

Tipo de artículo: Artículo original

Módulo de Posicionamiento Global para la Plataforma Web ULTRON 2.0

Global Positioning Module for ULTRON Web Platform 2.0

Jorge Luis Fernández Domínguez^{1*} , <https://orcid.org/0000-0002-2584-3579>

¹ Departamento de Sistemas Digitales, Facultad 2, Universidad de las Ciencias Informáticas

* Autor para correspondencia: jlfernandez@uci.cu

Resumen

En el centro de Representación y Análisis de Datos se desarrolla la plataforma web ULTRON 2.0. Esta posee funcionalidades para la representación de estructuras socioeconómicas. Con el avance tecnológico actual, la representación de objetos móviles constituye una necesidad para la toma de decisiones de determinadas entidades del país, y los sistemas de posicionamiento global juegan un papel fundamental en este proceso. La investigación realizada tuvo como objetivo desarrollar un módulo para la plataforma web ULTRON 2.0 que permita a los usuarios capacitados representar el posicionamiento global de objetos móviles en un mapa. Para guiar el proceso de desarrollo se utilizó la metodología AUP-UCI, adaptada a la actividad productiva en la universidad. Para la implementación del módulo se utilizó NodeJS y ExpressJS para la programación del lado del servidor, y AngularJS y Angular-Material para la programación del lado del cliente; además, se usó JavaScript como lenguaje de programación y Visual Studio Code como Entorno de Desarrollo Integrado. El módulo desarrollado es capaz de representar la ruta por donde el objeto hizo su recorrido, así como mostrar los datos obtenidos de la distancia, el tiempo y la velocidad media.

Palabras clave: sistemas de posicionamiento global, objetos móviles, mapa, ruta, AUP-UCI.

Abstract

The ULTRON web platform 2.0 is developed in the Data Representation and Analysis center. It has functionalities for the representation of socioeconomic structures. With the current technological advance, the representation of mobile objects is a necessity for the decision-making of certain entities in the country, and global positioning systems play a fundamental role in this process. The research carried out aims to develop a module for the ULTRON web platform 2.0 that allows trained users to represent the global positioning of mobile objects on a map. The AUP-UCI methodology was used to guide the development process, adapted to the productive activity in the university. For the module's implementation it was used NodeJS and ExpressJS for server-side programming, and AngularJS and Angular-Material for programming on the client side; in addition, JavaScript was used as programming language and Visual Studio Code as Integrated Development Environment. The developed module is able to represent the route where the object made its way, as well as show the obtained data of the distance, the time and the average speed.

Keywords: global positioning systems, mobile objects, map, route, AUP-UCI.

Recibido: 24/10/2021
Aceptado: 27/01/2022



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional**
(CC BY 4.0)

Introducción

En la historia de la humanidad siempre se han utilizado técnicas de posicionamiento, muchas de ellas por factores naturales, tales como las estrellas, el Sol, el viento, las formaciones rocosas, entre otros métodos. El hombre ha aprendido a interpretar estos elementos, haciéndolos imprescindibles en su orientación. Sin embargo, ha recurrido al uso de sus propias creaciones para obtener una mayor comprensión del espacio geográfico en que se encuentra.

La forma tradicional de posicionamiento, es decir, de conocer una posición, es utilizar un mapa localizando puntos conocidos mediante la interpretación del mismo y la observación real directa. Una vez localizada la posición actual se puede utilizar la brújula como un objeto clave para decidir hacia dónde ir.

Con el transcurso del tiempo, la era de la informatización se hace cada vez más potente. Surgen nuevos adelantos en la informática y cada día se aprecia más la dependencia del hombre hacia los ordenadores. Con el avance científico y tecnológico, el desarrollo de la información y el conocimiento han sido determinantes en los resultados económicos y beneficios sociales. Esto ha provocado un impulso de los mismos, así como de las necesidades de progreso en todas las esferas de la sociedad.

Acorde con los continuos cambios tecnológicos se encuentran los sistemas de posicionamiento global. Estos ofrecen de manera eficiente, en la obtención de información sobre la situación geográfica y la orientación, la posición geográfica en cualquier parte del planeta con una precisión hasta de centímetros.

Este servicio no se limita sólo a objetos estáticos. Entre los parámetros que ofrece, además de la latitud y longitud de la ubicación de un objeto móvil, se brinda la altitud a la que se encuentra, un aproximado del margen de error, y la velocidad a la que se desplaza. Esta última es medida mediante el tiempo que se demora en recorrer una distancia por lo que es una media y no la velocidad real a la que se desplaza el objeto móvil (Cruz Castro, y otros, 2013).

Los sistemas de posicionamiento global son, actualmente, una herramienta indispensable en todos los sistemas de transporte del mundo pues sirven de apoyo a la navegación aérea, terrestre y marítima. Los servicios de emergencia y socorro en casos de desastre dependen de los mismos para la localización y coordinación horaria de misiones para salvar vidas. Actividades cotidianas como operaciones bancarias, de telefonía móvil e incluso de las redes de distribución eléctrica, ganan en eficiencia gracias a la exactitud cronométrica que proporcionan. Agricultores,



topógrafos, geólogos e innumerables usuarios trabajan de forma eficiente, segura, económica y precisa gracias a sus señales accesibles y gratuitas.

