

Universidad de las Ciencias Informáticas

Facultad 6



Título: Módulo de análisis estadístico en su versión 2.0 para el RServer.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS.

Autores:

Julio Germán Hernández Gómez

Sandy Enrique Noa Rodríguez

Tutores:

Ing. Anaibis Álvarez Morales

Ing. Rachid Alí Grave de Peralta

La Habana, junio de 2013
“Año 55 de la Revolución”

DECLARACIÓN DE AUTORÍA

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Julio G. Hernández Gómez

Firma del autor.

Rachid Alí Grave de Peralta

Firma del tutor.

Sandy E. Noa Rodríguez

Firma del autor.

Anaibis Álvarez Morales

Firma del tutor.

DATOS DE CONTACTO

DATOS DE CONTACTO

Ing. Rachid Alí Grave de Peralta: Profesor graduado en Ingeniería de las Ciencias Informáticas en el año 2009, en la Universidad de las Ciencias Informáticas.

Se desempeña como profesor en el centro DATEC.

Contactar al correo rali@uci.cu

Ing. Anaibis Álvarez Morales: Profesora graduada en Ingeniería de las Ciencias Informáticas en el año 2011, en la Universidad de las Ciencias Informáticas.

Se desempeña como profesora en el centro DATEC.

Contactar al correo aamorales@uci.cu

AGRADECIMIENTOS

A los muchachos, Darlon, Dago, Liandry y Nesty por estar presentes cuando los necesitamos y hacer más soportable toda esta locura. Por ser nuestros mejores amigos y por sabernos entender siempre.

Sandy E.

A mis tutores Rachid y Anaibis por toda la atención prestada y por toda la ayuda ofrecida.

A mi madre, por sus consejos, por su preocupación, por sufrir los malos ratos conmigo, por sus rezos a Dios, por guiarme y por ser la mejor madre del mundo.

A mi abuelo, por incentivarme a seguir y por ser como un padre para mí.

A mi chica Yanet por su confianza, por su paciencia y espera, por comprenderme y por cuidar de la princesita hermosa que tenemos.

A mis tíos por apoyarme, por sentirse orgullosos de mí y por saber aconsejarme.

A mi niña, por ser simplemente mi razón de existir.

A mi amigo Frank por estar conmigo en el momento más difícil de mi vida y por todo lo que hizo por mí.

A mi amigo Carlos A. por ser un amigo un maestro y un hermano para mí.

Julio G.

A mis padres el sacrificio; gracias mami por preocuparte y en la misma medida ocuparte de mis cosas, por estar siempre a mi lado apoyándome.

A papá por sus consejos y su manera de motivarme.

A mi tía Juana por su acogida y por el apoyo brindado.

A mi tía Rosita por su afecto y por dejarse querer como a una madre.

A tata, la hermana que mis tíos me dieron, por todo lo que ha hecho por mí y por representar con su optimismo, sus sueños y ganas de vivir, un ejemplo a seguir.

A Yaily por escucharme, aguantarme, darme fuerzas para seguir, por lo que me ha ayudado desde que nos conocimos y sobre todo por su paciencia.

DEDICATORIA

Sandy E.

A la persona que más soñó verme graduar de Ingeniero en Ciencias Informáticas. A mi querida y adorada abuelita Gloria, por ser mi estrella, mi inspiración y por ser la persona que me ha transmitido las fuerzas desde el cielo para terminar.

A mi madre, pues sé que para ella todo esto es tan importante como para mí, por cuidarme exageradamente y por ser, quizás, quien más disfrute este momento.

Julio G.

A mis padres por darme vida y entregarme la suya.

RESUMEN

La estadística es una ciencia vinculada a la investigación científica. Los científicos que hacen uso de esta ciencia, comúnmente emplean programas de computación para realizar análisis estadísticos. Estos programas son privados y sus licencias suelen ser de coste elevado. Una alternativa a esta situación es hacer uso de R, software libre de excelente calidad y respaldado por una gran comunidad que potencia su escalabilidad. El RServer es un servidor que a través de un módulo de análisis estadístico llamado STATS, ofrece un conjunto de análisis propios de R; estos son fácilmente integrables en proyectos web escritos específicamente en el lenguaje de programación PHP. La presente investigación describe los aspectos teóricos, conceptuales y prácticos relacionados con el proceso de desarrollo del módulo de análisis estadístico en su versión 2.0 para el RServer y se centra en incrementar las funcionalidades de dicho módulo. Se realiza el análisis de la metodología, las herramientas y las tecnologías a emplear en la solución, así como de los sistemas estadísticos más renombrados a nivel mundial. Se ejecutan las pruebas de software pertinentes para determinar la calidad y el rendimiento del producto y finalmente se obtiene una capa de abstracción que permite, a otros proyectos, utilizar las distribuciones de Beta, de Cauchy, de F, las distribuciones Geométrica, Gamma, Uniforme y además las Correlaciones Canónicas, haciendo más fácil la obtención de los resultados y asegurando la exactitud y precisión en los mismos.

Palabras claves: estadística, módulo de análisis estadístico, RServer.

ÍNDICE DE CONTENIDO

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA SOBRE EL MÓDULO STATS DEL RSERVER.....	6
INTRODUCCIÓN	6
1.1 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA	6
1.1.1 Estadística	6
1.1.2 Módulo	7
1.1.3 Análisis estadístico de datos	7
1.1.4 Sistema de análisis estadístico	7
1.2 SISTEMAS DE ANÁLISIS ESTADÍSTICOS.....	8
1.2.1 SPSS (Paquete Estadístico para las Ciencias Sociales)	8
1.2.2 SAS (Sistema de análisis estadístico)	8
1.2.3 R (Paquete estadístico y lenguaje para matemáticas).....	9
1.2.4 Comparación general.....	9
1.3 MÉTODOS DE ANÁLISIS ESTADÍSTICOS	11
1.3.1 Distribución Beta	11
1.3.2 Distribución de Cauchy	11
1.3.3 Distribución F	12
1.3.4 Distribución Gamma.....	12
1.3.5 Distribución Geométrica	13
1.3.6 Distribución Uniforme.....	13
1.3.7 Correlaciones Canónicas	14
1.4 METODOLOGÍA, TECNOLOGÍAS Y HERRAMIENTAS EMPLEADAS EN LA SOLUCIÓN.....	15
1.4.1 Metodología de desarrollo de software.....	15
1.4.2 Herramientas CASE.....	16
1.4.3 Lenguaje de programación.....	17
1.4.4 Marco de trabajo	18
1.4.5 Ambiente de desarrollo	19
1.4.6 Servidor Web	19
1.4.7 Servicios web.....	20
CONCLUSIONES DEL CAPÍTULO.....	21

CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL MÓDULO STATS	22
INTRODUCCIÓN	22
2.1 MODELO DE DOMINIO	22
2.2 REQUISITOS DE SOFTWARE	24
2.2.1 Requisitos funcionales	24
2.2.2 Requisitos no funcionales	33
2.3 DIAGRAMA DE CASOS DE USO DEL SISTEMA	33
2.4 DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA	34
2.5 ARQUITECTURA ORIENTADA A SERVICIOS (SOA)	46
2.6 PATRONES DE DISEÑO Y DE ARQUITECTURA DE SOFTWARE	46
2.7 DIAGRAMA DE CLASES DEL DISEÑO	47
2.8 DIAGRAMA DE SECUENCIA	49
2.9 DIAGRAMA DE DESPLIEGUE	50
CONCLUSIONES DEL CAPÍTULO	51
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA DEL MÓDULO	52
INTRODUCCIÓN	52
3.1 MODELO DE IMPLEMENTACIÓN	52
3.1.1 Diagrama de componentes	52
3.2 ESTÁNDARES DE CODIFICACIÓN	53
3.3 IMPLEMENTACIONES SIGNIFICATIVAS	54
3.4 PRUEBAS DE SOFTWARE	56
3.4.1 Tipos de prueba	56
3.4.2 Pruebas de Caja Blanca	56
3.4.3 Pruebas de Caja Negra	58
3.4.4 Pruebas de rendimiento (Carga y Estrés)	61
CONCLUSIONES DEL CAPÍTULO	62
CONCLUSIONES	63
RECOMENDACIONES	64
REFERENCIAS	65
BIBLIOGRAFÍA	69
ANEXOS	72

INTRODUCCIÓN

Las ciencias de la computación constituyen una de las áreas de conocimiento que mayor evolución ha experimentado en los últimos años, esto trae consigo una revolución en las Tecnologías de la Información y las Comunicaciones (TIC) mundiales. Respecto a esto Fidel Castro expresó: "No hay más que asomarse a las puertas de la tecnología y las ciencias contemporáneas, para preguntarse si es posible vivir y conocer ese mundo del futuro sin un enorme caudal de preparación de conocimientos". (1).

La sociedad cubana tiene la urgente necesidad de ponerse a tono con el desarrollo actual de las TIC; estas agrupan los elementos y las técnicas usadas en el tratamiento y la transmisión de la información, resaltando potencialmente el uso de la informática como elemento fundamental para el procesamiento automático de los datos. En la actualidad, las ciencias informáticas proporcionan un entorno que asegura la integridad y la fiabilidad necesarias para la realización de procedimientos automatizados, esto ha conducido a que el desarrollo de productos informáticos se convierta en una de las principales plataformas de conocimiento, razón por la cual se busca la perfección en la construcción de estos dentro de la estadística como ciencia, permitiendo procesar grandes volúmenes de datos contables y garantizar que sea más ágil y confiable la realización de este proceso.

La estadística es la ciencia que estudia y recolecta los datos para explicar las condiciones regulares o irregulares de algún fenómeno o estudio aplicado, trata la clasificación y presentación de los hechos sujetos a una apreciación numérica como base de la explicación, descripción y comparación de los fenómenos. Murria R. Spiegel, (1991) define que: "La estadística estudia los métodos científicos para recoger, organizar, resumir y analizar datos, así como para sacar conclusiones válidas y tomar decisiones razonables basadas en tal análisis".

La estadística apoya la investigación y otras disciplinas con el fin de verificar o formular nuevas hipótesis. Es por ello que los procesos comerciales, administrativos y tecnológicos necesitan estos métodos en la toma de decisiones y en la predicción de valores futuros. Permite la obtención de indicadores, pasando por un proceso de establecimiento y verificación de hipótesis, trabajando en base a modelos (expresiones matemáticas) que potencialmente siguen las variables en estudio y que se comprueban utilizando los datos recogidos. También abarca la presentación y caracterización de la información para ayudar en el análisis e interpretación de datos (análisis estadístico). (2)

El análisis estadístico constituye un factor clave para lograr el éxito de muchas instituciones en el mundo. Gracias a la fusión de la informática con la estadística se han creado herramientas computacionales dedicadas a este fin.

La Universidad de las Ciencias Informáticas (UCI) es una universidad productiva que pretende informatizar el país y desarrollar la industria de software para contribuir al desarrollo económico del mismo. Su objetivo es producir software y servicios informáticos a partir de la vinculación estudio – trabajo como modelo de formación y forjar profesionales comprometidos con su Patria, altamente calificados en la rama de la informática. Dentro de la UCI existen varios centros productivos, uno de ellos es el Centro de Tecnologías de Gestión de Datos (DATEC).

DATEC es un centro especializado en Tecnologías de Gestión de Datos y Bioinformática, reconocido por sus soluciones y productos, con una activa colaboración con empresas, centros de investigación y desarrollo de alto nivel, así como con comunidades internacionales afines. Tiene la misión de crear bienes y servicios informáticos relacionados con la gestión de los datos, área del conocimiento que agrupa tanto a los sistemas de información, como a los denominados sistemas de inteligencia empresarial o de negocios, cuyo propósito fundamental es apoyar el proceso de toma de decisiones.

DATEC trabaja directamente con entidades como la Oficina Nacional de Estadística e Información (ONEI), institución que rige los principales procesos estadísticos en Cuba, y que para esto requiere el uso de sistemas informáticos. La ONEI emplea sistemas como el Paquete Estadístico para las Ciencias Sociales (SPSS), pero su licencia suele ser de coste elevado y su estructura no permite una fácil manipulación. El uso de dicha aplicación se debe a que el Centro no provee una herramienta completa para la realización de análisis estadístico, con el fin de suplir esta carencia hoy se encuentra en desarrollo el RServer.

El RServer es un servidor que ofrece un conjunto de análisis estadísticos propios del lenguaje R. Tiene como objetivo brindar una capa de abstracción que permita a otras aplicaciones informáticas, interactuar con las funcionalidades del paquete estadístico de R a través de servicios web. Debido a que está escrito en PHP sobre la arquitectura del marco de trabajo Symfony2, también puede ser usado como Interfaz de Programación de Aplicaciones (API) en software escritos sobre arquitecturas similares. RServer está estructurado por 2 módulos, TM para la minería de textos y STATS para el análisis estadístico de los datos.

Después de realizado el estudio del módulo de análisis estadístico del RServer, se ha determinado que el conjunto de funciones que proporciona es insuficiente, pues no se incluyen muchos de los análisis considerados básicos dentro de la estadística.

En su versión 1.0 el módulo de análisis estadístico del RServer permite realizar un conjunto de funciones estadísticas tales como: las distribuciones de Poisson, distribuciones de T-Student, distribuciones de Chi-Cuadrado, el análisis estadístico de datos de varianza para variables dependientes continuas y otras. Sin embargo, no permite calcular las variaciones de porcentajes que representan algún fenómeno, por ejemplo el tiempo diario que un estudiante dedica frente a una computadora. No cuenta además, con funciones que permitan detectar el tiempo que una máquina permanece encendida, no permite detectar la existencia de diferencias significativas entre muestras, esencial en todos aquellos casos en los que se quiere investigar la relevancia de un factor en el desarrollo y naturaleza de una característica. No implementa funciones que se apliquen a fenómenos de máxima incertidumbre, cuando no hay información que permita otorgar más probabilidad a unas zonas que a otras y carece de funciones que faciliten encontrar la combinación lineal de un juego de variables que tengan un alto grado de correlación con cualquier combinación lineal de un segundo juego de variables.

Por lo anteriormente expuesto se puede definir el siguiente **problema de la investigación**: ¿Cómo contribuir al incremento de las funciones que el RServer ofrece para la realización de análisis estadístico en el Centro de Tecnologías de Gestión de Datos (DATEC)?

Se define como **objeto de estudio de la investigación**: los sistemas de análisis estadístico de datos, enmarcado en **campo de acción**: los métodos de análisis estadístico de datos.

Planteando como **objetivo general**: Incrementar las funciones del módulo de análisis estadístico en su versión 2.0 para el RServer.

Objetivos Específicos:

- ✓ Realizar un estudio que permita caracterizar los sistemas de análisis estadísticos actuales y definir los métodos que serán incorporados al módulo de análisis estadístico en su versión 2.0.
- ✓ Fundamentar la metodología, las herramientas y las tecnologías empleadas en el desarrollo de la versión 2.0 del módulo de análisis estadístico para el RServer.

- ✓ Definir el análisis y diseño del módulo de análisis estadístico en su versión 2.0 para el RServer.
- ✓ Implementar las nuevas funcionalidades del módulo de análisis estadístico en su versión 2.0 para el RServer.
- ✓ Realizar pruebas al módulo de análisis estadístico en su versión 2.0 para el RServer.

Las **tareas de la investigación** que se proponen para dar cumplimiento a los objetivos son:

- ✓ Realización del estudio bibliográfico de los sistemas de análisis estadísticos existentes en el mundo para identificar sus principales características.
- ✓ Realización del estudio de los tipos de análisis estadísticos definidos en el estándar de R para entender su funcionamiento, sintaxis y aplicación práctica.
- ✓ Definición de los nuevos métodos del módulo de análisis estadístico en su versión 2.0 para incrementar las funcionalidades del RServer.
- ✓ Fundamentación de la selección de la metodología, las herramientas y las tecnologías a utilizar en el desarrollo del módulo de análisis estadístico en su versión 2.0 para el RServer.
- ✓ Especificación de los requisitos funcionales y no funcionales del sistema para establecer lo que el software debe hacer y bajo qué circunstancias debe hacerlo.
- ✓ Confección de los diagramas correspondientes al diseño para traducir las necesidades del cliente en modelos que representan la arquitectura del sistema.
- ✓ Definición de los patrones de arquitectura y diseño para establecer la estructura del módulo de análisis estadístico para el RServer.
- ✓ Implementación de la versión 2.0 del módulo de análisis estadístico para RServer, para dar cumplimiento a los requisitos definidos.
- ✓ Definición de los tipos de pruebas para determinar el comportamiento del software bajo determinadas circunstancias.
- ✓ Realización de las pruebas de caja blanca y de caja negra para determinar la calidad del sistema.

