

Arquitectura de Software para el desarrollo de herramienta Tecnológica de Costos, Presupuestos y Programación de obra¹

Software Architecture for the development of a Technological tool for Costs, Budgets and Work Scheduling

Javier Alfonso Cárdenas-Gutiérrez², Ender José Barrientos-Monsalve³, Leonardo Molina-Salazar⁴

Artículo recibido en marzo 16 de 2021; artículo aceptado en julio 02 de 2021

Este artículo puede compartirse bajo la [Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/) y se referencia usando el siguiente formato: Cárdenas-Gutiérrez, J.A., Barrientos-Monsalve, E. J. y Molina-Salazar, L. (2022). Arquitectura de Software para el desarrollo de herramienta Tecnológica de Costos, Presupuestos y Programación de obra. *I+D Revista de Investigaciones*, 17(1), 85-95.

Resumen

Los softwares académicos han demostrado ser fuertes herramientas que impulsan los procesos de aprendizaje dentro de todo tipo de instituciones educativas a lo extenso del mundo. Por lo tanto, implementar su uso se vuelve cada vez una tarea más imperativa debido al auge y crecimiento exponencial de la tecnología. El objetivo de este trabajo es diseñar una arquitectura de software para la materia Costos, Presupuestos y Programación de Obra, del programa de Ingeniería Civil de la Universidad Francisco de Paula Santander. La metodología se basa en el desarrollo teórico de las clases teóricas, fundamentando el proceso de manera lógica evitando contratiempos que en la práctica convencional podrían ocurrir. Los resultados muestran una arquitectura sólida y funcional, que permitirá posteriormente el desarrollo exitoso de un software académico y su mantenimiento. Esta herramienta permitirá el apalancamiento del proceso de aprendizaje de los estudiantes, debido a que incentiva el autoaprendizaje y el interés por la materia.

Palabras clave: Arquitectura de software, construcción, costos, desarrollo de software, presupuestos, programación de obra, software académico.

Abstract

Academic software has proven to be strong tools that drive learning processes within all types of educational institutions around the world. Therefore, implementing their use is becoming an increasingly imperative task due to the exponential growth of technology. The objective of this work is to design a software architecture for the subject Costs, Budgets and Work Scheduling, of the Civil Engineering program of the Universidad Francisco de Paula Santander. The methodology is based on the theoretical development of the theoretical classes, basing the process in

¹ Artículo de investigación, con enfoque cualitativo, resultado de un proyecto de investigación en curso, perteneciente al área de ingeniería civil, subárea de costos, presupuestos y programación de obra, desarrollado en el Grupo de Investigación en Transporte y Obras Civiles, fue financiado por la Universidad Francisco de Paula Santander (Cúcuta, Colombia). Dirección: Avenida gran Colombia N° 12E-96. Fecha de inicio: agosto 2020. Fecha de terminación: febrero 2021.

² Máster en Dirección de Proyectos. Vinculado al Grupo de Investigación en Transporte y Obras Civiles, Universidad Francisco de Paula Santander (Cúcuta, Colombia). Dirección: Avenida gran Colombia N.º 12E-96. PBX: (7) 6884239. ORCID ID: <https://orcid.org/0000-0002-9894-0177> Correo electrónico institucional: javieralfonosocg@ufps.edu.co

³ Doctor en Ciencias Gerenciales, Universidad Fundación de estudios superiores Comfanorte. Vinculado al Grupo de Investigación GTN, (Cúcuta, Colombia). Dirección: Avenida 5 con calle 15 centro. PBX: (7) 5829292. ORCID ID: <https://orcid.org/0000-0001-6673-0223> Correo electrónico institucional: ej_barrientos@fesc.edu.co

⁴ Estudiante de Ingeniería Civil., Universidad Francisco de Paula Santander. Vinculado al Grupo de Investigación en Transporte y Obras Civiles, Universidad Francisco de Paula Santander (Cúcuta, Colombia) Dirección: Avenida gran Colombia N.º 12E-96. PBX: (7) 6884239. ORCID ID: <https://orcid.org/0000-0003-1431-3633>, Correo electrónico institucional: davidleonardoms@ufps.edu.co

a logical way avoiding setbacks that in conventional practice could occur. The results show a solid and functional architecture, which will later allow the successful development of an academic software and its maintenance. This tool will leverage the learning process of students, because it encourages self-learning and interest in the subject.

Keywords: Software architecture, construction, costs, software development, budgeting, site scheduling, academic software.

Introducción

Las Tecnologías de la Información y las Comunicaciones (TIC) son un pilar de la economía, del Gobierno, de las relaciones culturales y sociales las cuales forman de manera integral a las personas. Las TIC influyen en el progreso del desarrollo humano y de la sociedad en general (Vinuesa & Córdova, 2019). Estas tecnologías abarcan un entorno de interés para la mayoría de las personas, volviéndose así cada vez más indispensable en la vida de todos nosotros (Rodríguez Calderón et al., 2019).

A su vez, el proceso de enseñanza debe permitir egresar a un estudiante preparado a las demandas del mundo profesional (Alonso Rodriguez, 2019). Sin embargo, el desarrollo exponencial de la tecnología influye de manera directa en los métodos utilizados dentro de las instituciones de educación superior, ya que, las estrategias didácticas quedan obsoletas con el transcurso del tiempo (Cova et al., 2008), obstaculizando los objetivos previstos dentro del proceso de aprendizaje (Hoyos Salcedo et al., 2021).

Por lo tanto, las aulas de clase y los métodos de enseñanza en general demandan nuevas herramientas pedagógicas que se adapten al desarrollo de la tecnología y que busquen el aprovechamiento de las ventajas que ofrecen al sector de la educación (Báez-Pérez et al., 2019). Estas tecnologías deben formar profesionales competentes y altamente capacitados que cumplan las exigencias del mundo actual (Alvarado Cardenas & Finol de Govea, 2019).

