

DOI: <https://doi.org/10.56712/latam.v4i1.560>

Desarrollo de software para la enseñanza de la Ingeniería: una contribución de la red ALFA para la educación tecnológica

Software development for the Engineering Teaching: a contribution of the ALFA network for technological education

Helga Karina Tolano Gutiérrez

Ktolano@uts.edu.mx
Universidad Tecnológica del Sur de Sonora
Ciudad Obregón, Sonora – México

María Anabell Covarrubias Díaz Couder

Anabell.covarrubias@lasallenoroeste.edu.mx
Universidad La Salle Noroeste
Ciudad Obregón, Sonora - México

Eusebio Jiménez López

ejimenezl@msn.com
Universidad Tecnológica del Sur de Sonora
Ciudad Obregón, Sonora - México

Lilia Zulema Gaytán Martínez

Zgaytan@uts.edu.mx
Universidad Tecnológica del Sur de Sonora
Ciudad Obregón, Sonora - México

Erika Ercilia Vásquez Moreno

Evazquez@uts.edu.mx
Universidad Tecnológica del Sur de Sonora
Ciudad Obregón, Sonora - México

Laura Olivia Amavizca Valdez

Lamavizca@uts.edu.mx
Universidad Tecnológica del Sur de Sonora
Ciudad Obregón, Sonora - México

Artículo recibido: 17 de marzo de 2022. Aceptado para publicación: 11 de abril de 2023.
Conflictos de Interés: Ninguno que declarar.

Resumen

El desarrollo de investigación teórica y aplicada es una actividad que se realiza en las universidades de todo el mundo. Sin embargo, cuando las universidades se integran en organizaciones como las redes de colaboración, el potencial para el desarrollo de ciencia, tecnología y educación se multiplica. En este artículo se presenta una experiencia en desarrollo de software educativo diseñado por las universidades que conforman la Red ALFA localizada en Cd. Obregón, Sonora, México. Fue aplicado el método evolutivo incremental para generar un proyecto de software relacionado con el simulador de un robot. El objetivo de este proyecto fue mostrar a los alumnos de ingeniería que no son de las carreras de Tecnologías de la Información y Comunicación la importancia de generar un producto de software que va más allá de la programación típica que se realiza para resolver un proyecto educativo en ingeniería y que dicho

producto puede ser compartido con otras universidades para mejorar la educación en ingeniería y tecnología.

Palabras clave: ingeniería de software, métodos evolutivos, educación en ingeniería

Abstract

The development of theoretical and applied research is an activity carried out in universities around the world. However, when universities are integrated into organizations such as collaborative networks, the potential for the development of science, technology and education is multiplied. This article presents an experience in the development of educational software designed by the universities that make up the ALFA Network located in Cd. Obregón, Sonora, Mexico. The incremental evolutionary method was applied to generate a software project related to a robot simulator. The objective of this project was to show engineering students who are not from the Information and Communication Technology careers the importance of generating a software product that goes beyond the typical programming that is carried out to solve an educational project in engineering and that said product can be shared with other universities to improve education in engineering and technology.

Keywords: software engineering, evolutionary methods, engineering education

Todo el contenido de LATAM Revista Latinoamericana de Ciencias Sociales y Humanidades, publicados en este sitio está disponibles bajo Licencia Creative Commons . 

Como citar: Tolano Gutiérrez, H. K., Covarrubias Díaz Couder, M. A., Jiménez López, E., Gaytán Martínez, L. Z., Vásquez Moreno, E. E., & Amavizca Valdez, L. O. (2023). Desarrollo de software para la enseñanza de la Ingeniería: una contribución de la red ALFA para la educación tecnológica. *LATAM Revista Latinoamericana de Ciencias Sociales y Humanidades* 4(1), 4219–4237. <https://doi.org/10.56712/latam.v4i1.560>

INTRODUCCIÓN

En el mundo actual, caracterizado por una sociedad compleja, globalizada, altamente diferenciada y especializada, las necesidades y problemáticas se han multiplicado, y para solucionarlas se requiere de la aplicación de conocimiento y de la colaboración entre diversos organismos como el gobierno, las universidades y las empresas. El modelo de redes de colaboración o cooperación, es ideal para sistematizar las relaciones entre varios organismos. Para el caso de las relaciones entre universidades, las redes de colaboración representan un buen mecanismo de colaboración ya que hacen más eficiente las actividades de investigación y la academia, y permite cumplir objetivos de estudio que serían difíciles de obtener si se trabaja de forma individual (Cárdenas, 2017).

La unión de grupos de trabajo interdisciplinarios e interinstitucionales facilitan la generación de conocimiento y la compartición de infraestructura es fundamental para la calidad académica y la producción científica. De acuerdo con Villanueva, Fernández y Palomares (2014), la experiencia y el tamaño de las redes personales facilitan la cantidad y calidad de la producción académica. Las redes organizativas sólo influyen en la cantidad de producción científica, las instituciones tienen capacidades, medios y activos, que pueden contribuir y aportar recursos que repercuten en la generación de conocimiento científico.

En términos generales, la cooperación hace referencia al conjunto de actividades realizadas entre o por instituciones (se pueden incluir actores diversos sociales) a través de múltiples modalidades y sobre la base del beneficio mutuo (Osorio, 2019), en tanto que las redes de cooperación se pueden definir como asociaciones de interesados que tienen como objetivo la consecución de resultados acordados conjuntamente a través de la participación y la colaboración mutua (Sebastián, 2000).

Las redes de colaboración tienen un alto impacto en la educación en ingeniería, pues los alumnos, profesores e investigadores participan en proyectos derivados de las empresas y centros de investigación, lo cual tiene beneficios importantes, pues se adquiere experiencia, recursos económicos y se proyecta a las universidades hacia la aplicación del conocimiento a problemáticas reales (Jiménez, Martínez y Luna, 2013). En este sentido en el año 2005 se formó una red de colaboración interinstitucional denominada "Red ALFA" en el sur de Sonora, México, integrada por tres instituciones de educación superior y una empresa (Jiménez, Ochoa y Martínez, 2004). La Red está integrada por la Universidad La Salle Noroeste, el Instituto Tecnológico Superior de Cajeme, la Universidad Tecnológica del Sur de Sonora y la Empresa Impulsora de Desarrollo Dinámico S.A. de C.V (Ahora Innovación en Ingeniería de Manufactura y Mantenimiento S de RL MI), todas estas organizaciones están ubicadas en Cd. Obregón Sonora (Martínez, Jiménez, Beltrán y Armenta, 2010). Esta red ha contribuido a resolver problemas con empresas Pymes (Martínez et al. 2012: Amavizca et al, 2014) y a desarrollar diversos proyectos académicos (Jiménez et al., 2014).

