

Herramienta de gestión de colas y tareas de procesamiento de neurodatos para la plataforma BrainSSys

Neurodata Processing Task and Queue Management Tool for the BrainSSys Platform

Arturo Orellana García ^{1*}	0000-0002-3652-969X
Denys Buedo Hidalgo ²	0000-0002-5031-2040
Dairon Canel Gómez ³	0000-0003-2700-6042
Jesús Enrique Fuentes González	0000-0001-7798-2768

^{1,3,4} Centro de Informática Médica, Universidad de las Ciencias Informáticas, La Habana, Cuba.

² Ministerio de Educación Superior, La Habana Cuba.

* Autor para la correspondencia: aorellana@uci.cu

RESUMEN

Las neurociencias constituyen un campo de la ciencia que estudia el sistema nervioso desde el funcionamiento neuronal hasta el comportamiento y cómo sus diferentes elementos interactúan. El Centro de Neurociencias de Cuba es la principal institución de investigación de neurociencia en el país, donde fueron identificadas distintas problemáticas asociadas al lanzamiento y gestión de colas de procesamiento de datasets de neurociencias. El procesamiento que se desarrolla sobre sus bases de datos se realiza de forma manual en sus servidores de HPC (High Performance Computer) debido a dificultades tecnológicas y financieras para acceder a los servicios que ofrecen las plataformas internacionales, por lo que los investigadores se ven obligados a programar instrucciones en consola para ejecutar tareas de procesamiento y análisis.

El objetivo de la presente investigación fue desarrollar una herramienta que automatice el proceso de gestión de colas y tareas de procesamiento de neurodatos para la plataforma BrainSSys. Se empleó Python como lenguaje de programación, PyQt5 como marco de trabajo y se utilizó el editor de código Sublime Text 3.

Se obtuvo como resultado una herramienta que permite automatizar la gestión de las colas de procesamiento en la plataforma BrainSSys. Las pruebas realizadas determinaron la aceptación y la evaluación funcional de la herramienta, siendo en ambos casos de gran valor para la calidad de la propuesta solución.

Palabras Clave: automatización; gestión de colas; neurociencias; servidores HPC.



ABSTRACT

Neuroscience is a field of science that studies the nervous system from neural functioning to behavior and how its different elements interact. The Cuban Neuroscience Centre is the main neuroscience research institution in the country, where different problems associated with the launch and management of neuroscience dataset processing queues has been identified. The processing of its databases is carried out manually on their HPC (High Performance Computer) servers due to technological and financial difficulties in accessing the services offered by international platforms, so researchers are forced to program instructions on the console to execute processing and analysis tasks.

The aim of this research was to develop a tool that automates the queue management process and neurodata processing tasks for the BrainSSys platform. Python was used as programming language, PyQt5 and Sublime Text 3 as framework and code editor respectively.

The result is a tool that automates the management of processing queues on the BrainSSys platform. The tests carried out determined the acceptance and functional evaluation of the tool, being in both cases of great value for the quality of the proposed solution.

Keywords: automation; queue management; neurosciences; processing; HPC servers.

Recibido: 05/12/2021

Aprobado: 17/03/2022

Introducción

El análisis biológico del cerebro es un área multidisciplinar que abarca varios niveles de estudio: desde el puramente molecular hasta el específicamente conductual y cognitivo. Comprende el nivel celular (neuronas individuales), los ensambles y redes pequeñas de neuronas (como las columnas corticales) y los ensambles grandes (como los propios de la percepción visual) incluyendo sistemas como la corteza cerebral o el cerebelo, e incluso, el nivel más alto del sistema nervioso. ⁽¹⁾

El propósito principal de las Neurociencias es entender cómo el encéfalo produce la marcada individualidad de la acción humana, es aportar explicaciones de la conducta en términos de actividades del encéfalo, explicar cómo actúan millones de células nerviosas individuales en el encéfalo para producir la conducta y cómo, a su vez, estas células están influidas por el medio ambiente, incluyendo la conducta de otros individuos. ⁽²⁾