Un desarrollo idóneo en la educación superior, garantiza el desarrollo económico, político y social de un país, por lo tanto, las estrategias utilizadas por los docentes universitarios deben permitir que el estudiante se desarrolle como una persona más reflexiva, crítica y creativa, con el fin de que pueda proponer y establecer soluciones asertivas y diferentes a todos los problemas personales o profesionales de una manera más competente (Taipe et al., 2020). De tal forma, promover y potencializar este tipo de servicios que ofrecen las instituciones de educación superior es una tarea completamente imperativa (Díaz et al., 2016).

Una de estas estrategias puede ser el software educativo, el cual es un conjunto de herramientas computarizadas, diseñadas específicamente para la enseñanza y el aprendizaje de diferentes temáticas dentro de cualquier

área del conocimiento (Almaguel Guerra et al., 2016), con el propósito de apalancar el proceso de aprendizaje y desarrollar conceptos, habilidades y destrezas de una manera más fácil (Burgos Moreira & Velasco García, 2018).

En el sector de la construcción se ha presentado un rápido incremento en el desarrollo de obras civiles, permitiendo que los proyectos de construcción sean cada vez más complejos, debido a las altas exigencias del mercado, estableciendo así una mayor competitividad y unos estándares cada vez más altos. Por lo tanto, el ingeniero civil debe apoyarse en instrumentos que le permitan realizar su trabajo de una manera más eficiente (Salazar Alzate, 2017).

En la construcción, la correcta gestión de los costos de un proyecto de construcción es una de las tareas más importantes debido a que se establecen los recursos necesarios para el óptimo desarrollo de las actividades, con el fin de generar productos de calidad que satisfagan las exigencias y las expectativas del cliente (Silva Giraldo et al., 2018). Por lo tanto, es indispensable tener dentro del material pedagógico herramientas tecnológicas que complementen el proceso de aprendizaje y acerquen al estudiante al entorno profesional.

El objetivo de este trabajo es establecer una arquitectura de software para el manejo de costos, presupuesto y programación de obra, dentro del programa de Ingeniería Civil de la Universidad Francisco de Paula Santander, con el fin de actualizar los métodos de enseñanza tradicionales que se implementan hoy día dentro de las aulas de clases. La metodología utilizada consiste en elaborar una secuencia lógica que permita el desarrollo teórico y práctico dentro de una herramienta tecnológica, complementando los conocimientos aprendidos en clase y desarrollando el interés y el autoaprendizaje. Con los resultados obtenidos se podrá desarrollar y mantener un software académico dentro del plan de estudios de Ingeniería Civil, dentro de la materia, Costos Presupuestos y Programación de obra. Este software es el primer software educativo dentro de este programa y fomentará el uso de este tipo de herramientas, además de impulsar el desarrollo de habilidades, desarrollando profesionales más íntegros y competentes dentro del campo laboral.

Este trabajo de investigación se enfoca en el diseño de un software para el estudio de costos y presupuestos

caracterizado por ser didáctico y flexible para aplicarlo al plan de estudios de la carrera de ingeniería civil de la UFPS, sin embargo, se resalta que a nivel profesional se usan otros programas, que para uso educativo requieren de licencias muy costosas, como lo son el Procore, QuickBooks Desktop Enterprise, CoConstruct, génesis 2.2.5, tablas de Excel personalizadas entre otros, esto a medida que cada sector tenga una necesidad para el uso de cada uno de ellos. Ante esto y con la gran variedad de programas existentes para el costo y presupuestos de obra el diseño del presente software se adapta a los escenarios educativos de formación de la carrera de ingeniería civil y con ello aportando en los procesos de formación de los estudiantes de este programa de la UFPS.

Metodología

La metodología para el desarrollo de esta investigación fue la siguiente:

1. Conceptualizaciones
2. Cronología y metodología del desarrollo de costos presupuestos y programación de obra
3. Casos de uso y diagrama de flujo del software
4. Diagrama de despliegue del software
5. Diagrama de paquetes del software

Desarrollo

Conceptualizaciones

Autoaprendizaje: Es el proceso autorregulado de una persona donde toma conciencia de sus propios procesos cognitivos, relacionando las variables cognitivas y emocionales al diseñar y aplicar una estrategia de aprendizaje (Cárcel Carrasco, 2016).

Análisis de precios unitarios (A.P.U) : Es un modelo matemático que adelanta el resultado, expresado en una moneda la situación relacionada a una actividad estudiada calculada en una unidad de medida dentro del concepto de obra (Benitez Varon & Camargo Pérez, 2020).

Calidad Educativa: Hace referencia a los procesos de formación en las instituciones de educación, que tiene como prioridad el bienestar de los estudiantes, con principios equitativos, relevancia pertinencia eficiencia y eficacia para cumplir con las políticas de educación (Marín-González et al., 2018).

Costos de la construcción: Es el valor expresado en una moneda que deriva de la complejidad de una propuesta de construcción, basado en sus especificaciones técnicas, la tecnología necesaria, la calidad del trabajo y sus garantías.

Costos directos: Hace referencia a los costos de los insumos necesarios para la realización de un proyecto. (Materiales, Maquinaria, Mano de Obra y Transporte).

Costos indirectos: Son aquellos costos que no tiene vínculo directo con la elaboración de un producto servicio, sin embargo, son consumidos durante el proceso. (Servicios, Arriendos, Empleados de planta).

Presupuesto: Es un estudio económico dirigido a cumplir una meta, se expresa en términos financieros y este se debe cumplir en un tiempo determinado (Jiménez López, 2017).