Cuba, a pesar de ser un país subdesarrollado, debido al proceso de Informatización de la Sociedad, ha decidido incorporar poco a poco estos servicios en distintas tareas de índole profesional. Para ello cuenta con la Universidad de las Ciencias Informáticas (UCI) como principal promotora de esta importante misión.

En el centro de Representación y Análisis de Datos, en dicha universidad, se desarrolla la plataforma web ULTRON, en su versión 2.0. La misma está compuesta por varios módulos que permiten la visualización, localización y consulta geográfica de objetivos socioeconómicos. En la actualidad todo el proceso de representación de datos en la plataforma se realiza a través de puntos, líneas y polígonos, los cuales constituyen estructuras de interés para una organización. Sin embargo, se necesita representar el posicionamiento global que adopta determinado objeto móvil, ya sea una persona o un vehículo. Este servicio favorecería la actividad productiva de determinadas entidades del país, pues contribuiría al ahorro de innumerables recursos, combustible y tiempo en la planificación de sus actividades. Por tal razón, la ausencia de esta funcionalidad, afectaría considerablemente el proceso de toma de decisiones de estas entidades.

Luego del análisis de la situación existente se identifica el siguiente problema de investigación: ¿Cómo representar el posicionamiento global de objetos móviles en la plataforma web ULTRON 2.0? Para dar solución al problema anteriormente planteado, se propone como objetivo general: Desarrollar un módulo para la plataforma web ULTRON 2.0 que permita la representación del posicionamiento global de objetos móviles en el mapa.

Materiales y métodos

Para la selección del sistema de posicionamiento global adecuado se analizan cuatro sistemas fundamentales que actualmente se usan en el mundo: GPS, GLONASS, BeiDou y Galileo. BeiDou es un sistema chino que funciona como sistema regional dentro de la región del Pacífico asiático. Galileo surge como una iniciativa europea para desarrollar un sistema de radionavegación mediante satélites dedicado a satisfacer las necesidades de la comunidad civil mundial. Estos sistemas constituyen soluciones más modernas y más avanzadas tecnológicamente, pero actualmente tienen sólo alcance regional. Tanto el sistema ruso GLONASS como el sistema norteamericano GPS, constituyen los más conocidos y usados de todos los sistemas de posicionamiento global. Aunque son muy similares, por su funcionamiento a través de señales satelitales, presentan diferencias en cuanto a la precisión de sus datos. La



precisión de GLONASS, hoy en día, es menor en términos medios que la del GPS. Si los receptores de GPS teóricamente permiten determinar la localización en un espacio abierto con un error de 3-4 metros, para los receptores de GLONASS el error es de 7-10 metros. Por tal razón, para la representación del posicionamiento global de objetos móviles en la plataforma web ULTRON 2.0 se hace uso del GPS.

Para la selección del formato de intercambio para GPS se analizan los ficheros GeoJSON, KML y GPX, los cuales son muy usados en la actualidad por diferentes usuarios en el mundo. A continuación se muestra una tabla donde se observan las principales diferencias entre estos ficheros (*Ver Tabla 1*).

Tabla 1. Comparación entre GPX, KML y GeoJSON

	GPX	KML	GeoJSON
Información que almacena	Puntos de interés, recorridos y rutas	Puntos de interés, polígonos y descripciones textuales	Puntos, líneas y polígonos
Estándar en que se basa	XML	XML y OGC	JSON y OGC
Cantidad de información	Mayor cantidad de información posible	Sólo información básica	Sólo información básica
Tamaño del archivo	Menos de 100KB	Más de 100KB	Más de 100KB

Se ha decidido seleccionar GPX (Formato de Intercambio GPS (*GPS eXchange Format*)) como formato de intercambio para GPS pues es un esquema basado en XML pensado para transferir y almacenar datos GPS entre aplicaciones y servicios web en Internet. Se puede usar para describir puntos de interés, recorridos y rutas. Fue diseñado desde el inicio para ser el estándar de intercambio debido a que hereda todos los beneficios de XML de interoperabilidad y extensibilidad (Foster, 2004).

La metodología empleada para el desarrollo de la presente investigación se basa en el método de desarrollo ágil: Proceso Unificado Ágil (AUP). Esta es una versión simplificada de RUP (Proceso Unificado de Software), que describe la forma de desarrollar aplicaciones de software de negocio, usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. En la UCI, se decide hacer una variación de esta metodología (AUP-UCI), de forma tal que se adapte al ciclo de vida definido para la actividad productiva en la universidad, logrando estandarizar el proceso de desarrollo de software. De las cuatro fases que propone AUP (Inicio, Elaboración, Construcción, Transición), se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la



misma, se unifican las restantes 3 fases de AUP en una sola titulada Ejecución y se agrega la fase de Cierre (Rodríguez Sánchez, 2015).

El esfuerzo principal en la disciplina Requisitos es desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto. Existen tres formas de encapsular los requisitos [Casos de Uso del Sistema (CUS), Historias de usuario (HU) y Descripción de Requisitos por Proceso (DRP)], agrupados en cuatro escenarios condicionados por el Modelado de Negocio (Rodríguez Sánchez, 2015). En la investigación se utiliza el Escenario No 4: Historias de Usuario. Este se aplica a proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente acompaña al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Además, es recomendado en proyectos no muy extensos, pues las Historias de Usuario no deben poseer demasiada información.