Un término importante a tener en cuenta en las Neurociencias es la Neuroinformática. Este término es utilizado en relación al campo de investigación de la neurociencia mediante la aplicación de modelos computacionales y herramientas analíticas. Tiene como objetivo facilitar de una manera estructurada y disciplinada que las nuevas tecnologías permitan a las diferentes áreas de investigación compartir datos y hallazgos para poder continuar con el estudio y entendimiento del cerebro.⁽³⁾

Se han desarrollado plataformas para el procesamiento y análisis de datos de neurociencias producto de las alianzas entre diferentes países, patrocinados por proyectos y financiamientos internacionales; entre ellos CBRAIN, un sistema web para el estudio de neurodatos y LORIS (Sistema Longitudinal de Investigación e Imágenes en Línea, por sus siglas en inglés), un software de gestión de proyectos y datos basado en la web para estudios de investigación de neuroimágenes⁽⁴⁾.

En Cuba se desarrollan investigaciones con el uso de la Neuroinformática y las Neurotecnologías en instituciones como el Centro de Neurociencias de Cuba (CNEURO). Dicha institución está dedicada a la investigación de Neurociencias, Neurotecnología y otras tecnologías médicas, que abarcan desde la investigación básica hasta el desarrollo, la producción y comercialización. La entidad lleva a cabo investigaciones de temas como neurociencia cognitiva, neuroinformática y neuroimagen funcional⁽⁵⁾.

Neurociencia y Neuroinformática

Los actuales estudiosos del cerebro, saben que para comprenderlo hay que derrumbar las barreras de las disciplinas tradicionales para señalar algunas de las áreas que han sido creadas, en gran parte para caracterizar los métodos de estudio.⁽⁶⁾

La Neuroinformática es un término utilizado en relación al campo de investigación que combina la investigación en neurociencia e informática para desarrollar herramientas innovadoras destinadas a la organización de datos neurocientíficos de gran volumen y alta dimensión. También se aplican modelos computacionales para integrar y analizar estos datos para eventualmente comprender la estructura y función del cerebro.⁽⁷⁾

El campo de la Neuroinformática ha crecido rápidamente para involucrar el desarrollo de herramientas modernas basadas en la tecnología de la información y la comunicación (TIC) con el objetivo de ayudar a comprender la función cerebral en la salud y la enfermedad. Para lograr este objetivo, es esencial crear grandes bases de datos donde sea posible compartir datos de neurociencia primaria, cuidadosamente estandarizados. Al igual que desarrollar herramientas de análisis poderosas y modelos computacionales.⁽⁷⁾

En las neurociencias se distinguen varios niveles de análisis, que en orden ascendente de complejidad comprenden:

- La neurociencia molecular, que estudia la complejidad molecular del sistema nervioso y las diversas moléculas que lo componen.
- La neurociencia celular, que presta atención al estudio de cómo todas estas moléculas trabajan juntas, proporcionando a las neuronas propiedades especiales como los tipos diferentes de neuronas y sus funciones.



- La neurociencia de sistemas, que estudia los circuitos y redes neuronales complejos que llevan a cabo una función común, por ejemplo, visión, movimiento y lenguaje.
- La neurociencia cognitiva, que investiga los procesos neuronales que están en la base de los niveles superiores de la actividad mental humana, como son la conciencia de sí mismo, el lenguaje, la imaginación, la creatividad, el sentido estético y el comportamiento moral.

El desarrollo de vanguardia en el campo de la Neuroinformática ha generado avances significativos en estas últimas décadas, entre ellos se incluyen:

- Recopilación y mejor organización de datos de neurociencia (tanto en laboratorios húmedos como simulados) y desarrollo de bases de conocimiento.
- Desarrollo de flujos de trabajo para una mejor gestión y manejo de datos, metadatos y otros documentos relacionados con la investigación.
- Desarrollo de herramientas para la adquisición, análisis y visualización de datos automatizados y multipropósito.
- Desarrollo de herramientas y creación de repositorios para la distribución de datos.
- Desarrollo de herramientas para la teoría, cálculo y simulación de fenómenos neuronales.
- Plataformas para la gestión y análisis de datos de neurociencias.