Software: Son los diferentes procesos, documentación, reglas y asociación de datos que forman parte de todas las operaciones lógicas dentro de un sistema computarizado (IEEE Std, 1993).

Software educativo: Son las herramientas diseñadas dentro de un programa de computador, que sirven para facilitar el proceso de enseñanza – aprendizaje mediante diferentes técnicas (Almaguel Guerra et al., 2016; Niño & fernández, 2019).

Software de calidad: Son las características y el desempeño de un sistema de información dentro del ciclo de vida de un programa computarizado (Callejas-Cuervo et al., 2017).

Tecnologías de la información y Comunicación: Las TIC constituyen a las herramientas que permiten potencializar cualquier tipo de procesos mediante el variado uso de recursos y conocimientos (Pizarro & Cordero, 2013).

Cronología y metodología del desarrollo de costos presupuestos y programación de obra

La cronología implementada dentro del programa de Ingeniería Civil, para la materia de Costos Presupuestos y Programación de obra, parte de la siguiente manera:

1. El proyecto parte de un pliego de condiciones y se establece una estructura de división del trabajo (E.D.T).
2. Se calculan las cantidades que serán necesarias para cada actividad.
3. Se establece una lista de materiales
4. Se establece una lista de maquinarias
5. Se establece una lista de rendimientos de mano de obra
6. Se realizan los A.P.U.
7. Se calculan el valor directo del proyecto
8. Se establecen los valores de los costos indirectos
9. Se calcula el valor total del proyecto
10. Se realiza la programación de obra.

La metodología para el desarrollo de este procedimiento generalmente es el siguiente:

1. Casi siempre existe ya una E.D.T establecida inicialmente, la cual está ligada a todos los aspectos técnicos, sin embargo, aunque exista debe revisarse que estén contemplados todos los aspectos del

proyecto, identificando cuales son las actividades que deben realizarse para poder establecer un presupuesto y una programación lo más detallado posibles.

2. Posteriormente de haber creado la E.D.T con las actividades que se ejecutaran, se procede a realizar la medición de cada uno de los ítems dentro del presupuesto, de manera automática o manual según sea el caso, para tener las mediciones lo más exactas posibles con respecto a lo que se ejecutará dentro del proyecto.
- A partir de aquí, el desarrollo de un presupuesto puede desarrollarse de diferentes maneras, sin embargo, para que exista un procedimiento lógico se propone que no existen listas de precios, de maquinaria, mano de obra.
3. En este punto, es indispensable tener conocimientos teóricos o prácticos acerca de los procesos constructivos debido a que se debe tener en cuenta que tipo de materiales, maquinaria o equipos son los más idóneos para realizar las diferentes tareas.
 4. Los materiales deben establecerse dentro de una lista, basados en la lista de actividades que se estableció dentro de la E.D.T., este será un proceso tedioso la primera vez, pero después se podrán utilizar estos ítems para otros proyectos.
 5. Este proceso debe realizarse para la maquinaria, los equipos, la mano de obra y el transporte.
 6. Posteriormente a tener estas bases de datos, se deben crear los diferentes A.P.U. considerando los diferentes elementos que requiere una actividad para ejecutarse de manera correcta, por lo tanto, la persona que realice esta tarea debe tener un poco de experiencia en este aspecto.
 7. Al realizar este procedimiento para cada una de las actividades, solo hace falta multiplicar el valor del A.P.U por la cantidad de unidades que tiene dicha actividad, de esta manera, se sumarian todos los ítems del presupuesto y se establecería el costo directo del proyecto.
 8. Posteriormente se debe determinar el costo indirecto del proyecto, el cual puede llegar a ser hasta de un 30% del costo total, sin embargo, cada uno de sus componentes deberá justificarse.
 9. El valor total del presupuesto se calcula sumando los costos directos e indirectos calculados anteriormente.
 10. El último paso es establecer la programación de obra, la cual se estimará según el tipo de programación realizada, en el programa de Ingeniería Civil, se utiliza el método de la ruta crítica y es la metodología que se implementará dentro de este software.

Casos de uso y diagrama de secuencia del software

Los casos de uso son los que dan el orden y la importancia a las acciones que requiere el software, este indica como debe ser construido el sistema, convirtiendo en pequeños conjuntos de múltiples mini proyectos, que permiten cumplir los objetivos del software de manera más detallada.

Esta arquitectura de software propone 8 casos de uso:

1. Registro e inicio de sesión
2. Gestión de presupuestos
3. Gestión de caracterizaciones
4. Gestión de ítems
5. Gestión de salarios
6. Gestión de A.P.U
7. Listo de compras
8. Gestión de la programación

Adicional a esto, para una mejor interpretación de este procedimiento, mediante un diagrama de secuencia se explicará el proceso de manera visual y secuencialmente para evitar errores y corregir los que se lleguen a cometer. Esto permitirá que los problemas se resuelvan de manera más fácil y se puedan encontrar soluciones rápidas (González Espinosa, 2019).

Diagrama de despliegue del software

El diagrama de despliegue mediante el lenguaje técnico unificado (U.M.L) es una vista física que representa los componentes físicos del sistema de información del software. En este diagrama se representan los componentes de hardware denominados dispositivos en los que se va a realizar la implementación de los componentes que vienen del desarrollo del software denominados artefactos y los protocolos o tipos de comunicación que se realizan entre estos para el correcto funcionamiento del sistema de información.

