



Universidad de las Ciencias
Informáticas
Facultad 3

SIPCSOL: Sistema para Cálculos Solares en Cuba

**Trabajo de diploma para optar por el título de Ingeniero
en Ciencias Informáticas**

Autor:

Aniel Perez Orozco

Tutor:

Ing. Pedro Manuel Alás Verdecia

La Habana, junio 2015
"Año 57 de la Revolución"

Declaración de autoría

Por este medio declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) para que haga el uso que estime pertinente.

Para que así conste firmo la presente a los ____ días del mes de _____ del 2015.

Autor:

Aniel Perez Orozco

Tutor:

Ing. Pedro Manuel Alás Verdecia

Datos de contacto

Datos del Tutor:

Nombre: Pedro Manuel Alás Verdecia.

Ingeniero en Ciencias Informáticas.

Correo electrónico: pmalas@uci.cu

Datos del Autor:

Nombre: Aniel Perez Orozco.

Correo electrónico: aporozco@estudiantes.uci.cu

"Suerte" es lo último que deseo de los que creen que la victoria se puede obtener por accidente.

...aunque miles de voces digan que no estás preparado, escucha esa voz solita que te dice ¡estás preparado! ¡Ahora depende de ti! Así que levántate y brilla."

Agradecimientos

- A mis padres, gracias por su amor y cariño, por sus mimos y sus regaños, por creer siempre en mí, porque gracias a su dedicación y esfuerzo hoy puedo alcanzar este logro.*
- A mis hermanos gracias por apoyarme siempre, por estar ahí cuando más los necesito, por soportar siempre mis malacrianzas y por dejarme ser el faro de esta familia.*
- A mis sobrinas, Daly y Dany que aunque no he podido estar a su lado en sus primeros años de vida, siempre las llevo presente, y que el solo saber que puedo ser un ejemplo para ellas me enorgullece.*
- A mis hermanas Misdre y Yaimy, gracias por su amistad incondicional, por cada consejo, nuestra amistad es un ejemplo de que no hay que tener la misma sangre para conformar una familia.*
- A mi tío José, gracias por tu apoyo y por haber sido un padre para mí en estos últimos años.*
- A mis tíos, mis primos y de forma general a toda mi familia que siempre se preocupan por mí y me ayudan a cumplir mis metas.*
- A Marian y Yaicel gracias por cargar conmigo, por compartir cada momento sin importa que fueran buenos o malos, siempre estaban ahí, gracias por ser mis hermanos acá en la universidad.*
- A Oda, gracias por poder contar con tu amistad, por tantas noches durmiendo juntos y sin poder tocarnos.*
- Ara por poder contar siempre con tu amistad, y por los celos que siempre tienes, esos son muestra de nuestra amistad*
- Jose gracias por soportarme y poder contar con tu amistad.*
- A Alice gracia por cada masaje que me distes en estos últimos días de estrés, y por dejarme ser tu amigo*
- A Gey, gracias por ser la amiga incondicional que siempre está dispuesta a ayudarme en cualquier circunstancia.*
- A Negrin, Roli, Sosa, Bauta, y a todo mi grupo en general con los que compartí estos 5 años.*
- A todas mis amistades, Addiel, Rodnier, Roisbel, Yunior y la familia de amigos.*
- A mi tutor Pedro que se ha convertido en un amigo, gracias a tu guía y confianza hoy puedo cumplir mis metas*
- A mis compañeros del proyecto Katia, Claudia, Julio, Dalinda, Rene, Alturo gracias a todos por ayudarme.*
- A todos los profes del Centro Cultural por soportarme cada día, en especial a Malcom y Yenirsy, por hacerme creer artista.*
- A mis compañeros de la FEU y de los Festivales, los de la nueva y vieja escuela; Ernesto gracias por creer y confiar en mí.*
- A mis amigos y conocidos que van dejando un poco de su experiencia en mi vida y por estar a mi lado en las buenas y en las malas.*

Dedicatoria

*A mis padres, porque creyeron en mí y porque me sacaron adelante,
dándome ejemplos dignos de superación y entrega, porque gracias a
ustedes, hoy puedo ver alcanzada mi meta, y porque el orgullo que sienten
por mí, fue lo que me hizo ir hasta el final.*

*Va por ustedes, por lo que valen, porque admiro su fortaleza y por lo que
han hecho de mí.*

*A mis hermanos, tíos, primos, abuelos y amigos,
Gracias por haber fomentado en mí el deseo de superación y el anhelo de
triunfo en la vida.*

*Mil palabras no bastarían para agradecerles su apoyo, su comprensión y
sus consejos en los momentos difíciles.*

*A todos, espero no defraudarlos y contar siempre con su valioso apoyo,
sincero e incondicional.*

RESUMEN

Numerosas son las aplicaciones que utilizan los datos de posición solar, principalmente las utilizadas en la Astronomía, Agronomía, Arquitectura y la Industria Energética. Existen modelos matemáticos astronómicos que permiten predecir la posición del sol y que se usan en varias aplicaciones informáticas de tipo escritorio, las cuales permiten obtener estos datos pero de forma individual. Sin embargo aún resultan insuficientes los resultados en la obtención íntegra de los parámetros solares. En la presente investigación se propone una herramienta informática que permite calcular parámetros solares, simular la trayectoria solar y de la sombra para un punto en el plano horizontal mediante Cartas Solares y Sombrigramas; brindando además información sobre los modelos matemáticos utilizados para la obtención de dichos parámetros. Para el desarrollo de la misma se hizo uso de la variación de la metodología “Proceso Unificado Ágil” (AUP, por sus siglas en inglés) en unión con el modelo CMMI-DEV v1.3, utilizando como marco de trabajo base Symfony en su versión 2.3.6. Al sistema SIPCSOL se le realizaron pruebas que demuestran su validez y que es posible su despliegue, cumpliendo con los requisitos definidos por el cliente.

Palabras claves: parámetros solares, sombrigramas, cartas solares, algoritmo.

ÍNDICE DE CONTENIDO

INTRODUCCIÓN.....	10
CAPÍTULO I.....	15
1.1. Análisis bibliométrico y documental.....	15
1.2. Marco teórico. Conceptos y definiciones.....	16
1.3 Materiales y métodos.....	17
1.4 Estado del Arte.....	19
1.5 Aplicación Web y de Escritorio.....	22
1.6 Metodología de desarrollo de <i>Software</i>	23
1.7 Herramientas a utilizar en la propuesta de solución.....	23
1.7.1 Lenguaje de modelado UML 2.0.....	24
1.7.2 Herramientas CASE (Computer Aided <i>Software</i> Environment).....	24
1.7.3 Modelado de prototipado web.....	25
1.7.4 Marco de Trabajo.....	26
1.7.5 Lenguaje de programación.....	28
1.7.6 Sistema Gestor de Base de Datos.....	31
1.7.7 Servidor web.....	31
1.6.8 Entorno de Desarrollo Integrado.....	32
1.8 Conclusiones Parciales.....	32
CAPÍTULO II DESCRIPCIÓN DE LA APLICACIÓN.....	33
2.1. Propuesta de solución.....	33
2.2 Modelo Conceptual.....	33
2.2.1 Diagrama del Modelo Conceptual.....	35
2.3 Captura de Requisitos.....	35
2.3.1 Requisitos funcionales.....	36
2.3.2 Requisitos no funcionales.....	37
2.3.3 Validación de requisitos.....	38
2.3.4 Historias de usuarios.....	38
2.4 Arquitectura.....	42
2.5 Patrones de diseño.....	45
2.5.1 Patrones de Diseño de Asignación de Responsabilidades o GRASP.....	45
2.3.1 Patrones de Comportamiento (Patrones GOF):.....	47
2.4 Diagrama de clases del Diseño con estereotipos web.....	47

2.5 Modelo de datos.....	48
Patrones de diseño de la base de datos.....	49
2.6 Conclusiones Parciales.....	50
CAPÍTULO III	52
3.1 Implementación.....	52
3.1.1 Diagrama de componentes	52
3.1.2 Diagrama de despliegue.....	54
3.1.3 Estándar de codificación empleado.....	55
3.2 Validación.....	56
3.2.1 Validación del diseño de la aplicación web	56
3.3 Pruebas	60
Pruebas Internas.....	60
3.3.1 Pruebas de Funcionalidad.....	60
3.3.2 Pruebas unitarias.....	60
3.4 Resultados de la prueba de rendimiento	62
3.5 Valoración del comportamiento de las variables de la investigación	63
3.5 Conclusiones Parciales.....	65
CONCLUSIONES GENERALES	66
RECOMENDACIONES.....	67
BIBLIOGRAFÍA REFERENCIADA.....	68
ANEXOS	74
Anexo 1 Síntesis del proceso de cálculo para llegar al Azimut de la Sombra y su Longitud.	74
Anexo 2 Simulación de la trayectoria de la sombra, mediante un Sombrigrama generado por el sistema SIPCSOL.....	75
Anexo 3 Simulación de la trayectoria solar, mediante una Carta Solar generado por el sistema SIPCSOL.....	76
Anexo 4: Tabla de requisitos no Funcionales	77
Anexo 5: Acta de Aceptación.....	79

INTRODUCCIÓN

Desde un principio el hombre primitivo distinguió los fenómenos causados por los dos principales movimientos de la tierra; el día y la noche asociados al movimiento de rotación, mientras que la traslación origina los cambios estacionales. Con este incipiente conocimiento de la trayectoria solar el hombre empezó a explicarse, de una forma u otra, el comportamiento del universo. Con el paso del tiempo el hombre adquirió conocimientos más profundos acerca de la trayectoria solar y con esto aprendió a medir el tiempo.

Claros testimonios de este conocimiento es la pirámide de KUKULKAN (Chichén Itzá 900 a 1250 D.C). Las primeras aplicaciones prácticas del conocimiento de la trayectoria solar se dieron en la medición del tiempo, a través de los relojes solares. Posteriormente, basados en un gran conocimiento gnomónico, se desarrollaron las cartas solares, de tal forma que para principios del siglo XVII se contaba ya con diagramas solares de alta precisión. Sin embargo la aplicación de la geometría solar en el diseño arquitectónico se remonta por lo menos al siglo V A.C. en Grecia (Fuentes Freixant, 2012).

La mayoría de las fuentes de energía usadas por el hombre se derivan indirectamente del Sol, ya que este puede, a través de toda su radiación lanzada ser aprovechada como energía para los humanos. Los combustibles fósiles preservan energía solar capturada hace millones de años mediante fotosíntesis. La energía hidroeléctrica usa la energía potencial del agua que se condensó en altura después de haberse evaporado por el calor del Sol.

Por otra parte, se encuentra la geometría solar, que constituye uno de los elementos más importantes dentro de los proceso de diseño arquitectónico ya que a través del conocimiento del comportamiento de la trayectoria de los rayos solares, tanto en su componente térmica como lumínica, se logra dar la óptima orientación al edificio, la mejor ubicación de los espacios interiores de acuerdo a su uso, y se pueden diseñar adecuadamente las aberturas y los dispositivos de control solar, logrando efectos directos de calentamiento, enfriamiento e iluminación, traducibles en términos de confort humano (Freixant, 2012).

Aplicando la geometría solar, se obtienen un conjunto de parámetros solares para predecir la posición del sol en un momento determinado. Estos parámetros solares lo constituyen: la declinación solar, el ángulo horario, la altura solar, el azimut solar, el azimut de la sombra, la altura de culminación y las horas de Sol.

El cálculo de estos parámetros permite aprovechar los beneficios de la trayectoria solar en diseños arquitectónicos, la agronomía, astronomía y su transformación en energía eléctrica; haciendo uso de modelos matemáticos a través de métodos numéricos que determinan las coordenadas solares de forma precisa para su posterior aplicación.

Llevar a cabo el cálculo de estos parámetros de forma manual, no garantiza precisión y rapidez pues es un proceso complejo, el cual involucra resolver varias ecuaciones trigonométricas, que sin la asistencia de *software*, la resolución de estos cálculos puede durar varios días. Además requiere de investigación previa por parte de las personas que desee obtener dichos parámetros

De igual forma el uso de herramientas existentes que realizan estos cálculos de forma individual es un proceso largo y costoso para los usuarios; pues los mismos al querer efectuar estos cálculos tienen que realizar una búsqueda en Internet, descargar estas herramientas y si las mismas no calculan en su totalidad los parámetros, tienen que realizar la misma operación varias veces.

Estas aplicaciones no brindan información sobre las ecuaciones básicas y algoritmos que se utilizan para realizar estos cálculos, lo que conlleva a que los usuarios tengan desconocimiento acerca de los modelos matemáticos utilizados para efectuar los cálculos en el sistema.

Otros de los principales problemas que tienen este *software* es que no logran gráficos que simulen la trayectoria solar y de la sombra, lo que trae consigo que los usuarios no puedan predecir con precisión la trayectoria del sol y el comportamiento de la sombra en un punto del plano horizontal para determinado rango de tiempo.

Lo antes expuesto constituye la problemática que origina esta investigación, por lo que se define el siguiente **problema a resolver** ¿Cómo realizar y publicar el cálculo de los parámetros solares en Cuba, de manera que se garantice precisión, rapidez y el acceso a estos por parte de los usuarios interesados?

Se determina como **Objeto de estudio**: el desarrollo de aplicaciones informáticas para obtener parámetros solares.

Se concreta el **Campo de acción**: sistemas de apoyo a la obtención de parámetros solares.

Se asume el **Objetivo General**: Desarrollar una aplicación web para realizar y publicar el cálculo de los parámetros solares en Cuba, de manera que se garantice precisión, rapidez y el acceso a estos por parte de los interesados.

Objetivos Específicos:

- ✓ Establecer los referentes teóricos de la investigación mediante un estudio sobre el modelo matemático y algoritmos que se utilizan para calcular las ecuaciones de los parámetros solares.
- ✓ Modelar y diseñar las funcionalidades del sistema para el cálculo y publicación de los parámetros solares en Cuba.
- ✓ Implementar el sistema para calcular y publicar los parámetros solares en Cuba.
- ✓ Validar la solución propuesta a partir de las pruebas internas, de liberación y de aceptación.

Hipótesis:

Si se desarrolla una aplicación web para realizar y publicar el cálculo de los parámetros solares en Cuba, entonces se garantizará precisión, rapidez y el acceso a estos por parte de los usuarios interesados.

Durante la investigación se han empleado un conjunto de **métodos científicos** como procedimientos lógicos, que se han seguido para la obtención y el procesamiento de la información.

Métodos teóricos

- ✓ *Histórico-Lógico*: permitió realizar un estudio sobre la estructura de los modelos matemáticos, herramientas que se llevan a cabo para el cálculo de los parámetros solares y las tendencias del uso actual de las mismas, con el fin de seleccionar los patrones más apropiados para implementar en la propuesta de solución y darle cumplimiento al objetivo general de la presente investigación.
- ✓ *Analítico-Sintético*: se empleó para realizar el estudio teórico de la investigación, permitiendo el análisis de documentos y la extracción de los elementos más importantes acerca del proceso de desarrollo de aplicaciones informáticas para obtener parámetros solares.
- ✓ *Inductivo-Deductivo*: fue utilizado para ordenar e integrar los algoritmos o modelos matemáticos para la obtención de los parámetros solares.
- ✓ *Modelación*: permitió la creación de una representación o modelo para obtener parámetros solares, haciendo posible el diseño de los prototipos funcionales y la representación de los requisitos del sistema informático.

Dentro de los **métodos de investigación empíricos** se emplearon:

- ✓ *Observación:* se empleó a través de la realización de análisis de algunas herramientas que se utilizan para el cálculo de los parámetros solares a nivel mundial y en Cuba.
- ✓ *Entrevista:* se aplicó a sujetos para comprobar la forma en que realizan los cálculos de estos parámetros, propiciando el levantamiento de requisitos del sistema y la obtención de información necesaria para el desarrollo de la investigación.
- ✓ *Medición:* permitió medir la calidad de la especificación de los requisitos, además de obtener una medida de la calidad del diseño para su validación y el comportamiento de las variables de la investigación.
- ✓ *Experimento:* se utilizó para recuperar y comparar los resultados que inducen a hacer cambios para obtener resultados aceptables por el cliente y captados en los requisitos.

El presente trabajo está estructurado de la siguiente manera:

Capítulo 1: Fundamentación teórica

En este capítulo se dan a conocer las cuestiones teóricas necesarias para la comprensión de la investigación, lo cual incluye el establecimiento del estado del arte a través del estudio de los modelos matemáticos y herramientas que se utilizan para el cálculo de los parámetros solares a nivel internacional y nacional. Se realiza el análisis y selección de la metodología de desarrollo de *software* a emplear en función de las necesidades y las características del grupo de desarrollo. Así como el estudio de las herramientas y tecnologías a emplear en el desarrollo de la solución.

Capítulo 2: Descripción de la aplicación

En este capítulo se realiza una breve descripción de los principales procesos del negocio y el análisis de la propuesta de solución. Luego se definen los requisitos funcionales, no funcionales y se describen las principales historias de usuario. Se expone todo el diseño de la aplicación con los artefactos definidos por la metodología para esta fase. Así como los resultados de la validación de los requisitos y el diseño, a través de la aplicación de técnicas y métricas respectivamente. Además se muestran varios diagramas para esclarecer las características de la propuesta de solución. Por último se muestra la arquitectura a la cual responde el sistema, la evidencia de los patrones de diseño utilizados, el diagrama de paquetes y el modelo de datos.

Capítulo 3: Implementación y validación de la propuesta

Se abordan aspectos significativos de la implementación del sistema, así como los productos de trabajo generados del proceso. Se realiza un resumen de los resultados obtenidos tras la aplicación de las pruebas realizadas al sistema. Por último se emite una valoración de las variables de la investigación antes y después de la puesta en práctica del sistema.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

El presente capítulo brinda una panorámica de los modelos matemáticos que permiten obtener los parámetros solares y los principales conceptos asociados al cálculo de los mismos, los cuales permitirán un mejor entendimiento de este trabajo. De igual forma se realiza el estado del arte sobre los *software* que se utilizan en el mundo para realizar estos cálculos. Por último, se describen un conjunto de herramientas y tecnologías a emplear en el desarrollo de la solución.

1.1. Análisis bibliométrico y documental

En esta sección se realiza un análisis bibliométrico con el objetivo de mostrar la novedad de la revisión bibliográfica realizada, basándose en las fechas de las publicaciones consultadas. Las bases de datos utilizadas son: *IEEE* y *Google Scholar*. Además, se relacionan los tipos de fuentes bibliográficas más citadas. Las fuentes bibliográficas son: artículos de revistas, libros, revistas artículos en congresos y sitios web. El análisis realizado se muestra en la siguiente tabla.

Tabla 1: Análisis bibliométrico y documental

Tipo de fuente bibliográfica	Cantidad consultada	Cantidad publicada en los últimos cinco años (2010-2015)
Artículos de revista	7	5
Libro	22	9
Artículos en congreso	4	3
Sitio web	19	10
Total:	52	27

De la tabla anterior se obtiene el resultado mostrado en la Figura 1, donde se muestra que el 52% de la literatura consultada pertenece a los últimos cinco años, por lo que se evidencia la actualidad de la bibliografía consultada



Figura1: Actualidad de la bibliografía consultada

1.2. Marco teórico. Conceptos y definiciones

La tierra da una vuelta sobre su eje cada 24 horas y completa una elipse alrededor del Sol cada 365.25 días, aproximadamente. La excentricidad de la órbita de la tierra es muy pequeña (0.01673). La distancia más corta entre la tierra y el Sol es el perihelio y la mayor el afelio. La distancia media Sol-tierra, r_0 , es una unidad astronómica (UA), y vale 1.496×10^8 Km (Santigosa, L.M.L y L.R, Junio 2010).

Los **equinoccios** son los momentos del año, en que el Sol forma un eje perpendicular con el ecuador y en el que la duración del día, es igual al de la noche; el equinoccio de otoño ocurre el 21 de septiembre y el de primavera el 21 de marzo, ambos establecen el comienzo de dichas estaciones (Iriondo, 2011).

Los **solsticios** son los momentos del año en que el Sol en su movimiento aparente, pasa por uno de los puntos de la eclíptica más alejado del ecuador y en el que se da la máxima diferencia de duración entre día y noche; suceden respectivamente el 21 de junio y el 21 de diciembre. Igualmente establecen el comienzo de las estaciones de verano e invierno (Iriondo, 2011).

La Geometría solar, trata en principio de los recorridos aparentes del Sol en el cielo y de su localización en distintas fechas y horas para, de ese modo, predecir su interacción con planos y volúmenes en el espacio, en función a su ubicación relativa (Amador, 2013).

Parámetros Solares: a los efectos de esta investigación los parámetros solares son todos aquellos que se obtienen con la aplicación de un algoritmo de geometría solar, entiéndase en este caso:

La declinación solar (δ), es el ángulo formado por una línea que una los centros de la tierra y el Sol y el plano ecuatorial (Santigosa, L.M.L y L.R, Junio 2010). Sus valores varían cada día entre $+ 23^\circ 26' 29,7''$ en el solsticio de verano hasta $- 23^\circ 26' 29,7''$ en el solsticio de invierno pasando por cero los días de los equinoccios (Lepetra, 2011).

