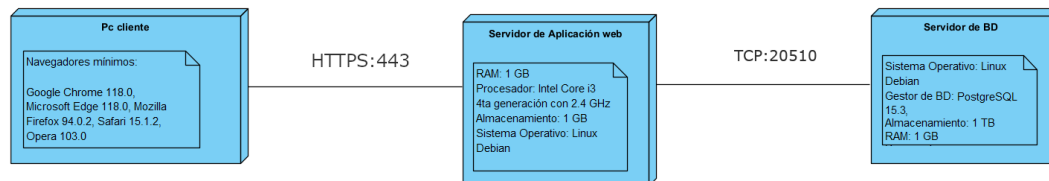


Figura 12. Diagrama de despliegue.

Fuente: Elaboración propia

Conclusiones parciales

Los principales conceptos asociados al modelo de dominio permitieron representar el vocabulario y los conceptos claves del dominio del problema. Con el establecimiento de los requisitos funcionales y no funcionales se esclarecieron las características a satisfacer en la propuesta de solución planteada. Bajo la guía de la metodología de desarrollo Scrum, se analizaron y realizaron las historias de usuario como principal artefacto, lo que resultó en la obtención de los prototipos no funcionales del sistema.

La definición de la arquitectura del sistema mediante el patrón arquitectónico MVC en el backend y el patrón MVVM en el frontend permitió delimitar las vistas, donde se construyen las interfaces gráficas de usuarios, la lógica del negocio para la definición de los controladores que gestionan las operaciones de los componentes, y el modelo para la persistencia de la información. Finalmente, se elaboró y describió el diagrama de clases y el modelo de datos que soporta la persistencia de la información en el sistema.

CAPÍTULO III: Implementación y validación del sistema para la gestión de eventualidades del seguimiento realizado a niños autistas

En el capítulo se presenta la definición de los estándares de codificación que debe tener en cuenta el desarrollador durante la etapa de implementación del software. Además, se mostrará una descripción de las pruebas empleadas para la evaluación de la herramienta desarrollada, presentando a su vez los resultados obtenidos durante dicho proceso.

3.1 Estándares de codificación

Los estándares de codificación son un conjunto de convenciones establecidas de antemano (denominaciones, formatos, etc.) para la escritura de código. Estos estándares varían dependiendo del lenguaje de programación elegido y además varían en cobertura, algunos son más extensos que otros (Patilla et al. 2021).

Durante la etapa de implementación del software, es fundamental que el desarrollador siga estándares de codificación para garantizar la calidad y la eficiencia del código. Estos estándares pueden incluir pautas sobre la estructura del código, la nomenclatura de variables y funciones, el manejo de errores, entre otros aspectos. Seguir estos estándares ayuda a mejorar la legibilidad, la mantenibilidad y la escalabilidad del software.

A continuación, se muestran los estándares utilizados para la correcta y organizada implementación del código:

1. **Convención de Nomenclatura:** Se utilizan nombres descriptivos para variables (ver figura 13), funciones y módulos. Además, se usa el estilo camel case para las variables y funciones (ejemplo: nombreVariable, nombreFuncion) y pascal case para los módulos (ejemplo: NombreModulo).

```
addOrUpdateCategory: {  
  type: Object,  
  default: () => ({ name: '', decription: '' })  
}
```

Figura 13. Nomenclatura de variables.

2. **Indentación:** Se utiliza una indentación (ver figura 14) consistente de 4 espacios para mejorar la legibilidad del código.

```

data() {
  return {
    tags: {
      name: '',
      color: ''
    }
  };
},

```

Figura 14. Indentación del código.

3. **Comentarios:** Se incluyen comentarios descriptivos (ver figura 15) para explicar el propósito y la funcionalidad del código. Se utilizan comentarios en línea (//) , comentarios de bloque (/* */) o comentarios en html (<!-- -->) según sea necesario.

```

//Dialog Metods
show() {
  this.$refs.dialog.show()
},

```

Figura 15. Comentarios.

4. **Modularidad:** Se divide el código en módulos reutilizables (ver figura 16) y separados por funcionalidad. Se utiliza el sistema de módulos de Node.js (require y module.exports) para importar y exportar funcionalidades entre módulos.

```

module.exports = {
  userGet,
  userPost,
  usersPut,
  usersDelete,
};

```

Figura 16. Modularidad.

5. **Manejo de Errores:** Se implementa un manejo adecuado de errores utilizando bloques try-catch para capturar y manejar excepciones (ver figura 17). Se utiliza el objeto Error para lanzar y personalizar errores.

```
try {
  const [length, user] = await Promise.all([
    User.count({ where: query }),
    User.findAll({
      where: query,
      offset: Number(offset),
      limit: Number(limit),
    }),
  ]);
  res.json({ length, user });
} catch (error) {
  console.log(error);
  res.status(500).json({
    msg: "Talk to the administrator",
  });
}
```

Figura 17. Try-Catch.

6. **Uso de Promesas o Async/Await:** Se utilizan promesas o async/await para manejar operaciones asíncronas de manera más legible y evitar el anidamiento excesivo de callbacks.

Las promesas (ver figura 18) son una biblioteca para programación asíncrona, una representación de primera clase de un valor que puede estar disponible en el futuro. Las promesas se utilizan en muchas bibliotecas de JavaScript existentes. Tiene como ventaja permitir un manejo fácil de las operaciones asíncronas (posibilitan que el programa continúe ejecutando otras tareas mientras espera la finalización de una tarea asíncrona) en una aplicación.

```
async created() {
  await this.init()
},
```

Figura 18. Promesas.

3.2 Verificación y evaluación de la aplicación

Es importante realizar pruebas exhaustivas para evaluar la herramienta desarrollada. En la presente investigación se incluyen pruebas unitarias, pruebas de integración, pruebas de rendimiento y pruebas de aceptación. Las pruebas permiten identificar posibles errores o fallos en el software y asegurarse de que cumple con los requisitos y las expectativas establecidas.