El documento está estructurado por tres capítulos:

Capítulo 1: Fundamentación Teórica sobre el Módulo STATS del RServer.

En el capítulo quedan sustentadas las bases teóricas de la investigación. Se exponen los conceptos fundamentales asociados al objeto de estudio del trabajo, se especifica la metodología, las herramientas y las tecnologías utilizadas para dar solución al problema en cuestión. Además se exponen las fórmulas y aplicaciones matemáticas de los nuevos métodos estadísticos a implementar en el módulo de análisis estadístico en su versión 2.0 para el RServer.

Capítulo 2: Análisis y Diseño de la Solución.

En este capítulo se describe el análisis de la solución propuesta utilizando UML¹ como lenguaje de modelado y Open UP como metodología de desarrollo de software. Incluye el modelo de dominio y la especificación de los requisitos funcionales y no funcionales que guían el desarrollo del software. Se determinan el actor y el caso de uso, los cuales quedan representados en el diagrama de casos de uso del sistema. Se modela el diagrama de clases del diseño y el diagrama de despliegue. Se define la arquitectura y los patrones de diseño que se van a emplear en la implementación del módulo, obteniéndose como resultado la estructura arquitectónica para el módulo de análisis estadístico del RServer.

Capítulo 3: Implementación y Prueba del Módulo STATS.

En el capítulo se generan los artefactos correspondientes a esta fase, donde se modela y describe el diagrama de componentes y se define un estándar de codificación para la implementación. Se definen las pruebas, especificando las técnicas, tipos y métodos de pruebas que se emplearán en la revisión de la aplicación, se diseñan y ejecutan los casos de prueba y se visualizan fragmentos de códigos de la aplicación.

Para finalizar, se presentan las conclusiones, las recomendaciones, la bibliografía, los anexos y un glosario de términos para brindar un mejor entendimiento de lo expuesto en todo el documento.

¹ **UML:** el lenguaje para modelamiento unificado es un lenguaje para la especificación, visualización, construcción y documentación de los artefactos de un proceso de sistema.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA SOBRE EL MÓDULO STATS DEL RSERVER

Introducción

En el presente capítulo quedan sustentadas las bases teóricas de la investigación. Se exponen los conceptos fundamentales asociados al objeto de estudio del trabajo. Se realiza el estudio del estado del arte, que recoge los diferentes sistemas de análisis estadísticos y evalúa sus características. Además se especifica la metodología, las herramientas y las tecnologías utilizadas para dar solución al problema en cuestión. A continuación se exponen conceptos que se consideran determinantes para el correcto entendimiento del problema que se trata y de la solución que se propone.

1.1 Conceptos asociados al dominio del problema

Es importante que todo profesional que utilice la estadística como herramienta auxiliar de trabajo, posea un mínimo de conocimientos y domine los conceptos que facilitarán el buen desarrollo de esta actividad.

1.1.1 Estadística

La estadística es la ciencia que tiene como objetivo reunir la información cuantitativa concerniente a individuos, grupos y series de hechos; y deducir de ello, basado en el análisis de estos datos, significados precisos o previsiones para el futuro.

Algunos autores tienen definiciones de la estadística semejantes a la anterior, aunque otros difieren un poco. Para José Enrique Chacón, la estadística se define como “la ciencia que tiene por objeto el estudio cuantitativo de los colectivos”. (3)

Harald Cramér se limita a establecer que “el principal objeto de la teoría estadística consiste en la investigación de la posibilidad de obtener inferencias válidas a partir de los datos estadísticos, y en la construcción de métodos para realizar dichas inferencias”. (4)

La más aceptada, sin embargo, es la de Óscar Vázquez Mínguez, que define la estadística como “La ciencia que tiene por objeto aplicar las leyes de la cantidad a los hechos sociales para medir su intensidad, deducir las leyes que los rigen y hacer su predicción próxima”. (5)

1.1.2 Módulo

Existen varios conceptos referentes al término módulo, aunque los autores de la presente investigación se adscriben a las definiciones dadas por Lastre, debido a que son las que mejor se ajustan al trabajo realizado.

Lastre define módulo como "...componente autocontrolado de un sistema, el cual posee una interfaz bien definida hacia otros componentes; algo es modular si es construido de manera tal que facilite su ensamblaje, acomodamiento flexible y reparación de sus componentes." (6)

También define que "en programación un módulo es una parte de un programa de ordenador". De las varias tareas que debe realizar un programa para cumplir con su función u objetivo, un módulo realiza una o varias tareas.

1.1.3 Análisis estadístico de datos

Una peculiaridad fundamental del trabajo estadístico práctico es el análisis de conjuntos de datos numéricos. Cualquier conclusión a la cual se llegue mediante una técnica o procedimiento estadístico lleva implícita necesariamente el análisis de cantidades o características cuantitativas.

Se define análisis estadístico como "conjunto de métodos, técnicas y procedimientos para el manejo de datos, su ordenación, presentación, descripción, análisis e interpretación, que contribuyen al estudio científico de los problemas planteados en distintos sectores y a la toma de decisiones". (7)

1.1.4 Sistema de análisis estadístico

Un sistema de análisis estadístico es un conjunto de programas informáticos específicamente diseñados para el análisis estadístico de datos, con el objetivo de resolver problemas de estadística descriptiva, inferencial o ambos. Es una variedad de programas y subprogramas conectados de modo que funcionan de manera conjunta; para pasar de uno a otro no se necesita salir del programa y volver a él. Un sistema de análisis estadístico permite aplicar a un mismo fichero de datos un conjunto ilimitado de procedimientos estadísticos de manera sincronizada, sin salir del programa.

1.2 Sistemas de análisis estadísticos

En la actualidad todo se enfoca en torno al software, es por lo que se han diseñado programas informáticos que son embebidos por diferentes sectores de la ciencia, algunos de estos son SPSS, SAS, R, Minitab, Statgraphics STATISTICA, Redatam y S-PLUS aunque solo algunos de ellos serán objeto de estudio en este trabajo.

1.2.1 SPSS (Paquete Estadístico para las Ciencias Sociales)

Es un software estadístico de análisis de datos lanzado al mercado en 1968. Originalmente se desarrolló para las ciencias sociales, por lo que ofrece un uso sencillo de las opciones, acceso rápido a datos y procedimientos, generación de salidas y gráficos. Contiene programas capaces de realizar desde un simple análisis descriptivo, hasta diferentes tipos de análisis multivariantes de datos. Dispone de un amplio conjunto de métodos estadísticos (series temporales, regresión logística y análisis de supervivencia). (8)

A partir del análisis de la herramienta se evidenciaron desventajas que imposibilitaron la utilización de este software. SPSS es un programa con una interfaz gráfica de usuario amigable, pero sólo a través de esta se accede a sus opciones por lo que solo se pueden abrir los datos y ejecutar cálculos mediante el uso de los botones de la interfaz gráfica, lo que puede considerarse una deficiencia si se necesita acceder, visualizar y aprender el código de sus funcionalidades. Además es un software de licencia propietaria.

1.2.2 SAS (Sistema de análisis estadístico)

Ha sido por largos años el software más utilizado en la comunidad estadística y, por lo tanto, también se ha propagado su uso entre investigadores de diferentes disciplinas. Es un sistema integrado de programas independientes, conjunto de programas de ordenador agrupados en módulos que trabajan simultáneamente. SAS está orientado fundamentalmente al análisis estadístico. Aunque permite también la grabación y recuperación de datos, el control de calidad, la investigación operativa y la econometría. (9)

Luego de la investigación realizada a este sistema se pudo constatar que su utilización puede tornarse un poco engorrosa, SAS, a diferencia de SPSS, es un programa que requiere el ingreso de comandos para ejecutar gran parte de sus rutinas y opciones. Por lo tanto, el usuario necesita del conocimiento de la sintaxis antes de su uso. El aprendizaje de sus comandos provocaría una demora potencial en la

realización de cualquier software que requiera funcionalidades estadísticas, además no es un producto gratis y su código fuente no se encuentra disponible.

1.2.3 R (Paquete estadístico y lenguaje para matemáticas)

R es un programa estadístico que proporciona un entorno en el que se puede realizar el análisis estadístico, es un sistema integrado para manejo de datos, simulaciones, cálculos y realización de gráficos. Además es un lenguaje de programación completo y orientado a objetos de uso libre, de distribución gratuita y de código abierto. R también es un programa basado sobre comandos, en el que se puede acceder a todos los procedimientos y opciones a través de sintaxis computacional. Fue oficialmente presentado en 1997 y es un software libre que se rige por la licencia general pública (GPL) de la fundación de software (GNU)². R ofrece una gran variedad de estadísticas (modelos lineales y no lineales, clásicas pruebas estadísticas, análisis de series de tiempo y clasificación), además de poseer técnicas gráficas y ser altamente extensible. También combina en un solo programa diferentes funciones estadísticas para realizar análisis más complejos. Originalmente fue concebido para el sistema operativo Unix, aunque puede utilizarse actualmente en versiones para Windows, Macintosh y Linux. (10)

Es importante especificar que tanto SPSS como SAS son programas comerciales y, por lo tanto, tienen una orientación y administración diferentes a las de R, además se enfocan en aquellos mercados y usuarios que les proporcionan los mayores beneficios.

1.2.4 Comparación general

A continuación se establece una comparación entre los programas estadísticos más utilizados en la actualidad, sobre la base de las características de cada uno de ellos.

² **GNU:** es un acrónimo recursivo que significa GNU No es Unix. El proyecto GNU fue iniciado por Richard Stallman con el objetivo de crear un sistema operativo completamente libre: el sistema GNU.

Aspecto	Programa estadístico		
	SPSS	SAS	R
Amigabilidad con el usuario	Excelente	Baja-Regular	Baja-Regular
Manipulación de datos	Baja	Buena	Buena
Calidad de gráficos	Regular	Buena-Excelente	Excelente
Control de procesos	Baja	Excelente	Excelente
Costo	U\$S 1500	U\$S 7200	Gratis
Código fuente disponible	No	No	Sí
Variedad análisis estadísticos	Buena	Buena-Excelente	Excelente
Documentación	Excelente	Buena	Buena-Excelente
Soporte técnico	Bueno	Bueno	Bajo
Sistema operativo	Windows®	Windows® Macintosh® Linux	Windows® Macintosh® Linux

Tabla 1. Comparación de aspectos generales entre los programas estadísticos SPSS, SAS y R

En este contexto, se presenta dicha comparación sobre la base de una serie de aspectos que demuestran que R es el software indicado para la realización del trabajo. Aunque R requiere conocer la sintaxis y/o los comandos antes de ejecutar un procedimiento, posee una fácil manipulación de datos ya que muestra los datos o una porción de ellos según se requiera, también es posible mostrar y editar los datos en una ventana al igual que SPSS y SAS.

En cuanto a calidad de gráficos SPSS ofrece una serie de gráficos que aunque pueden modificarse son difíciles de personalizar y aunque tanto SAS como R permiten el diseño personalizado de gráficos, SAS requiere del uso de diferentes rutinas, mientras que las sintaxis de R son más sencillas. R también ofrece una amplia gama de formatos en los cuales los gráficos pueden ser exportados, sin necesidad de mayor sintaxis.

R es totalmente gratuito y su código fuente se encuentra disponible en Internet, al igual que cada versión nueva. La instalación típica de R consiste de una serie de paquetes, y cada paquete nuevo que ha sido desarrollado y publicado también en Internet, desde donde se puede obtener y agregar al sistema. A diferencia de SPSS y de SAS, la distribución de R es actualizable y se le pueden incorporar paquetes adicionales por separado.

Por su parte, la variedad de análisis estadísticos en R resulta excelente pues la renovación e implementación de nuevos procedimientos en R es relativamente rápida, debido a la aparición de nuevos

procedimientos y/o paquetes en desarrollo, los cuales pueden ser instalados directamente. En cambio, SAS y SPSS demorarán años en implementar nuevos procedimientos, requiriendo, necesariamente una nueva versión del software. (11)

1.3 Métodos de análisis estadísticos

A continuación aparecen las fórmulas matemáticas de cada una de las distribuciones que se emplean para la implementación del módulo de análisis estadístico en su versión 2.0 para el RServer.

1.3.1 Distribución Beta

Aplicación: La distribución de probabilidad Beta es una función de densidad con dos parámetros definida en el intervalo cerrado $0 \leq x \leq 1$. Se utiliza frecuentemente como modelo para fracciones, tal como la proporción de impurezas en un producto químico o la fracción de tiempo que una máquina está en reparación.

Esta distribución se ha empleado de manera extensa en la representación de magnitudes físicas, cuyos resultados están acotados por algún intervalo finito; su utilidad también se evidencia al calcular límites de tolerancia, sin necesidad de recurrir a la distribución normal. (12) A continuación se presenta la función Beta que se define de la siguiente manera:

$$B(a, \beta) = \int_0^1 x^{a-1} (1-x)^{\beta-1} dx$$

Una utilidad importante de la función Beta es:

$$B(a, \beta) = \frac{\Gamma(a) \Gamma(\beta)}{\Gamma(a + \beta)}$$

Donde $\Gamma(X)$ es la función gama que se describe más adelante.

1.3.2 Distribución de Cauchy

Aplicación: La distribución de Cauchy es un ejemplo de una distribución que no tiene valor esperado, varianza o momentos definidos. La teoría mecánica y eléctrica y los problemas de medición y calibración son algunas de las áreas donde se utiliza esta distribución.

Entonces sea X una variable aleatoria con distribución de Cauchy con parámetros $\alpha = 0$ y $\beta = 1$:

$$f(x) = \frac{1}{\pi} \frac{1}{1+x^2}$$

Su función de densidad de probabilidades está dada por la ecuación anterior. (13)

1.3.3 Distribución F

Aplicación: Es una distribución de probabilidad de gran aplicación en la inferencia estadística, fundamentalmente cuando se trata de contrastar la igualdad de varianzas de dos poblaciones normales, técnica que permite detectar la existencia o inexistencia de diferencias significativas entre muestras diferentes y que es, por tanto esencial en todos aquellos casos en los que se quiere investigar la relevancia de un factor en el desarrollo y naturaleza de una característica.

Siendo x_1 y x_2 variables aleatorias ji-cuadrado con v_1 y v_2 grados de libertad, entonces si x_1 y x_2 son independientes:

$$F = \frac{x_1/v_1}{x_2/v_2}$$

Se dice que tiene una distribución F con v_1 grados de libertad del numerador y v_2 grados de libertad del denominador. (14)

1.3.4 Distribución Gamma

Aplicación: Los tiempos que tardan en revisar un motor de un automóvil ó avión tienen una distribución de frecuencias sesgadas. Las poblaciones asociadas a estas variables aleatorias frecuentemente tienen distribuciones que se pueden modelar adecuadamente por la función de densidad tipo Gamma. A fin de poder caracterizar la distribución gama, es necesario hacer una presentación de una de las funciones más importantes tanto en la teoría como en la práctica.

Definición: la función gama notada Γ , se define de la siguiente forma:

$$\Gamma(p) = \int_0^{\infty} x^{p-1} e^{-x} dx$$

Definida para $p > 0$

Sea X una variable aleatoria continua, que recorre o toma valores no negativos, se dice que X tiene una distribución de probabilidad gama si su función de probabilidad es:

$$f(x) = \frac{1}{\Gamma(a)\beta^a} x^{a-1} e^{(-\frac{x}{\beta})}$$

$$\text{Si } x > 0 \text{ a, } \beta > 0$$

El valor esperado y la varianza de la distribución gama están dados por:

$$E(X) = a\beta$$

$$VAR(X) = a\beta^2$$

La variación de los parámetros a y β , dan una diversidad de formas para algunas distribuciones de probabilidad particulares; estos parámetros se conocen como los parámetros de forma y escala. (15)

1.3.5 Distribución Geométrica

Aplicación: En situaciones donde ingenieros o administradores intentan determinar cuán ineficiente es un sistema durante el periodo de tiempo utilizado. En teoría de probabilidad y estadística, la distribución geométrica es la distribución de probabilidad discreta siguiente: (16)

$$g_p(k) \equiv P(X = k) = (1 - p)^{k-1} p \quad x \in \{1, 2, 3 \dots\}$$

1.3.6 Distribución Uniforme

Aplicación: Se aplica a fenómenos de máxima incertidumbre, cuando no hay información que permita otorgar más probabilidad a unas zonas que a otras. Dentro de las variables aleatorias continuas, la más sencilla es esta; su característica principal es establecer para todos los eventos asociados a un experimento la misma probabilidad en un intervalo dado, lo que quiere decir que la variable está distribuida uniformemente en un intervalo dado; una definición más precisa es la siguiente:

Sea X una variable aleatoria continua, definida en un intervalo (a, b) de los números reales, la variable aleatoria está distribuida de manera uniforme si su función de probabilidad es:

$$f(x; a, b) = \frac{1}{b - a} \text{ si } a \leq x \leq b$$

0 para cualquier caso.

