



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DESARROLLO DE UNA INTERFAZ GRÁFICA
PROTOTIPO PARA EL ENTRENAMIENTO A
DISTANCIA UTILIZANDO EL SIMULADOR DE AULA

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

PRESENTA:

ISRAEL HUMBERTO NÚÑEZ HERNÁNDEZ



DIRECTOR DE TESIS:
M. C. EDGAR SALAZAR SALAZAR

MÉXICO, D.F.

2004



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

ESTA TESIS NO SALE
DE LA BIBLIOTECA

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.

NOMBRE: Yo Israel Humberto
Núñez Hernández

FECHA: 16 enero 2007

FIRMA: Vanesa Núñez 

A mi familia,
a mis amigos,
a Isabel.

Agradecimientos

Deseo agradecer sinceramente al M.C Edgar Salazar Salazar por su asesoría y dirección en el presente trabajo, así como a los integrantes del H. Jurado, por sus valiosas aportaciones para la realización del mismo:

Presidente:	Dr. Juan Luis Francois Lacouture.
Vocal:	M.C. Edgar Salazar Salazar.
Secretario:	Dr. Héctor Ocampo Mancilla.
1er. Suplente:	Dr. Carlos Chávez Mercado.
2do. Suplente:	Dra. Cecilia Martín Del Campo Márquez.

También, agradezco a todas aquellas personas e instituciones que hayan participado directamente o indirectamente en la terminación del presente trabajo, ya sea con su tiempo, esfuerzo, apoyo o como inspiración.

ÍNDICE

INTRODUCCIÓN.	iv
CAPÍTULO 1. ANTECEDENTES.	1
1.1. Centrales nucleares.	2
1.2. Central Nuclear de Laguna Verde.	9
1.3. Simuladores.	11
1.4. Simulador réplica de la Central Nuclear de Laguna Verde.	12
1.5. Prototipo del Simulador de Aula para la CNLV.	13
1.6. Códigos RELAP / SCDAP.	15
CAPÍTULO 2: DEFINICIÓN DE REQUERIMIENTOS.	17
2.1. Descripción del problema.	18
2.2. Planteamiento de objetivos.	19
2.3. Especificación de requerimientos.	22
CAPÍTULO 3: DISEÑO.	24
3.1. Arquitectura del sistema.	25
3.2. Detalle de módulos.	27
CAPÍTULO 4. DESARROLLO.	32
4.1. Codificación de módulos.	33
4.2. Pruebas del sistema.	45
4.3. Análisis de resultados.	50
CONCLUSIONES.	51
• Conclusiones sobre el presente trabajo.	52
• Trabajos futuros.	52
BIBLIOGRAFÍA.	54
LISTA DE ACRÓNIMOS Y TÉRMINOS.	56

INTRODUCCIÓN

En el año de 1999, se integró a la División de Ingeniería Eléctrica de la Facultad de Ingeniería (DIE-FI) el Grupo de Ingeniería Nuclear (GRIN); el cual tiene como uno de sus principales objetivos desarrollar el Laboratorio de Análisis en Ingeniería de Reactores Nucleares (LAIRN).

El Grupo de Ingeniería Nuclear ha trabajado desde hace algunos años en el diseño de un prototipo de Simulador de Aula (SimAula), el cual consiste en un sistema de cómputo robusto, modular y re-configurable, que permite la simulación del funcionamiento de los principales sistemas de la Central Nucleoeléctrica Laguna Verde (CNLV), con la finalidad de crear un ambiente de simulación gráfica con propósitos de entrenamiento y análisis de procesos nucleares.

Este prototipo de Simulador de Aula incluye un sistema de visualización donde se observa el comportamiento de los procesos nucleares, mediante la representación virtual (emulación) de paneles de control, diagramas mímicos y pictóricos, así como gráficas de tendencia correspondientes a los principales sistemas de la central.

El Simulador de Aula se encuentra, actualmente, en el Laboratorio de Análisis en Ingeniería de Reactores Nucleares en las instalaciones de la División de Estudios de Postgrado (DEPFI) en Jiutepec, Morelos.

El presente trabajo pretende poner al alcance de usuarios que se encuentren geográficamente distantes, el entrenamiento o el análisis de accidentes en una central nuclear, utilizando el Simulador de Aula mediante la red de Internet.

El contenido de la presente tesis, se compone de seis capítulos.

El capítulo 1 contiene una descripción de los antecedentes, desarrollo, aplicaciones y capacidades del Simulador de Aula, además del código utilizado en el proceso de simulación (RELAP / SCDAP).

En el capítulo 2 se presentan los objetivos, haciendo gran énfasis en delimitar los alcances del presente trabajo; así como una descripción del problema en el que nos enfocaremos y la solución propuesta para el mismo.

El capítulo 3 contiene una descripción detallada de la arquitectura del sistema que desarrollaremos, así como también, una descripción detallada de los módulos que contendrá nuestro sistema.

En el capítulo 4 se muestra el desarrollo paso a paso de nuestro sistema, además de las pruebas realizadas al sistema a partir del desarrollo del mismo.

Finalmente, se mencionan las conclusiones generales y el trabajo futuro relacionado con el presente trabajo de Tesis, así como al Simulador de Aula.

CAPÍTULO 1

ANTECEDENTES

El Simulador de Aula es un sistema capaz de simular procesos utilizados en la generación de energía eléctrica en una Central Nuclear. A continuación, se presenta de manera general los aspectos más importantes de las Centrales Nucleares, su clasificación y funcionamiento, además de una visión general del Simulador de Aula, alcances, objetivos y las herramientas utilizadas en el proceso de simulación.

1.1. Centrales nucleares.

Una Central Nuclear es una instalación física donde se produce, mantiene y controla una reacción nuclear de fisión en cadena para producir energía eléctrica (una reacción nuclear en cadena ocurre cuando en una fisión nuclear aparecen neutrones libres, si en las proximidades hay más núcleos de Uranio estos neutrones producirán más fisiones). Estas centrales tienen cierta semejanza con las termoeléctricas convencionales, ya que también utilizan vapor a presión para mover turbogeneradores (un turbogenerador es una máquina termodinámica que tiene como función convertir la energía térmica en energía mecánica, está compuesto por turbina, generador eléctrico y condensador), pero en lugar de emplear combustibles fósiles para producirlo, aprovechan el calor que se obtiene al fisionar átomos de los isótopos U235 y PU239, en dispositivos denominados reactores (un reactor es un dispositivo en el cual se lleva a cabo un reacción nuclear en cadena controlada) (1).

El primer reactor construido en el mundo fue operado en 1942, en la Universidad de Chicago (USA), bajo la dirección del investigador Enrico Fermi.

Un Reactor Nuclear debe constar de tres componentes fundamentales, estos componentes son:

1. Combustible.

El Combustible en un reactor se encuentra en forma sólida, siendo el más utilizado el Uranio bajo su forma isotópica de U-235 (se denominan isótopos a los núcleos atómicos con el mismo número de protones pero con distinto número de neutrones, dos isótopos por tanto corresponden al mismo elemento químico pero tienen un peso atómico distinto). Sin embargo, hay elementos igualmente fisionables (una fisión es la ruptura en dos o más fragmentos del núcleo de un átomo pesado provocada por una partícula incidente, especialmente por un neutrón), como el Plutonio que es un transuránico producido por la captura neutrónica del Uranio.

El Uranio natural tiene poca cantidad de Uranio fisionable, alrededor del 0.7%, por lo que en la mayoría de los reactores se emplea combustible "enriquecido", es decir, combustible donde se aumenta la cantidad de Uranio 235 hasta un 3% para reactores térmicos del tipo PWR y BWR.

2. Moderador.

Los neutrones obtenidos de la fisión nuclear emergen con velocidades muy altas (20 000 Km/s). Para asegurar continuidad de la reacción en cadena, es decir, procurar que los "nuevos neutrones" sigan colisionando con los núcleos atómicos del combustible, es necesario disminuir la velocidad de estas partículas (2 Km/s). Se disminuye la energía cinética de los neutrones mediante choques con átomos de otro material llamado Moderador. Entre los Moderadores más comunes podemos citar a: el agua natural, el agua pesada D₂O (el agua pesada es agua formada con átomos de hidrógeno pesado, esto es, con los isótopos de deuterio) y el Carbono (grafito).

3. Refrigerante.

El calor generado por las fisiones se debe extraer del núcleo del reactor para producir el vapor que se requiere en la generación de energía eléctrica, y al mismo tiempo, mantener suficientemente baja la temperatura de los distintos elementos que se encuentran en su interior para que estos no sufran ningún deterioro. Para lograr este proceso se utilizan fluidos en los cuales se sumerge el núcleo. Se puede utilizar como refrigerante el agua, el agua pesada, el anhídrido carbónico, etc. Cuando se va a convertir el calor generado en energía eléctrica, el calor del refrigerante se transfiere a un fluido de trabajo, con el fin de producir vapor de agua o un gas a temperatura elevada, que después pueden ser utilizados, por ejemplo, en un sistema convencional de turbina-generador. En algunos Reactores el agua hierve dentro del núcleo del propio reactor, de manera que se puede utilizar directamente el calor generado en la fisión para producir vapor.

La conversión de calor en energía eléctrica se realiza en tres etapas:

1. La energía del combustible se utiliza para producir vapor a elevada presión y temperatura.
2. La energía del vapor se transforma en movimiento de una turbina.
3. Finalmente el giro del eje de la turbina se transmite a un generador que produce energía eléctrica.

En la Figura 1.1, se muestran los componentes de una Central Nuclear.

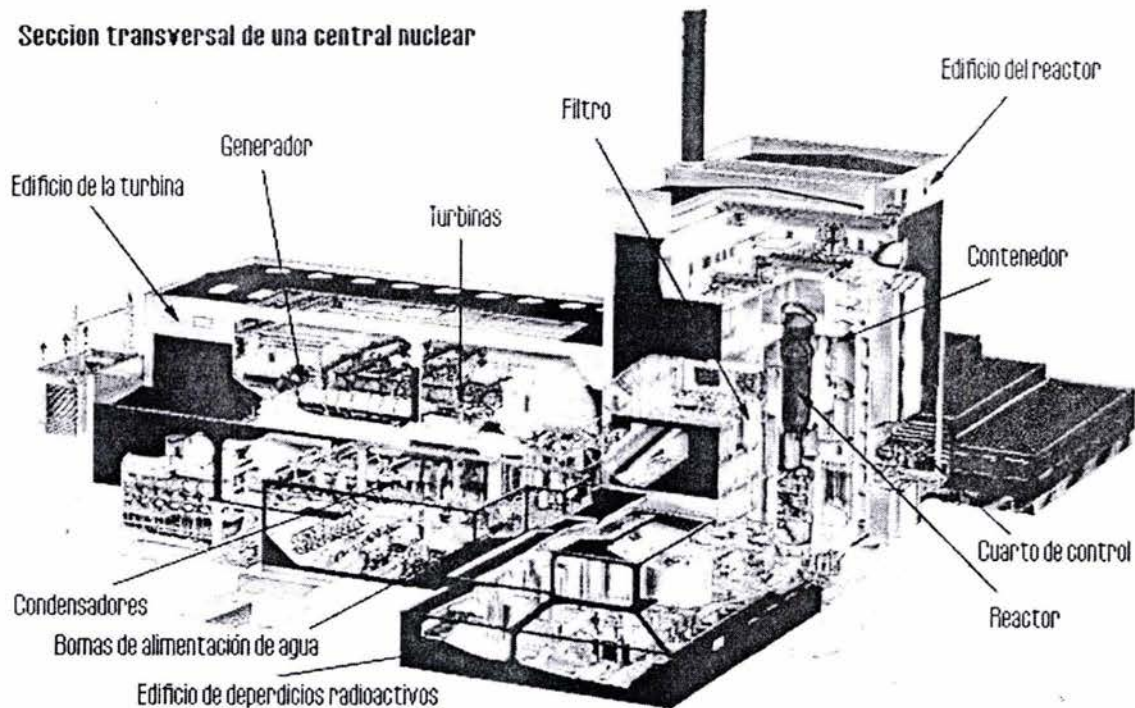


Fig.1.1 Elementos que componen una Central Nuclear.

Las diferentes combinaciones entre combustibles, moderadores y refrigerantes dan lugar a los diversos tipos de reactores. A continuación se mencionan los más comunes.

- Reactores de Agua Ligera a Presión (Pressurized Water Reactor - PWR).

En este tipo de reactor, los ensamblajes de combustible se encuentran dentro de una vasija a presión, llena de agua ligera, que desempeña el papel de moderador y refrigerante, y a pesar de la elevada temperatura que reina en su interior, no entra en ebullición debido a la presión interna en la vasija. Ver Figura 1.2.

El agua caliente se extrae del reactor y se envía al generador de vapor, que no es más que un intercambiador de calor, donde el agua, cede gran parte de su energía calorífica a otro volumen del mismo líquido para después retornar al reactor. Por su parte el agua que fue calentada en el generador de vapor, entra en ebullición, produciéndose así el vapor que sirve para mover al grupo turbogenerador, para que posteriormente sea condensado por un tercer circuito de agua procedente de un lago, río o una torre de refrigeración.

Este tipo de reactor utiliza como combustible uranio enriquecido.