CNEURO es coordinador del Programa Nacional de Ciencia y Tecnología de Neurociencias y Neurotecnologías, por lo que trabaja en estrecha colaboración con distintas entidades para el desarrollo de investigaciones. Una de ellas es la Universidad de Ciencias Informáticas (UCI), que ejecuta un proyecto de investigación nacional denominado BrainSSys para solucionar diferentes problemáticas asociadas a las tecnologías y a la Neuroinformática. El proyecto tiene entre sus objetivos desarrollar una plataforma digital para la estructuración, manejo de bases de datos multimodales de neurociencias y análisis de datos estandarizados del cerebro.

En reuniones efectuadas con expertos de CNEURO se identificaron distintas problemáticas asociadas al lanzamiento y gestión de colas de procesamiento de datasets de neurociencias, entre ellos:

- Los investigadores se ven obligados a programar instrucciones en consola para ejecutar tareas de procesamiento y análisis.
- En ocasiones se debe repetir el proceso debido a errores en las instrucciones lo cual atrasa la planificación de la investigación.
- La creación de colas de procesamiento y la distribución de las tareas en los servidores se ve afectada al ser manual y subjetiva, basada en la experiencia del investigador, lo que afecta a los nuevos trabajadores del centro por no contar con la experticia requerida.



Para dar solución al problema planteado se definió como objetivo desarrollar una herramienta para la plataforma BrainSSys que propicie la gestión de colas y tareas de procesamientos de datos de neurociencias.

Materiales y métodos

Métodos teóricos:

- Método analítico-sintético: utilizado para el análisis de la teoría y extracción de los principales conceptos a incluir en el marco teórico
- Método histórico-lógico: para investigar acerca de otras soluciones similares.
- Método inductivo-deductivo: utilizado en la aplicación de casos de pruebas al sistema, obteniendo conclusiones a partir de las respuestas proporcionadas por este.

Métodos empíricos:

- Entrevista: permitió la recolección de información a través de conversaciones planificadas con especialistas de Neurociencias.
- Observación: se utilizó como método para validar la propuesta de solución.
- Modelación: se empleó como recurso auxiliar en la búsqueda teórica para caracterizar y representar mediante diagramas el campo de acción de la presente investigación.

Se seleccionó la metodología de desarrollo Extreme Programming (XP), debido a las características de la solución a desarrollar. Para el modelado se utilizó UML en su versión 2.0 y la herramienta CASE Visual Paradigm 8.0. Para la persistencia de los datos se utilizó SQLite en su versión 3.8.

Como lenguaje de programación se utilizó Python 3.7 por la legibilidad de su código y por ser un lenguaje interpretado, dinámico y multiplataforma. Como entorno integrado de desarrollo se utilizó el editor de texto y de código fuente Sublime Text 3 por sus facilidades para codificar y la aceptación de la comunidad.

Como marco de trabajo se utilizó Qt y la colección de interfaces de una serie de módulos que unen Python 2.x y 3.x al marco de la aplicación Qt denominada PyQt por tener la documentación más vasta en comparación con otras bibliotecas de GUI de Python.