Diagrama de paquetes del software

El diagrama de paquete de U.M.L representará como vista física la jerarquía de directorios del sistema, este nos va a proporcionar una visualización de los paquetes que componen del sistema de información y sus dependencias. El sistema seguirá el patrón de diseño Modelo – Vista – Controlador (MVC) que se describirá más adelante. En este caso los paquetes representan los directorios actuales del sistema y un conjunto de agrupaciones que se especificaran más adelante.

Resultados

Los diferentes casos de uso se pueden observar en la Figura 1, donde se resaltan los que hacen parte de las funciones del super usuario y del usuario.

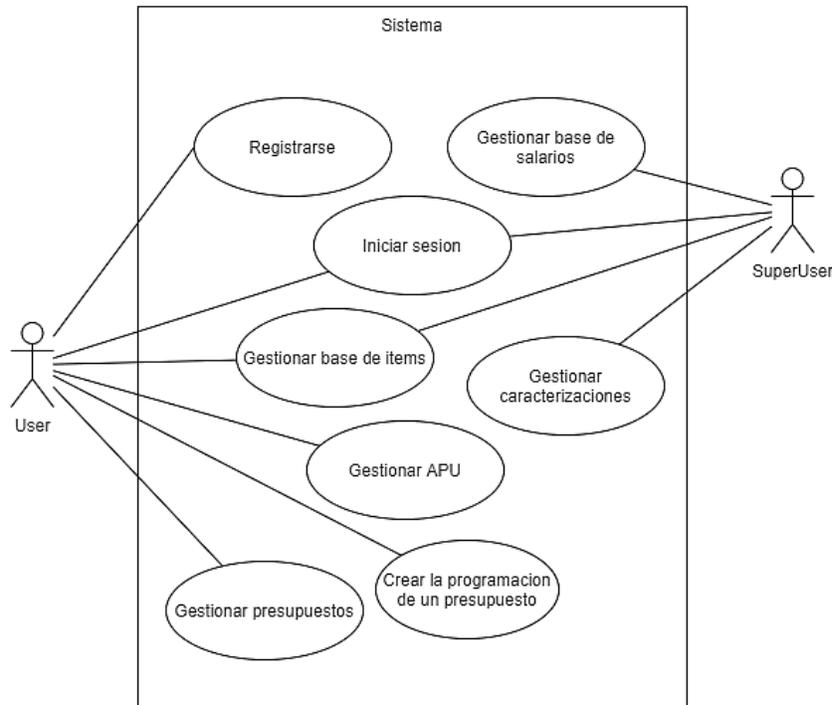


Figura 1. Casos de uso Fuente: Autores.

Casos de uso

- Registro e inicio de sesión: En este caso de uso, se realizará el procedimiento de registrarse mediante las credenciales establecidas, posteriormente el usuario podrá iniciar sesión y poder acceder a todos los servicios del sistema.
- Gestión de presupuestos: Este caso de uso se encarga de la gestión de los diferentes presupuestos que el usuario puede crear, editar o eliminar, en este caso de uso se pueden agregar, editar o eliminar, capítulos, actividades y calcular los valores monetarios de un proyecto
- Gestión de caracterizaciones: Las caracterizaciones serán uno de los aspectos más importantes de esta herramienta debido a que dividirá todos los elementos del sistema en grupos de procesos y grupos de actividades, todo esto con el fin de que la creación de los A.P.U, sea más eficaz mediante el uso de filtros.
- Gestión de ítems: Este caso de uso se encargará de agregar, editar o eliminar, los diferentes ítems del presupuesto, tales como, materiales, mano de obra, maquinaria y transporte.
- Gestión de salarios: En este caso de uso, el usuario podrá agregar los salarios que necesite para su personal, sin embargo, solo el super usuario podrá actualizar los precios del salario mínimo legal vigente, así como sus diferentes condicionales para el cálculo de las prestaciones sociales.

- Gestión de A.P.U: Dentro de este caso de uso, se podrán identificar los ítems que se utilizarán dentro de cada actividad y gracias a la caracterización de estos, su realización será mucho más rápida.
- Listado de compras: Este caso de uso, permitirá al usuario ver los diferentes materiales y sus cantidades para saber la cantidad de material que debe comprar, así como las maquinarias a alquilar y la mano de obra que necesita a lo largo de todo el proyecto.
- Gestión de la programación: Aquí se realizará toda la programación del proyecto mediante el método de la ruta crítica, adicional a esto, el software permitirá vincular los trabajadores a las diferentes actividades e identificará quienes se encuentran indispuestos. Será una programación dinámica y se podrá actualizar en el desarrollo del proyecto.

Diagrama de secuencia

Este diagrama de secuencia, que se muestra en la Figura 2, enseña de manera detallada el proceso general para la creación de un presupuesto y su respectiva programación, así como para la creación, edición o eliminación de los A.P.U.

De tal manera que, inicialmente el usuario debe registrarse e iniciado sesión, le permite seleccionar los presupuestos guardados, para editar o crear uno nuevo. El usuario puede agregar, capítulos y así mismo crear actividades, una vez creada las actividades se le permite

al usuario editar, crear o eliminar los A.P.U. seleccionando los diferentes ítems que a este lo conforman. Todo esto, con su respectiva caracterización de grupos de procesos y grupos de actividades, esto permitirá una mayor eficacia y productividad al momento

de desarrollar el presupuesto. Una vez creado todo el presupuesto, el usuario tendrá la opción de realizar su programación de obra, la cual relacionará a las personas que llevarán a cabo cada una de las actividades.

