

**Universidad de las Ciencias Informáticas**

**Facultad 5**



**Trabajo de Diploma para optar por el título de  
Ingeniero Informático**

**Título:**

**Propuesta de visualización de animaciones para objetos gráficos  
del Visualizador web del SCADA GALBA.**

**Autor(es):**.Yanisleyidis Rodríguez Tamayo.

**Tutor(es):** Ing. Yosvani Ramírez Martínez  
Ing. Ariannys Garrido Saroza.

**La Habana, junio de 2015**

**“Año 56 de la Revolución”**

A large, faded, grayscale portrait of Ernesto Guevara, the Argentine Marxist revolutionary, serves as the background for the page. He is wearing his iconic beret with a star on the side and has a slight smile. The text is overlaid on the lower half of his face.

*No se vive celebrando  
victorias, sino  
superando derrotas.*

*Ernesto Guevara*

# *Declaración de autoría*

---

Declaro ser la autora de la presente tesis: **“Propuesta de visualización de animaciones para objetos gráficos del Visualizador web del SCADA GALBA”** y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo el presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2015.

---

Firma del Autor

Yanisleydis Rodriguez Tamayo

---

Firma del Tutor

Ing. Yosvani Ramírez Martínez

---

Firma del Tutor

Ing. Ariannys Garrido Saroza

**Tutor:** Ing. Yosvani Ramírez Martínez  
Universidad de las Ciencias Informáticas  
La Habana, Cuba  
E-mail: [yramz@uci.cu](mailto:yramz@uci.cu)

**Tutor:** Ing. Ariannys Garrido Saroza  
Universidad de las Ciencias Informáticas  
La Habana, Cuba  
E-mail: [asaroza@uci.cu](mailto:asaroza@uci.cu)

### *Dedicado a toda mi familia:*

*Mis dos madres, padre, dos hermanos, dos primos, tía, tío y bisabuela. Especialmente a mi papá y abuela quien realmente nunca he podido llamar así, ya que la considero mi mamá.*

*Esto más que una dedicatoria es un regalo que le estoy haciendo para ambos, como muestra de que todos sus sacrificios no han sido en vano; y porque creo que es la forma mas bonita y adecuada que un hijo tiene para pagarle a sus padres, por todo esfuerzo y dedicación durante tantos años.*

*Le agradezco a todo mi familia en general:*

*A mi mamita Claribel, primeramente por haberme traído al mundo, y luego por estar ahí siempre para mí llena de amor y cariño. A mi otra mami Daisy y papá, por ser mis guías desde siempre, por brindarme apoyo en todas mis decisiones y haberme malcriado tanto. A mi tiita Zure por tanta confianza y cariño, le agradezco también desde el fondo de mi corazón a mi abuelita Candita, tios chino y Canty. A los más pequeños de casa: los dos bebés, también Leticia y Alexey por traer tantos momentos de felicidad a mi vida. A todas aquellas profesoras y entrenadoras que me impartieron clases y supieron ganarse mi aprecio y cariño. Especialmente a mis tutores que han sido mis amigos por encima de todo, que gracias a su excelente trabajo han contribuido a cumplir mi meta principal. A las grandes amigas y amigos que he conocido desde el primer año: Edith, Dámaso, Ernesto y Lyanis por brindarme ayuda cuando más los necesité, especialmente a Yanisleydis, que considero una hermana con quien hubiese querido compartir de todo corazón este puesto. A todos los que son y los que fueron mis compañeros de aula, por tantos momentos de risa y diversión. A Miguel Ángel por haber aguantado mis malas crianzas tanto tiempo sin protestas.*

La informática ha elevado a gran escala el desarrollo de los procesos industriales. Dentro de las aplicaciones informáticas que se utilizan para lograr un control eficiente de estos procesos, se encuentran los sistemas de supervisión, control y adquisición de datos, conocidos como SCADA por sus siglas en inglés. Estos sistemas permiten supervisar y controlar a distancia una instalación, proceso o sistema de características variadas.

En el Centro de Informática Industrial (CEDIN) de la Universidad de las Ciencias Informáticas (UCI), se desarrolla, en convenio con la Empresa de Petróleos de Venezuela S.A. (PDVSA), el SCADA Guardián del ALBA (GALBA). Como estrategia administrativa el centro está proyectando sus productos hacia una plataforma web, esto se evidencia con el Visualizador web que se está desarrollando actualmente en el módulo Interfaz Hombre-Máquina (HMI) del GALBA.

La presente investigación surge a partir de la necesidad de tener agrupadas un conjunto de animaciones que definan el comportamiento visual de los objetos gráficos permitiendo la simulación en tiempo real en el Visualizador web del HMI. Su desarrollo propicia que toda la información adquirida sea representada con los diversos eventos ocurridos en los procesos en tiempo real, siendo esto beneficioso para lograr una mejora de los dinamismos en los despliegues y contribuya a la toma de decisiones de los operadores. Para la construcción de esta solución se utilizó como guía de desarrollo la metodología Proceso Unificado Ágil (AUP), utilizando las técnicas de modelación establecidas por el Lenguaje Unificado de Modelado (UML). Se emplea para la implementación el lenguaje de programación JavaScript apoyado en los marcos de trabajo JQuery y AngularJs. Para validar que los resultados obtenidos eran los esperados se realizó un conjunto de pruebas utilizando el método de Caja Negra con la técnica partición de equivalencia. Durante la etapa de pruebas se obtuvieron un conjunto de no conformidades, las cuales fueron resueltas satisfactoriamente en cada iteración.

**Palabras Clave:** HMI, animaciones, objetos gráficos, Visualizador web.

<b>INTRODUCCIÓN</b> .....	1
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>6</b>
1.1 Introducción .....	6
1.2 Sistemas de supervisión, control y adquisición de datos .....	6
1.3 Sistema SCADA Guardián del ALBA .....	7
1.3.1 Interfaz Hombre-Máquina .....	8
1.3.2 Módulo de HMI web del sistema SCADA Guardián de ALBA .....	9
1.3.3 Visualización gráfica en el entorno web del HMI .....	10
1.3.4 Tecnologías para la visualización gráfica en el entorno web del HMI de los sistemas SCADA. ....	11
1.4 Metodología de desarrollo de software .....	12
1.5 Tecnologías y herramientas a utilizar .....	13
1.5.1 Herramientas CASE .....	13
1.5.2 Lenguaje de modelado .....	14
1.5.3 Entorno de desarrollo .....	14
1.5.4 Lenguajes de programación .....	15
1.5.4.1 JavaScript .....	15
1.5.4.2 HTML .....	16
1.5.4.3 CSS .....	16
1.5.5 Marco de trabajo para el desarrollo de aplicaciones web .....	16
1.5.5.1 JQuery .....	17
1.5.5.2 AngularJs .....	17
1.6 Conclusiones parciales: .....	18

<b>CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA.....</b>	<b>19</b>
2.1 Introducción: .....	19
2.2 Modelo de dominio.....	19
2.4 Especificación de requisitos .....	21
2.4.1 Requisitos funcionales.....	21
2.4.2 Requisitos no funcionales.....	22
2.5 Modelo de casos de uso del sistema .....	23
2.5.1 Descripción de los actores del sistema.....	23
2.5.2 Definición de los casos de uso del sistema.....	24
2.5.2 Descripción de los casos de usos del sistema.....	25
2.5 Patrones de arquitectura .....	35
2.5.1 Patrón arquitectónico Vista Vista-Modelo Modelo.....	35
2.6 Patrones de diseño .....	36
2.7 Conclusiones parciales .....	39
<b>CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS .....</b>	<b>40</b>
3.1 Introducción .....	40
3.2 Modelo de implementación .....	40
3.2.2 Diagrama de componentes.....	41
3.2.3 Descripción del diagrama de componentes .....	41
3.3 Modelo de despliegue .....	42
3.4 Estándares de codificación. ....	43
3.5 Pruebas del sistema.....	44

3.5.1 Prueba de Caja Negra .....	45
3.5.2 Resultados de las Pruebas .....	52
3.6 Beneficios del sistema .....	53
3.7 Conclusiones parciales .....	53
<b>CONCLUSIONES GENERALES .....</b>	<b>55</b>
<b>RECOMENDACIONES .....</b>	<b>56</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>57</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>59</b>

Tabla 1: Componentes del dominio.....	20
Tabla 2: Actor del sistema.....	24
Tabla 3: Descripción del caso de uso: Visualizar animación de traslación. ....	30
Tabla 4: Descripción del caso de uso: Visualizar animación de visibilidad. ....	25
Tabla 5: Descripción del caso de uso: Visualizar animación de texto.....	26
Tabla 6: Descripción del caso de uso: Visualizar animación de imagen.....	27
Tabla 7: Descripción del caso de uso: Visualizar animación de rotación.....	29
Tabla 8: Descripción del caso de uso: Visualizar animación de color.....	32
Tabla 9: Descripción del caso de uso: Visualizar animación de escalado. ....	33
Tabla 10: Descripción de los componentes del diagrama de componentes. ....	41
Tabla 11: Diseño del caso de prueba Visualizar animación de traslación .....	46
Tabla 12: Diseño del caso de prueba Visualizar animación de visibilidad .....	47
Tabla 13: Diseño del caso de prueba Visualizar animación de escalado .....	48
Tabla 14: Diseño del caso de prueba Visualizar animación de imagen.....	50
Tabla 15: Diseño del caso de prueba Visualizar animación de color.....	50
Tabla 16: Diseño del caso de prueba Visualizar animación de texto.....	51
Tabla 17: Diseño del caso de prueba Visualizar animación de rotar. ....	51

Figura 1: Esquema básico de un sistema de Adquisición, supervisión y control.....	7
Figura 2: Modelo conceptual del sistema. ....	20
Figura 3: Diagrama de Caso de Uso del Sistema.....	24
Figura 4: Representación del patrón arquitectónico Vista-Vista Modelo-Modelo .....	36
Figura 5: Fragmento de código del patrón Adaptador en la solución propuesta. ....	37
Figura 6: Fragmento de código del patrón Observador en la solución propuesta. ....	37
Figura 7: Fragmento de código del patrón Iterador en la solución propuesta.....	38
Figura 8: Fragmento de código del patrón Constructor en la solución propuesta. ....	39
Figura 9: Diagrama de componentes de la solución propuesta.....	41
Figura 10: Diagrama de despliegue de la solución propuesta.....	43
Figura 11: Iteraciones de las NC detectadas durante la prueba de Caja Negra. ....	52

## INTRODUCCIÓN

En la sociedad actual los avances tecnológicos marcan nuevos paradigmas en todas sus ramas. Los últimos avances científicos-técnicos y la gran cantidad de información existente en la actualidad han propiciado el surgimiento de una nueva era, la era de la información y el conocimiento. En este contexto donde se hace necesaria la gestión de la información de forma eficiente, las Tecnologías de la Información y las Comunicaciones (TIC) desempeñan un papel imprescindible. A su vez son un factor de vital importancia en la transformación y el desarrollo para el sector empresarial.

En las industrias se utilizan diferentes herramientas para la ejecución, monitoreo y control de los procesos industriales; las que al inicio de la automatización, eran bastante sencillas. Recientemente estas herramientas han ganado en complejidad y funcionalidad de forma exponencial, esto ha permitido una disminución considerable en la carga de trabajo de los operadores en cuanto a tareas se refiere, así como la peligrosidad en las labores que realizan. En la actualidad, los procesos industriales se monitorean y controlan mediante los sistemas de supervisión, control y adquisición de datos (SCADA, por sus siglas en inglés). Estos sistemas proporcionan gran información de los procesos de forma oportuna para la toma de decisiones y centralizan el funcionamiento de una empresa en una reducida cantidad de estaciones de trabajo, mejorando altamente la eficacia de dichos procesos.

Con la misión de formar profesionales comprometidos con su patria y altamente calificados en la rama de la Informática, capacitados para desarrollar aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación y servir de soporte a la industria cubana de la informática, se crea la Universidad de las Ciencias Informáticas (UCI). La universidad orienta su producción entorno a centros de desarrollo entre los que se destaca el Centro de Informática Industrial (CEDIN) perteneciente a la facultad 5, el cual está desarrollando varios software para contribuir a la automatización de procesos industriales, resaltando entre sus productos el sistema SCADA Guardián del ALBA (GALBA). ([UCI, 2015](#))

El sistema SCADA Guardián del ALBA es una aplicación para la supervisión y el control de procesos que se comunica con dispositivos de campo, está compuesta por varios módulos que interactúan entre si, los cuales integran funcionalidades de alto nivel que permiten mejoras significativas en los procesos

industriales. Como estrategia administrativa está proyectando sus productos hacia la plataforma web, por las diversas ventajas que esta presenta. Sin embargo el Visualizador web que se está desarrollando en el CEDIN, aunque permite simular los dispositivos de campo representados en las vistas, que se desean supervisar mediante sinópticos gráficos, carecen de comportamientos diferenciados que mejoren la representación gráfica. Una de las opciones para representar lo que sucede en el proceso en tiempo real, es variando las propiedades de los objetos gráficos mediante las animaciones. Actualmente estas animaciones no se encuentran implementadas en el Visualizador web del módulo Interfaz Hombre-Máquina (HMI por su siglas en inglés), provocando limitaciones en los posibles estados que pueda tomar el proceso que se esté controlando, ya que no se observa el comportamiento de los componentes gráficos en los despliegues teniendo en cuenta la actualización de la información en tiempo real, este inconveniente en ocasiones dificulta las decisiones que puedan tomar los operadores limitando las interpretaciones de las informaciones de una manera más rápida e intuitiva. Además no existen definiciones del comportamiento de los componentes gráficos de manera genérica para que puedan ser reutilizados y lograr la visualización de una manera más óptima.