Durante el transcurso de la investigación, se emplea PostgreSQL 9.6, el cual es un sistema de gestión de bases de datos objeto-relacional. Es uno de los sistemas de gestión de bases de datos de código abierto más potente del mercado. Es multiplataforma, soporta múltiples transacciones e integridad de datos. Presenta una estabilidad alta, gran seguridad de los datos y facilidad de administración e implementación de estándares (Zea Ordóñez, y otros, 2017). Con el fin de que PostgreSQL soporte objetos geográficos se ha desarrollado el módulo PostGIS, convirtiéndola en una base de datos espacial que puede usarse en Sistemas de Información Geográfica. Añade soporte para objetos geográficos que permite consultas de ubicación para ser ejecutadas en SQL. PostGIS 3.0 es una versión estable y liberada de PostGIS. Posee soporte de datos ráster y vectoriales. Es compatible con el servidor de mapas MapServer (Ramsey, y otros, 2015).

NodeJS 10.15.10 es un entorno de programación en la capa del servidor basado en el lenguaje de programación JavaScript. Fue creado con el enfoque de ser útil en la creación de aplicaciones web altamente escalables, como por ejemplo, servidores web. Aumenta la velocidad de desarrollo (NodeJS, 2018).

ExpressJS 4.17 es un framework del lado del servidor para el desarrollo de aplicaciones web con NodeJS. Es un marco de desarrollo rápido y flexible que cuenta con una gran comunidad de desarrollo, lo cual es importante por el volumen de documentación existente para el desarrollo de proyectos (Nallu, 2015).



Sequelize 3.0.0 es el framework ORM (Object-Relational Mapping) de NodeJS que soporta la conexión con los SGBD: MySQL, PostgreSQL, MariaDB y SQLite. Es fácil y flexible de usar al momento de mapear las tablas en Modelos de Objetos Relacionales (Pavón, 2017).

AngularJS 8.0 es un proyecto de código abierto, escrito en JavaScript que contiene un conjunto de librerías útiles para el desarrollo de aplicaciones SPA (Single Page Applications / Aplicaciones de una Sola Página). Este propone una serie de patrones de diseño para llevarlas a cabo. En pocas palabras, es lo que se conoce como un framework para el desarrollo, en este caso con programación del lado del cliente (Basalo, y otros, 2014).

Redis-Server 2.2 es un motor de base de datos libre de tipo clave-valor persistentes que residen en memoria RAM y envía el conjunto de datos almacenados al disco duro. Resuelve de manera sencilla y eficiente problemas que no necesitan la complejidad de las bases de datos relacionales, como lo es en la mayoría de los casos de usos el almacén y gestión de estructuras de datos temporales. (Younes, 2016).

MapServer 7.4.2 es un entorno de desarrollo o plataforma de código abierto para la publicación de datos espaciales y aplicaciones cartográficas en Internet/Intranet. Permite visualizar, consultar y analizar información geográfica a través de la red mediante la tecnología Internet MapServer. Ofrece la posibilidad de ser utilizado como servidor de mapas de terceros programas y admite múltiples formatos de datos vectoriales (Sphinx, 2014).

OpenLayers 4.0 es una biblioteca de código abierto del lado del cliente, que emplea JavaScript para mostrar mapas interactivos en los navegadores web. Ofrece una API (Application Programming Interface) para acceder a diferentes fuentes de información cartográfica en la red (LLasac Huaca, y otros, 2015).

Git 2.9 es un sistema de control de versiones distribuido, libre y de código abierto, diseñado para manejar desde pequeños hasta grandes proyectos, con velocidad y eficiencia (Loeliger, 2009).

JavaScript 1.8.5 fue diseñado para ser un lenguaje de elaboración de scripts que pudiera vincularse a archivos HTML. Este se ha convertido en un lenguaje con una implantación masiva en todos los entornos (personales y empresariales). Permite un rápido desarrollo de la interfaz de usuario, escribir código libre de errores y nuevo soporte para HTML5. Además, brinda soporte para múltiples idiomas, un amplio conjunto de plugins; así como una edición rápida, fácil y eficiente gestión de proyectos. Es un lenguaje robusto, seguro, multitarea y portable (Centeno Defas, y otros, 2016).



Resultados y discusión

Los requisitos son la base de todos los proyectos que involucren el desarrollo de software. Al acordar los requisitos para el sistema, se proporcionan las bases para la planificación del desarrollo de un proyecto de software. La ingeniería de requisitos sirve para determinar las necesidades que un cliente tiene en la implementación o desarrollo de software. Estas necesidades se obtienen mediante el proceso de recopilación, análisis y verificación de la especificación de los requisitos del software de una manera completa y correcta (García Velázquez, 2016). Según (Pressman, 2010) los requisitos de software constituyen las necesidades de los clientes, las funcionalidades y las restricciones que debe cumplir el sistema. Los mismos se clasifican en funcionales y no funcionales.

Los requisitos funcionales (RF) son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requisitos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer (Sommerville, 2016). A continuación, se identifican los requisitos funcionales del módulo a desarrollar:

RF1. Importar fichero GPX: El módulo debe importar ficheros GPX que contengan la información, recogida por el GPS, del recorrido del objeto.

RF2. Visualizar la ruta tomada por el objeto: El módulo debe representar en el mapa la ruta por donde el objeto realizó su recorrido.