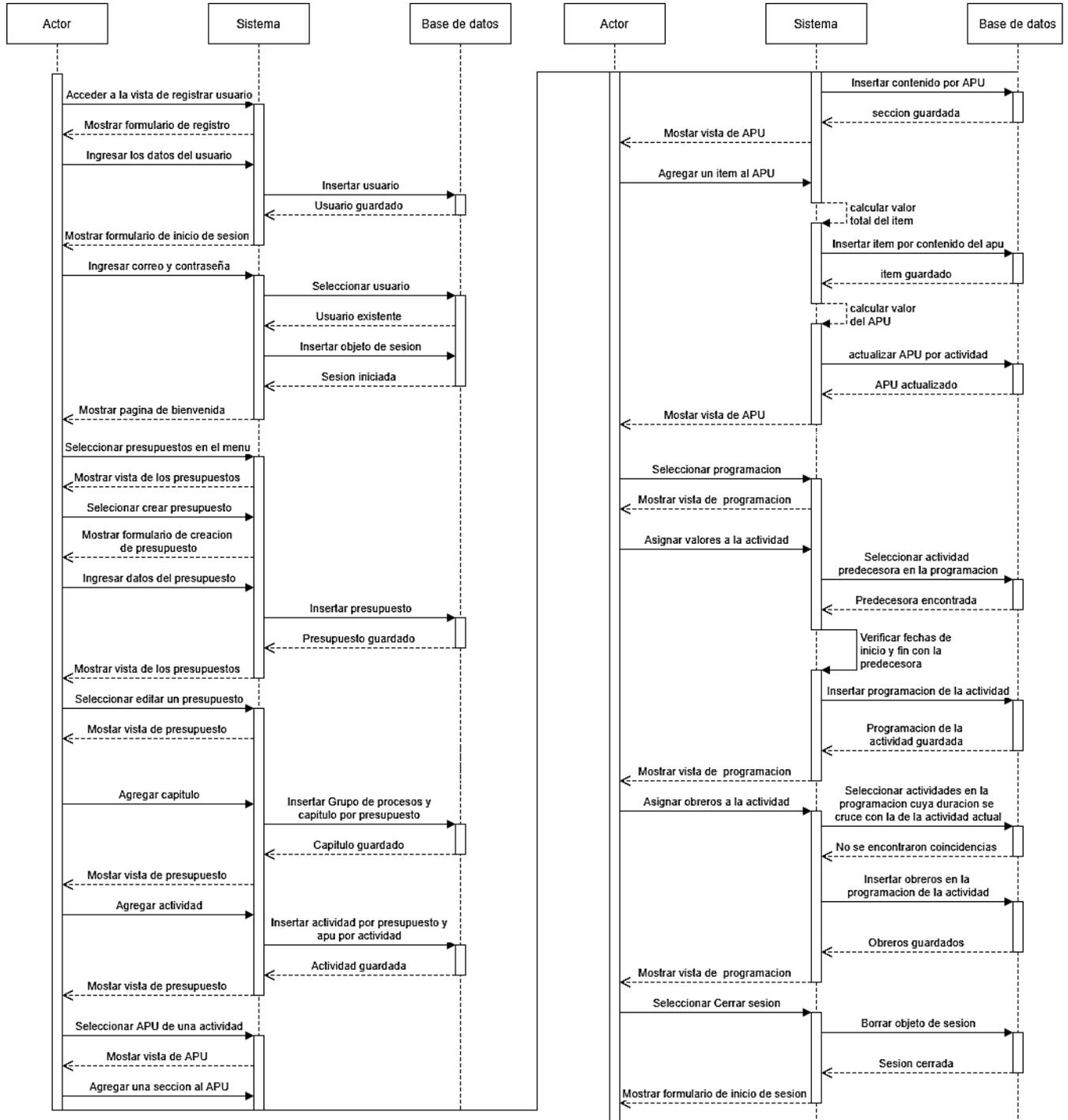


Figura 2: Diagrama de secuencia. Fuente: Autores

Diagrama de despliegue del software

En la figura 3, se puede observar el diagrama de despliegue del software. Donde el sistema está realizado bajo la arquitectura cliente servidor e implementa el patrón de diseño de Modelo Vista Controlador; el

funcionamiento de este modelo de arquitectura se basa en que un servidor en este caso el dispositivo denominado “servidor de aplicaciones – nginx”, contiene los archivos del sistema y los sirve por igual según la disponibilidad del servidor a un número indeterminado de usuarios denominados clientes.

