

**Universidad de las Ciencias Informáticas
Facultad 6**



**Título: Diseño e implementación del módulo de estadística
avanzada para la plataforma alasGRATO**

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

Autor(es): Marisel Valdés Arango

Elisa Fonseca Torrente

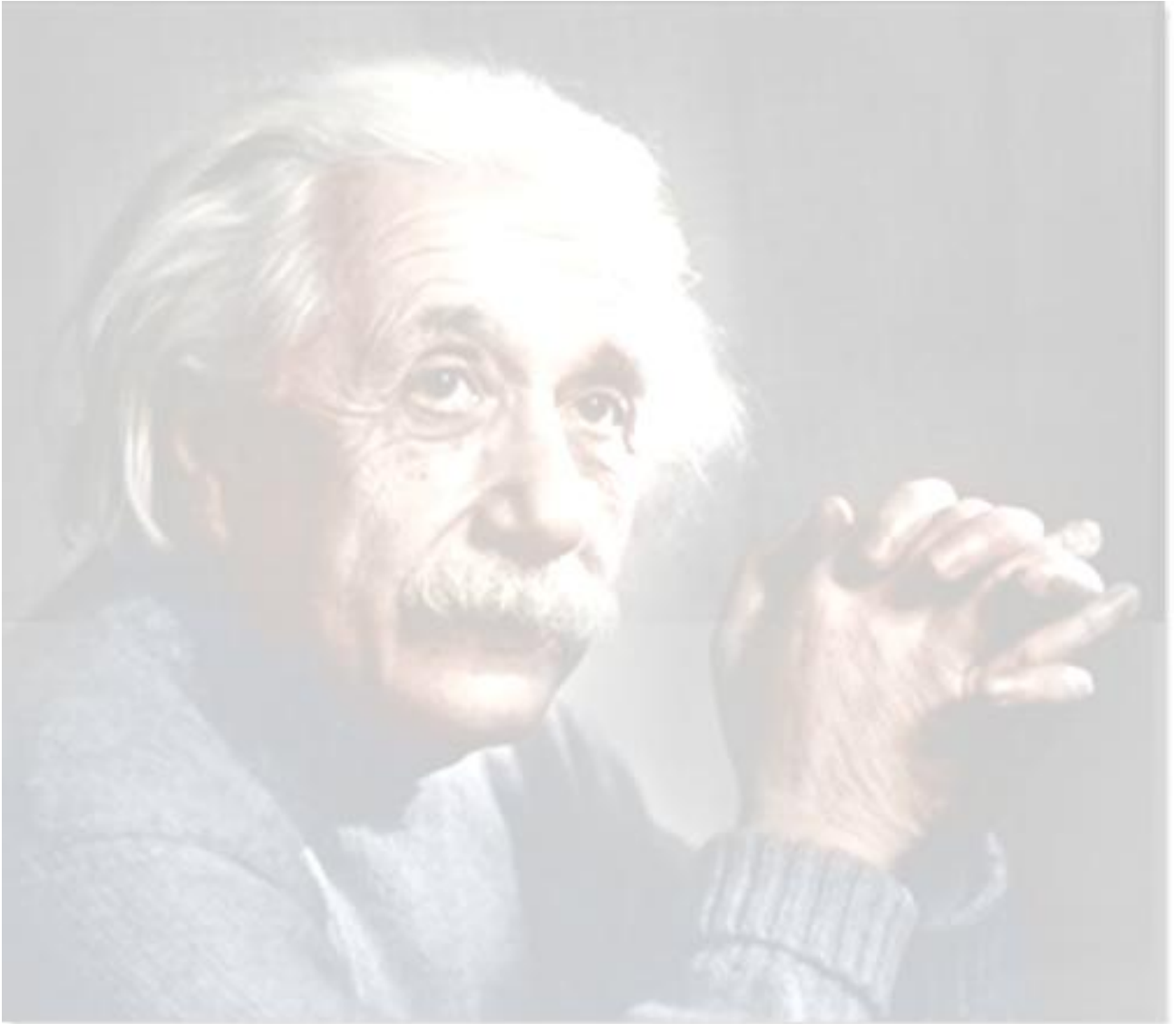
Tutor(es): Dr. Ramón Carrasco Velar

Ing. Tonyse De La Rosa Martín

Lic. Yunier E. Tejeda Rodríguez

Consultante: Diolisys Fontela González

“Julio de 2010”



“Uno debería guardarse contra aquellos que sermonean habitualmente a los jóvenes con la importancia del éxito como principal propósito en la vida. El estímulo más importante para el trabajo, en la escuela y en la vida, es el placer de trabajar, el placer de sus resultados, y el conocimiento del valor del resultado para la comunidad.”

Albert Einstein

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 _____.

Marisel Valdés Arango

Firma del Autor

Elisa Fonseca Torrente

Firma del Autor

Dr. Ramón Carrasco Velar

Firma del Tutor

Ing. Tonyse De La Rosa Martín

Firma del Tutor

Lic. Yunier Emilio Tejeda Rodríguez

Firma del Tutor

Ing. Dioletsys Fontela González

Firma del Consultante

DATOS DE CONTACTO

Tutores:

Dr. Ramón Carrasco Velar

Centro de Química Farmacéutica, Ciudad de La Habana, Cuba.

ramon.carrasco@cqf.sld.cu

rcarrasco@uci.cu

Ing. Tonyse De La Rosa Martín

Universidad de las Ciencias Informáticas, Ciudad de La Habana, Cuba.

tdelarosa@uci.cu

Lic. Yunier Emilio Tejeda Rodríguez

Universidad de las Ciencias Informáticas, Ciudad de La Habana, Cuba.

yuniere@uci.cu

Consultante:

Ing. Dioleisys Fontela González

Universidad de las Ciencias Informáticas, Ciudad de La Habana, Cuba.

dfontela@uci.cu

Agradecimientos:

A nuestros padres por guiarnos y apoyarnos en todo momento, por sus sabios consejos, por la confianza depositadas en nosotros.