El ángulo horario, es aquel que forman los rayos solares en un instante determinado con el plano meridiano y se mide desde el meridiano hacia el oeste, por ejemplo, a las 13:00 horas el ángulo horario es 15° al oeste (Marino, 2009).

Altura Solar (h_s), se define como el ángulo que forman los rayos solares en un instante determinado sobre un punto dado respecto al plano horizontal o como ¿Cuánto subo para mirar el Sol esta entre los valores 0 y 90 grados? (Santigosa, L.M.L y L.R, Junio 2010).

Azimut solar, se define como el ángulo medido sobre el plano del horizonte que forma la proyección horizontal de los rayos del Sol con la línea meridiana (dirección norte sur), este se

puede medir desde el sur o desde el norte, se puede definir también dando respuesta a la siguiente interrogante ¿Cuánto giro respecto al norte para observar el Sol? (Marino, 2009)

Altura de culminación o **culminación solar**, es la mayor altura solar que alcanza el Sol en el día, ocurre cuando el astro pasa por el meridiano del lugar, es decir, cuando su acimut medido desde el norte es 180° (Marino, 2009).

Azimut de la sombra y su longitud se determinan a partir de un grupo de parámetros solares (Anexo 1).

Sombrigrama, es un diagrama que se utiliza para saber con exactitud la sombra que reflejan las estructuras en función de la hora, día, mes, latitud y longitud en que se encuentre la edificación.

Carta Solar: diagrama que permite simular la trayectoria del Sol sobre un punto del plano horizontal.

El cálculo de estos parámetros solares, permite aprovechar los efectos beneficiosos de la energía proveniente del Sol, los cuales son utilizados en disimiles ramas de la ciencia como por ejemplo los diseños arquitectónicos de todo tipo relacionados con el asoleamiento y el control solar, en la astronomía, la agronomía y la transformación de los mismos en energía eléctrica (Marino, 2009).

Con la aparición de las nuevas tecnologías de la información y las comunicaciones varias han sido las formas de dar solución a los problemas referentes al cálculo de estos parámetros.

1.3 Materiales y métodos

Se realizó un estudio de los principales modelos matemáticos astronómicos empleados en los algoritmos de la geometría solar.

Las ecuaciones que se muestran a continuación representan una selección de las que permiten obtener los parámetros solares. Su selección se basa en variantes ofrecidas por varios autores y su análisis constituye la base para potenciar aspectos de rendimiento. Además de que los valores obtenidos en las ecuaciones seleccionadas son la base para generar los algoritmos (Verdecia, 2015).

Declinación solar (α)

La declinación puede ser calculada de varias formas. Según (Plasencia, 2007) los modelos matemáticos más empleados para dar cuenta de este fenómeno se muestran a continuación.

Según (Cooper, 1969) con la ecuación 1 expresada en grados:

$$\alpha = 23,45^\circ \times \sin(2\pi \times Dn + 284)/365 \quad (1)$$

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Tanto (Spencer, 1971) como (Michakskey, 1988) proponen utilizar las ecuaciones 2 y 3 para calcular la declinación con un resultado en radianes.

$$\alpha = 0,006918 - 0,399912 \times \cos(X) + 0,07257 \times \sin(X) - 0,006758 \times \cos(2X) + 0,000907 \times \sin(2X) - 0,00697 \times \cos(3X) + 0,001480 \times \sin(3X)$$

(2)

$$X = 2\pi \times \frac{Dn-1}{365} \quad (3)$$

Ecuación de Tiempo (EqT)

La diferencia de tiempo (en minutos) entre la hora civil y la hora solar varía durante un año precisamente por la excentricidad del movimiento de traslación y por la influencia de diversos fenómenos astronómicos. Para calcular estas diferencias de tiempo diarias EqT se utilizan a nivel mundial los modelos siguientes (Verdecia, 2015).

Las ecuaciones 4 y 5 son propuestas por (Spencer, 1971) y son ampliamente utilizadas a nivel mundial (Cooper, 1969).

$$EqT = 229,18 \times (0,000075 + 0,001869 \times \cos(\beta) - 0,032077 \times \sin(\beta) - 0,014615 \times \cos(2\beta) - 0,040849 \times \sin(2\beta))$$

(4)

Donde β se calcula a partir del Dn :

$$\beta = Dn \times (2\pi/365)$$

(5)

A opinión de (Plasencia 2007) un modelo matemático más preciso, aunque con saltos visibles para calcular la ecuación de tiempo EqT, es el propuesto por el laboratorio de Monitoreo de la Radiación Solar de la Universidad de Oregón en los Estados Unidos (SRML, 2007) en la ecuación 6:

$$EqT = \begin{cases} -14,2 \times \sin(\pi(d_n + 7)/111), & 1 \leq d_n \leq 106 \\ 4,0 \times \sin(\pi(d_n - 106)/59), & 107 \leq d_n \leq 166 \\ -6,5 \times \sin(\pi(d_n - 166)/80), & 167 \leq d_n \leq 246 \\ 16,4 \times \sin(\pi(d_n - 247)/113), & 247 \leq d_n \leq 365 \end{cases} \quad (6)$$

Las ecuaciones anteriores no dependen de la posición de un observador en la tierra y sus resultados son la base de las ecuaciones que aparecen en lo adelante. Solo tienen una variable de entrada Dn y esta normalmente toma los valores naturales entre 1 y 365, por lo tanto los algoritmos que utilicen estas ecuaciones tienen orden de complejidad prácticamente constante, O

(1) en la notación O grande. Los valores naturales de 1 a 365 de Dn son la generalidad en las aplicaciones que utilizan la Información para Predecir la Posición Solar (IPS). Las excepciones pueden encontrarse en aplicaciones de programas espaciales donde se consideran los valores entre 1 y 365 tal que $Dn = \{1 \alpha 365/\in\mathbb{R}\}$; en este caso Dn puede tomar n valores y el algoritmo pasa a tener un orden de complejidad $O(n)$. Al combinar los algoritmos para obtener la IPS se obtuvo en el peor de los casos una complejidad de $O(n^2)$.

1.4 Estado del Arte

Los modelos matemáticos mencionados anteriormente, constituyen la base para la obtención de algoritmos que se utilizan en disímiles herramientas informáticas para cálculos solares. Seguidamente se describen un conjunto de sistemas que son utilizados tanto a nivel internacional como nacional.

Software de cálculo solar en Cuba

Dig5.

Se trata de un software de escritorio para arquitectos estudiosos de la astronomía solar que es capaz de determinar las coordenadas solares para un punto cualquiera de La Tierra a cualquier hora del día y fecha del año. Dando además, los siguientes datos: Hora solar; hora oficial; declinación solar; ángulo horario; azimut de la sombra y su longitud a partir de la altura del punto proyectante. Esta aplicación permite obtener la simulación de diagramas de sombras o Sombrigramas pero no brinda información sobre las ecuaciones básicas o modelos matemáticos que se utilizan para llevar a cabo el cálculo de estos parámetros

Con estos datos, sobre todo los de altura y acimut solar, así como la longitud de la sombra y su correspondiente dirección respecto a la alineación norte sur, sirven para realizar diseños arquitectónicos de todo tipo relacionados con el asoleamiento y el control solar (Rosell, 2003).

Este software permite:

- ✓ La realización de proyectos complejos de Arquitectura con un alto contenido de parámetros solares en el diseño.
- ✓ El estudio solar de una región específica para futuros proyectos y conocimientos de la geografía del lugar con relación a los movimientos del Sol, su incidencia y efectos en las edificaciones.
- ✓ Propiciar el cálculo y conocimiento de parámetros de tiempo, como la salida, paso por el meridiano y puesta del Sol, así como la determinación de las horas de Sol para un día cualquiera del año. Lo que propicia diseños puntuales muy específicos que pueden llegar a ser espectaculares si se aprovechan los resultados del cálculo preciso de las coordenadas.

- ✓ Incluye una calculadora que determine por separado los parámetros solares sin que sea necesario imprimir los resultados con el único fin de conocerlos.

SolArq

Aplicación informática de escritorio que permite calcular los parámetros solares como la hora solar; hora oficial; declinación solar; ángulo horario; acimut de la sombra y su longitud a partir de la altura del punto proyectante. Esta aplicación no permite obtener la simulación de diagramas de sombras o Sombrigramas ni brinda información sobre las ecuaciones básicas o modelos matemáticos que se utilizan para llevar a cabo el cálculo de estos parámetros. Sin embargo esta aplicación está desarrollada en el lenguaje de programación C # y brinda los algoritmos para su posterior aplicación (Verdecia, 2014).

Sistemas internacionales para el cálculo de parámetros solares.

A continuación se describen un conjunto de sistemas para estudiantes y profesionales que realizan estudios sobre el cálculo de parámetros solares propuestas por el Centro de Estudios de la Energía Solar, primer Centro Internacional para la formación tecnológica de especialistas en energía solar.

Geoclock

Es un sistema informático que muestra las posiciones y coordenadas del Sol; además de la hora actual y las horas alrededor de todo el mundo. También permite ver la posición actual del Sol, las partes de la tierra en la luz del Sol; así como la salida del Sol local, puesta del Sol, el azimut del Sol y elevación. Este *software* se actualiza automáticamente cada pocos segundos.

Geoclock es privativa y muy recomendable ya que permite aproximarse a conocimientos de la geografía terrestre, posibilita el cálculo rápido de las distancias entre dos puntos del globo, la latitud y longitud, la posición de la tierra con respecto al sol, la situación horaria, etc. Cuenta con cuatro tipos de visores: un mapa del mundo, un mapa de los Estados Unidos (con gran parte de México y Canadá), una vista polar y una vista global desde el Espacio Exterior. Aspectos Técnicos: Windows 95 / 98 / NT / 2000, Tamaño comprimido: 610 k. Productor: GeoClock Sunlight Clock (Ahlgren, 2004)

Sunpath

Es privativa y permite calcular las posiciones solares para una secuencia de fechas y horas, salida del Sol y puesta del Sol, los solsticios y equinoccios, la hora solar a las conversiones de tiempo estándar, el horario de verano, y encuentra las fechas y horas en que el Sol está a una distancia

especificada por el usuario a partir de una determinada dirección en el cielo. También se puede proporcionar una visualización continua de la hora y la posición solar.

Permite imprimir los resultados directamente en una impresora, o guardar en un archivo o en archivos con formato para importar a los programas de trazado gráfico (Solar, Centro de Energía, 2012).

Shadows

En su versión básica es gratuito, permite el cálculo de sombras y diagramas para reloj solar, además posee un conjunto muy completo de funcionalidades y dispone de una completa documentación. Shadows posee una formidable utilidad pedagógica con la simulación de la sombra del estilo. El programa también calcula la sombra producida por el alero de un tejado para comprobar si el cuadrante está iluminado durante todo el año.

La simulación de la sombra puede utilizarse con provecho en las aulas para estudiar las variaciones de altura y azimut del Sol a lo largo del año, e introducir conceptos astronómicos como la inclinación del eje de rotación de la Tierra y las estaciones (Centro de Energía Solar, 2012).

RiseSet

RiseSet, es una sencilla aplicación con la que se podrá, una vez introducida la latitud, longitud y una fecha deseada, calcular y prever la hora exacta de la salida y puesta del Sol y la Luna con los crepúsculos, y hasta las fases lunares y eclipses.

Esta herramienta luego de haber proporcionado los datos de ubicación, el programa calculará la hora del encuentro tan esperado, con su altitud y los ángulos de azimut del Sol y la Luna, con la posibilidad de almacenar tres localizaciones para su acceso rápido (Centro de Energía Solar, 2012).

Una vez realizada la descripción de algunas aplicaciones informática para realizar estos cálculos, estas son sometidas a un análisis comparativo (Tabla 2) que responda al problema de la presente investigación, teniendo en cuenta el tipo de herramienta, el tiempo de respuesta, la cantidad máxima de parámetros que calcula, si brindan información sobre los modelos matemáticos utilizado para la obtención de estos parámetros, si permiten realizar la simulación de la trayectoria solar y de la sombra mediante las Cartas Solares y Sombrigramas.

Tabla 2: Comparación entre Aplicaciones para el cálculo de parámetros solares.

Sistemas	Cant.P	T-R	T-App	Infr.	M-S.	A-A.
----------	--------	-----	-------	-------	------	------

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Geoclock	2	30 segundos	Escritorio	No	No	No
Sunpath	3	11 milisegundo	Escritorio	No	No	No
Shadows	3	10 milisegundo	Escritorio	No	No	No
RiseSet		12 milisegundos	Escritorio	No	No	No
Dig5.	14		Escritorio	No	Si	No
SolArq	14	12 milisegundos	Escritorio	No	No	Si

Como resultado del análisis comparativo, el autor de la presente investigación arrojó a la siguiente conclusión, no existe una solución informática que realiza de forma íntegra los cálculos de los parámetros solares, la modelación mediante Sombrigramas y Cartas Solares, que además brinde información sobre las ecuaciones básicas o modelos matemáticos utilizados para calcular estos parámetros.

1.5 Aplicación Web y de Escritorio

Una aplicación web es un sistema informático que los usuarios utilizan accediendo a un servidor web a través de Internet o de una intranet. Las aplicaciones web son populares debido a la practicidad del navegador web como cliente ligero. La facilidad de actualizar y mantener aplicaciones web sin distribuir e instalar *software* en miles de potenciales clientes es otra razón de su popularidad (Artículo Tecnológico, 2010).

Se denominan aplicaciones de escritorio a toda aplicación que ha sido desarrollada para ser ejecutada en una plataforma específica, ya sea Windows, Linux, Mac u otra. El desarrollo sobre una plataforma normalmente implica que la aplicación "no" pueda ser ejecutada en otra. En la siguiente tabla se muestra

Tabla 3: Comparación entre Aplicaciones Web y Escritorio (Open Systems Developmen OSD, 2013)

Características	Web	Desktop
Personalización, actualización y soporte.	Es suficiente con realizar los cambios en el servidor WEB.	Los cambios hay que realizarlos en cada PC donde se tenga la aplicación.
Accesibilidad y cobertura.	Cualquier lugar con acceso a la red.	Solo en la PC donde se haya instalado previamente la aplicación.
Capacidad de usuarios concurrentes.	Alta, debido a la arquitectura basada en servicios web.	Baja, ya que la forma de diseño es centrada en un único usuario local.
Portabilidad.	El sistema puede ser usado a través de un navegador, sin necesitar grandes prestaciones de recursos en la PC.	Puede presentar incompatibilidades con el sistema operativo, así como dependencia de los recursos que posea la PC.
Infraestructura y movilidad.	Solo se tiene que conectar a la red.	Está restringido a la ubicación de la PC local.
Seguridad eléctrica y	Es responsabilidad del	Es responsabilidad de cada

lógica.	proveedor de servicio.	usuario que usa el sistema localmente.
----------------	------------------------	--

A partir de la comparación establecida anteriormente y en correspondencia con las necesidades existentes, entre las que resalta: la necesidad que la solución sea accesible desde distintas partes de nuestro país; el autor de este trabajo determina desarrollar la propuesta con enfoque web, lo cual brinda múltiples beneficios en cuanto a portabilidad, actualización y soporte, e infraestructura.

Para proseguir, se hace necesaria la selección de la metodología de desarrollo de *software* y las herramientas y tecnologías a emplear durante el desarrollo de la propuesta.

1.6 Metodología de desarrollo de Software

Se puede definir como metodología de desarrollo de *software*: un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda en la construcción de un *software* (Pressman, 2012).

En la actualidad existen diversas metodologías de desarrollo de *software* para guiar a los equipos de trabajo en la creación de un nuevo *software*. Atendiendo a su complejidad son separadas en dos grupos: metodologías tradicionales o formales y las ágiles o ligeras.

La Universidad de las Ciencias Informáticas adopta una nueva metodología por la cual se deben regir todos los proyectos productivos en la misma. Se basa en una variación de la metodología “Proceso Unificado Ágil” (AUP, por sus siglas en inglés) en unión con el modelo CMMI-DEV v1.3. Esta metodología establece distintas fases y disciplinas por las que se debe transitar durante el desarrollo. Las fases definidas son: Inicio, Ejecución (Elaboración, Construcción, Transición) y Cierre. Las disciplinas son: Modelo (Modelado del negocio, Requisitos, Análisis y Diseño), Implementación, Pruebas (Internas, Liberación y Aceptación) y Despliegue (opcional) (Sánchez, 2014).

La solución que se desea implementar se enmarca en la fase de Ejecución, en la cual se ejecutan las actividades requeridas para desarrollar el *software*. Durante la Ejecución se realizan los procesos de negocio, se refinan los requisitos, se elabora la arquitectura y el diseño, se implementa y se libera el producto (Sánchez, 2014).

1.7 Herramientas a utilizar en la propuesta de solución

Para definir las herramientas y tecnologías a utilizar durante el desarrollo de la propuesta, se realizó un estudio sobre las líneas de desarrollo utilizadas en el centro CEIGE (Centro de Informatización de Entidades) para aplicaciones web.

El centro contiene una línea de desarrollo orientada a la realización de aplicaciones web mediante la utilización del marco de trabajo Symfony2, con lenguaje PHP y otra línea orientada a

aplicaciones de escritorio que utiliza la tecnología Java; aunque es necesario aclarar que con la tecnología Java también se pueden realizar aplicaciones web mediante uso de marcos de trabajo como Spring¹.

La primera línea descrita responde a las necesidades del cliente, por lo que el autor hará uso de la misma para dar solución al problema de la presente investigación.

A continuación se realiza un análisis con respecto a las características y beneficios que brindarán las herramientas y tecnologías definidas en correspondencia con la línea de desarrollo seleccionada.

1.7.1 Lenguaje de modelado UML 2.0

El Lenguaje Unificado de Modelado (UML por sus siglas en inglés *Unified Modeling Language*) es un lenguaje estándar de modelado, para la visualización, especificación, construcción y documentación de los artefactos de sistemas en los que el *software* juega un papel importante. Permite a los desarrolladores visualizar los resultados de su trabajo en esquemas o diagramas estandarizados (Rumbaugh, 2007).

Este lenguaje además, proporciona un conjunto de elementos de modelado, anotaciones, relaciones y normas que pueden aplicarse a una actividad de desarrollo de *software*. Sin embargo, UML también puede ser empleado para modelar otros dominios, tales como el modelado de sistemas y el modelado de negocio (Leffingwell, 2003).

Para facilitar el uso de este lenguaje de modelado se utiliza la herramienta Visual Paradigm la cual permite aprovechar las facilidades de uso y la amplia gama de utilidades que brinda, a continuación una descripción de esta herramienta.

1.7.2 Herramientas CASE (Computer Aided Software Environment)

Las herramientas CASE comprenden un conjunto de programas de diferentes tipos empleados para ayudar a las actividades del proceso del *software* como el análisis de requisitos, el modelado de sistemas, la depuración y las pruebas (Sommerville, 2005).

Permiten incrementar la productividad, comunicarse de manera más eficiente con los usuarios, integrar las actividades y proporcionar continuidad de una fase a la siguiente durante todo el ciclo de vida del desarrollo de sistemas e integrar el trabajo que desempeñan en el sistema, reduciendo de esta forma el costo en términos de tiempo, recursos y dinero del proceso de desarrollo del *software* (Kendall, 2005).

Visual Paradigm 8.0 Enterprise Edition.

¹Spring: Marco de trabajo para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java.

Es una de las herramientas UML CASE del mercado, considerada como muy completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones.

Fue creada para el ciclo de vida completo del desarrollo de *software* que lo automatiza y acelera, lo que permite la captura de requisitos, análisis, diseño e implementación. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de las clases.

Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación. Está diseñada para usuarios interesados en sistemas de *software* de gran escala con el uso del acercamiento orientado a objeto, además apoya los estándares más recientes de las notaciones de Java y de UML. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros (Leffingwell, 2003).

1.7.3 Modelado de prototipado web

Para el prototipado web se decide emplear Axure RP Pro 6.5 pues permite crear prototipos navegables, con mejor calidad visual. Además la herramienta brinda un conjunto de bibliotecas muy atractivas.

Además, generar prototipos de diseño HTML, lo que permite la interacción de cualquier miembro del equipo de desarrollo o las partes interesadas con estos sin necesidad de instalar Axure. De esta forma se puede comprobar si el sistema de navegación de la aplicación web es el deseado y además el cliente podrá tener una idea clara y precisa de cómo será la aplicación final (Oficial, 2010).

Luego de la presentación de esta herramienta CASE para la representación visual de los principales componentes del desarrollo del *software* y para el prototipado web el Axure RP 6.5; se da paso a las tecnologías y herramientas propuestas por la línea de desarrollo, lo que establece el uso del marco de trabajo Symfony2 con el lenguaje de programación PHP, como servidor de aplicaciones Apache 2.2, Gestor de Base de Datos PostgreSQL 9.3 y como IDE de desarrollo el NetBeans en su versión 7.3.

1.7.4 Marco de Trabajo

Según Jorge Naula, un *framework*², es: “una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación”. Es decir; un *marco de trabajo* web se puede definir como un conjunto de componentes (por ejemplo: clases, descriptores y archivos de configuración en XML³) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web y es precisamente el que se utiliza en estos estudios (Euphoriait, 2009).