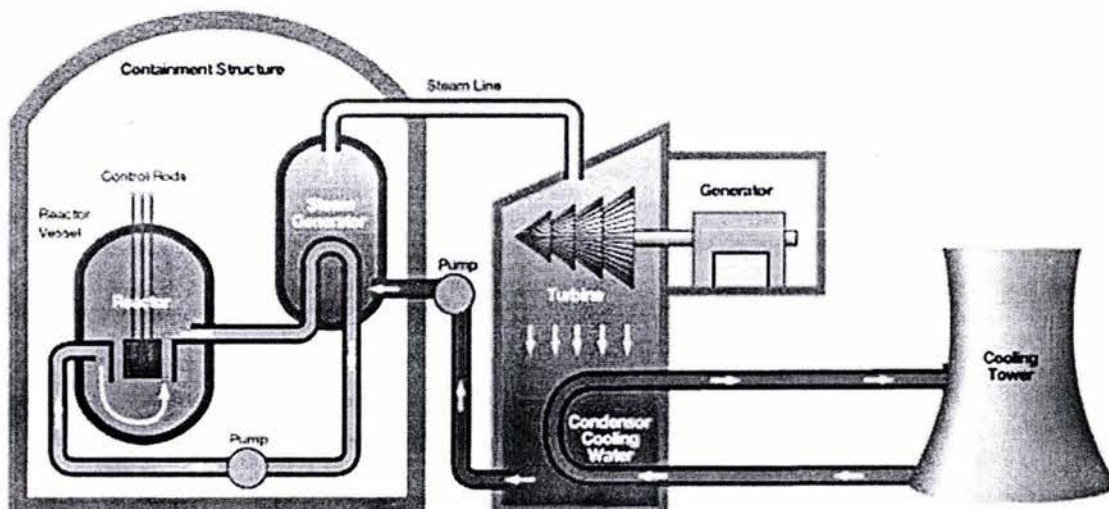


Fig.1.2 Diagrama de un Reactor PWR.

- Reactores de Agua Ligera en Ebullición (Boiling Water Reactor - BWR).

Se asemejan mucho a los PWR, ya que también emplean agua ligera como moderador y refrigerante, y como combustible uranio enriquecido. Ver Figura 1.3.

La diferencia estriba en que en los reactores BWR el agua entra en ebullición en el interior de la vasija, produciéndose así directamente, el vapor que se utilizará para mover el turbogenerador; el vapor posteriormente es condensado por el agua de enfriamiento procedente de una fuente independiente como un lago, río o el mar, y regresado en forma de agua caliente al reactor para repetir el ciclo.

Este tipo de reactor es el utilizado en Laguna Verde.

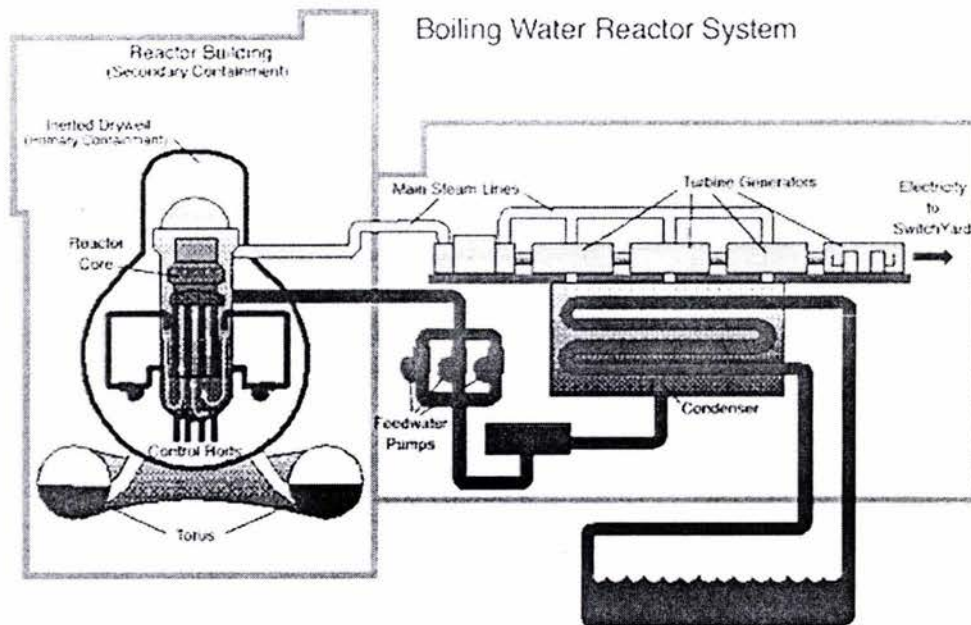


Fig.1.3 Diagrama de un Reactor BWR.

- Reactores de Agua Pesada a Presión (Pressurized Heavy Water Reactor - PHWR o CANDU).

La principal característica de este reactor, consiste en que utiliza uranio natural como combustible y agua pesada como moderador y refrigerante. Ver Figura 1.4.

El núcleo del reactor, se encuentra contenido dentro de un cilindro lleno de agua pesada denominado "calandria", el cual está atravesado axialmente por tubos llamados "tubos de presión", que contienen los ensambles de combustible. Por dentro de los tubos, circula agua pesada cuya temperatura, se eleva considerablemente sin llegar a entrar en ebullición, debido a la elevada presión que prevalece en el interior de los tubos.

El agua pesada caliente pasa a continuación a los generadores de vapor, en los que transfiere gran parte de su energía térmica a otro circuito que contiene agua ligera, regresando posteriormente al reactor, para continuar el proceso de refrigeración del mismo.

Por su parte el agua ligera al ser calentada entra en ebullición, produciéndose el vapor destinado a mover el turbogenerador; este vapor, como en los casos anteriores, es posteriormente condensado y regresado como agua caliente al generador de vapor, para repetir el ciclo.

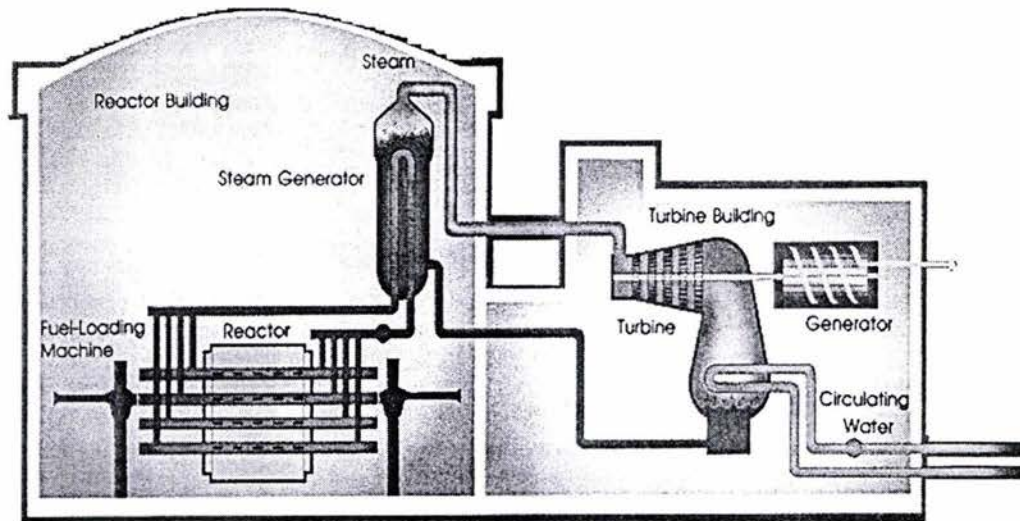


Fig.1.4 Diagrama de un Reactor PHWR.

- Reactores Enfriados por Bióxido de Carbono y Moderados por Grafito (GAS COOLED REACTOR - GCR).

A diferencia de los anteriores, este sistema no utiliza agua como enfriador, si no Bióxido de Carbono (CO_2); emplea grafito como moderador y uranio natural en forma metálica como combustible. Ver Figura 1.5.

En estos reactores la vasija está sustituida por un contenedor de concreto de gruesas paredes. El núcleo está formado por una gran cantidad de ensambles de combustible, localizados en el interior de una pila de bloques de grafito, por la que atraviesan los ductos en los que circula el gas refrigerante que es el Bióxido de Carbono.

Este gas arrastra el calor generado por la reacción nuclear y al circular posteriormente por el serpentín de un intercambiador de calor lleno de agua eleva la temperatura de ésta haciéndola hervir.

El vapor producido es conducido al turbogenerador para generar la energía eléctrica, posteriormente es condensado y regresado al intercambiador de calor, para repetir el ciclo.

El gas CO_2 , después de ceder gran parte de su calor al agua, es recirculado de nuevo a través del núcleo para mantenerlo refrigerado.

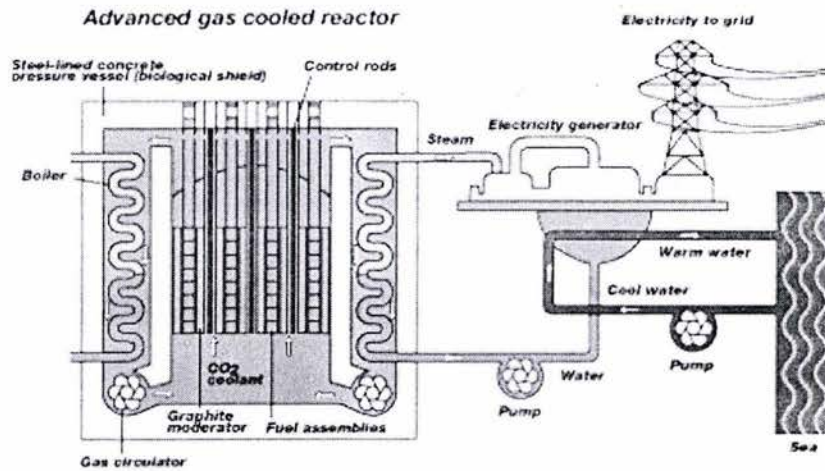


Fig.1.5 Diagrama de un Reactor GCR.

- Reactores Rápidos de Cría Enfriados por Sodio (LIQUID METAL FAST BREEDER REACTOR - LMFBR).

Este reactor funciona con neutrones rápidos, es decir, de elevada energía. Con los neutrones rápidos se puede aprovechar la propiedad que tiene el uranio 238 de convertirse en plutonio 239 mediante la captura de un neutrón libre, dando lugar a que de un núcleo no fisionable se obtenga otro que sí lo es y que por lo tanto puede ser utilizado en la manufactura de combustible. Este reactor utiliza combustible enriquecido al 20%, ya sea con uranio 235 o plutonio 239 y sodio líquido como refrigerante. Su peculiaridad es que el núcleo se rodea con un manto de uranio natural empobrecido, que al absorber neutrones poco moderados, se transforma en plutonio y de esta manera genera nuevo combustible. Ver Figura 1.6.

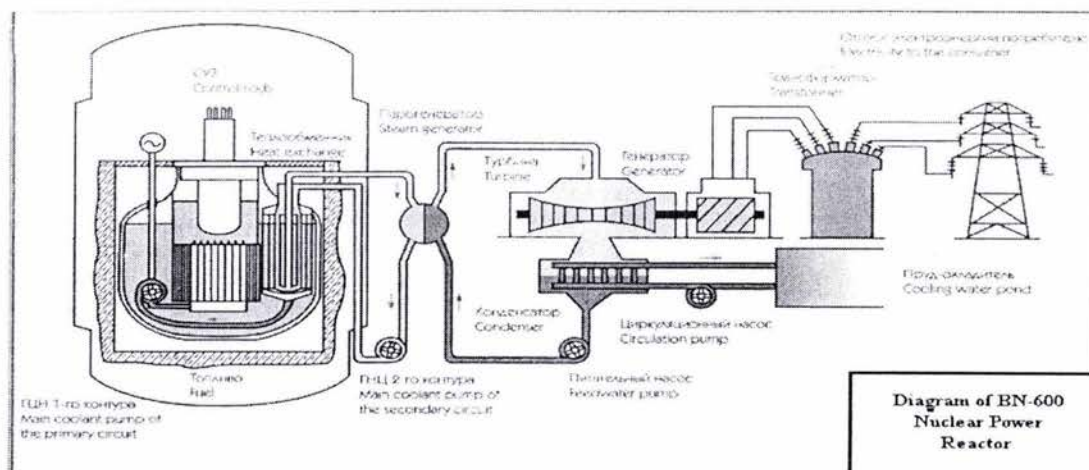


Fig.1.6 Diagrama de un Reactor LMFBR.

1.2. Central Nuclear de Laguna Verde.

En México, la producción de energía eléctrica por medios nucleares, se inició formalmente en julio de 1990, con el arranque de la operación comercial de la unidad No.1 de Laguna Verde, cuya capacidad es de 682.44 MWe (Mega Watts eléctricos), la capacidad de la central se duplicó en abril de 1995, al entrar en servicio la unidad No.2.

En la siguiente figura, se muestra una vista de la Central Nuclear de Laguna Verde.

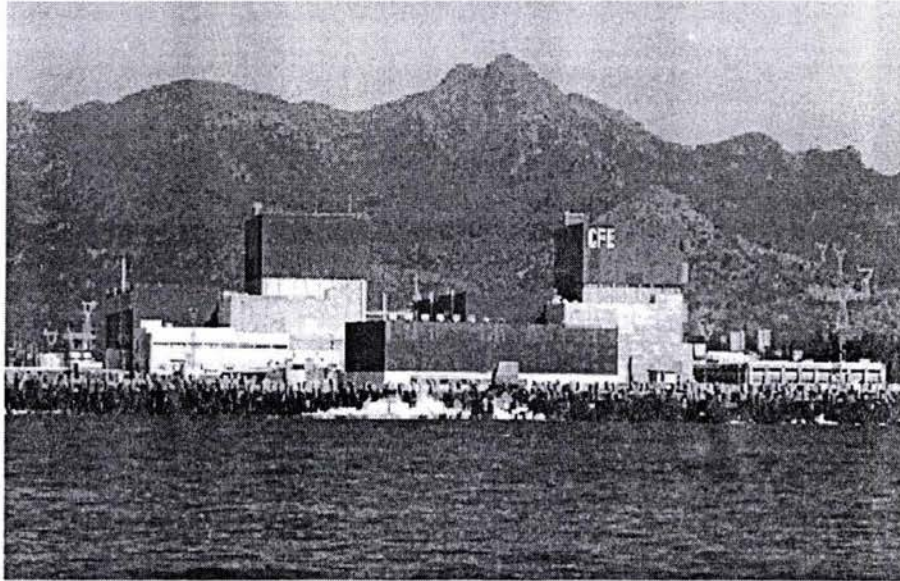


Figura 1.7 Vista de la Central Nuclear de Laguna Verde.