RF3. Seleccionar el color de la ruta: El módulo debe permitir que el usuario seleccione un color determinado para la representación de la ruta.

RF4. Determinar la distancia recorrida por el objeto: El módulo debe ser capaz de determinar la distancia recorrida por el objeto.

RF5. Determinar el tiempo de recorrido del objeto: El módulo debe determinar el tiempo que demoró el objeto en desplazarse.

RF6. Determinar la velocidad del objeto en su recorrido: El módulo debe determinar la velocidad a la que se movió el objeto durante su recorrido, dependiendo de los valores de la distancia y el tiempo alcanzados.

RF7. Exportar fichero PDF: El módulo debe exportar la información necesaria para el usuario en un fichero PDF.

RF8. Limpiar campos: El módulo debe quitar la información que se muestra en el mapa de la interfaz.

Los requisitos no funcionales (RNF) son aquellos que se refieren a las propiedades emergentes del sistema como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, definen las restricciones



del sistema como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema. Los requisitos no funcionales rara vez se asocian con características particulares del sistema; más bien, estos requisitos especifican o restringen las propiedades emergentes del sistema (Sommerville, 2016). El módulo debe cumplir con los siguientes requisitos no funcionales que han sido agrupados según su clasificación:

Requisitos de hardware:

RNF1. La PC cliente debe cumplir con los siguientes requisitos mínimos: Tarjeta de red con velocidad de 100Mbps y transmisión Full Duplex, una RAM de 2 GB o superior, procesador a 3 GHz o superior y almacenamiento de 40 GB en adelante.

RNF2. La PC servidor de mapas debe tener como mínimo 2 GB de RAM, 40 GB de disco duro, un procesador de 3 GHz y una tarjeta de red con velocidad de 100Mbps y transmisión Full Duplex.

RNF3. La PC servidor de base de datos debe tener como mínimo 2 GB de RAM, 40 GB de disco duro, un procesador de 3 GHz y una tarjeta de red con velocidad de 100Mbps y transmisión Full Duplex.

RNF4. La PC servidor de aplicaciones debe cumplir con los siguientes requisitos mínimos: RAM de 4 GB, 40 GB de disco duro, un procesador de 3 GHz y una tarjeta de red con velocidad de 100Mbps y transmisión Full Duplex.

Requisitos de software:

RNF5. La PC cliente debe poseer un navegador web que cumpla con los estándares de World Wide Web Consortium (W3C).

RNF6. La PC servidor debe tener NodeJS 10.15.10 o superior.

RNF7. La PC servidor debe tener PostgreSQL 9.6 como Sistema Gestor de Base de Datos.

RNF8. La PC servidor debe tener PostGIS 3.0 o superior como extensión de PostgreSQL para el soporte de datos espaciales.

RNF9. La PC servidor debe tener MapServer 7.4.2.

RNF10. La PC servidor debe tener Redis-Server 2.2 o superior.

Para el desarrollo del módulo se decide hacer uso del patrón arquitectónico Modelo-Vista-Controlador (MVC), pues es empleado en la plataforma web ULTRON 2.0 y es el que sigue el framework AngularJS en su versión 8.0. Según (Medina, 2016) el patrón Modelo-Vista-Controlador divide una aplicación, subsistema o módulo en tres áreas:



procesamiento, entrada y salida. El Modelo es el que encapsula los datos y la funcionalidad central; es independiente de las representaciones específicas de la salida o el comportamiento de la entrada. La Vista se encarga de presentar la información al usuario obteniendo los datos del modelo; puede haber múltiples vistas del modelo y cada una de ellas tiene asociado un componente Controlador. El Controlador es el que recibe y responde a las entradas o eventos (usualmente acciones del usuario), estos eventos son trasladados a solicitudes de servicio al modelo o a la vista.

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso. Se centra en cómo los requisitos, junto con otras restricciones del entorno de implementación, tienen impacto en el módulo, siendo la principal vía de acceso en la actividad de implementación (Jacobson, y otros, 2004). Entre los artefactos que genera el modelo de diseño se encuentra el Diagrama de Clases del Diseño donde se muestran las clases, interfaces y colaboraciones, así como las relaciones entre ellas. A continuación, se expone la representación del Diagrama de Clases del Diseño con Estereotipos Web “Determinar Valores”, correspondiente a la interfaz “Geolocalización” (Ver Figura 1). Esta representación contribuye a una correcta implementación del producto.