El Visualizador web del HMI representa la información adquirida mediante los diferentes componentes gráficos, sin ningún comportamiento asociado a los diversos eventos definidos, lo que trae como consecuencia:

- **Deficiente interpretación de valores:** El monitoreo de los procesos se ve afectado, pues cuando el nivel de detalle o información es elevado se hace engorroso leer e interpretar gran cantidad de textos. Como también se hace difícil la asociación de un texto a un determinado concepto o idea, lo cual no sucedería con imágenes, íconos y figuras que varían su comportamiento en dependencia de algún evento asociado, permitiendo una lectura e interpretación más rápida e intuitiva.
- **Deficiente simulación de los procesos:** Los objetos gráficos no muestran comportamiento diferenciado, por lo que no se evidencia una simulación verídica de los procesos que ocurren en el campo, como son: movimiento, rotación, cambio de color o de tamaño.

Teniendo en cuenta la situación planteada con anterioridad se define el siguiente **problema de investigación**: ¿Cómo contribuir al proceso de visualización gráfica en el Visualizador web del HMI del SCADA Guardián del Alba?

La investigación tiene como **objeto de estudio**: Los procesos de visualización gráfica en el entorno web de los HMI para sistemas SCADA.

Para dar solución al problema planteado se propone como **objetivo general** Desarrollar una propuesta para la visualización de las animaciones para elevar los niveles de interpretación de valores y simulación de los procesos en el Visualizador web del HMI del SCADA Guardián del Alba.

Todo lo anterior permite determinar como **campo de acción**: Visualización de animaciones de objetos gráficos en el entorno web de los HMI.

A partir de este objetivo general se derivan los siguientes **objetivos específicos**:

- Realizar el análisis del estado del arte de las principales tecnologías, metodologías y herramientas a utilizar en la implementación de la solución propuesta.
- Realizar el análisis y diseño del sistema.
- Implementar el sistema.
- Validar el correcto funcionamiento del sistema.

Se tiene como **idea a defender**: Con la integración de la propuesta de visualización de animaciones de objetos gráficos al Visualizador web del HMI, se obtiene dinamismo en los despliegues, mejora la interpretación de variables así como la simulación de los procesos, lo que favorece a la toma de decisiones de los operadores del sistema SCADA Guardián del Alba.

**Para dar cumplimiento a los objetivos de esta investigación se plantean las siguientes tareas investigativas:**

- Selección de las principales tecnologías, metodologías y herramientas a utilizar en la implementación del sistema.

- Elaboración de los fundamentos teóricos-metodológicos para el desarrollo de sistemas SCADA en el entorno web del HMI.
- Establecer relación entre principales proceso de visualización gráfica de los sistemas SCADA para el entorno web del HMI.
- Especificación de los requisitos del sistema.
- Definición de la arquitectura del diseño del sistema.
- Implementación del sistema.
- Selección de las funcionalidades a validar del sistema.<sup>3</sup>

Durante el desarrollo de la investigación se utilizaron en conjunto varios métodos y técnicas de obtención de información.

Como parte de los **métodos empíricos** que se identifica con el conocimiento que se obtiene a partir de la experiencia, de las propiedades y relaciones que se aprecian a través de los sentidos; se encuentran:

- **Test o pruebas:** Con el fin de obtener un resultado satisfactorio según los acuerdos y diseños planificados al inicio del trabajo, que se enfoca en la variación de los objetos gráficos mediante animaciones y para estar al tanto de que las propiedades de estos puede ser alterada en las configuraciones de los ordenadores. De esta forma estar convencidos de que la solución propuesta funcione correctamente.
- **Observación:** Se puso en práctica este método para conocer el funcionamiento existente en los despliegues del SCADA Guardián del ALBA mediante el comportamiento de los dispositivos de campo en las propiedades de los objetos gráficos empleados para la toma de decisiones de los operadores.
- **Consulta bibliográfica:** Empleada para consultar las fuentes de información relacionadas con los tipos de animaciones aplicadas a los objetos gráficos en entornos web de los HMI en sistemas SCADA.

Se utilizaron como **métodos teóricos** para reflejar las relaciones y leyes esenciales del objeto.

- **Análisis histórico–lógico:** Para la comprensión de la evolución de las animaciones asociadas a objetos gráficos en los entornos web de los HMI en sistemas SCADA.
- **Análisis y síntesis:** Con el propósito de consultar bibliografías actualizadas y analizar los elementos más importantes que puedan ser empleados como guía para llevar a cabo la investigación.
- **Hipotético-deductivo:** Al integrar la propuesta de visualización de animaciones de objetos gráficos al Visualizador web del HMI, se obtiene dinamismo en los despliegues, mejora la interpretación de variables así como la simulación de los procesos, lo que favorece a la toma de decisiones de los operadores del sistema SCADA Guardián del ALBA.

La estructura del documento se resume en los siguientes acápites:

**Capítulo 1. Fundamentación teórica:** Abarca conceptos relacionados con las animaciones asociadas a los objetos gráficos en entornos web de los sistemas SCADA. Además, se realiza un estudio de las principales herramientas, metodologías y tecnologías que se utilizan para el desarrollo de visualización de animaciones y se fundamenta la selección de estas para el desarrollo de la solución propuesta.

**Capítulo 2. Análisis y diseño del sistema:** Se lleva a cabo el diseño de la solución propuesta describiendo los patrones de diseño que intervienen en la solución del sistema, así como los requisitos funcionales y no funcionales, teniéndolos en cuenta para la representación mediante los diagramas de modelo de dominio y caso de uso.

**Capítulo 3. Implementación y pruebas:** Se pone de manifiesto el desarrollo y la realización de pruebas a la solución propuesta, teniendo como punto de partida para implementación los estándares de codificación, modelo de implementación y despliegue. Además se describen los posibles aportes realizados al centro CEDIN con la integración del sistema informático implementado al Visualizador web del SCADA GALBA.

---

# CAPÍTULO 1

## FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción

En el presente capítulo se engloban conceptos fundamentales asociados a los HMI en entornos web de los sistemas SCADA. Se analizan las principales tendencias, tecnologías, metodologías y software utilizados en la actualidad para el desarrollo de animaciones de diferentes objetos gráficos en el entorno web. Además se analizan y se fundamenta la selección de las herramientas, metodología y tecnologías para el desarrollo de la solución propuesta.

### 1.2 Sistemas de supervisión, control y adquisición de datos

Los sistemas SCADA, acrónimo de *Supervisory Control and Data Acquisition* (en español, supervisión, control y adquisición de datos), comprenden las soluciones de aplicación que requieren de la captura de información de un proceso o planta industrial, la cual es utilizada para realizar una serie de análisis o estudios con los que se pueden obtener valiosos indicadores de los procesos para mejorar la eficacia del monitoreo y control del proceso y de la toma de decisiones operacionales apropiadas. A su vez se comunican con los dispositivos de campo y controlan el proceso de forma automática desde la pantalla de un ordenador u otra tecnología de comunicación. Entre las funciones principales que posee se encuentran:

- Adquisición de datos: para recoger, procesar y almacenar la información recibida.
- Supervisión: para observar desde un monitor la evolución de las variables de control.
- Control: para modificar la evolución del proceso, actuando bien sobre los reguladores autónomos básicos (consignas, alarmas, menús) directamente sobre el proceso mediante las salidas conectadas. ([Montero, 2014](#))

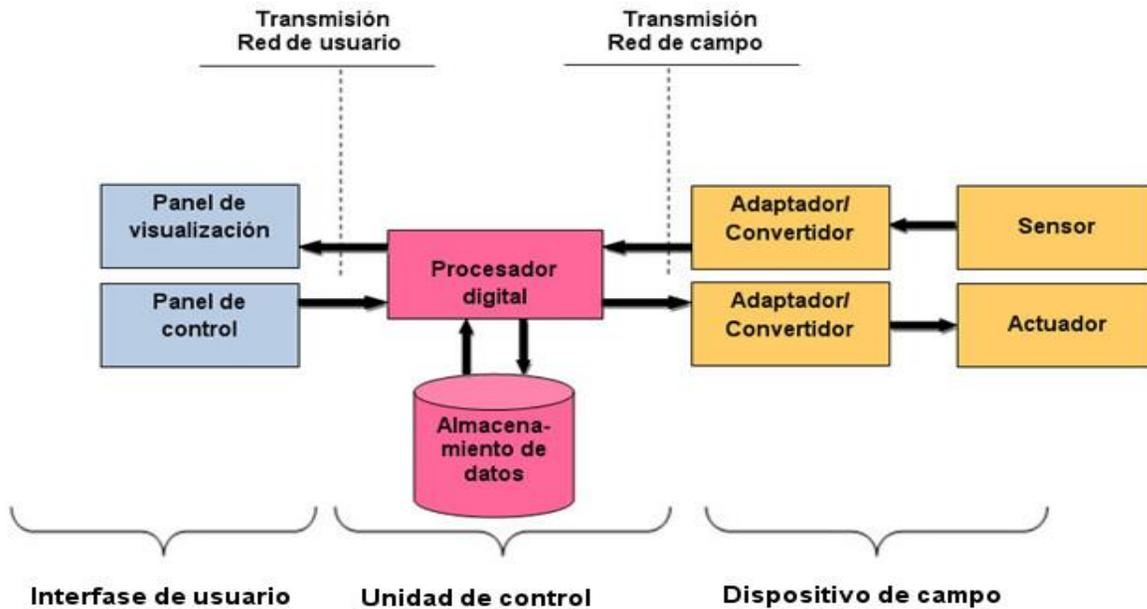


Figura 1: Esquema básico de un sistema de Adquisición, supervisión y control.

## 1.3 Sistema SCADA Guardián del ALBA

Desde el año 2006 se comienza el desarrollo en la UCI de un SCADA en convenio con la Gerencia AIT de la empresa Petróleos de Venezuela S.A (PDVSA), conocido en sus inicios como SCADA Nacional o SCADA PDVSA y a partir del 2008 en que fue presentado en la “Cumbre del ALBA” se comienza a conocer como SCADA GALBA, donde se proyectó una futura instalación del sistema en los países integrantes de esta organización. El software, que se da como solución es realizado en cooperación de distintos equipos de desarrollo de la UCI, DST-AIT PDVSA, Centro de Desarrollo de Automática Integral (CEDAI), Universidad Central “Marta Abreu” de Las Villas (UCLV), Instituto Superior Minero Metalúrgico de Moa (ISMMM), Universidad de los Andes (ULA), DBAccess<sup>1</sup>, IntelCom<sup>2</sup> y la empresa de Ingeniería de Software y Calidad Aplicada (Isca). El SCADA “Guardián del ALBA” es un software que integra las funcionalidades de alto nivel que permite la solución de aplicaciones de supervisión y control de procesos,

<sup>1</sup> Organización latinoamericana de proyección global proveedora de servicios de Tecnología de la Información.

<sup>2</sup> Empresa de ingeniería prestadora de servicios en soluciones integrales de telecomunicaciones en Venezuela.

utilizando para ello una arquitectura distribuida de módulos que permite escalar a aplicaciones de gran envergadura.

## 1.3.1 Interfaz Hombre-Máquina

Los sistemas HMI se pueden pensar como una “ventana<sup>3</sup>” de un proceso. Esta ventana puede estar en dispositivos especiales como paneles de operador o una computadora. Estos sistemas, en computadoras se les conoce también como software de interfaz hombre máquina (por sus siglas en inglés *Human Machine Interface*) o de monitoreo y control de supervisión. Las señales de los procesos son conducidas al mismo por medio de dispositivos como tarjetas de entrada/salida en la computadora, PLC (Controladores lógicos programables), RTU (Unidades remotas de I/O) o DRIVE (Variadores de velocidad de motores). Todos estos dispositivos deben tener una comunicación que entienda el HMI.