Su producción anual de energía eléctrica, que es del orden de 9564 GWh (Giga Watts hora), equivale al 6% de la producción total del país (2).

El reactor de la Central Nuclear de Laguna Verde es de tipo Boiling Water Reactor (BWR), el cual utiliza agua natural como moderador y enfriador, y uranio enriquecido como combustible.

En este tipo de reactor los elementos combustibles se encuentran dentro de una vasija a presión llena de agua. El agua entra en ebullición dentro de la vasija, produciéndose el vapor que se utiliza para mover el turbogenerador.

En la Figura 1.8 se muestra el diagrama de la Central Nuclear de Laguna Verde.

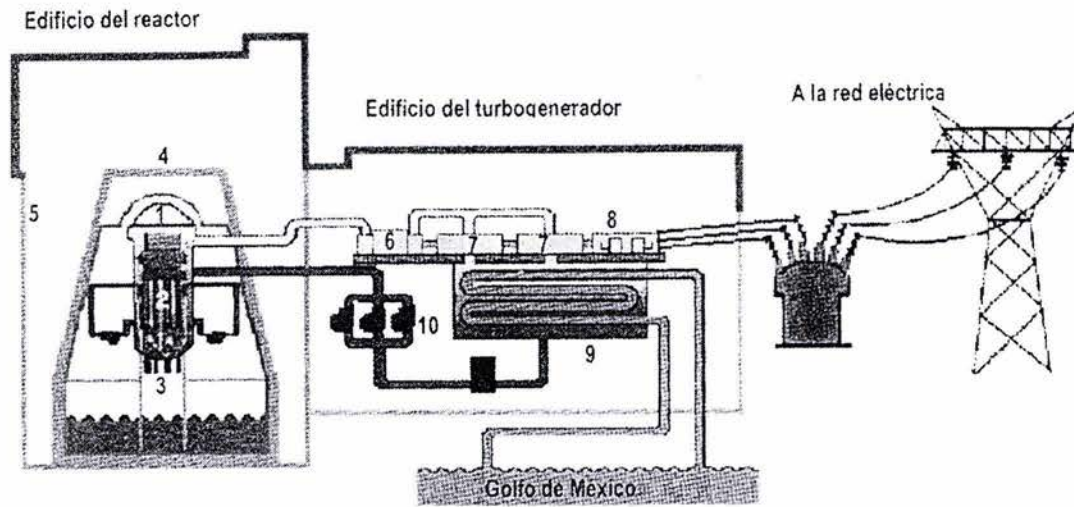


Figura 1.8 Diagrama de la Central Nuclear de Laguna Verde.

La Central Nuclear de laguna Verde consta de seis edificios principales. Ver Figura 1.9.

1. Edificio del Reactor.

En él se encuentra el reactor nuclear, sus sistemas auxiliares y los dispositivos de seguridad, la plataforma de recambio de combustible y la alberca de almacenamiento de combustible gastado.

2. Edificio del Turbogenerador.

Aloja a las turbinas de alta y baja presión, el generador eléctrico y su excitador, el condensador, los precalentadores de agua de alimentación y los recalentadores de vapor.

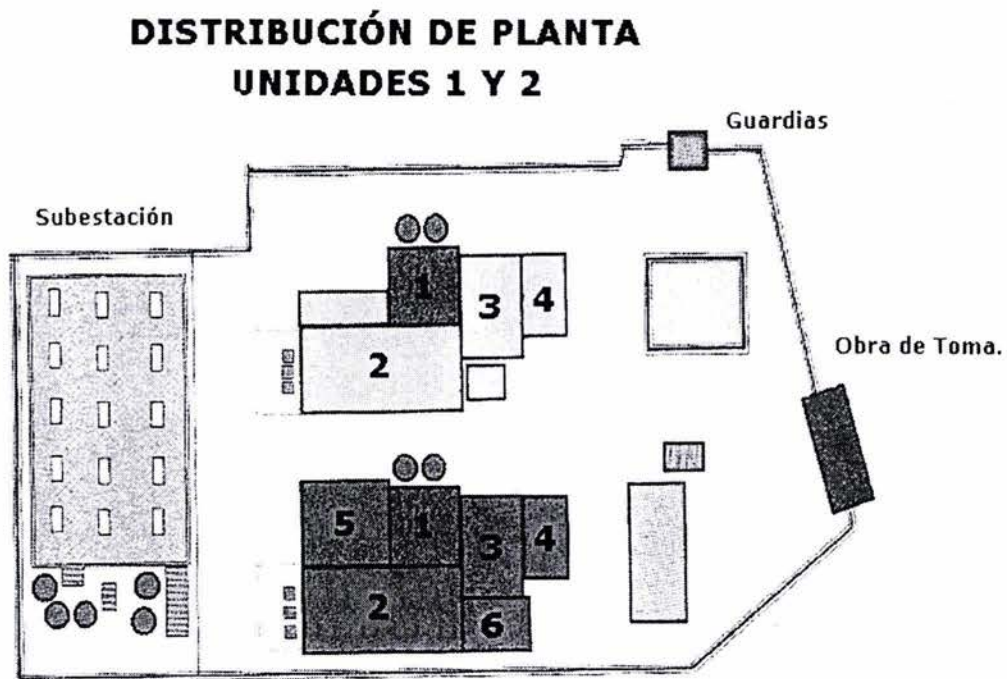
3. Edificio de Control.

En su interior está el cuarto de control y las computadoras de proceso, cuartos de cables, el sistema de aire acondicionado, el banco de baterías, los laboratorios radioquímicos y el acceso al personal del edificio del reactor.

4. Edificio de Generadores Diesel.

Aloja a los tres generadores diesel que se usan para el suministro de energía eléctrica a los sistemas de refrigeración de emergencia en caso de pérdida de energía exterior.

5. Edificio de Tratamiento de Residuos Radioactivos.
Aloja a los sistemas de tratamiento de residuos sólidos, líquidos y gaseosos de mediano y bajo nivel de radioactividad.
6. Edificio de la Planta de Tratamiento de Agua y Taller Mecánico.
Contiene la planta de agua desmineralizada de alta pureza para su uso en el ciclo de agua-vapor. También contiene el taller mecánico usado para reparación de equipos y mantenimiento.



1. Edificio del Reactor.
2. Edificio del Turbogenerador.
3. Edificio de Control.
4. Edificio de Generadores Diesel.
5. Edificio de Tratamiento de Residuos Radioactivos.
6. Edificio de la Planta de Tratamiento de Agua y Taller Mecánico.

Figura 1.9. Diagrama esquemático de la Central Nuclear de Laguna Verde.

1.3. Simuladores.

Un simulador es un programa o conjunto de programas de computadora, así como diversos componentes de monitoreo y / o control que en su conjunto permite reproducir y / o predecir el comportamiento de un sistema del mundo real (2, 3).

En cualquier tipo de simulación, es posible identificar básicamente tres procesos fundamentales:

- a) Inicialización y proceso de entrada,
- b) ejecución del modelo o modelos del proceso simulado, y
- c) el proceso de salida o de respuesta del sistema.

Normalmente estos tres procesos se ejecutan en forma iterativa hasta alcanzar un estado o comportamiento satisfactorio del sistema simulado.

Algunas de las ventajas de simular son:

- Ofrece la posibilidad de repetir, en condiciones idénticas y a partir de su modelación, procesos y fenómenos difíciles de lograr en condiciones reales y por tanto, estudiar sistemáticamente sus comportamientos.
- Elimina los riesgos que presenta la interacción con la realidad, tanto para dispositivos, instrumentos, etc.
- Permite la realimentación inmediata, pues los efectos que se logran en el funcionamiento del sistema, fenómeno o proceso que se simula, como resultado de introducir modificaciones en determinados parámetros, resultan inmediatos.

Los simuladores juegan un papel fundamental en los programas de entrenamiento para los operadores que laboran en las plantas de energía nuclear. El uso de simuladores especializados como herramientas de análisis es un complemento para un entrenamiento óptimo y más eficiente.

El objetivo de un entrenamiento empleando simuladores es asegurar que los operadores adquieran y practiquen habilidades específicas relacionadas con la operación segura y eficiente del sistema.

1.4. Simulador réplica de la Central Nuclear de Laguna Verde.

EL simulador réplica de la CNLV (un simulador réplica se caracteriza por reproducir el comportamiento y la distribución física de la instrumentación existente en el cuarto de control real de una central específica), está integrado por tableros de instrumentos idénticos al cuarto de control de la misma, así como por equipos y programas de cómputo que reproducen la operación y los efectos de las diversas acciones y maniobras de control que deben efectuar los operadores (5).

A continuación se presentan las ventajas que presenta el simulador réplica de la CNLV:

- Los operadores ejercitan las acciones y maniobras de control en una amplia gama de condiciones de operación normales y de emergencia.
- El entrenamiento se realiza en un cuarto de control idéntico al de la central.
- Es posible actualizar los escenarios de entrenamiento para incorporar las mejoras continuas que se realizan en los procedimientos de operación de emergencia y las actualizaciones de la propia planta. Esta modalidad, sin embargo, no es inmediata por que implica cambios en modelos y paneles.
- Se ahorran divisas al no tener que mandar a los operadores a adiestrarse al extranjero en simuladores que no tienen el mismo alcance ni la misma disposición de los instrumentos.

1.5. Prototipo del Simulador de Aula para la CNLV.

El uso de simuladores especializados y herramientas de análisis está en aumento como un complemento para un entrenamiento óptimo y más eficiente.

El concepto del Simulador de Aula se refiere a la representación y emulación de sistemas, componentes, paneles de control e instrumentación en diversos despliegues gráficos generados por computadora.

El desarrollo del Simulador de Aula se inició con el propósito de apoyar y entrenar al personal de operación de la Central Nuclear de Laguna Verde. El sistema facilitará la incorporación e integración de interfaces hombre-máquina a códigos computacionales para el análisis de seguridad de centrales nucleoelectricas; diseño y optimización de recargas de combustible nuclear; desarrollo, prueba y validación de modelos avanzados del reactor y estudios de inestabilidad. Así mismo, el sistema aumentará la capacidad para realizar estudios de sustitución de instrumentación convencional por sistemas digitales computarizados incorporando Ingeniería de Factores Humanos, y permitirá ampliar el campo de aplicación de nuevas técnicas computacionales asociadas a sistemas de información, diagnóstico y control, ambientes de realidad virtual y sistemas multimedia para educación a distancia.

La interfaz del operador es del tipo de manipulación directa, que simulan los paneles de control. Es una representación virtual de los instrumentos y controles cuya funcionalidad es similar a los del simulador réplica actual. Por medio de la manipulación directa, el sistema

permite al operador visualizar y manipular la instrumentación vía interacción con pantallas sensibles al tacto (touch-screen), el ratón o el teclado (6).

En la siguiente figura 1.10, se observa una vista del Simulador de Aula.



Figura 1.10. Simulador de Aula.

Actualmente el sistema contiene una representación de los paneles BB-09, BB-10 y BB-11, vasija del reactor y contención primaria, gráficas de tendencia, y despliegues de balance de planta de la CNLV.

Los paneles contienen la instrumentación asociada a los sistemas de refrigeración de emergencia, agua de servicio nuclear, recirculación e instrumentación y control nuclear. En la siguiente figura 1.11, se observan algunos ejemplos de los despliegues gráficos del Simulador de Aula.

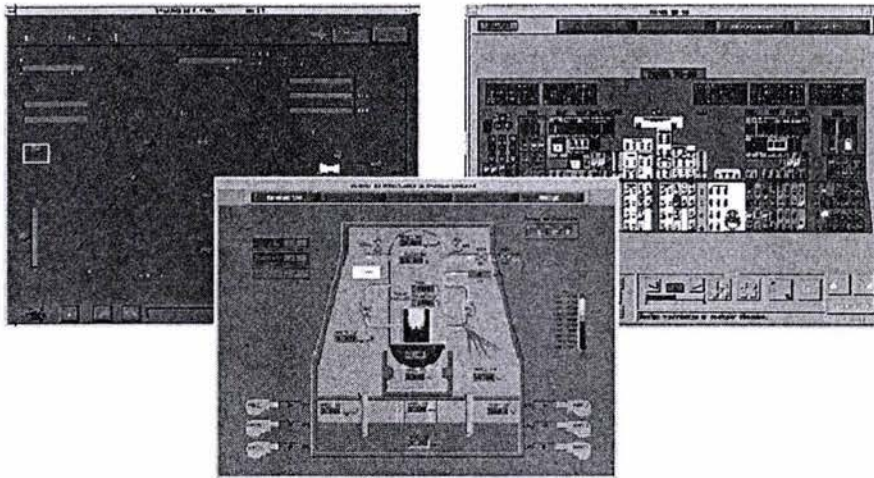


Figura 1.11. Despliegues gráficos del Simulador de Aula

1.6. Códigos RELAP / SCDAP.

Dentro del Simulador de Aula se ha implementado de manera dinámica, que los procesos nucleares involucrados sean controlados mediante los códigos nucleares RELAP / SCDAP, los cuales emplean modelos altamente especializados para representar los principales fenómenos físicos que se llevan a cabo en las Centrales Nucleoeléctricas.

El código RELAP / SCDAP se desarrolló en el Laboratorio Nacional de Energía y Ambiente de Idaho (Idaho National Engineering and Environmental Laboratory, INEEL) bajo el patrocinio de la Oficina de Investigación Reguladora Nuclear de la Comisión Reguladora Nuclear de los Estados Unidos NRC (Office of Nuclear Regulatory Research of the U.S. Nuclear Regulatory Commission) (7).