Durante el proceso de evaluación, se obtienen resultados que proporcionan información sobre el rendimiento, la estabilidad y la calidad del software. Estos resultados son fundamentales para tomar decisiones informadas sobre posibles mejoras o ajustes necesarios en el software.

3.2.1 Estrategia de pruebas

Una estrategia de pruebas de software es un plan que describe los pasos que se deben seguir para probar el software. Este plan incluye la planificación de la prueba, el diseño de casos de prueba, la ejecución de la prueba y la recolección y evaluación de los resultados (ver tabla 5). También se especifica cuánto tiempo, esfuerzo y recursos se necesitan para llevar a cabo estas pruebas (PRESSMAN, LOWE 2008)

Tabla 5. Estrategia de pruebas.

Pruebas	Método	Técnica
Pruebas unitarias	Caja blanca	Aislamiento y Enfoque
Pruebas de integración	Caja negra	Incremental
Pruebas de validación	Caja negra	Partición equivalente
Pruebas de rendimiento	Carga y estrés	Pruebas de carga

3.2.2 Pruebas unitarias

Las **pruebas de unidad** son una técnica de pruebas de software que se enfoca en evaluar el comportamiento de una unidad de código, como una función o un método, para asegurarse de que funciona correctamente. Estas pruebas se realizan generalmente por los desarrolladores y se centran en probar el código desde un punto de vista interno, es decir, sin tener en cuenta la estructura interna del sistema a probar. En lugar de eso, se evalúa el comportamiento del sistema a través de sus entradas y salidas.

El proceso para ejecutar este tipo de pruebas es el siguiente (Hidalgo Peña 2019):

- Analizar los requisitos y sus especificaciones.
- Seleccionar entradas válidas y no válidas de acuerdo con las especificaciones.
- Determinar las salidas esperadas para cada entrada.
- Diseñar los casos de pruebas (ver tabla 5 y 6) con las entradas seleccionadas.
- Ejecutar los casos de prueba.
- Comparar las salidas encontradas con las salidas esperadas.
- Determinar si el funcionamiento del software en prueba es apropiado

Aislamiento y Enfoque

Por una parte, en las pruebas unitarias, **aislar** significa evaluar una **unidad de código** (como una función o método) de manera **independiente**. El objetivo es verificar que esta unidad funcione correctamente sin depender de otras partes del sistema.

Por otra parte, las pruebas unitarias deben centrarse en **componentes individuales**. Se evalúa cada unidad de código por separado, sin considerar su contexto completo. Esto permite detectar errores específicos en esa unidad sin verse afectado por otros componentes.

En la realización de las pruebas se utilizó la herramienta Insomnia (ver figura 19) comprobando el correcto funcionamiento de los componentes independientes del backend.



Figura 19. Respuesta satisfactoria

Esta prueba realizada a la funcionalidad Crear evento se divide en 3 partes:

- 1- La respuesta de la solicitud debe ser correcta.
- 2- La carga útil de la respuesta debe ser válida, en este caso debe ser un objeto que dentro contenga un evento.
- 3- El evento debe ser correcto.

Como se muestra en la figura 19 la respuesta a la prueba es correcta. El mensaje de notificación “Tests Passed 1/1” indica que la ejecución fue satisfactoria. Además, se muestra el tiempo de ejecución de la prueba que es 1593 ms.

3.2.3 Pruebas de integración

Las pruebas de integración evalúan si los componentes individuales trabajan en conjunto de manera esperada. En otras palabras, se verifica el funcionamiento de los diferentes módulos del sistema una vez que se unen o agrupan en elementos mayores, comprobando su comportamiento frente a las comunicaciones entre ellos. El objetivo es localizar errores de interfaz y asegurar el correcto funcionamiento en conjunto de los componentes (Gómez Rodríguez 2015).

La integración incremental es un enfoque que, en lugar de construir y probar un programa en una sola etapa masiva, se divide el proceso en pequeños incrementos. Cada incremento se construye y verifica por separado. Esto tiene varias ventajas:

1. Aislamiento de Errores: Al trabajar con incrementos pequeños, los errores son más fáciles de identificar y corregir. No se ven afectados otros componentes del sistema.

2. Pruebas Exhaustivas de Interfaces: Cada incremento se somete a pruebas exhaustivas, lo que aumenta la confianza en las interfaces entre módulos.
3. Enfoque Sistemático: La metodología incremental permite aplicar un enfoque sistemático, en este se construye una base sólida antes de agregar nuevas funcionalidades.

En resumen, la integración incremental es una estrategia efectiva para desarrollar software de manera controlada y confiable. Para la realización de estas pruebas se utilizó la herramienta Insomnia para equilibrar capacidades y complejidad, crear y organizar solicitudes HTTPS.

A continuación, se muestra un ejemplo de prueba de integración realizada en la herramienta Insomnia. La información de prueba proporcionada (ver figura 20) es de una solicitud POST realizada al endpoint `/api/event`. Los principales componentes de la solicitud son:

Método de solicitud: El método de solicitud utilizado es POST, lo que indica que el cliente desea crear un nuevo recurso en el servidor.

URI de solicitud: El URI de la solicitud es `/api/event`.

Versión de HTTPS: La versión de HTTPS utilizada es HTTPS/1.1.

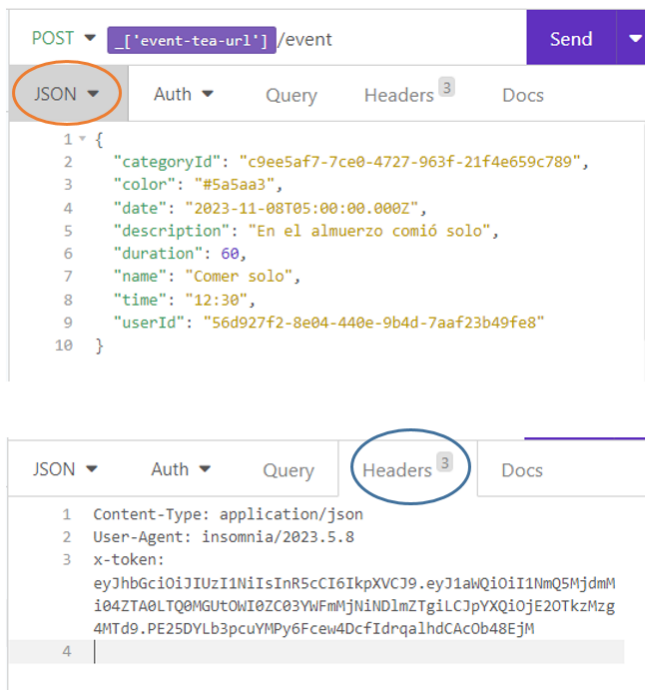
Content-Type: El tipo de contenido de la carga útil de la solicitud es `application/json`.

