**F\_AA\_225**

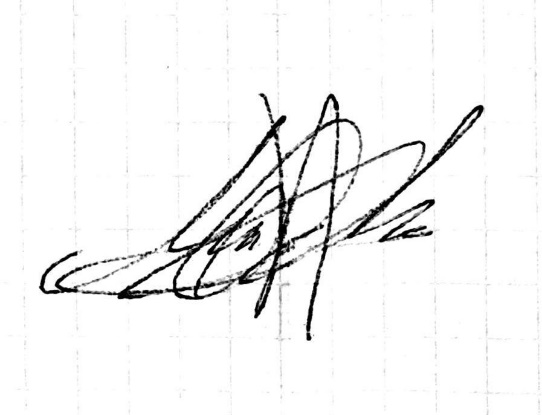
**FACULTAD /ESCUELA DE INGENIERÍA DE SISTEMAS**

**CARRERA/PROGRAMA DE INGENIERÍA EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

**PLAN DE TRABAJO DE TITULACIÓN/TESIS**

**TIPO DE TRABAJO DE TITULACIÓN: PROYECTO INTEGRADOR**

|  |  |
| --- | --- |
| **I.- INFORMACIÓN BÁSICA** | |
| **PROPUESTO POR:**  Luis Andrés Andrade Cabrera | **LÍNEA DE INVESTIGACIÓN:**  DICC-A3-L2 Creación y Gestión del Software |
| **AUSPICIADO POR:**  MSc. Vicente Adrián Eguez Sarzosa  PhD. Diana Cecilia Yacchirema Vargas | **FECHA:**  22/03/2021 |
| **II.- INFORMACIÓN DEL TRABAJO DE TITULACIÓN** | |
| 1. **Título del Trabajo de Titulación**   CREACIÓN DE PLATAFORMA DE SOFTWARE SUPERVISORIO EN JAVA PARA EL LABORATORIO DE REDES INDUSTRIALES DEL DACI - FIEE | |
| 1. **Planteamiento del Problema**   La utilización de software específico para el desarrollo de prácticas de laboratorio es clave para el aprendizaje significativo de los estudiantes de la carrera de Electrónica y Control y de la carrera de Automatización de la Facultad de Ingeniería Eléctrica y Electrónica de la Escuela Politécnica Nacional, ya que como parte de su perfil profesional deben desarrollar conocimientos y destrezas al implementar sistemas de control industrial [1].  El costo del software de control industrial que es utilizado para el laboratorio del Departamento de Automatización y Control Industrial (DACI), tiene precios por licencia, por ejemplo, del software AVEVA Edge, desde los $3,240 hasta los $20,740 dólares americanos [2]. El software en mención permite crear distintas pantallas gráficas, comunicarse usando más de 250 controladores disponibles a diversos tipos de PLC, registro de eventos que garanticen la trazabilidad de acciones iniciadas por los usuarios, etc.  Actualmente el DACI utiliza el software Wonderware® InTouch® HMI, que es una versión antigua de AVEVA Edge [3] y las prácticas de laboratorio no utilizan todas las opciones que el software ofrece siendo posible optar por altern­ativas de menor costo que tengan solo las opciones más utilizadas y permitir incluso que los estudiantes puedan tenerlo en sus computadores de manera legal.  Estas alternativas no son solo aplicables para el DACI, sino que se puede extender su uso a la pequeña y mediana empresa [4], incentivando su desarrollo y productividad mediante el uso eficiente de los sistemas de control y administración adecuada de la información sin afectar demasiado el presupuesto de dichas empresas.  El presente proyecto propone la creación de un software para la implementación de aplicaciones de control supervisorio bajo licencia libre y a bajo costo para el laboratorio de Redes Industriales del DACI, que permita el desarrollo de prácticas de laboratorio utilizando elementos y sus interacciones especificadas similares a los usados en implementaciones prácticas reales de sistemas de control industrial. Además, este software será desarrollado en el lenguaje de programación JAVA, de modo que permita su ejecución en diferentes sistemas operativos. | |
| 1. **Justificación**   3.1 Justificación Teórica  Como se especifica en [5] los sistemas de control supervisorio y adquisición de datos (SCADA, por sus siglas en inglés) permiten gestionar y controlar un sistema sea local o remoto mediante una interfaz gráfica que permite al usuario comunicarse con el sistema de control de un proceso industrial. La principal función de un SCADA es la de supervisar, esto significa que un operador no solo monitoreará las variables del proceso industrial, sino que puede actuar y modificar dichas variables en tiempo real, en síntesis, gracias a la interfaz brindada por un SCADA se puede observar un proceso, detectar errores y corregirlos en tiempo real.  La visualización de cada variable a ser controlada, dentro de una pantalla mostrada al usuario, se denomina Interfaz Humano - Máquina (HMI, por sus siglas en inglés). Un HMI básico solamente mostrará variables, pero no ofrecerá la posibilidad de ejecutar acciones para solución de problemas sin que el sistema deba detenerse [5], pero en un SCADA se añade la capacidad de interactuar con ellas en tiempo real, pues además de mostrar el estado de variables, es capaz de detectar si estas variables han ingresado en rangos de alarma para reportarlo al operador de forma inmediata, también se proporcionan pantallas con los comandos necesarios que ofrecen un mecanismo para solucionar los problemas encontrados en el proceso supervisado.  