El código RELAP5 / SCDAP es el resultado de la unión de los modelos que simulan los códigos RELAP5 / MOD3.2 y SCDAP. Los modelos del código RELAP5 calculan la respuesta termo-hidráulica de los principales sistemas de la planta, interacciones del sistema de control, cinética del reactor, y transporte de gases no condensables. Los modelos del código SCDAP calculan el sobrecalentamiento y la progresión del daño de las estructuras del núcleo y base de la vasija del reactor.

El código RELAP / SCDAP fue desarrollado como código de mejor estimación en la simulación de los sistemas de refrigeración en reactores de agua ligera durante eventos transitorios y / o secuencias de accidentes severos. El código modela el comportamiento en

conjunto del sistema refrigerante del reactor, núcleo del reactor, liberación de productos de fisión durante un accidente severo, así como accidentes de pérdida de refrigerante por ruptura (Loss Of Coolant Accidents, LOCA's), pérdida de potencia del sistema, pérdida de agua de alimentación, pérdida de flujo de recirculación y transitorios operacionales tales como transitorio anticipado sin SCRAM (Anticipated Transient Without SCRAM, ATWS). El término SCRAM (Safety Control Rod Axe Man) es usado como referencia a la inserción súbita de las barras de control para el paro en un reactor nuclear, especialmente en una emergencia.

El código incluye diversos modelos de componentes genéricos. Los modelos de estos componentes incluyen las barras de combustible, las barras de control, bombas, válvulas, tubos, estructuras de calor, cinética puntual del reactor, calentadores eléctricos, bombas de chorro, turbinas, separadores, acumuladores, y componentes del sistema de control.

Se incluyen procesos especiales para efectos tales como pérdida de la geometría del núcleo, variación del flujo por cambio repentino de área, ramificación de flujo, flujo estrangulado, seguimiento de concentración de boro, y transporte de gases no condensables.

El modelo del núcleo incluye, el sobrecalentamiento de las barras de combustible, deformación y ruptura, liberación de productos de fisión, oxidación prematura, fundición del zirconio, disolución de UO₂, flujo y enfriamiento del encamisado y combustible fundido, formación, comportamiento y transporte del material fundido. El código también modela barras de control y sus componentes.

CAPÍTULO 2

DEFINICIÓN DE REQUERIMIENTOS

En este capítulo se describe el problema al que nos enfrentaremos, así como nuestra propuesta de solución, alcances, objetivos y los resultados esperados al finalizar el presente trabajo de Tesis.

Tomando lo anterior como punto de partida, se determinarán los requerimientos de nuestro sistema para su desarrollo e implementación.

2.1. Descripción del problema.

Actualmente la simulación de los procesos nucleares se realiza en una computadora independiente (Standalone) mediante códigos (RELAP / SCDAP) que han sido integrados de forma dinámica e interactiva a algunos de los despliegues gráficos del Simulador de Aula; los cuales permiten la simulación y un análisis muy preciso del desarrollo de los procesos en condiciones de operación normal, eventos transitorios y secuencia de accidentes severos.

El método usado en la integración del código al simulador, consiste en que múltiples procesos puedan acceder de manera independiente a un segmento de memoria, para compartir entre sí los valores o los datos generados por el código. Dichos valores representan parámetros de entrada o salida cuya finalidad es el monitoreo de información o de control respectivamente (5).

Los despliegues gráficos de información asociados a transitorios y gráficas de tendencia fueron modificados para acceder al mismo segmento de memoria compartida que genera el código, con lo cual es posible desplegar la información de las variables de interés que genera el código RELAP / SCDAP en tiempo de simulación.

Para poder utilizar el código RELAP / SCDAP y simular una planta nuclear, es necesario formar un archivo de entrada. El código procesa la información contenida en el archivo de entrada en un formato establecido, en donde cada línea se identifica por un número y la información asociada a éste representa las condiciones de los componentes específicos del sistema como por ejemplo, bombas y válvulas. Dicho archivo de entrada puede contener los elementos necesarios para analizar cualquiera de los eventos dentro de un reactor.

El código RELAP / SCDAP genera archivos de salida, en los cuales, es posible analizar la información generada (variables y parámetros previamente definidos en el archivo de entrada) por el proceso de simulación.

Además, el despliegue de gráficos utilizados permite visualizar en forma integral los parámetros más importantes del reactor en forma simultánea lo que permite comprender un gran número de interacciones entre variables y componentes durante el evento transitorio simulado (ver Figura 2.1 y Figura 2.2).

Con el fin de complementar y extender los alcances del Simulador de Aula, se ha considerado, desarrollar un sistema capaz de visualizar los despliegues gráficos del Simulador de Aula desde cualquier computadora en cualquier parte del mundo, con la

finalidad de participar más activamente en programas de estudio de Ingeniería Nuclear en cualquier Universidad del Mundo, así como, en la capacitación del personal que labora en las Centrales Nucleares.

2.2. Planteamiento de objetivos.

Para saber cuáles serán los objetivos de nuestro sistema, nos serviremos de los objetivos fundamentales del Simulador de Aula, esto nos dará un parámetro para poder establecer los elementos que integrarán nuestro sistema así como los alcances del presente trabajo:

- Contar con un prototipo de Simulador de Aula con la finalidad de crear un ambiente de simulación gráfica con propósitos de entrenamiento y análisis de procesos nucleares.
- Contar con un sistema basado en computadora, integrado por hardware, software y modelos de simulación avanzados.
- Un sistema constituido por una interfaz gráfica, la cual consta de paneles virtuales de control, despliegues mímicos y pictóricos y gráficas de tendencia.

De donde podemos hacer dos conclusiones importantes:

Primero, el Simulador de Aula está dirigido al entrenamiento y capacitación del personal de operación que labora en las Centrales Nucleares; en específico de la Central Nuclear de Laguna Verde; segundo, el Simulador de Aula tiene como sus principales propósitos el análisis de procesos nucleares; y tercero, la participación en programas de estudio de Ingeniería Nuclear en las Universidades.

Teniendo esto como punto de partida, pasaremos ahora a describir el estado actual de desarrollo e implementación en la que se encuentra el Simulador de Aula.

Actualmente, el código RELAP / SCDAP ha sido incorporado al Simulador de Aula a los despliegues gráficos que se presentan a continuación:

1. Despliegue representando la Vasija del Reactor (6).

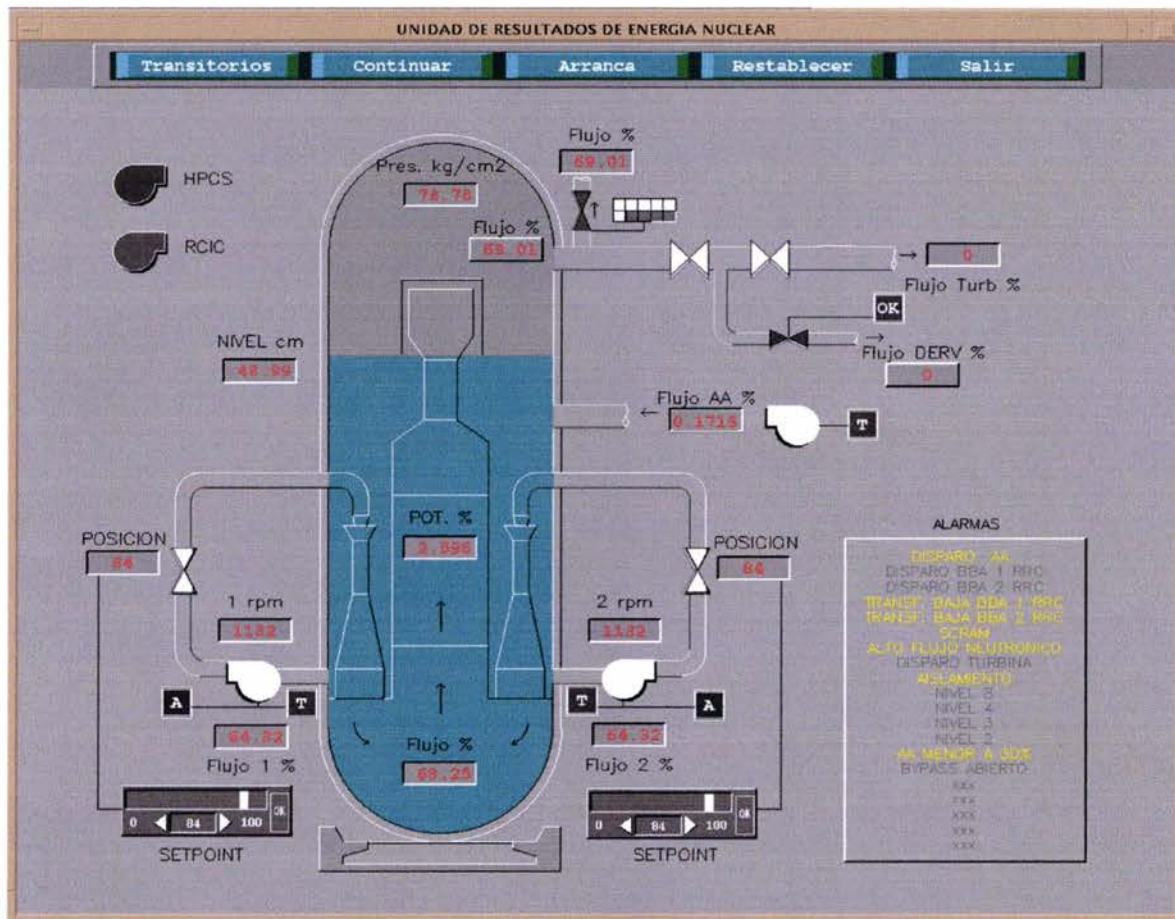


Figura 2.1 Vasija del Reactor.

2. Despliegue representando Gráficas de Tendencia de la Vasija del Reactor (6).

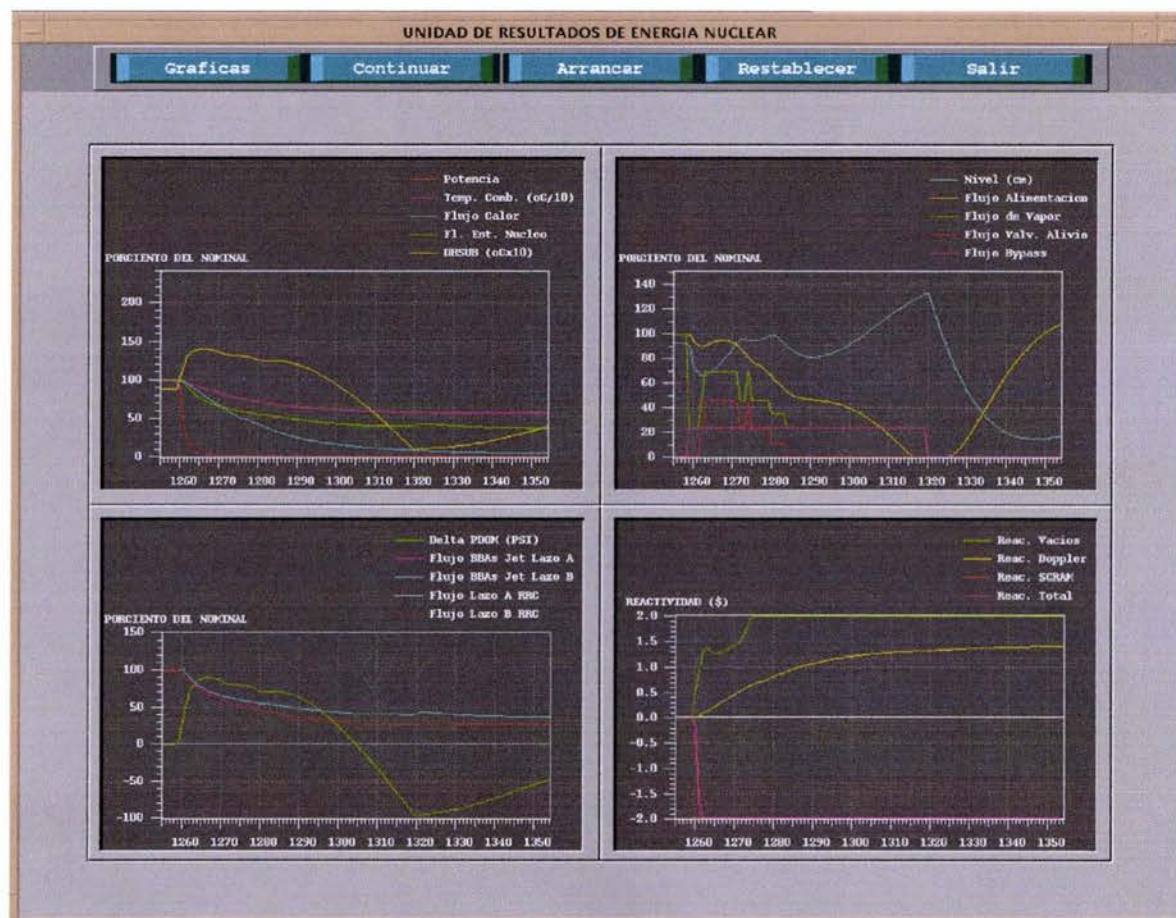


Figura 2.2 Graficas de Tendencia de la Vasija del Reactor.

Tomando en cuenta lo anterior, podemos continuar con nuestro trabajo.

Es importante plantear los objetivos y alcances, ya que el presente trabajo es sólo una etapa de todo un proceso de desarrollo e implementación que ha venido llevándose a cabo en el Simulador de Aula; considerándose que, posteriormente, se realizarán nuevas etapas en él.

El objetivo de la presente tesis es desarrollar y / o incorporar el software necesario para crear una interfaz gráfica prototipo para el entrenamiento a distancia utilizando el Simulador de Aula, mediante el uso de Internet.

Al finalizar la tesis se espera interactuar con el simulador de aula a distancia, utilizando Internet y como interfaz gráfica un navegador Web (Internet Explorer, Netscape, etc.), donde se mostrarán algunos de los despliegues más importantes del Simulador de Aula.