User-Agent: El agente de usuario que realiza la solicitud es `insomnia/2023.5.8`.

x-token: La solicitud incluye un encabezado x-token con un valor de token.

Content-Length: La longitud de la carga útil de la solicitud es de 270 bytes.

La carga útil de la solicitud (body o payload) contiene datos JSON con las siguientes propiedades de un evento: categoría del evento, el color, la fecha, la descripción, la duración, el nombre, la hora y el ID de usuario asociado con el evento.



■ Carga útil enviada por el usuario en formato JSON al backend.

■ Token del usuario enviado al backend en la Cabecera de la solicitud.

Figura 20. Prueba de integración. Crear evento.

La respuesta del backend enviada al frontend, correspondiente a la prueba anterior (ver figura 21) evidencia que el módulo Node.js Express ha creado un evento satisfactoriamente y devuelve el identificador del evento creado.

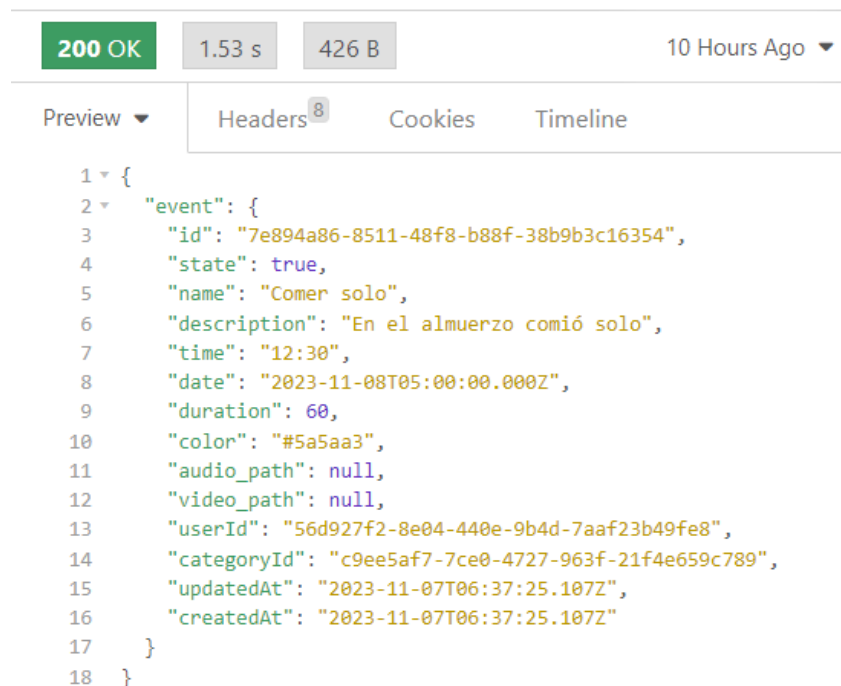


Figura 21. Respuesta recibida a la prueba de integración Crear evento.

3.2.4 Pruebas de validación

Las pruebas de validación se inician al culminar las pruebas de integración. En este punto, los componentes individuales han sido probados exhaustivamente, y el software se ensambla por completo como un paquete. Los errores de interfaz han sido detectados y corregidos.

En la etapa de validación a nivel del sistema, desaparece la distinción entre diferentes categorías de software. Las pruebas se centran en las acciones visibles por el usuario y en la salida del sistema reconocible por el usuario. Es decir, se evalúa si el software cumple con las expectativas y necesidades del usuario final. En resumen, la prueba de validación es crucial para garantizar que el software funcione correctamente en su conjunto y satisfaga las necesidades de los usuarios.

Partición de equivalencia

La partición de equivalencia es un método de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos a partir de los cuales se pueden derivar los casos de prueba. “La partición de equivalencia divide los datos en particiones (también conocidas como clases de equivalencia) de tal manera que se espera que todos los miembros de una partición dada sean procesados de la misma manera. Existen particiones de equivalencia tanto para valores válidos como no válidos.” (Montes Felix 2022)

En la técnica de particiones de equivalencia se realizó el “Diseño de caso de prueba” para ejecutar el flujo básico de las funcionalidades utilizando datos válidos e inválidos (ver tabla 6).

Tabla 6: Diseño de caso de prueba para validación Crear Evento

Escenario	Descripción	Nombre	Descripción	Categoría	Tiempo inicial	Tiempo final	Color	Audio	Imagen	Video	Respuesta del sistema	Flujo central
EC 1.1 Insertar datos correctos	Mediante este escenario se inserta en el sistema un evento.	V	V	V	V	V	V	V	V	V	El sistema crea el nuevo evento y muestra el mensaje: "El evento ha sido creado satisfactoriamente."	El usuario una vez autenticado en el Sistema selecciona el día en que desee crear un evento -El sistema muestra los campos para la creación del nuevo evento. -El usuario llena los campos obligatorios -El usuario presiona el botón: "Crear"
		Asearse solo	En la mañana fue solo al baño y se lavó los dientes solo	Aseo	22:01	22:10	#4de5e8	Sin audio	Sin imagen	Sin video		
		V	V	V	V	V	V	V	V	V		
		Asearse solo	En la mañana fue	Aseo	22:01	22:10	#4de5e8	Audio	Imagen	Video		

			solo al baño y se lavó los dientes solo									
EC 1.2	Mediant e este escenari o se introduc en datos para insertar un evento que ya existe en el sistema.	V	V	V	V	V	V	NA	NA	NA	El sistema divide el día del calendario en dos secciones diferentes, uno a la izquierda y otro a la derecha.	El usuario una vez autenticado en el Sistema se muestra la vista de calendario en la sección Día. -El usuario selecciona la sección en la cual desea crear el evento. -El usuario selecciona el mismo día de creado el evento anterior. -El sistema muestra los campos para la creación del nuevo evento.
Insertar element os repetido s.		Asears e solo	En la mañan a fue solo al baño y se lavó los dientes solo	Aseo	22:01	22:10	#4de 5e8					