A nuestras familias por brindarnos su apoyo y darnos su entera confianza de que si podíamos seguir adelante en los momentos más difíciles.

A nuestro pelu por siempre estar a nuestro lado, por ayudarnos a seguir adelante y depositar toda la confianza en nosotras de que este día si llegaría.

A nuestro Comandante en jefe Fidel Castro, a la Revolución y a la UCI por habernos dado la oportunidad de formarnos como ingenieros.

A todo lo que han hecho posible la elaboración de esta tesis y permitir que un día como hoy estemos aquí realizando nuestro sueño, en especial a Luis Gabriel, Roberto, Dairon, Edel, Félix y Edilberto.

A Dioleisys por su disposición permanente e incondicional, por su esmero, paciencia e infinita ayuda.

A nuestros amigos, por permitirnos disfrutar al máximo estos maravillosos cinco años a su lado, por ser partícipes de nuestros logros, desconciertos, malos y buenos momentos, marcando una etapa inolvidable en nuestras vidas: themura, salvi, julito, frometa, Silvio, yayi, kaly, ludmi, yili, sura, karen, yuya, imi, joana, anita, geo, yendis, osley, Arnaldo, yosmel y aliocha.

A las chicas y chicos del apartamento: alieķna, yanet, Tatiana, yumi, haymel, nela, mariela, yadi, osmay, yasi, jorge y pochy.

A todos los profesores que a lo largo de la carrera nos han brindado su apoyo y ayuda, en especial: al querido profesor Mariño, Sandy, Alberto, Anthony, nara y yuraisy.

A nuestros tutores.

A dios por permitir que hoy estemos aquí.

Dedicatoria:

Dedicatoria de Elisa:

En especial a mi madre que aunque no pudo estar aquí presente, sé que en este momento está pensando en mí, por darme su amor incondicional y por creer siempre en mí. Gracias mami chuli por ser la mejor madre del mundo. Te quiero mucho.

A mi papa por ayudarme y por quererme siempre.

A mi abuelo lu y más que eso mi padre.

A mi peluchino por toda su comprensión, amor, paciencia, por soportar mis malcriadeces, por darme fuerzas cuando más lo necesitaba, mi amor sin ti no hubiese podido llegar a ser quien soy hoy.

A mis tíos y más que eso mis padres por siempre ayudarme y creer en mí: papi lali, hermano cale y hermano miguel.

A mis tías por ser mis madres y hermanas: a tina, elisania, mersury y maría.

A mis hermanas: yisel (la flaquí) por sus consejos, por tenerme como su hermana a seguir, yare y yurima, las amo.

A mis primos que los adoro: rao, carlitin, fofi, pochy, daylet, yanet, ale, salet, odélito y piry.

A mi compañera de tesis chuchy por ser siempre mi amiga y mi hermana.

A mi amiga anicel por nuestra amistad de más de 7 años, por ser más que amiga hermana.

A mi otra familia mary y victor por acogerme como una hija más.

A mi suegra mercedes por sus consejos y su apoyo incondicional.

A mis amigos de la universidad que han sido los mejores amigos del mundo: salvi, frometa, julito, osmay, karen, temura, yayi, ludmi, kaly, sura, roxy, roberto, Dairon, luis, Silvio, alexey y geo.

A todos y todas los que me han apoyado y ayudado de una forma u otra a hacer posible este sueño.

Los quiero a todos(as).

Dedicatoria de Marisel:

A mis padres que son mi inspiración, por confiar en mí, darme la seguridad que permitió continuar mis estudios y realizar mis sueños, por su amor, dedicación y desvelo.

A mi hermano por apoyarme y estar siempre a mi lado.

A mis abuelos, tíos y primos por su colaboración.

A mis amigos y compañeros por soportarme estos cinco años: salvi, frometa, karen, temura, yayi, ludmi, kaly, sura, anita, yili, roberto, Dairon, luis y geo, gracias por su amistad sincera, por quererme y ayudarme en todo momento.

A los amigos y amigas que he conocido a lo largo de mi carrera: Haimel, Aliekna, Tatiana, Yanet, Mariela, Nela, Rosana, Darien, Manuel Alejandro, Arlhey, Felipe Yunier, Roilan, Hasiel Lincon, Yasiel, Silvio, Alvaro y otros que no están aquí pero ayudaron también.

Al piquete de la pandemia: Elieny, Yanara, Yadiurvis, Ana G, julito y su líder Imara, nos vemos en otras fiestas piquete.

A mi compañera de tesis, por su amor y apoyo durante estos cinco años, gracias por ser especial conmigo y compartir momentos agradables y momentos tristes.

Solo espero que todos siempre me tengan en su corazón y nunca olviden que pueden contar conmigo para lo que necesiten.

¡Gracias a todos!