El reto del presente trabajo de tesis consiste entonces en analizar y seleccionar las herramientas para, diseñar e implementar un sistema capaz de desplegar las vistas del Simulador de Aula a través de Internet.

Es decir, que el personal que labora en la CNLV o analistas que se encuentren en lugares lejanos, tengan la posibilidad desde cualquier computadora, de interactuar con el Simulador de Aula. Requiriéndose específicamente para ello, tener una conexión Internet.

2.3. Especificación de requerimientos.

Pasaremos ahora a plantear cuáles son los requerimientos del sistema que desarrollaremos en cuanto a equipo (software y hardware) e instalaciones necesarias.

El Simulador de Aula está implementado como un sistema de cómputo, integrado por hardware, software y modelos de simulación, entonces, necesitaremos de equipo de cómputo, con características de hardware mínimas especificadas por el software utilizado por el propio Simulador de Aula.

Dentro de GRIN existe el hardware y software en donde se encuentran instalados los entornos de desarrollo y de programación del Simulador de Aula, requeridos para los procesos que involucran al sistema. Dentro de estos tenemos hardware como estaciones de trabajo Alpha y Computadoras Personales. En cuanto al software requerido, dentro del GRIN existen instalados software que varía según la plataforma que se opere; dentro de los

diversos sistemas operativos podemos contar con UNIX, LINUX, WINDOWS, así como software para diseño de gráficos DATAVIEWS (en el cual se encuentran desarrollados los despliegues gráficos del Simulador de Aula). Además se ha utilizado software para el proceso de comunicación; como son los editores y compiladores de los lenguajes de alto nivel Fortran 90, C, C++ y Java.

En este momento el Simulador de Aula utiliza memoria compartida para depositar las variables que generan los códigos nucleares, las cuales son necesarias para realizar la interfaz gráfica con los que interactúa el usuario. Se pretende utilizar del lado del Servidor (donde se encuentran los códigos nucleares), un lenguaje para manipular las variables (C) y un servidor de Web (Apache). Con el software desarrollado utilizando estos paquetes, se pretende manipular las variables generadas por los códigos nucleares y ponerlas a disposición del navegador en Internet.

Del lado del cliente se generarán las gráficas de interacción utilizando el paquete de desarrollo de interfaz gráfica DATAVIEWS (usando un control ActiveX); por lo tanto, será necesario tener instalado una versión del software DATAVIEWS (12).

DATAVIEWS es una herramienta de software que permite desarrollar despliegues gráficos. Estos despliegues son usados para controlar y monitorear procesos (12).

Los despliegues gráficos son creados a través de un editor gráfico, que permite crear interfaces de usuario, desde el diseño de despliegues gráficos simples hasta definir la interacción con el usuario; contiene además, librerías con rutinas gráficas escritas en diferentes lenguajes de programación (C, C++, Java y VisualBasic), que trabajan conjuntamente con el editor gráfico. Las rutinas permiten trabajar desde funciones de dibujo básicas, hasta funciones para manipular despliegues gráficos complejos; de esta forma se pueden integrar despliegues gráficos en poderosas interfaces gráficas para una amplia gama de aplicaciones.

Existen diferentes versiones de evaluación del software DATAVIEWS disponibles en Internet a través de la página Web de DATAVIEWS.

CAPÍTULO 3

DISEÑO

El presente capítulo tiene como objetivo describir detalladamente la arquitectura utilizada para el despliegue de información a través de Internet, así como una descripción detallada de los elementos que la componen. Además, se define la metodología a seguir para el despliegue de las Vistas del Simulador de Aula a través de Internet.

3.1. Arquitectura del sistema.

Para dar solución a nuestro problema nos hemos decidido por una arquitectura Cliente / Servidor. La arquitectura Cliente / Servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en elementos independientes que cooperan entre sí para intercambiar información, servicios o recursos.

Se puede decir que la arquitectura Cliente / Servidor es la integración distribuida de un sistema de red, con los recursos, medios y aplicaciones que definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los Clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e información; estableciendo así un enlace de comunicación transparente entre los que conforman la estructura (8).

Los componentes de nuestra aplicación Cliente / Servidor serán: el Cliente, el Servidor y la Infraestructura de Comunicación (ver Figura 3.1) que a continuación se explica.



Figura 3.1. Arquitectura Cliente / Servidor.

Cliente

El Cliente es la entidad por medio de la cual un usuario solicita un servicio, realiza una petición o demanda el uso de recursos al Servidor.

Este elemento se encarga, básicamente, de la presentación de los datos y / o información al usuario en un ambiente gráfico.

Frecuentemente se comunican con procesos auxiliares que se encargan de establecer conexión con el servidor, enviar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad.

Servidor

El Servidor es la entidad física que provee un servicio y devuelve resultados; ejecuta el procesamiento de datos, aplicaciones y manejo de la información o recursos.

En el servidor se realizará la parte destinada a recibir las solicitudes del cliente y donde se ejecutarán todos los procesos necesarios para el correcto funcionamiento de nuestro sistema.

En el servidor existirán procesos que se encargan de recibir las solicitudes del cliente, activar un proceso o los procesos necesarios para satisfacer el pedido, recibir su respuesta y enviarla al cliente.

Además, debe manejar ínter bloqueos, la recuperación ante fallas, y otros aspectos afines.

Por las razones anteriores, la plataforma computacional asociada con los Servidores es más poderosa que la de los Clientes; para el desarrollo del Servidor, se utilizarán las estaciones de trabajo Alpha, donde actualmente se encuentra instalado el Simulador de Aula.

Infraestructura de comunicación.

Para que los clientes y servidores puedan comunicarse se requiere de una infraestructura lógica que proporcione los mecanismos básicos de direccionamiento y transporte. Dicha infraestructura abarca a todo el software distribuido necesario para el soporte de interacciones entre Clientes y Servidores.

Sus funciones son:

- Independizar las dos entidades. El Cliente y el Servidor no necesitan saber comunicarse entre ellos.
- Traducir la información de una aplicación y pasarla a la otra: acepta consultas y datos recuperándolos de la aplicación cliente, los transmite y envía la respuesta de regreso.
- Controlar las comunicaciones: da a la red las características adecuadas de desempeño, confiabilidad, transparencia y administración.
- Permitir manejar diferentes ambientes de computación.

Entre las principales características que debe satisfacer nuestra arquitectura Cliente / Servidor, se pueden destacar las siguientes:

- El Servidor presentará a todos sus clientes una interfaz única y bien definida.
- El Cliente no necesitará conocer la lógica del servidor, sólo su interfaz externa.
- El Cliente no dependerá de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.

Nuestra aplicación tendrá tres funciones fundamentales:

1. Administración de los datos.
2. Lógica de la aplicación (procesos).
3. Lógica de la presentación (interfaz de usuario).

Los procesos se efectúan mediante el uso de los dispositivos que forman parte del hardware; a su vez los datos y programas que constituyen parte del software interactúan entre sí realizando las funciones lógicas necesarias para correr una aplicación, misma que genera un despliegue de información para el usuario (en nuestro caso una presentación en forma gráfica).

3.2. Detalle de módulos.

El software DATAVIEWS permite desplegar y animar Vistas en una arquitectura Cliente / Servidor; en donde el cliente será el navegador Web y en el Servidor Web además de las paginas Web, estarán localizados las Vistas del Simulador de Aula.

La siguiente figura 3.2 ilustra esta relación.

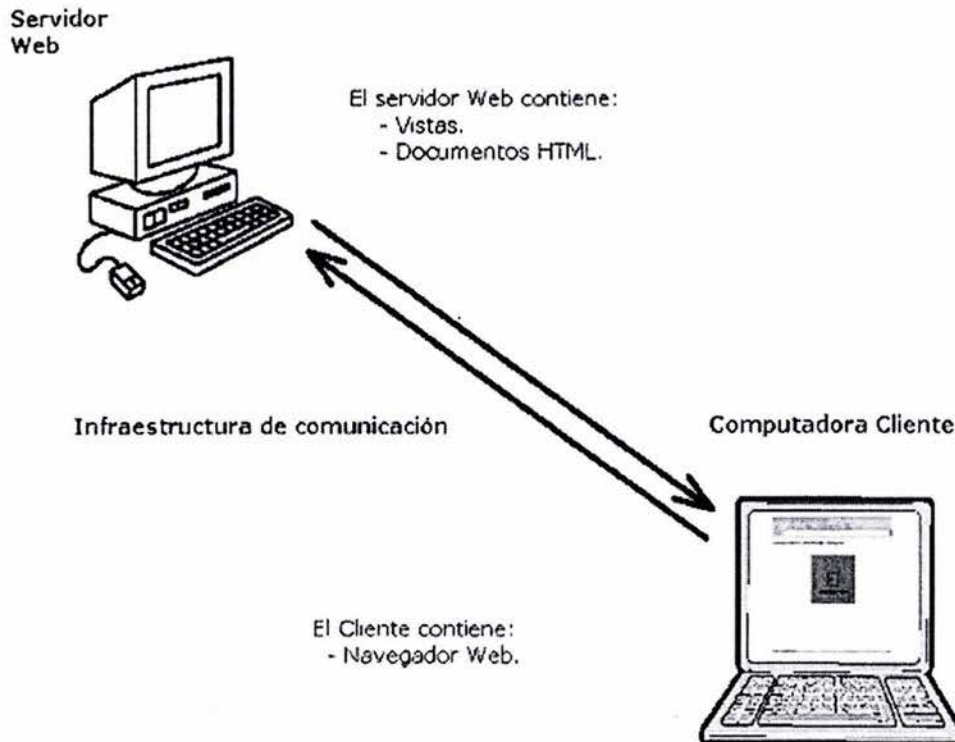


Figura 3.2. Relación entre DATAVIEWS y arquitectura Cliente / Servidor.

Esta relación permite monitorear datos desde prácticamente cualquier parte del mundo.

El software proporciona además funcionalidad para desplegar gráficas sofisticadas, el envío de datos dinámicamente a las gráficas y a objetos; y responder a eventos generados por la interacción del usuario a través de objetos.

Cliente.

El Cliente (navegador Web) desplegará una página Web que contiene una referencia a una Vista.

EL navegador Web puede encontrarse en cualquier máquina que tenga acceso a Internet o a una Intranet local.

El trabajo del navegador Web será dado una referencia hacia una porción de información en Internet, mejor conocida como URL (Uniform Resource Location), debe ser capaz de acceder a la información u operar de alguna forma basándose en los contenidos de dicha referencia. En el caso de los documentos Web, esto significa que el navegador Web debe ser capaz de comunicarse con el Servidor mediante el protocolo HTTP.

Cuando se accede a la página Web que contiene la Vista, se pueden usar objetos de entrada para acceder a otras vistas o desplegar otras páginas Web simplemente dando clic en objetos que se encuentran en la Vista.

Servidor.

El Servidor, será la base del desarrollo en el presente trabajo de Tesis. El Servidor deberá contar con los recursos necesarios para el correcto funcionamiento de los despliegues gráficos del Simulador de Aula, siendo el cliente, el encargado de la presentación de los datos a los usuarios.

Los recursos necesarios para el correcto funcionamiento de los despliegues gráficos incluyen páginas Web, archivos que contienen las Vistas, y además, la fuente de datos externos necesaria para el proceso de simulación.

El código RELAP / SCDAP funcionará como una fuente de datos externos; una fuente de datos externos es un programa ejecutable o un archivo de texto que alimenta comandos hacia la Vista a través del navegador Web. Estos comandos tendrán formato de texto y serán interpretados por la Vista en el navegador Web.

Los comandos que serán enviados establecerán un valor a una variable dentro de la memoria en la Vista. En nuestro caso la fuente de datos externos será un archivo de texto el cual contendrá los datos que genera el código RELAP / SCDAP.

Una vez que el Servidor cuente con los recursos anteriormente mencionados, el Servidor estará en posibilidad de procesar las peticiones hechas por el navegador Web y enviar la información necesaria (utilizando para ello el protocolo HTTP, HyperText Transfer Protocol), para que el navegador pueda presentar la información correctamente.

Infraestructura de Comunicación.

Los navegadores Web utilizan el protocolo HTTP, para comunicarse con los Servidores Web (9).

HTTP es un protocolo Cliente / Servidor que maneja el intercambio de información entre los clientes y los servidores Web.

HTTP se basa en sencillas operaciones de solicitud/respuesta. Un Cliente establece una conexión con un Servidor y envía un mensaje con los datos de la solicitud. El Servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado.

Cada solicitud contiene una dirección URL, que es una cadena que define un componente Web o un objeto como puede ser una página HTML, un archivo multimedia o una aplicación CGI (Common Gateway Interface).

Las principales características del protocolo HTTP son:

- La comunicación entre los Clientes y Servidores se realiza a partir de caracteres de 8 bits. De esta forma, se puede transmitir cualquier tipo de documento: texto, binario, etc., respetando su formato original.
- Permite la transferencia de objetos multimedia. El contenido de cada objeto intercambiado está identificado por su clasificación MIME.
- Existen tres métodos básicos que un Cliente puede utilizar para dialogar con el Servidor: GET, para recoger un objeto, POST, para enviar información al servidor y HEAD, para solicitar las características de un objeto.
- Cada operación HTTP implica una conexión con el Servidor, que es liberada al término de la misma.
- No mantiene estado. Cada petición de un Cliente a un Servidor no es influida por las transacciones anteriores. El Servidor trata cada petición como una operación totalmente independiente del resto.

Etapas de una transacción HTTP.

Cada vez que un Cliente realiza una petición a un Servidor, se ejecutan los siguientes pasos:

- Un usuario accede a una URL, seleccionando un enlace de un documento HTML o introduciéndola directamente en el campo Location del navegador Web.
- El navegador Web descodifica la URL, separando sus diferentes partes.
- Se abre una conexión TCP / IP con el Servidor.
- Se realiza la petición.
- El Servidor devuelve la respuesta al Cliente.
- Se cierra la conexión.
- Este proceso se repite en cada acceso al Servidor HTTP.