El HMI del sistema SCADA Guardián de ALBA está compuesto por dos ambientes de trabajo, el de Configuración o Editor que permite administrar la configuración del proyecto y sus recursos; y el de Ejecución o Visualizador que permite ejercer control y supervisar los recursos que han sido configurados con anterioridad en el ambiente Editor. A continuación se describen algunas de las funcionalidades que permiten estos ambientes:

- El Editor permite crear un proyecto y administrar la configuración de sus recursos (Seguridad, Histórico, Adquisición y HMI). Posibilita la creación y el diseño de los despliegues que se visualizarán en el ambiente de Ejecución utilizando para ello los componentes necesarios para representar la información al operador que son asociados a su vez con los puntos(valores configurados para adquirir información del campo) que representan, dando la posibilidad el sistema de modificar todos los recursos que se definan. Permite la creación y edición de reportes, la definición de los usuarios bajo un esquema de privilegios identificado y la configuración de los grupos donde se almacenarán los datos del sistema.

---

<sup>3</sup> Parte fundamental de la interfaz gráfica de la computadora, especialmente con sistemas operativos que permiten el multiprocesamiento.

- El Visualizador permite adquirir datos del campo en forma continua, almacenar y mostrar información en forma de sumarios y gráfica, ejercer control y supervisión de los despliegues configurados en el ambiente de configuración, así como visualizar las alarmas que se detecten y afecten el comportamiento del proceso supervisado. Permite la autenticación de usuarios en el sistema. Crea una sesión para cada usuario autenticado, con sus respectivos privilegios sobre la herramienta y maneja los tiempos de expiración de la sesión. ([Mandado, 2009](#))

Como se observa, los visualizadores de hoy en día ofrecen a los operadores las más sofisticadas técnicas de supervisión y control, brindando u ofreciendo la posibilidad de visualizar las operaciones que se realizan en una planta industrial con los sinópticos gráficos, haciendo sumamente intuitivo la operación de una planta específica.

## 1.3.2 Módulo de HMI web del sistema SCADA Guardián de ALBA

El módulo de HMI web del sistema SCADA Guardián de ALBA está compuesto por un cliente<sup>4</sup> web y un servidor HMI web, que se comunican haciendo uso del protocolo de comunicación WebSocket<sup>5</sup>. Éste módulo brinda las funcionalidad convencionales de un SCADA, permite a los usuarios autorizados estar en contacto directo con el sistema, realizar la supervisión y la adquisición de datos. A su vez es capaz de conectarse con la capa de comunicaciones del SCADA Guardián del ALBA, a través del servidor HMI y adquirir la información de los puntos, despliegues, las últimas 5 alarmas emitidas por el sistema, el sumario de puntos, el sumario de dispositivos, el sumario de subcanales, el sumario de alarmas, los detalles de un punto, detalles de los dispositivos y detalles de subcanales para visualizarla. En el cliente web se representa la información adquirida mediante componentes gráficos, sin embargo la interpretación de la información, y el estado que toman los procesos son limitados por los inconvenientes de los deficientes de la interpretación de valores y simulación de los procesos.

---

<sup>4</sup> Nombre que reciben aquellas computadoras que forman parte de una red y que reciben la información y el software de otro principal llamado servidor.

<sup>5</sup> Tecnología que proporciona un canal de comunicación bidireccional, Está diseñada para ser implementada en navegadores y servidores web, pero puede utilizarse por cualquier aplicación cliente/servidor.

## 1.3.3 Visualización gráfica en el entorno web del HMI

A cada objeto gráfico o componente se le pueden asociar animaciones o parámetros con el objetivo de obtener un efecto visual deseado en función de variables o eventos asignados. Un objeto, según sus características se puede animar de varias formas ya sea usándolo como entrada de datos, como una barra de desplazamiento, como un pulsador que realizará una acción, cambiando el color de sus líneas, cambiando su tamaño, desplazándolo, cambiando su color de relleno o cambiando el color de un texto. ([ISEF, 2010](#))

En dependencia del tipo de animaciones ya sean básicas o avanzadas pueden provocar que el objeto cambie de apariencia reflejando cambios que ocurren en los valores almacenados en la base de datos. La información visualizada correctamente permite que los operadores puedan tomar decisiones informadas y oportunas en el momento adecuado, realizar acciones correctivas para reducir costos e incidentes operacionales, y mejorar la productividad en toda la operación.

Las animaciones básicas incluyen:

- Animación de color: Cada valor o rango de valores de objeto se asociará con un color determinado, permitiendo efectuar un cambio en este.
- Animación de texto: Por medio de esta animación es posible mostrar, en diversos formatos, la información de los valores de un objeto.
- Animación de imagen: Muestra una o varias imágenes en un componente en dependencia de los valores que tomen los objetos.

Las animaciones avanzadas incluyen:

- Rotación: Esta animación permite cambiar el ángulo de rotación del objeto para que este pueda girar hacia la derecha o hacia la izquierda.
- Traslación: Se basa en desplazar el objeto horizontal o verticalmente.
- Visibilidad: Esta animación permita ocultar o mostrar un objeto.

- Escalado: Esta animación permite que un objeto pueda variar su forma original, logrando un aumento en la misma.

## 1.3.4 Tecnologías para la visualización gráfica en el entorno web del HMI de los sistemas SCADA.

- **SCADA Movicon:** Para la visualización gráfica emplea la tecnología "XML<sup>6</sup>-dentro", la cual ofrece una única plataforma para proyectar estaciones de supervisión SCADA de grandes plantas, también contribuye a reducir costos manteniendo en la empresa un solo software de visualización y control para todas las necesidades y los proyectistas pueden reducir drásticamente los tiempos de desarrollo, y los usuarios pueden disponer de soluciones potentes, abiertas, flexibles y de simple mantenimiento. ([AUTC, 2015](#))
- **SCADA InTouch:** Para la visualización gráfica emplea la tecnología: ArchestrA Graphics y Objetos de Aplicaciones:—los poderosos componentes de objetos dentro de la System Platform 3.0 resumen los gráficos ArchestrA InTouch 10.0 junto con lógica de la aplicación, I/O, almacenamiento de historia y datos, permitiendo ingeniería de soluciones drag and drop. Esto implica un increíble ahorro de tiempo, lo que también mejora la consistencia y mantenimiento de las aplicaciones. ([Intouch, 2015](#))

Al realizar el estudio acerca de las tecnologías empleadas para la visualización gráfica en los sistemas SCADA Movicon e InTouch se pudo apreciar que utilizan librerías de animaciones para la visualización de objetos o imágenes en su interfaz gráfica; aunque hay que destacar que ambos son software privativos por lo que no muestran el código fuente empleado para la construcción de los mismos. Y aunque las tecnologías que emplean son muy potentes con diversas ventajas también poseen ciertas dificultades que resultan inconvenientes para lograr dinamismo en el Visualizador web de HMI del SCADA Guardián del ALBA, y se debe a que requieren de constante mantenimiento ya que estas tecnologías están ligadas a herramientas que no precisan de una fuerte programación, lo que trae consigo altos presupuestos.

---

<sup>6</sup> Lenguaje de marcación extendido. Tecnología que permite un mejor uso de los documento en la Web de Internet, principalmente para el comercio electrónico.

## 1.4 Metodología de desarrollo de software

Una metodología de desarrollo de software es un conjunto de técnicas, herramientas y procedimientos que le permiten a los desarrolladores definir las actividades a realizar para la construcción de un nuevo producto de software.

Existen numerosas propuestas de metodologías, por una parte están las metodologías tradicionales, que se centran en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas, documentación detallada y notaciones para el modelado donde el resultado final sería un proceso de desarrollo más complejo. Entre estas se pueden encontrar: RUP (*Rational Unified Procces*) y MSF (*Microsoft Solution Framework*). Por otra parte las metodologías ágiles le dan un mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con interacciones muy cortas, centrándose en el factor humano y en el producto de software. Entre estas la más destacadas hasta el momento son: XP (*Extreme Programming*), OpenUp y Proceso Unificado Ágil (AUP).

Teniendo en cuenta lo planteado con anterioridad se decide utilizar para el desarrollo del sistema informático la metodología AUP debido a que la UCI actualmente cuenta con 14 centros productivos, cada uno de estos centros se dedica al desarrollo de software y/o servicios empleando el uso de diferentes metodologías de desarrollo entre robustas y ágiles. A pesar de la variedad de metodologías usadas, se ha comprobado que muy pocos proyectos la aplican en su totalidad. Las diferencias entre estas metodologías no radica únicamente en los productos de trabajos que proponen o en sus roles, sino en su forma de planificar el proyecto y realizar las estimaciones del tiempo. Factor determinante en la culminación exitosa de todo desarrollo de software. Para lograr erradicar los problemas detectados, se decide escoger una metodología para ser adaptada a lo que ya la Universidad ha estado proponiendo como ciclo de vida de los proyectos, sin alejarse de lo que hasta el momento se ha trabajado e introducir la menor cantidad de cambios posibles.

Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos) exigiéndose así que el proceso sea

configurable. Se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI.

El Proceso Unificado Ágil de Scott Ambler o *Agile Unified Process* en inglés es una versión simplificada del RUP. Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP.

Con la adaptación de AUP que se propone para la actividad productiva de la UCI se logra estandarizar el proceso de desarrollo de software, dando cumplimiento además a las buenas prácticas que define CMMI-DEV v1.3. Se logra un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajos. Se redujo a 1 la cantidad de metodologías que se usaban y de más de 20 roles en total que se definían se redujeron a 11. ([Informaticas, 2014](#))

## 1.5 Tecnologías y herramientas a utilizar.

Las herramientas y tecnologías escogidas para el desarrollo de un proyecto deben ser seleccionadas cautelosamente, ya que pueden suponer el fracaso de éste o pueden aumentar su complejidad, para lo cual se deben conocer cuáles son las distintas alternativas y las necesidades del proyecto.

A continuación se abordarán algunas de las tecnologías y herramientas que mantienen un frecuente empleo en el desarrollo de visualización de animaciones para un Visualizador web, de las cuales se realizará una selección para el desarrollo del sistema propuesto.

### 1.5.1 Herramientas CASE

Las herramientas CASE (*Computer Aided Software Engineering*) fueron diseñadas para aumentar la productividad en el desarrollo de software, reduciendo el costo de los mismos en términos de tiempo y dinero. Corresponden a diversas aplicaciones informáticas, que incluyen un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de un proyecto a desarrollar. Además facilitan el uso de las distintas metodologías propias de la ingeniería del software incluyendo AUP.

Se seleccionó Visual Paradigm 8.0 porque es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Para modelar los diagramas de la solución propuesta por su estabilidad de ejecución en diferentes sistemas operativos y la facilidad de abrir y trabajar con un modelo UML utilizando el mismo programa sin importar el sistema operativo y sin afectar en absoluto el trabajo hecho. Este software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Es una herramienta libre y multiplataforma. ([Paradigm, 2014](#))

## 1.5.2 Lenguaje de modelado

Son un conjunto estandarizado de símbolos que facilita la modelación de un diseño de software. Se usan en combinación con una metodología de desarrollo de software para avanzar de una especificación inicial a un plan de implementación. Para la especificación y documentación del producto el lenguaje de modelado es esencial, ya que constituye un pilar para el mantenimiento del software, su posterior actualización y soporte.

Se decide utilizar el Lenguaje de Modelado Unificado (UML), debido a que se emplea para especificar, visualizar, construir y documentar los artefactos de la solución propuesta, es un lenguaje estándar, fácil de aprender, y ofrece una amplia variedad de diagramas para mostrar la solución propuesta desde varias perspectivas. Está especialmente diseñado para apoyar un estilo de desarrollo iterativo e incremental y presenta tecnología orientada a objetos. Ayuda al usuario a entender la realidad de la tecnología y la posibilidad de que reflexione antes de invertir y gastar grandes cantidades en proyectos que no estén seguros en su desarrollo. Tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, hasta la implementación y configuración con los diagramas de despliegue. ([UML, 2013](#))

## 1.5.3 Entorno de desarrollo

# Capítulo 1: Fundamentación teórica

---

Un entorno de desarrollo integrado, conocido también como IDE (*Integrated Development Environment*) por sus siglas en inglés, es un programa informático compuesto por un conjunto de herramientas de programación que facilitan el desarrollo de aplicaciones web. Un IDE puede denominarse como un entorno de programación, esto significa que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Python, Java, PHP, C#, Delphi, Visual Basic.

Se propone como IDE de desarrollo Brackets porque es una excelente herramienta de diseño y desarrollo web de código abierto que ha sido creado con tecnologías web. Entre sus características principales se destacan que es un editor de código sencillo y ligero, sin paneles de opciones ni barras de iconos que sobrecarguen la interfaz, permite editar los estilos CSS directamente en el documento HTML sin necesidad de abrir el archivo, brinda sugerencias de código para las etiquetas y atributos HTML, los selectores y propiedades de las hojas de estilo CSS y las palabras reservadas, variables locales, argumentos y nombres de propiedades de JavaScript. Es modular y extensible mediante extensiones. Brackets integra el navegador con un editor web, por lo que se puede trabajar directamente en el editor de código y ver los resultados simultáneamente en el navegador; con la gran ventaja de no tener que recordar los cambios y luego regresar al editor para ingresarlos manualmente. ([Medios, 2015](#))

## 1.5.4 Lenguajes de programación

Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes. Tiene la capacidad de especificar, de forma precisa, cuáles son los datos que debe trabajar un equipo informático, de qué modo deben ser conservados o transferidos dichos datos y qué instrucciones debe poner en marcha la computadora ante ciertas circunstancias.