												-El usuario rellena los campos con los mismos datos. -El usuario presiona el botón: "Crear". - El sistema divide el día del calendario en dos secciones diferentes, uno a la izquierda y otro a la derecha
EC 1.3 Insertar datos incompletos.	Mediante este escenario o no se introducen todos los datos para crear un evento.	I	NA	NA							El sistema muestra debajo del componente un mensaje en color rojo indicando: "Este campo es obligatorio."	El usuario una vez autenticado en el Sistema selecciona el día en que desee crear un evento -El sistema muestra los campos para la creación del nuevo evento. -El usuario llena los campos obligatorios y los no obligatorios
		Vacío										

												-El usuario presiona el botón: "Crear" -El sistema no le permite crear el evento.
EC 1.4 Insertar datos incorrectos	Mediante este escenario se introducen datos incorrectos.	NA	NA	NA	I	NA					El sistema muestra debajo del componente un mensaje en color rojo indicando: "Introduzca una hora válida."	El usuario una vez autenticado en el Sistema selecciona el día en que desee crear un evento -El sistema muestra los campos para la creación del nuevo evento. -El usuario llena los campos obligatorios y los no obligatorios -El usuario presiona el botón: "Crear" -El sistema no le permite crear el evento.
					30:00							
		NA	NA	NA	NA	I						
						30:00						
EC 1.5 Cancelar	Se cancela la	NA	NA	NA							El sistema al presionar el botón "Cancelar" se cierra	El usuario una vez autenticado en el Sistema

operació n.	creación del evento.										el formulario y se muestra la vista del Calendario	selecciona el día en que desea crear un evento -El sistema muestra los campos para la creación del nuevo evento. -El usuario llena los campos obligatorios y los no obligatorios -El usuario presiona el botón: "Cancelar" -El sistema cierra el formulario
----------------	----------------------------	--	--	--	--	--	--	--	--	--	--	--

Para obtener la calidad del funcionamiento de los requisitos funcionales se llevaron a cabo tres iteraciones, las cuales arrojaron los resultados (ver tabla 7). Las no conformidades encontradas fueron resueltas en su totalidad.

Tabla 7: Resultado de las pruebas de validación.

Iteración	Cantidad de no conformidades	Asociadas a
Primera	5	Errores de validación de campos
Segunda	2	Errores de validación de campos
Tercera	0	-

3.2.5 Pruebas de rendimiento

La evaluación de carga y estrés se refiere, por lo general, a la práctica de analizar el comportamiento de una aplicación bajo cargas o entradas intensas. Estas pruebas se llevan a cabo con el objetivo de verificar si el sistema cumple con los requisitos de rendimiento en situaciones críticas, como: número máximo de usuarios accediendo simultáneamente a los servicios proporcionados, manejo de documentos extremadamente grandes, capacidad para procesar transacciones concurrentes por minuto, tiempo de respuesta ante las solicitudes, permiten asegurar que el sistema funciona de manera óptima incluso en condiciones desafiantes. (Baró & Carla, 2019).

Para la realización de esta prueba (ver tabla 8) se utilizó la herramienta Apache JMeter. Las pruebas se realizaron desde la computadora con 8GB de RAM, microprocesador INTEL(R) CORE(TM) i5-12500H con 2.50 GHz y sistema operativo Windows 11. Se describen las variables que miden el resultado de las pruebas de carga y estrés realizadas al módulo:

Muestra: Representa la cantidad de datos o transacciones procesadas durante la prueba. Una muestra grande proporciona resultados más confiables.

Media (promedio): Calcula la suma de los valores medidos dividida por la cantidad de muestras. Indica el valor central de los datos.

Mediana: Es el valor medio de una serie de datos ordenados. Es útil para identificar valores atípicos.

Mínimo: Representa el valor más bajo observado en la muestra. Indica el rendimiento mínimo.

Máximo: Es el valor más alto observado en la muestra. Indica el rendimiento máximo.

Línea 90 %: Se refiere al percentil 90 de los datos. Representa el valor por debajo del cual se encuentra el 90% de las muestras.

% Error: Calcula la diferencia entre los valores medidos y los valores esperados. Un alto porcentaje de error puede indicar problemas.

Rendimiento: Mide la cantidad de operaciones o transacciones realizadas por unidad de tiempo (por ejemplo, transacciones por segundo).

kb/s (kilobytes por segundo): Indica la velocidad de transferencia de datos. Es relevante para evaluar la eficiencia en la comunicación.

Como se muestra en la tabla (ver tabla 8), se simularon las peticiones realizadas al módulo por un total de 100, 500 y 1000 usuarios simultáneamente en cada caso, los cuales realizan hasta 5 peticiones por segundo.

Tabla 8. Pruebas de Cargas y Estrés.

Usuarios	Muestras	Media	Min	Max	%Error	Rendimiento	Kb/s
100	100	21692	15037	23726	0.00%	4,1/sec	2,75
500	500	60031	60001	60068	0.00%	8,2/sec	2,59
1000	1000	51194	33993	69346	0.00%	14,2/sec	9,91

3.2.6 Evaluación del objetivo de la investigación

Para la valoración de la aplicación desarrollada en entornos reales, fueron encuestados seis padres de niños autistas, tres educadores y cuatro médicos que participaron en el cuasi experimento (ver tabla 9). Las variables evaluadas fueron las siguientes:

Satisfacción del cliente: el cliente considera que la aplicación satisface las necesidades de registro de información digital para apoyar la toma de decisiones.