El sistema de protocolos TCP / IP (Transmisión Control Protocol / Internet Protocol) establece las bases de la mayor parte de Internet. El protocolo TCP se encarga de fragmentar la información en pequeños paquetes para después volverlos a juntar en un destino final. El protocolo IP tiene como función el verificar que estos paquetes vayan dirigidos correctamente hacia un mismo destino (10).

Tomando en consideración lo anterior podemos definir una serie de pasos que serán necesarios (metodología); lo cual nos permitirá desplegar las Vistas ya creadas y para las cuales se les ha sido integrado el código RELAP / SCDAP, así como también para otras Vistas que se encuentren actualmente en desarrollo:

1. *Generación del archivo de texto, el cual servirá como una fuente de datos externos.*
2. *Modificación de las vistas, para que puedan ser desplegadas dentro del Navegador.*
3. *Creación de la página Web, donde estará contenida la Vista.*
4. *Configuración del servidor Web.*

CAPÍTULO 4

DESARROLLO

En el presente capítulo se describe detalladamente la metodología definida para visualizar a través de Internet y utilizando un navegador Web, las Vistas del Simulador de Aula. Dicha metodología permite la incorporación de nuevas tecnologías utilizadas en Internet al Simulador de Aula para complementar y extender su funcionalidad.

4.1. Codificación de módulos.

A continuación se describe detalladamente la metodología utilizada para desplegar las Vistas del Simulador de Aula en Internet.

Generación del archivo de texto.

Para la generación del archivo de texto que servirá como fuente de datos externos, se tomó como base la arquitectura e implementación ya creada en la utilización de los despliegues.

El código RELAP / SCDAP (desarrollado en FORTRAN), utiliza bloques de memoria dinámica que corresponden a arreglos matriciales para depositar el valor de las variables utilizadas en el proceso de simulación. Conociendo el valor de los índices para cada una de las variables, se extrae entonces las variables de interés por medio de rutinas de lectura y escritura.

El código RELAP / SCDAP fue modificado para que el valor de las variables utilizadas en el proceso de simulación se escribieran a memoria compartida agregándose para ello funciones utilizando rutinas de lectura y escritura propias del lenguaje de programación C.

La rutina de lectura y escritura define una estructura que representa cada uno de los objetos que se tienen en una Vista (campos de un canal de información). La rutina se utiliza como enlace entre las variables del código, programado en FORTRAN, y las variables de la interfaz gráfica y los despliegues de información (Vistas) programados en C.

Los despliegues gráficos llamados Vistas se modificaron para acceder al mismo segmento de memoria compartida que genera el código por medio de la rutina descrita anteriormente. Dicho desarrollo ha sido implementado utilizando el Lenguaje de Programación C.

Para poder llevar a cabo nuestra integración, es necesario, en primer lugar identificar las rutinas de acceso a memoria compartida mencionadas anteriormente. Una vez identificadas estas rutinas es necesario modificarlas para que el valor de los datos leídos, sean guardados en un archivo de texto de forma secuencial.

Para poder insertar datos a un archivo de texto de forma secuencial, utilizando el lenguaje de programación C, es necesario, definir un apuntador que servirá como referencia al archivo de texto, por medio de este apuntador se puede definir el nombre del archivo, además del tipo de acceso para el mismo (lectura, escritura, sobre-escritura o combinaciones de estas). Una vez creado y referenciado el apuntador, se puede insertar o

0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
1.0
1.0
1.0
93.0
84.0
84.0
84.0
84.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
1.0
1.0
0.0
72.0
0.0

Lista 4.1. Valores contenidos en el archivo.

Los valores presentados se encuentran definidos dentro de un archivo de configuración de la Vista y representan los valores que contiene la Vista al iniciar el proceso de Simulación.

wvas	0.0
wgdm	100.0
waa	100.0
wder	0.0
Alarma1	0.0
Alarma2	0.0
Alarma3	0.0
Alarma4	0.0
Alarma5	0.0
Alarma6	0.0
Alarma7	0.0
Alarma8	0.0
Alarma9	0.0
Alarma10	0.0
Alarma11	0.0
Alarma12	0.0
Alarma13	0.0
Alarma14	0.0
Alarma15	0.0
Alarma16	0.0
Alarma17	0.0
Alarma18	0.0
Alarma19	0.0
Alarma20	0.0
rcc_bba_1	1.0
rcc_bba_2	1.0
aa_bba	1.0
znivel_dyn	93.0
pos_fcv_1	84.0
pos_fcv_2	84.0
fcv1	84.0
fcv2	84.0
srv1	0.0
srv2	0.0
srv3	0.0
srv4	0.0
srv5	0.0
srv6	0.0
srv7	0.0
srv8	0.0
srv9	0.0
srv10	0.0
hpcs	0.0
rcic	0.0
srvs	0.0
msiv	1.0
tsv	1.0

bypass_v	0.0
pdm	72.0
wturb	0.0

Lista 4.2. Relación entre variables y valores contenidos en el archivo.

Modificación de las Vistas.

Dado que las Vistas deberán ser una representación lo más cercana posible a las Vistas que se tienen actualmente, éstas no deberán sufrir modificaciones, sólo aquellas que sean necesarias para que puedan ser desplegadas por el navegador Web.

El uso de archivos en formato ASCII (American Standard Code for Information Interchange) es crítico en entornos de desarrollo multiplataforma por que los servidores Web pueden estar instalados en diferentes plataformas, así como también, los navegadores Web.

El software DATAVIEWS permite guardar archivos en dos tipos de formatos, binario y ASCII. Por medio de las opciones del editor gráfico de DATAVIEWS es posible configurar el formato en el cual las vistas se pueden guardar (Binario o ASCII).

Creación de la página Web.

El software DATAVIEWS permite configurar e interactuar con las Vistas que se desplegarán, a través de los siguientes mecanismos: propiedades, métodos y eventos (12).

- Propiedades.

Las propiedades controlan características como cuáles Vistas serán desplegadas, cómo serán desplegadas, así como el comportamiento en ciertos puntos mientras la aplicación esté corriendo.

Las propiedades de una Vista pueden ser asignadas en tiempo de diseño, el valor especificado durante este proceso es la opción inicial que tendrá la Vista al presentarse en pantalla. Todas las propiedades tienen valores iniciales por default, por lo que no es necesario asignar un valor a todas ellas, para una Vista en particular.

A continuación se presentan algunas propiedades:

- ViewFile. Especifica el nombre del archivo que contiene la Vista que será desplegada.
- AutoUpdate. Especifica si la vista se actualizará o no, en el intervalo indicado por la propiedad UpdatePeriod.
- UpdatePeriod. Especifica el intervalo de tiempo, en milisegundos, entre cada actualización de la Vista.
- ReadDataOnUpdate. Especifica si la Vista leerá de la fuente de datos, antes de cada actualización.

- Métodos.

Los métodos permiten actuar sobre las Vistas en tiempo de ejecución. Los métodos permiten modificar las propiedades definidas para una Vista en particular; los métodos son creados utilizando lenguajes de programación y su implementación varía de acuerdo al lenguaje seleccionado.

- Eventos.

Los eventos son generados por el Usuario al interactuar con la Vista. Un evento ocurre cuando el usuario da clic con el ratón en un objeto de la Vista. Por medio de las propiedades de la Vista se puede especificar los objetos que generan eventos.

Si la Vista está contenida dentro de una página Web, la Vista procesa los eventos como peticiones al Servidor. Sin embargo, no es necesario procesar todos los eventos generados por el usuario a través de peticiones al Servidor. Los eventos pueden ser procesados utilizando tecnología asociada al navegador Web, como son los lenguajes de scripts que se ejecutan en el navegador.

Para poder llevar a cabo este proceso, es necesario insertar la Vista en un archivo HTML, (HyperText Markup Language) para controlar el tamaño en que la Vista será desplegada y para alimentar la Vista con datos. El archivo HTML puede estar localizado en un servidor Web, en una máquina localmente o en una red.

HTML se basa en SGML (Standard Generalized Markup Language), que es un sistema que se utiliza para el procesamiento de documentos. SGML se utiliza para describir la estructura general de varios tipos de documentos (13).

HTML, debido a que es un subconjunto de SGML, es un lenguaje para describir documentos estructurados. Lo cual tiene su fundamento en que la mayoría de los documentos tienen elementos comunes. El lenguaje HTML, sirve para incrustar hipervínculos y describir la estructura lógica de documentos Web.

Un documento Web es un archivo de texto simple (ASCII) con comandos HTML incrustados que indican los elementos, estructura y formato del documento, así como vínculos a otros documentos Web.

El Lenguaje HTML se basa en etiquetas para definir la presentación de la información que será desplegada por el navegador Web. HTML define 3 etiquetas básicas que se usan para describir la estructura general del documento.

1. <HTML> es la primera etiqueta en cada documento HTML. Indica que el contenido del archivo se encuentra en lenguaje HTML. Todo el texto y etiquetas del documento deben estar dentro de las etiquetas <HTML> y </HTML>. (inicial y final respectivamente)
2. La etiqueta <HEAD> especifica que las líneas de los puntos inicial y final de la etiqueta son el prólogo del resto del archivo.
3. El resto del documento incluyendo todo el texto y otro contenido (imágenes, vínculos, etc.), se encierra dentro de la etiqueta <BODY>.

Para poder desplegar una Vista utilizando un navegador Web es necesario utilizar la tecnología ActiveX. ActiveX es una tecnología de software que posibilita y coordina la colaboración entre diferentes programas independientes (14).

El Control ActiveX es un objeto de programa que se utiliza, como su nombre lo indica, para controlar el desarrollo y especialmente las actividades del usuario en un programa. Los Controles ActiveX son más bien componentes del Sistema Operativo y están enlazados a programas de aplicaciones a través de puertos de comunicación. Con ActiveX es posible integrar completamente programas diferentes, eso significa que en el caso de una Vista, este no se ejecuta de forma separada o independiente si no se ejecuta dentro del navegador Web.

Los Controles ActiveX se incorporan a los documentos Web junto con los elementos HTML OBJECT y PARAM.

La etiqueta <OBJECT> se organiza para especificar la identificación del Control, qué información debe utilizar, los parámetros para la presentación, el comportamiento del Control y finalmente algunos parámetros del mismo.

La etiqueta <OBJECT> tendrá la siguiente forma:

```
<OBJECT ID="vassim" CLASSID="CLSID:5BDCB7B3-51BE-11D2-9D0F-000000000000"  
STYLE="POSITION:ABSOLUTE;TOP:8pt;LEFT:8pt;WIDTH:800pt;HEIGHT:600pt;">
```

En esta etiqueta se indica el nombre del Control que deseamos desplegar en la página Web en este caso "vassim". El atributo CLASSID identifica el tipo de objeto especificando el GUID, (Global Unique Identifier) el atributo GUID es una clave única utilizada para identificar el Control; además de configurar algunas propiedades del despliegue de la Vista dentro del navegador como es la posición y el tamaño de ésta (que se especifica dentro del atributo STYLE). El atributo CLASSID proporciona al navegador suficiente información para encontrar, identificar y ejecutar el control.

La mayor parte de los Controles ActiveX que se colocan en una página Web tienen atributos que se pueden modificar a los cuales a menudo se hace referencia como propiedades de control. Estas propiedades se pueden establecer usando la etiqueta <PARAM> dentro de la etiqueta <OBJECT>. Por ejemplo:

```
<PARAM NAME="ViewFile" VALUE="./views/vasija.view">
```

En esta etiqueta se define una propiedad llamada ViewFile la cual contiene la ruta del archivo (vasija.view) donde está contenida la Vista. De esta forma, es posible asignar un valor a las propiedades que contiene una Vista.

En el siguiente ejemplo se asigna a la propiedad AutoUpdate el valor de False:

```
<PARAM NAME="AutoUpdate" VALUE=False>
```

El cual, es el valor que tiene esta propiedad al presentarse la Vista en el navegador Web.

Por medio de funciones (realizadas en un lenguaje de scripts), es posible modificar el valor de las propiedades definidas para una Vista. Existen dos lenguajes ampliamente utilizados: JavaScript y VBScript.

Para poder insertar una función en la página Web que contiene la Vista, es necesario utilizar la etiqueta <SCRIPT>, de la siguiente forma:

```
<SCRIPT LANGUAGE="VBScript">  
  Sub StartButton_Click()  
    vassim.AutoUpdate = True  
  end sub  
</SCRIPT>
```

En el ejemplo anterior se define una función llamada StartButton_Click() utilizando VBScript (como se observa en el atributo LANGUAGE), que contiene únicamente una instrucción representada por la línea:

```
vassim.AutoUpdate = True
```

que modifica la propiedad AutoUpdate de la Vista identificada con el nombre vassim (en el atributo ID de la etiqueta OBJECT), asignándole el valor de True.

La función definida anteriormente puede ser ejecutada desde los elementos definidos por las etiquetas HTML, utilizando para ello el nombre de la función, como se observa a continuación:

```
<INPUT TYPE="BUTTON" NAME="ButtonStart" VALUE="Start"  
ONCLICK="VBScript:StartButton_Click()">
```

La etiqueta anterior, define un elemento HTML, que es un botón, el cual contiene la leyenda Start y usando el método ONCLICK, ejecuta la función StartButton_Click().

Así también, es posible crear una función StopButton_Click(), para cambiar el valor de la propiedad AutoUpdate a el valor False, que tenia originalmente.

```
vassim.AutoUpdate = False
```

De esta forma, es posible agregar funciones que permiten modificar las propiedades de la Vista, permitiendo cierta interacción entre los usuarios y las Vistas.

Configuración del servidor Web.