Con el propósito de mostrar las experiencias de la colaboración entre universidades de la Red Alfa, en este artículo se presenta una síntesis de un proyecto relacionado con el desarrollo de software aplicado a la enseñanza de la ingeniería, en particular la construcción de un simulador robótico desarrollado con un método de Ingeniería de Software (Jiménez et al, 2019). El objetivo principal del proyecto fue motivar a los alumnos de ingeniería de la Universidad Tecnológica del Sur de Sonora y de la Universidad La Salle Noroeste, ambas pertenecientes a la Red Alfa, a que conocieran las ventajas de desarrollar simuladores robóticos en términos de la Ingeniería de Software (Montilva y Barrios, 2021), en particular la aplicación del Método Evolutivo Incremental (Pressman, 2002) para que aprovecharan al máximo el desarrollo de los programas que semestre a semestre realizan para sus materias, proyectos y tesis. Cabe señalar que el desarrollo técnico del simulador fue presentado por Jiménez et al. (2019) en un congreso de Mecatrónica y en este

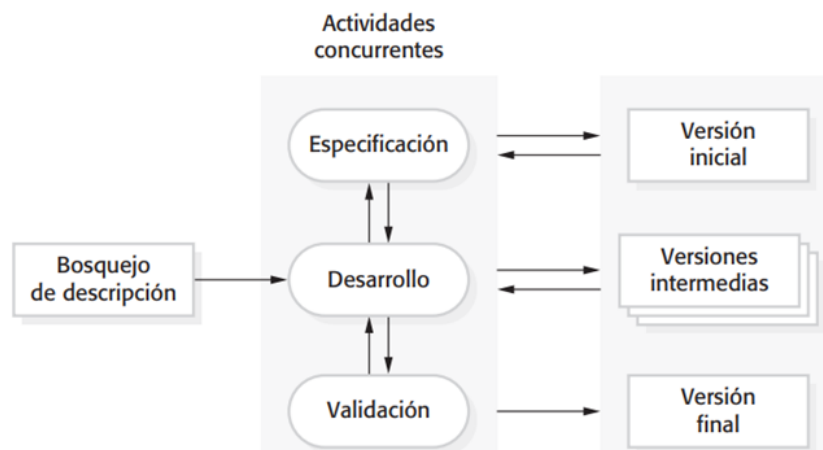
artículo se presenta como caso de éxito de la Red Alfa y la evaluación por parte de los alumnos acerca del uso de la Ingeniería de Software.

MÉTODO

La investigación presentada en este artículo es tipo descriptiva con enfoque cualitativo y el diseño de la misma es no experimental de corte transversal. La investigación se consideró descriptiva debido a que se mostrará una experiencia de desarrollo de software educativo y se realizarán valoraciones sobre las ventajas que se obtienen al pasar de un desarrollo de un software personal a uno desarrollado con Ingeniería de Software. El método de ingeniería de Software que se aplicó para el desarrollo del simulador se muestra en la Figura 1.

Figura 1

Desarrollo incremental



Fuente: Sommerville, 2011.

Para la selección del método de Ingeniería de Software utilizado para el desarrollo del simulador se consideraron los aspectos siguientes:

- No fue necesario seleccionar un método de ingeniería de software novedoso, ya que lo que importaba era que los alumnos de ingeniería aprendieran más la conceptualización que el propio desarrollo.
- El método evolutivo incremental fue seleccionado debido a que es más intuitivo, a pesar de que en la actualidad se usan más otros métodos, como por ejemplo las metodologías ágiles (Navarro, Fernández y Morales, 2013).

Para el desarrollo del simulador robótico se procedió de la siguiente manera:

- El grupo de noveno semestre se dividió en tres equipos de tres alumnos.
- A cada grupo se le entregó el programa y los modelos que habían desarrollado en una clase anterior.
- Se les dio material y pláticas acerca de Ingeniería de Software y los pasos del método evolutivo incremental.
- Se dejó a su consideración el número de incrementos y el método de diseño del producto de software.
- El software en donde se realizaría la base del desarrollo podía ser MATLAB o Mathematica.

Cabe señalar que los alumnos de ingeniería a los cuales se pretendía motivar eran aquellos que no estudian Tecnologías de la Información y la Comunicación, debido a que estos alumnos llevan materias de Ingeniería de Software en su carrera.

Instrumento

Para poder conocer las opiniones de los alumnos de ingeniería (que no son de las carreras de Tecnologías de la información y la Comunicación (TICs)) se diseñó el instrumento mostrado en la Tabla 1.

Tabla 1

Cuestionario aplicado a los alumnos

Preguntas	Respuestas
1. ¿Habías escuchado con anterioridad lo que es un producto de software?	SI ____ NO ____
2. ¿Tenías noción acerca del concepto de Ingeniería de Software?	SI ____ NO ____
3. ¿Conocías algún método de ingeniería de software?	SI ____ NO ____
4. ¿Se te complicó entender el método evolutivo incremental?	SI ____ NO ____
5. ¿Consideras necesario que en tu plan de estudios se integre una materia de ingeniería de software?	SI ____ NO ____
6. ¿Consideras importante transformar tu software personal a un producto de software?	SI ____ NO ____
7. ¿Pudiste notar algunas diferencias entre tu programa o producto personal de software del simulador y entre el simulador desarrollado con el método de las entregas incrementales?	SI ____ NO ____
8. ¿Usarías en el futuro un método de ingeniería de software para transformar tus programas a un producto de software?	SI ____ NO ____
9. ¿Consideras complicado usar un método de ingeniería de software para el desarrollo de simuladores robóticos?	SI ____ NO ____
10. ¿Recomendarías a otros alumnos de ingeniería usar un método de ingeniería de software para transformar sus programas en producto de software?	SI ____ NO ____

Fuente: Elaboración propia.