RESUMEN

La Facultad 6 de la Universidad de Ciencias Informáticas desarrolla un proyecto denominado “Plataforma para la Predicción de Actividad Biológica en Compuestos Orgánicos” (alasGRATO). Esta plataforma cuenta con una base de datos de gran tamaño formada por moléculas y sus descriptores asociados, llegando al extremo en que puede haber más descriptores que individuos por lo que se encuentra elevada redundancia en la información que ellos contienen, por lo que se quiere integrar el módulo de estadística avanzada para reducir la multicolinealidad entre sus descriptores. En el presente trabajo de diploma se desarrolla una herramienta computacional para la predicción de actividad biológica en compuestos orgánicos a partir de descriptores moleculares basada en Regresión Mínimos Cuadrados Parciales (PLS) que se adhiera a la plataforma alasGRATO.

PALABRAS CLAVES

Actividad biológica, descriptores, multicolinealidad, Regresión de Mínimos Cuadrados Parciales (PLS).

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA SOBRE LA ESTADÍSTICA CLÁSICA Y AVANZADA	4
1.1 Aplicaciones de las técnicas estadísticas clásica y avanzada	4
1.2 Técnicas multivariadas	5
1.2.2 Análisis de Componentes Principales (ACP).....	5
1.2.3 Análisis de Discriminante Lineal (ADL).....	6
1.2.4 Método de Mínimos Cuadrados Parciales (PLS)	6
1.3 Fundamentación de metodología de desarrollo a utilizar	7
1.3.1 OpenUp/Basic.....	7
1.4 Roles desempeñados	8
1.4.1 Rol desempeñado por el Analista.....	8
1.4.2 Rol desempeñado por el Desarrollador	8
1.5 Artefactos generados	8
1.5.1 Artefactos a ser generados por el Analista	9
1.5.2 Artefactos a ser generados por el Desarrollador	9
1.6 Lenguajes de programación y herramientas utilizadas en el desarrollo de la solución	9
1.6.1 Lenguaje de programación	9
1.6.1.1 Java.....	9
1.6.2 Entorno Integrado de Desarrollo (IDE)	10
1.6.2.1 NetBeans.....	10
1.6.3 Herramientas.....	11
1.6.3.1 Herramienta CASE	11
1.6.3.1.1 Visual Paradigm para UML	11
1.6.3.2 Herramienta estadística	11
1.6.3.2.1 El R	11
1.6.3.3 Herramientas manejadoras de Plug-ins.....	12
1.6.3.1 Front-End GRATO	12
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA PROPUESTO	14
2.1 Modelo conceptual del sistema	14
2.1.1 Glosario de términos del dominio	14
2.2 Especificación de los requisitos del software	15
2.2.1 Requisitos funcionales.....	15
2.2.2 Requisitos no funcionales	15
2.3 Modelo de Caso de uso del sistema	16
2.3.1 Actores del sistema.....	16
2.3.2 Diagrama de caso de uso del sistema.....	17
2.3.3 Descripción textual de los Casos de Uso del sistema	18
2.3.3.1 Caso de Uso Cargar Fichero	18
2.3.3.2 Caso de Uso Realizar Predicción.....	18

2.3.3.4 Guardar Resultado	19
2.4 Conclusiones	20
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	21
3.1 Objetivos del diseño	21
3.2 Patrones utilizados en la solución del problema	21
3.2.1 Patrones de diseño	21
3.2.2 Patrones de arquitectura.....	23
3.3 Diagrama de clases del diseño	25
3.4 Diagramas de Interacción: secuencia	26
3.4.1 Diagramas de secuencia.....	26
3.5 Diagrama de despliegue	27
3.6 Conclusiones	28
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA	29
4.1 Diagrama de componentes	29
4.2 Solución en la implementación del problema	30
4.3 Pruebas del sistema	33
4.3.1 Configuración del entorno de pruebas.....	33
4.3.2 Diseño de las pruebas de unidad (caja negra).....	34
4.3.2.1 Casos de prueba.....	34
4.3.2.2 Pruebas de unidad (caja negra) para el caso de uso cargar fichero.....	34
4.4 Pantalla principal	36
4.5 Conclusiones	36
CONCLUSIONES	38
RECOMENDACIONES	39
REFERENCIAS BIBLIOGRÁFICAS	40
BIBLIOGRAFÍA	43
ANEXOS	50
GLOSARIO DE TERMINOS	52

Introducción

INTRODUCCIÓN

La bioinformática es una nueva disciplina que surge como respuesta al creciente aumento de los volúmenes de datos biológicos y estructurales, así como la facilidad de compartir esta información a través de Internet. Esta ciencia relaciona la biología, la bioquímica, la química, la farmacología, la informática y las tecnologías de la información con vista al análisis, organización y distribución de la información biológica. En tal sentido, ha facilitado grandes avances en las investigaciones biomédicas, por ejemplo: en el diagnóstico, tratamiento y prevención de diversas enfermedades, lo cual incide en una mejoría de la calidad de vida. Por ello el desarrollo de la industria farmacéutica guarda una estrecha relación con la bioinformática, la cual permite la búsqueda, cada vez más eficiente, de nuevos fármacos más específicos y con menores efectos secundarios, para el tratamiento de diversas patologías.

El proceso de obtención de nuevos fármacos es un proceso altamente costoso que requiere investigaciones de gran complejidad. En el pasado, el descubrimiento de moléculas candidatas a fármacos con determinada acción o efecto terapéutico, solía ser en la mayoría de los casos por azar, provocando en muchos casos reacciones adversas. [1]

En la actualidad, este proceso tiene un enfoque más seguro y diferente. Por lo general, se identifica primero la diana o blanco terapéutico y luego, a partir de técnicas disímiles se evalúan los compuestos candidatos.