Aplicando la definición de función de distribución acumulada $F(x) = P(X \leq x) = \int_{-\infty}^x f(t)dt$ y el segundo teorema fundamental del cálculo $P(a \leq x \leq b) = F(b) - F(a)$ se tiene la siguiente distribución de probabilidad acumulada para una distribución de probabilidad uniforme continua:

$$F(x) = P(X \leq x) = \int_{-\infty}^x \frac{dt}{b-a} = \frac{1}{b-a} \int_a^x dt \text{ si } a \leq x \leq b$$

Al aplicar las propiedades de la integral, se llega al siguiente resultado:

$$\begin{aligned} &0 \text{ si } x < a \\ F(x) = P(X \leq x) &= \frac{x-a}{b-a} \text{ si } a \leq x \leq b \\ &1 \text{ si } x > b \end{aligned}$$

La distribución anterior, cuando $a = 0$ y $b = 1$, se conoce como distribución uniforme sobre el intervalo unitario $(0,1)$, fundamental en la generación de números aleatorios, instrumento muy utilizado en la simulación por computador. (17)

1.3.7 Correlaciones Canónicas

Aplicación: Las correlaciones canónicas fueron introducidas por Hotelling en 1935,1936 como una respuesta a simples problemas con bastas implicaciones. Encuentra la combinación lineal de un juego de variables, las cuales tienen un alto grado de correlación con cualquier combinación lineal de un segundo juego de variable. (18)

Formalmente, dado un juego de variables explicativas $\{X_1, X_2, \dots, X_q\}$ y un juego de variables dependientes $\{Y_1, Y_2, \dots, Y_p\}$. La siguiente ecuación establece la relación entre dos juegos de variables. (19)

$$a_1 \cdot Y_1 + a_2 \cdot Y_2 + \dots + a_p \cdot Y_p = b_1 \cdot X_1 + b_2 \cdot X_2 \dots + b_q \cdot X_q$$

1.4 Metodología, tecnologías y herramientas empleadas en la solución

Para el desarrollo del módulo resulta necesario realizar el estudio de la metodología, las herramientas y las tecnologías definidas por el grupo de arquitectura del departamento de Soluciones Integrales de DATEC. En el presente documento se caracterizan las utilizadas para el desarrollo de la solución.

1.4.1 Metodología de desarrollo de software

El proceso de desarrollo de software es un conjunto de actividades que guían los esfuerzos de las personas implicadas en un proyecto. Una metodología de desarrollo de software constituye una secuencia de pasos y procedimientos que deben seguirse para desarrollar un software. Esta tiene como objetivo: registrar los requisitos de un sistema informático, proporcionar un método sistemático de desarrollo, construir un sistema bien documentado y fácil de mantener. Una metodología específica cómo dividir un proyecto en etapas, qué tareas se llevan a cabo en cada etapa, qué restricciones deben aplicarse, qué técnicas y herramientas se emplean y cómo se controla y gestiona un proyecto. (20)

Open Up (Proceso Unificado Abierto)

Es una metodología de desarrollo de software que estructura el ciclo de vida de un proyecto en cuatro fases: Concepción, Elaboración, Construcción y Transición.

1. Concepción: Primera de las 4 fases en el ciclo de vida de un proyecto, proporciona el entendimiento del propósito y los objetivos, obteniendo suficiente información para conocer qué debe hacer el proyecto. El objetivo de esta fase es capturar las necesidades de los stakeholder³ en los objetivos del ciclo de vida para el proyecto.

2. Elaboración: Es la segunda de las 4 fases del ciclo de vida del Open UP donde se tratan los riesgos significativos para la arquitectura. El propósito de esta fase es establecer la elaboración de la arquitectura del sistema.

3. Construcción: Esta fase está enfocada al diseño, implementación y prueba de las funcionalidades para desarrollar un sistema completo. El propósito de esta fase es completar el desarrollo del sistema basado en la arquitectura definida.

³ **Stakeholder:** el término agrupa a trabajadores, organizaciones sociales, accionistas y proveedores, entre muchos otros actores clave que se ven afectados por las decisiones de una empresa.

4. Transición: Es la última fase, cuyo propósito es asegurar que el sistema sea entregado a los usuarios, y evalúa la funcionalidad y rendimiento del último entregable de la fase de construcción.

Beneficios de su uso

Es apropiado para proyectos pequeños y de bajos recursos, permite disminuir las probabilidades de fracaso del proyecto e incrementar las probabilidades de éxito.

Permite detectar errores tempranos a través de un ciclo iterativo.

Evita la elaboración de documentación, diagramas e iteraciones innecesarios requeridos en la metodología Rational Unified Process (RUP).

Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas. (21)

Open Up preserva la esencia de RUP pues es iterativo e incremental, dirigido por casos de usos y centrado en la arquitectura. Esta metodología de desarrollo ofrece a los miembros del equipo una amplia visibilidad de los puntos de decisión a través del proyecto, permitiendo una supervisión efectiva y una correcta toma de decisiones en el momento adecuado. (22) Además el proyecto debe ser realizado a corto plazo y el equipo de desarrollo es muy pequeño, factores que traen consigo que la metodología sea la más adecuada para el desarrollo del módulo.

1.4.2 Herramientas CASE

CASE son las siglas de: Computer Aided Software Engineering; y en su traducción al español significa Ingeniería de Software Asistida por Computación. Representan la forma que permite modelar los procesos de negocio de las empresas, estas son un elemento importante, que permite llevar adelante el proyecto de forma eficaz. Algunos ejemplos de herramientas CASE son: ERwin, EasyCASE, Oracle Designer, PowerDesigner y Visual Paradigm. (23)

Herramienta CASE seleccionada para el modelado

Se selecciona para efectuar el modelado de la solución propuesta el Visual Paradigm para UML, debido a que es una herramienta profesional que se encuentra disponible en plataformas como Windows y Linux, tiene licencia gratuita y soporta aplicaciones web, permitiendo crear los diagramas correspondientes durante todo el ciclo de vida de desarrollo del software.

Entre sus características se destacan las siguientes:

- ✓ Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- ✓ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ✓ Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- ✓ Generación de documentación: brinda la posibilidad de documentar todo el trabajo sin necesidad de utilizar herramientas externas.
- ✓ Tiene apoyo adicional en cuanto a generación de artefactos automáticamente. (24)

1.4.3 Lenguaje de programación

Cualquier lenguaje artificial que pueda utilizarse para definir una secuencia de instrucciones para procesar por un ordenador se puede denominar lenguaje de programación. Un lenguaje de programación está formado de un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Además permite especificar de manera precisa sobre qué datos debe operar una computadora, cómo deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural. Hoy existen un gran número de lenguajes, tales como Java, C, C#, C++, PHP, Java Script, Shell, Python, Pascal y Perl. (25)

Hypertext Pre-processor (PHP)

PHP es un lenguaje de programación de uso general de script del lado del servidor, diseñado originalmente para la creación de páginas web dinámicas. Puede ser incrustado dentro de código HTML⁴ y generalmente se ejecuta tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. (26)

Se selecciona este lenguaje de programación web para la realización del módulo de análisis estadístico, fundamentalmente porque se ajusta a las necesidades de implementación y por las valiosas

⁴ **HTML:** lenguaje de marcado de hipertexto. Es un lenguaje que se utiliza para la creación de páginas web.

características y ventajas que brinda. Además, los productos desarrollados en el centro de DATEC están implementados en dicho lenguaje, lo que permite una futura integración entre ellos y el módulo.

Ventajas de este lenguaje:

- ✓ Se caracteriza por ser un lenguaje muy rápido.
- ✓ Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para la Universidad. (27)

PHP 5.3.10

Es importante señalar que para la solución propuesta se utiliza PHP en su versión 5.3.10 debido a las características que incorpora. Esta nueva versión se centra no sólo en la corrección de errores, sino además en la simplificación del código, así como su reutilización de una manera sencilla y rápida. Entre sus ventajas más destacables se encuentran la buena administración de la memoria y un mejor rendimiento, de esta forma el servidor trabaja de manera más rápida. (28)

1.4.4 Marco de trabajo

En el desarrollo de software, un marco de trabajo o Framework en su traducción al inglés, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser fácilmente organizado y desarrollado. Típicamente, incluye soporte de programas, bibliotecas y un lenguaje interpretado que ayuda a desarrollar y unir los diferentes componentes de un proyecto. (29) Entre los marcos de trabajo que se pueden citar aparecen Zend Framework, CakePHP, CodeIgniter, Seagull y Symfony. Para el desarrollo de la propuesta de solución se utilizará Symfony 2.0.1

Symfony 2.0.1

Después de seleccionar PHP como lenguaje de programación, se elige como marco de desarrollo para el lado del servidor Symfony en su versión 2.0.1 pues este ha sido probado en numerosos proyectos reales de la Universidad, presenta utilidades que simplifican el desarrollo de las aplicaciones web con PHP y está diseñado para optimizar el desarrollo de las mismas. Es fácil de utilizar pues cuenta con una API de

desarrollo muy intuitivo y es de código abierto, lo que permite su libre uso y distribución. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web, lo que mantiene el código organizado y permite a la aplicación evolucionar fácilmente en el tiempo. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. (30) En esta versión desaparecen los plugins y se sustituyen por un elemento mucho más potente llamado bundle (módulo) modificable. Consta de buena seguridad, validación y es muy versátil. Al mismo tiempo es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. (31)

1.4.5 Ambiente de desarrollo

Los Entornos de Desarrollo Integrados, por sus siglas en inglés (IDE) proporcionan herramientas que dan soporte al proceso de desarrollo del software, incluyendo editores para escribir y editar programas, y depuradores para localizar errores lógicos. En el presente trabajo se hace uso de Net Beans IDE 7.2.

NetBeans IDE 7.2

NetBeans ofrece una versión del IDE hecho a medida para el desarrollo de sitios web, puede editar, ejecutar o depurar archivos PHP. NetBeans IDE soporta muchos de los frameworks PHP más populares. En su versión 7.2 introduce soporte para Symfony 2 con la ayuda de las plantillas Twig. El IDE además proporciona autocompletado de código y reconoce la sintaxis de este framework. Es una herramienta de código abierto donde los programadores pueden escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Además es un producto libre y gratuito sin restricciones de uso, tiene soporte para desarrollar aplicaciones web. (32)

1.4.6 Servidor Web

Un servidor web es un programa informático que utiliza el protocolo de transferencia de hipertexto (HTTP)⁵ para recibir las peticiones de información de un programa cliente (navegador) en el ordenador del usuario, aunque el equipo donde se ejecuta recibe la misma denominación. Es la tecnología que apela a programas informáticos que tienen implícitos, que procesan aplicaciones ejecutando conexiones con el cliente, concibiendo o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. Es

⁵ **HTTP:** protocolo mediante el cual se transfieren las páginas web a un ordenador.

un programa especialmente diseñado para transferir datos de hipertexto, es decir, páginas web con todos sus elementos. Los servidores web están alojados en un ordenador que cuenta con conexión a Internet y se encuentra a la espera de que algún navegador realice una petición. (33)

Apache 2.2

Apache es un servidor HTTP potente y flexible, es un software de código abierto y uso gratuito que implementa el protocolo HTTP/1.1 y la noción de sitio virtual, corre sobre una multitud de plataformas y sistemas operativos y posee una estrecha relación con la mayoría de los lenguajes de desarrollo por parte del servidor. Presenta entre otras características mensajes de error altamente configurables e intercambio de contenido. Aunque RServer se puede montar sobre diferentes servidores HTTP, se utiliza Apache 2.2 porque fue el servidor utilizado en la primera versión del módulo, además por sus características y ventajas expuestas anteriormente. (34)

1.4.7 Servicios web

Un servicio web es un componente de software almacenado en una computadora, el cual se puede utilizar mediante llamadas a métodos desde una aplicación u otro componente de software en otra computadora a través de una red. Es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades. (35) Los servicios web son muy prácticos, pueden aportar gran independencia entre la aplicación que usa el servicio web y el propio servicio, esta flexibilidad será cada vez más importante, dado que la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños es cada día más utilizada. (36)

Protocolo de Simple Acceso a Objetos (SOAP)

En el núcleo de los servicios Web se encuentra SOAP, que proporciona un mecanismo estándar para empaquetar mensajes. SOAP es el protocolo que define como se establece el intercambio de información mediante XML⁶, y UDDI⁷ es el protocolo empleado para publicar la información del servicio web y que

⁶ **XML:** es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos.

⁷ **UDDI:** es un registro diseñado para almacenar de forma estructurada los datos.

permite comprobar los servicios web que se encuentran disponibles. (37) Además, SOAP no está vinculado con ningún lenguaje, no se encuentra fuertemente asociado a ningún protocolo de transporte y tampoco está atado a ninguna infraestructura de objeto distribuido. (38)

Conclusiones del capítulo

Se realizó el estudio de diferentes sistemas de análisis estadísticos y se arribó a la conclusión de que R es el software más apropiado para ejecutar los análisis en el RServer. Se estudiaron los modelos matemáticos de las funciones a implementar y se definieron la metodología, las herramientas y tecnologías a utilizar para el desarrollo del software.

Como metodología se utilizó Open UP, con la que el centro ha tenido muy buenos resultados ya que permite la construcción de los artefactos necesarios para el desarrollo del proyecto. Se empleó Visual Paradigm en su versión 6.4 para el modelado del sistema y como lenguaje de programación PHP en su versión 5.3, con el cual se implementan las clases necesarias para la comunicación con el servidor de análisis estadístico.

CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL MÓDULO STATS

Introducción

En el presente capítulo se pone de manifiesto el proceso ingenieril de desarrollo del software. Se describe el análisis de la solución propuesta utilizando UML como lenguaje de modelado y Open UP como metodología de desarrollo. Incluye el modelo de dominio y la especificación de los requisitos funcionales y no funcionales que guían el desarrollo del software. Se determinan el actor y el caso de uso, los cuales quedan representados en el diagrama de casos de uso del sistema, se modela el diagrama de clases del diseño, además se define la arquitectura y los patrones de diseño que se van a emplear en la implementación, obteniéndose como resultado la estructura arquitectónica para el módulo STATS del RServer 2.0.

2.1 Modelo de dominio

El modelo del dominio constituye una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. El modelo de dominio puede utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema y puede ser tomado como el punto de partida para el diseño del software.

Importante: Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes de software. No se trata de un conjunto de diagramas que describen clases de software, u objetos de software con responsabilidades. (39)

A continuación se representa el modelo de dominio de la aplicación.

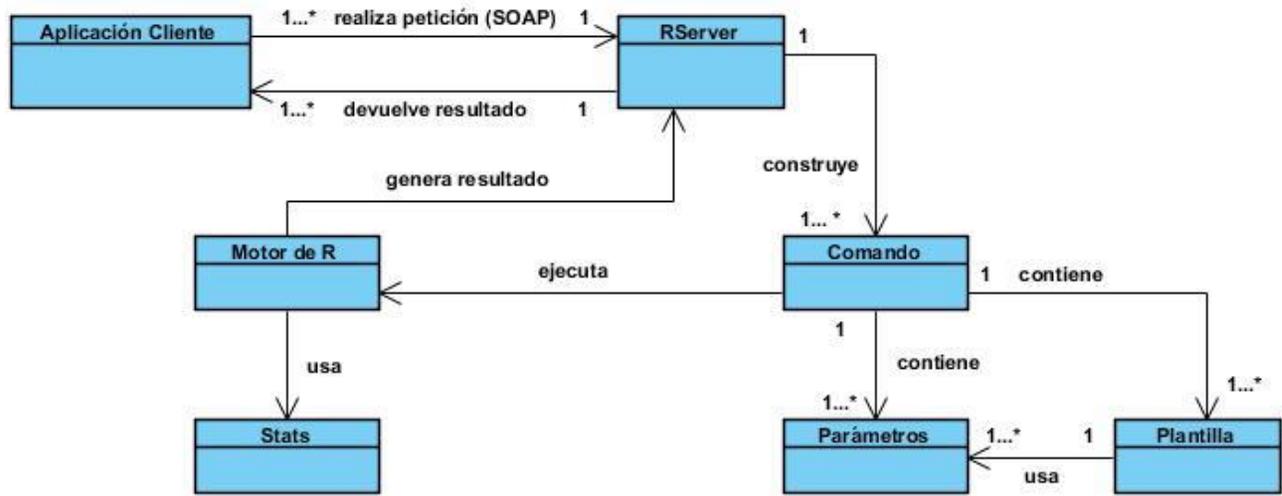


Ilustración 1. Modelo de Dominio

Descripción del modelo de dominio

La figura 1 representa el diagrama de dominio del sistema RServer, donde la Aplicación Cliente realiza una petición por el protocolo SOAP al RServer enviando un conjunto de parámetros, este a su vez construye un Comando que está constituido por las Plantillas y por los Parámetros enviados por la Aplicación Cliente. La Plantilla estructura los Parámetros para su cálculo posterior, una vez conformado el comando, este se ejecuta en R. R emplea un paquete llamado Stats que tiene los métodos de análisis estadísticos necesarios para la implementación del módulo STATS. Finalmente se genera un resultado que el RServer devuelve a dicha Aplicación Cliente en formato JSON.