RESULTADOS

La propuesta de utilizar métodos de Ingeniería de Software nació por el hecho de que en los proyectos que desarrollan los alumnos de ingeniería de las universidades que pertenecen a la Red ALFA, sobre todo en aquellos en donde se dejan la aplicación de programación (y que no son de las carreras de computación), solo diseñan software al nivel más bajo, es decir, la mayoría sólo construían los programas o software personal para resolver un caso particular y que solo es entendible por el desarrollador, por lo que dicho software no se aprovecha. Para que los alumnos de ingeniería puedan transformar su programa en un producto de software (producto que tiene un ciclo de vida, es generado por una gestión de software, recibe mantenimiento y tiene características visibles para los clientes) deben hacer uso de la Ingeniería de Software (IS) ya que esta busca apoyar el desarrollo de software profesional, en lugar de la programación individual, e incluye técnicas que apoyan la especificación, el diseño y la evolución del programa, ninguno de los cuales son normalmente relevantes para el desarrollo de software personal (Sommerville, 2011).

En esta sección se presentan los resultados: primero se describen algunos ejemplos de la manera en que los alumnos desarrollan sus programas y segundo se presentan los resultados del simulador desarrollado con el método evolutivo incremental (Jiménez et al. 2019).

Programación individual o desarrollo de software personal

Las tareas o proyectos que se desarrollan en diversas clases de ingeniería, en particular en Ingeniería Mecatrónica, requieren realizar programas de cómputo, en donde se realicen los cálculos necesarios para alguna aplicación y se muestran los resultados numéricos y gráficos. El camino que sigue el desarrollo de software personal para una aplicación en robótica es el siguiente:

- Estudio de los movimientos del robot.
- Modelación matemática de los movimientos de un robot.
- Modelación numérica de la solución de las ecuaciones resultantes.
- Generación de un prototipo virtual en un software CAD (Computer Aided Design) para obtener datos geométricos para alimentar los programas.
- Selección del lenguaje de programación o de la plataforma de cálculo simbólico.
- Programación de los modelos.
- Visualización de los resultados numéricos y gráficos.
- Prueba de los resultados.

Los pasos descritos anteriormente se aplican en general para el desarrollo de simuladores y para generarlos, los alumnos se conforman en equipos de trabajo, debido a que los proyectos no son fáciles y sencillos. Se ha observado que los desarrollos de los programas tienen las siguientes características:

- Los alumnos desarrollan la programación según lo enseñado por sus profesores anteriores (de programación) o bien usando una lógica personal con la que al final obtienen el resultado funcional que se busca, es decir, un programa funcional.
- Se ha observado que, al no usar una metodología formal, los alumnos tienen problemas para encontrar los errores en sus programas lo que implica que se tarden más en desarrollarlo.
- Algunos alumnos prefieren desarrollar sus programas en lenguajes como el "C++" en vez de utilizar lenguajes de cálculo formal como el software Mathematica o Matlab. Los programas desarrollados en C++ a menudo sólo él o los alumnos que programan lo entienden y en general no son transferibles.
- La codificación de los programas es confusa y es visible la falta de etiquetados.
- Cuando se requiere realizar un nuevo programa similar en general los alumnos vuelven a programar de cero. Los programas a menudo son poco reutilizables.

La Tabla 2 muestra la codificación en Mathematica 6.0 de un programa de un simulador robótico desarrollado por un alumno de la Universidad La Salle Noroeste y la Figura 2 muestra la calidad gráfica (Romero, 2010).

Tabla 2

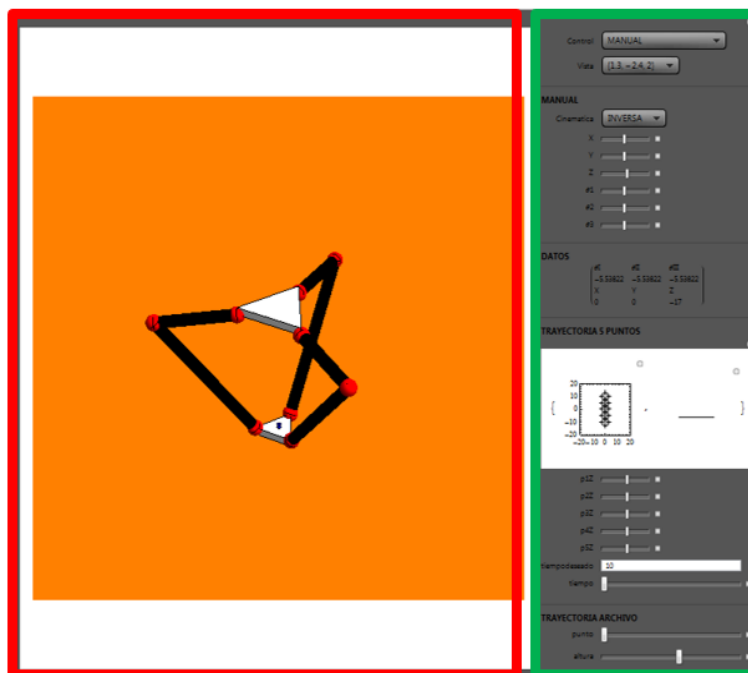
Código para simular un robot Delta en Mathematica 6.0

<pre> GdePuntos [Dibujo_] :=Module[{NumOrd,m,Δy,Δx, Tabla,equL,AngIni,AngFin,Rad,Incl,IncC,incx }, total={{Dibujo[[1,2]],Dibujo[[1,3]]}}; NumOrd=Dimensions[Dibujo][[1]]; Dimensiones=0; For[Ord=1,Ord<=NumOrd,Ord++, Which[Dibujo[[Ord,1]]=="1", {Δy=(Dibujo[[Ord,5]]-Dibujo[[Ord,3]]); Δx=(Dibujo[[Ord,4]]-Dibujo[[Ord,2]]); If[Δx≠0, {m=Δy/Δx; If[Δx>0,incx=0.005,incx=-0.005]; j=Table[{i,m Dibujo[[Ord,2]]+Dibujo[[Ord,3]],{i,Dibujo [[Ord,2]],Dibujo[[Ord,4]],incx}}}, {If[Δy>0,Incl=0.005,Incl=-0.005]; </pre>	<p>Generador de puntos para el modo TRAYECTORIA POR ARCHIVO, carga un archivo con extensión .txt y crea una tabla con las coordenadas (x, y) por las que pasa el dibujo creado por dicho archivo.</p>
---	---

Fuente: Romero, 2010.

Figura 2

Salida gráfica del simulador de un robot Delta en Mathematica 6.0



Fuente: Romero, 2010.