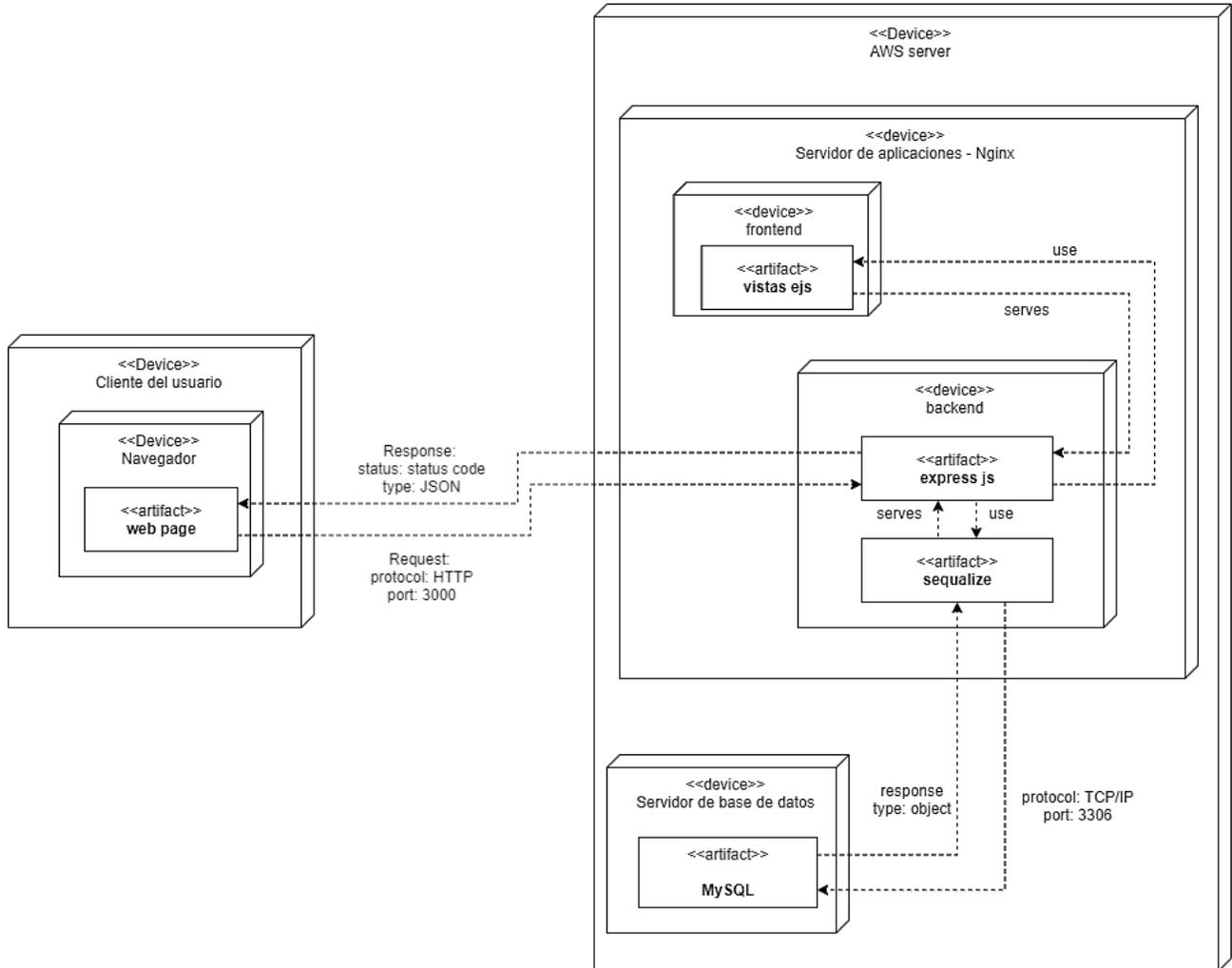


Figura 3: Diagrama de despliegue del software. Fuente: Autores.

Diagrama de paquetes del software

En la figura 4, se puede observar el diagrama de paquetes, el cual representa la jerarquía de los directorios de

paquetes UML, el cual proporciona una visualización de los diferentes paquetes que componen del sistema de información y sus dependencias. Este sistema sigue el patrón de diseño Modelo – Vista -Controlador.

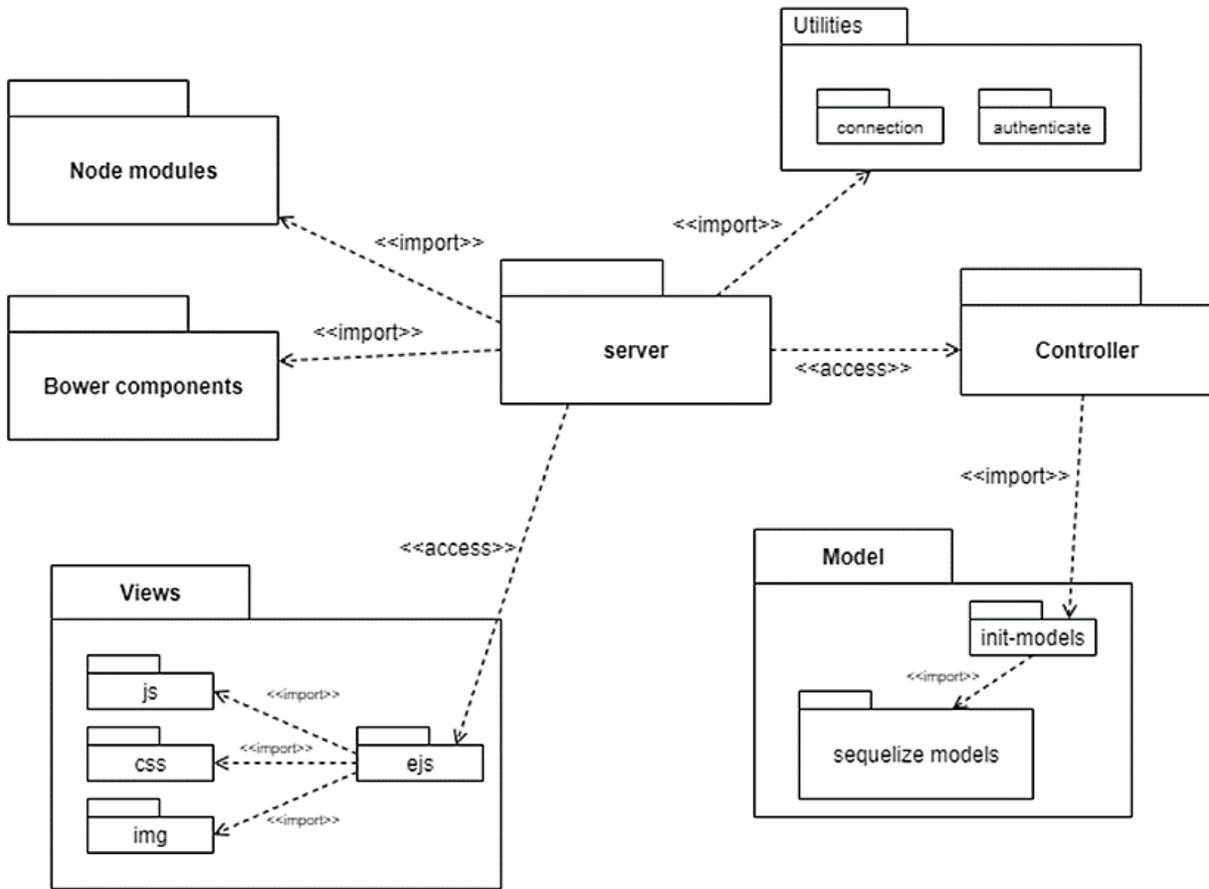


Figura 4: Diagrama de Paquetes. Fuente: Autores.