Resultados y discusión

A continuación, se presenta un análisis del estado del arte de las herramientas que permiten los procesos gestión de colas y tareas de procesamiento de neurodatos. El análisis fue útil para identificar las formas de gestión de los procesamientos de datos de neurociencias. Se analizaron de acuerdo a las siguientes características:

- Acceso al repositorio
- Acceso a los datos
- Tipo de datos
- Tecnología / Lenguajes
- Si es de código abierto o no
- Organización a la que pertenece

La Plataforma Canadiense de Investigación de Imágenes Cerebrales (CBRAIN) es una plataforma de investigación colaborativa basada en la web, ⁽⁸⁾ desarrollada en respuesta a los desafíos planteados por la investigación de neuroimágenes con gran cantidad de datos y con uso intensivo de ordenadores. CBRAIN ofrece un acceso transparente a fuentes de datos remotas, sitios de computación distribuidos y una serie de herramientas de procesamiento y visualización dentro de un entorno controlado y seguro.

CBRAIN usa una conexión con el repositorio SSL y SSH, para acceder a los datos es necesario un inicio de sesión, como tipo de datos utiliza archivos planos, imágenes y metadatos textuales; No especifican las tecnologías utilizadas y no utilizan código abierto. La institución pertenece a la CONP (Canadian Open Neuroscience Platform).

Longitudinal Online Research and Imaging System (LORIS) es un sistema modular y extensible de gestión de datos basado en la web que integra todos los aspectos de un estudio multicéntrico: ⁽⁹⁾ desde la adquisición de datos heterogéneos (imagen, clínica, comportamiento y genética) hasta el almacenamiento, el procesamiento y, finalmente, la difusión. Proporciona una plataforma segura, fácil de usar y racionalizada para automatizar el flujo de los ensayos clínicos y los estudios multicéntricos complejos. Una organización interna centrada en el sujeto permite a los investigadores capturar y posteriormente extraer toda la información, longitudinal o transversal, de cualquier subconjunto del estudio.

Loris utiliza una conexión SSL al repositorio, para acceder a los datos es necesario un inicio de sesión, como tipo de datos utiliza archivos planos, imágenes, metadatos textuales y visualizaciones 3D; Utilizan Linux, Apache, MySQL, PHP (LAMP), no utilizan código abierto y pertenecen a la CONP (Canadian Open Neuroscience Platform).

La Biblioteca de imágenes cerebrales (BIL), ^[10] es un recurso público nacional que permite a los investigadores depositar, analizar, extraer, compartir e interactuar con grandes conjuntos de datos de imágenes cerebrales. BIL abarca la deposición de conjuntos de datos, la integración de conjuntos de datos en un sistema de búsqueda accesible en la web, la redistribución de conjuntos de datos y un enclave computacional para permitir a



los investigadores procesar conjuntos de datos en el lugar y compartir conjuntos de datos restringidos y previos al lanzamiento.

BIL usa una conexión con el repositorio SSL, Bridges-2, Neocórtex y Sistema VM, para acceder a los datos es necesario un inicio de sesión, como tipo de datos utiliza imágenes (tiff, jpg-2000) y datos de neuronas (swc); No especifican las tecnologías utilizadas y no utilizan código abierto. La institución pertenece a la Iniciativa BRAIN, de la Universidad de Pittsburgh

El repositorio C-BIG ^[11] se lanzó internamente en 2019 como el repositorio de datos clínicos y bioespecímenes de Canadian Open Neuroscience Platform (CONP), y públicamente en noviembre de 2020 para la comunidad de investigación en general. Este repositorio se construyó utilizando LORIS para recolectar bioespecímenes y datos clínicos de imágenes y genéticos de pacientes con enfermedades neurológicas y controles sanos.

El repositorio permite el seguimiento y la difusión de datos de muestras biológicas sin procesar o derivados, y admite metadatos y estadísticas resumidas en múltiples niveles de control de acceso, incluido el acceso para investigadores genuinos y profesionales de la atención clínica a través del novedoso modelo de Acceso Registrado. C-BIG presenta las mejores prácticas actuales de intercambio de datos, como FAIR, e incluye captura de procedencia, además de registro completamente auditable y otros esfuerzos de estandarización.