A pesar de que el programa es funcional y que cumple con las especificaciones mínimas se pudo observar que: 1) el programa a pesar de que tiene una explicación como se muestra en la Tabla 2 la codificación está confusa, 2) El alumno batalló mucho para hacer las correcciones, 3) El programa fue desarrollado con la lógica propia del alumno y 4) El programa se transfirió a otros alumnos para hacer simulaciones y su comprensión fue muy complicada y tardada.

Otro caso se presentó en el desarrollo de una aplicación para la construcción de funciones desarrollada por alumnos y profesores de la Universidad Tecnológica del Sur de Sonora (Jiménez et al. 2011). La Figura 3 muestra la salida gráfica de un programa que fue desarrollado para que los alumnos pudieran practicar el tema de las funciones matemáticas (Jiménez et al. 2014).

Figura 3

Salida gráfica de un software para construir funciones desarrollado en ASP.NET 2005

The screenshot shows a web application interface with a blue header containing the text "FUNCCIONES" and a registered trademark symbol. Below the header, there are three main sections:

- Datos de la Función:** This section contains four input fields: "Dominio:", "Conjunto del Dominio:", "ContraDominio:", and "Conjunto del ContraDominio:". A blue button labeled "Nueva Función" is positioned to the right of the "Conjunto del Dominio:" field.
- Construir Función:** This section features two dropdown menus for "Dominio:" and "ContraDominio:". To their right are three blue buttons: "Construir", "Borrar Sel", and "Borrar Todo". A checkbox labeled "Cambiar Elemento" is located above a large empty rectangular box.
- Resultado:** This section contains a blue button labeled "¿Es Función?" on the left, a text input field with three dots "..." in the center, and a blue button labeled "Graficar" on the right.

Fuente: Jiménez et al. 2014.

A continuación, se muestran las librerías del lenguaje de programación utilizadas para desarrollar el software para construir funciones:

Tabla 3

Librerías del programa para la construcción de funciones

```

Esta es la pantalla principal

using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Globalization;
using System.ComponentModel;
using ZedGraph;

using System.Drawing;
using System.Web.SessionState;
    
```

Fuente: Jiménez et al, 2011.

El desarrollo de este software, al igual que el simulador del robot delta, cumple con los requisitos funcionales, ya que es posible construir funciones, clasificarlas y graficarlas. Sin embargo, fue posible advertir que: 1) El código no tenía comentarios que ayudarán a su comprensión, 3) Fue desarrollado usando la lógica personal de los alumnos, 3) No es flexible, es decir, solo tiene la aplicación para el cual fue desarrollado y 4) Fue complicado realizar correcciones.

Como se ha visto anteriormente, el desarrollo de software personal cumple con objetivos limitados y es funcional. Sin embargo, es necesario fomentar una cultura de desarrollo de software cumpliendo con metodologías específicas que ayuden a los alumnos de ingenierías a generar un producto de software.

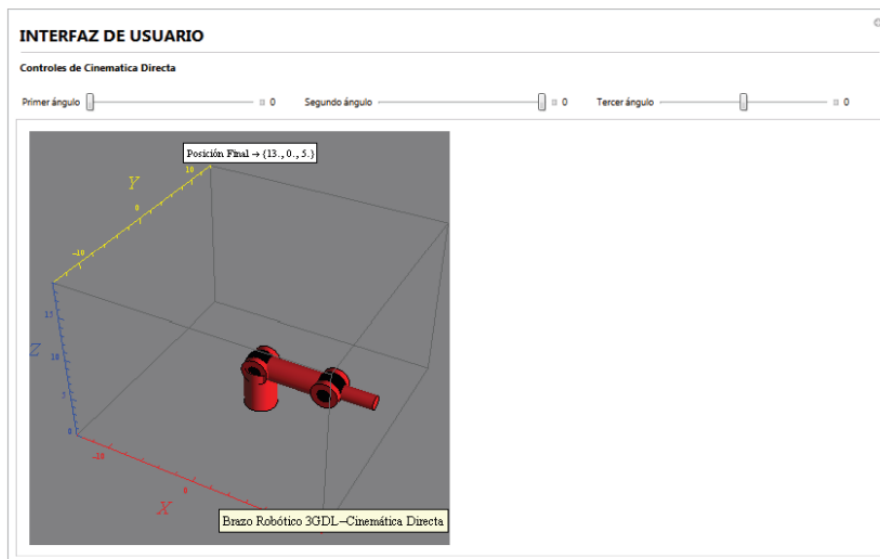
Desarrollo de un simulador mediante el método del método evolutivo incremental

Con el propósito de mostrar a los alumnos de ingeniería la diferencia entre un desarrollo personal de software de un simulador y un producto de software se presenta a continuación una síntesis de la construcción de un simulador robótico (Jiménez et al. 2019) mediante el método evolutivo incremental (Sommerville, 2011).

Para iniciar el desarrollo del simulador con el método evolutivo incremental se les permitió a los alumnos llevar su programa personal que habían realizado en una materia anterior. La figura 4, muestra la salida gráfica del simulador (Jiménez et al. 2019).

Figura 4

Salida gráfica de un simulador de un robot programado en Mathematica 8



Fuente: Jiménez et al. 2019.

Los alumnos fueron instruidos sobre aspectos básicos de la Ingeniería de Software y específicamente sobre el método evolutivo incremental. De acuerdo con Rodríguez (2009), es necesario definir de cuantos incrementos debe llevar el desarrollo y llevar a cabo los pasos siguientes:

Primer incremento

- Análisis
 - Detección de la necesidad.
 - Planteamiento del problema.
 - Análisis de requerimientos.
 - Especificación de la interfaz.
- Diseño.
 - Diseño de la interfaz del usuario.
 - Diseño de la estructura del sistema.
- Codificación
- Pruebas
 - Implementación y prueba de unidades (codificación).
 - Integración y prueba del sistema.

Iteración y evaluación para el incremento siguiente.

Los alumnos eligieron seleccionar el número de incrementos, para el caso presentado en este artículo fueron elegidos tres incrementos los se enlistan a continuación:

- Primer incremento: Cinemática directa.
- Segundo incremento: Cinemática inversa.
- Unión de interfaces.

Con el objetivo de sintetizar el desarrollo del simulador solo se presentarán los resultados generales del segundo incremento (Zarza, 2018).

La explicación general del problema del simulador es la siguiente:

“Se desea construir un simulador de un robot de 3GDL (Grados de Libertad) bajo la plataforma de cálculo formal Mathematica 8 utilizando Cuaterniones unitarios (Jiménez et al. 2023). El sistema debe ser capaz de resolver el problema inverso y el problema directo, y mostrar los resultados en una pantalla de manera que puedan ser visible los movimientos del robot y que al variar los datos pueda ser posible calcular las nuevas posiciones”.