Entre los objetivos principales de un *marco de trabajo* se encuentran: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo, como el uso de patrones.

Symfony2.3.6

Este marco de trabajo fue diseñado con el objetivo de optimizar la creación de las aplicaciones web con el uso de sus características. Puede ser utilizado en plataformas Unix, Linux y Windows; y está desarrollado en lenguaje PHP. Posee además una librería de clases que permite reducir el tiempo de desarrollo; pero requiere de instalación, configuración y utilización de líneas de comando.

En este marco de trabajo están embebidos diferentes patrones de diseño los cuales se agrupan en 3 categorías: creacionales, estructurales y de comportamiento (Hamon, 2014). A continuación se describen los mismos.

Creacionales: tratan con las formas de crear instancias de objetos. El objetivo de estos patrones es de abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados.

- ✓ **Factory Method** (Método de fabricación): Centraliza en una clase constructora la creación de objetos de un subtipo de un tipo determinado, ocultando al usuario la casuística para elegir el subtipo que crear.

Estructurales: Los patrones estructurales describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionales pueden ser incluso objetos simples u objetos compuestos.

- ✓ **Adapter** (Adaptador): Adapta una interfaz para que pueda ser utilizada por una clase que de otro modo no podría utilizarla.

²**Framework:** La traducción de *framework*(o en plural *frameworks*) en idioma español es marco de trabajo.

³**XML:** Siglas en inglés de *eXtensible Markup Language* ('lenguaje de marcas extensible').

- ✓ **Composite** (Objeto compuesto): Permite tratar objetos compuestos como si de uno simple se tratase.
- ✓ **Decorator** (Envoltorio): Añade funcionalidad a una clase dinámicamente.

Comportamiento: Los patrones de comportamiento ayudan a definir la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos.

- ✓ **Iterator** (Iterador): Permite realizar recorridos sobre objetos compuestos independientemente de la implementación de estos.
- ✓ **Mediator** (Mediador): Define un objeto que coordine la comunicación entre objetos de distintas clases, pero que funcionan como un conjunto.
- ✓ **Strategy** (Estrategia): Permite disponer de varios métodos para resolver un problema y elegir cuál utilizar en tiempo de ejecución.
- ✓ **Template Method** (Método plantilla): Define en una operación el esqueleto de un algoritmo, delegando en las subclasses algunos de sus pasos, esto permite que las subclasses redefinan ciertos pasos de un algoritmo sin cambiar su estructura.

Este marco de trabajo contiene abundante documentación y está diseñado para trabajar de forma modular en bundles⁴ y ha sido ideado para aprovechar todas las nuevas características de PHP 5.3. Entre sus ventajas se puede mencionar que incorpora el patrón Modelo-Vista-Controlador (MVC)⁵, soporta AJAX⁶, plantillas y conexión y trabajo con la bases de datos con la integración del ORM Doctrine (Alfonso, 2012).

ORM Doctrine 2

Doctrine 2 es el mapeador de objetos relacionales (ORM) para PHP 5.3.0+ que provee una persistencia transparente para los objetos PHP. Este usa el patrón de mapeo de datos, lo que proporciona una completa separación entre la lógica del dominio/negocio de la persistencia en un sistema de gestión de bases de datos relacionales.

El beneficio de Doctrine para los programadores es la habilidad de concentrarse únicamente en la lógica de negocio orientada a objetos y preocuparse de la persistencia solo como una tarea secundaria. Lo cual no significa que la persistencia no sea importante para Doctrine 2, sin embargo se sostiene el criterio de que existen considerables beneficios para la programación orientada a objetos si la persistencia y las entidades son perfectamente separadas (Mapper, 2012).

⁴**Bundles:** Conjunto de archivos que implementan una única funcionalidad.

⁵**MVC:** Siglas de Modelo Vista Controlador. Es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación.

⁶**AJAX:** Acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML).

Twig 2.1

Twig es un motor de plantillas, integrado con Symfony2 para crear las plantillas de la aplicación, las cuales tienen una sintaxis concisa, fáciles de leer y de escribir. La característica más importante de Twig es que está implementado con herencia entre plantillas, lo que permite crear un “esqueleto” base que contiene todos los rasgos comunes de las interfaces ahorrando así código y tiempo en el trabajo. Twig se caracteriza por ser rápido, seguro y flexible:

Rápido: Twig compila las plantillas hasta código PHP regular optimizado. El costo general en comparación con código PHP regular se ha reducido al mínimo.

Seguro: Twig tiene un modo de recinto de seguridad para evaluar el código de plantilla que no es confiable. Esto permite utilizar Twig como un lenguaje de plantillas para aplicaciones donde los usuarios pueden modificar el diseño de la plantilla.

Flexible: Twig es alimentado por flexibles analizadores léxico y sintáctico. Esto permite al desarrollador definir sus propias etiquetas y filtros personalizados (Twig, 2012).

Twitter Bootstrap 3.1

Twitter Bootstrap es una colección de herramientas de *software* libre para la creación de sitios y aplicaciones web el cual es compatible con gran parte de los navegadores web. Bootstrap 3.1 es un *framework* diseñado para simplificar el proceso de creación de diseños web. Para ello ofrece una serie de plantillas CSS y de ficheros JavaScript, los cuales permiten obtener:

- ✓ Interfaces que funcionen de forma perfecta en los navegadores actuales y correctamente en los no tan actuales.
- ✓ Un diseño que pueda ser visualizado de forma correcta en distintos dispositivos y a distintas escalas y resoluciones.
- ✓ Una mejor integración con las bibliotecas que se suelen usar habitualmente, como por ejemplo jQuery.
- ✓ Un diseño sólido basado en herramientas actuales y potentes (Bootstrap, Twitter).

1.7.5 Lenguaje de programación

Un lenguaje de programación es un lenguaje artificial que se utiliza para expresar programas de ordenador. Está formado por un conjunto de símbolos, palabras claves utilizables y por reglas gramaticales para construir sentencias sintáctica y semánticamente correctas (Sala, 2003).

A continuación se hace un análisis de las características de los lenguajes de programación a utilizar en la solución.

Lenguaje de programación del lado del servidor

PHP 5.3

PHP es un lenguaje interpretado de alto nivel, multiplataforma, embebido en páginas HTML (*Hyper Text Markup Language*, Lenguaje para el Formato de Documentos de Hipertexto). La característica más potente y destacable de este lenguaje es su soporte para una gran cantidad de bases de datos, además de otros servicios que usen protocolos como IMAP⁷, SNMP⁸, NNTP⁹, POP3¹⁰, HTTP y derivados.

La mayoría de su sintaxis es similar a C, Java y Perl; y la meta de este lenguaje es permitir escribir a los desarrolladores web, páginas dinámicas de una manera rápida y fácil (Sæther, 2009).

Además posee gran capacidad de procesamiento de texto, donde se incluyen las Expresiones Regulares Compatibles de Perl, muchas extensiones y herramientas para el acceso y análisis de documentos XML (Lenguaje de Marcado Extensible, por sus siglas en inglés). También es potente, de alto rendimiento, de fácil aprendizaje y de escaso consumo de recursos (Gutierrez, 2009).

Lenguajes de programación del lado del cliente

JavaScript 1.8.5

JavaScript es uno de los lenguajes de script u orientado a documento, creado por la empresa Netscape, para añadir interactividad a las páginas web. Es compacto y basado en objetos, diseñado para el desarrollo de aplicaciones cliente-servidor (Maza, 2012).

Es un lenguaje interpretado e independiente de plataforma que permite incluir macros en páginas web. Estas macros se ejecutan del lado del cliente y no en el servidor. La característica de JavaScript que más simplifica la programación es que, aunque el lenguaje soporta varios tipos de datos, no es necesario declarar el tipo de las variables, argumentos de funciones ni valores de retorno de las funciones. El tipo de las variables cambia implícitamente cuando es necesario, lo que ayuda a programar con rapidez macros sencillas.

La principal ventaja de JavaScript es que eso sucede sin ningún tipo de transmisión de datos a través de Internet, de tal forma que cuando un usuario escribe algo en un formulario, no es necesario que sea enviado al servidor, verificado y devuelto (Boticario, 2001).

⁷ **IMAP**: Protocolo de Acceso a Mensajes Electrónicos, por sus siglas en inglés.

⁸ **SNMP**: Protocolo Simple de Administración de Red, por sus siglas en inglés.

⁹ **NNTP**: Protocolo para la Transmisión de Noticias en Red, por sus siglas en inglés.

¹⁰ **POP3**: Protocolo de Oficina de Correo u Oficina Postal.

Flotcharts 1.8

Flot es una biblioteca java script, que permite realizar varios tipos de gráficos; es fácil de usar, con sólo unas pocas líneas de código, se puede hacer gráficos de distintos tipos, como gráfico de líneas, gráfico circular, gráfico de barras, gráfico de la zona, carta apilada. Flot también es compatible con la carta de actualización en tiempo real y actualización Ajax, tiene amplia documentación donde se pueden encontrar ejemplos, el uso y métodos.

El rendimiento es bueno y es compatible con todo tipo de tipos de gráficos. También hay plugins para Flot utilizar (Schnur, 2014).

HTML 5

HTML, siglas de Hyper Text Markup Language (lenguaje de marcado de hipertexto), en su versión 5, es usado para describir la estructura y el contenido en forma de texto. Este puede describir hasta un cierto punto, la apariencia de un documento. Además se ha convertido en el formato más fácil para la creación de páginas web debido a su sencillez y no hay que compilar el código para ver si funciona. Se puede ver en forma inmediata el resultado del trabajo y también es usado para complementar el texto con objetos tales como imágenes (Eguiluz, 2010).

Este lenguaje se ha elegido por sus innumerables características, de las cuales se destacan que es un lenguaje estático para el desarrollo de sitios web que permite describir hipertexto presentando el texto de forma estructurada y agradable. Se utiliza para definir texto, tablas, y otros elementos que forman parte del diseño de la página web.

CSS 3

Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML por sus siglas en inglés de eXtensible Markup Language (lenguaje de marcas extensible) y por extensión en XHTML (por sus siglas del inglés). Permite separar la estructura de un documento de su presentación. La información de estilo puede ser adjuntada tanto como un documento separado o en el mismo documento HTML (Eguiluz, 2010).

Las ventajas de utilizar CSS es que tienen un control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo. Por otro lado los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad ya que una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Facilitando que el documento HTML en sí mismo sea más claro de entender y se consiga reducir considerablemente su tamaño (Eguiluz, 2010).

1.7.6 Sistema Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que administran y gestionan la información contenida en una base de datos. Ayuda a realizar la definición de los datos, el mantenimiento de la integridad de estos dentro de la base de datos, el control de su seguridad y privacidad; así como la manipulación de los mismos (Álvarez, 2007).

PostgreSQL 9.3

Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo la licencia BSD y con su código fuente disponible libremente. Utiliza un modelo cliente/servidor y usa multiprocesos para garantizar la estabilidad del sistema ya que un fallo en uno de los procesos no afecta el resto y el sistema continúa con su funcionamiento. Funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema (Caswell, 2007).

PostGIS 9.1

Es una extensión que convierte el sistema de base de datos PostgreSQL en una base de datos espacial. La combinación de ambos es una solución perfecta para el almacenamiento, gestión y mantenimiento de datos espaciales (Ramsey, 2014).

- ✓ PostGIS es *software* libre, tiene licencia GNU General Public License (GPL).
- ✓ Es compatible con los estándares de OGC.
- ✓ Soporta tipos de datos espaciales, índices espaciales y tiene cientos de funciones espaciales (+ 890 en la versión 2.0). Permite importar y exportar datos a través de varias herramientas de conversión (shp2pgsql, pgsq2shp, ogr2ogr, dxf2postgis).

1.7.7 Servidor web

Un servidor web es un programa que sirve datos en forma de páginas web, hipertextos o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. La comunicación de estos datos entre cliente y servidor se hace por medio del protocolo HTTP. Con esto, un servidor Web se mantiene a la espera de peticiones HTTP, que son ejecutadas por un cliente HTTP; lo que se conoce como un navegador web (Enciclopedia Cubana en la Red.).

Apache 2.2

El Servidor Apache 2.2 es un servidor web de código abierto para plataformas Unix, GNU/Linux, Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1. Entre sus principales ventajas se encuentran que es modular, código abierto, multiplataforma, extensible y fácil de proveer soporte (del Castillo, 2000).

1.6.8 Entorno de Desarrollo Integrado

Un Entorno de Desarrollo Integrado (*Integrated Development Environment*) es un programa informático compuesto por un conjunto de herramientas de programación que les permite a los programadores escribir, compilar, depurar y ejecutar programas (Editorbfb, 2011).

NetBeans 7.3

Es un producto libre y gratuito sin restricciones de uso, teniendo el código fuente disponible para su reutilización bajo las licencias CDDL (Licencia Común de Desarrollo y Distribución) y la GPL (General Public License). Facilita la programación proporcionando el auto-completamiento y la inserción automática de código, una descripción sencilla sobre la estructura del programa, opciones de refactorización, depuración, entre otros.

1.8 Conclusiones Parciales

El análisis de los principales conceptos asociados al dominio del problema facilitó una mejor comprensión del negocio. Además el establecimiento del estado del arte permitió obtener las tendencias actuales de las aplicaciones y sistemas que calculan los parámetros solares tanto nacionales como internacionales. Lo que ratificó la necesidad de crear una solución para el problema planteado e incluir y adaptar algunas de las funcionalidades estudiadas de dichos sistemas.

Luego del estudio comparativo entre aplicaciones web y de escritorio ante el problema planteado, se decidió desarrollar la solución de tipo web.

La selección de la metodología AUP, permitirá guiar el proceso de desarrollo de *software*. Además se realizó un análisis detallado de las herramientas y tecnologías propuestas para lograr una mejor comprensión de cada una de ellas.

CAPÍTULO II DESCRIPCIÓN DE LA APLICACIÓN.

En el presente capítulo se hace una descripción de la propuesta de solución a través de los requisitos funcionales y no funcionales, así como las particularidades del diseño. Se muestran aspectos esenciales de la arquitectura del sistema y los patrones de diseño empleados. También se hace alusión a los productos de trabajo inherentes a las fases Inicio y Ejecución (Modelado del negocio, Requisitos, Análisis y diseño).

2.1. Propuesta de solución

Una vez concluido el estudio de algunas aplicaciones que realizan cálculos de parámetros solares, se decide desarrollar una aplicación web que permita obtener estos parámetros en su totalidad, utilizando los algoritmos que propone (Verdecia, 2014) en el sistema SolArq.

La solución que se desea, permitirá además brindar información sobre los modelos matemáticos utilizados para efectuar estos cálculos en el sistema y obtener Sombrigramas y Cartas Solares que simulen la trayectoria solar y el recorrido de la sombra para un punto del plano horizontal, teniendo como guía el sistema Dig5.

2.2 Modelo Conceptual

Luego de un análisis del problema en cuestión se llega a la conclusión de que en el negocio de la presente investigación, no se definen concretamente los procesos del mismo, pues se manejan conceptos que quedarán recogidos en el modelo conceptual. El dinamismo del sistema se concibe a partir del uso de prototipos que marcarán la secuencia de actividades a seguir.

Un modelo conceptual tiene como objetivo identificar y explicar los conceptos significativos en un dominio del problema, identificando los atributos y las asociaciones existentes entre ellos. Puede ser visto, también como una representación de las cosas, entidades, ideas, conceptos u objetos del “mundo real” o dominio de interés. Es importante diferenciar que dichas entidades u objetos no son componentes de *software*. (Moreno, 2009)

CAPÍTULO 2: DESCRIPCIÓN DE LA APLICACIÓN.

Este modelo contribuye posteriormente a identificar algunas clases que se utilizarán en el sistema, lo que ayuda a los clientes y desarrolladores a utilizar un vocabulario común que permita entender el contexto en que se emplaza el sistema.

Para esto es necesario tener un conocimiento sólido del funcionamiento del objeto de estudio para hacer una correcta captura de los requisitos y construir un sistema correcto. Ahora bien, para un mayor entendimiento del modelo conceptual se dejan claros todos los conceptos que intervienen en el mismo:

Usuario: Persona que interactúa con el sistema.

Roles: Contiene información sobre los diferentes roles que pueden tener los usuarios una vez ingresados en el sistema.

Artículos: Muestra información sobre las ecuaciones básicas que se tienen en cuenta para llevar a cabo el cálculo de los parámetros solares, así como los algoritmos o modelos matemáticos que se utilizaron en el sistema para efectuar los mismos.

Sombrigrama: diagrama de sombra, que permite simular la trayectoria de la sombra, en un periodo de tiempo, de un punto de referencia por encima del plano horizontal (Anexo 2).

Carta Solar: diagrama que permite simular la trayectoria del Sol sobre un punto del plano horizontal (Anexo 3).

Estudio: tiene información sobre los datos de entrada para efectuar los cálculos de los parámetros solares y a este se le asocia los Sombrigramas y las Cartas Solares.

Parámetro Solar: son todos aquellos que se obtienen con la aplicación de un algoritmo de geometría solar.

CAPÍTULO 2: DESCRIPCIÓN DE LA APLICACIÓN.

2.2.1 Diagrama del Modelo Conceptual.

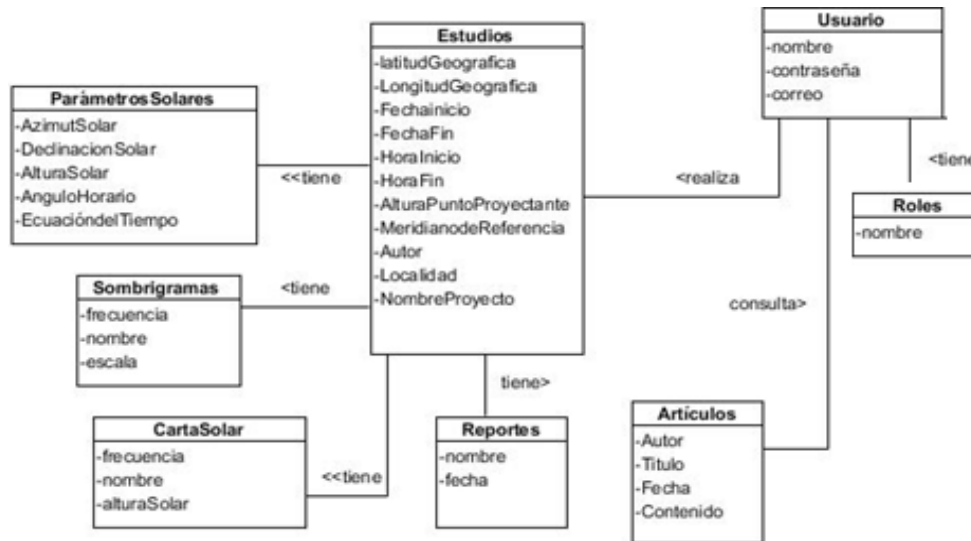


Figura 2: Diagrama del Modelo Conceptual.

2.3 Captura de Requisitos.

La ingeniería de requisitos es el conjunto de actividades implicadas en descubrir, documentar y mantener un conjunto de requisitos del software a desarrollar. La captura de los mismos es un proceso en el cual los datos son extraídos a partir de la aplicación de técnicas de captura de información (Martínez, 2012).

En la fase de Ejecución, de la metodología seleccionada para el desarrollo de la solución se define la disciplina Requisitos, esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto.

La entrevista, tormenta de ideas y la revisión de sistemas fueron las técnicas empleadas para la captura de requisitos.

La entrevista permitió tomar conocimiento del problema y comprender los objetivos de la solución. Mientras que el uso de la técnica tormenta de idea, permitió acumular nuevas ideas y/o información, proporcionadas por varios expertos.

La revisión de sistemas, fue aplicada con el objetivo de realizar un estudio a las herramientas Dig5 y SolArq, la cual permitió definir nuevas funcionalidades y adquirir los algoritmos que propone SolArq para realizar estos cálculos.

CAPÍTULO 2: DESCRIPCIÓN DE LA APLICACIÓN.

2.3.1 Requisitos funcionales.

Los requisitos funcionales son aquellos que indican lo que debe hacer el producto, son las capacidades con las que debe cumplir el mismo. Como resultado de la aplicación de las técnicas de captura de requisitos empleadas se obtuvieron 24 requisitos funcionales.

Estos requisitos fueron agrupados por objetivos según el estándar para la Agrupación de Requisitos Funcionales (ARF) de la IEEE 830:

- **ARF1-Administración:** esta agrupación le permite al administrador general gestionar los usuarios de la aplicación, los cuales a partir del rol que desempeñen se les otorgarán privilegios para trabajar sobre los demás módulos del sistema. Además permite la administración de los artículos que se publican en la portada de la aplicación.
- **ARF2-Portada:** esta agrupación le permite al usuario consultar artículos donde se refleja información sobre cuáles son las ecuaciones básicas y los algoritmos que se utilizan en el sistema para obtener los parámetros solares.
- **ARF3-Estudios:** el usuario puede en esta agrupación mostrar y gestionar los 2 tipos de estudios (simple o avanzado) según el rol que tenga asignado.