Los laboratorios del DACI implementan sistemas SCADA utilizando controladores lógicos programables (PLC, por sus siglas en inglés) de modelo MicroLogix 1100 que como se referencia en [6] permiten comunicarse a través de la red mediante el protocolo industrial Ethernet/IP™, entonces en este proyecto integrador se busca desarrollar un software para la implementación de aplicaciones de control supervisorio, es decir, un HMI/SCADA que tendrá las características no solo de visualización de las variables del proceso, sino que también podrá interactuar con estas variables y modificarlas en tiempo real, al acceder a los controladores PLC del proceso.  3.2 Justificación Metodológica.  Para el desarrollo del software propuesto se aplicará la metodología SCRUM [7] que permite el desarrollo bajo procesos iterativos e incrementales y permitirá obtener versiones de software con valor para el DACI durante todo el desarrollo del proyecto, es decir, el Departamento irá recibiendo versiones útiles a través de cada iteración y permitirá ir afinando los requerimientos deseados para el funcionamiento adecuado del software con los dispositivos del laboratorio de Redes Industriales. Utilizar esta metodología viene dado por su éxito al momento de cumplir con las expectativas y su flexibilidad ante los cambios, lo que permitirá al final del desarrollo obtener un producto maduro y preparado para trabajar en ambientes reales.  3.3 Justificación Práctica  Este trabajo de titulación busca crear un software para la implementación de aplicaciones de control supervisorio que permita el desarrollo de las prácticas del laboratorio de Redes Industriales del DACI, este software permitirá desarrollar el HMI para los sistemas SCADA que son estudiados e implementados en este departamento, es decir, se requiere de una funcionalidad semejante por ejemplo a las ofrecidas por AVEVA Edge. Finalmente indicar que se cuenta con el apoyo del DACI para la obtención de requerimientos, pruebas en entornos reales y verificación de cumplimiento de las funcionalidades esperadas. | |
| 1. **Objetivo General**   Implementar una plataforma de software para el desarrollo de aplicaciones de control supervisorio constituido por una aplicación de escritorio en JAVA que permita la comunicación con PLCs MicroLogix 1100 a través del protocolo EtherNet/IP™ para el DACI. | |
| 1. **Objetivos Específicos**    * Determinar los requisitos del sistema a nivel funcional y el alcance del software supervisorio.    * Diseñar la arquitectura e interfaces de usuario buscando asemejarse a HMIs profesionales.    * Implementar una aplicación de escritorio en JAVA que permita la interoperabilidad entre sistemas operativos, y se conecte a sistemas SCADA implementados en los laboratorios.    * Aplicar pruebas de aceptación para comprobar que se cumplan con los requerimientos solicitados por el DACI.    * Comprobar la calidad del producto a través de encuestas, para medir la usabilidad y facilidad de uso del sistema. | |
| 1. **Metodología**   La metodología a ser usada en el desarrollo de este proyecto de titulación es SCRUM, que busca mediante procesos iterativos e incrementales [7] satisfacer las necesidades del cliente mediante la priorización de las características que desean ser obtenidas. En este caso particular será una plataforma de software que permita el desarrollo de sistemas de control supervisorio constituido por una aplicación de escritorio en JAVA que permita la comunicación con PLCs MicroLogix 1100 a través del protocolo Ethernet/IP™ para el DACI.  El proyecto entonces se estructurará con base en las 5 fases propias de la metodología SCRUM [8]:   * Inicio: En esta fase se busca identificar las necesidades básicas del SPRINT, [8], el cual es un mini proyecto que permita responder preguntas de esta fase que se desea, como se lo desea y para cuando se lo desea, estas preguntas serán respondidas por el Jefe de Departamento del DACI, así como profesores que requieran esta plataforma de software. De esta fase se debe obtener una lista de requerimientos priorizada del producto. * Planificación y estimación: Esta es una de las fases más importantes, aquí se definirán las historias de usuario [9], mismas que son descripciones de lo que un usuario quiere hacer dentro de un producto de software para obtener algo que considera valioso, también se define las tareas y la iteración de estas, es decir, los tiempos de entrega y una lista ordenada para clasificar el trabajo en orden de prioridad. * Implementación: En esta fase es donde se aplica el ciclo de desarrollo de software, es decir las fases de análisis, diseño, desarrollo y pruebas de la plataforma de software que se está desarrollando. En esta fase se crearán los entregables, se iniciará el diseño de la arquitectura y de las interfaces de usuario, después se iniciará el desarrollo de la aplicación en JAVA. * Revisión y Retrospectiva: Luego de que se tienen entregables a modo de prototipo, se realiza una revisión del proceso de trabajo, definiendo opiniones y soluciones a los problemas que se vayan encontrando. * Lanzamiento: En esta fase se realizan los envíos de entregables y se hace una retrospectiva del proyecto en total. Es aquí donde se pueden realizar las pruebas de aceptación finales, así como las encuestas para comprobar la calidad del producto. | |