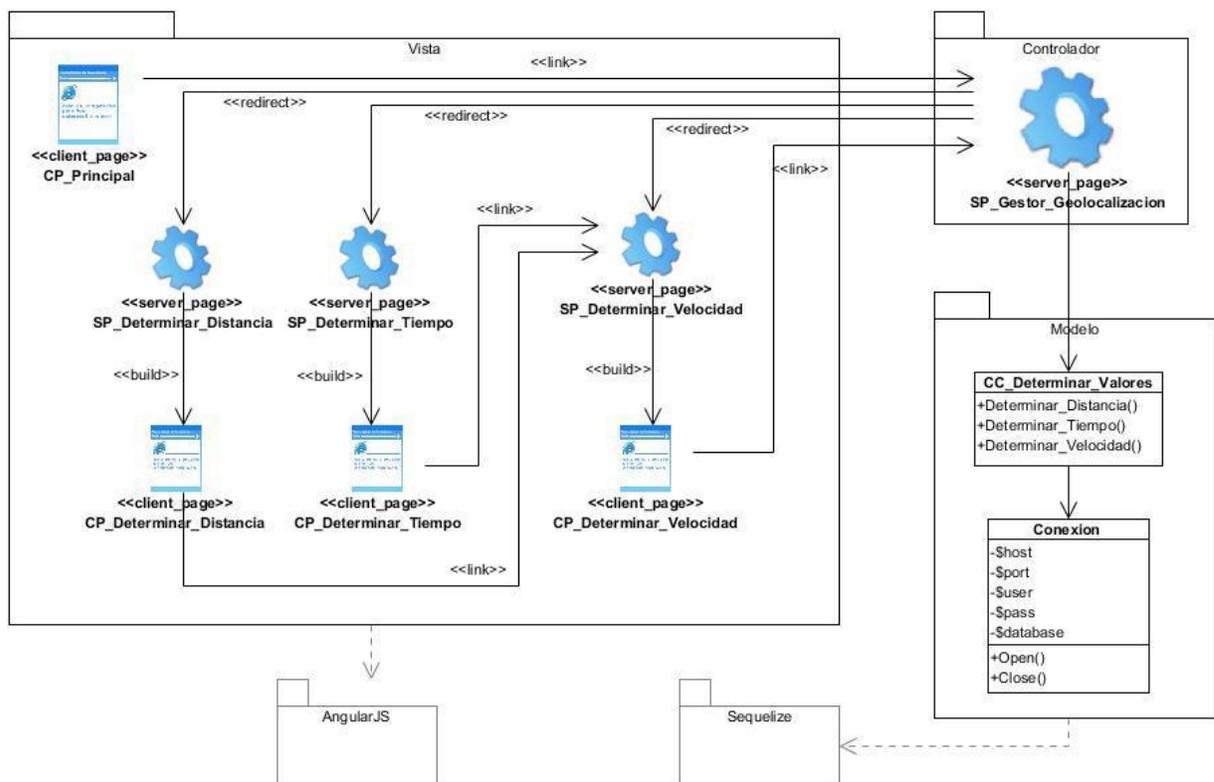


Figura 1: Diagrama de Clases del Diseño con Estereotipos Web: “Determinar Valores”



Los patrones de diseño constituyen la base para la búsqueda de soluciones a problemas comunes en el proceso de desarrollo de software. Se caracterizan por su reusabilidad, flexibilidad y aplicabilidad a diferentes problemas de diseño en diversas circunstancias. El patrón es un esquema de solución que se aplica a un tipo de situación, esta aplicación del mismo no es mecánica, sino que requiere de adaptación y matrices (Larman, 2003). Los patrones GRASP son una colección de principios de diseño orientados a objetos que guían la asignación de responsabilidades sobre los objetos. Son un intento de documentar lo que los diseñadores expertos probablemente conozcan intuitivamente (Trellini, 2015). Para el desarrollo del módulo se hace uso de los siguientes patrones, según (Trellini, 2015):

Controlador: Es el encargado de asignar a clases específicas la responsabilidad de controlar el flujo de eventos de la aplicación. La utilización del patrón se evidencia en la clase SP_Gestor_Geolocalizacion, la cual controla el flujo de eventos del módulo.

Experto: Es el encargado de asignar responsabilidades a las clases que tienen la información necesaria para cumplir dicha responsabilidad. De este modo se obtendrá un diseño manteniendo un bajo acoplamiento, incrementar la claridad, el encapsulamiento y la reusabilidad. Este patrón se aplica en la clase Conexión, pues esta cuenta con la información suficiente para llevar a cabo una responsabilidad, que en este caso es establecer las conexiones.

Creador: Es el responsable de crear una nueva instancia de una clase con la información necesaria para realizar la creación del objeto. De este modo se ayuda a mantener un bajo acoplamiento, incrementar la claridad, el encapsulamiento y la reusabilidad. Este patrón se ve aplicado cuando las clases servidoras SP_Determinar_Distancia, SP_Determinar_Tiempo y SP_Determinar_Velocidad construyen respectivamente a las clases cliente CP_Determinar_Distancia, CP_Determinar_Tiempo y CP_Determinar_Velocidad.

Bajo Acoplamiento: Es el encargado de lograr una menor dependencia y bajo impacto de cambio entre las clases. De tal forma que, en caso de producirse una modificación en alguna de ellas, no incremente el acoplamiento a un nivel que produzca resultados negativos típicos del alto acoplamiento entre las clases. La utilización de este patrón se ve empleado en todo el diseño del módulo ya que cada clase sólo se relaciona con otras que necesite. De esta forma, si se realiza algún cambio en alguna de ellas se afectarían lo menos posible las otras clases.

Alta Cohesión: Es el encargado de que la información que contiene una clase debe de ser coherente, altamente relacionada con dicha clase y que la misma no realice una excesiva cantidad de trabajo. De este modo se garantiza una mejor claridad y facilidad de comprensión del diseño. Se simplifica el mantenimiento y la introducción de



mejoras, se logra una menor dependencia y se especifican los propósitos de cada objeto en el sistema. Este patrón se ve reflejado en la clase CC_Determinar_Valores pues contiene la información y los métodos correspondientes a ella y no se sobrecarga con información de otras clases.