### 1.5.4.1 JavaScript

Para interactuar con el Visualizador web del HMI se propone el lenguaje JavaScript porque es un lenguaje libre por lo que cualquier persona puede utilizarlo sin comprar una licencia. JavaScript es un lenguaje de

# Capítulo 1: Fundamentación teórica

---

programación interpretado y se define como orientado a objetos, basado en prototipos. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. Además se emplea para realizar tareas y operaciones en el marco de la aplicación únicamente cliente, sin acceso a funciones del servidor. ([Flanagan, 2002](#))

## 1.5.4.2 HTML

*Hyper Text Markup Language* es un lenguaje de hipertexto que permite escribir textos de forma estructurada. Compuesto por etiquetas que marcan el inicio y el fin de cada elemento del documento permitiendo describir las páginas web. Los documentos HTML deben tener la extensión html o htm, para que puedan ser visualizados en los navegadores, los que se encargan de interpretar ese código y mostrar a los usuarios las páginas resultantes.

HTML consta de varios componentes entre los que están los elementos y sus atributos, tipos de dato y la declaración de tipo de documento. Describe la estructura y el contenido en forma de texto empleando etiquetas (<Body>...</Body> o <P>...</P>) para ello. Debido a que es un estándar independiente de fabricantes y marcas puede ser interpretado por todos los navegadores, siendo sus aplicaciones muy rápidas, con mucho desarrollo y ocupan poco espacio. ([W3Schools, 2015](#))

## 1.5.4.3 CSS

Es un lenguaje de hojas de estilos creados para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML7, siendo la mejor forma de separar los contenidos y su presentación y muy necesario para crear páginas web complejas. El separar la definición de los contenidos y la definición de su aspecto, mejora la accesibilidad del documento, reduciendo la complejidad de su mantenimiento y garantizando que sus páginas web tengan un estilo coherente en todo el sitio. ([w3schools, 2015](#))

## 1.5.5 Marco de trabajo para el desarrollo de aplicaciones web

---

<sup>7</sup> Extensible HyperText Markup Language (Lenguaje de marcado de Hipertexto Extensible)

# Capítulo 1: *Fundamentación teórica*

---

En el desarrollo de software un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Teniendo como base las habilidades que se ponen en práctica en los marcos de trabajo para desarrollar aplicaciones web, se decide utilizar dos frameworks para la implementación de dos propuestas, con la finalidad de garantizar la calidad de las animaciones en los objetos gráficos, y brindar la posibilidad de emplear la propuesta de visualización más compatible con el sistema al que se integrará, o ambas en caso conveniente.

## 1.5.5.1 JQuery

Se decide utilizar JQuery para el desarrollo de la solución propuesta porque es una biblioteca de JavaScript, creada inicialmente por John Resig<sup>8</sup>, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. JQuery es la biblioteca de JavaScript más utilizada.

Es un framework para javascript que permite desarrollar sitios de manera más dinámica y fácil; ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. Entre sus características principales se encuentra la manipulación de la hoja de estilos CSS, los efectos y animaciones personalizadas, es compatible con los navegadores Mozilla Firefox 2.0 o superior, Internet Explorer 6 o superior, Safari 3 o superior, Opera 10.6 o superior y Google Chrome 8 o superior. A su vez permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX. ([Resig, 2014](#))

## 1.5.5.2 AngularJs

Se decide utilizar AngularJs para el desarrollo de la solución propuesta porque contiene un conjunto de librerías útiles para el desarrollo de aplicaciones web y propone una serie de patrones de diseño para

---

<sup>8</sup> Programador y empresario, conocido por ser el Evangelista de JavaScript de la Fundación Mozilla, creador y líder de desarrollo de la biblioteca JavaScript jQuery, autor de un libro técnico sobre JavaScript y de uno de los blogs más leídos, con una media de 13.000 suscriptores.

llevarlas a cabo. Este framework brinda la posibilidad que los programadores mejoren el HTML que se lleve a cabo. Que puedan producir un HTML que de manera declarativa, genere aplicaciones que sean fáciles de entender, incluso para alguien que no tiene conocimientos profundos de informática. El objetivo es producir un HTML altamente semántico, es decir, que cuando lo leas entiendas de manera clara qué es lo que hace o para qué sirve cada cosa. ([Desarrolloweb, 2015](#))

## 1.6 Conclusiones parciales

En el presente capítulo se describieron los sistemas SCADA, específicamente el sistema SCADA Guardián del ALBA, el módulo de HMI web del sistema SCADA Guardián de ALBA, el proceso de visualización gráfica en el entorno web del HMI de los sistemas SCADA y las tecnologías utilizadas para lograr dicho proceso de visualización.

Después de haber analizado las metodologías, herramientas y tecnologías que hoy en día enriquecen a la informática se ha decidido implementar dos propuestas para la visualización de animaciones para objetos gráficos en el Visualizador web del HMI del SCADA GALBA, utilizando la herramienta Brackets, por las posibilidades que brinda. Para el lenguaje de programación JavaScript, a partir del cual se derivan los frameworks JQuery y AngularJs, que han sido seleccionados para realizar la solución propuesta que será integrada al HMI para entornos web en sistemas SCADA; como metodología de desarrollo se ha decidido utilizar AUP por la abundante documentación y el establecimiento de su empleo en los proyectos productivos de la Universidad de las Ciencias Informáticas.

## CAPÍTULO 2

### ANÁLISIS Y DISEÑO DEL SISTEMA

#### 2.1 Introducción

En el presente capítulo se reflejan las actividades realizadas en los procesos de análisis y diseño de la solución propuesta; proceso que será guiado por la metodología de desarrollo AUP. En el mismo se exponen los artefactos más importantes que describen el flujo normal de eventos que ocurren en el sistema tales como la descripción del modelo de dominio, así como los diagramas correspondientes al sistema, incluyendo la especificación de los requisitos, puesto que estos contribuyen a la base para la realización del modelo de casos de uso.

#### 2.2 Modelo de dominio

Un paso esencial en la construcción de un software es descomponer el problema, conformando un modelo conceptual compuesto por conceptos u objetos significativos. El modelo de dominio es la representación visual de los conceptos u objetos más importantes de un negocio y se utiliza con el objetivo de contribuir a la comprensión de los requisitos del sistema. ([Sommerville, 2005](#)). El modelo de dominio que se muestra a continuación describe el flujo de acciones de la solución propuesta:

# Capítulo 2: Análisis y Diseño del Sistema

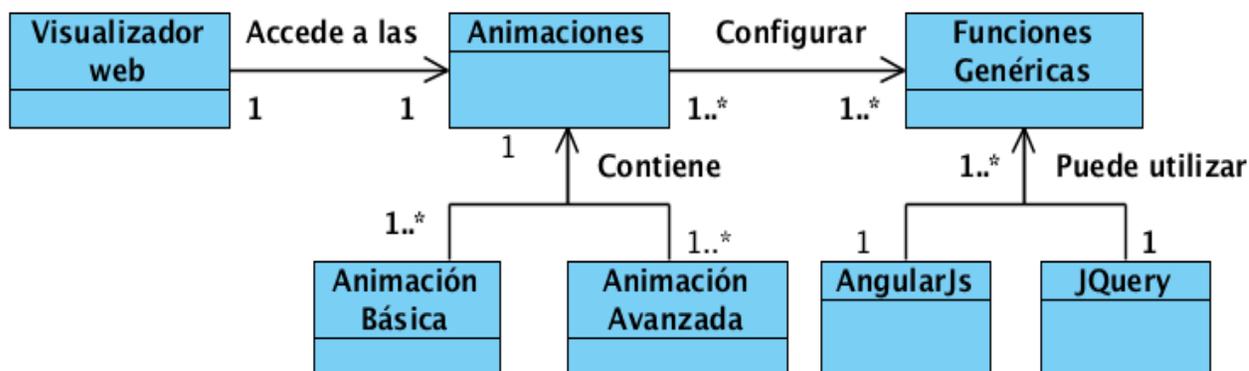


Figura 2: Modelo conceptual del sistema.

El modelo conceptual propuesto representa los conceptos del mundo real y las relaciones entre ellos. En este caso el flujo de acciones se describe de la siguiente forma:

## Descripción de los conceptos del dominio

Tabla 1: Componentes del dominio

Conceptos del dominio	Descripción
Visualizador Web	Es el encargado de mostrar la variación de las propiedades de los objetos gráficos mediante el conjunto de animaciones definidas y asignadas a los objetos.
Animaciones	Compuesta por un conjunto de animaciones creadas con funciones genéricas, que podrán ser reutilizadas para diferentes objetos, dándoles apariencia en tiempo real.
Funciones Genéricas	Configuran todas las animaciones, donde se podrá variar los parámetros de las propiedades de las mismas.
Animación Básica	Contiene las funciones genéricas con las animaciones de

	color, texto e imagen.
Animación Avanzada	Contiene las funciones genéricas con las animaciones de traslación, rotación, escalado y visibilidad.
AngularJs	Dinamiza documentos HTML permitiendo crear la función correspondiente a la animación solicitada.
JQuery	Permite la creación de animaciones para Visualizadores web, para simplificar el uso de animaciones, manejo de eventos mediante la creación de funciones.

### 2.4 Especificación de requisitos

Conocer los requisitos del sistema que se va a desarrollar es el primer paso en el desarrollo del mismo, de esta manera se reducirá el número de cambios que habrá que realizar en el producto. Los requisitos son una descripción de las necesidades o deseos de un producto. La meta primaria de esta fase es identificar y documentar lo que en realidad se necesita. ([Mario, 2010](#))

A partir del modelo del dominio presentado se realizó el levantamiento de los requisitos, los cuales están divididos en dos tipos: los funcionales y los no funcionales. A continuación se especifican los requisitos definidos para el desarrollo del sistema propuesto.

#### 2.4.1 Requisitos funcionales

Los requisitos funcionales (RF) del sistema propuesto se identificaron de acuerdo a las capacidades o condiciones que este debe cumplir. Los mismos expresan una especificación detallada de cómo reacciona el sistema a una entrada particular y cómo se comporta ante situaciones particulares, es decir, sus entradas, salidas y excepciones. Teniendo en cuenta las funcionalidades que el sistema debe proveer se identificaron 7 RF los cuales se definen a continuación:

➤ **RF1 Visualizar animación de color.**

Esta funcionalidad permitirá que la animación del color se interpole entre valores discretos de

colores para valores intermedios de la variable asociada al objeto.

➤ **RF2 Visualizar animación de texto.**

Esta funcionalidad permitirá mostrar, en diversos formatos, la información de los valores de las variables.

➤ **RF3 Visualizar animación de imagen.**

Esta funcionalidad permite que un objeto en función del valor que tome una variable pueda mostrar el intercambio de imágenes asociadas a una condición determinada.

➤ **RF4 Visualizar animación de rotación.**

Esta funcionalidad permitirá cambiar el ángulo de rotación (girar desde 0 hasta 360 grados) del objeto en función de una variable.

➤ **RF5 Visualizar animación de traslación.**

Esta funcionalidad permitirá desplazar el objeto en el eje horizontal, diagonal o vertical del despliegue, en función del valor de una variable asociada al movimiento. Cuando la variable adquiera valores dentro de los puntos para los cuales fue definido el movimiento, el objeto se trasladará desde un punto inicial hasta uno final.

➤ **RF6 Visualizar animación de visibilidad.**

Esta funcionalidad permitirá que el objeto se oculte o se visualice cuando la variable asociada cumpla con una condición seleccionada.

➤ **RF7 Visualizar animación de escalado.**

Esta funcionalidad permitirá escalar (aumentar o disminuir de tamaño) un objeto en función de los valores que adquiera una variable.

## 2.4.2 Requisitos no funcionales

Los requisitos no funcionales (RNF) son las propiedades o cualidades que el sistema debe tener, para que sea un producto atractivo, usable, rápido y confiable. Existen múltiples categorías para clasificar los requisitos no funcionales, siendo las siguientes las más representativas acumulando un total de 3 RNF para el sistema propuesto.

➤ **RNF1: Usabilidad:**

**RNF1.1:** El sistema debe estar disponible para su posterior reutilización en cualquier Visualizador web de un SCADA

➤ **RNF2: Eficiencia:**

**RNF2.1:** El sistema debe permitir que los componentes gráficos puedan utilizar una misma animación, sin tener que reimplementar la funcionalidad específica para cada uno de ellos.

➤ **RNF3: Compatibilidad:**

**RNF3.1:** El sistema debe ser compatible con los navegadores Firefox en su versión 13 o superior, Chrome en su versión 8 o superior, Opera en su versión 10 o superior, Safari en su versión 5 o superior e Internet Explorer en su versión 10 o superior.