- El paquete server representa es una agrupación de un conjunto de archivos de configuración que se encuentran en la raíz del sistema la cual consiste de los siguientes archivos: el archivo de inicialización del servidor y los archivos que referencian las versiones de las dependencias de los paquetes de Node modules y Bower components
- El paquete Node modules y Bower components representan las librerías externas y sus versiones de las que depende el sistema, posee una relación de importación con el paquete server ya que usa estas librerías para funcionar correctamente.
- El paquete Directorio que contiene el archivo de la conexión con la base de datos, la autenticación de usuarios del sistema tiene una relación de importación con el paquete server ya que este realiza la invocación de los métodos y objetos dentro de estos archivos para la funcionalidad de algunos métodos.
- El paquete views se refiere al directorio que guarda los objetos referentes al front-end y sus componentes, es decir aquello que será accesible en las terminales del cliente una vez renderizadas las plantillas desde el servidor. Posee una relación de acceso con el paquete

server ya que este realiza invocaciones a las plantillas dentro del sub-paquete ejs.

- El paquete controller (en conjunto con el paquete modelo) se refiere al directorio donde almacenan los objetos referentes al Back-end, en este se almacenan las funciones que se realizarán antes de procesar los datos a la base de datos, tiene una relación de acceso con el paquete server porque este realiza el enrutamiento hacia este cuando se es invocada alguna función desde el front-end y tiene una relación de importación con el paquete modelo porque a través del sub paquete init-models define e invoca los modelos con los que almacenara los objetos en la base de datos.
- El paquete modelo se refiere al directorio donde se almacenan la definición de los objetos que se encuentran en la base de datos, es decir las tablas, sus relaciones y sus atributos.

Conclusiones

Se realizó exitosamente esta arquitectura de software académico con base a las necesidades y procedimiento implementados dentro del programa de Ingeniería Civil.

Los casos de usos utilizados son consecuentes a las unidades teóricas vistas dentro del contenido curricular del curso, estos casos de uso permitieron representar de manera más gráfica la interacción de todo el sistema en general y los servicios que esta herramienta prestará.

El diagrama de despliegue y el diagrama de paquetes permitirán una correcta distribución de la información del sistema de manera jerárquica, con el fin de que este trabaje en las mejores condiciones, bajo parámetros que permitan su alta productividad y velocidad ante un número indeterminado de usuarios simultáneos.

Esta propuesta, permitirá un desarrollo idóneo y eficaz de este software académico, el cual es indispensable debido a que dentro del programa de Ingeniería Civil, existe escases de clases dinámicas con manejo de recursos tecnológicos, por lo tanto, implementar esta herramienta marcará un antes y un después dentro de la pedagogía impartida en las clases, actualizando su contenido programático, generando mayor interés en las clases, manteniendo la vanguardia ante el desarrollo de la tecnología y formando profesionales más idóneos ante los retos y las exigencias de la sociedad actual.

Así mismo, esta propuesta permitirá a los estudiantes del programa de Ingeniería Civil de la Universidad Francisco de Paula Santander, manejar una herramienta tecnológica propia y gratuita que diferencie de otras herramientas mucho más costosas, de difícil acceso y comprensión, contará con lo que denominamos “caracterizaciones”, donde cada uno de los ítems (materiales, mano de obra, maquinaria y transporte) así como sus capítulos y actividades estarán relacionadas a grupos de procesos y grupos de actividades. Esta agrupación de ítems aumentará la productividad y generará un entorno intuitivo y de poca dificultad que permitirá la rápida aprehensión de los procedimientos que conllevan a desarrollar cualquier presupuesto de obra.

Referencias

- Almaguel Guerra, A., Alvarez Mora, D., Pernía Nieves, L. A., Mota Pimentel, G. J., & Coello León, C. (2016). Software educativo para el trabajo con matrices. *Revista Digital: Matemática, Educación e Internet*, 16(2), 1–12. <https://doi.org/10.18845/rdmei.v16i2.2525>
- Alonso Rodriguez, A. (2019). *Software Educativo para la asignatura Introducción a la Ingeniería Civil* [Universidad Central “Marta Abreu” de las Villas]. <https://dspace.uclv.edu.cu/handle/123456789/11672>
- Alvarado Cardenas, M. E., & Finol de Govea, A. (2019). Diseño de un software para la enseñanza de Inglés Técnico en Ingeniería: enfoque teórico de su elaboración. *Revista UCMAule*, 57, 39–62.