Todo esto ha traído consigo grandes gastos y tiempo de espera, estimándose que el desarrollo completo de una molécula toma alrededor de 12-15 años y sólo 1 de cada 5.000 llega a ser un medicamento comercializable. A pesar de la gran cantidad de técnicas y tecnología moderna con la que cuenta la humanidad hoy, el tiempo de desarrollo de un fármaco sigue siendo elevado.

Según un artículo publicado por la Revista Médica de Chile, en 1960 el tiempo promedio de desarrollo era de 8,1 años contra los 15,3 años que tomó en 1992, aumentando también los gastos asociados al proceso. En 1978 el costo total de desarrollo era de \$54 millones de dólares; en 1990 era de \$231 millones y en 1997 llegó a los \$ 500 millones. [2]

Por esta razón, en los últimos años la industria farmacéutica ha reorientado sus investigaciones y prestado más atención a aquellos métodos que permitan una selección racional o diseño de nuevos compuestos con propiedades deseadas. Uno de los nuevos enfoques es la utilización de métodos computacionales que relacionan la estructura química con la actividad biológica.

Introducción

La gran mayoría de estos métodos computacionales, son muy costosos e inaccesibles para Cuba. Es por ello, que con el fin de garantizar la independencia tecnológica cubana, la Facultad 6 de la Universidad de las Ciencias Informáticas está desarrollando su propia plataforma para la predicción de actividad biológica, llamada Plataforma para la Predicción de Actividad Biológica de Compuestos Orgánicos (alasGRATO), la cual está implementada en Java para su uso multiplataforma.

Esta plataforma cuenta con bases de datos de gran tamaño formada por moléculas y sus descriptores asociados, llegando al extremo en que puede haber más descriptores que individuos. La generalidad de esos descriptores parte de formulismos que se basan en la matriz de conectividad de los vértices o aristas del grafo químico por lo que se encuentra elevada redundancia en la información que ellos contienen, es decir, hay multicolinealidad en los descriptores. Por ello se plantea como **problema científico** de la presente investigación: ¿cómo insertar técnicas de estadística clásica y avanzada en la Plataforma alasGRATO?

Teniendo como **objeto de estudio** las técnicas de estadística clásica y avanzada enmarcado en el **campo de acción** la regresión de mínimos cuadrados parciales aplicada al diseño de fármacos.

Con el fin de solucionar el problema planteado se definió como **objetivo general**: desarrollar una herramienta computacional para la predicción de actividad biológica en compuestos orgánicos a partir de descriptores moleculares basada en Regresión Mínimos Cuadrados Parciales (PLS) que se adhiera a la plataforma alasGRATO.

Entre los **objetivos específicos** propuestos se encuentran:

1. Diseñar una herramienta que contemple la regresión de mínimos cuadrados parciales (PLS).
2. Implementar la herramienta diseñada.
3. Implementar un *plug-in* para el módulo de estadística avanzada de la plataforma alasGRATO.

Para poder cumplir los objetivos y lograr una solución adecuada del problema científico se plantean las siguientes **tareas de la investigación**:

1. Estudio de los conceptos básicos de regresión de mínimos cuadrados parciales (PLS).
2. Elaboración del modelo de dominio correspondiente.
3. Definición de los requisitos funcionales y no funcionales.
4. Desarrollo del diagrama de caso de uso del sistema.
5. Descripción de los casos de uso correspondientes al sistema.

Introducción

6. Implementación de la herramienta.
7. Realización de las pruebas de validación.

El presente documento está estructurado en cuatro capítulos. En los que se describen los métodos y procedimientos a seguir para dar cumplimiento a los objetivos trazados. A continuación, una breve descripción de cada uno de ellos.

Capítulo 1: Fundamentación teórica sobre la estadística clásica y avanzada.

En este capítulo comprende el estudio y el análisis de los elementos fundamentales a tener en cuenta para la solución del problema planteado. Se argumenta la selección de las herramientas, metodología y el lenguaje de programación que se usarán para dar solución al problema científico y se definen los roles y artefactos generados.

Capítulo 2: Características del sistema propuesto.

Este capítulo describe la propuesta de solución para el problema científico de esta investigación, mostrando los procesos del negocio mediante el Modelo de Dominio. Se definen los requisitos funcionales y no funcionales de la herramienta, se muestran los actores del sistema y las acciones que realizan y se presenta el diagrama de caso de uso del sistema y las descripciones detalladas correspondientes a cada caso de uso.

Capítulo 3: Análisis y diseño del sistema.

En este capítulo se hace referencia al flujo de trabajo análisis y diseño, en el cual se describen las clases más importantes, además se muestran los diferentes diagramas de clases pertenecientes a cada uno de los casos de uso. También se realiza el modelo de despliegue de la aplicación.

Capítulo 4: Implementación y prueba del sistema.

Está enfocado a la implementación del sistema con el objetivo de darle solución a los requisitos funcionales. Se representan los diagramas de componentes de los CU. Se muestran algunas pantallas de la aplicación con sus descripciones y se representa todo lo relacionado con las pruebas realizadas al sistema.

Capítulo I

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA SOBRE LA ESTADÍSTICA CLÁSICA Y AVANZADA

En este capítulo comprende el estudio y el análisis de los elementos fundamentales a tener en cuenta para la solución del problema planteado. Se argumenta la selección de las herramientas, metodología y el lenguaje de programación que se usarán para dar solución al problema científico y se definen los roles y artefactos generados.