Aplicabilidad del sistema: el cliente considera que la concepción de la aplicación y sus componentes, permiten su aplicabilidad en diversos entornos.

Utilidad del sistema: es significativo el aporte de información que proporciona el sistema a las organizaciones como punto de partida para apoyar la toma de decisiones.

Tabla 9. Cuadro lógico de ladov.

3- ¿Cuál es su criterio sobre el sistema para la gestión de eventualidades?	1- ¿Considera usted apropiado no llevar un registro digital de la gestión de eventualidades de los niños autistas en Cuba?								
	Sí			No sé			No		
	2- ¿Utilizaría usted el sistema de gestión de eventualidades de los niños autistas en Cuba?								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6
Me da la mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	6
No sé decir	2	3	6	3	3	3	6	3	4

Esta técnica también permite obtener el índice de satisfacción grupal (ISG), para lo cual se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica (ver tabla 10) que oscila entre +1 y - 1 de la siguiente forma: Máxima satisfacción (+ 1), Satisfecho (0.5), No definido (0), Insatisfecho (- 0.5) y Máxima insatisfacción (- 1). A partir de la cantidad de respuestas es posible calcular el Índice de Satisfacción Grupal (ISG) siguiendo la siguiente fórmula:

En esta fórmula A, B, C, D y E representan la cantidad de sujetos que interactúan con la aplicación y N representa el total de sujetos. La escala de valores del índice grupal que se toma al aplicar la técnica se encuentra entre:

- -1 y -0.5 Insatisfacción.

- -0.49 y 0.49 Contradicción.
- 0.5 y 1 Satisfacción

Los resultados obtenidos fueron los siguientes:

Tabla 10. Resultados de las escalas de satisfacción.

Total de usuarios(N)	13	Escala
Máxima satisfacción	7	A
Satisfecho	5	B
No definido	0	C
Insatisfecho	1	D
Máxima insatisfacción	0	E

$$\text{ISG} = A (+1) + B (+0,5) + C (0) + D (-0,5) + E (-1)/N$$

$$\text{ISG} = 7 (+1) + 5 (+0,5) + 0(0) + 1 (-0,5) + 0 (-1)/13$$

$$\text{ISG} = 9/13$$

$$\text{ISG} = 0.69$$

A partir de los datos de satisfacción individual, se determinó el ISG, obteniéndose un valor de 0.69, lo cual se traduce en una clara satisfacción con el sistema propuesto para apoyar al registro de eventualidades de los niños autistas.

Los resultados obtenidos y los criterios emitidos validan la fortaleza de la propuesta, reflejándose una opinión positiva respecto a la satisfacción con el sistema, su utilidad y aplicabilidad.

Conclusiones parciales

La definición de los estándares de codificación a tener en cuenta para la implementación del sistema, permitió garantizar que el código posea una alta calidad y legibilidad, tenga menos errores y pueda ser mantenido y escalado fácilmente, así como reutilizado por otros desarrolladores. En tanto que la realización de las pruebas de software facilitó encontrar errores existentes para posteriormente solucionarlos, posibilitando determinar y asegurar la fortaleza de la aplicación. Por último, la aplicación de la técnica ladov, mostró un alto grado de satisfacción grupal con el

sistema propuesto valorándose positivamente su utilidad y aplicabilidad en entornos reales.

CONCLUSIONES

Una vez realizadas las tareas trazadas en la presente investigación se concluye lo siguiente:

- El estudio de diferentes sistemas para la gestión de eventualidad de los niños autistas permitió obtener una idea temprana de los principales conceptos asociados a la investigación.
- El levantamiento de requisitos aplicando Historias de Usuario permitió realizar una correcta descripción y modelación del problema, obteniéndose 29 requisitos funcionales y cuatro requisitos no funcionales.
- La arquitectura propuesta evidenció que la propuesta solución cuenta con un alto grado de flexibilidad y escalabilidad ante posibles modificaciones.
- El empleo de iteraciones de Scrum promovió un flujo constante de retroalimentación con el cliente y poder corregir oportunamente las no conformidades encontradas.
- Las pruebas de software aplicadas al sistema, garantizaron el cumplimiento de las necesidades y requisitos establecidos por el cliente.
- La validación del objetivo de la investigación con criterio de expertos reflejó que el sistema contribuye a mejorar la gestión de eventualidades de niños autistas en Cuba.

RECOMENDACIONES

Con el cumplimiento del marcado objetivo en la presente investigación y luego de alcanzado el resultado esperado, se recomienda:

- Implementar las funcionalidades necesarias para permitir crear notificaciones para eventos programados con antelación.
- Agregar las funcionalidades necesarias para compartir información personal con determinados usuarios.
- Brindar la posibilidad de visualizar gráficamente información estadística de eventos por categorías y etiquetas permitiendo analizar de manera óptima los avances y/o retrocesos del niño.

BIBLIOGRAFÍA

ANDRADE, Idanis Perdomo, 2022. Revisión sobre el uso de las TIC'S en la Ciencia. *Revista Latinoamericana de Educación Científica, Crítica y Emancipadora*. Vol. 1, número 2, pp. 01-18.

ARMADA GONZÁLEZ, Guillermo, 2019. *Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas* [en línea]. bachelorThesis . Universidad de las Ciencias Informáticas. Facultad 1. Recuperado a partir de : <https://repositorio.uci.cu/jspui/handle/123456789/10071> [accedido 17 septiembre 2023]. Accepted: 2022-04-05T12:39:39Z

Atención, convivencia y apoyo, claves para la atención del Autismo - Cubadebate, 2021 *Cubadebate - Cubadebate, Por la Verdad y las Ideas* [en línea]. Recuperado a partir de : <http://www.cubadebate.cu/noticias/2021/04/02/atencion-convivencia-y-apoyo-claves-para-la-atencion-del-autismo/> [accedido 16 septiembre 2023].