C-Big usa una conexión con el repositorio SSH; para acceder a los datos es necesario un inicio de sesión, como tipo de datos utiliza imágenes EEG (EEG-BIDS) y MEG (MEG-BIDS); Utilizan LORIS, MySQL, React, JavaScript y no utilizan código abierto. La institución pertenece a la CONP (Canadian Open Neuroscience Platform).

NeuroVault, ^[12] es una plataforma web que permite a los investigadores almacenar, compartir, visualizar y decodificar mapas del cerebro humano. Este nuevo recurso puede mejorar la forma en que se presentan, difunden y reutilizan los experimentos de cartografía del cerebro humano.

NeuroVault usa una conexión con el repositorio SSH; para acceder a los datos es necesario un inicio de sesión, como tipo de datos utiliza imágenes EEG y MEG (NIFTI). Utilizan Python, Django, Docker, JavaScript y código abierto. La institución pertenece a la Neuroscience Information Framework

El repositorio facilita el depósito y el intercambio de mapas estadísticos, proporciona una visualización atractiva y una decodificación cognitiva de los mapas que pueden mejorar los esfuerzos de colaboración y la legibilidad de los resultados. Al mismo tiempo, también proporciona una API para que los investigadores de métodos descarguen los datos, realicen análisis potentes o creen nuevas herramientas ^[13].

Las características analizadas son válidas en el entorno para el que fueron desarrolladas, sin embargo, no son útiles en su totalidad para Cuba. Esto se debe a que en su mayoría no son de código abierto, pertenecen o están asociados a instituciones / organizaciones extranjeras por lo que no cumplen con las políticas de soberanía del país. De ahí que no sea posible dar soporte, ni personalizarlas a las características propias de CNEURO. Por eso



se decide el desarrollo de una solución propia para la gestión de colas de tareas de procesamientos, tomando en cuenta las características deseadas por la institución y partiendo de la experiencia internacional.

La herramienta propuesta en esta investigación gestiona las colas y tareas de procesamientos de datos de neurociencias para la plataforma BrainSSys. Se crea, como se ha planteado, con el objetivo de responder a la necesidad de automatización de los procesos señalados y para subsanar las deficiencias inherentes al tratamiento manual.

Los modelos de dominio pueden utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de software o de otro tipo. Similares a los mapas mentales utilizados en el aprendizaje, el modelo de dominio es utilizado por el analista como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir. (Fig. 1)

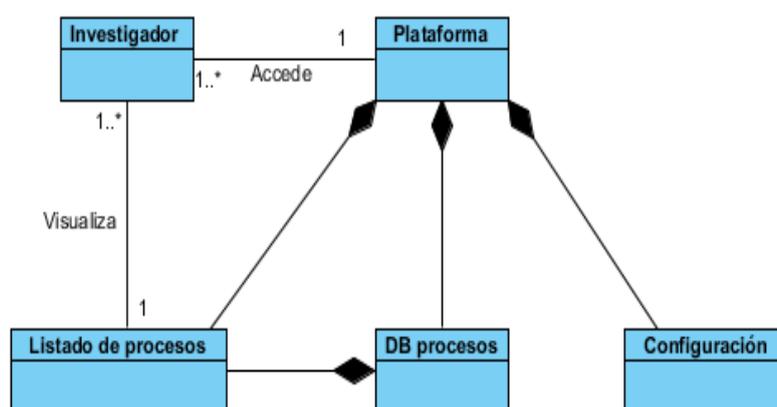


Fig. 1- Modelo conceptual del sistema (Elaboración propia)

La obtención de requisitos es uno de los pasos fundamentales en el desarrollo de software. Esta tarea está encaminada a identificar los requerimientos del cliente, analizar las necesidades y especificar los requisitos de la propuesta de solución. En la presente investigación se identificaron 4 requisitos funcionales y 5 requisitos no funcionales.

Como requisitos funcionales se identificó el RF1: Listar procesos (la aplicación debe permitir mostrar un listado con todos los procesos), el RF2: Eliminar proceso (la aplicación debe permitir eliminar los procesos listados), el RF3: Modificar proceso (la aplicación debe permitir modificar los datos de los procesos añadidos al sistema) y el RF4: Añadir proceso (la aplicación debe permitir adicionar procesos a la lista de procesos).