Si se desea publicar documentos en Internet, se necesita algún tipo de Servidor el cual proporciona a los navegadores Web los documentos y recursos que solicitan. Un servidor Web utiliza el protocolo HTTP para escuchar las solicitudes de los documentos por parte de los navegadores Web.

El servidor Web seleccionado para enviar los datos y que contendrá las Vistas es el Servidor HTTP Apache.

El Servidor Apache es un servidor HTTP de código abierto que trabaja en sistemas operativos como son UNIX, LINUX y WINDOWS. El servidor Apache ha sido el servidor Web más utilizado en Internet. (Más del 63% de los sitios en Internet en la actualidad usan Apache como su servidor Web) (15).

El Servidor Apache se vale de directorios para organizar la información así también como los diferentes sitios Web. Los documentos Web son colocados en directorios creados bajo el directorio "htdocs" dentro del directorio de instalación de Apache (16).

Dentro del servidor Web, nuestra aplicación tendrá la siguiente estructura:

/ vassim

El directorio "vassim" contendrá los archivos Web y el archivo que servirá como fuente de datos externos.

/ vassim / views

El directorio "views" contendrá además de las Vistas, los archivos necesarios para las mismas.

Para poder acceder a los documentos Web es necesario indicar al navegador Web el URL haciendo referencia al directorio vassim o un documento Web dentro de este directorio.

Antes de que se pueda desplegar una Vista en el navegador Web, se necesita indicar el contenido de los datos que está enviando hacia el navegador Web. Para una correcta comunicación entre el Cliente y el Servidor es necesario dar de alta la descripción MIME de las Vistas en el Servidor, para que reconozca que está enviando una Vista al navegador Web. El servidor reconoce el tipo de archivo que está enviando por su extensión. Cada extensión de archivo está asociado con su correspondiente tipo MIME.

Los recursos u objetos que actúan como entrada o salida de un comando HTTP están clasificados por su descripción MIME (Multipurpose Internet Mail Extension). De esta forma, el protocolo puede intercambiar cualquier tipo de dato, sin preocuparse de su contenido. La transferencia se realiza en modo binario, byte a byte, y la descripción MIME permitirá que el navegador Web trate adecuadamente los datos (12).

Es posible asociar más de un tipo de extensión de archivo, con la misma descripción MIME. Esta asociación es usualmente hecha en los archivos de configuración del Servidor Web.

La descripción MIME de las Vistas creadas con el software DATAVIEWS es:

```
text/x-dvweb
```

Los archivos de configuración de Apache se encuentran en el directorio "conf", dentro del directorio de instalación de Apache. El archivo de configuración que contiene la descripción de los tipos MIME es "mime.types", el cual es necesario modificar agregando la siguiente línea:

```
text/x-dvweb dvv dvweb v
```

De esta forma, se asocia las extensiones con que se guardan los archivos de las Vistas (.dvv, .dvweb y .v) a su tipo MIME en el Servidor, lo que permite una correcta comunicación entre el navegador Web y el Servidor.

Como se ha descrito anteriormente, se puede acceder a los recursos a través de Internet por medio de la dirección URL; indicándola directamente en el navegador Web, o en su caso, a través de las ligas (incrustadas en páginas Web) a estos recursos.

Cuando se hace referencia a una página Web que contiene una Vista, se crea una instancia del control ActiveX en el navegador Web. El control es el encargado de interpretar los datos que contiene la Vista, este control envía peticiones al Servidor para solicitar los recursos necesarios para el correcto funcionamiento de la Vista, como es, el archivo que contiene los datos para la simulación (fuente de datos externos).

Una vez que el Servidor ha terminado de enviar el archivo, el control está listo para presentar los datos y con esto empezar el proceso de simulación. El control lee los datos y asigna a cada objeto en la Vista el valor definido para éste en el archivo. El orden en que se

encuentran los datos, depende directamente de cómo fue definido en la etapa de diseño de la Vista y el cual corresponde al orden establecido en la estructura del canal de información.

Al terminar de asignar al último objeto su valor en la Vista, y presentar los valores en la Vista, el control interpreta los siguientes datos que contiene el archivo como los nuevos valores para cada objeto; el proceso anterior se repite hasta terminar de leer los últimos datos que se encuentran en el archivo.

El control es destruido (se liberan los recursos asignados a éste), en el momento en que se cierra la ventana del navegador Web o el usuario sale de la página Web que contiene la Vista.

4.2. Pruebas del sistema.

Los despliegues gráficos que verá el usuario, deberán corresponder o ser lo más cercano posible a los despliegues gráficos que se tienen actualmente. Para lo cual se deberá cambiar algunas de sus propiedades de las Vistas, como son el tamaño de la misma, la velocidad con que son leídos los datos, etc.

En las siguientes figuras 4.2 y 4.3 se observan los despliegues gráficos obtenidos de la aplicación utilizando Internet Explorer como navegador Web.

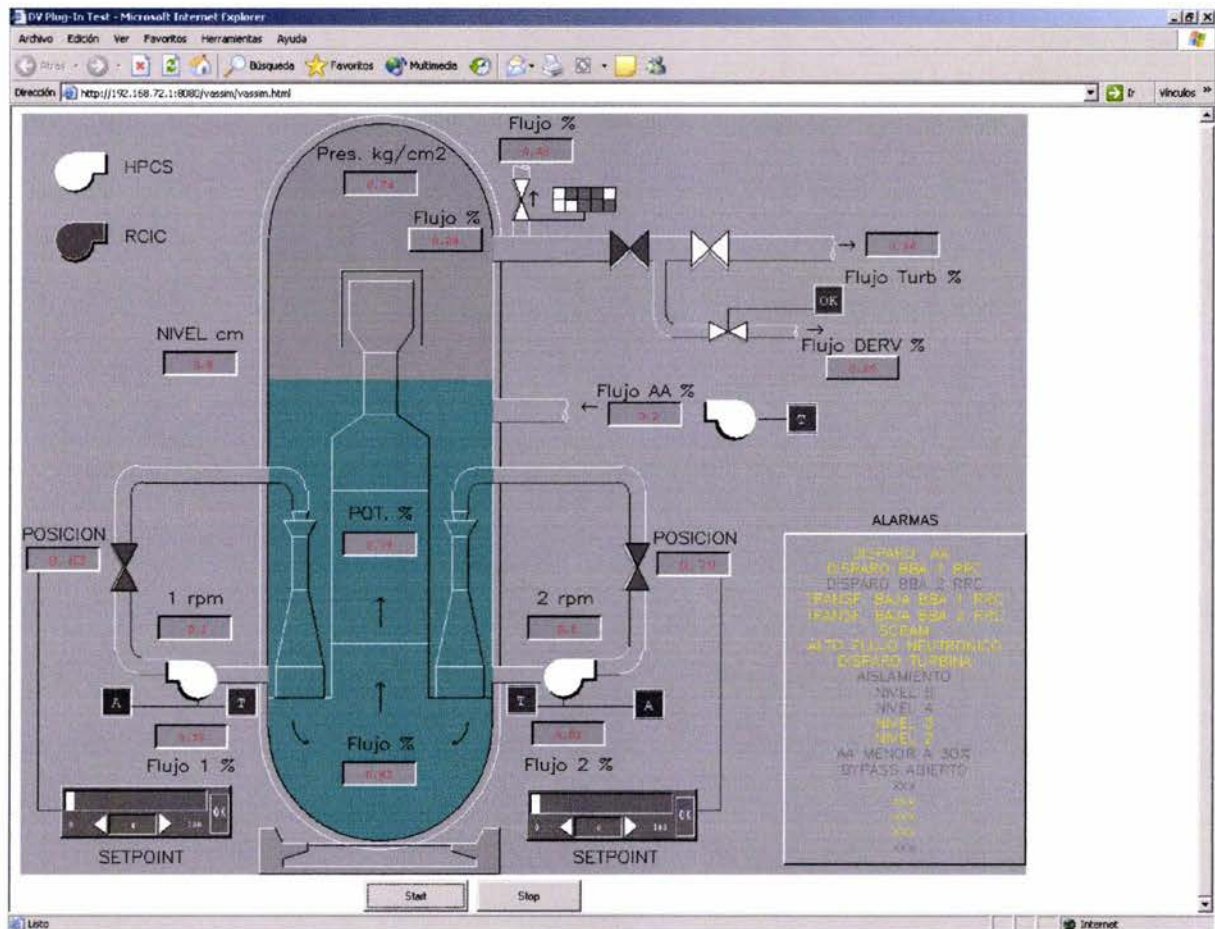


Figura. 4.2 Vasija del Reactor en Internet.

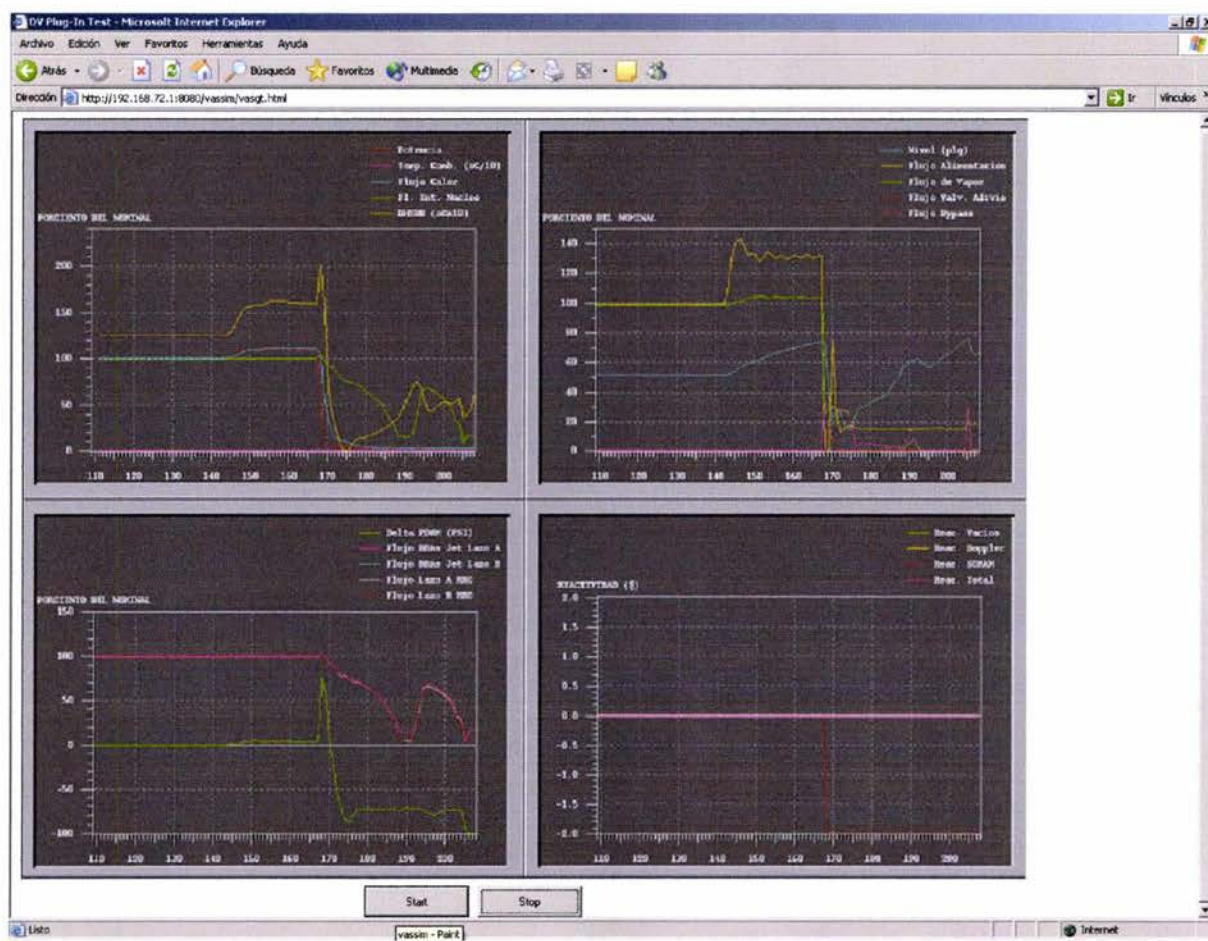


Figura. 4.3 Gráficas de Tendencia de la Vasija del Reactor en Internet.

Durante la etapa de desarrollo se decidió instalar el Servidor Apache, así como también la aplicación Web completa en una computadora Alpha, la cual utiliza como su sistema operativo UNIX; en donde además, se encuentra actualmente instalado todo el entorno de desarrollo del Simulador de Aula. (Despliegues gráficos, código RELAP / SCDAP, etc.)

Sin embargo, dentro del GRIN son utilizadas computadoras personales (las cuales utilizan diferentes versiones de WINDOWS como su sistema operativo) para el desarrollo de proyectos. Son en estas computadoras donde están instalados los programas Clientes (navegadores Web).

Con base en lo anterior, se considera importante la portabilidad de la aplicación, entendiéndose como portabilidad la facilidad de transferir las aplicaciones a diferentes entornos de hardware y software. Lo cual consiste en que los despliegues gráficos puedan estar colocados en cualquier computadora con cualquier sistema operativo y que estos además puedan ser vistos utilizando diferentes navegadores en distintos sistemas operativos.

La portabilidad no sólo debe ser vista desde un solo punto de vista ya sea del Servidor o del Cliente, si no de ambos, ya que debido a las características de Internet, estos dos elementos (Servidor y Cliente), pueden estar localizados en diferentes tipos de computadoras con diferentes Sistemas Operativos.

Por parte del Servidor se deberá garantizar que tanto las Vistas como la propia configuración del Servidor no sufran cambios a excepción de los expuestos anteriormente en este capítulo. Es decir que la metodología descrita será correcta no importando el Servidor que se utilice o del sistema operativo en el que se trabaje.

En el caso del Servidor Apache, existen versiones del mismo para diferentes sistemas operativos. Para cada una de estas versiones no existirá cambio alguno en la manera de cómo está organizado el sitio.