ÁVILA, Eder, 2021. Influencia de los datos enlazados en la generación y gestión del conocimiento. [en línea]. 2021. Recuperado a partir de : https://www.scielo.sa.cr/scielo.php?script=sci_arttext&pid=S1659-41422021000100180 [accedido 5 agosto 2023].

BUILES, Jovani Alberto Jiménez, BEDOYA, Diego León Ramírez y BEDOYA, John Willian Branch, 2019. Metodología de desarrollo de software para plataformas educativas robóticas usando ROS-XP. *Revista Politécnica*. Vol. 15, número 30, pp. 55-69. DOI 10.33571/rpolitec.v15n30a6.

CEVALLOS MOYA, Brayan Enrique y PILATAXI GÓMEZ, Jonathan Vladimir, 2022. *Desarrollo de una aplicación web y móvil para potenciar el aprendizaje lógico matemático en niños de 2 a 3 años del centro infantil Luceritos del sur de la ciudad de Quito* [en línea]. bachelorThesis . Ecuador: Universidad Técnica de Cotopaxi (UTC). Recuperado a partir de : <http://repositorio.utc.edu.ec/handle/27000/9161> [accedido 17 septiembre 2023]. Accepted: 2022-12-05T21:33:53Z

DAVEY, Graham C. L., 2019. Clinical Psychology: Revisiting the Classic Studies. . pp. 1-256.

DEL SOLE, Alessandro, 2021. Introducing Visual Studio Code. En : DEL SOLE, Alessandro (ed.), *Visual Studio Code Distilled: Evolved Code Editing for Windows*,

macOS, and Linux, pp. 1-15. Berkeley, CA: Apress. ISBN 978-1-4842-6901-5. DOI 10.1007/978-1-4842-6901-5_1.

Día a Día, diario visual para personas con autismo o dificultades de comunicación, sin fecha *Fundación Orange* [en línea]. Recuperado a partir de : <https://fundacionorange.es/aplicaciones/dia-a-dia/> [accedido 17 septiembre 2023].

DINA, Jenny et al., 2023. Influencia de la educación en la sociedad del conocimiento en la educación superior universitaria | Delectus. [en línea]. 31 julio 2023. Recuperado a partir de : <https://revista.inicc-peru.edu.pe/index.php/delectus/article/view/212> [accedido 5 agosto 2023].

Documentación de Vue.js, sin fecha [en línea]. Recuperado a partir de : <https://es.vuejs.org/v2/guide/instance.html> [accedido 20 septiembre 2023].

FERNÁNDEZ, Eduardo et al., 2020. Aplicación web para la gestión de diplomas digitales en centros de capacitación mediante firma electrónica y blockchain - ProQuest. [en línea]. abril 2020. Recuperado a partir de : <https://www.proquest.com/openview/e3b85a7260c758fd48b22ce5c52f0ced/1?pq-origsite=gscholar&cbl=1006393> [accedido 5 agosto 2023].

FLOREZ MENA, 2019. Ventajas de la implementación del modelo pedagógico cognitivo de plataforma Moodle para el proceso de enseñanza aprendizaje | Revista Ideales. [en línea]. 2019. Recuperado a partir de : <https://revistas.ut.edu.co/index.php/Ideales/article/view/1963> [accedido 5 agosto 2023].

FLOREZ PORTELA, Laura Alejandra, 2023. En la trama del tejido. *instname:Universidad de Bogotá Jorge Tadeo Lozano*. DOI 10/31103. Accepted: 2023-04-26T15:01:44Z

GARCÍA DÍAZ y María del Mar, 2022. Reflexión sobre las TIC: implicaciones en el desarrollo de los menores. [en línea]. 23 mayo 2022. Recuperado a partir de : <https://riucv.ucv.es/handle/20.500.12466/2796> [accedido 5 agosto 2023].

GARCÍA-GONZÁLEZ, José R. et al., 2020. Diseño teórico de la investigación: instrucciones metodológicas para el desarrollo de propuestas y proyectos de investigación científica. *Información tecnológica*. Vol. 31, número 6, pp. 159-170. DOI 10.4067/S0718-07642020000600159.

GARCÍA-HOLGADO, A. et al., 2020. UML. Unified Modeling Language. . DOI 10.5281/zenodo.3688621. Accepted: 2020-02-26T16:29:03Z

GAUCHAT, Juan Diego, 2012. *El gran libro de HTML5, CSS3 y Javascript*. Marcombo. ISBN 978-84-267-1782-5. Google-Books-ID: szDMIRzwzuUC

GAVILANEZ VAQUE, Gina Thalia y OCHOA MOROCHO, Darwin Joel, 2022. *Aplicación web basado en React.js y Node.js para la automatización del proceso de gestión y control de pasantes de la empresa Viamatica S.A*” [en línea]. bachelorThesis . Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas. Carrera de Ingeniería en Sistemas Computacionales. Recuperado a partir de : <http://repositorio.ug.edu.ec/handle/redug/65219> [accedido 20 septiembre 2023]. Accepted: 2022-12-06T20:55:20Z

GUERRERO, Carlos A., SUÁREZ, Johanna M. y GUTIÉRREZ, Luz E., 2013. Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. *Información tecnológica*. Vol. 24, número 3, pp. 103-114. DOI 10.4067/S0718-07642013000300012.

GUZMÁN, Guido et al., 2017. Nuevas tecnologías: Puentes de comunicación en el trastorno del espectro autista (TEA). *Terapia psicológica*. Vol. 35, número 3, pp. 247-258. DOI 10.4067/S0718-480820170003000247.

HERNÁNDEZ, Jairo Enrique Rodríguez, 2020. DE LOS SISTEMAS DE GESTIÓN AL MODELO INTEGRADO DE PLANEACIÓN Y GESTIÓN EN EL SECTOR PÚBLICO: UNA REVISIÓN DEL CASO COLOMBIANO. *REAd. Revista Eletrônica de Administração (Porto Alegre)*. Vol. 26, pp. 137-175. DOI 10.1590/1413-2311.281.97181.