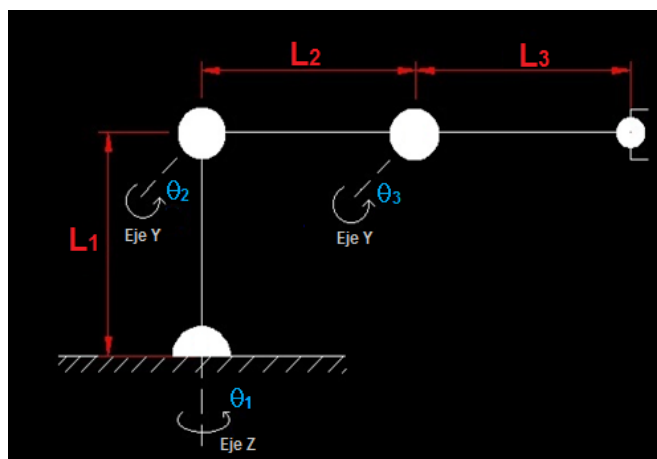
Con el análisis previo, fue posible definir los incrementos necesarios para el desarrollo del producto de software (simulador).

El segundo incremento trata de la cinemática inversa el cual se describe a continuación:

“El problema cinemático inverso consiste en determinar los ángulos ($\theta_1, \theta_2, \theta_3$) y los ejes de las rotaciones (Z, Y, Y) del robot mediante un modelo matemático si se da de entrada la trayectoria o puntos de la trayectoria que describe el punto terminal del robot (ver Figura 5).”

Figura 5

Modelo matemático de trayectoria



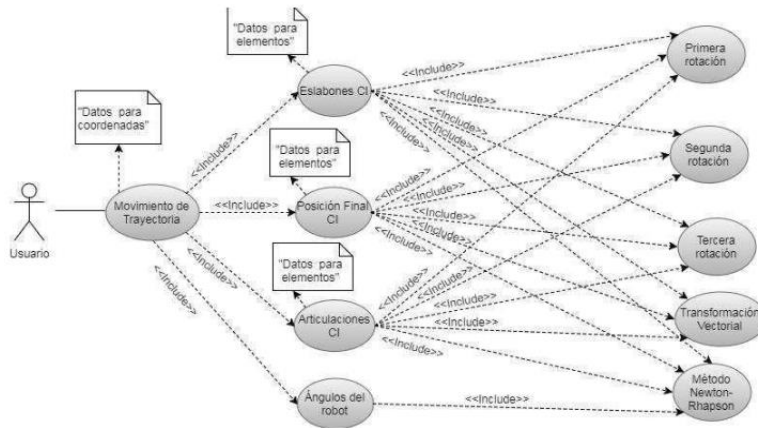
Los requerimientos necesarios para el desarrollo del simulador se resumen en los siguientes puntos (Jiménez et al. 2019):

- Requerimientos Funcionales: Los datos requeridos y de salida, así como las fórmulas o modelos de la cinemática inversa.
- Requerimientos no funcionales: a) Se debe mostrar en la pantalla los datos numéricos de la posición final del robot, b) Se requiere mostrar los datos numéricos de los ángulos y c) Se deben dibujar los elementos de trabajo.
- Requerimientos del Usuario: A) La manipulación de los controles debe ser sencilla, B) No se deben modificar los rangos de movimientos preestablecidos del robot y C) El usuario podrá visualizar los ejes coordenados.
- Requerimientos del sistema.

Para definir los requerimientos del sistema se utilizó una herramienta de UML (Lenguaje Unificado de Modelado) (Ahmad, Iqbal, Ashraf, Truscan y Porres, 2019). El caso de uso que se desarrolló para segundo incremento del simulador se llamó "Caso de uso: Cinemática inversa" y se observa en la Figura 6, y el diagrama de secuencia se presenta en la Figura 7.

Figura 6

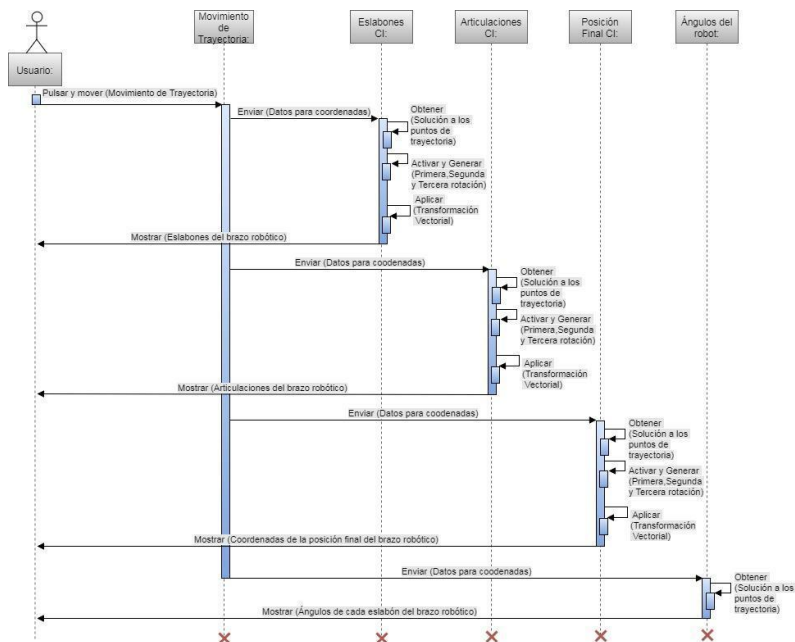
Diagrama de caso de uso del segundo incremento



Fuente: Zarza, 2018.

Figura 7

Diagrama de secuencia



Fuente: Elaboración propia.

La Tabla 4 muestra parte de la codificación del segundo incremento la cual fue desarrollada en Mathematica 8.