## 2.5 Modelo de casos de uso del sistema

El modelo de casos de uso representa un esquema que recoge las funcionalidades del sistema que se automatizan y determina el uso que tendrá el mismo desde el punto de vista del usuario. ([Mario, 2010](#))

Para el sistema propuesto el modelo de casos de uso está compuesto por los actores que intervienen, los requisitos funcionales principales agrupados en casos de uso y la relación que existe entre estos y los actores. Como el sistema propuesto es una propuesta de visualización con animaciones básicas y avanzadas, incluirá varias funciones genéricas con el fin de que al ser invocadas proporcionen dinamismo en los diferentes objetos gráficos de la interfaz del Visualizador web al que será integrada la solución propuesta.

### 2.5.1 Descripción de los actores sistema

Los actores representan un tipo de usuario del sistema y el conjunto de casos de uso al que estos tienen acceso define su rol global en el sistema ([Mario, 2010](#)). En la solución propuesta se identificó como actor el siguiente:

# Capítulo 2: Análisis y Diseño del Sistema

Tabla 2: Actor del sistema.

Actor	Descripción
Visualizador web	Su responsabilidad es configurar los parámetros de las animaciones ya sean básicas o avanzadas para su posterior visualización.

## 2.5.2 Definición de los casos de uso del sistema

Un caso de uso (CU) se representa mediante un óvalo y proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico (Mario, 2010). Teniendo en cuenta lo planteado se decide agrupar los 7 RF en 7 CU. A continuación se puede apreciar el diagrama de CU del sistema.

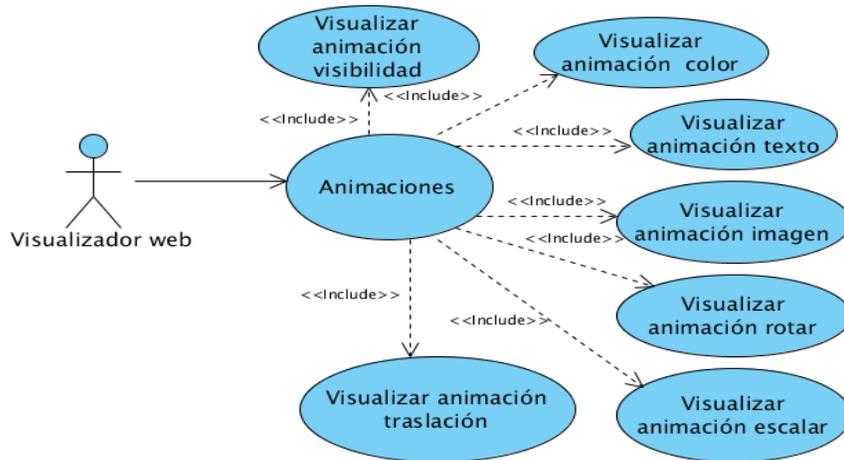


Figura 3: Diagrama de Caso de Uso del Sistema.

En el diagrama de CU del sistema presentado se puede apreciar que el Visualizador web tiene integrado una propuesta de visualización de animaciones la cual brinda diferentes funciones genéricas desarrolladas con JQuery y AngularJs, que incluyen las funcionalidades de visualizar animación de color, visualizar animación de texto, visualizar animación de imagen, visualizar animación de rotación, visualizar animación de traslación, visualizar animación de visibilidad y visualizar animación de escalado.

## 2.5.2 Descripción de los casos de usos del sistema

Especificar los casos de uso del sistema permite entender las necesidades del cliente y proveer una visión que posibilite emprender el desarrollo del mismo ([Mario, 2010](#)). A continuación se describen los CU de la solución propuesta.

**Tabla 3: Descripción del caso de uso: Visualizar animación de visibilidad.**

<b>Caso de uso</b>	Visualizar Animación de visibilidad
<b>Actores</b>	Visualizador web
<b>Propósito</b>	El CU se realiza con el objetivo de mostrar u ocultar un objeto dado un evento realizado.
<b>Resumen</b>	El CU se lleva a cabo con la condición de que el objeto se pondrá visible cuando la variable asociada cumpla con la condición seleccionada.
<b>Precondiciones</b>	Debe de habilitar la animación invocando la función de visibilidad.
<b>Referencias</b>	RF1
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>

## Capítulo 2: Análisis y Diseño del Sistema

<p>1. El CU se inicia cuando el visualizador web invoca la función genérica para la visibilidad.</p>	<p>2. El sistema requiere los parámetros de configuración de la función genérica de visibilidad.</p>
<p>3. El Visualizador web selecciona la variable o expresión que va a definir la visibilidad.</p> <p>a) Seleccionar la condición de comparación.</p> <p>b) Seleccionar el valor con el cual se va a comparar el valor de la variable.</p> <p>c) Seleccionar la condición de comparación.</p>	<p>4. El CU finaliza mostrando u ocultando el objeto gráfico en la interfaz del Visualizador web.</p>
<p><b>Pos condiciones</b></p>	<p>El sistema mantiene visible el objeto si la condición seleccionada esta en correspondencia con la variable asociada.</p>

Tabla 4: Descripción del caso de uso: Visualizar animación de texto.

<b>Caso de uso</b>	Visualizar animación de texto
<b>Actores</b>	Visualizador web
<b>Propósito</b>	El CU se realiza con el objetivo de variar el formato de un texto asociado a un objeto.
<b>Resumen</b>	El CU se lleva a cabo cuando se requiere mostrar en diversos formatos, la información de los valores de las variables.

## Capítulo 2: Análisis y Diseño del Sistema

<b>Precondiciones</b>	Debe de habilitar la animación invocando la función de texto.
<b>Referencias</b>	RF2
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
1. El CU se inicia cuando el visualizador web invoca la función genérica para la animación de texto.	2. El sistema requiere los parámetros de configuración de la función genérica de texto.
3. El Visualizador web selecciona si se desea que se muestre el valor de la variable o un mensaje asociado al valor.  a) Seleccionar la variable o expresión cuyo valor, o resultado en caso de la expresión, será mostrado como texto.  b) Asignar formato en que se mostrará el texto, este formato puede utilizarse para representar los números con determinadas cifras decimales.	4. El CU finaliza mostrando los formatos definidos para el texto asociado al objeto.
<b>Pos condiciones</b>	El sistema mantiene el formato del objeto en función a los valores asociados a la configuración del mismo.

Tabla 5: Descripción del caso de uso: Visualizar animación de imagen.

## Capítulo 2: Análisis y Diseño del Sistema

<b>Caso de uso</b>	Visualizar animación de imagen
<b>Actores</b>	Visualizador web
<b>Propósito</b>	El CU se realiza con el objetivo de reemplazar un objeto por otro.
<b>Resumen</b>	El CU se lleva a cabo cuando se desea cambiar un objeto por otro dado un evento
<b>Precondiciones</b>	Debe de habilitar la animación invocando la función de imagen.
<b>Referencias</b>	RF3
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
1. El CU se inicia cuando el visualizador web invoca la función genérica para la animación de imagen.	2. El sistema requiere los parámetros de configuración de la función genérica de imagen.
3. El Visualizador web selecciona la variable que va a definir la animación. a) Indicar el valor del nuevo objeto. b) Seleccionar la variable asociada al objeto que será reemplazado. c) Asignar valor del nuevo objeto a la variable del objeto a reemplazar.	4. El CU finaliza mostrando el cambio del objeto gráfico en la interfaz del Visualizador web.
<b>Pos condiciones</b>	El sistema mantiene el cambio del objeto en función del valor de la variable asociada al evento.

## Capítulo 2: Análisis y Diseño del Sistema

Tabla 6: Descripción del caso de uso: Visualizar animación de rotación.

<b>Caso de uso</b>	Visualizar animación rotación
<b>Actores</b>	Visualizador web
<b>Propósito</b>	El CU se realiza con el objetivo de rotar un objeto para variar el ángulo de ubicación del mismo.
<b>Resumen</b>	El CU se lleva a cabo cuando se desea cambiar el ángulo de rotación del objeto en función de una variable.
<b>Precondiciones</b>	Debe de habilitar la animación invocando la función de rotación.
<b>Referencias</b>	RF4
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
1. El CU se inicia cuando el visualizador web invoca la función genérica para la rotación.	2. El sistema requiere los parámetros de configuración de la función genérica de rotación.

## Capítulo 2: Análisis y Diseño del Sistema

<p><b>3.</b> El Visualizador web selecciona la variable o expresión que va a definir la rotación.</p> <p>a) Indicar el valor de la variable que se corresponde con el valor inicial del intervalo (en grados) de rotación.</p> <p>b) Indicar el valor de la variable que se corresponde con el valor final del intervalo (en grados) de rotación.</p> <p>c) Se debe dar la posibilidad de configurar un centro que permita la rotación del objeto respecto al dicho centro. Para ello se debe:</p> <p>Habilitar un punto para centro de rotación.</p> <p>Seleccionar valor de coordenada X.</p> <p>Seleccionar valor de coordenada Y.</p>	<p><b>4.</b> El CU finaliza mostrando el dinamismo del objeto gráfico rotándolo según la posición deseada.</p>
<p><b>Pos condiciones</b></p>	<p>El sistema mantiene la rotación del objeto en función de los valores de las variables asociadas a los ángulos de las posiciones.</p>

Tabla 7: Descripción del caso de uso: Visualizar animación de traslación.

<b>Nombre del Caso de Uso</b>	Visualizar animación de traslación
<b>Actor</b>	Visualizador web.
<b>Propósito</b>	Permitir que el Visualizador web pueda configurar los parámetros para poder desplazar un objeto horizontal,

## Capítulo 2: Análisis y Diseño del Sistema

	verticalmente o diagonal.
<b>Resumen</b>	El CU se inicia cuando el Visualizador web configura los parámetros de la animación de traslación, la variable asociada toma valores dentro de los puntos para los cuales fue definido el movimiento y finalmente el objeto se irá moviendo desde el punto inicial hasta el final.
<b>Precondiciones</b>	Se debe habilitar la animación invocando la función de traslación.
<b>Referencias</b>	RF5
<b>Prioridad</b>	Crítico
<b>Flujo Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El CU se inicia cuando el visualizador web invoca la función genérica para la traslación.	2. El sistema requiere los parámetros de configuración de la función genérica de traslación.

## Capítulo 2: Análisis y Diseño del Sistema

<p><b>3.</b> El Visualizador web define el movimiento, invocando dicha función.</p> <p>a) Seleccionar el objeto o imagen al que se le aplicará la animación de traslación.</p> <p>b) Indicar el valor de la variable para el cual se aplicará el desplazamiento desde su posición actual hasta el punto final en el ejeX.</p> <p>c) Indicar el valor de la variable para el cual se aplicará el desplazamiento desde su posición actual hasta el punto final en el ejeY.</p> <p>d) Indicar la velocidad con la cual se efectuara el movimiento en segundos.</p>	<p><b>4.</b> El CU finaliza mostrando el dinamismo del objeto gráfico desplazándose a la posición deseada en la interfaz del Visualizador web.</p>
<p><b>Pos condiciones</b></p>	<p>El sistema mantiene la traslación del objeto en función del valor de una variable asociada al movimiento.</p>

**Tabla 8: Descripción del caso de uso: Visualizar animación de color.**

<b>Caso de uso</b>	Visualizar animación de color
<b>Actores</b>	Visualizador web
<b>Propósito</b>	El CU se realiza con el objetivo de cambiarle el color a un objeto determinado
<b>Resumen</b>	El CU se lleva a cabo cuando se decide cambiar el color de un objeto en

## Capítulo 2: Análisis y Diseño del Sistema

	función de los valores de una variable.
<b>Precondiciones</b>	Debe de habilitar la animación invocando la función de color.
<b>Referencias</b>	RF6
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
1. El CU se inicia cuando el visualizador web invoca la función genérica para la animación de color.	2. El sistema requiere los parámetros de configuración de la función genérica de color.
3. El Visualizador web selecciona la variable o expresión que define el cambio del color.  a) Editar correspondencia entre los valores discretos de la variable con los colores deseados para cada valor.  b) Permitir que se interpole entre valores discretos de colores para valores intermedios de la variable.	4. El CU finaliza mostrando el cambio de color del objeto gráfico en la interfaz del Visualizador web.
<b>Pos condiciones</b>	El sistema visualiza el color del objeto en función del valor asignado a la variable asociada.

Tabla 9: Descripción del caso de uso: Visualizar animación de escalado.

<b>Caso de uso</b>	Visualizar animación de escalado
<b>Actores</b>	Visualizador web

## Capítulo 2: Análisis y Diseño del Sistema

<b>Propósito</b>	El CU se realiza con el objetivo de aumentar o disminuir la dimensión de un objeto.
<b>Resumen</b>	El CU se lleva a cabo cuando se decide escalar un objeto en función de los valores que tome una variable.
<b>Precondiciones</b>	Debe de habilitar la animación invocando la función de escalado.
<b>Referencias</b>	RF7
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
1. El CU se inicia cuando el visualizador web invoca la función genérica para el escalado.	2. El sistema requiere los parámetros de configuración de la función genérica de escalado.
3. El Visualizador web permite habilitar o no la animación de escalado. a) Seleccionar la variable o expresión que va a definir el escalado. b) Indicar el valor de la variable que se corresponde con el valor inicial del intervalo (en porcentaje) de escalado. c) Indicar el valor de la variable que se corresponde con el valor final del intervalo (en porcentaje) de escalado.	4. El CU finaliza mostrando el escalado en el objeto gráfico variando la dimensión del mismo en la interfaz del Visualizador web.
<b>Pos condiciones</b>	El sistema permitirá modificar la escala aumentando o disminuyendo su tamaño en relación

a un punto fijo de referencia en el mismo objeto.