HIDALGO MOLINA, Juan Ignacio y GARCÍA CASTRO, Diana Stefanie, 2021. *Análisis comparativo de Jamstack VS NODE.JS en el desarrollo de páginas y aplicaciones web* [en línea]. bachelorThesis . Recuperado a partir de : <http://dspace.ups.edu.ec/handle/123456789/20908> [accedido 17 septiembre 2023]. Accepted: 2021-09-15T20:35:24Z

HILLARY, Alyssa y DALTON, Elisabeth, 2019. Augmentative and Alternative Communication for Speaking Autistic Adults: Overview and Recommendations | Autism in Adulthood. [en línea]. 13 abril 2019. Recuperado a partir de :

<https://www.liebertpub.com/doi/full/10.1089/aut.2018.0007> [accedido 16 septiembre 2023].

Index | Node.js v20.6.1 Documentation, sin fecha [en línea]. Recuperado a partir de : <https://nodejs.org/dist/latest-v20.x/docs/api/> [accedido 17 septiembre 2023].

LIZANO, Palma y ALBERTO, Kléber, 2015. *Sistema de gestión de calidad y seguridad BASC (Business Alliance for Secure Commerce) para la Certificación de un operador logístico de la empresa United Parcel Service*. [en línea]. masterThesis . Universidad Técnica de Ambato. Facultad de Ciencia e Ingeniería en Alimentos. Maestría en Producción más Limpia. Recuperado a partir de : <https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/23390> [accedido 17 septiembre 2023]. Accepted: 2016-06-29T22:52:52Z

LOSAVIO, Francisca y ESTEVES, Yuly, 2015. Modelo del Negocio para Análisis del Dominio del Software Educativo: un enfoque centrado en la calidad del producto. *SAPIENS*. Vol. 16, número 1, pp. 49-76.

MINA ÁLVAREZ, Claudia Marcela y LASSO MOSQUERA, Sandra Paola, 2019. Evaluación del sistema de control interno a partir del Modelo Estándar de Control Interno MECI, en la Institución Educativa Sagrado Corazón de Puerto Tejada, Cauca. [en línea]. Recuperado a partir de : <https://bibliotecadigital.univalle.edu.co/entities/publication/ed8fe37c-0d2a-4060-98be-8b9d165dc1cd> [accedido 17 septiembre 2023].

MAR CORNELIO, Omar et al., 2020. Sistema de Laboratorios Remoto para el estudio de la Microbiología y Parasitología Médica. *Revista Cubana de Informática Médica* [en línea]. Vol. 12, número 2. Recuperado a partir de : http://scielo.sld.cu/scielo.php?script=sci_abstract&pid=S1684-18592020000200005&lng=es&nrm=iso&tlng=es [accedido 25 septiembre 2023].

NARVÁEZ CABALLERO, Nikolai Alexander et al., 2020. Propuesta metodológica para el desarrollo del pensamiento creativo desde el diseño de la ingeniería concurrente. *Sophia*. Vol. 16, número 2, pp. 207-218. DOI 10.18634/sophiaj.16v.2i.976.

OLSSON, Mikael, 2019. *CSS3 Quick Syntax Reference: A Pocket Guide to the Cascading Style Sheets Language*. Berkeley, CA : Apress. ISBN 978-1-4842-4902-4.

OSVALDO GEORGE, Rodríguez Pérez y ANNARELIS, Pérez Pupo, 2023. Atención a la pareja infértil: diseño de un sistema de Gestión Hospitalaria. Centro Territorial

Holguín, 2021. En : *Jorcienciapdcl 2023* [en línea]. 3 marzo 2023. Recuperado a partir de : <https://jorcienciapdcl.sld.cu/index.php/jorcienciapdcl23/2023/paper/view/285> [accedido 17 septiembre 2023].

PARDO, César et al., 2020. Modelo de referencia para la adopción e implementación de Scrum en la industria de software. *Investigación e Innovación en Ingenierías*. Vol. 8, número 3, pp. 14-28.

PÉREZ ZÚÑIGA, Ricardo et al., 2018. La sociedad del conocimiento y la sociedad de la información como la piedra angular en la innovación tecnológica educativa. *RIDE. Revista Iberoamericana para la Investigación y el Desarrollo Educativo*. Vol. 8, número 16, pp. 847-870. DOI 10.23913/ride.v8i16.371.

PSYCHIATRY, Indian, 2013. Diagnostic and statistical manual of mental disorders 5: A quick glance - PMC. [en línea]. 2013. Recuperado a partir de : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3777342/> [accedido 16 septiembre 2023].

QUEVEDO, Juan Ignacio, 2022. *Educación digital e inclusión: herramientas digitales utilizadas en educación y su aplicación en alumnos con autismo* [en línea]. bachelorThesis . Recuperado a partir de : <https://repositorio.21.edu.ar/handle/ues21/26309> [accedido 5 agosto 2023]. Accepted: 2023-02-06T12:29:34Z

RODRÍGUEZ HERNÁNDEZ, Dayanis María, MARTÍNEZ VENCE, Leyanis y FUMERO MORENO, Danna, 2023. El registro del derecho de autor en la legislación cubana. [en línea]. Recuperado a partir de : <https://riu.austral.edu.ar/handle/123456789/2263> [accedido 17 septiembre 2023]. Accepted: 2023-08-03T20:01:08Z

ROSALES VERA, Helen Noemi, 2023. *Aplicación de la NIC 7 en la empresa Junta Administradora de Agua Potable y Alcantarillado Ayangue, cantón Santa Elena, provincia de Santa Elena, año 2022* [en línea]. bachelorThesis . La Libertad: Universidad Estatal Península de Santa Elena. 2023. Recuperado a partir de : <https://repositorio.upse.edu.ec/handle/46000/10078> [accedido 17 septiembre 2023]. Accepted: 2023-08-29T16:55:33Z

SANROMA-GIMENEZ, MÓNICA, LAZARO-CANTABRANA, José Luis y GISBERT-CERVERA, MERCÉ, 2017. Mobile technology: A tool for improving the digital inclusion for people with ASD. [en línea]. 2017. Recuperado a partir de :

http://www.scielo.edu.uy/scielo.php?pid=S1688-70262017000200173&script=sci_abstract&lng=en [accedido 21 agosto 2023].