1.1 Aplicaciones de las técnicas estadísticas clásica y avanzada.

La estadística es una ciencia con base matemática referente a la recolección, análisis e interpretación de datos, que busca explicar condiciones regulares en fenómenos de tipo aleatorio.

En la actualidad se aplica en diferentes ciencias ya que su campo es bastante amplio; ejemplo en las ciencias sociales, ciencias naturales, problemas industriales o empresariales tales como estudio del mercado y control de calidad, censo de población, estudios del comportamiento humano, economía, finanzas, el comercio, la educación, en la biología y la medicina.[3]

La estadística comenzó a ser empleada en la experimentación médica, se suele citar como primer ensayo clínico aleatorio, el efectuado por el Consejo Británico de Investigación Médica en 1946, sobre el uso de la estreptomina en el tratamiento de la tuberculosis pulmonar. [4]

También en América la metodología estadística comenzaba su exitosa carrera en el mundo de la investigación médica. En 1954 se llevó a cabo el mayor y más costoso estudio médico jamás realizado, para evaluar la eficacia de la vacuna de Salk como protección contra la poliomielitis. Puesto que la incidencia anual de polio era sólo del orden de 1/2000, fue necesaria la participación de alrededor de 2 millones de niños, con un coste superior a 5 millones de dólares, para comprobar definitivamente la eficacia de la vacuna. Después de la segunda guerra mundial los avances recientes en física, economía y telecomunicaciones, abrieron nuevos campos para la utilización de técnicas estadísticas, entre las que cabe destacar el análisis de series temporales, la teoría de juegos, teoría del caos y redes neuronales. [5]

En Cuba el uso de las técnicas estadísticas son aplicadas en diversas ramas de la ciencia ejemplo en la medicina para las investigaciones epidemiológicas prevaleciendo en los factores de riesgo del cáncer, el comportamiento de los mismos constituye la base científica para brindar medidas preventivas, así como establecer acciones de salud encaminadas a disminuir el riesgo de la población. Para este análisis se aplicó una encuesta en las 14 provincias del país, donde se obtuvo información de la

Capítulo I

prevalencia de algunos factores demográficos y factores de riesgo. Los métodos multivariados permitieron analizar todos los factores en conjunto y diferenciar a la población por sus hábitos de vida. [6]

La aplicación de las técnicas estadísticas multivariadas están presentes en distintas áreas o ramas de las ciencias, por ejemplo en: investigación de mercados (para identificar características de los individuos con el propósito de determinar qué tipo de personas compran determinado producto); en el sistema de educación de cualquier tipo de especialidad (para conocer los estudiantes que tendrán éxito y concluirán satisfactoriamente sus estudios); en la agricultura (al estudiar la resistencia de determinado tipo de cosechas a daños por plagas y sequías); en el deporte (para conocer a partir de medidas antropométricas las posibilidades de obtener buenos resultados en un deporte específico); en la psicología (al estudiar la relación entre el comportamiento de adolescentes y actitudes de los padres) y en la economía (para conocer el nivel de desarrollo de un territorio en relación con otros y realizar inferencias a partir de variables económicas fundamentales).[7]

1.2 Técnicas multivariadas

Los métodos multivariados son un conjunto de técnicas que permiten interpretar y visualizar conjuntos grandes de datos (tanto en individuos como en variables), a partir de su simplificación o reducción. Tienen como objetivo encontrar relación entre variables e individuos. [8]

1.2.2 Análisis de Componentes Principales (ACP)

El objetivo principal que persigue el ACP es la representación de las medidas numéricas de varias variables en un espacio de pocas dimensiones, permite reducir la dimensionalidad de los datos, transformando el conjunto de p variables originales en otro conjunto de q variables no correlacionadas llamadas componentes principales. Entre sus usos más frecuentes esta reducir la dimensionalidad de la matriz de datos con el fin de evitar redundancias y destacar relaciones, como técnica de análisis exploratorio permite descubrir interrelaciones entre los datos y de acuerdo con los resultados.

Existen dos formas básicas de aplicar el ACP:

- **Método basado en la matriz de covarianzas**, que se usa cuando los datos son dimensionalmente homogéneos y presentan valores medios similares.

Capítulo I

- **Método basado en la matriz de correlación**, cuando los datos no son dimensionalmente homogéneos o el orden de magnitud de las variables aleatorias medidas no es el mismo.

1.2.3 Análisis de Discriminante Lineal (ADL)

Es una técnica que persigue uno de los siguientes objetivos:

- Describir las diferencias existentes entre esos grupos sobre la base de los valores que toman ciertas variables sobre los individuos de cada uno de los grupos.
- Clasificar nuevos individuos en alguno de los grupos preexistentes en función de los valores que toman ciertas variables para esos individuos.

Técnicamente, podemos decir que el análisis discriminante tratará de encontrar funciones de X variables cuyos valores separen o discriminen lo más posible a los grupos existentes. Estas funciones, denominadas funciones o ejes discriminantes, serán combinaciones lineales de las variables originales de la forma:

$$Y = a_0 + a_1X_1 + a_2X_2 + \dots + a_pX_p$$

1.2.4 Método de Mínimos Cuadrados Parciales (PLS)

El objetivo que persigue es explicar la relación entre dos conjuntos de variable X y Y , así como la variabilidad tanto en X como en Y , maximizando la covarianza entre ellos.