## 2.5 Patrones de arquitectura

Durante el desarrollo de un sistema de software, se necesita seleccionar diferentes patrones, los cuales permiten obtener una buena estructura y organización que hace eficiente el funcionamiento de dicho sistema. Estos patrones no son más que una guía para realizar alguna acción. Especifican un conjunto predefinido de subsistemas con sus responsabilidades y una serie de recomendaciones para organizar los distintos componentes. Comprenden el diseño de más alto nivel de la estructura del sistema, expresan un esquema organizativo y estructural para sistemas de software, lo que posibilita un mejor entendimiento del problema que se le quiera dar solución. ([Wilson, 2015](#))

### 2.5.1 Patrón arquitectónico Vista Vista-Modelo Modelo

Para desarrollar la propuesta de visualización de animaciones no se precisa del empleo de un patrón arquitectónico. Teniendo en cuenta que la misma debe integrarse al Visualizador web del módulo HMI, esta debe adaptarse al patrón arquitectónico que posee, para el cual se empleó Modelo Vista Vista-Modelo (MVVM por sus siglas en inglés). Al integrar la solución propuesta al Visualizador web su ubicación sería en la capa intermedia Vista-Modelo. En la capa Vista del Visualizador web se invocan las funciones genéricas implementadas en la solución propuesta, para esto se configuran los parámetros necesarios en cada función para su posterior utilización. En la capa Vista-Modelo se encuentran las funciones genéricas que en dependencia de los datos configurados y el tipo de animación solicitada permite que se pueda visualizar en la capa Vista el resultado obtenido. A continuación se describen las principales características del patrón mencionado.

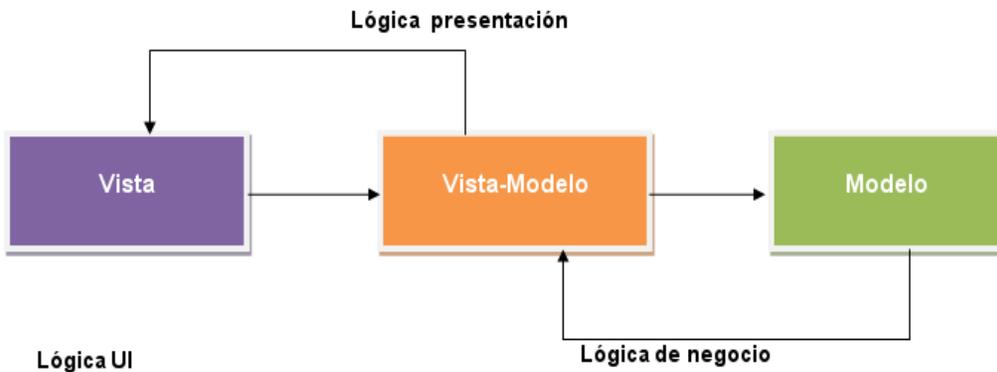


Figura 4: Representación del patrón arquitectónico Vista-Vista Modelo-Modelo.

Es una variante de otro patrón muy conocido, llamado *Model-View-Controller*, o MVC. Este patrón se emplea en un gran número de marcos; en especial, el marco de aplicaciones web de uso extenso Ruby on Rails, además de ASP.NET MVC de Microsoft, AngularJs. No sólo se emplea en las aplicaciones web, sino que también se usa extensamente desde las aplicaciones de escritorio hasta las aplicaciones móviles. ([Microsoft, 2015](#))

Está compuesto por 3 entidades:

- Modelo: representa el dominio del negocio: acceso a datos, clases modelos, reglas del negocio;
- Vista: figura la interfaz de usuario de la aplicación.
- Vista-Modelo: resulta ser una capa intermedia entre el modelo y la vista, con el objetivo de procesar todas las peticiones que tenga la vista hacia el modelo, además de ocuparse del manejo de las reglas del negocio.

## 2.6 Patrones de diseño

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de un software, es decir brindan una solución ya probada y documentada a problemas de desarrollo de software

## Capítulo 2: Análisis y Diseño del Sistema

---

que están sujetos a contextos similares. Esto provee las siguientes ventajas: reduce los esfuerzos de desarrollo y mantenimiento, mejora la seguridad informática, eficiencia y consistencia del diseño, proporciona un considerable ahorro en la inversión. ([etnassoft, 2015](#))

- **Adaptador (Adapter):** El patrón adaptador traduce una determinada clase o interfaz en un modelo compatible para todo escenario. Convierte la interfaz de una clase en otra interfaz que el cliente espera. Permite a las clases trabajar juntas, lo que de otra manera no podría hacerlo debido a sus interfaces incompatibles.

```
angular.module('animation',[]);  
  
angular.module('animation').service('animationSrv',function($interval){  
  
    this.A_trasladar=function(Obj,posx,posy,vel)
```

Figura 5: Fragmento de código del patrón Adaptador en la solución propuesta.

La figura 5 es una representación de un fragmento de código donde la creación de todas las funcionalidades es convertida en un servicio que podrá ser utilizado en cualquier momento dentro del uso de una etiqueta determinada que se desee animar, en este caso se ejemplifica con el cambio de imagen.

- **Observador (Observer):** El patrón observador consiste en un objeto, llamado 'sujeto', que mantiene una lista de dependencias, 'observadores' (define una dependencia del tipo uno-a-muchos entre objetos), las cuales le notifican automáticamente de cualquier cambio.

```
angular.module('exa',[]);  
angular.module('exa').directive('animate',function($interval){
```

Figura 6: Fragmento de código del patrón Observador en la solución propuesta.

## Capítulo 2: Análisis y Diseño del Sistema

---

La figura 6 es una representación de un fragmento de código donde se crea una directiva<sup>9</sup> (animate) para ser incluida en el cuerpo (body) del código HTML; la misma incluirá varias funciones genéricas con todas las animaciones de la solución propuesta que podrán ser invocadas desde dicho HTML.

- **Iterador (*Iterator*):** El patrón iterador se encarga de recorrer un conjunto de elementos para interactuar de forma individual con cada uno de ellos, pueden presentar diferentes tipos de recorrido sobre la estructura (recorrido de principio a fin y recorrido con saltos)

```
$interval(function(){
    var actual_image = array_image.shift();
    Obj.attr({
        'src':actual_image,
    });
    array_image.push(actual_image);
},2000);
};
```

Figura 7: Fragmento de código del patrón Iterador en la solución propuesta.

La figura 7 es una representación de un fragmento de código de la función genérica: cambio de imagen, donde se debe indicar una lista de imágenes que serán mostradas una a continuación de otra en un intervalo de tiempo definido entre dichas imágenes.

- **Constructor (*Builder*):** El constructor es un patrón básico para lograr la abstracción de los pasos necesarios para la creación de un objeto. El propósito de este patrón es separar la construcción de un objeto complejo de su representación, de forma que el mismo proceso de construcción pueda crear diferentes presentaciones.

---

<sup>9</sup> Instrucciones que se utilizan para construir programas fuente que puedan ser fácilmente modificados y compilados en diferentes entornos de ejecución.

```
link:function(scope,element,attr){
    scope.axisx = parseInt(scope.axisx);
    scope.axisy = parseInt(scope.axisy);
    scope.velocity = parseInt(scope.velocity);
    animationSrv.A_trasladar(element,scope.axisx,scope.axisy,scope.velocity);
},
```

Figura 8: Fragmento de código del patrón Constructor en la solución propuesta.

La figura 8 es una representación de un fragmento de código de la función genérica: trasladar, donde teniendo en cuenta la definición de posición de un objeto, se puede variar la misma a partir de los valores indicados en los ejes x, y para un movimiento horizontal, vertical y diagonal con la velocidad deseada.

## 2.7 Conclusiones parciales

En el presente capítulo se realizó el análisis y diseño de la solución propuesta. Se definió el modelo del dominio, el cual refleja el punto de partida de la solución. Se especificaron los requisitos del sistema que permitieron identificar las funcionalidades con las que este contará y que darán respuesta a las necesidades del Visualizador web. A raíz de esto se determinaron 10 requisitos, 3 no funcionales y 7 funcionales, los cuales fueron encapsulados en el diagrama de caso de uso del sistema.

Se especificaron funcionalidades y características que posee el sistema a través del cual surgió el modelado del dominio. Se tuvo en cuenta la arquitectura basada en patrones que se utilizó para el Visualizador web (MVVM) para especificar la ubicación de la solución propuesta en dicho patrón, directamente en la entidad Vista-Modelo. Partiendo de los frameworks utilizados se seleccionaron los patrones de diseño: Adaptador, Observador, Iterador y Constructor

## CAPÍTULO 3 IMPLEMENTACIÓN Y PRUEBAS

### 3.1 Introducción

Como resultado del análisis y diseño de la solución propuesta, se procede a la implementación del sistema en términos de componentes: ejecutables y código fuente. Para lograr una correcta implementación, es necesario que la solución propuesta cumpla con las expectativas del cliente, asegurando que sea operacional y que funcione de acuerdo a los requisitos definidos. Derivándose entonces la incorporación de pruebas al sistema, actividad para garantizar la calidad del software. En el presente capítulo se exponen además de los aspectos mencionados anteriormente, aquellos relacionados con los estándares de codificación que permiten la creación de un código fácil de entender.

### 3.2 Modelo de implementación

Para la composición física de la implementación del sistema se realizó el modelo de implementación, el cual está integrado por un conjunto de componentes. Este modelo permitió definir una organización del código, implementar los elementos de diseño y finalmente probar y desarrollar los componentes como unidades. La implementación es esencial en el proceso de poner en práctica el diseño de la solución propuesta precisando los pasos clave del análisis realizado; logrando un resultado que cumpla con las especificaciones realizadas por el cliente. Esta fase es un proceso de fabricación, en el que se pone énfasis a la gestión de los recursos y el control de las operaciones para optimizar los costos, la planificación y la calidad del producto final.

Para conformar el modelo de implementación se realizaron dos diagramas fundamentales, el diagrama de componentes y el diagrama de despliegue, los cuales se describen a continuación.

## 3.2.2 Diagrama de componentes

Para visualizar la estructura general del sistema se generó el diagrama de componente, el cual permitió describir el modelo de implementación y las relaciones entre los elementos de dicho modelo. El diagrama de componentes se representa como un grafo unido por medio de las relaciones de dependencia que existe entre los componentes del sistema y muestra un conjunto de ficheros relacionados entre sí para lograr una completa funcionalidad del sistema. ([Alvayrdgur, 2013](#))

A continuación se presenta el diagrama de componentes de la solución propuesta:

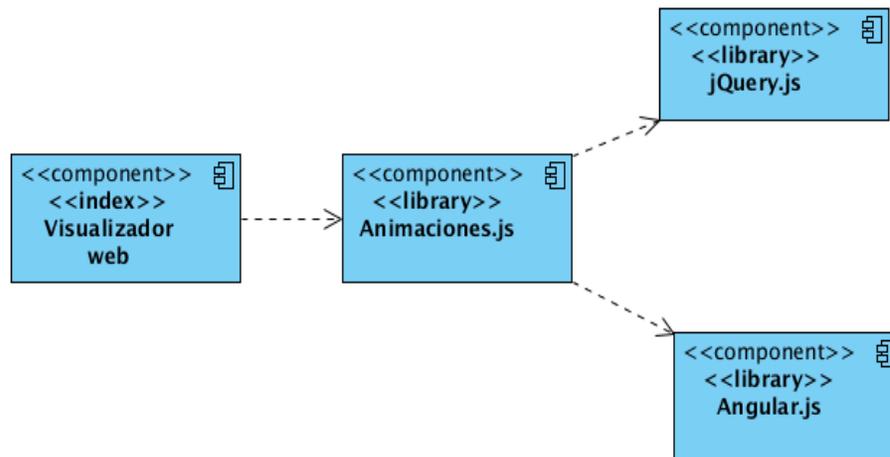


Figura 9: Diagrama de componentes de la solución propuesta.

El flujo de acciones del diagrama de componentes presentado se define de la siguiente forma:

## 3.2.3 Descripción del diagrama de componentes

Para una mejor comprensión del diagrama presentado en la siguiente tabla se resumen algunos de los componentes utilizados y el propósito de cada uno de estos:

Tabla 10: Descripción de los componentes del diagrama de componentes.

Componente	Propósito
------------	-----------

Propuesta de visualización de animaciones para objetos gráficos del Visualizador web del SCADA GALBA.