SANTIESTEBAN CATALLOPS, Leonardo José, 2019. *Sistema de gestión de información tecnológica para la Facultad 1 de la Universidad de las Ciencias Informáticas* [en línea]. bachelorThesis . Universidad de las Ciencias Informáticas. Facultad 1. Recuperado a partir de : <https://repositorio.uci.cu/jspui/handle/123456789/10067> [accedido 17 septiembre 2023]. Accepted: 2022-04-04T17:30:39Z

SAZO, Jose Antonio, 2022. Una revisión narrativa sobre las estrategias para la intervención de la comunicación en niños y niñas con alto riesgo de trastorno del espectro del autismo | UCMaule. [en línea]. 19 diciembre 2022. Recuperado a partir de : <https://revistaucmaule.ucm.cl/article/view/928> [accedido 16 septiembre 2023].

SCHWABER, Ken, 1997. SCRUM Development Process. En : SUTHERLAND, Jeff et al. (eds.), *Business Object Design and Implementation*, pp. 117-134. London : Springer. 1997. ISBN 978-1-4471-0947-1. DOI 10.1007/978-1-4471-0947-1_11.

SEVILLA ROBLES, Miguel Ángel, 2020. Resumen sobre Internet. [en línea]. Recuperado a partir de : <http://148.202.167.116:8080/xmlui/handle/123456789/3088> [accedido 5 agosto 2023]. Accepted: 2020-03-26T18:08:00Z

STEPHANIDIS, Constantine y ANTONA, Margherita, 2013. *Universal Access in Human-Computer Interaction: User and Context Diversity: 7th International Conference, UAHCI 2013, Held as Part of HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part II*. Springer. ISBN 978-3-642-39191-0. Google-Books-ID: 0iW5BQAAQBAJ

TELENCHANA CHIMBO, Daniel Iván, 2022. *Aplicación web usando el framework Angular para el control de historias clínicas de los pacientes del consultorio médico Fisio&Trauma de la ciudad de Ambato* [en línea]. bachelorThesis . Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Ingeniería en Sistemas Computacionales e Informáticos. Recuperado a partir de : <https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/34812> [accedido 5 agosto 2023]. Accepted: 2022-03-30T15:56:40Z

UCHIHA, Jesus, 2015. Ingeniería de software Ian Sommerville 7ma edición - [PDF Document]. *vdocumento.com* [en línea]. 8 agosto 2015. Recuperado a partir de :

<https://vdocumento.com/ingenieria-de-software-ian-sommerville-7ma-edicion.html>
[accedido 20 septiembre 2023].

MAR CORNELIO, Omar et al., 2020. Sistema de Laboratorios Remoto para el estudio de la Microbiología y Parasitología Médica. *Revista Cubana de Informática Médica* [en línea]. Vol. 12, número 2. Recuperado a partir de : http://scielo.sld.cu/scielo.php?script=sci_abstract&pid=S1684-18592020000200005&lng=es&nrm=iso&tlng=es.

GÓMEZ, Ángel Pisco et al., 2017. *FUNDAMENTOS SOBRE LA GESTIÓN DE BASE DE DATOS*. 3Ciencias. ISBN 978-84-947995-6-3. Google-Books-ID: H0VBDwAAQBAJ

HIDALGO PEÑA, Massiel, 2019. *Portal web para el Movimiento Juvenil Martiano* [en línea]. bachelorThesis . Universidad de las Ciencias Informáticas. Facultad 1. Recuperado a partir de : <https://repositorio.uci.cu/jspui/handle/123456789/10089> [accedido 20 octubre 2023]. Accepted: 2022-04-06T13:24:02Z

PATILLA, Hubner Janampa et al., 2021. Modelo de Gestión de Desarrollo de Software Ágil mediante Scrum y Kanban sobre la Programación Extrema - ProQuest. [en línea]. julio 2021. Recuperado a partir de : <https://www.proquest.com/openview/e9496bfb82c7643bfe4661484604a07b/1?pq-origsite=gscholar&cbl=1006393> [accedido 20 octubre 2023].

PRESSMAN, Roger y LOWE, David, 2008. *Web Engineering: A practioner's approach*.

SOMMERVILLE, Ian, 2005. *Ingeniería del software*. Pearson Educación. ISBN 978-84-7829-074-1. Google-Books-ID: gQWd49zSut4C

GÓMEZ RODRÍGUEZ, Nuria, 2015. *Las pruebas de integración como proceso de la calidad del software en el ámbito de las telecomunicaciones* [en línea]. bachelorThesis . Recuperado a partir de : <https://e-archivo.uc3m.es/handle/10016/25775> [accedido 6 noviembre 2023]. Accepted: 2017-11-08T17:35:09Z

MONTES FELIX, Frank David, 2022. Ejecución de pruebas funcionales para certificar el sistema de agendamiento y realización de teleconsultas médicas “SPARTM”. *Universidad Nacional Mayor de San Marcos* [en línea]. Recuperado a partir de : <https://cybertesis.unmsm.edu.pe/handle/20.500.12672/17923> [accedido 7 noviembre 2023]. Accepted: 2022-04-06T23:36:15Z

The Collaborative API Development Platform - Insomnia, 2023 [en línea]. Recuperado a partir de : <https://insomnia.rest/>

GUTIERREZ, Vargas y PABLO, Juan, 2022. APOYO EN EL DESARROLLO DE REQUERIMIENTOS Y CORRECCIÓN SOBRE INCIDENCIAS REPORTADAS EN LA EMPRESA INDRA COMPANY. [en línea]. Recuperado a partir de : <http://repositorio.ucp.edu.co/handle/10785/12536> [accedido 10 noviembre 2023]. Accepted: 2023-03-18T16:14:02Z