El modelo de implementación está compuesto por componentes que establecen la composición física de la implementación del sistema. También describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación, además en el lenguaje o los lenguajes de programación utilizados, y cómo dependen los componentes unos de otros (Jacobson, y otros, 2004).

Un componente es una parte modular, desplegable y reemplazable de un sistema, o sea, son las piezas reutilizables a partir de las cuales se pueden construir los sistemas. Emparejado a ello los diagramas de componentes son empleados para describir la estructura física del modelo de implementación del sistema y mostrar las relaciones entre sus elementos. De este modo se refleja una estructura de alto nivel. Representan las dependencias entre componentes de software incluyendo componentes de código fuente, componentes del código binario, y componentes ejecutables. Se rigen por la filosofía de la alta cohesión interna y el bajo acoplamiento externo (Jacobson, y otros, 2004). A continuación, se presenta el Diagrama de Componentes “Determinar Valores” (Ver Figura 2).

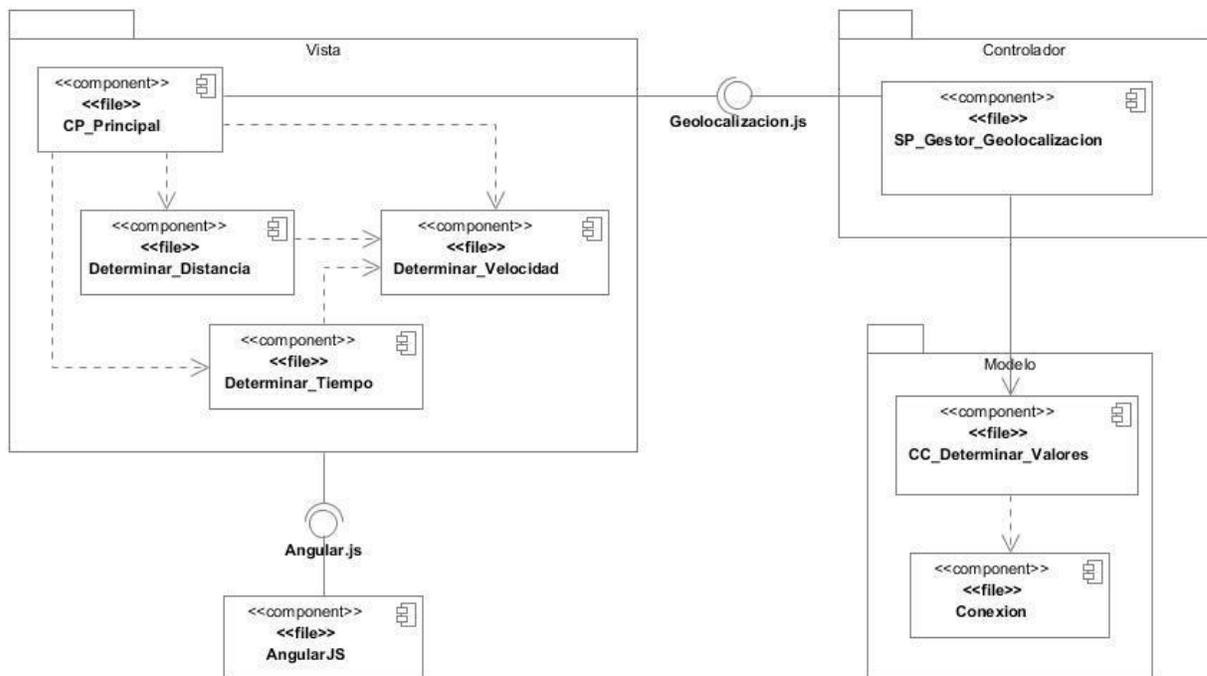


Figura 2: Diagrama de Componentes: “Determinar Valores”



El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo (Jacobson, y otros, 2004). El Diagrama de Despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución, los vínculos de comunicación entre ellos y las instancias de los componentes. El diagrama está compuesto por nodos, dispositivos y conectores. En resumen, el Diagrama de Despliegue es un modelo de objetos que describe los componentes, dispositivos, hardware y software que se desplegarán en el ambiente seleccionado (Pressman, 2010). Con el objetivo de representar los recursos de cómputo que necesita el módulo para su correcto funcionamiento se muestra a continuación el Diagrama de Despliegue correspondiente a la propuesta de solución (Ver Figura 3).