A continuación se muestra una lista con los requisitos funcionales del sistema.

Tabla 4: Lista de Requisitos funcionales.

Número	Requisito	Descripción
RF#1	Crear estudio avanzado.	Permite crear un nuevo estudio.
RF#2	Mostrar listado de estudios avanzados.	Muestra el listado de estudios avanzados.
RF#3	Actualizar estudios avanzados.	Permite modificar los datos de los estudios avanzados.
RF#4	Eliminar estudio avanzado.	Permite eliminar un estudio avanzado del listado.
RF#5	Realizar estudio simple.	Permite realizar un Estudio Simple.
RF#6	Exportar estudio.	Permite exportar los resultados de los estudios avanzados realizados.
RF#7	Obtener parámetros solares.	Permite obtener los parámetros solares asociado a cada estudio.

CAPÍTULO 2: DESCRIPCIÓN DE LA APLICACIÓN.

RF#8	Crear Sombrigramas	Permite crear un nuevo Sombrigrama.
RF#9	Mostrar listado de Sombrigramas.	Muestra el listado de Sombrigramas.
RF#10	Actualizar Sombrigramas.	Permite modificar los datos de los Sombrigramas.
RF#11	Eliminar Sombrigrama.	Permite eliminar un sombrigrama del listado
RF#12	Crear Carta Solar.	Permite crear una carta solar.
RF#13	Mostrar listado de Cartas Solares.	Muestra el listado de Cartas Solares.
RF#14	Actualizar Carta Solar.	Permite modificar los datos de una Carta Solar.
RF#15	Eliminar Carta Solar.	Permite eliminar una Cara Solar.
RF#16	Autenticar usuario	Permite que un usuario de autentique en el sistema luego de haber sido registrado en el mismo.
RF#17	Añadir usuario.	Permite Añadir un usuario al Sistema.
RF#18	Mostrar listado de usuarios.	Muestra el listado de usuario.
RF#19	Modificar usuario.	Permite Modificar los datos de un usuario al Sistema.
RF#20	Eliminar usuario.	Permite Eliminar un usuario al Sistema.
RF#21	Modificar artículo	Permite Modificar un usuario al Sistema.
RF#22	Mostrar listado de artículos.	Muestra el un listado con los artículos registrados en el sistema.
RF#23	Añadir artículo.	Permite añadir a la portada los artículos con información de cómo se realizan los cálculos de parámetros en el sistema.
RF#24	Eliminar artículo	Permite eliminar de la portada los artículos con información de cómo se realizan los cálculos de parámetros en el sistema.

2.3.2 Requisitos no funcionales.

Los requisitos no funcionales (RNF) se refieren a las características o cualidades que debe tener el producto para hacerlo atractivo, usable, rápido y confiable. Para el desarrollo de la aplicación web propuesta se definieron 26 RNF los que se clasificaron en RNF de usabilidad, confiabilidad, eficiencia, soporte, restricciones de diseño, de interfaz, de licencia, estándares aplicables y seguridad.

CAPÍTULO 2: DESCRIPCIÓN DE LA APLICACIÓN.

Estos elementos no funcionales, constituyen los requisitos de la aplicación a implementar y fueron validados, partiendo de que fueron definidos en conjunto con el cliente. La tabla de RNF se muestra en el Anexo 4.

2.3.3 Validación de requisitos

Con el objetivo de verificar si los requisitos del *software* obtenidos definen el sistema que el cliente desea, se llevó a cabo un proceso de validación de los mismos, para el cual se emplearon las siguientes técnicas:

- ✓ **Revisiones de los requisitos:** se realizaron revisiones a cada uno de los requisitos por parte del equipo de desarrollo y por el cliente. Las revisiones internas generaron un total de 9 no conformidades de tipo técnicas y de ortografía, las cuales fueron corregidas satisfactoriamente en tiempo. Con el cliente se desarrollaron 2 revisiones. En la primera, el mismo quedó satisfecho con el trabajo efectuado por el equipo de desarrollo, sin embargo se añadieron detalles a 6 de los requisitos funcionales para su perfeccionamiento. En el segundo encuentro, se añadieron especificaciones a 3 requisitos funcionales y se aprobaron finalmente los requisitos, generándose de este encuentro el acta de aceptación de requisitos.
- ✓ **Construcción de prototipos:** mediante los prototipos se le mostró al cliente un modelo ejecutable del portal web que le permitió tener una visión preliminar de cómo sería el sistema y a través de la interacción con los mismos se comprobó si satisfacía sus necesidades, los mismos fueron aprobados por el cliente con un alto nivel de satisfacción.

2.3.4 Historias de usuarios.

Las historias de usuarios (HU) según la metodología de *software* empleada en la presente solución se define como una de las formas que se utilizan para encapsular requisitos, esta forma se evidencia en el cuarto escenario condicionado por Modelado del Negocio.

En la siguiente tabla se puede apreciar el listado de las mismas organizadas según las ARF:

Tabla 5: Distribución de HU por ARF.

CAPÍTULO 2: DESCRIPCIÓN DE LA APLICACIÓN.

ARF	Historias de Usuarios	Sprint ¹¹
Administración	HU_1: Autenticar usuario.	1
	HU_2: Administrar usuario.	1
	HU_3: Administrar artículos	1
Estudios	HU_4: Realizar estudio simple	2
	HU_5: Gestionar estudio avanzado	2
	HU_6: Gestionar Sombrigramas	2
	HU_7: Gestionar Carta Solar	2

A continuación se ilustra la historia de usuario HU_5: Gestionar estudio avanzado.

Tabla 6: Descripción HU_5: Gestionar estudio avanzado.

Historia de Usuario
Número: HU_2 Nombre Historia de Usuario: Gestionar Estudio
Modificación de Historia de Usuario Número: 3.
Usuario: Aniel Perez Orozco.
Prioridad en Negocio: Alta.
Riesgo del Desarrollo: Media.
<p>Descripción: El usuario una vez autenticado accede al módulo <i>Estudios</i> y selecciona la opción <i>Estudio Avanzado</i>, El sistema muestra una lista de estudios generados por el usuario, en el cual se le permitirá <i>adicionar, modificar y eliminar</i> estudios avanzados.</p> <p><i>Adicionar estudio:</i> permitirá añadir un nuevo estudio los siguientes datos iniciales: nombre del proyecto, localidad, latitud y longitud geográfica, altura del punto proyectante, meridiano de referencia, la fecha de inicio y la hora de inicio, fecha fin y hora fin, además selecciona el tipo de algoritmo(Alás, Alás-Cooper o Alás-Spencer) por donde desea obtener los parámetros solares y el intervalo en que desea que se realice estos cálculo(horas o minutos), algunos de los</p>

¹¹**Sprint:** Terminología para definir las iteraciones en la metodología.

CAPÍTULO 2: DESCRIPCIÓN DE LA APLICACIÓN.

datos que el usuario debe brindar al sistema son: longitud y latitud geográfica, altura del punto proyectante, si es un estudio simple o avanzado, etc. Además el usuario seleccionará el tipo de algoritmo a utilizar para realizar los cálculos y luego obtendrá dichos resultado (fracción del día, ángulo horario, ecuación del tiempo, altura de culminación, altura solar, azimut solar, azimut de la sombra, longitud de la sombra, declinación solar y la hora de paso por el meridiano).

Modificar: Permitirá al usuario modificar algunos de los datos iniciales: nombre del proyecto, localidad, latitud y longitud geográfica, altura del punto proyectante, meridiano de referencia, la fecha de inicio y la hora de inicio, fecha fin y hora fin, además selecciona el tipo de algoritmo (Alás, Alás-Cooper o Alás-Spencer) por donde desea obtener los parámetros solares y el intervalo en que desea que se realice estos cálculo(horas o minutos), algunos de los datos que el usuario debe brindar al sistema son: longitud y latitud geográfica, altura del punto proyectante, si es un estudio simple o avanzado, etc. Además el usuario seleccionará el tipo de algoritmo a utilizar para realizar los cálculos y luego obtendrá dichos resultado (fracción del día, ángulo horario, ecuación del tiempo, altura de culminación, altura solar, azimut solar, azimut de la sombra, longitud de la sombra, declinación solar y la hora de paso por el meridiano).

Eliminar: El usuario podrá eliminar un estudio de la lista.

Mostrar Estudio Avanzado: Se muestran los resultados del cálculo de los parámetros solares en función del intervalo de tiempo seleccionados por el usuario y se generan estos resultados en formato excel.

Observaciones: El usuario tiene que seleccionar un tipo de algoritmo para calcular los parámetros de forma obligatoria, sino el sistema no efectuara los cálculos.

Prototipo de Interface:

- ✓ Lista de estudio

CAPÍTULO 2: DESCRIPCIÓN DE LA APLICACIÓN.

✓ Añadir estudio

Añadir Estudio

Tipo de Estudio
 Estudio Simple Estudio Avanzado

Fecha Inicio: [] Fecha Fin: []
 Localidad: [] Hora Inicio: [] Hora Fin: []
 Proyecto: [] Oficial Solar

Latitud Geográfica: [] Meridiano de Referencia: []
 Longitud Geográfica: [] Este Oeste
 Altura del Punto Proyectante: [] Realizado por: []

[Obtener Parámetros]

Parámetros Solares

Nd: [] Altura Solar: []
 Fd: [] Horas de Sol: []
 Hora Solar: [] Azimut Solar: []
 Ángulo Horario: [] Declinación Solar: []
 Ecuación del Tiempo: [] Hora de salida del Sol: []
 Altura de Culminación: [] Hora de la puesta del Sol: []
 Hora de paso por el meridiano: []

Coordenadas Solares
 Azimut de la Sombra: []
 Longitud de la Sombra: []

Coordenadas Cartesianas
 X: []
 Y: []

[Aceptar] [cancelar]

✓ Mostrar estudio

Estudio Avanzado

Tipo de Estudio
 Estudio Simple Estudio Avanzado

Fecha Inicio: [] Fecha Fin: []
 Localidad: [] Hora Inicio: [] Hora Fin: []
 Proyecto: [] Oficial Solar

Latitud Geográfica: [] Meridiano de Referencia: []
 Longitud Geográfica: [] Este Oeste
 Altura del Punto Proyectante: [] Realizado por: []

[Obtener Parámetros]

Parámetros Solares

Nd: [] Altura Solar: []
 Fd: [] Horas de Sol: []
 Hora Solar: [] Azimut Solar: []
 Ángulo Horario: [] Declinación Solar: []
 Ecuación del Tiempo: [] Hora de salida del Sol: []
 Altura de Culminación: [] Hora de la puesta del Sol: []
 Hora de paso por el meridiano: []

Coordenadas Solares
 Azimut de la Sombra: []
 Longitud de la Sombra: []

Coordenadas Cartesianas
 X: []
 Y: []

[Generar PDF] [Generar documento] [cancelar]

✓ **RF#1: Crear estudio avanzado.**

✓ **RF#2: Mostrar resultados estudios avanzados.**

CAPÍTULO 2: DESCRIPCIÓN DE LA APLICACIÓN.

- ✓ **RF#3: Actualizar estudios avanzados.**
- ✓ **RF#4: Eliminar estudio avanzado.**
- ✓ **RF#6: Exportar estudio.**
- ✓ **RF#8: Obtener parámetros solares.**

2.4 Arquitectura

La Arquitectura de *Software* se define, a grandes rasgos, como una vista del sistema que incluye los componentes principales del mismo, su conducta, y las formas en que interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones (Clements, 1996).

Los patrones arquitectónicos son aquellos que expresan un esquema organizativo estructural fundamental para sistemas de *software*. Un patrón de arquitectura de *software* es un esquema genérico probado para solucionar un problema particular recurrente que surge en un cierto contexto. Este esquema se especifica describiendo los componentes, con sus responsabilidades, relaciones, y las formas en que colaboran (Larman, 1999.).

El desarrollo de la solución define, a partir del marco de trabajo Symfony2, una arquitectura basada en cuatro capas: (Capa de Presentación, la Capa de Negocio, la Capa de Acceso a Datos y la Capa de Datos), con el uso del patrón arquitectónico **Modelo-Vista-Controlador (MVC)**, muy beneficioso en aplicaciones que manejan gran cantidad de datos y transacciones complejas, donde se requiere una mejor separación de conceptos facilitando la programación en diferentes capas de manera paralela e independiente (Gómez, 2011).

MVC sugiere la separación del *software* en 3 estratos: Modelo, Vista y Controlador, que serán explicados brevemente.

- ✓ **Modelo:** Es la representación de la información que maneja la aplicación. Básicamente son los datos puros que puestos en contexto del sistema proveen de información al usuario a la aplicación misma (Gómez, 2011).

CAPÍTULO 2: DESCRIPCIÓN DE LA APLICACIÓN.

- ✓ **Vista:** Es la representación del modelo en forma gráfica disponible para la interacción con el usuario. En el caso de una aplicación web, la “vista” es una página HTML con contenido dinámico sobre el cual el usuario puede realizar operaciones (Gómez, 2011).
- ✓ **Controlador:** Es la capa encargada de manejar y responder las solicitudes del usuario, procesando la información necesaria y modificando el modelo en caso de ser necesario (Gómez, 2011).

La arquitectura propuesta, además posee la característica de tener varios complementos transversales a las capas para garantizar seguridad, tratamiento de excepciones, entre otros aspectos. Cada uno de los módulos o bundles se estructuran arquitectónicamente igual. Las entidades del dominio son accesibles en la sub-capa servidor de la Capa de Presentación, la Capa de Negocio y la Capa de Acceso a Datos.

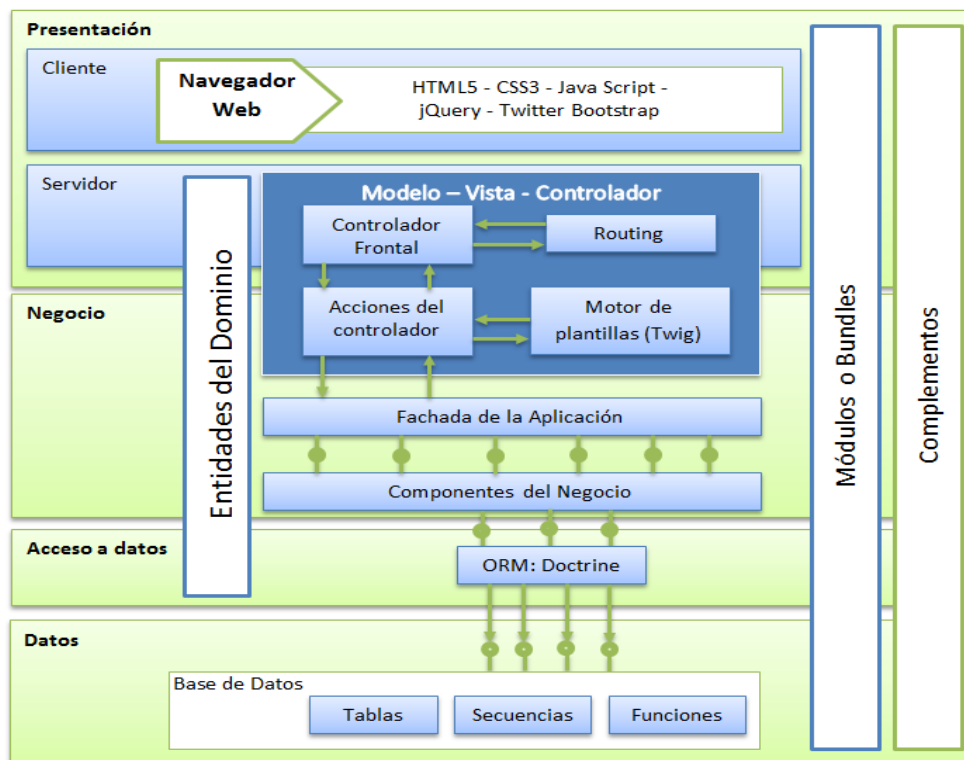


Figura 3: Esquema de la arquitectura.

- ✓ **Capa de Presentación:** en esta capa están los componentes con los que el usuario va a interactuar (páginas). Permitiendo alcanzar las funcionalidades que brinda el

CAPÍTULO 2: DESCRIPCIÓN DE LA APLICACIÓN.

controlador y mostrar o capturar la información a través de los diferentes elementos que comprende, a decir: *twigs*, formularios, *grids*, entre otros.

Está se divide en dos sub-capas: cliente y servidor. En la primera se visualiza mediante el navegador web (utilizando tecnologías como HTML5, CSS3, JavaScript, Twitter-Bootstrap y JQuery) datos procesados. En la segunda se maneja la lógica de control así como la construcción de páginas y formularios.

- ✓ **Capa de Negocio:** Su diseño depende directamente del negocio específico al que se refiera cada sistema. Esta capa recibe una petición del nivel superior y gestiona o procesa la misma. Finalmente envía una respuesta continuando el proceso en el punto donde se inició dicha petición. Contiene las clases gestoras, encargadas del manejo de la lógica de negocio. Dichas clases son gestionadas mediante una fachada de aplicación que desacopla los componentes del negocio de las clases controladoras.
- ✓ **Capa de Acceso a datos:** Contiene las entidades y repositorios que mapea Doctrine como marco de trabajo para la comunicación con el servidor de datos. Gestiona las peticiones de la capa de negocio consultando la BD y retornando los datos que se recuperan al gestor correspondiente. Es importante destacar que las entidades del dominio se encuentran en esta capa pero son accesibles además desde las capas superiores.
- ✓ **Capa de datos:** En esta capa se encuentra el gestor de base de datos PostgreSQL y en él un conjunto de esquemas, tablas, vistas y procedimientos almacenados que permiten persistir la información con la que trabaja la aplicación y manejar su almacenamiento.
 - ✓ **Complementos:** Los Complementos son un grupo de facilidades y mejoras que permiten lograr una arquitectura más flexible y adaptable tales como gestión de seguridad, de validaciones, de mensajería y de configuración así como tratamiento de excepciones y otras.
 - ✓ **Módulos (Bundles):** Cada módulo constituye una estructura de carpetas que se organizan según la arquitectura que se describe en la figura 3. Permiten utilizar funcionalidades construidas por terceros o empaquetar sus propias funcionalidades para distribuirlas y reutilizarlas.

2.5 Patrones de diseño

En términos generales, un patrón es un conjunto de información que proporciona respuesta a un grupo de problemas similares, es decir, un patrón es una solución a un problema en un contexto (Grasp, 2011).

Además del uso del patrón arquitectónico MVC, en la solución propuesta se emplean distintos tipos de patrones de diseño como son: de asignación de responsabilidad, de comportamiento; los cuales se describen a continuación:

2.5.1 Patrones de Diseño de Asignación de Responsabilidades o GRASP (Patrones Generales de *Software* para Asignar Responsabilidades).

Los patrones GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable (Grosso, 2011).

Estos patrones se describen a continuación:

- ✓ **Experto:** este patrón es aplicado en todas las clases debido a que cada una de ellas es experta pues contienen la información necesaria para cumplir con las responsabilidades que le fueron asignadas; facilitando así el entendimiento, extensión y mantenimiento del sistema. Además de conservar el encapsulamiento y contribuir al bajo acoplamiento (Grosso, 2011).



Figura 4: Empleo del patrón Experto en la clase Estudio.

CAPÍTULO 2: DESCRIPCIÓN DE LA APLICACIÓN.

- ✓ **Creador:** el patrón se evidencia en las clases controladoras que, para cada uno de los módulos o funcionalidades de la aplicación, son las encargadas de crear las instancias de los objetos que manejan, favoreciendo así la reutilización y el bajo acoplamiento (Grosso, 2011).

```

| public function simpleAction()
| {
|     $estudioSimple = new EstudioSimple() ;
|     $formEstudioSimple = $this->createForm(new
EstudioSimpleType(), $estudioSimple);
|
|     return array(
|         "form" => $formEstudioSimple->createView()
|     );
| }

```

Figura 5: Evidencia del patrón Creador en la clase *EstudioSimpleController.php*.

- ✓ **Bajo Acoplamiento:** se proporciona un bajo acoplamiento en el diseño debido a que las clases existentes tienen asignadas responsabilidades de tal forma que estas no dependan en gran medida de otras, permitiendo de esta forma tener sistemas más robustos y de fácil mantenimiento (Grosso, 2011).

Esta característica permitió potenciar la reutilización y disminuyó la dependencia entre las clases. Se evidencia en la clase *EstudioAvanzado*.

```

<?php
namespace Sunarq\CoreBundle\Entity;
use Doctrine\ORM\Mapping as ORM;
/**
 * EstudioAvanzado
 *
 * @ORM\Table()
 * @ORM\Entity
 */
class EstudioAvanzado extends Estudio
{
    /**
     * @var integer
     *
     * @ORM\Column(name="id", type="integer")
     * @ORM\Id
     * @ORM\GeneratedValue(strategy="AUTO")
     */

```

Figura 6: Evidencia del patrón Bajo Acoplamiento en las clases: *EstudioAvanzado*.