Este método cuenta con el algoritmo estándar para calcular las componentes latentes que es Lineal Iterativo de Mínimos Cuadrados Parciales (NIPALS) el cual cuenta con 8 pasos:

$$1. t_j = \sum_{m=1}^M x_m w_{mj} = Xw_j$$

$$2. u_j = \sum_{k=1}^K y_k c_{kj} = Yc_j$$

$$3. b_j = \frac{u'_j t_j}{u'_j u_j}, \quad j = 1, \dots, a$$

$$4. \hat{u}_j = t_j b_j, \quad j = 1, \dots, a$$

Capítulo I

$$5. p'_j = \frac{t'_j X}{t'_j t_j}, \quad j = 1, \dots, a$$

$$6. q'_j = \frac{\hat{u}'_j Y}{\hat{u}'_j \hat{u}_j}, \quad j = 1, \dots, a$$

$$7. E = X - t_j p'_j, \quad j = 1, \dots, a$$

$$8. F = Y - \hat{u}_j q'_j, \quad j = 1, \dots, a$$

En el algoritmo NIPALS se extraen secuencialmente las variables latentes como una combinación lineal de las variables de entrada y salida, se determinan los coeficientes de la regresión lineal $u_j = t_j b_j + e_j, j = 1, \dots, a$ entre cada par de vectores *scores* t y u correspondientes, obteniendo la regresión estimada. Luego los vectores *loadings* de entrada y de salida son calculados de la regresión de la matriz X sobre t_j y la matriz Y sobre \hat{u}_j respectivamente, se actualizan las matrices predictivas y respuestas respectivamente. [9]

1.3 Fundamentación de metodología de desarrollo a utilizar

Una metodología de desarrollo es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo producto de *software*. Una metodología representa el camino para desarrollar *software* de una manera sistémica. Estas persiguen tres necesidades principales:

- Mejores aplicaciones, que conduzcan a una mejor calidad.
- Un proceso de desarrollo controlado.
- Un proceso normalizado en una organización, no dependiente de un personal.

La metodología a utilizar en el desarrollo del presente trabajo es *OpenUp/Basic* según se definió en la arquitectura definida para el proyecto *alasGRATO*. [10]

1.3.1 OpenUp/Basic

Conserva las características principales del modelo de desarrollo *RUP*, incluye el desarrollo iterativo, permite identificar los requisitos operacionales del sistema, prever las interacciones con los usuarios y prevenir los posibles riesgos en el desarrollo del sistema. Es una forma de desarrollo ágil y ligera, consiste en equipos a los cuales se les asigna una fase del desarrollo que tienen que complementarse

Capítulo I

entre sí para obtener un buen producto final. No puede ser una sola persona la que realice todo el trabajo pues esto podría ocasionar que se pierda de vista ciertas características importantes, por ejemplo para un proyecto pequeño constituyen equipos de tres a seis personas e implican tres a seis meses de esfuerzo del desarrollo. Describe el conjunto mínimo de roles, y para cada uno un conjunto de tareas a realizar y artefactos que obtendrán en el desarrollo del *software*. [11]

1.4 Roles desempeñados

De acorde a la metodología seleccionada para el desarrollo de la solución del sistema, se definieron los roles a ser desempeñados partiendo de los roles que propone la misma. Entre ellos se encuentra:

1.4.1 Rol desempeñado por el Analista

Según *OpenUp*, el Analista necesita poseer experiencia en la identificación y comprensión de los problemas y oportunidades, así como la capacidad de articular las necesidades que están asociadas con el problema clave a resolver o la oportunidad de ser realizado. Además, debe tener conocimiento de los ámbitos de negocio y la tecnología o la capacidad de absorber y comprender rápidamente esa información. Este rol puede ser asignado de la siguiente manera:

Uno o más miembros del equipo realizan esta función exclusivamente. Este enfoque comúnmente adoptado es adecuado para las necesidades complejas que son difíciles de reunir. [12]

1.4.2 Rol desempeñado por el Desarrollador

Es el responsable para el desarrollo de una parte del sistema, incluido diseñarlo para ajustar en la arquitectura posibles prototipos de interfaz de usuario y, a continuación, la implementación, la unidad de pruebas, y la integración de los componentes que forman parte de la solución. Según *OpenUp*, el desarrollador necesita conocimientos técnicos y experiencia suficiente para definir y crear soluciones técnicas dentro del proyecto. Debe tener la capacidad de comprender y ajustarse a la arquitectura, de identificar y construir las pruebas de desarrollador que cubran el requerido comportamiento de los componentes técnicos.

1.5 Artefactos generados

En dependencia de la metodología y los roles definidos a ser utilizados en el desarrollo de la solución del sistema se generan los siguientes artefactos:

Capítulo I

1.5.1 Artefactos a ser generados por el Analista

- Requerimientos del sistema.
- Casos de Uso.
- Modelo de Casos de Uso. [13]

1.5.2 Artefactos a ser generados por el Desarrollador

- Estructura.
- Diseño.
- Implementación.

1.6 Lenguajes de programación y herramientas utilizadas en el desarrollo de la solución

Para el desarrollo de una aplicación es sumamente importante una buena selección del lenguaje de programación y de las herramientas a utilizar ya que determinan el tiempo de desarrollo, los costos y el resultado general de todo el proceso.

Las herramientas y el lenguaje de programación a utilizar en el desarrollo del presente trabajo son *Visual Paradigm* como herramienta CASE, R como herramienta estadística, *NetBeans* como entorno integrado de desarrollo y JAVA como lenguaje de programación según se definió en la arquitectura del proyecto alasGRATO.

1.6.1 Lenguaje de programación

Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; permitiendo al desarrollador comunicarse con los dispositivos de *hardware* y *software* existentes.[14]

1.6.1.1 Java

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. Hace uso de la definición de entidades formadas por métodos y variables que reciben el nombre de clases.