Tabla 4

Código del segundo incremento desarrollado en Mathematica 8

Código en Mathematica	Explicación
Las funciones llamadas: "Multiplicación", "Conjugado", "Rotación" y "Vectores de rotación", son funciones básicas que hacen posible el cálculo para las funciones llamadas: "Vectores de rotación rotados 1", "Vectores de rotación rotados 2", "Primera rotación", "Segunda rotación" y "Tercera rotación".	
<pre>Multiplicacionpq[p_, q_] := { p[1] q[1] - p[2] q[2] - p[3] q[3] - p[4] q[4], p[1] q[2] + p[2] q[1] + p[3] q[4] - p[4] q[3], p[1] q[3] + p[3] q[1] - p[2] q[4] + p[4] q[2], p[1] q[4] + p[4] q[1] + p[2] q[3] - p[3] q[2] };</pre>	<p><i>Multiplicación</i> Función que realiza la multiplicación de dos cuaterniones y da su correspondiente resultado.</p>
<pre>Conjugado[p_] := {p[1], -p[2], -p[3], -p[4]};</pre>	<p><i>Conjugado</i> Función que realiza la operación de invertir a signo negativo la parte imaginaria del quaternion.</p>
<pre>Rotacionpq[p_, q_] := Multiplicacionpq[Multiplicacionpq[p, q], Conjugado[p]];</pre>	<p><i>Rotación</i> Función que realiza el cálculo matemático entre el eje de rotación y el elemento a rotar. El cual dará como resultado las coordenadas finales para posicionar el elemento rotado.</p>

Fuente: Zarza, 2018.

Una vez desarrollada la codificación el siguiente paso fue realizar las pruebas del segundo incremento. Este paso es muy importante en el método evolutivo incremental, ya que el desarrollador debe verificar que el programa funciona en forma correcta y que cumple con sus requerimientos. La verificación fue complicada ya que los modelos matemáticos que resultan del problema cinemático inverso fueron no lineales, por lo que se debió aplicar un método numérico para resolver las ecuaciones, en este caso se utilizó el método de Newton-Raphson (Cruz, García y Olais, 2022). Los pasos seguidos para realizar las pruebas del segundo incremento fueron los siguientes:

- Se desarrolló a mano el modelo numérico (desglose del método de Newton-Raphson) del robot y se programó en el paquete Matlab.
- Se desarrollaron algunas interacciones tomando de entrada un punto (X, Y) del brazo del robot hasta que se presentó convergencia.
- Se obtuvieron los parámetros de Cuaterniones en radianes que definen los ejes y el ángulo de rotación.
- Se convirtieron los radianes a grados.
- Los datos del punto (X, Y) de entrada al Matlab se le proporcionaron al simulador desarrollado en Mathematica.
- Se obtuvieron los resultados y se graficaron.

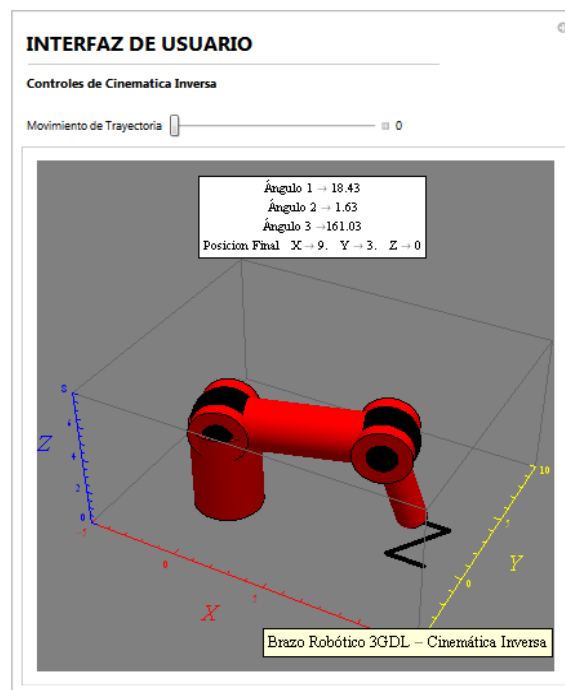
La Tabla 5 muestra los ángulos obtenidos de Matlab para cada eslabón y la Figura 8 muestra el resultado en el simulador.

Tabla 5

Ángulos de cada eslabón del robot manipulador obtenidos de Matlab

ÁNGULOS DE CADA ESLABÓN DEL ROBOT	
Eslabón 1	Ángulo 1: 18.434948822967740512
Eslabón 2	Ángulo 2: 1.6273419080852469243
Eslabón 3	Ángulo 3: 161.03442501155089417

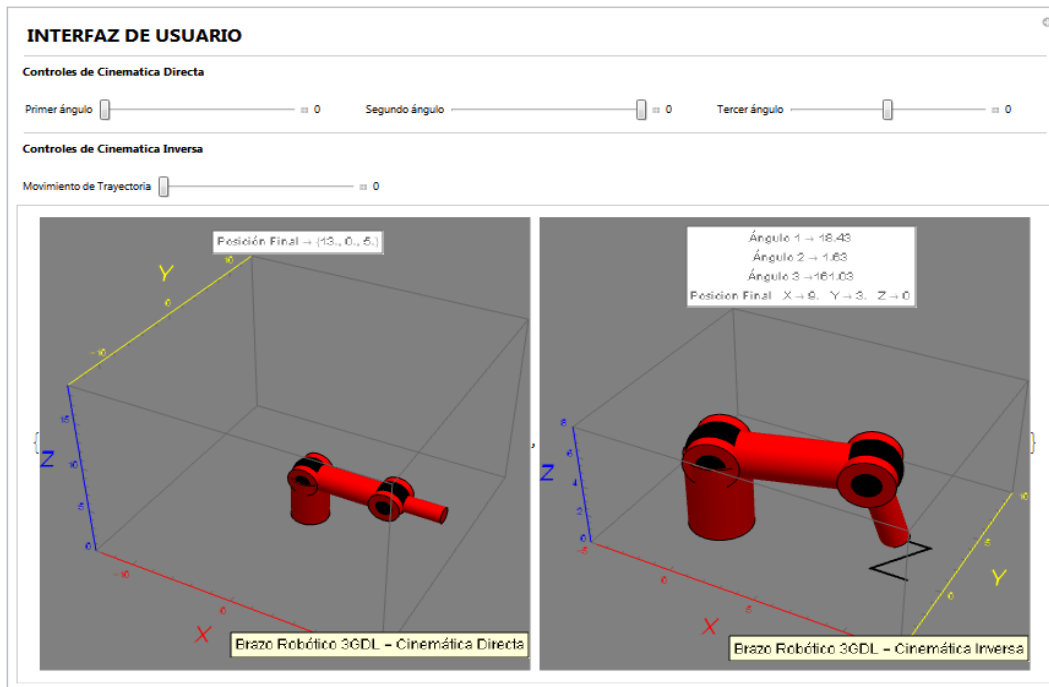
Fuente: Elaboración propia.