- ✓ **Alta Cohesión:** el patrón se evidencia debido a que a cada una de las clases se le asignaron responsabilidades de tal forma, que estén estrechamente relacionadas entre sí y no realicen un trabajo excesivo (Grosso, 2011).

```
<?php

namespace Sunarq\CoreBundle\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Template;
use Sunarq\CoreBundle\Entity\EstudioSimple ;
use Sunarq\CoreBundle\Form\EstudioSimpleType ;

/**
 * @Route("/estudio/simple")
 */
class EstudioSimpleController extends Controller
{
```

Figura 7: Evidencia del patrón Alta Cohesión en las clases: *EstudioSimpleController*.

2.3.1 Patrones de Comportamiento (Patrones GOF):

En la presente solución no se define el uso de un nuevo patrón de comportamiento, debido a que se hará uso de los patrones de GOF que utiliza el marco de trabajo Symfony los cuales son descritos en el Capítulo 1.

2.4 Diagrama de clases del Diseño con estereotipos web.

El propósito del diagrama es el de representar las diferentes clases que se utilizan en el sistema, así como las relaciones que existen entre estas.

Para el Sistema, se elaboraron un total de 5 diagramas, a continuación se muestra el diagrama de clases con estereotipo web para la clase gestionar Estudios avanzado.

CAPÍTULO 2: DESCRIPCIÓN DE LA APLICACIÓN.

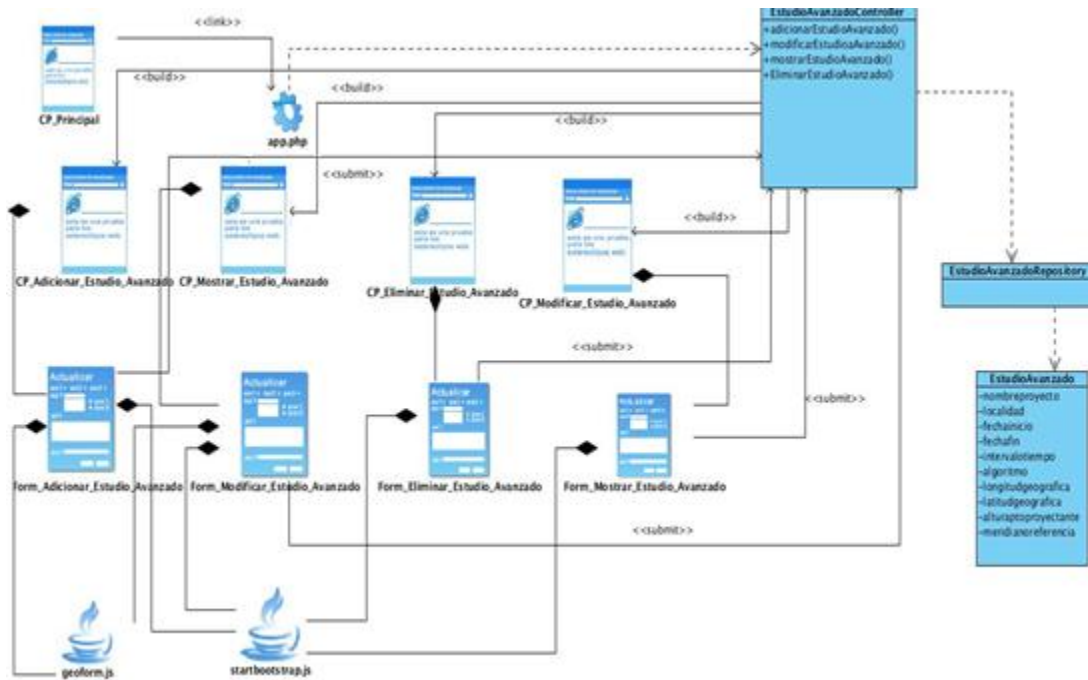


Figura 8: Diagrama de clases del diseño con estereotipos web: Gestionar estudio avanzado.

2.5 Modelo de datos.

El modelo de datos es un conjunto de conceptos que sirven para describir la estructura, semántica y las relaciones existen entre las entidades de una base de datos importantes para el funcionamiento del negocio y muestra los datos que serán contenidos en el sistema (Fernández, 2011).

A continuación se muestra la estructura de la Base de Datos del sistema SIPCSOL, la cual permite realizar la representación lógica de los datos procesados por el sistema . Este modelo muestra las entidades de datos, los atributos asociados a cada una de ellas y las relaciones entre las mismas.

CAPÍTULO 2: DESCRIPCIÓN DE LA APLICACIÓN.

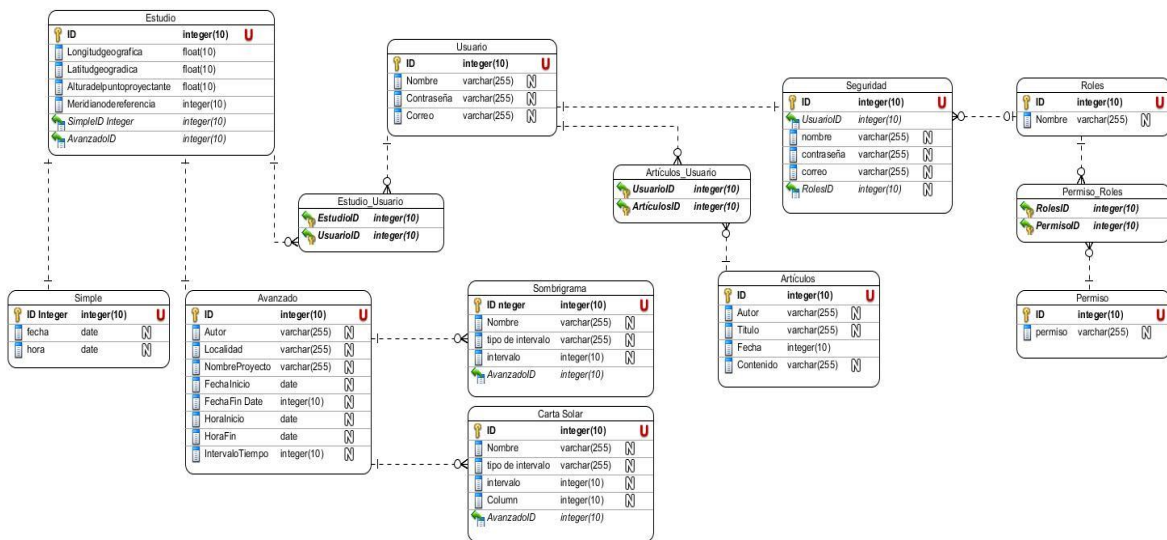


Figura 9: Modelo de datos.

El modelo de datos obtenido cuenta con un total de 13 tablas que definen los tipos de actividades que existirán en el sistema (Estudio simple, Estudio avanzado, Carta Solar, Sombragrama, Artículos, Usuario,).

Durante el modelado de la base de datos se emplearon los siguientes patrones de diseño:

Patrones de diseño de la base de datos.

- ✓ **Llaves subrogadas:** El uso de este patrón es bastante generalizado en la base de datos pues la mayoría de las entidades contienen una llave única entera auto-incremental, lo que facilita reducir el costo de las búsquedas en la base de datos. EL uso de este patrón constituye una protección ante los cambios debido a que la lógica del negocio no está en las llaves y evita la contención (bloqueo) de la base de datos, debido a la rapidez de los mecanismos de generación que provee el sistema.

CAPÍTULO 2: DESCRIPCIÓN DE LA APLICACIÓN.

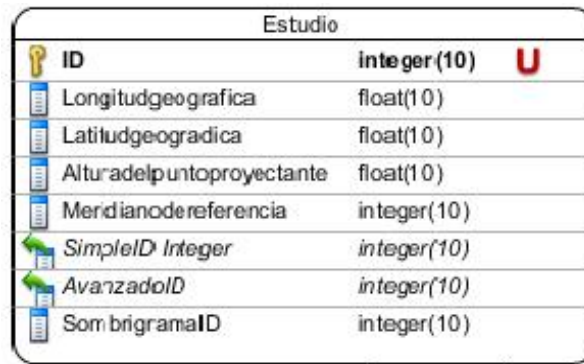


Figura 10: Uso del patrón Llaves Subrogadas en la entidad Estudio.

- ✓ **Control de Acceso Basado en Roles (RBAC, por sus siglas en inglés):** este patrón se implementa mediante la asignación de roles a los diferentes usuarios del portal, a través de los cuales dichos usuarios obtienen los permisos para realizar determinadas funciones en el mismo; lo que simplifica la gestión de los usuarios debido a que el manejo de los permisos de estos se reduce solo a asignarle el rol apropiado.

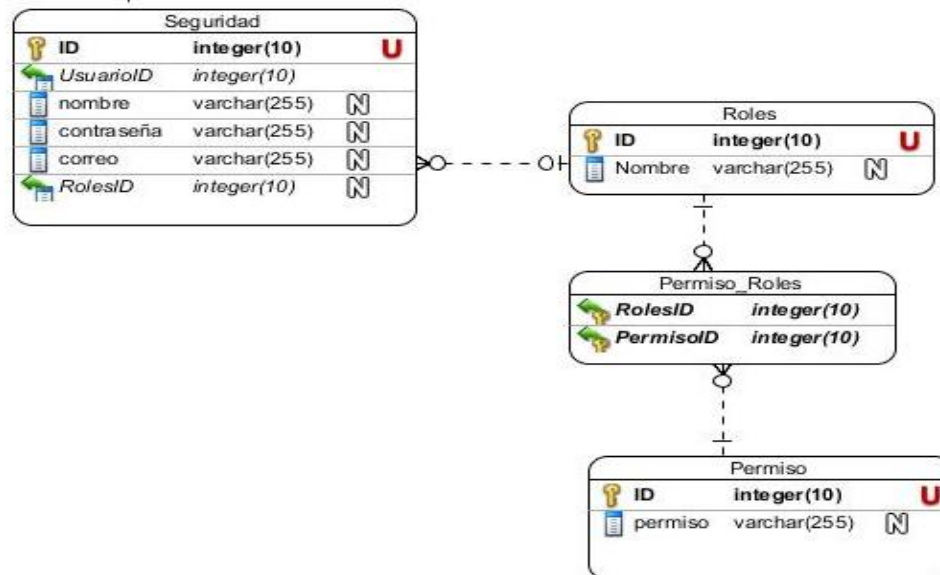


Figura 11: Empleo del patrón Control de Acceso Basado en Roles (RBAC).

2.6 Conclusiones Parciales.

En el presente capítulo se realizó la descripción de la aplicación mediante el uso de la metodología seleccionada en su primera etapa (Inicio y Ejecución).

CAPÍTULO 2: DESCRIPCIÓN DE LA APLICACIÓN.

Se presentó la propuesta de solución que sugiere la investigación, basada en el problema y el estado del arte establecido en el Capítulo 1. El flujo de trabajo generó el Modelo Conceptual, el cual reúne los principales conceptos del sistema y las relaciones que se establecen entre ellos.

Como parte de la realización del segundo flujo (Captura de Requisitos), se aplicaron técnicas de ingeniería de requisitos para su captura. Definiéndose un total de 24 RF, y 26 RNF los que fueron descritos y agrupados por categorías.

Se realizó un estudio sobre todas las HU definidas como punto de partida para el diseño y realización de los prototipos del sistema. Lo que se concretó con la elaboración de las Plantillas de Historias de Usuario, que aglutinan su descripción, junto al prototipo y los requisitos a los que responde la HU.

Posteriormente se realizó un análisis sobre la arquitectura a la cual responde el sistema, resaltando el uso del patrón arquitectónico MVC y el comportamiento de los componentes que interactúan de forma transversales a las capas para garantizar seguridad, tratamiento de excepciones, entre otros aspectos. Luego, se realizó un análisis sobre el uso de los patrones de diseño utilizados durante el desarrollo de la aplicación.

Se elaboraron los diagramas de clases del diseño con estereotipo web con el propósito de representar las principales clases y la forma en que se relacionan las mismas. Por último, para describir la estructura, semántica y las relaciones existen entre las entidades de una base de datos se realizó el Modelo de Datos de la Aplicación y se validaron los requisitos, aplicando un conjunto de técnicas.

CAPÍTULO III

IMPLEMENTACIÓN Y VALIDACIÓN

En la realización de este capítulo se describe el proceso de implementación de la aplicación en términos de componentes; donde se detalla mediante el diagrama de despliegue como quedará distribuida la aplicación. Se muestran los resultados obtenidos de la aplicación de las métricas y técnicas empleadas para validar los requisitos y el diseño del sistema; además de los resultados alcanzados luego de la realización las pruebas. De ello, se generan los principales productos de trabajo definidos por la metodología, correspondientes a la fase de ejecución, en sus disciplinas (Implementación y Pruebas internas, de liberación y de aceptación).

3.1 Implementación

La implementación constituye una de las fases más importantes del desarrollo de *software*. En ella se toman como punto de partida los resultados obtenidos en el diseño, implementándose el sistema en términos de componentes como ficheros de código binario, código fuente, scripts y ejecutables. Su importancia reside en que se obtiene como consecuencia un sistema ejecutable, siendo esto uno de los principales objetivos en el desarrollo de *software* (Arizaca, 2009).

3.1.1 Diagrama de componentes

Un diagrama de componentes representa la separación de un sistema de *software* en componentes físicos (por ejemplo archivos, módulos, paquetes, base de datos, código fuente, binario o ejecutable etc.) y muestra las dependencias y organización existente entre estos componentes (Arizaca, 2009).

En el diagrama de componentes de la Figura 12, se evidencian las interacciones entre el los componentes de Symfony2 y los del sistema SIPCSOL. Symfony2 cuenta con los componentes *Security* (para la seguridad), *Routing* (para las rutas de acceso), *Config* (para las configuraciones), *Mensajería* (para la gestión de mensajes informativos y de errores), *Excepciones* (para el manejo de las excepciones generadas) y *Ctr_Frontal* es el controlador frontal que recibe cada petición realizada al sistema, entre otros. Estos componentes permiten un adecuado manejo de la seguridad, las configuraciones, las excepciones y errores del subsistema.

Mientras que el sistema SIPCSOL está compuesto por tres paquetes fundamentales, Controlador Modelo y Vista.

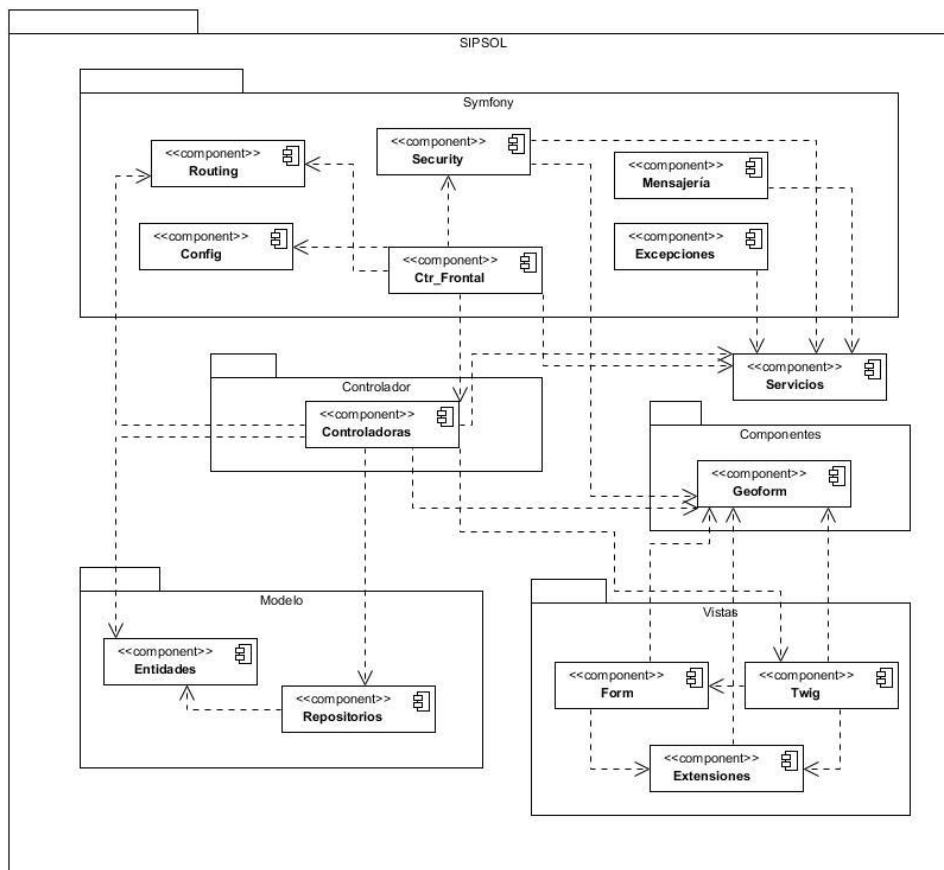


Figura 12: Diagrama de componentes.

El paquete Controlador contiene el componente Controladoras, responsable de dar respuesta a las peticiones realizadas por los usuarios, posee una relación con el componente Routing para el manejo de las rutas definidas para el módulo y el componente Servicios que ofrece los servicios públicos.

El componente Controladoras se relaciona, además, con el componente Entidades, que a su vez es usado por el componente Repositorio el cual contiene las clases repositorios encargadas de realizar las consultas a la base de datos. El paquete Vista incluye los componentes Form (encargado del trabajo con los formularios), Twig (para las interfaces de usuarios y plantillas del módulo) y Extensiones (posee los JavaScript y CSS).

De igual forma, estos componentes pueden ser agrupados y organizados atendiendo a la arquitectura del sistema para lograr un mejor entendimiento de sus funciones, distribución e interacciones entre sí.

3.1.2 Diagrama de despliegue

Los diagramas de despliegue proveen una vista de cómo los componentes se despliegan en el transcurso de la infraestructura del sistema. Permite comprender la correspondencia entre la arquitectura de *software* y la de hardware.

El siguiente diagrama de despliegue muestra la forma en que se van a comunicar los componentes del sistema. La PC cliente (usuario) se va a comunicar por el protocolo HTTP¹² para utilizar las funcionalidades que se encuentran en el Servidor web. Este a su vez se comunica con el Servidor de Base de Datos por medio del protocolo TCP/IP¹³ para la obtención de la información que maneja el sistema, también con el protocolo TCP/IP.

Además de ello, para el despliegue del sistema a implementar se tiene conectada al Servidor web por el protocolo HTTP la PC_Cliente (Admin) la cual cuenta con una impresora con conexión USB donde se podrá obtener de forma física los estudios e información generada en la aplicación figura 13

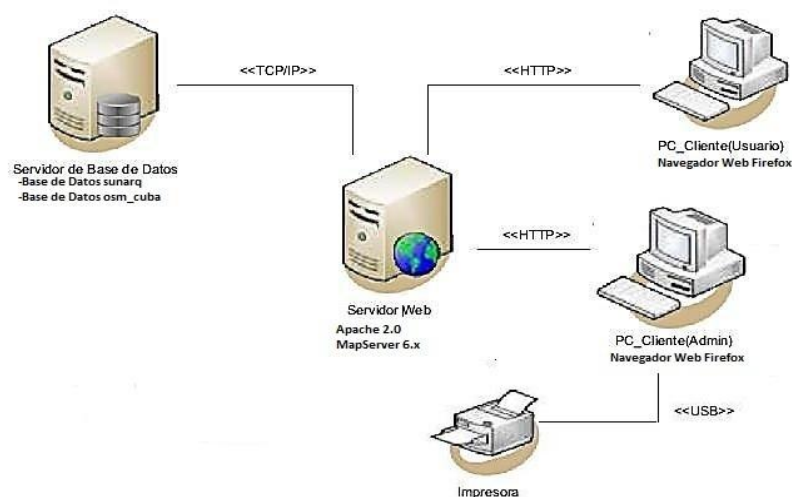


Figura 13: Diagrama de despliegue.

¹² HTTP: Protocolo de transferencia de hipertexto.

¹³ TCP/IP: Protocolo de Control de Transmisión.

3.1.3 Estándar de codificación empleado

Con el objetivo de lograr una estandarización en la programación de la aplicación web se decide aplicar el estilo de escritura empleados en estándares de codificación *CamelCase*, específicamente la variante *lowerCamelCase*. El cual facilitará la lectura, comprensión y mantenimiento del código.

A continuación se describe a grandes rasgos las convenciones de nomenclatura.

General:

- ✓ Se exceptúan el uso de las tildes y la letra ñ, la que será sustituida por nn.
- ✓ En todo momento se utilizarán nombres que sean claros, concretos y libres de ambigüedades. Ejemplo: "idestudio" y no solamente "id".

El nombre de todas las variables y métodos comenzarán con letra minúscula y si este está compuesto por varias palabras, todas las palabras internas que lo componen comienzan con mayúscula. Ejemplo: "ecuaciondeTiempo (EstudioSimple \$estudioSimple)".

Indentación:

- ✓ El contenido siempre se indentará con *tabs*, nunca utilizando espacios en blanco.

Clases:

- ✓ El nombre de las clases comenzará con mayúscula si este está compuesto por varias palabras, todas las palabras internas que lo componen comienzan con mayúscula también. Ejemplo: "EstudioAvanzado".
- ✓ Intentar mantener los nombres de las clases descriptivos y simples. Usar palabras completas, evitar acrónimos y abreviaturas, a no ser que la abreviatura sea mucho más conocida que el nombre completo, como URL o HTML.