| 1. **Plan de Trabajo**   Para el desarrollo del proyecto aplicando la metodología SCRUM se seguirá el siguiente plan:   1. Determinar los requerimientos del software supervisorio a través de reuniones con los encargados del DACI. 2. Aplicar el marco de trabajo SCRUM.    1. Inicio       1. Definir la visión del proyecto       2. Identificar al SCRUM Master y a los Interesados (Stakeholders)       3. Crear la lista de requerimientos priorizada del producto.    2. Planificación y estimación       1. Definir las historias de usuario       2. Crear de tareas       3. Definir la lista de tareas del Sprint    3. Implementación       1. Creación de Entregables    4. Revisión y Retrospectiva       1. Demostración y validación del Sprint       2. Retrospectiva del Sprint    5. Lanzamiento       1. Enviar entregables       2. Retrospectiva del proyecto 3. Realizar pruebas al software supervisorio    1. Diseñar y ejecutar pruebas de rendimiento    2. Seleccionar y aplicar una encuesta de usabilidad    3. Procesar y evaluar los datos obtenidos    4. Realizar las mejoras que se consideran necesarias 4. Elaboración del Informe final del Proyecto | |
| 1. **Bibliografía**   [1] “Ingeniería en Electrónica y Control - Facultad de Ingeniería Eléctrica y Electrónica - FIEE”. https://fiee.epn.edu.ec/index.php/oferta-academica/pregrado/ingenieria-en-electronica-y-control (consultado dic. 12, 2020).  [2] “Wonderware California”. https://california.wonderware.com/Licenses/LicenseInfo.aspx (consultado dic. 12, 2020).  [3] S. Gamboa y D. Maldonado, “INTRODUCCIÓN A INTOUCH HMI”. FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA, nov. 2020, [En línea]. Disponible en: <http://ciecfie.epn.edu.ec/Automatizacion/Instrumentacion/Redes/2020B/P1_CP-Rindustriales.pdf>.  [4]C. Coro y C. Fabricio, “Importancia del uso del sistema scada para el desarrollo empresarial”, ago. 2019, Consultado: mar. 15, 2021. [En línea]. Disponible en: http://dspace.uniandes.edu.ec/bitstream/123456789/10386/1/PIUSDSIS024-2019.pdf.  [5] E. Pérez-López y E. Pérez-López, “Los sistemas SCADA en la automatización industrial”, *Revista Tecnología en Marcha*, vol. 28, núm. 4, pp. 3–14, dic. 2015.  [6] Rockwell Automation, *MICROLOGIX PROGRAMMABLE CONTROLLERS*. 2019.  [7] “What Is Scrum Methodology? & Scrum Project Management”, *Digite*, sep. 30, 2019. https://www.digite.com/agile/scrum-methodology/ (consultado ene. 04, 2021).  [8] “▷ Fases de la Metodología Scrum 【5 Etapas y 16 Procesos】”, *ComparaSoftware Blog*, jul. 10, 2020. https://blog.comparasoftware.com/fases-metodologia-scrum/ (consultado ene. 31, 2021).  [9] “User Stories: What They Are And Why And How To Use Them”. https://www.digite.com/agile/user-stories/ (consultado ene. 31, 2021). | |
| 1. **Cronograma**  |  |  |  |  | | --- | --- | --- | --- | | Fase | Actividades | Duración (horas) | Semanas | | Determinar los requerimientos del software supervisorio. | Determinar los requerimientos de software supervisorio mediante reuniones con los encargados del DACI | 10 | 1 | | Aplicación del Framework Scrum | Inicio | 10 | 1 | | Planificación y estimación de cada sprint | 4 | 8 | | Implementación | 11 | | Revisión y retrospectiva | 4 | | Lanzamiento | 4 | | Pruebas | Pruebas de Rendimiento | 8 | 1 | | Pruebas de Usabilidad | 8 | | Procesar y evaluar los datos obtenidos | 4 | 10 | | Realizar las mejoras que se consideren necesarias | 10 | | Informe Final | Elaboración del Informe Final | 20 | 3 | | TOTAL (semanas) | | | 24 | | TOTAL (horas) | | | 420 | | |



Firma

|  |  |  |
| --- | --- | --- |
| Luis Andrés Andrade Cabrera |  |  |
| Proponente 1 |  |  |
| Email: luis.andrade03@epn.edu.ec  Telf.: +593997645903 |  |  |

Firma Firma



|  |  |  |
| --- | --- | --- |
| MSc. Vicente Adrián Eguez Sarzosa |  | PhD. Diana Cecilia Yacchirema Vargas |
| DIRECTOR |  | CODIRECTORA |
| Email: adrian.eguez@epn.edu.ec  Telf.: +59322976300,4729 |  | Email: diana.yacchirema@epn.edu.ec  Telf.: +593 990739368 |