Definición de las entidades y conceptos principales

- ✓ Aplicación Cliente: Es una aplicación informática que consume el servicio remoto en otra computadora conocida como servidor.
- ✓ Comando: Es la instrucción u orden que se proporciona al sistema informático. En el caso del RServer es un comando bash que ejecuta el código R.
- ✓ Plantilla: Una plantilla es un archivo de texto. La plantilla define el nombre de la función y la estructura que van a tener los parámetros en R.
- ✓ Parámetros: Son valores que utiliza la plantilla para generar el cálculo con el código R.

- ✓ Motor de R: Es el sistema ejecutándose, es cuando R procesa los datos enviados por la aplicación.
- ✓ Stats: Este paquete contiene las funciones para cálculos estadísticos.

2.2 Requisitos de software

Una parte fundamental del trabajo está dirigida a lograr una correcta especificación de los requisitos de software, estos representan la descripción de los servicios y restricciones de un sistema de software, es decir, lo que el software debe hacer y bajo qué circunstancias debe hacerlo. (40)

Los requisitos de software se dividen en dos categorías:

2.2.1 Requisitos funcionales

Son declaraciones de los servicios que debe proporcionar el sistema, la manera en que este debe reaccionar a entradas y de cómo se debe comportar en situaciones particulares. Estos describen lo que el sistema debe hacer, aunque en algunos casos los requisitos funcionales también pueden declarar explícitamente lo que el sistema no debe hacer. Deben ser completos, consistentes, no ambiguos, precisos, verificables y modificables; de esta forma se garantiza el cumplimiento de los objetivos planteados. Para el desarrollo del sistema se han definido los siguientes requisitos funcionales:

RF_1 Realizar análisis estadístico de datos por el método de la densidad de Beta.

Entrada: `dbeta(x, shape1, shape2, ncp = 0, log = FALSE)`

Descripción: Calcula la densidad de Beta, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

RF_2 Realizar análisis estadístico de datos por el método de la distribución acumulada de Beta.

Entrada: `pbeta(q, shape1, shape2, ncp = 0, lower.tail = TRUE, log.p = FALSE)`

Descripción: Calcula la distribución acumulada de Beta, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

RF_3 Realizar análisis estadístico de datos por el método de la distribución de cuantiles de Beta.

Entrada: `qbeta(p, shape1, shape2, ncp = 0, lower.tail = TRUE, log.p = FALSE)`

Descripción: Calcula la distribución de cuantiles de Beta, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

RF_4 Realizar análisis estadístico de datos para generar números aleatorios de Beta.

Entrada: `rbeta(n, shape1, shape2, ncp = 0)`

Descripción: Genera los números aleatorios de Beta, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

Parámetros (Argumentos)

x, q: vector de cuantiles.

P: vector de probabilidades.

N: número de observaciones. Si la longitud (n) > 1, la longitud se toma para ser el número requerido.

shape1, shape2: parámetros positivos de la distribución Beta.

Ncp: no centralidad parámetro.

log, log.p: lógico; si es TRUE, p probabilidades se dan como $\log(p)$.

lower.tail: lógico, si es TRUE (por defecto), las probabilidades son $P[X \leq x]$, de lo contrario, $P[X > x]$.

RF_5 Realizar análisis estadístico de datos por el método de la densidad de Cauchy.

Entrada: `dcauchy(x, location = 0, scale = 1, log = FALSE)`

Descripción: Calcula la densidad de Cauchy, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

RF_6 Realizar análisis estadístico de datos por el método de la distribución acumulada de Cauchy.

Entrada: `pcauchy(q, location = 0, scale = 1, lower.tail = TRUE, log.p = FALSE)`

Descripción: Calcula la distribución acumulada de Cauchy, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

RF_7 Realizar análisis estadístico de datos por el método de la distribución de cuantiles de Cauchy.

Entrada: `qcauchy(p, location = 0, scale = 1, lower.tail = TRUE, log.p = FALSE)`

Descripción: Calcula la distribución de cuantiles de Cauchy, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

RF_8 Realizar análisis estadístico de datos para generar números aleatorios de Cauchy.

Entrada: `rcauchy(n, location = 0, scale = 1)`

Descripción: Genera los números aleatorios de Cauchy, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

Parámetros (Argumentos)

x, q: vector de cuantiles.

P: vector de probabilidades.

N: número de observaciones. Si la longitud (n) > 1, la longitud se toma para ser el número requerido.

location, scale: parámetros de localización y escala.

log, log.p: lógico; si es TRUE, p probabilidades se dan como log (p).

lower.tail: lógico, si es TRUE (por defecto), las probabilidades son $P[X \leq x]$, de lo contrario, $P[X > x]$.

RF_9 Realizar análisis estadístico de datos por el método de la densidad de F.

Entrada: `df(x, df1, df2, ncp, log = FALSE)`

Descripción: Calcula la densidad de F, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON

RF_10 Realizar análisis estadístico de datos por el método de la distribución acumulada de F.

Entrada: `pf(q, df1, df2, ncp, lower.tail = TRUE, log.p = FALSE)`

Descripción: Calcula la distribución acumulada de F, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

RF_11 Realizar análisis estadístico de datos por el método de la distribución de cuantiles de F.

Entrada: `qf(p, df1, df2, ncp, lower.tail = TRUE, log.p = FALSE)`

Descripción: Calcula la distribución de cuantiles de F, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

RF_12 Realizar análisis estadístico de datos para generar números aleatorios de F.

Entrada: `rf(n, df1, df2, ncp)`

Descripción: Genera los números aleatorios de F, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

Parámetros (Argumentos)

x, q: vector de cuantiles.

P: vector de probabilidades.

N: número de observaciones. Si la longitud (n) > 1, la longitud se toma para ser el número requerido.

DF1, DF2: grados de libertad.

Ncp: no centralidad parámetro. Si se omite el F central es asumida.

log, log.p: lógico; si es TRUE, p probabilidades se dan como log (p).

lower.tail: lógico, si es TRUE (por defecto), las probabilidades son $P [X \leq x]$, de lo contrario, $P [X > x]$.

RF_13 Realizar análisis estadístico de datos por el método de la densidad de Gamma.

Entrada: `dgamma(x, shape, rate = 1, scale = 1/rate, log = FALSE)`

Descripción: Calcula la densidad de Gamma, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

RF_14 Realizar análisis estadístico de datos por el método de la distribución acumulada de Gamma.

Entrada: `pgamma(q, shape, rate = 1, scale = 1/rate, lower.tail = TRUE, log.p = FALSE)`

Descripción: Calcula la distribución acumulada de Gamma, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

RF_15 Realizar análisis estadístico de datos por el método de la distribución de cuantiles de Gamma.

Entrada: `qgamma(p, shape, rate = 1, scale = 1/rate, lower.tail = TRUE, log.p = FALSE)`

Descripción: Calcula la distribución de cuantiles de Gamma, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

RF_16 Realizar análisis estadístico de datos para generar números aleatorios de Gamma.

Entrada: `rgamma(n, shape, rate = 1, scale = 1/rate)`

Descripción: Genera los números aleatorios de Gamma, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

Parámetros (Argumentos)

`x, q`: vector de cuantiles.

`P`: vector de probabilidades.

`N`: número de observaciones. Si la longitud (`n`) > 1, la longitud se toma para ser el número requerido.

`Rate`: una forma alternativa para especificar la escala.

`shape, scale`: parámetros de forma y escala. Debe ser positivo, escalar en sentido estricto.

`log, log.p`: lógico; si es TRUE, `p` probabilidades se dan como $\log(p)$.

`lower.tail`: lógico, si es TRUE (por defecto), las probabilidades son $P[X \leq x]$, de lo contrario, $P[X > x]$.

RF_17 Realizar análisis estadístico de datos por el método de la densidad Geométrica.

Entrada: `dgeom(x, prob, log = FALSE)`

Descripción: Calcula la densidad de Geométrica, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

RF_18 Realizar análisis estadístico de datos por el método de la distribución acumulada Geométrica.

Entrada: `pgeom(q, prob, lower.tail = TRUE, log.p = FALSE)`

Descripción: Calcula la distribución acumulada Geométrica, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

RF_19 Realizar análisis estadístico de datos por el método de la distribución de cuantiles Geométrica.

Entrada: `qgeom(p, prob, lower.tail = TRUE, log.p = FALSE)`

Descripción: Calcula la distribución de cuantiles Geométrica, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

RF_20 Realizar análisis estadístico de datos para generar números aleatorios Geométrica.

Entrada: `rgeom(n, prob)`

Descripción: Genera los números aleatorios Geométrica, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

Parámetros (Argumentos)

x, q : vector de cuantiles que representan el número de errores en una secuencia de ensayos de

Bernoulli se produce antes de éxito.

P: vector de probabilidades.

N: número de observaciones. Si la longitud (n) > 1, la longitud se toma para ser el número requerido.

Prob: probabilidad de éxito en cada ensayo. $0 < \text{prob} \leq 1$.

log, log.p: lógico; si es TRUE, p probabilidades se dan como log (p).

lower.tail: lógico, si es TRUE (por defecto), las probabilidades son $P [X \leq x]$, de lo contrario, $P [X > x]$.

RF_21 Realizar análisis estadístico de datos por el método de la densidad Uniforme.

Entrada: `dunif(x, min=0, max=1, log = FALSE)`

Descripción: Calcula la densidad de Uniforme, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

RF_22 Realizar análisis estadístico de datos por el método de la distribución acumulada Uniforme.

Entrada: `punif(q, min=0, max=1, lower.tail = TRUE, log.p = FALSE)`

Descripción: Calcula la distribución acumulada Uniforme, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

RF_23 Realizar análisis estadístico de datos por el método de la distribución de cuantiles Uniforme.

Entrada: `qunif(p, min=0, max=1, lower.tail = TRUE, log.p = FALSE)`

Descripción: Calcula la distribución de cuantiles Uniforme, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

RF_24 Realizar análisis estadístico de datos para generar números aleatorios Uniforme.

Entrada: `runif(n, min=0, max=1)`

Descripción: Genera los números aleatorios Uniforme, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

RF_25 Realizar análisis estadístico de Correlaciones Canónicas entre dos matrices.

Entrada: `cancor(matrix1, matrix2, xcenter, ycenter)`

Descripción: Calcula la correlación canónica entre matrices, ejecutando una plantilla predeterminada y pasándole la entrada correspondiente. La plantilla ejecuta el método en R y se devuelve el resultado final.

Salida: String en formato JSON.

Parámetros (Argumentos)

`x, q`: vector de cuantiles.

`P`: vector de probabilidades.

`N`: número de observaciones. Si la longitud (`n`) > 1, la longitud se toma para ser el número requerido.

`min, max`: límites inferior y superior de la distribución. Debe ser finito.

`log, log.p`: lógico; si es TRUE, `p` probabilidades se dan como $\log(p)$.

`lower.tail`: lógico, si es TRUE (por defecto), las probabilidades son $P[X \leq x]$, de lo contrario, $P[X > x]$.

`matrix1`: matriz numérica que contiene las coordenadas `x`.

`matrix2`: matriz numérica que contiene las coordenadas `y`.

`xcenter`: vector lógico o numérico.

`ycenter`: análoga a `xcenter` pero para los valores de `y`.

2.2.2 Requisitos no funcionales

Estos requerimientos a menudo se aplican al sistema en su totalidad y no a características o servicios individuales del sistema. Por su parte, son los que imponen restricciones en el diseño o la implementación.

(41) Los requisitos no funcionales son:

RNF_Software

Se debe tener instalado en el servidor donde se despliegue el RServer, R en su versión 2.13, Symfony 2.0., Apache 2.0 o superior y PHP 5.3 o superior y GNU Linux.

RNF_Hardware

El sistema requiere de una memoria RAM de 256 MB con un ordenador Pentium III o equivalente. Se necesita una capacidad de almacenamiento en el disco duro de al menos 20GB para la instalación de las tecnologías necesarias para el despliegue de la aplicación.

RNF_Restricciones de Diseño

El sistema debe ser implementado en el lenguaje de programación PHP 5.3.

Se utiliza SOAP para la conexión con el servidor y el envío de parámetros se hace en formato JSON

Se requiere el marco de trabajo Symfony 2.0 para el desarrollo del sistema del lado del servidor.

2.3 Diagrama de casos de uso del sistema

Los diagramas de caso de uso documentan el comportamiento de un sistema desde el punto de vista del usuario y los casos de uso son una unidad coherente de funcionalidad, expresada como transacción entre los actores y el sistema, estos determinan los requisitos funcionales del sistema. El propósito de la vista de casos de uso es enumerar los actores y los casos de uso, y demostrar las relaciones que puedan existir entre ellos. (42) Una vez especificado los requisitos funcionales, estos se engloban en un caso de uso, denominado Realizar Análisis Estadístico.

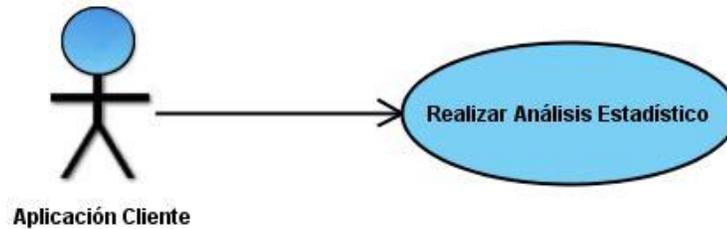


Ilustración 2. Diagrama de Casos de Uso del Sistema

Actor	Descripción
Aplicación Cliente	Programa implementado en cualquier lenguaje de programación que sea capaz de conectarse con el RServer a través del protocolo SOAP.

Tabla 2. Descripción del actor del sistema

2.4 Descripción del caso de uso del sistema

El caso de uso Realizar Análisis Estadístico inicia cuando la Aplicación Cliente realiza la petición del servicio mediante el protocolo SOAP. El servidor construye un comando que contiene la plantilla a renderizar con los parámetros de entrada. El motor de R ejecuta dicho comando y el RServer devuelve la respuesta a la aplicación cliente en formato JSON. La siguiente tabla describe las secciones del caso de uso.

Objetivo	Realizar un análisis estadístico con los parámetros que la aplicación cliente envía por el protocolo SOAP.
Actores	Aplicación Cliente.
Resumen	Realiza el análisis estadístico de los parámetros enviados por la aplicación cliente. Estos parámetros tienen un formato JSON. A partir de ellos, se ejecuta la funcionalidad correspondiente con el tipo de análisis y a esta funcionalidad se le pasa el resto de los

	parámetros que serían los datos a calcular. Una vez realizado el cálculo, el RServer devuelve una respuesta a la aplicación cliente en formato JSON.	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	El cliente ha enviado los parámetros correctamente.	
Postcondiciones	Se realizó correctamente el análisis y se envió un resumen del análisis en formato JSON.	
Flujo de eventos		
Flujo básico. Realizar análisis estadístico de datos		
	Actor	Sistema
	Envía los parámetros a calcular en formato JSON.	
		<p>Según los parámetros recibidos por la aplicación cliente que serían los datos a calcular, se ejecuta la funcionalidad correspondiente al tipo de análisis y se envía a una de las secciones siguientes.</p> <ol style="list-style-type: none"> 1- Realizar análisis estadístico de datos por el método de la distribución acumulada de Beta. 2- Realizar análisis estadístico de datos por el método de la distribución de cuantiles de Beta. 3- Realizar análisis estadístico de datos por el método de la densidad de Beta. 4- Realizar análisis estadístico de datos para generar números aleatorios de Beta. 5- Realizar análisis estadístico de datos por el método de la distribución acumulada de Cauchy. 6- Realizar análisis estadístico de datos por el método de la distribución de cuantiles de Cauchy.