# Capítulo 3: Implementación y pruebas

---

Visualizador web	Es donde se incluirán las animaciones para hacer la llamada de las funciones genéricas y para mostrar los objetos con la animación asociada.
Animaciones	Agrupar las funciones genéricas con animaciones (rotación, traslación, escalado, visibilidad, texto, imagen y color).
JQuery.js	Librería que se utilizó para crear las funciones de las animaciones con JQuery.
Angular.js	Librería que se utilizó para crear las funciones de las animaciones con AngularJs.

## 3.3 Modelo de despliegue

Con el objetivo de proveer una descripción de la distribución física del sistema se realiza el modelo de despliegue, mediante el cual se muestran las relaciones físicas de los distintos nodos que componen la solución propuesta. Los diagramas de despliegue son los complementos de los diagramas de componentes, que unidos proporcionan la vista de implementación del sistema. A su vez describen la topología del sistema, es decir, la estructura de los elementos de hardware y el software que consumen. Igualmente muestran las relaciones físicas de los distintos nodos que componen un sistema. ([Alvayrdgur, 2013](#))

A continuación se muestra el modelo de despliegue del sistema en donde los nodos representados mediante estereotipos se conectan mediante soportes bidireccionales.

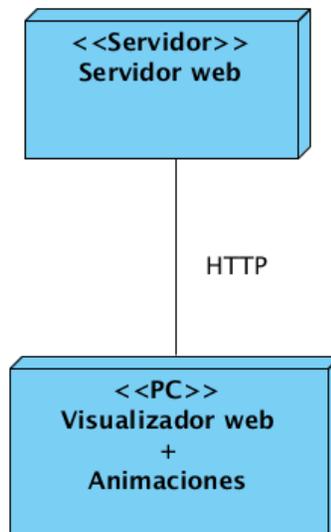


Figura 10: Diagrama de despliegue de la solución propuesta.

En el diagrama presentado se observa la relación existente entre el software y el hardware, donde se especifica que el servidor web responderá a las peticiones que se realizan desde el Visualizador web a través del protocolo HTTP<sup>10</sup> para el acceso a las funciones de las animaciones, que permitirán que se dinamicen los objetos deseados.

### 3.4 Estándares de codificación

Los estándares de codificación, también conocidos como estilos de programación o convenciones de código, son convenios para escribir código fuente en ciertos lenguajes de programación. Permiten que el código en consecuencia sea mantenible y que todos los participantes lo puedan comprender en un menor tiempo. (Calleja, 2014)

---

<sup>10</sup> Protocolo de transferencia de hipertexto, define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web, sigue el esquema petición-respuesta entre un cliente y un servidor.

# Capítulo 3: Implementación y pruebas

---

Es debido a estos argumentos que la codificación de las funciones genéricas para la propuesta de visualización de las animaciones, debe cumplir con ciertos requisitos, algunos de los cuales son detallados a continuación:

- Los nombres de las variables deben seguir la notación *lowerCamelCase*.
- Las funciones genéricas tienen como prefijo la letra “A” seguido por el carácter “\_” y el nombre de la acción de animación.
- Todo el código desarrollado tendrá una indentación<sup>11</sup> de 3 espacios.
- Todas las funciones tienen que estar comentadas explicando el funcionamiento de cada una de ellas. Para ello se utilizará el estándar de documentación de gnome. El formato de los comentarios tienen la siguiente forma:

```
/**  
    * nombre_funcion:  
    * @par: descripción de los parámetros  
    * Descripción de la función.  
*/
```

## 3.5 Pruebas del sistema

Una vez finalizada la Implementación del sistema, se procede a la realización de las pruebas que permitirán comprobar el funcionamiento del sistema, con el objetivo de evaluar o valorar la calidad del producto a través de la búsqueda de errores y validar el cumplimiento de los requisitos del sistema. Las pruebas de software son las investigaciones empíricas y técnicas cuyo fin es proporcionar información objetiva e independiente sobre la calidad del producto. Esta actividad forma parte del proceso de control de la calidad global. Las pruebas son un conjunto de actividades dentro del desarrollo de software y dependiendo del tipo de pruebas, estas actividades podrán ser implementadas en cualquier momento del

---

<sup>11</sup> Mover un bloque de texto hacia la derecha insertando espacios o tabuladores, para así separarlo del margen izquierdo y mejor distinguirlo del texto adyacente.

proceso de desarrollo. Para que las pruebas tengan éxito es necesario realizar casos de pruebas que tengan probabilidad de descubrir los errores en el sistema y utilizar técnicas que guíen este proceso. Para probar la solución propuesta se empleó el método de Caja Negra. ([PRESSMAN, 2010](#))

## 3.5.1 Prueba de Caja Negra

Son las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba (CP) pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene. ([PRESSMAN, 2010](#))

Estas pruebas permiten encontrar:

- Funciones que estén incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

Para aplicar la prueba al sistema se definieron por casos de pruebas teniendo en cuenta un conjunto de entradas, condiciones de ejecución y los resultados esperados. Se utilizó la técnica de partición de equivalencia, para aplicarla se tuvo en cuenta lo siguiente:

### Partición de Equivalencia

La técnica Partición de equivalencia es la más usada en el método Caja Negra, la cual divide el dominio de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. El diseño de estos casos de prueba para la partición de equivalencia se basa en la evaluación de las clases de equivalencia. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada. En las tablas que se muestran a continuación se puede apreciar los diseños de casos de pruebas de los casos de uso definidos.

## Capítulo 3: Implementación y pruebas

### Condiciones de ejecución

1. La propuesta de visualización de las animaciones debe estar incluida en el Visualizador web.
2. En el Visualizador web debe invocarse la función "A\_trasladar".

### CP Visualizar animación de traslación.

Tabla 11: Diseño del caso de prueba Visualizar animación de traslación.

Escenario	Variable	Descripción	Respuesta del sistema	Flujo central
EC 1.1: Trasladar el objeto en el eje vertical.	Obj(id)	Permite que el objeto se traslade en el eje vertical.	En el Visualizador web se muestra un desplazamiento del objeto solo verticalmente.	->Indicar el parámetro del identificador del objeto.
	posx(# =0)			->Indicar valor igual a cero para el eje X.
	posy(# ≠0)			->Indicar cualquier valor diferente de cero para el eje Y.
	Vel(#seg)			->Indicar valor de velocidad en segundos.
EC 1.2: Trasladar el objeto en el eje horizontal.	Obj(id)	Permite que el objeto se traslade en el eje horizontal.	En el Visualizador web se muestra un desplazamiento del objeto solo horizontalmente.	->Indicar el parámetro del identificador del objeto.
	posx(# ≠0)			->Indicar valor diferente de cero para el eje X.
	posy(# =0)			

## Capítulo 3: Implementación y pruebas

	Vel(#seg)			->Indicar valor igual a cero para el eje Y. ->Indicar valor de velocidad en segundos.
EC 1.3: Trasladar el objeto en su diagonal.	Obj(id)	Permite que el objeto se traslade en su diagonal.	En el Visualizador web se muestra un desplazamiento del objeto solo diagonalmente.	->Indicar el parámetro del identificador del objeto.
	posx(# ≠0)			->Indicar valor diferente de cero para el eje X.
	posy(# ≠0)			->Indicar valor diferente de cero para el eje Y.
	Vel(#seg)			->Indicar valor de velocidad en segundos.

### CP Visualizar animación de visibilidad.

#### Condiciones de ejecución

1. La propuesta de visualización de las animaciones debe estar incluida en el Visualizador web.
2. En el Visualizador web debe invocarse la función "A\_visible".

Tabla 12: Diseño del caso de prueba Visualizar animación de visibilidad.

Escenario	Variable	Descripción	Respuesta del sistema	Flujo central
-----------	----------	-------------	-----------------------	---------------

## Capítulo 3: Implementación y pruebas

EC 1.1: Mostrar un objeto	Obj(id)	Permite mostrar el objeto deseado.	En el Visualizador web se muestra el/los objeto(s) especificados.	->Indicar el parámetro del identificador del objeto.
	cond(si)			->Indicar el parámetro con la cadena "si".
EC 1.2: Ocultar un objeto	Obj(id)	Permite ocultar el objeto deseado.	En el Visualizador web se oculta el/los objeto(s) especificados.	->Indicar el parámetro del identificador del objeto.
	cond(no)			->Indicar el parámetro con la cadena "no".

### CP Visualizar animación de escalado.

#### Condiciones de ejecución

1. La propuesta de visualización de las animaciones debe estar incluida en el Visualizador web.
2. En el Visualizador web debe invocarse la función "A\_escalado".

**Tabla 13: Diseño del caso de prueba Visualizar animación de escalado.**

Escenario	Variable	Descripción	Respuesta del sistema	Flujo central
EC 1.1: Aumentar	Obj(id)	Permite escalar un objeto aumentando el	En el Visualizador web se muestra el	->Indicar el parámetro del

## Capítulo 3: Implementación y pruebas

porcentaje de un objeto.	pc(# >100%)	porcentaje del mismo.	escalado del objeto mientras aumenta su tamaño original.	identificador del objeto. ->Indicar parámetro con el valor de porcentaje mayor al 100%(actual). ->Indicar parámetro con el valor de la velocidad en segundos.
	vel(#seg)			
EC 1.2: Disminuir porcentaje de un objeto.	Obj(id)	Permite escalar un objeto disminuyendo el porcentaje del mismo.	En el Visualizador web se muestra el escalado del objeto mientras disminuye su tamaño original.	->Indicar el parámetro del identificador del objeto. ->Indicar parámetro con el valor de porcentaje menor al 100%(actual). ->Indicar parámetro con el valor de la velocidad en segundos.
	pc(# <100%)			
	vel(#seg)			

### CP Visualizar animación de imagen.

#### Condiciones de ejecución

1. La propuesta de visualización de las animaciones debe estar incluida en el Visualizador web.
2. En el Visualizador web debe invocarse la función "A\_imagen".

## Capítulo 3: Implementación y pruebas

Tabla 14: Diseño del caso de prueba Visualizar animación de imagen.

Escenario	Variable	Descripción	Respuesta del sistema	Flujo central
EC 1.1: Intercambiar imágenes.	Obj1(id)	Permite mostrar el intercambio de dos objetos o imágenes.	En el Visualizador web se muestra dinámicamente el intercambio de dos objetos o imágenes.	->Indicar el parámetro del identificador del objeto1.
	Obj2(id)			->Indicar el parámetro del identificador del objeto2.

### CP Visualizar animación de color.

#### Condiciones de ejecución

1. La propuesta de visualización de las animaciones debe estar incluida en el Visualizador web.
2. En el Visualizador web debe invocarse la función "A\_color".

Tabla 15: Diseño del caso de prueba Visualizar animación de color.

Escenario	Variable	Descripción	Respuesta del sistema	Flujo central
EC 1.1: Cambio de color.	Obj(id)	Permite mostrar la interpolación entre dos colores definidos.	En el Visualizador web se muestra dinámicamente el intercambio entre dos colores, pasando por la gama de colores	->Indicar el parámetro del identificador del objeto.
	color1(name o #RGB)			->Indicar el nombre en inglés del primer color o identificador RGB del

## Capítulo 3: Implementación y pruebas

	color2(name o #RGB)		existentes entre ellos.	mismo. ->Indicar el nombre en inglés del segundo color o identificador RGB del mismo.
--	---------------------	--	-------------------------	--

### CP Visualizar animación de texto.

#### Condiciones de ejecución

1. La propuesta de visualización de las animaciones debe estar incluida en el Visualizador web.
2. En el Visualizador web debe invocarse la función "A\_texto".

Tabla 16: Diseño del caso de prueba Visualizar animación de texto.

Escenario	Variable	Descripción	Respuesta del sistema	Flujo central
EC 1.1: Intercambiar textos.	text1 (id)	Permite mostrar el intercambio de dos textos.	En el Visualizador web se muestra dinámicamente el intercambio de dos textos.	->Indicar el parámetro del identificador del texto1.
	text2(id)			->Indicar el parámetro del identificador del texto2.

### CP Visualizar animación de rotar.

#### Condiciones de ejecución

1. La propuesta de visualización de las animaciones debe estar incluida en el Visualizador web.
2. En el Visualizador web debe invocarse la función "A\_rotar".

Tabla 17: Diseño del caso de prueba Visualizar animación de rotar.

# Capítulo 3: Implementación y pruebas

Escenario	Variable	Descripción	Respuesta del sistema	Flujo central
EC 1.1: Rotar un objeto.	Obj(id)	Permite mostrar la rotación de un objeto hasta un ángulo de 360°.	En el Visualizador web se muestra dinámicamente la rotación del objeto especificado hasta un ángulo de 360°.	->Indicar el parámetro del identificador del objeto.
	angulo(0>#<360)			->Indicar el valor del giro que efectuará el objeto.

## 3.5.2 Resultados de las Pruebas

A continuación se describen las no conformidades detectadas en cada iteración y cuáles de estas fueron significativas.