Los actores del negocio son aquellas personas que interactúan con el negocio para beneficiarse de sus resultados. Ellos se agrupan en diferentes roles los cuáles son descritos a continuación:

- Administrador: posee todos los privilegios de la herramienta.
- Investigador: Crea y lista procesos mediante su interacción con la plataforma.



Las metodologías ágiles como XP generan artefactos ingenieriles como las Historias de Usuario (HU) que sustituyen a los documentos de especificación funcional, y a los casos de uso. Se utilizaron con el fin de especificar los requisitos del software y fueron escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar.⁽¹⁴⁾ A través de una tabla fueron descritas brevemente las características deseadas. La HU también se utilizó para estimar el tiempo que le tomó al equipo de desarrollo para realizar las entregas, se programaron en un tiempo aproximado de 2 semanas.

La arquitectura empleada para el desarrollo del sistema es Modelo-Vista-Controlador (Fig. 2). Esta arquitectura divide la presentación e interacción de los datos del sistema. El sistema se estructura en tres componentes lógicos que interactúan entre sí.⁽¹⁵⁾

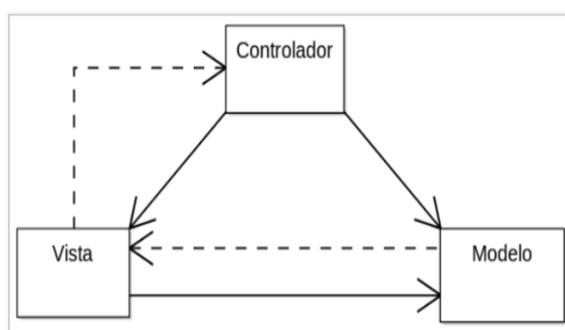


Fig. 2- Arquitectura para el desarrollo del sistema. (Elaboración propia).

Se hizo uso también del modelo CRC (Clase-Responsabilidad-Colaboración) que utiliza tarjetas de índices reales o virtuales con el objetivo de desarrollar una representación organizada de las clases.

Las responsabilidades se representaron como los atributos y operaciones relevantes para la clase. Los colaboradores se representaron como aquellas clases que se requieren para dar a una clase la información necesaria a fin de completar una responsabilidad, es decir que una colaboración implica una solicitud de información o de cierta acción.⁽¹⁶⁾ (Tabla 1)

Tabla 1- Tabla de tarjeta CRC asociada a la clase ProcessTable. (Elaboración propia).

Nombre de la clase: ProcessTable	
Responsabilidades	Clases Relacionadas
editProcess()	QApplication
createProcess()	EditProcess
elimProcess()	CreateProcess
listProcess()	Process
	ProcessListHandler



Los patrones de diseño son soluciones a problemas típicos y recurrentes que se pueden encontrar a la hora de desarrollar una aplicación. Establecen un lenguaje común para programar. Identifican clases, instancias, roles, colaboraciones y distribución de responsabilidades. Proveen un esquema para refinar componentes de un sistema de software y la forma en que se relacionan entre sí. ⁽¹⁷⁾

Estos patrones describen una estructura generalmente recurrente de comunicación de componentes que resuelve un problema de diseño general dentro de un contexto particular, lo cual permite un lenguaje de programación de alto nivel. Los patrones de diseño utilizados GRASP: Creador, Controlador, Experto, Bajo Acoplamiento y Alta Cohesión, en conjunto con los estándares de codificación especificados permitieron estructurar la solución respondiendo a diversos problemas en la implementación de los requisitos funcionales.