	<p>7- Realizar análisis estadístico de datos por el método de la densidad de Cauchy.</p> <p>8- Realizar análisis estadístico de datos para generar números aleatorios de Cauchy.</p> <p>9- Realizar análisis estadístico de datos por el método de la distribución acumulada de F.</p> <p>10- Realizar análisis estadístico de datos por el método de la densidad de F.</p> <p>11- Realizar análisis estadístico de datos por el método de la distribución de cuantiles de F.</p> <p>12- Realizar análisis estadístico de datos para generar números aleatorios de F.</p> <p>13- Realizar análisis estadístico de datos por el método de la distribución acumulada de Gamma.</p> <p>14- Realizar análisis estadístico de datos por el método de la distribución de cuantiles de Gamma.</p> <p>15- Realizar análisis estadístico de datos por el método de la densidad de Gamma.</p> <p>16- Realizar análisis estadístico de datos para generar números aleatorios de Gamma.</p> <p>17- Realizar análisis estadístico de datos por el método de la distribución acumulada Geométrica.</p> <p>18- Realizar análisis estadístico de datos por el método de la distribución de cuantiles Geométrica.</p> <p>19- Realizar análisis estadístico de datos por el método de la densidad Geométrica.</p> <p>20- Realizar análisis estadístico de datos para generar números aleatorios Geométrica.</p> <p>21- Realizar análisis estadístico de datos por el método de la distribución de cuantiles Uniforme.</p> <p>22- Realizar análisis estadístico de datos por el método de la</p>
--	---

		distribución acumulada Uniforme. 23- Realizar análisis estadístico de datos por el método de la densidad Uniforme. 24- Realizar análisis estadístico de datos para generar números aleatorios Uniforme. 25- Realizar análisis estadístico de datos por el método de Correlaciones Canónicas entre matrices.
		Termina el caso de uso.
Flujos alternos		
Nº 1. Datos enviados mal estructurados		
	Actor	Sistema
		Enviará un mensaje de error. Por lo que es recomendable visitar la ayuda para ver la estructura que deben llevar los parámetros.
Sección 1: “Realizar análisis estadístico de la distribución acumulada de Beta”		
Flujo básico. Realizar análisis estadístico de la distribución acumulada de Beta		
	Actor	Sistema
		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían ser un vector de cuantiles, dos parámetros positivos de la distribución Beta, un parámetro que determina la centralidad de una gráfica t-Student. Si se omite, el estudio se realiza con la gráfica centralizada en 0, en este caso se puede omitir pues no se va a graficar y dos variables lógicas, todos en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
		Devuelve el resultado en formato JSON.
Sección 2: “Realizar análisis estadístico de la distribución de cuantiles de Beta”		
Flujo básico. Realizar análisis estadístico de la distribución de cuantiles de Beta		
	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían ser un

		vector de probabilidades, dos parámetros positivos de la distribución Beta, un parámetro que determina la centralidad de una gráfica t-Student. Si se omite, el estudio se realiza con la gráfica centralizada en 0, en este caso se puede omitir, pues no se va a graficar y dos variables lógicas. Todos, en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.
Sección 3: “Realizar análisis estadístico de la densidad de Beta”		
Flujo básico. Realizar análisis estadístico de la densidad de Beta		
	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían ser un vector de cuantiles, dos parámetros positivos de Beta, un parámetro que determina la centralidad de una gráfica. Si se omite, el estudio se realiza con la gráfica centralizada en 0, en este caso se puede omitir, pues no se va a graficar y una variable lógica. La plantilla va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.
Sección 4: “Realizar análisis estadístico para generar números aleatorios de Beta”		
Flujo básico Realizar análisis estadístico generando números aleatorios de Beta		
	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían ser un vector de cuantiles, dos parámetros positivos de Beta, un parámetro que determina la centralidad de una gráfica. Si se omite, el estudio se realiza con la gráfica centralizada en 0, en este caso se puede omitir, pues no se va a graficar. La plantilla va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.
Sección 5: “Realizar análisis estadístico de la distribución acumulada de Cauchy”		

Flujo básico. Realizar análisis estadístico de la distribución acumulada de Cauchy		
	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían ser un vector de cuantiles, un parámetro de localización, un parámetro de escala y dos variables lógicas. Todos, en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.
Sección 6: “Realizar análisis estadístico por el método de la distribución de cuantiles de Cauchy”		
Flujo básico. Realizar análisis estadístico de la distribución de cuantiles de Cauchy		
	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían ser un vector de probabilidades, un parámetro de localización, un parámetro de escala y dos variables lógicas. Todos, en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.
Sección 7: “Realizar análisis estadístico de la densidad de Cauchy”		
Flujo básico. Realizar análisis estadístico de la densidad de Cauchy		
	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían ser un vector de cuantiles, un parámetro de localización, un parámetro de escala y una variable lógica. Todos, en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.
Sección 8: “Realizar análisis estadístico de datos para generar números aleatorios de Cauchy”		
Flujo básico. Realizar análisis estadístico generando números aleatorios de Cauchy		
	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos

		parámetros. Los parámetros de esta sección deberían ser números de observaciones, un parámetro de localización y uno de escala. Todos, en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.

Sección 9: “Realizar análisis estadístico de datos por el método de la distribución acumulada de F”

Flujo básico. Realizar análisis estadístico de la distribución acumulada de F

	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían ser un vector de cuantiles, dos grados de libertad, un parámetro que determina la centralidad de una gráfica (se omite) y dos variables lógicas. Todos, en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.

Sección 10: “Realizar análisis estadístico de datos por el método de la densidad de F”

Flujo básico. Realizar análisis estadístico por el método de la densidad de F

	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían ser un vector de cuantiles, dos grados de libertad, un parámetro que determina la centralidad de una gráfica (se omite) y dos variables lógicas. Todos, en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.

Sección 11: “Realizar análisis estadístico de datos por el método de la distribución de cuantiles de F”

Flujo básico. Realizar análisis estadístico de la distribución de cuantiles de F

	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos

		parámetros. Los parámetros de esta sección deberían ser un vector de probabilidades, dos grados de libertad, un parámetro que determina la centralidad de una gráfica (se omite) y dos variables lógicas. Todos, en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.

Sección 12: “Realizar análisis estadístico de datos para generar números aleatorios de F”

Flujo básico. Realizar análisis estadístico generando números aleatorios de F

	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían un número de observaciones, dos grados de libertad, un parámetro que determina la centralidad de una gráfica. Si se omite, el estudio se realiza con la gráfica centralizada en 0, en este caso se puede omitir. Todos, en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.

Sección 13: “Realizar análisis estadístico de datos por el método de la distribución acumulada de Gamma”

Flujo básico. Realizar análisis estadístico de la distribución acumulada de Gamma

	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían ser un vector de cuantiles, un parámetro de forma y escala, este debe ser positivo y escalar en sentido estricto, un parámetro que especifique la escala y dos variables lógicas. Todos, en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.

Sección 14: “Realizar análisis estadístico de datos por el método de la distribución de cuantiles de Gamma”

Flujo básico. Realizar análisis estadístico de la distribución de cuantiles de Gamma		
	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían ser un vector de probabilidades, un parámetro de forma y escala, este debe ser positivo y escalar en sentido estricto, un parámetro que especifique la escala y dos variables lógicas. Todos, en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.
Sección 15: “Realizar análisis estadístico de datos por el método de la densidad de Gamma”		
Flujo básico. Realizar análisis estadístico de la densidad de Gamma		
	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían ser un vector de cuantiles, un parámetro de forma y escala, este debe ser positivo y escalar en sentido estricto, un parámetro que especifique la escala, el propio parámetro escala y una variable lógica. Todos, en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.
Sección 16: “Realizar análisis estadístico de datos para generar números aleatorios de Gamma”		
Flujo básico. Realizar análisis estadístico de datos generando números aleatorios de Gamma		
	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían ser un número de observaciones, un parámetro de forma y escala, una variable alternativa para especificar la escala y un parámetro escala. Todos, en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.

Sección 17: “Realizar análisis estadístico de datos por el método de la distribución acumulada Geométrica”

Flujo básico. Realizar análisis estadístico de la distribución acumulada Geométrica

	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían ser un vector de cuantiles, la probabilidad de éxito y dos variables lógicas. Todos, en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.

Sección 18: “Realizar análisis estadístico de datos por el método de la distribución de cuantiles Geométrica”

Flujo básico. Realizar análisis estadístico de la distribución de cuantiles Geométrica

	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían ser un vector de probabilidades, la probabilidad de éxito del ensayo y dos variables lógicas. Todos, en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.

Sección 19: “Realizar análisis estadístico de datos por el método de la densidad Geométrica”

Flujo básico. Realizar análisis estadístico de datos por el método de la densidad Geométrica

	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían ser un vector de cuantiles, la probabilidad de éxito del ensayo y una variable lógica. Todos, en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.

Sección 20: “Realizar análisis estadístico de datos para generar números aleatorios Geométrica”

Flujo básico. Realizar análisis estadístico de datos generando números aleatorios Geométrica		
	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían ser el número de observaciones y la probabilidad de éxito del ensayo. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.
Sección 21: “Realizar análisis estadístico de datos por el método de la distribución de cuantiles Uniforme”		
Flujo básico. Realizar análisis estadístico de la distribución de cuantiles Uniforme		
	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían ser un vector de probabilidades, una variable mínima y una variable máxima que son los límites superior e inferior de la distribución y dos variables lógicas. Todos, en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.
Sección 22: “Realizar análisis estadístico de datos por el método de la distribución acumulada Uniforme”		
Flujo básico. Realizar análisis estadístico de datos por el método de la distribución acumulada Uniforme		
	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían ser un vector de cuantiles, una variable mínima y una variable máxima que son los límites superior e inferior de la distribución y dos variables lógicas. Todos, en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.

Sección 23: “Realizar análisis estadístico de datos por el método de la densidad Uniforme”		
Flujo básico. Realizar análisis estadístico de la densidad Uniforme		
	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Los parámetros de esta sección deberían ser un vector de cuantiles, una variable mínima y una variable máxima que son los límites superior e inferior de la distribución y otra variable lógica. Todos, en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.
Sección 24: “Realizar análisis estadístico de datos para generar números aleatorios Uniforme”		
Flujo básico. Realizar análisis estadístico generando números aleatorios Uniforme		
	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Cuyos parámetros de esta sección deberían ser un número de observaciones, una variable mínima y una variable máxima que son los límites superior e inferior de la distribución. Todos, en ese orden. Esta a la vez va al motor de R y realiza el cálculo.
2.		Devuelve el resultado en formato JSON.
Sección 25: “Realizar análisis estadístico por el método de Correlaciones Canónicas entre matrices”		
Flujo básico. Realizar análisis estadístico de Correlaciones Canónicas entre matrices		
	Actor	Sistema
1.		Ejecuta una plantilla predeterminada en R, pasándole dichos parámetros. Cuyos parámetros de esta sección deberían ser xcenter: vector lógico o numérico de longitud p1, que describe cualquier centrado que se realiza en los valores de x antes del análisis y ycenter, análoga a xcenter, pero para los valores de y. Esta a la vez va al motor de R y realiza el cálculo.

2.	Devuelve el resultado en formato JSON.
----	--

Tabla 3. Descripción del caso de uso del sistema

Para dar cumplimiento a todas las secciones del caso de uso Realizar Análisis Estadístico proporcionando el uso de cada funcionalidad que se pone de manifiesto a través de un servicio web, se implementa el sistema sobre la arquitectura SOA.

2.5 Arquitectura orientada a servicios (SOA)

La arquitectura orientada a servicios permite que la funcionalidad de la aplicación se exponga y consuma como un conjunto de servicios. Los servicios usan una forma estándar de interacción que les permite ser invocados, publicados y descubiertos. Los servicios SOA proveen interfaces con alcance de aplicación en vez de interfaces del nivel de componente u objeto. Con SOA, se puede reemplazar un servicio sin tener que preocuparse por la tecnología fundamental; la interfaz es lo que importa, y está definida en un estándar universal de servicios Web y XML.

2.6 Patrones de diseño y de arquitectura de software

Los patrones de diseño y los patrones arquitectónicos ofrecen soluciones a problemas de construcción del software. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. En comparación con los patrones de diseño, los patrones arquitectónicos tienen un nivel de abstracción mayor. (43) Durante la elaboración del sistema se hizo empleo de diferentes patrones, dentro de los cuales se destacan la inyección de dependencias, el patrón experto, el controlador, el controlador frontal y el modelo-vista-controlador.

MVC (Patrón Modelo Vista Controlador)

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP). El modelo se encarga de la abstracción de la lógica relacionada con los datos. En el diagrama de clases del diseño se modeló solamente el controlador, puesto que la vista está dada por el WSDL que genera en XML el controlador para la interacción de la aplicación con el servidor, esto provee la interfaz de comunicación que representa

la vista abstracta según define SOA. No es objetivo de este trabajo usar bases de datos, por lo que no se presenta un modelo; pero se considera una opción para posteriores implementaciones del módulo. (44)

Inyección de dependencias

Es un patrón de diseño, en el que se suministran objetos a una clase en lugar de ser la propia clase quien cree los objetos. "Inyección de Dependencia es donde los componentes reciben sus dependencias a través de sus constructores, métodos o directamente en los campos." (45)

Experto

Consiste en asignar una responsabilidad al experto en información: la clase que cuenta con información necesaria para cumplir la responsabilidad. (46)

Controlador

Es un operador artificial de todos los eventos del sistema de un caso de uso. Se utiliza para asignar la responsabilidad del manejo de mensajes de los eventos de un sistema, donde un evento del sistema es un evento de alto nivel generado por un actor externo; es un evento de entrada externa. A partir de aquí el controlador ofrece una guía para operaciones posteriores. (47)

Controlador Frontal

Es un patrón de diseño que se basa en usar un controlador como punto inicial para la gestión de las peticiones. Todas las peticiones web son manejadas por un solo controlador frontal, que es el punto de entrada único de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL escrita o seleccionada por el usuario. (48)

2.7 Diagrama de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y las interfaces en una aplicación, estas describen la estructura del sistema. A continuación se expone el diagrama de clases del diseño que se generó para lograr el entendimiento de la solución.

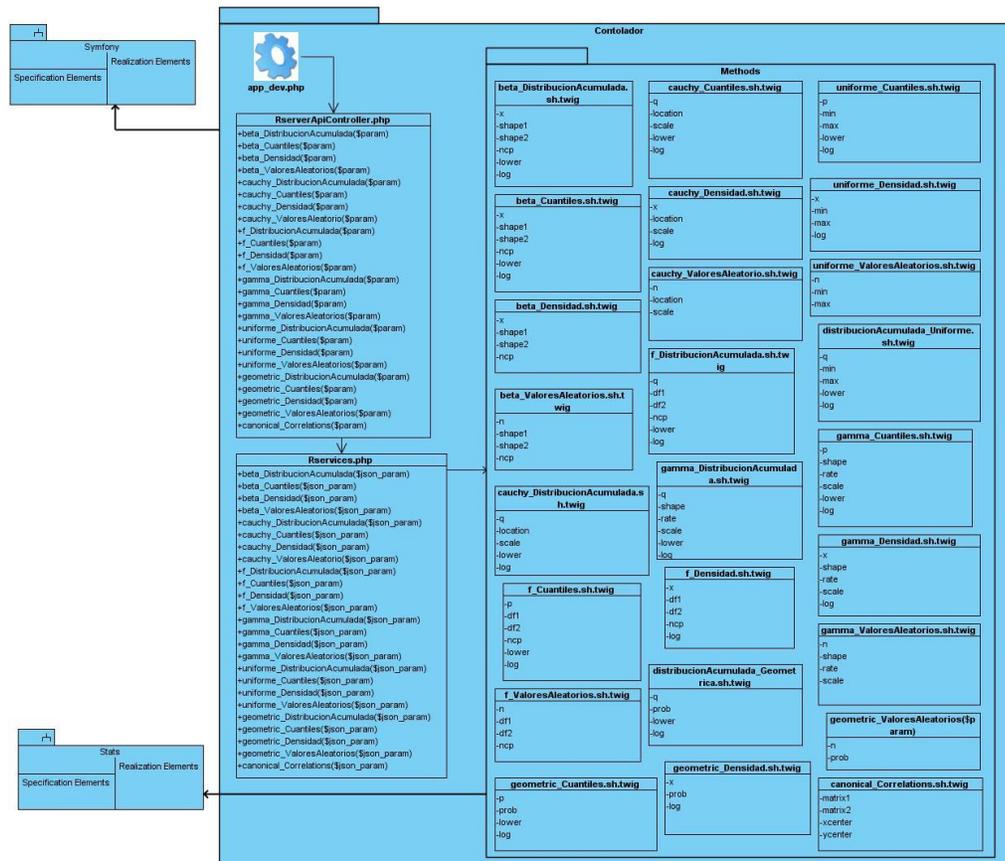


Ilustración 3. Diagrama de Clases del Diseño

Descripción del diagrama de clases del diseño

Inicialmente se encuentra el controlador frontal `app_dev.php`, el mismo inicia el flujo de Symfony. A la clase `RserverApiController.php` llegan los parámetros enviados por la aplicación cliente, esta a su vez le pasa los parámetros a `Rservices.php` mediante la inyección de dependencia previamente configurada. `Rservices.php` es la clase que se encarga de vincular el motor de R con el lenguaje PHP, la misma ejecuta la plantilla correspondiente al tipo de análisis en cuestión que se encuentra en el paquete `Methods`. Estas plantillas utilizan un paquete implementado en R llamado `Stats`.