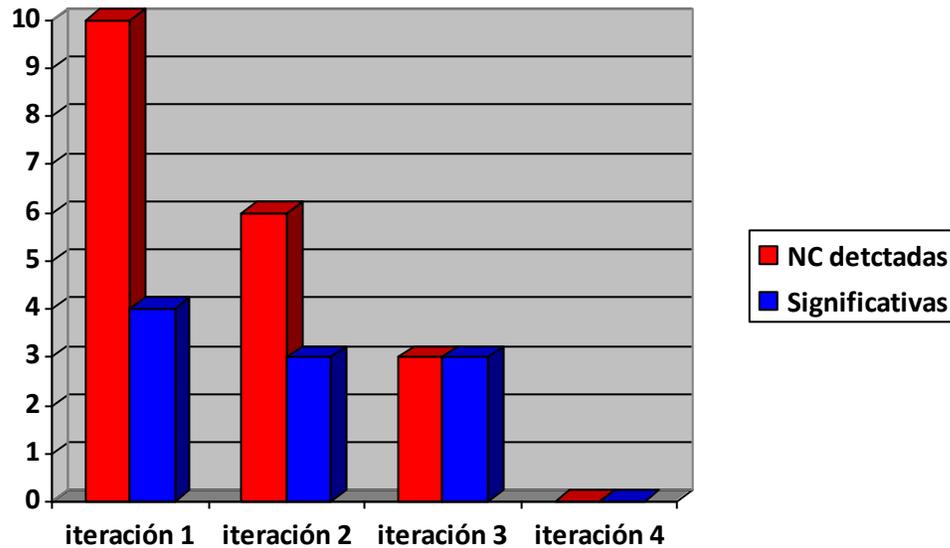


Figura 11: Iteraciones de las NC detectadas durante la prueba de Caja Negra.

# Capítulo 3: Implementación y pruebas

---

Cada incidencia detectada en el desarrollo del sistema fue resuelta a raíz del trabajo continuo del desarrollador, con un total de 10 no conformidades encontradas en la primera iteración, 6 en la segunda y 3 en la tercera, las cuales se dividieron en significativas y no significativas, como aparece representado en el gráfico de la figura 11. El resultado de las pruebas realizadas al sistema propuesto fue satisfactorio. Finalmente en la cuarta iteración se comprobó que el resultado obtenido fue el esperado, lo que significa que la solución propuesta está lista para ser integrada al Visualizador web del HMI del SCADA GALBA.

## 3.6 Beneficios del sistema

La solución del sistema propuesto que se ha desarrollado podrá ser integrada sin dificultades al Visualizador web implementado en el CEDIN, propiciando mejoras significativas con respecto al anterior modo de la representación de la información adquirida a través de los componentes gráficos en el sistema SCADA GALBA. Con el resultado obtenido en la investigación los operadores pueden beneficiarse ya que se mejoró la representación gráfica de los objetos que intervienen en los procesos de una planta industrial. Permite observar el comportamiento de los componentes gráficos en los despliegues teniendo en cuenta la actualización de la información en tiempo real, mejorando la interpretación de la misma de una manera más rápida e intuitiva por parte de los operadores.

El sistema permitirá la reutilización de sus funcionalidades con distintos fines deseados a la hora de representar la acción de un evento ocurrido. Proporciona la optimización de grandes cantidades de líneas de código. Posibilita además invocar la función correspondiente a la acción del objeto e indicar los valores pertinentes para mostrar dinamismo en dicho objeto, adquiriendo de esta forma una apariencia en tiempo real.

## 3.7 Conclusiones parciales

En el capítulo expuesto se construyó el modelo de implementación del sistema, para ello se realizó el diagrama de componentes en el que se definieron los componentes necesarios para el correcto funcionamiento de la solución propuesta y se describieron cada uno de ellos para lograr una mayor comprensión acerca de su función. Se definió el modelo de despliegue el cual muestra los elementos

## *Capítulo 3: Implementación y pruebas*

---

necesarios del sistema, así como los estándares de codificación empleados para lograr una mejor organización y comprensión del código desarrollado para las distintas funciones genéricas. Una vez terminado el flujo de trabajo de Implementación se pasó al flujo de trabajo de prueba, donde se aplicó el método de caja negra, empleando la técnica de partición de equivalencia. Cada no conformidad detectada fue satisfactoriamente resuelta.

## CONCLUSIONES GENERALES

El presente trabajo tuvo como base la investigación y el desarrollo científico encaminados al desarrollo e integración de una propuesta de animaciones para el Visualizador web en el módulo HMI del SCADA GALBA, para elevar los niveles de interpretación de valores y simulación en el proceso de visualización gráfica.

Se realizó el análisis y diseño del sistema e implementación y prueba de Caja Negra con las funcionalidades definidas obteniendo el logro de un sistema que brinde las posibilidades representar en tiempo real los estados que toman los procesos que se estén controlando.

Como resultado del trabajo realizado se logró la implementación de un sistema que brinda las siguientes ventajas:

- Se beneficiarán los operadores, porque con las animaciones se mejoró la representación gráfica de los objetos o procesos que intervienen en una planta industrial.
- Modificar según un evento el comportamiento de los componentes gráficos en los despliegues.
- Los textos muestran un comportamiento diferenciado.
- Representación la información adquirida asociada a diversos eventos definidos dinámicamente.
- Reutilización de la propuesta de visualización de las animaciones para aplicaciones con características similares.
- Fácil uso de la aplicación.

El estudio en las diferentes temáticas de los procesos de visualización gráfica para los entornos web de la interfaz hombre-máquina en un sistema de supervisión y control, se ven reflejado en el resultado alcanzado.

## RECOMENDACIONES

Integrar nuevas animaciones a la propuesta de visualización de animaciones como son:

- Redimensionar
- Parpadeo
- Rellenado
- Movimiento compuesto
- Transparencia

## REFERENCIAS BIBLIOGRÁFICAS

**Alvayrdgur, Eduardo Rivera. 2013.** Scribd. Arquitectura de Software II. [En línea] 1 de junio de 2013. [Citado el: 1 de abril de 2015.] <http://es.scribd.com/doc/7884665/Arquitectura-de-Software-II-Diagrama-de-Componentes-y-Despliegue>.

**AUTC. 2015.** Movicon. [En línea] 2015. [Citado el: 15 de marzo de 2015.] <http://www.autc.com.pe/index.php/progea/movicon11.html>.

**Calleja, Arias. 2014.** Estándares de codificación. [En línea] Carmen, 2014. [Citado el: 2 de abril de 2015.] <http://www.chubut.gov.ar/informatica/docs/EstandaresCodificacion.pdf>.

**Desarrolloweb. 2015.** Qué es Angular? *Desarrolloweb.com*. [En línea] 2015. [Citado el: 15 de marzo de 2015.] <http://www.desarrolloweb.com/articulos/que-es-angularjs-descripcion-framework-javascript-conceptos.html>.

**etnassoft. 2015.** Patrones de diseño. [En línea] 2015. [Citado el: 27 de febrero de 2015.] <http://www.etnassoft.com/2011/05/26/patrones-de-diseno-utilizados-por-jquery/>.

**Flanagan, David. 2002.** JavaScript : The Definitive Guide. s.l. : 4<sup>a</sup> Edición edición, 2002.

**Informáticas, Universidad de las Ciencias. 2014.** Metodología de desarrollo para la actividad productiva de la UCI. *Programa de mejora*. 2014.

**Intouch, Scada Wonderware. 2015.** Scada Wonderware Intouch. [En línea] 2015. [Citado el: 15 de marzo de 2015.] <http://plc-hmi-scadas.com/SCADA-INTOUCH/>.

**ISEF.2010.** Instituto Schneider Electric de Formación.Bac de Roda 52, Edificio A – 1<sup>a</sup> Planta. [En línea] 27 de Octubre de 2010.[Citado el: 2 de noviembre del 2015] [http://www.equiposdidacticos.com/pdf/catalogos/Manual\\_formacion\\_Vijeo\\_Designer.pdf](http://www.equiposdidacticos.com/pdf/catalogos/Manual_formacion_Vijeo_Designer.pdf).

**Larman, Craig.** Introducción al Análisis y Diseño Orientado a Objetos. s.l. : Primera Edición, 1999 y Segunda Edición.

**Mandado, Enrique. 2009.** Autómatas programables y sistemas de automatización. Barcelona : s.n., 2009.

**Mario, Campusano Zuluaga Jorge. 2010.** Diseño de Sistemas. 2010.

# Referencias Bibliográficas

---

- Medios, Ruben. 2015.** Brackets. *Un editor de código web para diseñadores y desarrolladores eficientes.* [En línea] 2015. [Citado el: 2 de noviembre de 2015.] <http://blog.opencademy.com/desarrollo-web/brackets-un-editor-de-codigo-web-para-disenadores-desarrolladores-eficientes/>.
- Microsoft.2015.**Centro de desarrollo de Windows. [En línea] 2015. [Citado el: 15 de abril de 2015] <https://msdn.microsoft.com/es-es/library/windows/apps/jj160324.aspx>.
- Montero, Dagoberto, Barrantes, David B. y Quiros, Jose M. 2014.** Introducción a los Sistemas de Control, Supervisión y Adquisición de datos(SCADA). [En línea] 2014. [Citado el: 30 de octubre de 2014.] <http://www.scribd.com/doc/13473499/Introduccion-a-Los-Sistemas-SCADA>.
- Paradigm, Visual. 2014.** UML CASE tool for software development. [En línea] 2014. [Citado el: 10 de diciembre de 2014.] <http://www.visual-paradigm.com/product/vpum/>.
- PRESSMAN. 2010.** Ingeniería del software. [aut. libro] 2003. marzo. McGraw-Hill Interamericana de España : 9786071503145, 2010.
- . 2010. Ingeniería de Software Un Enfoque Práctico. [aut. libro] ROGER S. 2010.
- Resig, John. 2014.** jQuery 1.11 Released. 2014.
- Sommerville. 2005.** Ingeniería del software. *Pearson Educación.* s.l. : 714 p. 9788478290741, 2005.
- UCI. 2015.** [En línea] 2015. [Citado el: 7 de marzo de 2015.]
- UML. 2013.** Object Management Group. [En línea] 2013. [Citado el: 7 de diciembre de 2015.] <http://www.uml.org/>
- w3schools. 2015.** w3schools.com. [En línea] 2015. [Citado el: 2015 de enero de 2015.] <http://www.w3schools.com/css/default.asp>.
- W3Schools. 2015.** w3schools.com.HTML. [En línea] 2015. [Citado el: 27 de enero de 2015.] <http://www.w3schools.com/html/default.asp>.
- Wilson, Jose Manuel Ayala. 2015.** Blog sobre arquitectura. [En línea] 2015. [Citado el: 10 de abril de 2015.] <http://jmaw.blogspot.com/2011/04/h2-margin-bottom-0.html>.

## GLOSARIO DE TÉRMINOS

**Framework:** En el desarrollo de software es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

**SCADA:** Sistema para la automatización de procesos industriales que permite la supervisión y el control de dichos procesos.

**Software:** Es la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo.

**Caso de uso (CU):** Secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

**Directiva:** Instrucción empleada en ensamblador que no tiene equivalente en lenguaje máquina; generalmente proporciona información al ensamblador acerca del entorno en que ha de ensamblarse el programa fuente.

**Biblioteca:** Colección de programas, rutinas y subrutinas almacenadas y probadas, por lo general traducidas al lenguaje de máquina, que están a disposición de los programadores, para ser utilizadas con un objetivo determinado.

**XML:** Lenguaje de marcación extendido. Tecnología que permite un mejor uso de los documento en la Web de Internet, principalmente para el comercio electrónico.

**Cliente:** Nombre que reciben aquellas computadoras que forman parte de una red y que reciben la información y el software de otro principal llamado servidor.

**Ventana:** Parte fundamental de la interfaz grafica de la computadora, especialmente con sistemas operativos que permiten el multiprocesamiento.

**Artefacto:** Un artefacto es un producto tangible resultante del proceso de desarrollo de software. Algunos artefactos como los casos de uso, diagrama de clases u otros modelos UML ayudan a la descripción de la función, la arquitectura o el diseño del software.

**Análisis:** Conjunto de documentaciones que acompañan a uno o varios programas y en las que se indican las características fundamentales que reúnen dichos programas.

**Aplicación:** Determinado programa de propósito general destinado a solucionar problemas del usuario en muy diversas áreas: gestión de ficheros de datos, realización de cálculos, escritura de documentos, control de almacenes, contabilidad, etc. Una aplicación tiene como mínimo un programa y puede tener o no ficheros de datos.

**Desplazamiento:** Dirección relativa al segmento que forma parte, junto con el selector, de una dirección lógica (virtual).

**Etiqueta:** Nombre simbólico que permite identificar un bloque de líneas en un programa o archivo de proceso por lotes.

**Función:** Módulo de un programa que realiza una función determinada y que siempre devuelve información a la parte del programa que lo llamó.

**Iteración:** Proceso o conjunto de procesos que se repiten un número determinado de veces hasta lograr los resultados deseados.