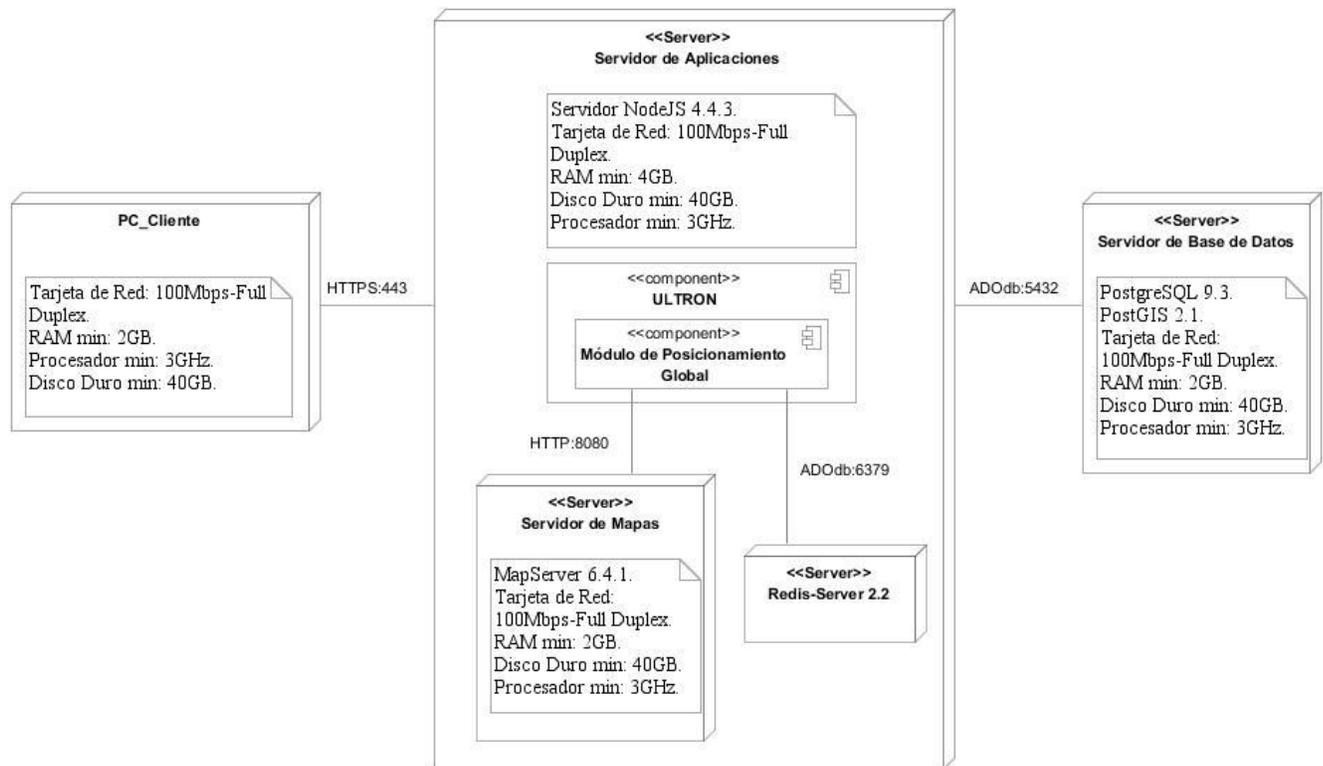


Figura 3: Diagrama de Despliegue

Para el despliegue del módulo se debe contar al menos con una PC para el especialista encargado de representar el posicionamiento global de un objeto móvil. La PC debe conectarse al servidor de aplicaciones web usando el protocolo HTTPS, por el puerto 443. Este, a su vez, se comunica con el servidor de base de datos utilizando el protocolo ADOdb, por el puerto 5432. El servidor de aplicaciones web incluye a la plataforma web ULTRON 2.0



junto al módulo desarrollado, al servidor de mapas y Redis-Server. Los dos últimos constituyen servidores virtuales. La comunicación entre la plataforma y el servidor de mapas se realiza a través del protocolo HTTP, usando el puerto 8080. Mientras, la comunicación con Redis-Server se logra usando el protocolo ADOdb, por el puerto 6379.

Como resultado del desarrollo del módulo de posicionamiento global para la plataforma web ULTRON 2.0 se obtiene una aplicación que brinda a los usuarios la posibilidad de representar el posicionamiento global de objetos móviles en un mapa. La solución permite importar ficheros GPX que contienen la información que recoge el GPS durante la trayectoria del objeto. Así, usa estos ficheros para trazar la ruta recorrida y determinar los valores de distancia, tiempo y velocidad media alcanzados. Cada ruta especifica la posición de los puntos inicial y final, así como la de determinados puntos de interés. La información que se maneja en el módulo es almacenada en una base de datos, posibilitando la existencia de un historial de todas las representaciones que realiza el usuario. Además, el usuario puede exportar los datos obtenidos y la imagen de cada ruta en un fichero PDF.

Conclusiones

A partir del contenido del trabajo presentado, de los antecedentes revisados en la literatura y su análisis, se arriba a las siguientes conclusiones:

La aplicación de la metodología de desarrollo AUP-UCI posibilitó que se obtuvieran los artefactos generados del proceso ingenieril, proporcionando comprensión a la solución desarrollada y favoreciendo la rigurosidad en su diseño. Los elementos significativos de las herramientas y tecnologías a utilizar lograron satisfacer las necesidades existentes para el desarrollo del módulo de posicionamiento global para la plataforma web ULTRON 2.0.

La identificación de los requisitos del módulo propició un mejor entendimiento de la solución propuesta al equipo de desarrollo en cuanto a las funcionalidades y características que debe poseer.

La selección del patrón arquitectónico Modelo-Vista-Controlador y la utilización de los patrones de diseño GRASP contribuyeron al diseño de la solución, evidenciando la utilización de buenas prácticas durante el desarrollo de la solución propuesta.

Con la implementación del módulo, se logró obtener un software que permite la representación del posicionamiento global de objetos móviles en un mapa, a través del uso del Sistema de Posicionamiento Global (GPS).



Conflictos de intereses

El autor de la presente investigación declara que no posee conflictos de intereses.