Definición de las clases principales

- ✓ `RserverApiController.php`: Esta clase provee el servicio web brindado por el `RServer` y genera la descripción de los métodos en el lenguaje descriptor de servicios web (WSDL).

- ✓ Rservices.php: Contiene las funcionalidades y renderiza las plantillas twig, vincula el motor de R con el lenguaje PHP.
- ✓ Plantilla.sh.twig: Posee el nombre de la función y la estructura de los parámetros en R.

2.8 Diagrama de secuencia

El diagrama de secuencia se encuentra entre los diagramas de interacción, ellos modelan el comportamiento dinámico de un sistema, muestran la forma en que un grupo de objetos interactúan entre sí a lo largo del tiempo. Este diagrama consta de objetos, mensajes entre estos objetos y una línea de vida del objeto representada por una línea vertical. (49)

A continuación se muestra el diagrama de secuencia de el caso de uso Realizar Análisis Estadístico que cuenta con un total de 25 secciones y todas siguen el mismo flujo, lo único que cambia es la funcionalidad que se va a ejecutar, por lo que solo es distinta la plantilla a renderizar. Por estas razones no se hace necesario plasmar todos estos diagramas, ni siquiera en los anexos.

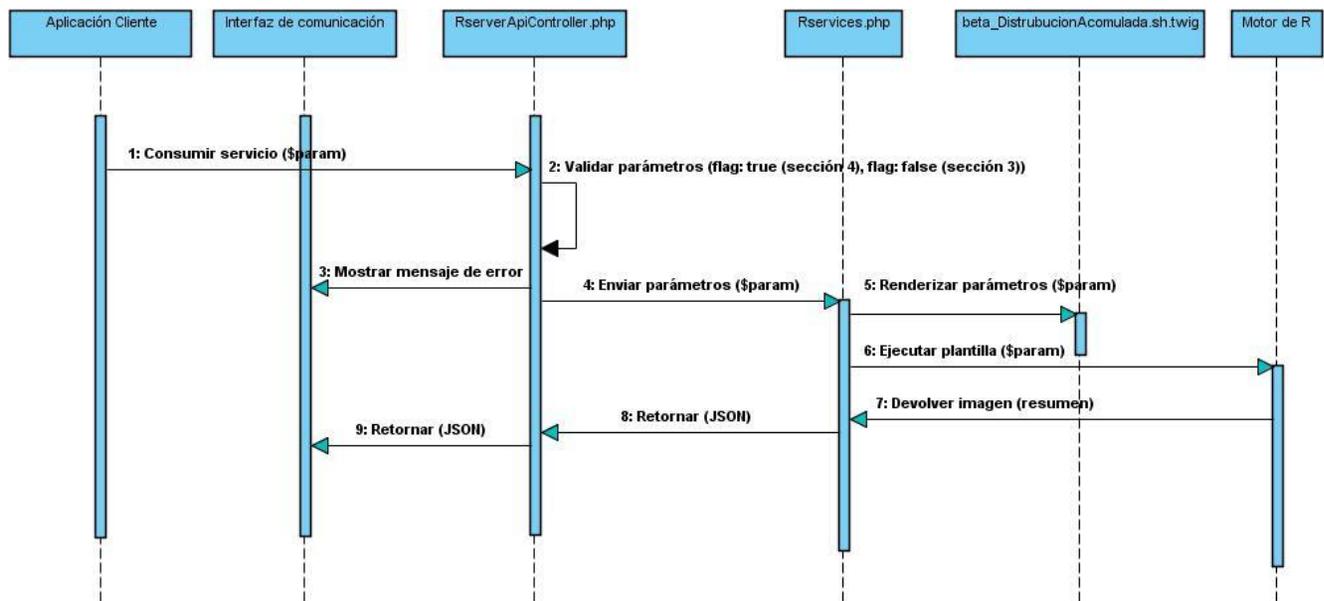


Ilustración 4. Diagrama de Secuencia del Caso de Uso: Realizar Análisis Estadístico, Sección Realizar Análisis Estadístico de Datos por el método de la distribución acumulada de Beta

Descripción del diagrama de secuencia

La Aplicación Cliente envía una petición al RServer pasándole un conjunto de parámetros, la clase RserverApiController.php valida que los datos entrados estén correctamente estructurados, de no ser así, muestra un mensaje de error, de lo contrario continúa ejecutándose correctamente. Dicha clase envía esos parámetros ya evaluados y codificados a la clase Rservices.php, esta los decodifica y le pasa los parámetros a la plantilla predeterminada para el tipo de análisis en cuestión. R realiza el cálculo predeterminado y devuelve un resumen a Rservices.php, esta clase vuelve a codificarlo y lo retorna a RserverApiController.php, esta lo decodifica una vez más y finalmente retorna el resultado en un formato de JSON.

Aclaración: La manera en la que el usuario inserta los datos y accede al servidor es completamente transparente al presente trabajo de diploma. La manera en que recibe la respuesta generada por el sistema no está definida en el alcance de la tesis, pues depende de la persona que utilice los servicios del RServer diseñar un modelo que le permita capturar los datos, sin embargo, como esto puede ser genérico se representa en el diagrama una interfaz de comunicación que constituye una forma correcta de devolver el resultado.

2.9 Diagrama de despliegue

"Los diagramas de despliegue muestran la configuración física de un sistema, revelando que piezas de software se ejecutan sobre que piezas de hardware." (50)

La figura 5 presenta una vista actualizada del diagrama de despliegue, estos se utilizan para visualizar la topología de los componentes físicos de un sistema donde los componentes de software se han implementado.



Ilustración 5. Diagrama de Despliegue

Descripción del diagrama de despliegue

El nodo PC Cliente representa donde va a estar la aplicación cliente que va a consumir el servicio que brindará el RServer 2.0, mediante el protocolo SOAP. Puede ser cualquier aplicación que desee realizar o graficar un análisis estadístico de los datos. En el nodo RServer 2.0 es donde será desplegada la aplicación, quedando allí instalado el motor de cálculo y las aplicaciones necesarias para el correcto funcionamiento del mismo.

Conclusiones del capítulo

El capítulo ayuda a traducir las necesidades del cliente en modelos que representan la estructura y el diseño arquitectónico del sistema, además de explicar los procedimientos que ejecuta. Se confeccionó el modelo de dominio, se hizo el levantamiento de los requisitos funcionales y no funcionales del sistema, quedando los requisitos funcionales agrupados en un caso de uso que dio lugar a la elaboración del diagrama de casos de uso del sistema. Además se elaboró el diagrama de clases del diseño, el diagrama de secuencia y el diagrama de despliegue. Se ofreció una breve explicación de cada uno de los modelos para obtener un mejor entendimiento y se definieron un conjunto de patrones a seguir para la integración de nuevas funcionalidades y módulos al RServer 2.0. Todos estos elementos dan paso al capítulo de implementación y pruebas.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA DEL MÓDULO

Introducción

En el presente capítulo se generan los artefactos correspondientes a esta fase, donde se modela y describe el diagrama de componentes y se define el estándar de codificación para la implementación. Se definen las técnicas, tipos y métodos de pruebas que se emplearán en la revisión de la aplicación, se diseñan y ejecutan los casos de prueba y se visualizan fragmentos de códigos de la aplicación en PHP y en R.

3.1 Modelo de implementación

El modelo de implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos. (51)

3.1.1 Diagrama de componentes

Los diagramas de componentes muestran como el sistema está dividido en componentes y las dependencias que existen entre ellos, además de proveer una vista arquitectónica de alto nivel del sistema.

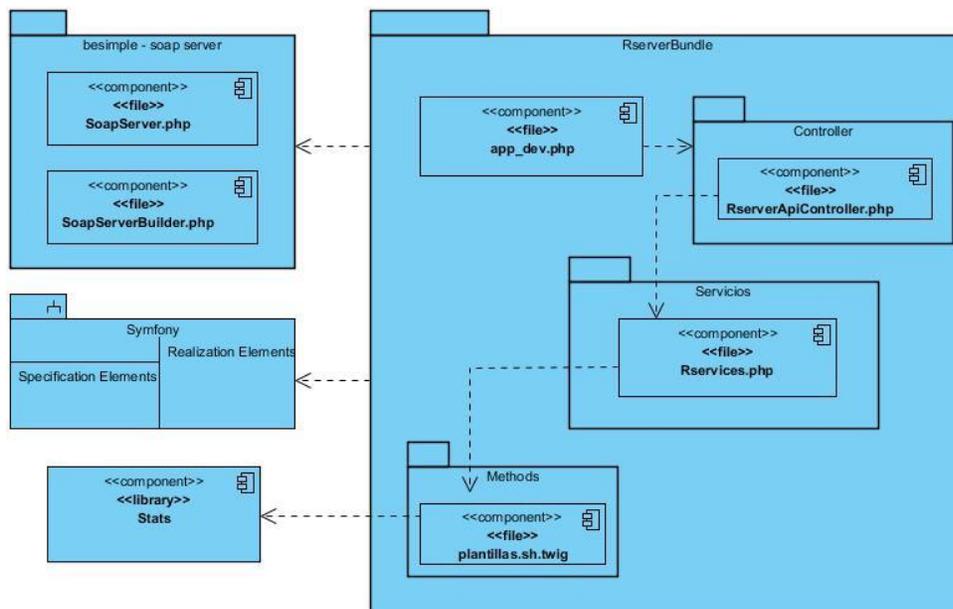


Ilustración 6. Diagrama de Componentes

Descripción del diagrama de componentes

El diagrama representa como queda la estructura de la aplicación. Dentro del RserverBundle (módulo de análisis estadístico) se encuentra el controlador frontal, que da inicio al paquete de controladores. También se encuentra la clase RserverApiController.php, clase que provee el servicio web y que pasa los parámetros a la clase Rservices.php, esta última es la que vincula al lenguaje R con PHP. Rservices.php ejecuta las plantillas que se encuentran en el subsistema de componentes Methods, estas plantillas usan el paquete Stats del motor de R. RserverBundle utiliza el bundle besimple-soap server para el envío de JSON a la aplicación cliente.

3.2 Estándares de codificación

La legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema de software. La mantención del código es la facilidad con que el sistema de software puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores, o mejorar el rendimiento. Un código fuente completo debe reflejar un estilo armonioso, que permita a futuros programadores continuar la implementación si fuera necesario.

Estilo de codificación utilizado

- ✓ Los nombres de las clases están escritos con el estándar “UpperCamelCase”.
- ✓ Los nombres de las variables están escritos con la Notación C.
- ✓ Los nombres de los métodos están escritos con un híbrido entre el estándar “lowerCamelCase” y la Notación C.
- ✓ Los delimitadores del cuerpo de la función están: el de apertura, en la misma línea que la declaración de la función, separado por un espacio. La llave de cierre está en una línea aparte y no tiene más código que la propia llave de cierre.
- ✓ Todas las etiquetas php deben ser completas (<?php ?>) no reducidas (<? ?>).
- ✓ Las variables usadas tienen nombres que las identifican, su nombre es coherente con lo que contienen.

- ✓ El arreglo multidimensional (o asociativo), se pone en líneas separadas, e indentadas una tabulación respecto al nombre de la variable que lo contiene.
- ✓ Los operadores se escriben con espacios a ambos lados.

3.3 Implementaciones significativas

Siguiendo los patrones de diseño propuestos en el capítulo anterior, el epígrafe muestra la implementación de métodos incorporados al módulo de análisis estadístico del RServer. Inicialmente se muestra el flujo del cálculo de la distribución de cuantiles de Beta, desde el servicio hasta la plantilla programada en R y posteriormente el flujo de cálculo de la Correlación Canónica entre dos matrices de datos.

```
/**
 * @Soap\Method("beta_Cuantiles")
 * @Soap\Param("params", phpType = "string")
 * @Soap\Result(phpType = "string")
 */
public function beta_CuantilesAction($params) {
    $rservices = $this->get('rserver.stats');
    $result = $rservices->beta_CuantilesAction($params);
    return $this->result($result);
}
```

Ilustración 7. Método que devuelve la distribución de cuantiles de Beta en formato JSON

En la figura se muestra un segmento de código que visualiza cómo se calcula el análisis estadístico de la distribución de cuantiles de Beta. En la parte superior, mediante notaciones de Symfony se describe el nombre de la función, los parámetros de entrada y el resultado que se obtiene. El método recibe un parámetro de tipo JSON y se crea la variable `$rservices` para la inyección de dependencias. Usando esta variable se puede acceder a todas las funcionalidades de la clase `Rservices.php`. Posteriormente la variable `$result` guardará el resultado que devuelve la función de `Beta_CuantilesAction ()` (Ilustración 8). Por último se devuelve un objeto JSON, en una respuesta SOAP.

```

public function beta_CuantilesAction($json_params) {
    $params = json_decode($json_params, true);
    $result = array();

    try {
        $comando = $this->plantilla->render(
            'RserverBundle:Methods:beta_Cuantiles.sh.twig', $params);
        $data = exec($comando, $out);
    } catch (\Exception $exc) {
        $data = array('success' => false, 'error_code' => $exc->getCode(), 'mensaje' =>
            $exc->getMessage());
    }
    $result['success'] = true;
    $result['data'] = $out;
    return json_encode($result);
}

```

Ilustración 8. Método que calcula la distribución de cuantiles de Beta

Esta funcionalidad recibe un objeto JSON, que es decodificado y guardado en la variable `$params`. Se inicializa la variable `$result` como un arreglo, donde se guardará la respuesta. Luego en un bloque `try catch` para capturar posibles errores, son encerradas las líneas, dentro de estas se guardan en una variable la ejecución de los parámetros en una plantilla definida para ese método (Ilustración 9) y en la otra se ejecuta dicho comando. De haber algún error es capturado y luego devuelto como respuesta. De ocurrir una ejecución correcta, se devuelve el resultado codificado como un JSON.

```

#!/bin/sh
R --slave << EOF
qbeta(c({{p}}), ({{shape1}}), ({{shape2}}), ncp = {{0}}, lower.tail = {{lower_tail}}, log.p =
{{log_p}})
EOF

```

Ilustración 9. Método que calcula la distribución de cuantiles de Beta en el lenguaje R

Este es el cálculo que se realiza de la distribución de cuantiles en el lenguaje R, que está implementado predeterminadamente en una plantilla llamada `Beta_Cuantiles.sh.twig`. Las dos primeras líneas de código

son las que levantan el motor de R y la última es el cierre. La tercera línea es la función de Cuantiles de Beta (qbeta), donde p es un vector de probabilidades, shape1 y shape2 son parámetros positivos de la distribución Beta, ncp es un parámetro que determina la centralidad de una gráfica, que si se omite, el método se realiza con este valor en 0, lower.tail y log.p son dos variables lógicas.

3.4 Pruebas de software

Un instrumento adecuado para determinar el estado de la calidad de un producto de software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. (52)

Las pruebas de software implican la aplicación de condiciones controladas, estas deben incluir tanto situaciones normales como anormales. El objetivo de las siguientes pruebas es encontrar errores para determinar situaciones en las que el sistema responde de manera incorrecta.

3.4.1 Tipos de prueba

Existen las pruebas de caja negra y las pruebas de caja blanca, las técnicas de caja negra o funcional realizan pruebas sobre la interfaz del programa a probar, entendiendo por interfaz las entradas y salidas de dicho programa. Mientras que las técnicas de caja blanca o estructural, se basan en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa. Se comprueban los caminos lógicos del software proponiendo casos de prueba que examinen que están correctas todas las condiciones y/o bucles para determinar si el estado real coincide con el esperado o afirmado. Esto genera gran cantidad de caminos posibles por lo que hay que dedicar esfuerzos a la determinación de las condiciones de prueba que se van a verificar. (53)

3.4.2 Pruebas de Caja Blanca

Se toma el método canonical_Correlations anteriormente explicado, para ejecutar las pruebas de caja blanca, donde se hacen separaciones del código en bloques para después proceder a la realización de las pruebas requeridas.

```

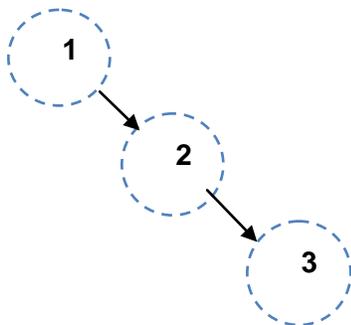
public function canonical_CorrelatiosActions ($json_params) {
$params = json_decode ($json_params, true);
$result = array ();
Try {
$comando = $this->plantilla->render (
'RserverBundle: Methods: canonical_correlations.sh.twig', $param);
$data = exec ($comando, $out);
}
catch (\Exception $exc){
$data = array ('success'=> false, 'error_code' => $exc ->getCode (), 'mensaje'
=> $exc -> getMessage ());
}

$result['success'] = true;
$result['data'] = $out;

Return json_encode ($result);
}

```

Quedando conformado el grafo de la siguiente manera:



Prueba del Camino Básico

La prueba del camino básico es una técnica de prueba de la Caja Blanca propuesta por Tom McCabe. Esta técnica permite obtener una medida de la complejidad lógica de un diseño. Se obtiene un conjunto de caminos independientes, se construye el Grafo de Flujo asociado y se calcula su complejidad ciclomática.