<https://doi.org/10.29035/ucmaule.57.39>

- Báez-Pérez, A. A., Soto-Vergel, Á. J., & Herrera-Rubio, J. E. (2019). Enseñanza de sistemas de radiocomunicaciones terrestres con línea de vista mediante software educativo. *Revista Educación En Ingeniería*, 14(28), 78–87. <https://doi.org/10.26507/rei.v14n28.996>
- Benitez Varon, C. M., & Camargo Pérez, J. F. (2020). *Tipificación y análisis de precios unitarios de estructuras para la aducción de agua en vivienda de suelo rural para el Departamento de Cundinamarca* (pp. 1–58). Universidad Distrital Francisco José de Caldas. Facultad de Ingeniería. <https://repository.udistrital.edu.co/bitstream/handle/11349/25575/CamargoPerezJairoFelipe2020.pdf?sequence=1&isAllowed=y>
- Burgos Moreira, B. de los A., & Velasco García, L. E. (2018). *Los recursos informáticos en la calidad del aprendizaje receptivo: Diseño de un entorno Web educativo* [Universidad de Guayaquil. Facultad de Filosofía, letras y ciencias de la educación]. <http://repositorio.ug.edu.ec/bitstream/redug/35929/1/BFILO-PD-INF1-18-010.pdf>
- Callejas-Cuervo, M., Alarcón-Aldana, A. C., & Álvarez-Carreño, A. M. (2017). Modelos de calidad del software, un estado del arte. *Entramado*, 13(1), 236–250. <https://doi.org/10.18041/entramado.2017v13n1.25125>
- Cárcel Carrasco, F. J. (2016). Desarrollo de habilidades mediante el aprendizaje autónomo. *3C Empresa, Investigación y Pensamiento Crítico*, 5(3), 52–60. <https://doi.org/10.17993/3cemp.2016.050327.52-60>
- Cova, Á., Arrieta, X., & Aular de Durán, J. (2008). Revisión de modelos para evaluación de software educativos. *Télématique. Revista Electrónica de Estudios Telemáticos*, 7(1), 94–116. <https://www.redalyc.org/articulo.oa?id=78470106>
- Díaz, J. C., Ucán, J. P., Aguilera, A., & Toscano, A. B. (2016). Asistente escolar para los estudiantes de Ingeniería de Software: una aplicación móvil. *ReCIBE. Revista Electrónica de Computación, Informática, Biomédica y Electrónica*, 5(3), 1–24. <http://40.71.171.92/bitstream/handle/123456789/1020/71-137-1-SM.pdf?sequence=1&isAllowed=y>
- González Espinosa. Jenniffer Xiomara. (2019). *Diagrama de flujo y su relación con la vida cotidiana* (pp. 1–25). Universidad Técnica de Machala. Facultad de Ciencias Empresariales. http://repositorio.utmachala.edu.ec/bitstream/4800/0/14847/1/E-4389_GONZALEZ ESPINOSA

JENNIFFER XIOMARA.pdf

- Hoyos Salcedo, E. A., Acosta Minoli, C. A., Arisitzabal Zapata, J. H., Mesa Mazo, M. J., Trujillo Salazar, C. A., Rincón Penagos, J. A., Gutiérrez Rodríguez, Á., & Jaime Pastor, A. (2021). Influencia de un software educativo en la consolidación del aprendizaje de superficies cuádricas. *Tecné, Episteme y Didaxis: TED*, 49, 123–142. <https://doi.org/10.17227/ted.num49-9574>
- IEEE Std. (1993). *IEEE Software Engineering Standard. Glossary of Software Engineering Terminology, 729-1993*. IEEE Press.
- Jiménez López, L. (2017). *Presupuestos en la construcción* (1 ed). Ediciones Paraninfo S.A. <https://www.paraninfo.co/catalogo/9788428338035/presupuestos-en-la-construccion>
- Marín-González, F., Cabas, L. de J., Cabas, L. C., & Paredes-Chacín, A. J. (2018). Formación Integral en Profesionales de la Ingeniería. Análisis en el Plano de la Calidad Educativa. *Formación Universitaria*, 11(1), 13–24. <https://doi.org/10.4067/s0718-50062018000100013>
- Niño, J. A., & fernández, F. H. (2019). Una mirada a la enseñanza de conceptos científicos y tecnológicos a través del material didáctico utilizado. *Revista Espacios*, 40(15), 4. <http://www.revistaespacios.com/a19v40n15/a19v40n15p04.pdf>
- Pizarro, G., & Cordero, D. (2013). Las TIC: Una herramienta tecnológica para el desarrollo de las competencias lingüísticas en estudiantes universitarios de una segunda lengua. *Revista Electrónica Educare*, 17(3), 277–292. https://www.scielo.sa.cr/scielo.php?script=sci_artext&pid=S1409-42582013000300013
- Rodríguez Calderón, W., Pallares Muñoz, M. R., & Vargas Guerra, A. (2019). Experiencia educativa apoyada en TICs para el cálculo de centroides y momentos de inercia de figuras planas de contorno recto. *III Congreso Internacional En Inteligencia Ambiental, Ingeniería de Software y Salud Electrónica y Móvil AmITIC 2019*, 62–66. <https://core.ac.uk/download/pdf/234021242.pdf>
- Salazar Alzate, M. F. (2017). *Impacto económico del uso de BIM en el desarrollo de proyectos de construcción en la ciudad de Manizales*. Universidad Nacional de Colombia.
- Silva Giraldo, C. A., Dugarte Mendoza, J. S., & Majía Jálabe, A. (2018). Impacto de los costos de calidad en la ejecución de los proyectos de construcción en Colombia. *Revista EAN. Escuela de Administración de Negocios. Edición Especial*, 33–54.
- Taipe, C., Yancachajlla, U., & Flores, H. (2020). Aprendizaje de la dinámica de una partícula a través del software Interactive Physics en estudiantes de ingeniería. *Revista Innova Educación*, 2(2), 330–346. <https://doi.org/10.35622/j.rie.2020.02.007>
- Vinueza, M., & Córdova, J. (2019). El uso de los programas educativos en el proceso de enseñanza-aprendizaje en ingeniería del software: Una aproximación al estado del arte. *Revista Inclusiones*, 6(2), 155–177. <http://revistainclusiones.org/index.php/inclu/articloe/view/63>