Una vez concluido el segundo incremento se procedió a unir las interfaces del problema cinemático directo e inverso (tercer incremento). La salida gráfica del tercer incremento se muestra en la Figura 9 y representa al desarrollo del simulador robótico.

Figura 9

Interfaz de usuario del simulador en su tercer incremento



Fuente: Zarza, 2018.

DISCUSIÓN

La transformación de un software personal a un producto de software desarrollado con metodologías apropiadas, fue clave para que los alumnos de ingeniería comprendieran que los programas que realizan en sus clases o proyectos pueden convertirse en un producto más elaborado y con potencia incluso de comercialización. El proceso de desarrollo del simulador fundamentado con el método evolutivo incremental fue claramente superior a la programación de un modelo (aunque esto ya se sabe de hace mucho tiempo, pero no era conocido por los alumnos que no son del área de TICs), ya que está documentado sistemáticamente, probado-validado, y puede ser escalado. A pesar de que el método de las entregas incrementales se vuelve pesado y tedioso según la complejidad del producto y del número de incrementos, este fue de gran utilidad para mostrar el camino para el desarrollo de un producto de software a los alumnos de ingeniería.

Para el caso de estudio presentado en este trabajo, se pudo constatar que los alumnos batallaron para desarrollar el simulador robótico, dado que no estaban acostumbrados a utilizar una metodología formal. Los problemas no se presentaron en el conocimiento del modelado del robot puesto que ya lo habían realizado en otra clase, ni en el método de solución, ni en el desconocimiento de los paquetes de cálculo formal, más bien fueron en la manera de modelar el diseño (casos de usos y diagramas de secuencias) y en las pruebas y validación. La elección de los incrementos fue otro problema que superaron los alumnos ya que no sabían cómo particionar el problema. A pesar de los inconvenientes presentados, los alumnos lograron adaptarse al método de las entregas incrementales y transformaron su programa a un producto de software.

Con relación a la evaluación de las opiniones de los alumnos (ver Tabla 1) se obtuvieron los resultados siguientes:

Las respuestas del cuestionario del Sí, en todas las preguntas fueron el mínimo 0% y un máximo del 100% y el NO tuvo un mínimo de 10% y un máximo 100%.

La pregunta 1, ¿Habías escuchado con anterioridad lo que es un producto de software?, tuvo el máximo del 100% en NO ya que los alumnos desconocían que era un producto de software. La pregunta 2, ¿Tenías noción acerca del concepto de Ingeniería de Software?, tuvo un 95% de NO puesto que no tenían conocimiento sobre Ingeniería de Software. La pregunta 3, ¿Conocías algún método de ingeniería de software?, tuvo resultados similares a la pregunta 2. La pregunta 4, ¿Se te complicó entender el método evolutivo incremental?, tuvo un 50% de NO y un 50% de respuesta de Sí, y esto resultó lógico ya que ciertamente algunos alumnos se les complicó aplicar el método evolutivo de las entregas incrementales más que otros. La pregunta 5, ¿Consideras necesario que en tu plan de estudios se integre una materia de ingeniería de software?, tuvo un valor similar a la 4, debido a que la mitad consideró que SI era necesario que las carreras de ingeniería tuvieran una materia de ingeniería de software y la otra mitad respondió que NO. La pregunta 6, ¿Consideras importante transformar tu software personal a un producto de software?, alcanzó un 70% de Sí y un 30% de NO. La pregunta 7, ¿Pudiste notar algunas diferencias entre tu programa o producto personal de software del simulador y entre el simulador desarrollado con el método de las entregas incrementales?, fue respondida con un 90% de Sí y un 10% que NO. La pregunta 8, ¿Usarías en el futuro un método de ingeniería de software para transformar tus programas a un producto de software?, arrojó que 60% de SI y 40% de NO. La pregunta 9, ¿Consideras complicado usar un método de ingeniería de software para el desarrollo de simuladores robóticos?, tuvo resultados similares a la pregunta 4. Finalmente, la pregunta 10, ¿Recomendarías a otros alumnos de ingeniería usar un método de ingeniería de software para transformar sus programas en producto de software?, tuvo un 75% de SI, esto es, que si recomendarían utilizar el método de ingeniería de software a otros alumnos de otras ingenierías.

Finalmente, el desarrollo del simulador y de otros proyectos educativos, muestra que la cooperación y la colaboración entre profesores y alumnos de distintas universidades es valiosa y necesaria. En este caso, la Red ALFA representa un instrumento valioso para que los profesores y alumnos de Universidad La Salle Noroeste y la Universidad Tecnológica de Sur de Sonora puedan desarrollar proyectos en conjunto en beneficio de la docencia, la investigación y la transferencia de conocimientos de ambas instituciones.

CONCLUSIONES

En este artículo se presenta un resumen del desarrollo de un simulador robótico en cuyo diseño se aplicó el método de desarrollo de software Evolutivo Incremental. Las principales conclusiones se resumen a continuación.

La colaboración entre instituciones de educación superior, da como resultado el desarrollo de proyectos educativos de gran proyección. El ejemplo presentado en este trabajo fue producto de la colaboración interinstitucional de la Universidad La Salle Noroeste y la Universidad Tecnológica del Sur de Sonora bajo la Red ALFA. Con respecto al desarrollo del simulador robótico los resultados del cuestionario, mostraron que en general los alumnos aceptaron que era importante transformar sus programas a un producto de software debido a las ventajas que ofrece, ya que a pesar de que en algunas preguntas la mitad respondía de manera afirmativa y la otra mitad negativa, la tendencia es positiva puesto que algunos de ellos batallaron más que otros ya que desconocían el método Evolutivo Incremental. La sistematización que ofrece este método para el desarrollo de software fue bien apreciada por los alumnos (que no son de las carreras de TICs) ya que se puede seguir una metodología probada, que tiene los pasos necesarios para el diseño, desarrollo y validación del producto.

Sin lugar a duda que muchos productos de software personal o programas que se desarrollan en las universidades por medio de clases o de proyectos, pueden ser transformados a productos de software en versiones primitivas (no comerciales directamente) pero que pueden ser la base de un producto de más alto valor (producto comercial). Por consiguiente, es de suma importancia promover en las ingenierías la continuidad de los proyectos para que con el tiempo adecuado y las metodologías apropiadas se logren desarrollar productos de software que puedan ser transferidos a las empresas.