Complejidad Ciclomática (La complejidad de McCabe $V(G)$). Es un indicador del número de caminos independientes que existen en un grafo.

$$V(G) = a - n + 2$$

$$V(G) = r$$

$$V(G) = c + 1$$

Donde:

a - número de arcos o aristas del grafo.

n - número de nodos.

r - número de regiones cerradas del grafo.

c - número de nodos de condición.

$V(G)$ - marca el límite mínimo de casos de prueba para un programa. Cuando $V(G) > 10$ la probabilidad de defectos en el módulo o programa crece mucho entonces quizás sea apropiado dividir el módulo.

Quedando de la siguiente forma:

$$a = 2, n = 3, r = 1, c = 0$$

$$V(G) = 2 - 3 + 2 = 1$$

$$V(G) = 1$$

$$V(G) = 0 + 1 = 1$$

Resultado de esta prueba: $V(G)$ no es mayor que 10, por lo que la probabilidad de defectos en el módulo no crece mucho, entonces no es necesario dividir dicho módulo.

3.4.3 Pruebas de Caja Negra

Dentro del método de Caja Negra la técnica de Partición de Equivalencia es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de pruebas

que descubran clases de errores, reduciendo así el número de clases de prueba que hay que desarrollar. (54)

A continuación se muestran casos de pruebas con algunas de las secciones del caso de uso Realizar Análisis Estadístico. Ver más en Anexos.

Descripción general: El caso de uso realiza el análisis estadístico de datos, según el tipo de análisis que se desea calcular. Inicia cuando la Aplicación Cliente realiza la petición del servicio mediante el protocolo SOAP y culmina cuando se tiene el resultado.

Condición de ejecución: Todos los valores insertados para la realización del cálculo deben estar correctamente estructurados.

Escenario	Descripción	Parámetros	Respuesta del sistema	Flujo central
EC 2.1 Calcular la distribución de cuantiles de Beta de forma correcta.	Se realiza el cálculo correctamente y se devuelve a la aplicación cliente un resumen del cálculo de la distribución de cuantiles de Beta.	V Se envían los parámetros correctamente, correspondientes con el tipo de análisis en cuestión.	Se envía el resultado de dicho cálculo a la aplicación cliente.	1- Recibir datos enviados a calcular. 2- Validar datos enviados, verificar que se encuentren correctamente estructurados. 3- Realizar análisis estadístico. 4- Enviar resultado obtenido por el sistema.
EC 2.2 Calcular la distribución de cuantiles de Beta de forma incorrecta.	Se le envía a la aplicación cliente un mensaje de error.	F No se envían los parámetros correspondientes con el tipo de análisis en cuestión o faltan datos.	Se envía un mensaje de error.	1- Recibir datos enviados a calcular. 2- Validar datos enviados, verificar que se encuentren correctamente estructurados. 3- Mostrar un mensaje de error.

Tabla 4. Realizar Análisis Estadístico sección Realizar análisis estadístico de la distribución de cuantiles de Beta

Escenario	Descripción	Parámetros	Respuesta del sistema	Flujo central
EC 4.1 Generar los números aleatorios de Beta de forma correcta.	Se realiza el cálculo correctamente y se devuelve a la aplicación cliente un resumen con los números aleatorios de Beta	V Se envían los parámetros correctamente correspondientes con el tipo de análisis en cuestión.	Se envía el resultado de dicho cálculo a la aplicación cliente.	1- Recibir datos enviados a calcular. 2- Validar datos enviados, verificar que se encuentren correctamente estructurados. 3- Realizar análisis estadístico. 4- Enviar resultado obtenido por el sistema.

	generados.			
EC 4.2 Generar los números aleatorios de Beta de forma incorrecta.	Se le envía a la aplicación cliente un mensaje de error.	F No se envían los parámetros correspondientes con el tipo de análisis en cuestión o faltan datos.	Se envía un mensaje de error.	1- Recibir datos enviados a calcular. 2- Validar datos enviados, verificar que se encuentren correctamente estructurados. 3- Mostrar un mensaje de error.

Tabla 5. Realizar Análisis Estadístico sección Realizar análisis estadístico generando números aleatorios de Beta

Se realizaron otras dos pruebas de Caja Negra: una para comprobar el correcto funcionamiento de los métodos de análisis estadísticos y otra para comprobar el WSDL, esta última se realizó cuando el módulo estaba implementado al 100%.

Comprobar el correcto funcionamiento de los métodos de análisis estadísticos: en esta prueba se confeccionó una clase llamada MethodController.php, en la misma se implementaron todas las funcionalidades correspondientes a los métodos de análisis estadísticos con datos bien estructurados, para simular el proceso del módulo.

```

public function canonical_CorrelationsAction() {
    $params = array(
        'x' => "1,2,3,4,5,6",
        'nfx' => 2,
        'ncx' => 3,
        'xbyrow' => 'FALSE',
        'y' => "1,2,3,4,5,6",
        'nfy' => 2,
        'ncy' => 3,
        'ybyrow' => 'FALSE',
        'xcenter' => 'TRUE',
        'ycenter' => 'TRUE'
    );
    $json_params = json_encode($params);

    $rserver = $this->get('rserver.stats');
    $result = $rserver->canonical_CorrelationsAction($json_params);
    echo "<pre>";
    echo $result;

    die;
}

```

Ilustración 10. Ejemplo de código con datos insertados

Comprobar el WSDL: Se implementa un cliente en PHP para comprobar el correcto funcionamiento del servicio web.

```
<?php

$wsdl = "http://localhost/Rserver/web/app_dev.php/ws/Rserver?wsdl";
$client = new SoapClient($wsdl);

$params = array(
    "q" => "0.5, 0.4, 0.3, 0.2",
    "shape1" => 2,
    "shape2" => 3,
    "ncp" => 0,
    "lower_tail" => "TRUE",
    "log_p" => "FALSE"
);

$json_params = json_encode($params);

try {
    $result = $client->beta_DistribucionAcumulada($json_params);
    echo $result;
} catch (Exception $e) {
    var_dump($e);
}
?>
```

Ilustración 11. Cliente implementado en PHP

3.4.4 Pruebas de rendimiento (Carga y Estrés)

Se realizaron pruebas de rendimiento (carga y estrés) para comprobar la velocidad con la que el sistema se desempeña dado varias muestras de usuarios conectados de manera concurrente. Para la realización de esta prueba se utilizó la herramienta JMeter.

Los resultados de la prueba se muestran en la figura, donde los tiempos de respuestas obtenidos como resultado son expresados en milisegundos (ms), donde 1 segundo equivale a 1000 milisegundos. Como resultado se observa que son tiempos de espera admisibles en el uso de una computadora, pues son tiempos que no llegan a 3 segundos.

Ver Resultados en Árbol

Nombre: Ver Resultados en Árbol

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Escribir en Log Sólo Errores

Muestra #	Start Time	Thread Name	Label	Tiempo de Muestr...	Status	Bytes
1	15:57:52.742	Grupo de Hilos 1-1	/	60019		3285
2	15:57:52.765	Grupo de Hilos 1-2	/	60026		3285
3	15:57:52.800	Grupo de Hilos 1-3	/	60008		3285
4	15:57:52.827	Grupo de Hilos 1-5	/	60006		3285
5	15:57:52.812	Grupo de Hilos 1-4	/	60042		3285
6	15:57:52.849	Grupo de Hilos 1-6	/	60065		3285
7	15:57:52.869	Grupo de Hilos 1-7	/	60064		3285
8	15:57:52.891	Grupo de Hilos 1-8	/	60057		3285
9	15:57:52.915	Grupo de Hilos 1-9	/	60049		3285
10	15:57:52.936	Grupo de Hilos 1-10	/	60047		3285
11	15:57:52.957	Grupo de Hilos 1-11	/	60058		3285
12	15:57:52.987	Grupo de Hilos 1-12	/	60958		3285
13	15:57:53.011	Grupo de Hilos 1-13	/	60935		3285
14	15:57:53.101	Grupo de Hilos 1-17	/	61897		3285
15	15:57:53.080	Grupo de Hilos 1-16	/	61929		3285
16	15:57:53.035	Grupo de Hilos 1-14	/	61976		3285
17	15:57:53.056	Grupo de Hilos 1-15	/	61955		3285
18	15:57:53.140	Grupo de Hilos 1-18	/	62813		3285
19	15:57:53.143	Grupo de Hilos 1-19	/	62827		3285
20	15:57:53.211	Grupo de Hilos 1-22	/	62762		3285
21	15:57:53.285	Grupo de Hilos 1-25	/	62688		3285
22	15:57:53.260	Grupo de Hilos 1-24	/	62714		3285
23	15:57:53.233	Grupo de Hilos 1-23	/	62743		3285
24	15:57:53.164	Grupo de Hilos 1-20	/	62844		3285
25	15:57:53.200	Grupo de Hilos 1-21	/	62814		3285
26	15:57:53.307	Grupo de Hilos 1-26	/	63703		3285
27	15:57:53.368	Grupo de Hilos 1-29	/	63659		3285
28	15:57:53.501	Grupo de Hilos 1-35	/	63527		3285

No. de Muestras 50 Última Muestra 64349 Media 62562 Desviación 1554

Ilustración 12. Prueba de carga y estrés en JMeter

Conclusiones del capítulo

En el capítulo se dio cumplimiento a los objetivos trazados en la investigación. Se comprueban las respuestas que genera el sistema dada las distintas situaciones que se pudieran presentar con la inserción de los datos y se verifica cada método implementado a través de una clase MethodController.php, construida especialmente para probar las funcionalidades. El módulo de análisis estadístico para RServer resulta completamente funcional y los resultados generales de las pruebas fueron satisfactorios.

CONCLUSIONES

A partir de los objetivos específicos planteados, se desarrolla el módulo de análisis estadístico en su versión 2.0 para el RServer, contribuyendo, en gran medida, al incremento de las funcionalidades que ofrece dicho servidor y dando cumplimiento al objetivo general del trabajo.

- ✓ El estudio de las bibliografías de los sistemas de análisis estadísticos actuales, permitió a los autores del trabajo establecer una comparación referente a varios de los elementos considerados importantes en el uso de estas herramientas. Considerando a R como el programa más adecuado para ejecutar los análisis estadísticos en el RServer.
- ✓ Se realizó el estudio de los métodos estadísticos en R y se definieron las nuevas funcionalidades que serán incorporados al módulo de análisis estadístico en su versión 2.0.
- ✓ Se fundamentó la metodología, las herramientas y las tecnologías empleadas en el desarrollo de la versión 2.0 del módulo de análisis estadístico para el RServer. Las herramientas y tecnologías utilizadas dieron soporte a las necesidades del equipo de trabajo y la metodología guió el proceso de desarrollo del software.
- ✓ Se realizó el análisis y diseño del módulo de análisis estadístico en su versión 2.0 para el RServer, obteniéndose como resultado los artefactos generados para la implementación.
- ✓ Se implementaron las nuevas funcionalidades del módulo de análisis estadístico en su versión 2.0 para el RServer y se dio cumplimiento a todos los requisitos funcionales.
- ✓ Se realizaron pruebas al módulo de análisis estadístico en su versión 2.0 para el RServer y se demostró que el producto está listo para ser usado en distintos ambientes y circunstancias.

Finalmente se logra una capa de abstracción que permite, a otros proyectos, utilizar las distribuciones de Beta, de Cauchy, de F, las distribuciones Geométrica, Gamma, Uniforme y además, las Correlaciones Canónicas, haciendo más fácil la obtención de los resultados y asegurando la exactitud y precisión en los mismos.

RECOMENDACIONES

- ✓ Se recomienda desarrollar nuevas versiones del módulo de análisis estadístico para el RServer, que ofrezcan una mayor cantidad de funciones y que incorporen la realización de gráficos.
- ✓ Usar la arquitectura propuesta, para la incorporación de los nuevos módulos y nuevos métodos estadísticos en el RServer.
- ✓ Diseñar una interfaz de usuario para la interacción con el RServer.

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS

1. **Ruz, Fidel Castro.** *Seminario Nacional para Educadores (VI)*. La Habana : Pueblo y Educación, 2005.
2. **Ruiz Muñoz, David.** Biblioteca Virtual. [Online] [Cited: 12 10, 2012.]
<http://www.eumed.net/coursecon/libreria/drm/0.htm>.
3. **Muñoz, David Ruiz.** *Manual de Estadística*. s.l. : eumed.net, 2004. 84-688-6153-7.
4. *Mathematical Methods of Statistics*. 1951.
5. **Ruiz Muñoz, David.** *Manual de Estadística*. s.l. : eumed.net, 2004. 84-688-6153-7.
6. **OpenCourseWare de la Universidad de Sevilla.** OpenCourseWare de la Universidad de Sevilla. [Online] 9 16, 2008. [Cited: 2 20, 2013.] http://ocwus.us.es/metodos-de-investigacion-y-diagnostico-en-educacion/analisis-de-datos-en-la-investigacion-educativa/Bloque_I/page_03.htm/.
7. —. OpenCourseWare de la Universidad de Sevilla. [Online] 9 16, 2008. [Cited: 2 20, 2013.]
http://ocwus.us.es/metodos-de-investigacion-y-diagnostico-en-educacion/analisis-de-datos-en-la-investigacion-educativa/Bloque_I/page_03.htm/.
8. **Salas, Christian.** *Ecología austral*. [Online] 6 19, 2008. [Cited: 2 21, 2013.]
http://www.scielo.org.ar/scielo.php?pid=S1667-782X2008000200007&script=sci_arttext.1667-782X.
9. **Rodríguez Coma, Magdalena.** Secretaría de Descentralización. [Online] [Cited: 12 09, 2012.]
<http://sd.pcm.gob.pe/contenido/902/Eje%20.%20Prog%20Inform%20Eval%20Cuant%20-%20Curso%20SAS%20Doc1%20-%20Prof%20Rod.pdf>.
10. **Institute for Statistics and Mathematics of the WU Wien.** The R Project for Statistical Computing. [Online] [Cited: 12 6, 2012.] <http://www.r-project.org/>.
11. **Salas, Christian.** *Ecología austral*. [Online] 6 19, 2008. [Cited: 2 21, 2013.]
http://www.scielo.org.ar/scielo.php?pid=S1667-782X2008000200007&script=sci_arttext.1667-782X.
12. **Cardona, Diego, Rivera, Miller, Romero, Jesús.** *DOCUMENTOS DE INVESTIGACIÓN. Una aproximación de la variable aleatoria a procesos de toma de decisión que implican condiciones de riesgo e incertidumbre*. Bogotá D.C : Universidad del Rosario, 2010. 0124-8219.
13. **Evangelina Fernández, Aída.** *FUNCIÓN CARACTERÍSTICA*. Depto de Matemática, Facultad de Ciencias Exactas y Tecnología, Universidad Nacional de Tucumán.
14. **Marques de Sá, Joaquim P.** *Applied Statistics Using SPSS, STATISTICA, MATLAB and R*. New York: Springer-Verlag Berlin Heidelberg, 2007. 978-3-540-71971-7.