Nombre de variables:

- ✓ No se utilizarán nombres de variables que puedan ser ambiguos.
- ✓ Se procurará evitar dar nombres sin sentido a variables temporales. Por ejemplo: temp, i, tmp.
- ✓ Las variables booleanas deben tener nombres que sugieran respuestas o contenidos de tipo Sí/No, por ejemplo: "esCooper".
- ✓ Los nombres de las variables booleanas deben ser positivos, por ejemplo: "esCooper".

3.2 Validación

Con los productos de trabajos obtenidos en la disciplina Análisis y diseño se comenzará con la implementación, donde a partir de los resultados obtenidos anteriormente se implementará el sistema en términos de componentes, ficheros, código fuente, scripts ejecutables y similares. Con el objetivo de certificar esta disciplina se realizó la validación de los resultados obtenidos con el uso de métricas.

Según (Presman, 2007) las métricas para el modelado del diseño cuantifican los atributos de diseño de manera tal que le permiten al ingeniero de *software* evaluar la calidad del diseño. Estas métricas de modelado del diseño incluye: métricas arquitectónicas, métricas al nivel de componentes, métricas de diseño de interfaz y métricas especializadas en el diseño orientado a objeto.

3.2.1 Validación del diseño de la aplicación web

Para comprobar la calidad y fiabilidad del diseño del sistema SIPCSOL, se emplearon las métricas basadas en clases: Tamaño Operacional de Clase (TOC) y Relaciones entre Clases (RC), permitiendo el resultado de ambas métricas validar el diseño propuesto en la investigación. A continuación se muestra un resumen de los resultados de la aplicación de ambas métricas.

Tamaño Operacional de Clases (TOC)

La métrica TOC es una métrica especializada en diseño orientado a objeto, midiendo características de clases, además de las correspondientes a comunicación y colaboración. Está dado por el número de métodos y atributos asignados a una clase para evaluar los siguientes atributos de calidad: responsabilidad, complejidad de implementación y la reutilización de las clases del diseño. Es importante destacar que para esta métrica, la responsabilidad y la complejidad son inversamente proporcionales a la reutilización, por lo que a mayor responsabilidad y complejidad de implementación de una clase, menor será su nivel de reutilización (Lorenz, 1994). El tamaño operacional de una clase se puede determinar empleando medidas para saber el número total de operaciones que contiene.

A continuación se muestra el resultado de la aplicación de la métrica TOC en las 16 clases que juegan un papel fundamental en los procesos principales del sistema informático.

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN

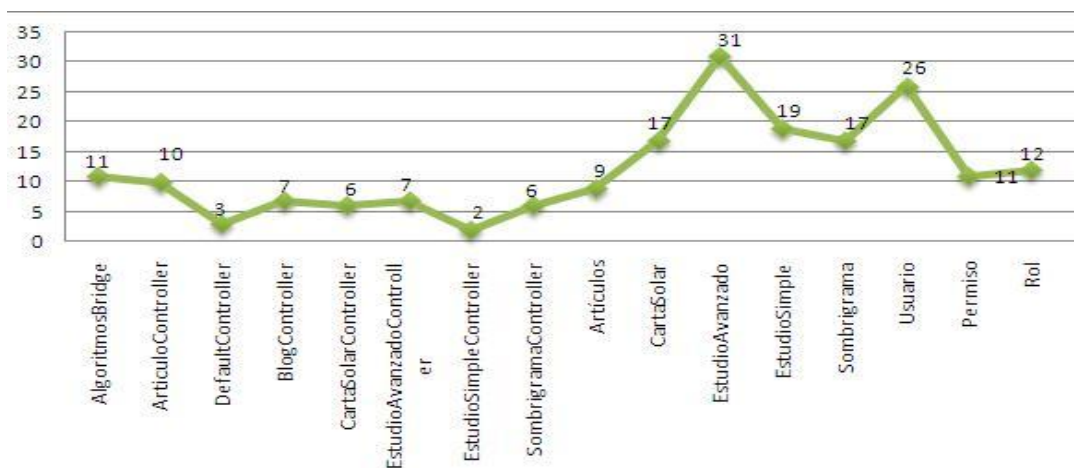


Figura 14: Cantidad de procedimiento por clases.

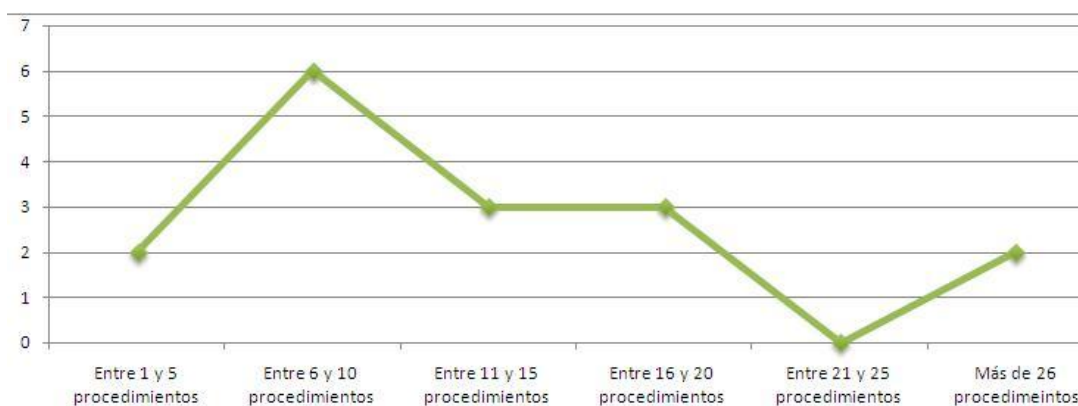


Figura 15: Resumen de cantidad de procedimientos.

Aplicando como umbral un promedio de 12,125 procedimientos (194 procedimientos en total /16 clases existentes), se obtienen los valores de los atributos de calidad: responsabilidad de las clases, complejidad al implementar las mismas, así como sus niveles de reutilización.

Tabla 7: Valores de las variables de calidad: Responsabilidad, Complejidad y Reutilización.

Nivel	Responsabilidad		Complejidad		Reutilización	
	Cantidad de Clases	Promedio	Cantidad de Clases	Promedio	Cantidad de Clases	Promedio
Baja	8	50	8	50	2	12
Media	6	37,5	6	37,5	6	38
Alta	2	12,5	2	12,5	8	50

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN



Figura 16: Resultado de las variables: Responsabilidad, Complejidad y Reutilización.

Luego de aplicada la métrica TOC se observa que las clases del diseño de la aplicación web no se encuentran sobrecargadas en cuanto a responsabilidades y el nivel de complejidad de las mismas no es muy alto, lo que favorece en gran medida la reutilización de estas.

Relaciones entre Clases (RC)

La métrica RC permite evaluar el acoplamiento, la complejidad de mantenimiento, la reutilización y la cantidad de pruebas de unidad necesarias para probar una clase teniendo en cuenta las relaciones existentes entre ellas. Luego de aplicar la métrica RC, se arrojaron los siguientes resultados:

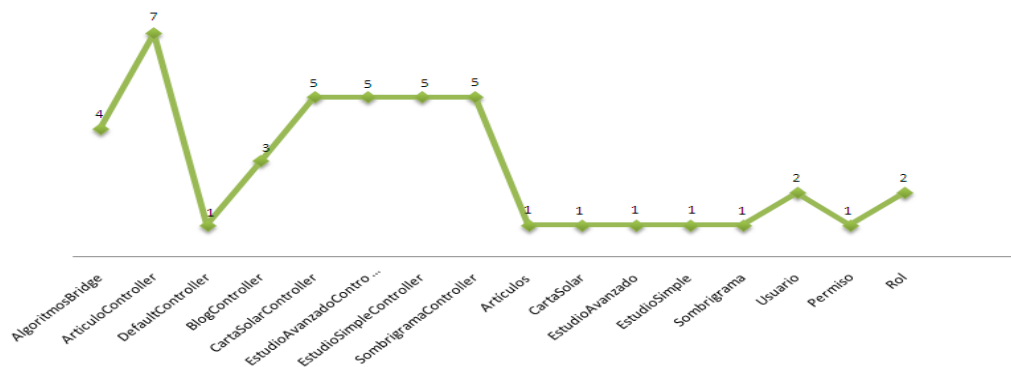


Figura 17: Cantidad de relaciones de uso por clases.

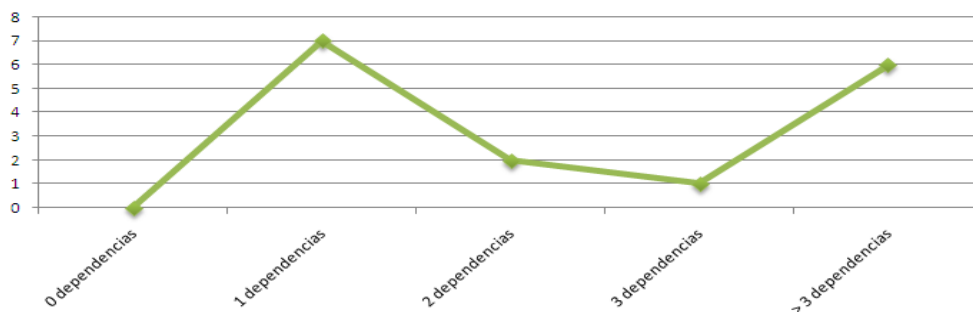


Figura 18: Resumen de relaciones de uso.

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN

Luego de calcular los valores de las variables de calidad: acoplamiento, complejidad de mantenimiento, reutilización y cantidad de pruebas, aplicando como umbral un promedio 3 asociaciones de uso por clase; obteniendo los siguientes resultados:

Tabla 8: Valores de las variables: Acoplamiento, Complejidad, Reutilización y Cantidad de Pruebas.

Nivel	Acoplamiento		Complejidad		Reutilización		Cant. Pruebas	
	Cantidad de Clases	Promedio	Cantidad de Clases	Promedio	Cantidad de Clases	Promedio	Cantidad de Clases	Promedio
Ninguno	0	0	-	-	-	-	-	-
Bajo	9	56,25	10	62,5	1	6,25	10	62,5
Medio	1	6,25	5	31,25	5	31,25	5	31,25
Alto	6	37,5	1	6,25	10	62,5	1	6,25

Una vez aplicada la métrica RC y teniendo en cuenta el umbral definido para validar el diseño, se obtiene como resultado que las clases promueven el bajo acoplamiento; la complejidad de mantenimiento y la cantidad de pruebas no son altas, y en consecuencia, el grado de reutilización es alto.

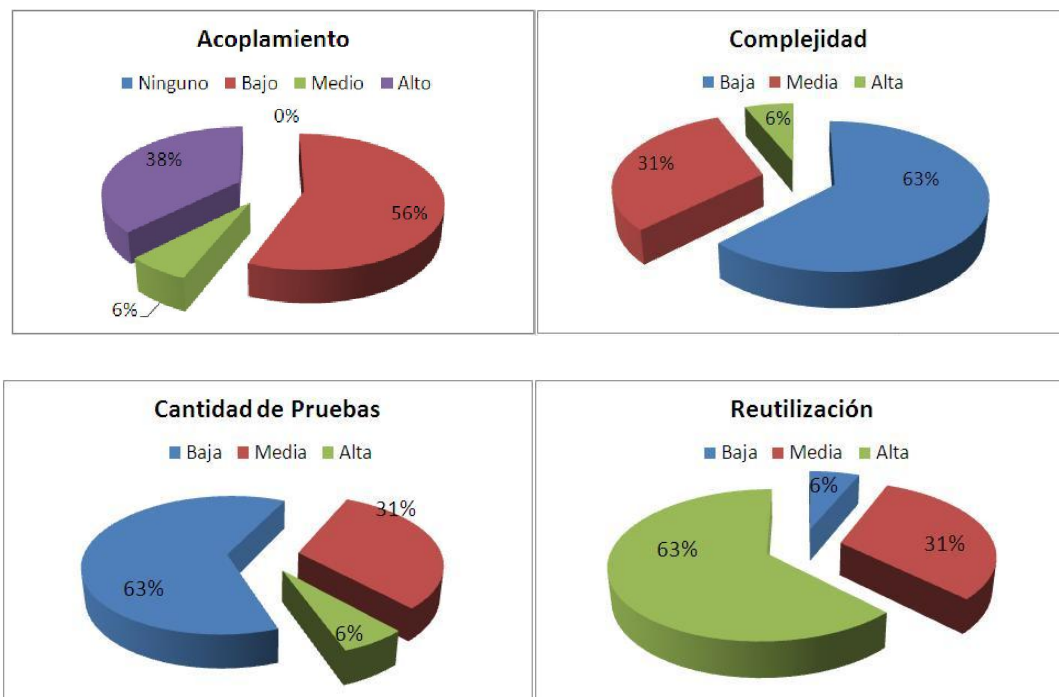


Figura 19: Resultado de las variables Acoplamiento, Complejidad, Reutilización y Cantidad de Pruebas.

En sentido general, los resultados obtenidos de la aplicación de las métricas TOC y RC demuestran que el diseño del sistema SIPCSOL no es complejo, que las clases presentan bajo acoplamiento y un alto grado de reutilización.

3.3 Pruebas

Las pruebas son un instrumento adecuado para verificar el estado de la calidad de un producto en las cuales un sistema o componente es ejecutado bajo condiciones o requisitos especificados. A la aplicación se le realizaron pruebas internas y de aceptación.

Pruebas Internas

En esta disciplina se verifica el resultado de la implementación probando cada construcción incluyendo tanto las construcciones internas como las intermedias, así como las versiones finales a ser liberadas. Se genera el artefacto de prueba diseño de casos de prueba.

3.3.1 Pruebas de Funcionalidad

El principal objetivo de técnica de Caja Negra es comprobar que las funcionalidades de la aplicación se realizan de forma correcta y responden a las necesidades del cliente, apoyándose en los casos de pruebas diseñados para cada funcionalidad, evidenciadas en el producto de trabajo Diseños de casos de Prueba, generado en la disciplina pruebas internas de la etapa de ejecución de la metodología Variación AUP-UCI.

Para comprobar la calidad de la herramienta SIPCSOL se realizaron 2 iteraciones por el Departamento de desarrollo de componentes del Centro CEIGE. Las no conformidades (NC) encontradas se clasificaron en: NC de la aplicación y NC del documento. En la siguiente figura se muestran las iteraciones realizadas y la cantidad de NC encontradas en cada iteración. En el anexo 5 se muestra el Acta de validación del Producto.

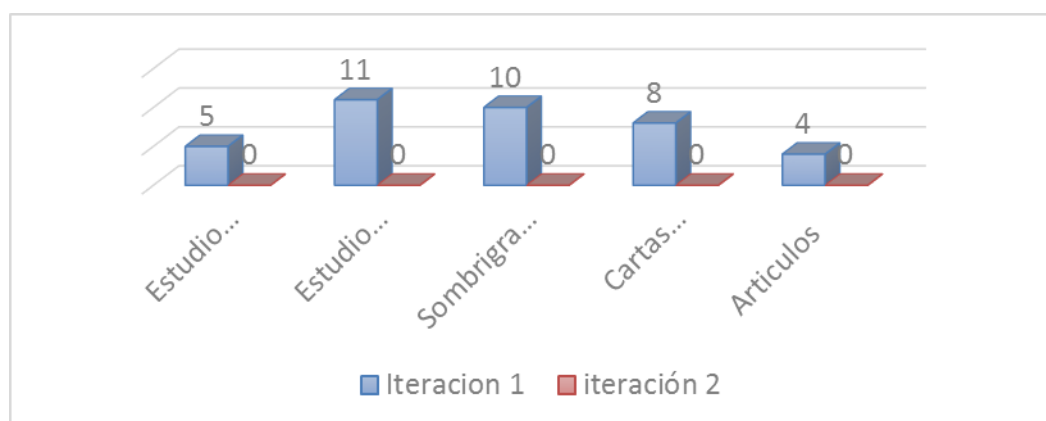


Figura 20: Relación de las NC, aplicando las pruebas de Caja Negra

3.3.2 Pruebas unitarias

Para comprobar que cada sentencia de código se ejecuta al menos una vez, se realizaron pruebas al código de las funcionalidades más complejas desde el punto de vista de la programación en cada uno de los módulos. Para ello se empleó el método de caja blanca con la técnica del *Camino Básico*; a partir de la obtención de la medida de la complejidad de un

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN

procedimiento o algoritmo y la obtención de un conjunto básico de caminos de ejecución de este, los cuales son utilizados para obtener los casos de prueba.

Seguidamente se muestra el proceso de pruebas realizado a el método `ecuaciondeTiempo()` perteneciente al módulo `AppBundle`, específicamente al `bridge AlgoritmosBridge`.

Para obtener los casos de prueba a partir de la técnica seleccionada se debe construir el grafo de flujo correspondiente al código de la funcionalidad. Luego se determina la complejidad ciclomática $V(G)$ del grafo resultante, la cual es un indicador del número de caminos independientes que existen en un grafo, es decir, es cualquier camino dentro del código que introduce por lo menos un nuevo conjunto de sentencias de proceso o una nueva condición(Rojas, 2010).

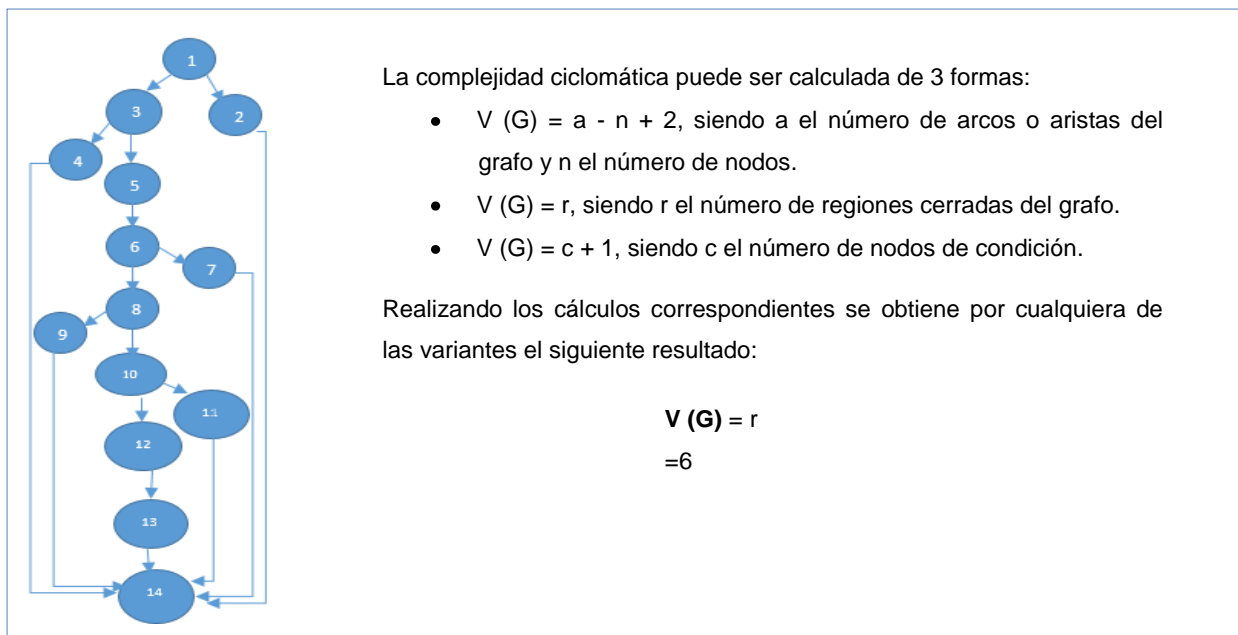


Figura 21: Grafo de flujo del código de la función `ecuaciondeTiempo()` y cálculo de su complejidad ciclomática.

Por lo que el conjunto de caminos básico sería:

Camino Básico 1: 1-2-14

Camino Básico 2: 1-3-4-14

Camino Básico 3: 1-3-5-6-8-10-11-14

Camino Básico 4: 1-3-5-6-7-14

Camino Básico 5: 1-3-5-6-8-9-14

Camino Básico 6: 1-3-5-6-8-10-11-14

Camino Básico 7: 1-3-5-6-8-10-11-12-13-14

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN

Luego se definen los casos de prueba para cada uno de los caminos básicos obtenidos. A continuación se muestra el resultado de las pruebas aplicadas a los caminos básicos 1 y 4.

Camino Básico 1: if (algoritmo == “Alas”) {ecuaciónTiempo= Valor1} retornar ecuaciónTiempo.

Camino Básico 4: if (algoritmo == “Alas-Cooper”) {ecuaciónTiempo = Valor2} entonces retornar ecuaciónTiempo.

Con la realización de estas pruebas se logró garantizar que todas las sentencias del programa fueran ejecutadas al menos una vez.

3.4 Resultados de la prueba de rendimiento

Para la realización de estas pruebas se utiliza la herramienta Apache JMeter específicamente para realizar pruebas de carga y estrés.