Sus principales características son:

Orientado a objetos: soporta las características esenciales del paradigma de la programación orientado a objetos: encapsulamiento, herencia y polimorfismo.

Capítulo I

Robusto: elimina el uso de apuntadores para referenciar áreas de memoria, además libera al desarrollador de la necesidad de desalojar la memoria que la aplicación ya no usa. Además, requiere la declaración explícita tanto de los tipos de datos como de los métodos.

Multiplataforma: el mismo código Java que funciona en un sistema operativo, funciona en cualquier otro sistema operativo que tenga instalada la máquina virtual de Java.

Multitareas: a pesar de que las capacidades multitareas que pueden ser implementadas en Java dependen en gran parte del sistema operativo en el cual se ejecuten, dígase Windows o Unix, dichas capacidades superan en gran manera a los entornos de flujo único que ofrecen otros lenguajes de programación. Permite la ejecución concurrente de varios procesos ligeros o hilos de ejecución. [15]

1.6.2 Entorno Integrado de Desarrollo (IDE)

Un IDE es un programa compuesto por un conjunto de herramientas para un desarrollador que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). [16]

1.6.2.1 NetBeans

NetBeans IDE es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. [17]

Además, es un producto libre y gratuito sin restricciones de uso. NetBeans es una aplicación de código abierto ("open source") diseñada para el desarrollo de aplicaciones fácilmente portables entre las distintas plataformas, haciendo uso de la tecnología Java. Dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web, control de versiones, colaboración entre varias personas, creación de aplicaciones compatibles con teléfonos móviles, resaltado de sintaxis y por si fuera poco sus funcionalidades son ampliables mediante la instalación de packs. [18]

Capítulo I

1.6.3 Herramientas

1.6.3.1 Herramienta CASE

Las herramientas CASE son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de *software* reduciendo el coste de los mismos en tiempo y dinero. Son usadas en algunas fases del desarrollo de sistemas de información, incluyendo análisis, diseño y programación. [19]

1.6.3.1.1 Visual Paradigm para UML

Es una herramienta de modelado que emplea UML, diseñada para apoyar a los arquitectos de sistema, desarrolladores y diseñadores para acelerar el proceso de análisis y diseño de aplicaciones complejas. Además, soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, ingeniería inversa, generar código desde diagramas y generar documentación.

También ofrece un diseño centrado en casos de uso y enfocado al negocio que genera un *software* de mayor calidad, el uso de un lenguaje estándar común para todo el equipo de desarrollo que facilita la comunicación, un modelo y código que permanece sincronizado en todo el ciclo de desarrollo. Disponibilidad en múltiples plataformas y versiones, así como de integrarse en los principales IDE. [20]

1.6.3.2 Herramienta estadística

1.6.3.2.1 El R

R es un paquete estadístico de última generación al mismo tiempo que un lenguaje de programación para análisis estadístico y gráfico, lo cual lo hace muy versátil.

Dentro de los lenguajes de programación se puede clasificar como un lenguaje orientado a objetos de tipo interpretado. Una de las ventajas que trae consigo es la gran flexibilidad, una gran potencia y un tiempo de aprendizaje corto. Actualmente se encuentran disponibles más de 800 paquetes los cuales se agrupan en vistas, es decir, según la función que realizan, proporciona un amplio abanico de herramientas estadísticas (modelos lineales y no lineales, análisis de series temporales, algoritmos de clasificación y agrupamiento, etc.) y gráficas. [21]

Puede integrarse con distintas bases de datos y existen librerías que facilitan su utilización desde lenguajes de programación interpretados como Perl y Python.

Capítulo I

Otra de las características de R es su capacidad gráfica, que permite generar gráficos con la calidad. R posee su propio formato para la documentación basado en LaTeX.

También puede usarse como herramienta de cálculo numérico, campo en el que puede ser tan eficaz como otras herramientas específicas tales como GNU Octave y su versión comercial, MATLAB. Se ha desarrollado una interfaz, RWeka para interactuar con Weka que permite leer y escribir ficheros en el formato arff y enriquecer R con los algoritmos de minería de datos de dicha plataforma. [22]

1.6.3.3 Herramientas manejadoras de Plug-ins

Un *plug-in* es un módulo de *hardware* o *software* que añade una característica o un servicio específico a un sistema más grande, es decir, es una aplicación que puede incorporarse a otra para extender las funcionalidades de ese sistema.[23]

El uso de *plug-ins* se ha convertido en la actualidad en una gran tendencia para todos en el mundo. Los grandes productores de *software* realizan aplicaciones que sean capaces de tener soporte para *plug-in* con el fin de ampliar su aplicación y convertirla en todo un éxito para el mundo digital.

Existen múltiples herramientas que permiten desarrollar *plug-ins*. Estas tienen su funcionalidad principal, pero el usuario les puede ir incorporando otras mediante *plug-ins* ya creados. A continuación se mencionara una de esas herramientas.

1.6.3.1 Front-End GRATO

Surgió como una necesidad del Proyecto alasGRATO en el departamento de Bioinformática de la Facultad 6 de la Universidad de las Ciencias Informáticas para mejorar la interacción de los usuarios con la plataforma. El Front-End GRATO es dinámico y multiplataforma, y se encarga de la portabilidad de los *plug-ins* garantizando:

- Unificar la carga de los *plug-ins* a través de descriptores XML.
- Ganar en el manejo de memoria en ejecución controlando todos los componentes visuales desde un mismo marco.
- Mantener el principio de conservación, pues en la evolución del Front-End, los *plug-ins* desarrollados siempre serán compatibles. [24]

Debido a todo lo planteado se escogió al Front-End GRATO como la herramienta manejadora de *plug-ins* a utilizar para el desarrollo de los *plug-ins* para la integración del módulo de estadística clásica y avanzada.