El patrón creador ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases (Fig. 3). La nueva instancia deberá ser creada por la clase que tiene la información necesaria para realizar la creación del objeto, o usa directamente las instancias creadas del objeto, o almacena o maneja varias instancias de la clase. Una de las consecuencias de usar este patrón es la visibilidad entre la clase creada y la clase creador.

```
21 class Process():
22     def __init__(self, idprocess, taskType,
23                 desc, owner, execServer, currentStatus,
24                 workdirSize, timeSubm, lastUpdated):
25         self.idprocess = idprocess
26         self.taskType = taskType
27         self.desc = desc
28         self.owner = owner
29         self.execServer = execServer
30         self.currentStatus = currentStatus
31         self.workdirSize = workdirSize
32         self.timeSubm = timeSubm
33         self.lastUpdated = lastUpdated
```

Fig. 3- Ejemplo de patrón creador (Elaboración propia).

El GRASP de experto en información es el principio básico de asignación de responsabilidades. Consiste en la asignación de una responsabilidad a la clase que cuenta con la información necesaria para llevarla a cabo. Determina cuál es la clase que debe asumir una responsabilidad a partir de la información que posee cada una. La creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. (Fig. 4)



```
def getId(self):  
    return self.idprocess  
  
def getTaskType(self):  
    return self.taskType  
  
def getDesc(self):  
    return self.desc
```

Fig. 4- Ejemplo de patrón experto (Elaboración propia)

Un controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado (Fig. 5). Sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto permite aumentar la reutilización de código y a la vez tener un mayor control.

```
def setId(self, setElement):  
    self.idprocess = setElement  
  
def setTaskType(self, setElement):  
    self.taskType = setElement  
  
def setDesc(self, setElement):  
    self.desc = setElement
```

Fig. 5- Ejemplo de patrón controlador (Elaboración propia).

En la perspectiva del diseño orientado a objetos, la cohesión o cohesión funcional es una medida que muestra cuan relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme, clases con responsabilidades moderadas en un área funcional que colaboran con las otras para llevar a cabo las tareas.

El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, las conoce y recurre a ellas. Propone tener las clases lo menos ligadas entre sí, de tal forma que, en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases. El bajo acoplamiento soporta el diseño de clases más independientes y reutilizables, lo cual reduce el impacto de los cambios y acrecienta la oportunidad de una mayor productividad.

Los modelos de datos aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos. El diseño de una base de datos consiste en definir la estructura de los datos que debe tener un sistema de información determinado (Fig. 6). Para ello se suelen seguir por regla general fases en el proceso de diseño, definiendo para ello el



modelo conceptual, el lógico y el físico. Generalmente es usado para describir la representación lógica y física de la información persistente manejada por el sistema.

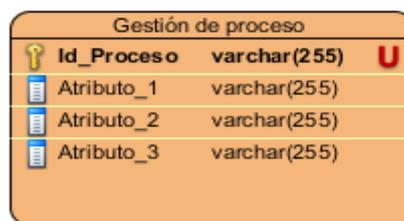


Fig. 6- Modelo Entidad-Relación (Elaboración propia).

La especificación de las tareas de programación permitió obtener un desglose por unidades de las funcionalidades encapsuladas mediante las historias de usuario. Los diseños de los casos de prueba de aceptación y unidad permitieron guiar la ejecución de las actividades de verificación y validación para un correcto funcionamiento de la aplicación. Las no conformidades resultantes por iteraciones permitieron realizar un análisis incremental sobre el funcionamiento de la herramienta, logrando así todos los objetivos propuestos.

Conclusiones

Se constató en la literatura estudiada que el tema es novedoso, actual y pertinente; además contribuyó a la elaboración de los fundamentos teórico-metodológicos de la investigación, así como la identificación de los principales problemas presentes en la actualidad con respecto al ámbito de las neurociencias en Cuba, determinando características aplicables a la solución propuesta.

El ambiente de desarrollo está acorde con el tipo de proyecto y las tendencias internacionales y permitió establecer la infraestructura tecnológica para ejecutar la propuesta.