Por otro lado en la parte del Cliente, tenemos el inconveniente de que los diferentes Navegadores Web no siguen un estándar de cómo incorporan la tecnología Web.

Los Controles ActiveX pueden ser utilizados sin ningún inconveniente en las recientes versiones de Internet Explorer, lo cual garantiza que los despliegues gráficos podrán ser vistos utilizando el sistema operativo WINDOWS en cualquiera de sus recientes versiones. Sin embargo, los navegadores como Netscape utilizan plug-ins para poder desplegar

diferente tipo de información; el plug-in del software DATAVIEWS está disponible para las versiones de Netscape utilizando los sistemas operativos UNIX y LINUX.

Con estas dos opciones se logra cubrir un gran campo de sistemas computacionales que son los sistemas operativos más utilizados en la actualidad.

Como ya se ha dicho, las vistas que serán desplegadas en Internet podrán ser vistas por cualquier usuario que quiera hacer análisis y entrenamiento, en cualquier parte del mundo requiriendo para ello acceso a Internet.

Surge entonces el problema de no conocer específicamente los recursos con los que cuentan los posibles usuarios que accederán a las Vistas. Lo cual puede influir en un deterioro en el desempeño.

La tecnología utilizada del lado del Cliente permite mejorar el desempeño, ya que la tecnología que permite desplegar las Vistas se encuentra del lado del Cliente. En nuestro caso las Vistas se ejecutan del lado del Cliente utilizando la tecnología ActiveX.

Cuando un usuario accede a una página Web que contiene una Vista, es necesario enviar toda la información al navegador Web para que este pueda presentar correctamente los datos (archivos de las Vistas, fuente de datos externos, imágenes, etc).

Para evitar el envío excesivo de información, es necesario que los datos contenidos en el archivo que sirve como fuente de datos externos, reflejen la funcionalidad de la Vista utilizando la menor cantidad posible de datos. Esto evitará, que la Vista tarde demasiado tiempo en cargar los datos para su despliegue.

La metodología y tecnología descrita anteriormente en este capítulo, permite desplegar otros despliegues gráficos (creados o no para el Simulador de Aula), inclusive si estos despliegues aun no han sido integradas al código RELAP / SCDAP.

Esto es posible debido a que los archivos que contienen las Vistas y otros elementos de que componen la Vista tienen un formato estándar (ASCII); además los datos que son leídos del archivo de datos fuente simplemente están ordenados de forma secuencial sin ningún otro tipo de formato.

4.3. Análisis de resultados.

La presente Tesis pretende crear un sistema capaz de tener disponible en Internet parte del trabajo ya realizado en el Simulador de Aula; no se interpretarán los resultados obtenidos a detalle sino de forma general; poniéndose un mayor énfasis en el formato en el que se presenta la información (interfaz del usuario), así como su nivel de utilidad. Además de una valoración de los métodos y herramientas utilizadas durante el desarrollo.

El adaptar los despliegues gráficos con que cuenta el Simulador de Aula actualmente a los despliegues gráficos que se obtienen en el navegador Web, no fue necesario hacer modificaciones que interfirieran con el estado actual de desarrollo del Simulador de Aula; además de que las herramientas utilizadas permiten un uso adecuado de los recursos.

Los despliegues gráficos (como se puede constatar al comparar directamente las imágenes mostradas en los capítulos anteriores) son una representación exacta de los despliegues gráficos que se tienen actualmente. Además, los valores contenidos en el archivo "default.dat" (fuente de datos externos), corresponden a los valores presentados en el proceso de simulación, gracias a que estos valores son obtenidos directamente de este proceso.

Como se pudo constatar en el presente capítulo el sitio Web puede estar localizado en cualquier computadora no importando el sistema operativo, no obstante que, en dicha computadora, no se encuentre instalado el entorno de desarrollo del Simulador de Aula; los despliegues gráficos podrán estar disponibles, gracias a que los datos que serán desplegados están contenidos en un archivo, que a pesar de que, es generado por el Simulador de Aula, éste es independiente de aquél.

Tener un archivo que sirve como fuente de datos externos, conlleva a que los datos que serán desplegados podrán cambiar según las necesidades; es decir, que el detalle con que será descrito el transitorio o cualquier otro fenómeno que se desee simular, utilizando los despliegues gráficos, dependerá únicamente de los datos que se encuentren en el archivo.

CONCLUSIONES

Finalmente, el presente capítulo presenta las conclusiones relacionadas con el trabajo realizado en la presente Tesis, y presentando además, el trabajo futuro relacionado con el Simulador de Aula.

● Conclusiones sobre el presente trabajo.

La metodología descrita a través de los capítulos anteriores permite que los despliegues gráficos del Simulador de Aula se encuentren disponibles a los usuarios en Internet.

Los resultados obtenidos, además de cumplir con los objetivos de la Tesis, presentan una gran ventaja de complementar el desarrollo ya realizado en el Simulador de Aula.

Esto es posible gracias a que no se ha interferido con las etapas de desarrollo anteriores, lo que permitirá que las siguientes etapas de desarrollo no tengan que ajustarse a las nuevas incorporaciones realizadas en el presente trabajo si no en la etapa de desarrollo en el que se encontraba anteriormente. En las modificaciones se ha mantenido la integridad del código, de forma que los cálculos no son afectados por la manipulación de la información de salida.

Sin embargo, actualmente, las Vistas no contemplan el procesamiento de eventos generados por los usuarios. La interacción con los usuarios se limita a la presentación de las variables involucradas en los procesos de simulación.

● Trabajos futuros.

Como se ha mencionado anteriormente, las Vistas no han sufrido modificación alguna en su funcionamiento, por lo tanto, la interacción con los usuarios a través de Internet, se limita a los eventos creados con un lenguaje de scripts.

Para poder generar una mayor interacción con los usuarios es necesario procesar los eventos directamente en las Vista; esto se logra, modificando las propiedades, asociando a los objetos que contiene la Vista un recurso en el Servidor por medio del URL.

Una vez presentada en el navegador Web, la Vista podrá interactuar con el usuario utilizando las funciones creadas con el lenguaje de scripts, o enviando peticiones al Servidor.

Por otra parte, resulta necesario, realizar acciones encaminadas a mejorar el proceso de enseñanza a distancia. Esto es, la creación de todo un sitio Web el cual contenga los recursos (páginas Web, imágenes, archivos multimedia, etc.) para poder llevar a cabo una sesión de entrenamiento y / o enseñanza a distancia tomando como temática el trabajo realizado en el Simulador de Aula.

Por este motivo, es indispensable, crear páginas Web destinadas a la comprensión y explicación detallada de los fenómenos físicos que se presentan a través de las Vistas del Simulador de Aula (además de las páginas Web que contienen las Vistas), así como, la interacción permitida por la Vista, a través de los métodos definidos, además de los requerimientos de software necesarios para poder ver las Vistas. Esto permitirá, que los usuarios puedan conocer y aprovechar mejor la información que se presenta en los procesos de simulación.

Como parte integral del Simulador del Aula, se trabajará en el desarrollo de modelos específicos para la Central Nuclear de Laguna Verde; específicamente la vasija del reactor y sus sistemas asociados, el núcleo y la contención primaria que permita la simulación en operación normal, transitorios, accidentes postulados y eventos especiales, cuyos resultados serán presentados en despliegues gráficos, gráficas de tendencia y tableros virtuales con los que ya cuenta el Simulador de Aula.

Por otra parte se trabajará en el desarrollo del módulo interactivo de control para el código RELAP / SCDAP, el cual permitirá en tiempo simulación, la utilización de los despliegues de instrumentación virtual existentes para la manipulación de válvulas, bombas, lógica de disparos y demás componentes mediante pantallas sensibles al tacto.

Además, en una nueva etapa de desarrollo se pretende que las variables que modelan los diferentes fenómenos simulados, sean insertadas en una base de datos diseñada específicamente para este propósito.

Lo anterior permitirá que las variables estén disponibles no sólo para las aplicaciones con los que se cuenta actualmente (por medio de memoria compartida), si no incluso a aplicaciones en cualquier lenguaje de programación que posea la capacidad de comunicarse con la base de datos, utilizando para ello sentencias SQL (Standard Query Language).

Lo anterior permitirá que el Simulador de Aula pueda crecer hasta convertirse en un sistema muy completo y robusto. Así como también, se realicen nuevos ajustes a la metodología descrita en el presente trabajo de Tesis, para cubrir los requerimientos que se presenten, sin interferir con la etapa de desarrollo actual del Simulador de Aula.

BIBLIOGRAFÍA

1. García y García, Enrique. "Los Reactores Nucleares y la Producción de Electricidad". Ediciones del Sector Eléctrico. México. 1980.
2. Comisión Federal de Electricidad. "CFE – Generación". <http://www.cfe.gob.mx/lagver>. México. 2003.
3. Collacott, Ralph. "Simulators: International Guide". Addison-Wesley. USA. 1985.
4. Law, Averill; Kelton, David y otros. "Simulation Modeling and Analysis". McGraw-Hill. USA. 1999.
5. Salazar, Humberto. "Incorporación de los Códigos RELAP / SCDAP al Simulador de Procesos Nucleares para Análisis y Entrenamiento en Aula", Tesis de licenciatura. UNAM. México. 2002.
6. Zayas Pérez, B & Chávez Mercado, Carlos, "Interfaz Gráfica Hombre-Máquina Avanzada para un Simulador de Procesos Nucleares para Entrenamiento en Aula". México Nuclear, Revista de la Sociedad Nuclear Mexicana, año 2 No. 5. Abril-Junio. 1996.
7. The SCDAP/RELAP5 Development Team. "SCDAP/RELAP5/MOD3.2 Code Manual". Idaho National Engineering and Environmental Laboratory. USA. 1997.
8. Departamento de Control de Calidad y Auditoría Informática. "Sistemas en Arquitectura Cliente / Servidor". <http://sistemas.dgsc.unam.mx/publica/pdf/clienteservidor.PDF>. DGSCA-UNAM. 2001.
9. Carballar, José A. "Internet". Addison-Wesley. México. 2000.
10. Connner & Stevens, "Interconectividad de Redes con TCP-IP". Prentice Hall. México. 2001.

11. Kernighan, Brian & Ritchie Dennis, "The C Programming Language". Prentice Hall. USA. 1978.
12. General Electric, "DataViews", <http://www.gefanuc.com/dataviews/>.
13. Musciano, Chuck & Kennedy, Bill. "HTML, The definitive guide". O'Reilly & Associates. USA. 1996.
14. Sanders & Kaufman. "Programación con ActiveX". Prentice Hall. México. 1997.
15. Jez. "Web Server Survey".
http://news.netcraft.com/archives/2003/04/13/april_2003_web_server_survey.html. USA. 2003.
16. Apache Software Foundation, "Apache HTTP Server" , <http://httpd.apache.org/>. USA.

LISTA DE ACRÓNIMOS Y TÉRMINOS

A

ActiveX	Componentes de software que se puede insertar en una página Web para proporcionar una funcionalidad que no está directamente disponible en HTML.
Apache	Servidor HTTP, más usado en Internet.
ASCII	(American Standard Code of Information Exchange). Estándar mundial de 128 caracteres, letras, números y símbolos aceptados por la mayoría de editores y procesadores de texto
ANSI	(American National Standards Institute). Organismo norteamericano encargado de llevar estándares en la industria.
Arquitectura	Término que se refiere a la estructura general de un sistema.

B

Browser	Un programa que permite al usuario navegar el Web. Los más populares son Netscape e Internet Explorer.
Base de Datos	Conjunto de datos relacionados que se almacenan de forma que se pueda acceder a ellos de manera sencilla, con la posibilidad de relacionarlos, ordenarlos en base a diferentes criterios, etc.

C

CGI	(Common Gateway Interface). Interface de intercambio de datos estándar en WWW a través del cual se organiza el envío de recepción de datos entre visualizadores y programas residentes en servidores WWW
-----	--

Cliente	Cualquier elemento de un sistema de información que requiere un servicio mediante el envío de solicitudes al Servidor.
CNLV	Central Nuclear de Laguna Verde.
D	
DEPFI	División de Estudios de Postgrado de la Facultad De Ingeniería.
DIE-FI	División de Ingeniería Eléctrica – Facultad de Ingeniería
Default	Término designado a los valores que toma un sistema en ausencia de los valores especificados.
G	
GRIN	Grupo de Ingeniería Nuclear.
H	
HTML	(HyperText Markup Language). Lenguaje de programación que permite al usuario crear documentos de hipertexto para su publicación en Internet.
HTTP	(HyperText Transfer Protocol). Protocolo de comunicación de datos que permite la transmisión de documentos de hipertexto. Es el protocolo en el que está basado Internet.
I	
Internet	Red global que permite el intercambio de información.
Interfaz	Conexión física y funcional entre dos componentes de "hardware", entre dos aplicaciones o entre un usuario y una aplicación.
L	
LAIRN	Laboratorio de Análisis en Ingeniería de Reactores Nucleares.

M

MIME (Multipurpose Internet Mail Extension). Extensiones al que están asociados los archivos.

R

RELAP / SCDAP Códigos utilizados para simular algunos de los fenómenos que se presentan en las Centrales Nucleares.

S

SCRAM Safety Control Rod Axe Man.

Servidor Cualquier elemento de un sistema de información que recibe peticiones de los clientes y devuelve los resultados.

SimAula Simulador de Aula.

SQL (Standard Query Language). Lenguaje de sentencias que permite manipular los datos en una base de datos.

StandAlone Computadora independiente la cual no se encuentra en red.

T

TCP – IP (Transfer Control Protocol – Internet Protocol). Protocolos encargados de gestionar la comunicación entre dos computadoras en Internet

U

URL (Uniform Resource Location). Es la dirección estandarizada y detallada de páginas Web.

W

WWW (World Wide Web). Sistema de información basado en páginas que contienen hipertexto y gráficos.