REFERENCIAS BIBLIOGRÁFICAS

15. **Cardona, Diego, Rivera, Miller, Romero, Jesús.** *DOCUMENTOS DE INVESTIGACIÓN. Una aproximación de la variable aleatoria a procesos de toma de decisión que implican condiciones de riesgo e incertidumbre.* Bogotá D.C : Universidad del Rosario, 2010. 0124-8219.
16. **Marques de Sá, Joaquim P.** *Applied Statistics Using SPSS, STATISTICA, MATLAB and R.* New York: Springer-Verlag Berlin Heidelberg, 2007. 978-3-540-71971-7.
17. **Cardona, Diego, Rivera, Miller, Romero, Jesús.** *DOCUMENTOS DE INVESTIGACIÓN. Una aproximación de la variable aleatoria a procesos de toma de decisión que implican condiciones de riesgo e incertidumbre.* Bogotá D.C : Universidad del Rosario, 2010. 0124-8219.
18. **Muller, Keith E.** *Understanding Canonical Correlation Through the General Linear Model and Principal Components.* The American Statistician, 1982.
19. **Gorunescu, Florin.** *Intelligent Systems Reference Library Volume 12.* Springer-Verlag Berlin Heidelberg, 2011. 978-3-642-19720-8.
20. **Roberto Hernando.** Metodologías de desarrollo de software. [Online] [Cited: 12 08, 2012.] http://www2.rhernando.net/modules/tutorials/doc/ing/met_soft.html.
21. **EcuRed.** EcuRed. [Online] [Cited: 2 21, 2013.] <http://www.ecured.cu/index.php/OpenUp>.
22. **Yang, Michael.** OpenUP. [Online] [Cited: 12 6, 2012.] <http://epf.eclipse.org/wikis/openup/>.
23. **C&TA.** Ciencia y Técnica Administrativa. [Online] [Cited: 12 08, 2012.] <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>.
24. **EcuRed.** EcuRed. [Online] [Cited: 12 09, 2012.] http://www.ecured.cu/index.php/Visual_Paradigm.
25. **kioskea.net.** Lenguajes de programación. [Online] [Cited: 12 6, 2012.] <http://es.kioskea.net/contents/langages/langages.php3>.
26. **NetBeans.org.** NetBeans. [Online] [Cited: 12 09, 2012.] <http://netbeans.org/features/php/index.html>.
27. **ProgramacionDesarrollo.** ProgramacionDesarrollo. [Online] 02 05, 2011. [Cited: 12 06, 2012.] <http://programaciondesarrollo.es/ventajas-de-php/>.
28. **agapea.com.** agapea.com. [Online] [Cited: 3 10, 2013.] <http://www.agapea.com/libros/PHP-5-9788441517851-i.htm>.
29. **Rivas, José Manuel.** Codigonexo. [Online] 4 3, 2013. [Cited: 4 10, 2013.] <http://www.codigonexo.com/definiciones/framework/>.
30. **MaestrosWeb.** MaestrosWeb. [Online] [Cited: 12 09, 2012.] <http://www.maestrosdelweb.com/editorial/curso-symfony2-introduccion-instalacion/>.

REFERENCIAS BIBLIOGRÁFICAS

31. **Symfony.es.** Symfony. [Online] [Cited: 12 09, 2012.]
<http://www.symfony.es/noticias/2010/02/17/sflive2010-asi-es-symfony-2/>.
32. **NetBeans.org.** NetBeans. [Online] [Cited: 12 9, 2012.] <http://netbeans.org/features/php/index.html>.
33. **Consumoteca.** Consumoteca consumidores bien informados. [Online] 08 12, 2009. [Cited: 12 09, 2012.] <http://www.consumoteca.com/diccionario/servidor-web>.
34. **Formación online en Nuevas Tecnologías.** Formación online en Nuevas Tecnologías. [Online] 3 2012, 17. [Cited: 2 21, 2013.] <http://www.digitalllearning.es/blog/apache-servidor-web-configuracion-apache2-conf/>.
35. **HackTimes.** Seguridad en Web Services (WS). [Online] 06 27, 2011. [Cited: 12 12, 2012.]
http://www.hacktimes.com/seguridad_en_web_services/.
36. **EcuRed.** EcuRed. [Online] [Cited: 12 09, 2012.] http://www.ecured.cu/index.php/Servicios_Web.
37. **HackTimes.** Seguridad en Web Services (WS). [Online] 6 27, 2011. [Cited: 3 20, 2013.]
http://www.hacktimes.com/seguridad_en_web_services/.
38. **González C., Benjamín.** desarrolloweb.com. [Online] 07 07, 2004. [Cited: 12 09, 2012.]
<http://www.desarrolloweb.com/articulos/1557.php>.
39. **Larman, Craig.** *UML y Patrones. 2ª Edición.* Madrid : Prentice Hall, 2003. 8420534382.
40. **Sommerville, Ian.** *Ingeniería de Software. 6ta. Edición.* s.l. : Addison Wesley, 2012.
41. —. *Ingeniería del Software.* Madrid : Pearson, 2005. 84-7829-074-5.
42. **Rumbaugh, James, Jacobson, Ivar and Booch, Grady.** *El Lenguaje Unificado de Modelado.* Madrid : Addison Wesley, 2000. 84-7829-037-0.
43. **Bass, Len, Clements, Paul and Kazman, Rick.** *Software Architecture in Practice (2nd Edition).* s.l. : Addison Wesley, 2005.
44. **LibrosWeb.** LibrosWeb. [Online] [Cited: 12 07, 2012.]
http://www.librosweb.es/symfony_1_1/capitulo_2/el_patron_mvc.html.
45. **Potencier, Fabien.** Fabien Potencier. [Online] 3 26, 2009. [Cited: 2 12, 2013.]
<http://fabien.potencier.org/article/11/what-is-dependency-injection>.
46. **EVA.** Entorno Visual de Aprendizaje. [Online] [Cited: 2 12, 2013.]
http://eva.uci.cu/file.php/161/Documentos/Materiales_basicos/Materiales_basicos_de_la_Unidad_3/UML_y_Patrones/04_Parte_IV_Fase_del_Disenio_1_.pdf.

REFERENCIAS BIBLIOGRÁFICAS

47. —. Entorno Visual de Aprendizaje. [Online] [Cited: 2 12, 2013.]
http://eva.uci.cu/file.php/161/Documentos/Materiales_basicos/Materiales_basicos_de_la_Unidad_3/UML_y_Patrones/04_Parte_IV_Fase_del_Diseño_1_.pdf.
48. **LIBROSWEB**. LIBROSWEB. [Online] [Cited: 12 8, 2012.]
http://librosweb.es/symfony_1_2/capitulo_6/el_controlador_frontal.html.
49. **ALTOVA**. ALTOVA. [Online] [Cited: 2 21, 2013.] <http://www.altova.com/es/umodel/sequence-diagrams.html>.
50. **Fowler, Martin**. *UML Distilled 3era edición*. s.l. : Addison-Wesley, 2003.
51. **MeRinde.net**. MeRinde. [Online] [Cited: 3 12, 2013.]
http://merinde.net/index.php?option=com_content&task=view&id=495&Itemid=291.
52. **PruebasDeSoftware.com**. PruebasDeSoftware. [Online] [Cited: 3 12, 2013.]
<http://pruebasdesoftware.com/laspruebasdesoftware.htm>.
53. **EVA**. Tipos de prueba de Caja Blanca y Negra. [Online] [Cited: 3 20, 2013.]
<http://eva.uci.cu/mod/resource/view.php?id=8649&subdir=/Comun>.
54. **EcuRed.com**. EcuRed. [Online] [Cited: 3 25, 2013.]
http://www.ecured.cu/index.php/Pruebas_de_caja_negra.

BIBLIOGRAFÍA

- agapea.com.** agapea.com. [Online] [Cited: 3 10, 2013.] <http://www.agapea.com/libros/PHP-5-9788441517851-i.htm>.
- ALTOVA.** ALTOVA. [Online] [Cited: 2 21, 2013.] <http://www.altova.com/es/umodel/sequence-diagrams.html>.
- Bass, Len, Clements, Paul and Kazman, Rick.** *Software Architecture in Practice (2nd Edition)*. s.l. : Addison Wesley, 2005.
- C&TA.** Ciencia y Técnica Administrativa. [Online] [Cited: 12 08, 2012.] <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>.
- Cardona, Diego, Rivera, Miller, Romero, Jesús.** *DOCUMENTOS DE INVESTIGACIÓN. Una aproximación de la variable aleatoria a procesos de toma de decisión que implican condiciones de riesgo e incertidumbre*. Bogotá D.C : Universidad del Rosario, 2010. 0124-8219.
- Consumoteca.** Consumoteca consumidores bien informados. [Online] 08 12, 2009. [Cited: 12 09, 2012.] <http://www.consumoteca.com/diccionario/servidor-web>.
- EcuRed.** EcuRed. [Online] [Cited: 12 09, 2012.] http://www.ecured.cu/index.php/Visual_Paradigm.
- EcuRed.** EcuRed. [Online] [Cited: 12 09, 2012.] http://www.ecured.cu/index.php/Servicios_Web.
- EcuRed.** EcuRed. [Online] [Cited: 2 21, 2013.] <http://www.ecured.cu/index.php/OpenUp>.
- EcuRed.com.** EcuRed. [Online] [Cited: 3 25, 2013.] http://www.ecured.cu/index.php/Pruebas_de_caja_negra.
- EVA.** Entorno Visual de Aprendizaje. [Online] [Cited: 2 12, 2013.] http://eva.uci.cu/file.php/161/Documentos/Materiales_basicos/Materiales_basicos_de_la_Unidad_3/UML_y_Patrones/04_Parte_IV_Fase_del_Disenio_1_.pdf.
- EVA.** Tipos de prueba de Caja Blanca y Negra. [Online] [Cited: 3 20, 2013.] <http://eva.uci.cu/mod/resource/view.php?id=8649&subdir=/Comun>.
- Evangelina Fernández, Aída.** *FUNCIÓN CARACTERÍSTICA*. Depto de Matemática, Facultad de Ciencias Exactas y Tecnología, Universidad Nacional de Tucumán.
- Formación online en Nuevas Tecnologías.** Formación online en Nuevas Tecnologías. [Online] 3 2012, 17. [Cited: 2 21, 2013.] <http://www.digitallearning.es/blog/apache-servidor-web-configuracion-apache2-conf/>.
- Fowler, Martin.** *UML Distilled 3era edición*. s.l. : Addison-Wesley, 2003.
- González C., Benjamín.** desarrolloweb.com. [Online] 07 07, 2004. [Cited: 12 09, 2012.] <http://www.desarrolloweb.com/articulos/1557.php>.
- Gorunescu, Florin.** *Intelligent Systems Reference Library Volume 12*. Springer-Verlag Berlin Heidelberg, 2011. 978-3-642-19720-8.
- HackTimes.** Seguridad en Web Services (WS). [Online] 06 27, 2011. [Cited: 12 12, 2012.] http://www.hacktimes.com/seguridad_en_web_services/.
- Ingeniería del Software*. Madrid : Pearson, 2005. 84-7829-074-5.
- Institute for Statistics and Mathematics of the WU Wien.** The R Project for Statistical Computing. [Online] [Cited: 12 6, 2012.] <http://www.r-project.org/>.

- kioskea.net.** Lenguajes de programación. [Online] [Cited: 12 6, 2012.]
<http://es.kioskea.net/contents/langages/langages.php3>.
- Larman, Craig.** *UML y Patrones. 2ª Edición.* Madrid : Prentice Hall, 2003. 8420534382.
- LibrosWeb.** LibrosWeb. [Online] [Cited: 12 07, 2012.]
http://www.librosweb.es/symfony_1_1/capitulo_2/el_patron_mvc.html.
- LIBROSWEB.** LIBROSWEB. [Online] [Cited: 12 8, 2012.]
http://librosweb.es/symfony_1_2/capitulo_6/el_controlador_frontal.html.
- MaestrosWeb.** MaestrosWeb. [Online] [Cited: 12 09, 2012.]
<http://www.maestrosdelweb.com/editorial/curso-symfony2-introduccion-instalacion/>.
- Marques de Sá, Joaquim P.** *Applied Statistics Using SPSS, STATISTICA, MATLAB and R.* New York: Springer-Verlag Berlin Heidelberg, 2007. 978-3-540-71971-7.
Mathematical Methods of Statistics. 1951.
- MeRinde.net.** MeRinde. [Online] [Cited: 3 12, 2013.]
http://merinde.net/index.php?option=com_content&task=view&id=495&Itemid=291.
- Muller, Keith E.** *Understanding Canonical Correlation Through the General Linear Model and Principal Components.* The American Statistician, 1982.
- Muñoz, David Ruiz.** *Manual de Estadística.* s.l. : eumed.net, 2004. 84-688-6153-7.
- NetBeans.org.** NetBeans. [Online] [Cited: 12 09, 2012.] <http://netbeans.org/features/php/index.html>.
- OpenCourseWare de la Universidad de Sevilla.** OpenCourseWare de la Universidad de Sevilla. [Online] 9 16, 2008. [Cited: 2 20, 2013.] http://ocwus.us.es/metodos-de-investigacion-y-diagnostico-en-educacion/analisis-de-datos-en-la-investigacion-educativa/Bloque_I/page_03.htm/.
- Potencier, Fabien.** Fabien Potencier. [Online] 3 26, 2009. [Cited: 2 12, 2013.]
<http://fabien.potencier.org/article/11/what-is-dependency-injection>.
- ProgramacionDesarrollo.** ProgramacionDesarrollo. [Online] 02 05, 2011. [Cited: 12 06, 2012.]
<http://programaciondesarrollo.es/ventajas-de-php/>.
- PruebasDeSoftware.com.** PruebasDeSoftware. [Online] [Cited: 3 12, 2013.]
<http://pruebasdesoftware.com/laspruebasdesoftware.htm>.
- Rivas, José Manuel.** Codigonexo. [Online] 4 3, 2013. [Cited: 4 10, 2013.]
<http://www.codigonexo.com/definiciones/framework/>.
- Roberto Hernando.** Metodologías de desarrollo de software. [Online] [Cited: 12 08, 2012.]
http://www2.rhernando.net/modules/tutorials/doc/ing/met_soft.html.
- Rodríguez Archilla, Pablo.** *Web services y arquitecturas orientadas al servicio.* Universidad de Granada, 2010.
- Rodríguez Coma, Magdalena.** Secretaría de Descentralización. [Online] [Cited: 12 09, 2012.]
<http://sd.pcm.gob.pe/contenido/902/Eje%208.%20Prog%20Inform%20Eval%20Cuant%20-%20Curso%20SAS%20Doc1%20-%20Prof%20Rod.pdf>.
- Ruiz Muñoz, David.** Biblioteca Virtual. [Online] [Cited: 12 10, 2012.]
<http://www.eumed.net/cursecon/libreria/drm/0.htm>.
- Rumbaugh, James, Jacobson, Ivar and Booch, Grady.** *El Lenguaje Unificado de Modelado.* Madrid : Addison Wesley, 2000. 84-7829-037-0.

- Ruz, Fidel Castro.** *Seminario Nacional para Educadores (VI)*. La Habana : Pueblo y Educación, 2005.
- Salas, Christian.** Ecología austral. [Online] 6 19, 2008. [Cited: 2 21, 2013.]
http://www.scielo.org.ar/scielo.php?pid=S1667-782X2008000200007&script=sci_arttext.1667-782X.
- Sommerville, Ian.** *Ingeniería de Software. 6ta. Edición*. s.l. : Addison Wesley, 2012.
- Symfony.es.** Symfony. [Online] [Cited: 12 09, 2012.]
<http://www.symfony.es/noticias/2010/02/17/sflive2010-asi-es-symfony-2/>.
- Yang, Michael.** OpenUP. [Online] [Cited: 12 6, 2012.] <http://epf.eclipse.org/wikis/openup/>.

ANEXOS

 **UCI**
Universidad de las Ciencias
Informáticas

**Centro de Tecnologías de Gestión de Datos
DATEC**

La Habana, 16 de Mayo del 2013
"Año 54 de la Revolución".

ACTA DE ACEPTACIÓN

De una parte, el Centro de Tecnologías de Gestión de Datos, en lo sucesivo DATEC, de la Universidad de las Ciencias Informáticas, representado en este acto por: Abel's Juan Abreu Medina, y de otra parte los estudiantes: Sandy E. Noa Rodríguez y Julio G. Hernández Gómez.

Primero: Que en cumplimiento de los requisitos funcionales han sido efectuadas las implementaciones correspondientes.

CONSIDERANDO: Que los hitos realizados han sido desarrollados con la calidad requerida y bajo las condiciones pactadas y aprobadas por **Las Partes**.

CONSIDERANDO: Que los hitos que se han ejecutado cumplen con los requerimientos establecidos.

POR TANTO: **Las Partes** acuerdan formalizar mediante la presente Acta, la aceptación del producto: Módulo de análisis estadístico en su versión 2.0 para el R Server.

Y para que así conste, se extiende la presenta **Acta** en dos (2) ejemplares, rubricados por **Las Partes**.

Entregan



Recibe

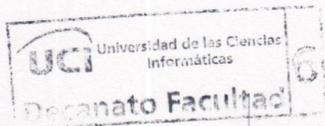


Ilustración 13. Acta de aceptación