Contribución de los autores

1. Conceptualización: Jorge Luis Fernández Domínguez
2. Curación de datos: Jorge Luis Fernández Domínguez
3. Análisis formal: Jorge Luis Fernández Domínguez
4. Investigación: Jorge Luis Fernández Domínguez
5. Metodología: Jorge Luis Fernández Domínguez
6. Administración del proyecto: Jorge Luis Fernández Domínguez
7. Recursos: Jorge Luis Fernández Domínguez
8. Software: Jorge Luis Fernández Domínguez
9. Validación: Jorge Luis Fernández Domínguez
10. Visualización: Jorge Luis Fernández Domínguez
11. Redacción – borrador original: Jorge Luis Fernández Domínguez
12. Redacción – revisión y edición: Jorge Luis Fernández Domínguez

Financiamiento

Se financió por medios propios.

Referencias

- Basalo, Alberto y Álvarez, Miguel Angel. 2014. DesarrolloWeb.com. *¿Qué es AngularJS?* [En línea] agosto 28, 2014. [Citado el: noviembre 25, 2018.] <http://www.desarrolloweb.com/articulos/que-es-angularjs-descripcion-framework-javascript-conceptos.html>.
- Centeno Defas, Elvis Paúl y Cordonez Suntasig, Sergio Paúl. 2016. *Implementación de un sistema de gestión documental administrativa*. Latacunga, Ecuador : s.n., 2016.
- Cruz Castro, Liester, Rodríguez Larrazabal, Yudiel y Céspedes Zamora, Carlos Karen. 2013. *SISTEMA PARA LA LOCALIZACIÓN GPS EN TIEMPO REAL SOBRE DISPOSITIVOS MÓVILES*. La Habana : XI CONGRESO CUBANO DE INFORMÁTICA Y GEOCIENCIAS (GEOINFO'2013), 2013.



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

- Foster, Dan. 2004. *GPX: the GPS Exchange Format*. [En línea] 2004. [Citado el: noviembre 23, 2018.] <http://www.topografix.com/gpx.asp>.
- García Velázquez, Luis Ambrosio. 2016. RECI Revista Iberoamericana de las Ciencias Computacionales e Informática. [En línea] julio 2016. [Citado el: enero 23, 2019.] <http://reci.org.mx/index.php/reci/article/view/49/221>.
- Jacobson, Booch y Rumbaugh. 2004. *El Proceso Unificado de Desarrollo. Vol I, Vol II*. La Habana: Félix Varela : s.n., 2004.
- Larman, Craig. 2003. *UML y Patrones 2da Edición*. España : Pearson Educación : s.n., 2003.
- LLasac Huaca, Juan Carlos y Alvarado Sánchez, Daniel Fernando. 2015. *BistStrema*. [En línea] marzo 5, 2015. [Citado el: noviembre 25, 2018.] <https://dspace.ups.edu.ec/bitstream/123456789/10112/1/UPS%20-%20ST001647.pdf>.
- Loeliger, Jon. 2009. *Version control with Git*. s.l. : O'Reilly, 2009. ISBN: 978-0-596-52012-0.
- Medina, Cesar Julio Bustacara. 2016. Model-View-Controller Pattern. [En línea] 2016. [Citado el: enero 24, 2019.] https://sophia.javeriana.edu.co/~cbustaca/docencia/DSBP-2016-03/presentaciones/Model_View_Controller_Pattern.pdf.
- Nallu, Surya. 2015. *Introduction to Express.js*. [En línea] febrero 6, 2015. [Citado el: noviembre 25, 2018.] http://www.teach.cs.toronto.edu/~csc309h/winter/posted_tutorials/Introduction_to_Express.pdf.
- NodeJS. 2018. *Node.js*. [En línea] Joyent Inc., 2018. [Citado el: noviembre 25, 2018.] <http://nodejs.org/>.
- Pavón, Santiago. 2017. *Desarrollo del Servidor Quiz Sequelize*. [En línea] 2017. [Citado el: noviembre 25, 2018.]
- Pressman, Roger. 2010. *Ingeniería del Software. Un enfoque práctico*. Séptima Edición. s.l. : McGraw-Hill, 2010. ISBN: 978-607-15-0314-5.
- Ramsey, P y Santilli, S. 2015. *Postgis Documentation*. 2015.
- Rodríguez Sánchez, Tamara. 2015. *Metodología de desarrollo para la Actividad productiva de la UCI*. La Habana : s.n., 2015.
- Sommerville, Ian. 2016. *Software engineering*. Tenth Edition. s.l. : Pearson Education, 2016. ISBN: 978-0-13-394303-0.
- Sphinx. 2014. MapServer. [En línea] 2014. [Citado el: noviembre 25, 2018.] <http://www.mapserver.org/es/about.html#about>.
- Trellini, Ariel. 2015. *Arquitectura y Diseño de Sistemas*. 2015.



Younes, Georges. 2016. *Integration Challenges of Pure Operation-based CRDTs in Redis*. Lisboa : s.n., 2016. ISBN: 978-1-4503-4775-4.

Zea Ordóñez, Mariuxi Paola, Molina Ríos, Jimmy Rolando y Redrován Castillo, Fausto Fabián. 2017. *Administración de bases de datos con PostgreSQL*. Alicante : Área de Innovación y Desarrollo, S.L., 2017. ISSN: 978-84-946684-6-3.



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional**
(CC BY 4.0)