Esta prueba consistió en definir 100 hilos, los cuales constituyen la cantidad de usuarios que acceden concurrentemente al sistema haciendo peticiones al servidor. En la figura 22 se ilustran los resultados de dicha prueba.

Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
/mapcache	2400	117	26	63	7	4013	0,00%	466,4/sec	7902,5
Total	2400	117	26	63	7	4013	0,00%	466,4/sec	7902,5

Figura 22: Valores correspondientes a la prueba de rendimiento para 100 hilos.

La figura anterior representa el Informe agregado que genera la herramienta JMeter, en la cual se recoge el comportamiento de los siguientes indicadores:

- ✓ Número de muestras: cantidad de muestras de peticiones utilizadas para la URL
- ✓ Media: tiempo promedio en milisegundos transcurridos para un conjunto de resultados
- ✓ Mediana: mediana aritmética (elemento de una serie ordenada de valores crecientes de manera que divide en dos partes iguales).
- ✓ Min: tiempo mínimo transcurrido para las muestras de peticiones de una determinada URL.
- ✓ Max: tiempo máximo transcurrido para las muestras de peticiones de determinada URL.
- ✓ % Error: porcentaje de las peticiones con error.

Como se puede apreciar el tiempo promedio es de 1.17 segundos al realizar 2400 requerimientos al servidor.

El tiempo total utilizado para los 100 hilos se puede calcular con la siguiente fórmula:

Tiempo Total = #Muestras * Media = 2400 * 26=62400 milisegundos.

El tiempo promedio total requerido por cada hilo, se puede calcular de la siguiente manera:

Tiempo Total $/(1000 * 60 * \text{cantidad de hilos}) = 62400/(1000 * 60 * 100) = 0,0104$ minutos

3.5 Valoración del comportamiento de las variables de la investigación

Antes de la implantación del sistema, el cálculo de los parámetros solares en Cuba, se realizaba de manera manual, o utilizando algunas herramientas informáticas. Los resultados eran incongruentes, de igual forma, el acceso a estas herramientas era limitado; lo que conllevaba a:

- ✓ Se destinaba gran cantidad de tiempo al proceso de cálculo de los parámetros y la obtención de diagramas que simulan la trayectoria solar.
- ✓ La obtención de los parámetros y los Sombrigramas que se obtenían no eran precisos, ya que los datos aportados por estas herramientas solo estaban referidos a Ciudad de La Habana, latitud 23,09° y Santiago de Cuba, latitud 20,01°, además, los datos se dan para los días 21 de cada mes a intervalos de una hora con precisión de grados solamente.
- ✓ Las herramientas utilizadas para la obtención de los resultados no brindan información sobre los modelos matemáticos utilizados.

Todas estas deficiencias afectaban directamente garantizar precisión, rapidez y el acceso por parte de los usuarios interesados.

Variable Dependiente: garantizar precisión rapidez y acceso por parte de los usuarios interesados.

Variable Independiente: aplicación web para realizar y publicar el cálculo de los parámetros solares en Cuba.

A partir de los resultados del diagnóstico inicial, obtenido a través de la aplicación de diferentes instrumentos y una vez introducida la herramienta informática SIPCSOL se arriba a la siguiente conclusión; se logró mejorar la realización y publicación de los cálculos de parámetros solares, evidenciándose con los resultados emitidos por el sistema, a través de la interacción generada mediante la web de los usuarios finales del proceso.

Se registra una reducción considerable del tiempo en la realización de los cálculos. La estimación del tiempo en horas, se realizó a través de la medición del mismo por la técnica (Cálculo del Tiempo Estándar o Tipo), donde se midió el tiempo estándar en la ejecución manual de una de las operaciones más complejas: *Realizar estudio avanzado*.

Con el uso de la técnica se realizaron un total de tres mediciones a varios expertos en diferentes momentos, antes de implantar el sistema y luego de su puesta en práctica. Los resultados fueron los siguientes:

Tabla 9: Medición del tiempo del proceso de forma manual: Realizar estudio Avanzado (Antes del sistema).

Medición 1	Tt (Tiempo Total)
Realizar estudio avanzado	1.46 horas
Medición 2	Tt (Tiempo Total)
Realizar estudio avanzado	1.20 horas
Medición 3	Tt (Tiempo Total)
Realizar estudio avanzado	1.42 horas
Tiempo Estándar ($\Sigma(Tt)$)	4.08 horas

Tabla 11: Medición del tiempo del proceso: Realizar estudio (Con el sistema).

Operaciones	Tt (Tiempo Total)
Realizar estudio avanzado	0.0134 horas
Operaciones	Tt (Tiempo Total)
Realizar estudio avanzado	0.0021 horas
Operaciones	Tt (Tiempo Total)
Realizar estudio avanzado	0.0224 horas
Tiempo Estándar ($\Sigma(Tt)$)	0.0379horas

La medición anterior permite demostrar que existe una reducción aproximada del tiempo en la ejecución del proceso de 4,0421 horas.

El sistema muestra gran precisión en los resultados de las operaciones realizadas, debido a que dichos resultados no están referidos solamente a Ciudad de La Habana, latitud 23,09° y Santiago de Cuba, latitud 20,01°. Permite obtener los parámetros solares para cualquier lugar del territorio nacional y con los valores de coordenadas geográficas de cada lugar. Por otra parte, los resultados emitidos no se dan para los días 21 de cada mes, a intervalos de una hora, con precisión de grados solamente, sino que se muestran a distintos intervalos de horas y minutos y en el formato 21°35'26 (grados, minutos y segundos).

3.5 Conclusiones Parciales

En este capítulo se realizó la validación del componente arrojando resultados importantes como:

Se validaron las fases de desarrollo abarcada durante el desarrollo del producto como buenas prácticas permitiendo obtener un producto de calidad.

Se realizaron pruebas de *software*, arrojando resultados satisfactorios en el desarrollo del sistema como producto.

Por último, se realizó una valoración del comportamiento de las variables que forman parte del problema de la investigación, demostrando que con el sistema desarrollado, la realización y publicación de los parámetros solares se logró mejorar, evidenciándose una reducción del tiempo en el procesamiento de obtención de los parámetro, garantizando además la precisión de los resultados emitidos y la accesibilidad del sistema.

CONCLUSIONES GENERALES

CONCLUSIONES GENERALES

Sobre la base del análisis, interpretación y sistematización de las indagaciones teóricas y empíricas, a continuación se presentan las siguientes conclusiones de la investigación:

1. Se realizó el marco teórico de la investigación donde se establecieron las tendencias marcadas en cuanto al desarrollo de *software* para cálculos solares y los modelos matemáticos que estos utilizan, lo que permitió fundamentar la necesidad de desarrollar el sistema SIPCSOL.
2. Se desarrolló una aplicación web que permitió realizar y publicar el cálculo de los parámetros solares en Cuba, garantizando precisión, rapidez y al acceso al sistema por parte de los usuarios interesados.
3. Las pruebas realizadas a SIPCSOL, demuestran que es posible su despliegue, cumpliendo con los requisitos definidos por el cliente.

RECOMENDACIONES

El autor de este trabajo recomienda:

1. Divulgar los principales resultados de la investigación, a partir de la presentación y aplicación de la propuesta, a instituciones que realicen el cálculo de parámetros solares en Cuba.
2. Integrar futuras versiones del sistema con la incorporación del mapa global.

BIBLIOGRAFÍA REFERENCIADA

PostGIS — Spatial and Geographic Objects for PostgreSQL. [online]. [Accessed 3 June 2015]. Available from: <http://postgis.net/>

CAROLINA MARTINEZ. ISSUU - Patrones de Diseño. Ingeniería de Software 1 - CI3711 de Carolina Martinez. [online]. [Accessed 3 June 2015]. Available from: http://issuu.com/carolinamartinez24/docs/patronesdedise__o_aj2015

JUAN CARLOS GONZALEZ MORENO. Tema 1 Ingeniería de Requisitos. [online]. 2009. [Accessed 3 June 2015]. Available from: <http://www.slideshare.net/jcgmoreno/tema-1-ingeniera-de-requisitos>

Open Systems Developmen OSD.<http://www.osdglobal.com/faq/desarrollo-software/comparativo-web-vs-escritorio>.

Ahlgren, Joseph R. 2004.Sun directamente sobre la cabeza. Republica de Colombia : s.n., 2004.

Alfonso, X. C. G. Y. J. M.

Mosaic. 2007. Introducción a los Sistemas de Gestión de Contenido . [En línea] 2007. [Citado el: 09 de 02 de 2015.] <http://mosaic.uoc.edu/articulos/cms1204.html..>

2010.Algunas Conceptos, Definiciones y Ecuaciones Usadas en la Geometría Solar. 2010.

Álvarez, Sara. 2007. Sistemas gestores de bases de datos. Introducción a este concepto y características especiales. [En línea] 2007. . [En línea] 2007. [Citado el: 09 de 02 de 2015.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html..>

Amador, Rubén Rodríguez. 2013.Geometría Solar. Santiago de Chile : s.n., 2013.

Areces, G.A., Diaz, Marquez Iskael, "Diseño e implementación de las capas de negocio y acceso a datos de los módulos Planificación y Ejecución de Visitas Familiares". 2008.

ARIZACA, R. E. Artefacto: Diagrama de Componentes. La Paz, Bolivia, 2009.

Artículo Tecnológico. Rodrigo. 2010. <http://culturacion.com/2010/01/caracteristicasdeunaaplicacion-web> : s.n., 2010.

Atlas Comparativo de la Defensa en América Latina y Caribe/Edición 2010. [En línea] [Citado el: 12 de Noviembre de 2013.]. <http://www.resdal.org/atlas/atlas10-15-cuba.pdf> (pag.187). [En línea]

Bartle, Phil. Información para la gestión y gestión de la información. [En línea] [Citado el: 15 de Noviembre de 2013.] <http://www.scn.org/mpfc/modules/mon-miss.htm>. [En l

Bootstrap, Twitter. Página Oficial. [En línea] [Citado el: 09 de 02 de 2015.] <http://bootstrapdocs.com/v2.1.1/docs/>.

Boticario, Jesus González. 2001. Capítulo 6. JavaScript". Aprender y formar en Internet. [En línea] Madrid, España : International Thomson Editors Spain Paraninfo, S. A, 2001. [Citado el: 09 de 02 de 2105.]

Castro, Elizabeth.HTML con XHTML y CSS. s.l. : Amaya.

BIBLIOGRAFÍA REFERENCIADA

- Caswell, Brian, Beale, Jay y Baker, Andrew. 2007. Snort Intrusion Detection and Prevention Toolkit. . [En línea] Kurasa : Syngress, 2007. 978-1-59749-099-3., 2007. [Citado el: 09 de 02 de 2015.]
- Centro de Energía Solar. 2012.Parque Industria PISA. Sevilla, España : s.n., 2012.
- Centro de Estudios de Ingeniería en Sistemas CEIS CUJAE. GREHU. Sistema de Gestión de Recursos Humanos. [En línea] 2003. [Citado el: 12 de 01 de 2014.]
<http://grehu.cujae.edu.cu/index.asp>. [En línea]
- Cepeda Asqui, Jessica Paulina y Tene Reino, Blanca Georgina . 2012. Investigación de la Herramienta Case para el Desarrollo del Sistema Académico Educativo en el Centro de Educación Básica “Dr. Nicanor Larrea León, Basada en la Arquitectura .Net Framework. [En línea]
- Cezanne Software HR. Cezanne. [En línea] 2009-2014. [Citado el: 18 de 12 de 2013.]
<http://cezannehr.com/es/software-rrhg/gestion-de-las-personas/>. [En línea]
- Clements, P. 1996.A Survey of Architecture Description Languages”, in Proceedings of the International Workshop on Software Specification and Design. . Alemania. : s.n., 1996.
- Comparativo de la Defensa en América Latina y Caribe/Edición 2010. [En línea] [Citado el: 12 de Noviembre de 2013.]. [ttp://www.resdal.org/atlas/atlas10-15-cuba.pdf](http://www.resdal.org/atlas/atlas10-15-cuba.pdf) (pag.182). [En línea]
- Danysoft. 2013. Embarcadero ER/Studio. [En línea] 2013. <http://www.codegear-shop.com/Embarcadero-ER/Studio/es>. [En línea]
- Davis,G. y Olsón., (1985) Management Information Systems: Conceptual foundations, Structure and Development. 2a ed.Nueva York: McGrawhill, Citado en Moreiro, G. (1998) Introducción al estudio de la información y la documentación. Medellín: Editorial Unive.
- del Castillo San Félix, Alvaro. 2000. El servidor de web Apache: Introducción práctica: Apache 1.x y 2.0 alpha. . [En línea] 2000. [Citado el: 09 de 02 de 2015.]
<http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/x31.html>..
- Del Pino Valdarrama, Santiago Luis. 2005 .Programación extrema en pocos minutos: planificando la transición.]. Cuba : Tono. Revista Técnica de la Empresa de Telecomunicaciones de Cuba, S.A., 2005. 3, págs. 41-44. 18135056. [En línea : s.n., 2005 .
- Del Pino Valdarrama, Santiago Luis. 2005. Programación extrema en pocos minutos: planificando la transición. Cuba : Tono. Revista Técnica de la Empresa de Telecomunicaciones de Cuba, S.A., 2005. 3, págs. 41-44. 18135056. [En línea]
2005. Diagrama de clases. [En línea] 2005. ; Available from: http://www-2.dc.uba.ar/materias/isoft1/is1-2005_2/apuntes/SlidesDC.pdf..
- Editorbfb. 2011. Qué es un entorno de desarrollo integrado, IDE. Programación Desarrollo. . [En línea] 2011. [Citado el: 09 de 02 de 2015.] <http://programaciondesarrollo.es/que-es-un-entorno-de-desarrollo-integrado-ide/>..
- Eguiluz, Javier.Introducción a CSS.
- . 2010.Introducción a JavaScript. 2010.
- Enciclopedia Cubana en la Red. Enciclopedia Cubana en la Red. [En línea] [Citado el: 09 de 02 de 2015.]
http://www.ecured.cu/index.php/Servidores_Web#.C2.BFQu.C3.A9_es_un_Servidor_Web.3F.

BIBLIOGRAFÍA REFERENCIADA

- Euphoriait. 2009. http://euphoriait.com/articulos/framework_web. [En línea] 10 de 06 de 2009. [Citado el: 09 de 02 de 2015.]
- Fernández, R.J., 2010. Modelo de Datos. 2010.
- . 2011. Modelo de Datos. 2011.
- Florida Solar Energy Center. 2012. Florida, Estados Unidos : s.n., 2012.
- Freixant, Victor Armando Fuentes. 2012. Geometría Solar. Arquitecturabioclimática. Universidad Autónoma Metropolitana- Azcapotzalco : Trillas México, 2012.
- Fuentes Freixant, Victor Armando . 2012. Geometria Solar. Mexico : s.n., 2012.
- Fuerzas Armadas Revolucionarias. Sitio Web de la defensa de Cuba. www.cubadefensa.cu. [En línea] [Citado el: 12 de Noviembre de 2013.] 2013. <http://www.cubadefensa.cu/far.html>. Ley de la defensa nacional, N° 75 - 21/12/1994, Art. 34. [En línea]
- Garrucha, MSc. Manuel Piloto. Sistema de Gestión del Conocimiento para una Consultoría de Inteligencia Empresarial. [En línea] [Citado el: 15 de Noviembre de 2013.] [http://www.intempres.pco.cu/Intempres2006/Intempres2006/Evaluacion de trabajos/Manuel_Pilo](http://www.intempres.pco.cu/Intempres2006/Intempres2006/Evaluacion%20de%20trabajos/Manuel_Pilo). [En línea]
- Gómez, José Jorge Márquez. 2011. Arquitectura del Software. http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rivera_l_a/capitulo2.pdf. [En línea] 2011.
- Grosso, Andrés. 2011. Patrones-Grasp. [En línea] 21 de 3 de 2011. [Citado el: 26 de 3 de 2015.] <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.
- Gutierrez, Andres Felipe. Kumbia PHP Framework, ¿Porque Programar debería ser más fácil?
- . Kumbia PHP Framework, ¿Porque Programar debería ser más fácil?
- Hamon, Hugo. 2014. Identifying Design Patterns in the Symfony Framework . Istanbul, Turkey : s.n., 2014.
- http://librosweb.es/bootstrap_3/capitulo_1.html. [En línea]
- http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_0017_04/1/TD_0017_04.pdf. [En línea]
- http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_05231_12/1/TD_05231_12.pdf. [En línea]
- Informática en Salud <http://informatica2009.sld.cu/Members/mirnacabrera/plataforma-para-la-administracion-procesamiento-y-transmision-de-la-informacion-en-el-sistema-de-salud-sisalud/> [En línea] 2013. [Citado el: 12 de 01 de 2014.]. [En línea]
- Iriondo, Graciela. 2011. Calendario Cósmico, capt. 9: El tiempo es circular. [En línea] 2011.
- Julio, G. d. J. and C. R. Cristóbal. (1999). "La Gestión del Conocimiento y el Factor Humano. Pasos para Equilibrar sus funciones.
- Kendall, Kenneth E. y Kendall, Julie E. 2005. Análisis y Diseño de Sistemas. Sexta Edición. México: Pearson Educación de México, S.A. de C.V., 2005. 970-26-0577-6. [En línea] 2005. [Citado el: 08 de 02 de 2015.]

BIBLIOGRAFÍA REFERENCIADA

- Larman, Craig. 1999.. UML y Patrones. Introducción al análisis y diseño orientado a objetos. México : PRENTICE HALL, . ISBN 970-17-0261-1 : s.n., 1999.
- Leffingwell, Dean y Widrig, Don. 2003."Using Software Engineering Techniques for Business Modeling. The Unified Modeling Language". Managing software requirements : a use case approach. United States of America : Pearson Education, Inc., 2003. 0-321-1224. 2003.
- . 2003."Using Software Engineering Techniques for Business Modeling. The Unified Modeling Language". Managing software requirements : a use case approach. United States of America : Pearson Education, Inc., 2003. 0-321-1224 : s.n., 2003.
- Lepetra., JOSE JAVIER GARCIA-BADELL. 2011.CALCULO DE LA ENERGIA SOLAR. 2011.
- Leyva Samada, Lisandra Isabel. 2009.Flujo de Investigación para la Metodología Ágil SXP. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. Cuba : s.n., 2009.
- . 2009.. Flujo de Investigación para la Metodología Ágil SXP. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. [En línea] 2009. [Citado el: 08 de 02 de 2015.] http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_24.
- Mapper, Object Relational. 2012. <http://www.doctrine-project.org/projects/orm.html>. [En línea] 2012. [Citado el: 09 de 02 de 2015.]
- Marino, ING. MARÍA CELESTE GARDEY. 2009.Introduccion a la geomertia solar. Facultad Regional Mendoza Universidad Tecnológica Nacional : s.n., 2009.
- Martínez, Ivette carolina. 2005.Ingeniería del Software I. Universidad Simón Bolivar , Venezuela : s.n., 2005.
- Maza, Miguel Ángel Sánchez. 2012.JavaScript. s.l. 2012.
- Milicias de Tropas Territoriales. Sitio Web de la defensa de Cuba. www.cubadefensa.cu. [En línea] [Citado el: 12 de Noviembre de 2013.] 2013. [http:// www.cubadefensa.cu/mtt.html](http://www.cubadefensa.cu/mtt.html). [En línea]
- Moreiro,G., (1998) Introducción al estudio de la información y la documentación. Medellín: Editorial de Antioquía.
- NYS Diseño Interactivo. . RRHH. [En línea] 2001. [Citado el: 18 de 12 de 2013.] http://www.rrhhWeb.com/includes_contenido/que_es.html. [En línea]
- Oficial, Axure RP Web. 2010. Axure RP Web Oficial. [En línea] 2010. [Citado el: 09 de 02 de 2015.] <http://www.axure.com/features>.
- OSD, Open Systems Development. <http://www.osdglobal.com/faq/desarrollo-software/comparativo-web-vs-escritorio>.
2011. Patrones Grasp. [En línea] 2011. [Citado el: 19 de 3 de 2015.] <http://www.buenastareas.com/ensayos/Patrones-Grasp/1896730.html>..
- Patrones Grasp. 2011; Available from: <http://www.buenastareas.com/ensayos/Patrones-Grasp/1896730.html>.
- Peñalver et al, G.M., A. García, S. 2010.,METODOLOGÍA ÁGIL PARA EL DESARROLLO DE SOFTWARE , in 1er Congreso Iberoamericano de Ingeniería de proyectos . Chile : s.n., 2010.

BIBLIOGRAFÍA REFERENCIADA

—. 2010.SXP, METODOLOGÍA ÁGIL PARA EL DESARROLLO DE SOFTWARE , in 1er Congreso Iberoamericano de Ingeniería de proyectos . Chile. : s.n., 2010.

Peñalver et al, G.M., A. García, S., SXP, METODOLOGÍA ÁGIL PARA EL DESARROLLO DE SOFTWARE , in 1er Congreso Iberoamericano de Ingeniería de proyectos . 2010: Chile.

Perales, Pedro E. Campos, Valladares, Ariel Guilarete y PREPARACIÓN, Angel Luis Cos Palmero. LA.