REFERENCIAS

Ahmad T., Iqbal J., Ashraf A., Truscan D., y Porres I. (2019). Model-based testing using UML activity diagrams: A systematic mapping study, *Computer Science Review*, 33, pp. 98-112

Amavizca L.O., García A.C., Jiménez E., Duarte G. L., Vázquez J.C. (2014). Aplicación de la metodología semi-ágil ICONIX para el desarrollo de software: implementación y publicación de un sitio WEB para una empresa SPIN - OFF en el Sur de Sonora, México. Twelfth LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2014) "Excellence in Engineering to Enhance a Country's Productivity" July 22 - 24, 2013 Guayaquil, Ecuador, pp. 1-10.

Cárdenas M. (2017). Redes de colaboración científica en el Instituto Politécnico Nacional de México, *Investigación Administrativa*, vol. 50, núm. 127, pp. 192-210.

Cruz S.R., García, M. & Olais, J.M. (2022). NAO robots as context to teach numerical methods. *Int J Interact Des Manuf*, 16, 1337–1356.

Jiménez E., Cepeda M.H., Gutiérrez E.K., Valdez L.O., Peñuñuri T., y Reyes L.A. (2014). Desarrollo de un software para la generación, validación y clasificación de funciones: una contribución a la enseñanza de las matemáticas. In *Proceedings of COPEC World Congress (Vol. 5)*.

Jiménez E., Cepeda M.H., Sánchez L.M., Félix C.A., Núñez A.R. Gutiérrez E.K., Valdez L.O., Ojeda R., Reyes L.A. y Peraza R. (2011). Desarrollo de un programa para generar y clasificar programas. Primera edición. Universidad Tecnológica del Sur de Sonora. México.

Jiménez E., Espinoza R.N., López A.A., Núñez E., Limón P.A., Fierro K. y Cuenca F. (2023). Quaternion Modeling of a Delta Planar Robot and Training of an Enhanced Multilayer Neural Network to Solve the Inverse Kinematic Problem. *Digitalization and Management Innovation A.J. Tallón-Ballesteros and P. Santana-Morales (Eds.)*. IOS Press, pp. 655-661.

Jiménez E., García L.A., Núñez E., Bojórquez G.I., Navarro L.B., Juárez H., Luna G., y Amavizca L.O. (2014). Experiencias en investigación y docencia en la carrera de Ingeniería Mecatrónica en la Universidad la Salle Noroeste. Twelfth LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2014) "Excellence in Engineering to Enhance a Country's Productivity" July 22 - 24, Guayaquil, Ecuador, pp. 1-10.

Jiménez E., Martínez V.M. y Luna G. (2013). Algunas Consideraciones Sobre las Redes de Colaboración: Experiencias en el Sur de Sonora, México, Eleventh LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2013) "Innovation in Engineering, Technology and Education for Competitiveness and Prosperity" August 14 - 16, Cancún, México. *Algunas Consideraciones Sobre las Redes de Colaboración: Experiencias en el Sur de Sonora, México*, pp. 1-2.

Jiménez E., Ochoa F., Martínez V. (2004). "Red Alfa: Red Universitaria Empresarial". Informe Interno de Investigación No 1. RED ALFA – ULSA Noroeste. Sonora México.

Jiménez E., Zarza M., Rivera J.O. Contreras A., Amavizca L.O. y Vázquez J.C. (2019). Desarrollo de un Simulador Robótico Utilizando el Método Evolutivo de las Entregas Incrementales. *Tecnificación Mecatrónica*. Primera edición. Asociación Mexicana de Mecatrónica A.C. México, capítulo 28, pp. 372-384

Martínez V.M., Jiménez E., Beltrán Y., Armenta I. (2010). La innovación como medio para el desarrollo económico regional: creación de empresas de base tecnológica Spin-off, *Memorias del XVI Congreso Internacional Anual de la SOMIM 22 al 24 de septiembre, 2010 monterrey, nuevo león, México*.

Martínez, V. M., Jiménez, E., Arellano, L., Uzeta, C. R., Sotelo, R., & Reyes, L. A. (2012). Desarrollo de equipos para la manufactura de muebles rústicos de la etnia guarijía: un ejemplo de transferencia tecnológica. XII International Conference on Engineering and Technology Education, pp. 190-194.

Montilva J.A. y Barrios J. (2021). Ingeniería del Software: un enfoque basado en procesos. Primera Edición. Universidad de Los Andes. Venezuela.

Navarro A., Fernández J. D., Morales J. (2013). Revisión de metodologías ágiles para el desarrollo de software, PROSPECTIVA, vol. 11, núm. 2, pp. 30-39

Osorio C. (2019). La educación CTS: un espacio para la cooperación iberoamericana, Revista CTS, nº 42, vol. 14, pp. 99-114.

Pressman, R.S., (2002). Ingeniería del Software. Un enfoque práctico, quinta edición, España.

Rodríguez, L. (2009). Propuesta de un método de ingeniería de software para sistemas mecatrónicas: simulador de un péndulo balístico. Tesis de Maestría en Ciencias de la Ingeniería Mecatrónica. Instituto Tecnológico Superior de Cajeme.

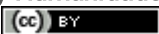
Romero F. (2010). Modelación y simulación de la cinemática inversa y directa de un robot delta de 3 grados de libertad con el método geométrico. Tesis de Licenciatura en Ingeniería Mecatrónica. Universidad La Salle Noroeste.

Sebastián J. (2000). Las Redes de Cooperación como modelo organizativo y funcional para la I+D, 7(15), pp. 97-111.

Sommerville I. (2011). Ingeniería de Software, Novena edición Pearson Educación, México.

Villanueva Á., Fernández A. y Palomares D. (2014). Propiedades relacionales de las redes de colaboración y generación de conocimiento científico: ¿Una cuestión de tamaño o equilibrio? Revista Española De Documentación Científica, 37(4), e068.
<https://doi.org/10.3989/redc.2014.4.1143>

Zarza M. (2018). Desarrollo de un programa de software de la cinemática directa e inversa de un robot manipulador de 3 GDL mediante el modelo de las entregas incrementales. Tesis de Licenciatura en Ingeniería Mecatrónica. Universidad La Salle Noroeste. México.

Todo el contenido de **LATAM Revista Latinoamericana de Ciencias Sociales y Humanidades**, publicados en este sitio está disponibles bajo Licencia **Creative Commons** .