Las funcionalidades que permiten la gestión de las colas de procesamiento, fueron implementadas a partir de la fase de análisis y diseño.

La calidad del producto final fue validada al aplicar pruebas de software a la solución en todas sus etapas de desarrollo lo cual determinó la aceptación y la evaluación funcional de la herramienta.

Agradecimientos

La investigación que da origen a los resultados presentados en la presente publicación recibió fondos de la Oficina de Gestión de Fondos y Proyectos Internacionales bajo el código PN305LH13-038.



Referencias

1. Manes, F., & Niro, M. (2014): Usar el cerebro. Barcelona: Paidós.
2. Kandell, E. R., Jessell, T. M., & Schwartz, J. H. (1997): Neurociencia y Conducta. Madrid: Prentice Hall.
3. Adey, & Sally. (2008): Reverse engineering the brain. 45(6).
4. MCIN. (2021): McGill Centre for Integrative Neuroscience. Retrieved from MCIN: <https://mcin.ca>
5. CNEURO. (2021): Centro de Neurociencias de Cuba. (CNEURO) Retrieved Mayo 5, 2021, from CNEURO: <https://www.cneuro.cu/>
6. Rosell., R., Juppet, M. F., Ramos, Y., Ramírez, R. I., & Barrientos, N. (2020): Neurociencia aplicada como nueva herramienta para la educación. Opcion, Regular (92), 792-818.
7. Linne, M.-L. (2018): Neuroinformatics and Computational Modelling as Complementary Tools for Neurotoxicology Studies. Basic & Clinical Pharmacology & Toxicology, 123, 56–61. doi:Doi: 10.1111/bcpt.13075
8. Sherif, T., Rioux, P., Rousseau, M.-E., Kassis, N., Beck, N., Reza, A., Evans, A. (2014): CBRAIN: a web-based, distributed computing platform for collaborative neuroimaging research.
9. Das, S., Zijdenbos, A., Harlap, J., Vins, D., & Evans, A. (2012): LORIS: a web-based data management system for multi-center studies. Frontiers in neuroinformatics.
10. BIL. (2021): The Brain Image Library. Retrieved from <https://www.brainimagelibrary.org>
11. C-BIG. (2021): C-BIG Repository. Retrieved on from <https://www.mcgill.ca/neuro/open-science/c-big-repository>
12. NeuroVault. (2021, Enero): Retrieved Mayo 5, 2021, from <https://neurovault.org/>
13. Gorgolewski, K. J., Varoquaux, G., Rivera, G., & Schwarz, Y. (2015): NeuroVault.org: a web-based repository for collecting and sharing unthresholded statistical maps of the human brain. Frontiers, 9(8), 1-9. doi:10.3389/fninf.2015.00008
14. Joskowicz, J. (2008): Reglas y prácticas en eXtreme Programming. vol. 22.
15. Sommerville, I. (2011): Ingeniería de software Novena edición. México.
16. Pressman, R. S. (2010): Ingeniería de software un enfoque práctico, Séptima edición. México, D. F.: Miembro de la Cámara Nacional de la Industria Editorial Mexicana, Reg.
17. Larman, C. (2004): Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. ISBN 0-13-148906-2.

Conflictos de interés

Los autores declaran que no existen conflictos de intereses.

Declaración de autoría

Dr.C. Arturo Orellana García: Dirigió el proyecto, proporcionó documentación, realizó el análisis e interpretación de los resultados, generó estadísticas, elaboró y aprobó el informe final.



MSc. Denys Buedo Hidalgo: Proporcionó documentación y asesoramiento científico técnico sobre los procesos y formas de gestión y almacenamientos en Computadoras de Altas Prestaciones (HPC), revisó y contribuyó a la redacción del documento.

Ing. Dairon Canel Gómez: Realizó el análisis y diseño del software. Desarrolló las funcionalidades de la investigación y documentó el artículo.

Jesús Enrique Fuentes González: contribuyó al desarrollo del software propuesto.