Ponjuán,G., (2000). Aplicaciones de Gestión de información en las organizaciones. El profesional de la información y su dominio de las técnicas y herramientas de la Gestión. Tesis de Doctorado. Cuba, Departamento de Bibliotecología y Ciencia de la Informa.

Pressman, Roger S. 2001. 2001.Ingeniería del Software: una tecnología estratificada. Ingeniería del Software. Un enfoque práctico. Quinta Edición. España : McGraw Hill : s.n., 2001.

Pressman, Roger S. 2001. Ingeniería del Software: una tecnología estratificada. Ingeniería del Software. Un enfoque práctico. Quinta Edición. España : McGraw Hill, 2001. [En línea]

Rodríguez Sala, Jesús Javier. 2003. Introducción a la programación: teoría y práctica. s.l. :. [En línea] Editorial Club Universitario, 2003. 9788484542742., 2003. [Citado el: 09 de 02 de 2015.]

Rodríguez Sánchez, T. 2014. Metodología de desarrollo para la Actividad productiva de la UCI. [aut. libro] T. RODRÍGUEZ SÁNCHEZ. Cuba : s.n., 2014.

ROJAS, J. A. B., EMILIO. 2007. [Disponible en:
<http://gemini.udistrital.edu.co/comunidad/grupos/arquisoft/fileadmin/Estudiantes/Pruebas/HTML%20-%20Pruebas%20de%20software/node26.html> .

Romero, Gladys Marsi Peñalver. 2008. [En línea] MA-GMPR-UR2 Metodología ágil para proyectos de software libre. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. [En línea] 2008. , 2008. [Citado el: 8 de 02 de 2015.]
http://repositorio_institucional.uci.cu/jspui/handle/id.

—. 2008. MA-GMPR-UR2 Metodología ágil para proyectos de software libre. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. [En línea] 2008. . [En línea] 2008. [Citado el: 08 de 02 de 2015.] http://repositorio_institucional.uci.cu/jspui/handle/id.

Romero, Gladys Marsi Peñalver. 2008. MA-GMPR-UR2 Metodología ágil para proyectos de software libre. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. [En línea] 2008. http://repositorio_institucional.uci.cu/jspui/handle/id. [En línea]

Rosell, Domingo Antonio Alás. 2003. Empresa de Diseño e Ingeniería CREVER Las Tunas, Cuba : s.n., 2003.

Rumbaugh, James , Jacobson, Ivar y Booch, Grady. 2007.El Lenguaje Unificado de Modelado Manual de Referencia. s.l. : Addison-Wesley Iberoameri, 2007. 9788478290871. 2007.

Sæther Bakken, Stig, Aulbach, Alexander y Schmid, Egon. 2000. PHP Manual. s.l: iUniverse Inc., 2000. Vol. 2. 978-0595132287. [En línea] 2000. [Citado el: 09 de 02 de 2015.]

Santigosa, L.M.L y L.R;. Junio 2010.Relaciones Astronómicas Sol-Tierra . Junio 2010.

Software., Arquitectura del. Available from:
http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rivera_l_a/capitulo2.pdf. [En línea]

BIBLIOGRAFÍA REFERENCIADA

- Software-talk. 2012. Netbeans vs Eclipse: An IDE comparison. Software Talk. . [En línea] 2012. [Citado el: 09 de 02 de 2015.] <http://software-talk.org/blog/2012/01/netbeans-vs-eclipse-an-ide-comparison/>..
- Sommerville, Ian. 2005. Ingeniería del software. Séptima Edición. Madrid. España: Pearson Educación. S. A., 2005. 84-7829-074-5. [En línea] 2005. [Citado el: 07 de 02 de 2015.]
- Sommerville, Ian. 2005. Ingeniería del software. Séptima Edición. Madrid. España: Pearson Educación. S. A., 2005. 84-7829-074-5. [En línea]
- Sommerville, Ian. 2005 .Ingeniería del software. Séptima Edición . Madrid. España: Pearson Educación. S. A., 2005. 84-7829-074-5. [En línea] : s.n., 2005 .
2013. Symfony en español. [En línea] 12 de marzo de 2013. [En línea] 12 de 3 de 2013. [Citado el: 23 de 3 de 2015.] <http://gitnacho.github.com/symfony-docs-es/book/doctrine.html>..
2012. Symfony. [En línea] [Citado el: 12 de noviembre de 2012.] . [En línea] 11 de 12 de 2012. [Citado el: 23 de 3 de 2015.] <http://www.symfony.com/>..
- Twig. [En línea] <http://twig.sensiolabs.org/>.
- Twig. 2012. <http://twig.sensiolabs.org/>. [En línea] 2012. [Citado el: 09 de 02 de 2015.]
- Verdecia, Pedro Manuel Alás. 2014. Cuba : s.n., 2014.
- Villaverde, P. O. F. and R. A. P. Gutiérrez. "La Gestión por Procesos y los Recursos Humanos. Presentación del software RH-CITMA " Retrieved 5 de febrero de 2008.
- Woodman L. (1985) Information management in large organizations. En: Information management from strategies to action. London: ASLIB.
2012. ZOEDDEV. [En línea] 2012. [Citado el: 12 de 2 de 2015.] <http://www.zoedev.com/javascript/la-mejor-biblioteca-de-graficos-con-html5-y-javascript-highcharts-js/>.

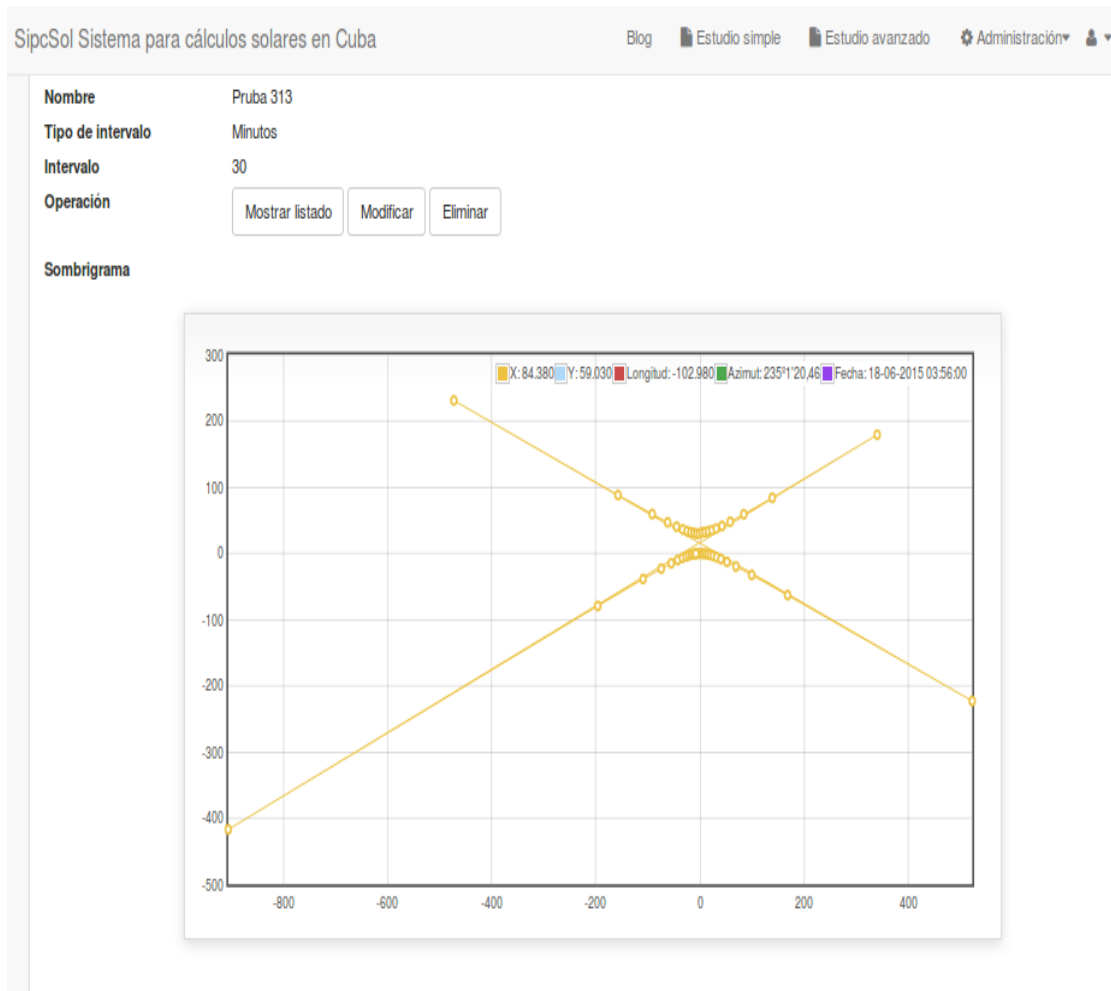
ANEXOS

Anexo 1 Síntesis del proceso de cálculo para llegar al Azimut de la Sombra y su Longitud.

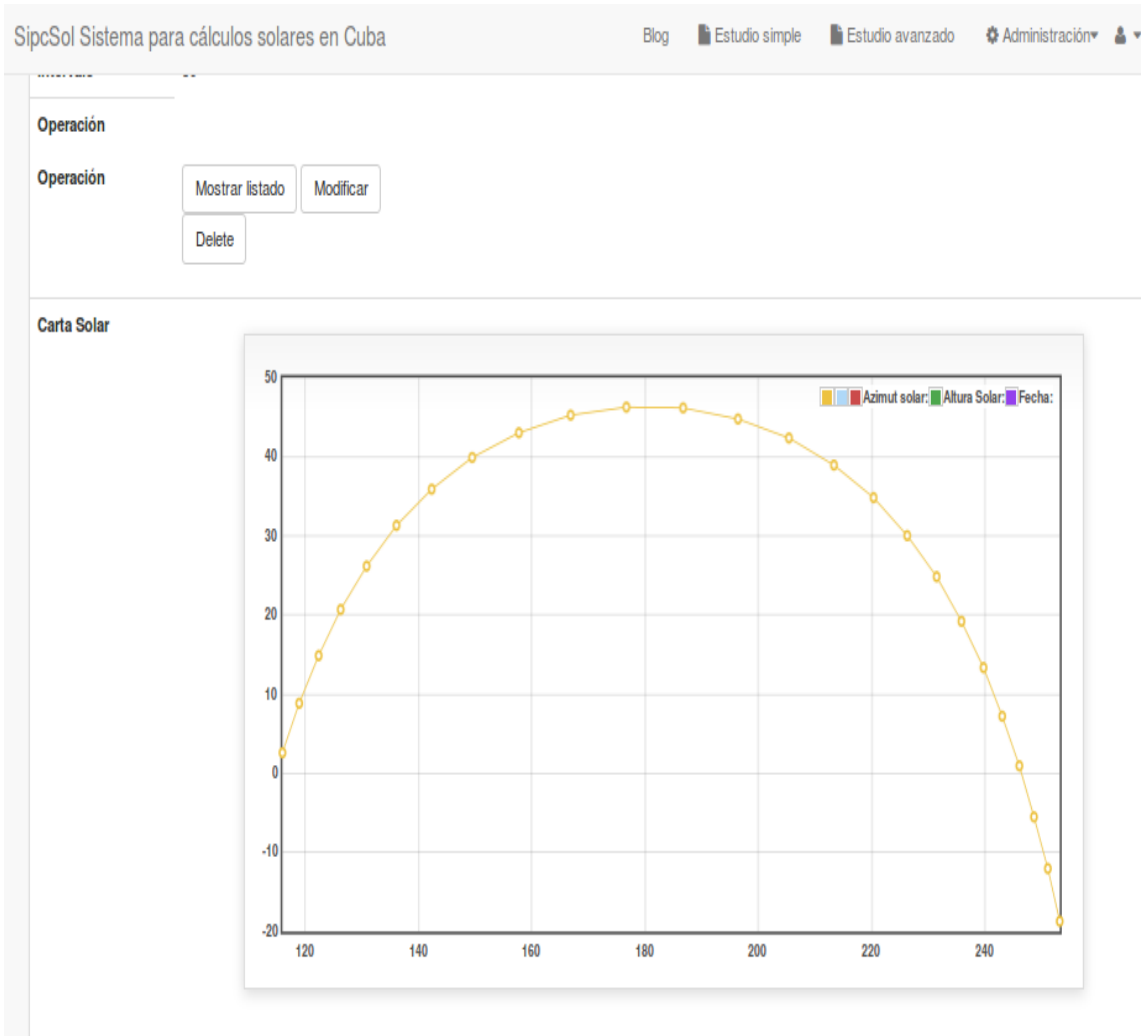
Suponiendo que se conocen los datos de latitud, longitud, fecha y hora se describen los pasos a seguir en secuencia.

- Cálculo de la fracción de día donde intervienen la fecha, la hora y la longitud geográfica.
- Cálculo de $4L$ donde interviene la longitud geográfica y el meridiano de referencia.
- Cálculo de la hora solar con la ecuación de rectificación horaria.
- Determinación del N_d donde intervienen todos los días anteriores a la fecha más la fracción de día. Para cada hora que se evalúe se calcula un N_d ya que varía la fracción de día.
- Cálculo de la ecuación de tiempo E_t donde interviene el N_d . Para cada hora que se evalúe es necesario calcular una E_t porque el N_d también varía.
- Cálculo de la declinación solar δ en el que interviene también el N_d .
- Cálculo del ángulo horario H en el que interviene la hora solar rectificada. Hay que tener en cuenta que el ángulo horario se mide a partir de la 12:00 hora solar que es cuando el Sol pasa por el meridiano.
- Cálculo de la altura solar h en el que intervienen la latitud, la declinación y el ángulo horario.
- Cálculo del acimut solar A . Ver condicionales en documento a parte.
- Cálculo del acimut de la sombra A_s a partir del acimut solar.
- Cálculo de la longitud de la sombra según la altura del punto proyectante y la tangente de la altura solar.
- Cálculo de la hora en que el Sol pasa por el meridiano despejando hora oficial cuando la hora solar es 12:00 en la ecuación de rectificación horaria, teniendo en cuenta el valor $4L$ y el valor de la ecuación de tiempo.
- Cálculo del parámetro ω según la ecuación dada y en la que intervienen la latitud y la declinación solar para ser usado en el cálculo de las horas de sol.
- Cálculo de las horas de sol H_{sol} para el día. Aquí interviene el parámetro ω .
- Dividir entre dos el Parámetro H_{sol} entre dos, es decir determinar la mitad.
- Cálculo de la hora de salida del Sol restando la mitad de las horas de sol a la hora en que el Sol pasa por el meridiano. Estas horas son en hora oficial.
- Cálculo de la hora de la puesta sumando las horas de sol a la hora de salida del Sol.

Anexo 2 Simulación de la trayectoria de la sombra, mediante un Sombrigrama generado por el sistema SIPCSOL.



Anexo 3 Simulación de la trayectoria solar, mediante una Carta Solar generado por el sistema SIPCSOL.



Anexo 4: Tabla de requisitos no Funcionales

Requisitos No Funcionales		
Tipo	No.	Descripción
Usabilidad	Usuario Final	RNF#1 El sistema será usado por el siguiente personal: <ul style="list-style-type: none"> • Ingenieros Agrónomos • Arquitectos • Astrónomos • Cualquier usuario interesado en obtener parámetros solares en Cuba.
	Aplicación Informática	RNF#2 El producto constituye una aplicación web enfocada en las necesidades de los Ingenieros agrónomos, Arquitectos, astrónomos y cualquier usuario que esté interesado en los obtener parámetros solares en Cuba.
		RNF#3 El software tendrá siempre la posibilidad de ayuda disponible para cualquier tipo de usuario, lo que le permitirá un avance considerable en la explotación de la aplicación en todas sus funcionalidades.
Finalidad	RNF#4	Este sistema está enfocado al cálculo de parámetros solares en Cuba.
Ambiente	RNF#5	El sistema debe presentar una interfaz de usuario fácil de entender y usar.
	RNF#6	El tiempo de respuesta brindado por el sistema será menor de 5 segundos, dependiendo de las prestaciones del entorno de trabajo.
	RNF#7	El sistema se desarrollará con tecnología PHP versión 5 o superior.
	RNF#8	Se empleará como Gestor de Base de Datos PostgreSQL en su versión 9.1.
	RNF#9	Se empleará como Servidor de Aplicaciones Web Apache en su versión 2.0.
	RNF#10	Como Sistema Gestor de Base de Datos se empleará PostgreSQL.
	RNF#11	Como Entorno de Desarrollo Integrado (IDE) se empleará NetBeans en su versión 7.2.
	RNF#12	Como marco de trabajo a utilizar se empleará Symfony 2 en su versión 1.4.4.1.
	RNF#13	El sistema se desarrollará en la plataforma Linux en el Sistema Operativo Ubuntu 11.10.
	RNF#14	Las computadoras de los clientes solo requerirán de un navegador Internet Explorer o Firefox en sus versiones 6 y 16.0.2, respectivamente o superior.
	RNF#15	El idioma de todas las interfaces de la aplicación será el español.
	RNF#16	El sistema mostrará el nombre del usuario que está autenticado en el sistema.
	RNF#17	El sistema contará con un menú que permite acceder a todas

		las funcionalidades para entrar los datos y procesar la información.
	RNF#18	En el sistema no existirán más de 3 interfaces para lograr una funcionalidad completa.
Confiabilidad	RNF#19	El tiempo máximo de inactividad es de 10 min y una vez que el sistema quede inactivo el usuario deberá autenticarse nuevamente.
	RNF#21	Cuando ocurra una excepción se restablecerá la información al estado anterior de forma que no queden inconsistencias en la misma.
	RNF#22	El sistema impondrá campos obligatorios para garantizar la integridad de la información que se introduce por el usuario.
	RNF#23	El sistema no permite la entrada de datos incorrectos.
	RNF#24	Ante el fallo de una funcionalidad del sistema, el resto de las funcionalidades que no dependen de esta deberán seguir funcionando.
Eficiencia	RNF#25	El sistema no excede los 3 para efectuar acciones de cálculo de los parámetros solares de un estudio simple y 30 segundos para los estudios avanzados (esta cifra no incluye los retardos por concepto de tráfico de red).
	RNF#26	El número de usuarios concurrentes que el sistema puede alojar es 100.
Restricciones del diseño	RNF#27	La iconografía utilizada será única en cada caso, permitiendo representar todos los conceptos del dominio de la aplicación con un ícono distintivo.
	RNF#28	Las etiquetas de cada funcionalidad y los campos de cada interfaz tendrán títulos asociados a su función.
	RNF#29	El sistema presentará los términos capitalizados, es decir, la primera palabra tendrá su primera letra en mayúsculas.
Interfaz	RNF#30	El sistema presentará una interfaz legible y simple de usar
	RNF#31	El sistema deberá ser independiente de la plataforma.
	RNF#32	La información de los gráficos no se solapa.
Requisitos de licencia	RNF#33	Para el desarrollo del software se emplearán las siguientes licencias: Licencia PostgreSQL Licencie para el Gestor de Base de Datos Postgres, licencia Apache 2.0 para el Servidor Web Apache. Licencia GPL para el Sistema Operativo Ubuntu 11.10.
Estándares Aplicables	RNF#34	Se empleará el estilo de codificación <i>CamelCase</i> , específicamente la variante <i>lowerCamelCase</i> .
	RNF#35	Se empleará el estándar para aplicaciones web W3C (<i>World Wide Web Consortium</i> , por sus siglas en inglés).
Seguridad	RNF#36	Que las informaciones relacionadas con los parámetros solares sean accesibles por los usuarios según su rol.
	RNF#37	El sistema brindará la posibilidad de establecer permisos sobre acciones, garantizando que solo acceda a la información quien esté autorizado.
	RNF#38	El sistema mostrará las funcionalidades de acuerdo a quien esté autenticado en el mismo.
	RNF#39	El sistema deberá garantizar la protección ante acciones no autorizadas.

Anexo 5: Acta de Aceptación.



CENTRO DE INFORMATIZACIÓN DE ENTIDADES
DEPARTAMENTO DE DESARROLLO DE COMPONENTES

FECHA 19/ 06/2015

A quien pueda interesar:

Por este medio se hace constar que la solución SIPCSOL Sistema para Cálculos Solares en Cuba del autor Aniel Perez Orozco fue sometida a una revisión técnica en la cual se detectaron 38 no conformidades que fueron resueltas quedando esta solución estable y lista para su posterior uso.

Como parte del desarrollo de la solución se elaboraron y entregaron los siguientes artefactos:

1. Modelo conceptual (1).
2. Historias de usuarios (7).
3. Modelo de diseño (4).
4. Diseños de casos de prueba (20).
5. Modelo de datos (1)
6. Código fuente.

Para que así conste firman a continuación los miembros del equipo que realizó la revisión, el autor y los tutores del trabajo.

Dado a los 19 días del mes de junio de 2015.

Nombre y apellidos	Firma
Revisores: Ing. Claudia Bravo Batista	
Autor (es): Aniel Perez Orozco	
Tutor: Ing. Pedro Manuel Alás Verdecia	

J' Departamento de Desarrollo de Componentes
René R. Bauta Camejo