Capítulo I

1.7 Conclusiones

En este capítulo se realizó un estudio de la estadística clásica y avanzada, así como el análisis de los diferentes métodos estadísticos, decidiéndose hacer uso del método: PLS, por sus ventajas para el desarrollo de la solución. Se decidió utilizar como metodología de desarrollo OpenUp, Visual Paradigm como herramienta CASE, como herramienta estadística el R y como herramienta manejadora de *plugin* el Front-End GRATO, como lenguaje de programación JAVA y el entorno integrado de desarrollo NetBeans. Sobre la base de la metodología elegida, se expuso como roles a desempeñar el de Analista a partir del cual se obtendrán los artefactos Requerimientos del sistema, Casos de Uso y el Modelo de Casos de Uso y el rol de Desarrollador el cual generará los artefactos Estructura, Diseño e Implementación.

Capítulo II

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA PROPUESTO

Este capítulo describe la propuesta de solución para el problema científico de esta investigación, mostrando los procesos del negocio mediante el Modelo de Dominio. Se definen los requisitos funcionales y no funcionales de la herramienta, se presenta el diagrama de caso de uso del sistema y las descripciones detalladas correspondientes a cada caso de uso.

2.1 Modelo conceptual del sistema

El modelo conceptual es una representación visual de los conceptos u objetos del mundo real significativo para un problema o área de interés. Representa clases conceptuales del dominio del problema. Representa conceptos del mundo real, no de los componentes de *software*. [25]

Los casos de uso son una herramienta para el análisis de requerimientos, pero realmente no están orientados a objetos.

Se representa en UML con un diagrama de clases en el que se muestra conceptos u objetos del dominio del problema (clases conceptuales), asociaciones entre las clases conceptuales y atributos de estas. No muestra comportamiento y las clases conceptuales no pueden tener métodos. Una clase conceptual puede ser una idea o un objeto físico (símbolo, definición y extensión).

2.1.1 Glosario de términos del dominio

Especialista: persona capacitada para interactuar con la aplicación.

Predicción: acción que el sistema realiza para emitir un resultado el cual brinda la opción de realizar predicción por PLS.

PLS: interfaz donde se muestra la selección de las variables y opciones estadísticas para realizar la predicción por mínimos cuadrados parciales.

Fichero: objeto donde se guarda la información solicitada, el cual se encuentra ubicado en un repositorio.

Resultado: interfaz donde se muestra los resultados obtenidos ya sea numérico o gráfico.

Capítulo II

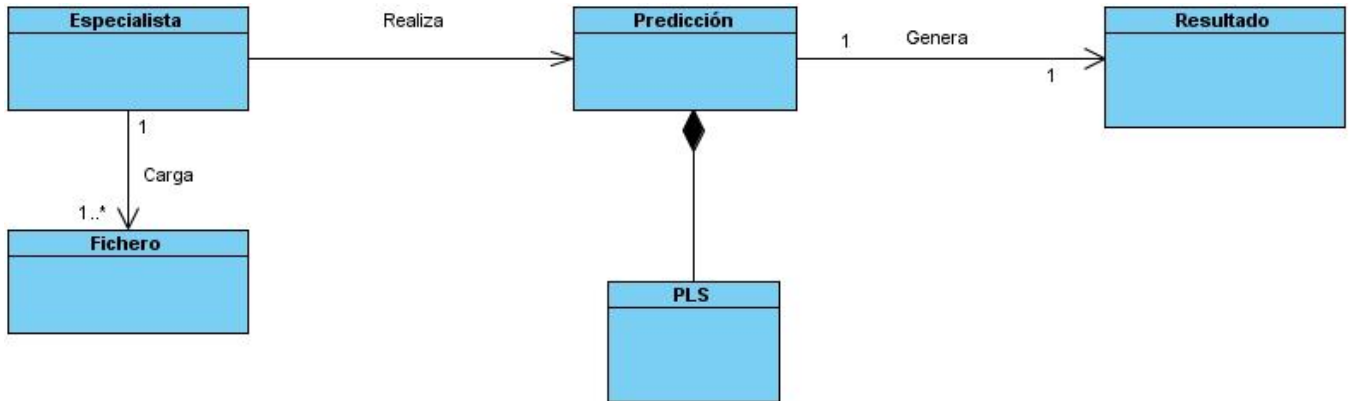


Figura 2.1: Modelo conceptual del sistema.

2.2 Especificación de los requisitos del software

El objetivo de la captura de requisitos es guiar el desarrollo de *software* hacia el sistema correcto, definiendo objetivos generales concretos de manera tal que tanto el negocio como los actores se beneficien. Los requisitos han sido descritos como una característica del sistema a entregar, especificados por escrito, posibles de probar o verificar, lo más abstracto y conciso posible para evitar malas interpretaciones. Se pueden clasificar en funcionales y no funcionales.

2.2.1 Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. La aplicación a desarrollar a través de la realización del presente trabajo debe cumplir los siguientes requisitos:

RF1 Cargar Fichero.

RF2 Realizar Predicción.

RF3 Guardar Resultado.

2.2.2 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. La aplicación a desarrollar será un módulo del proyecto productivo alasGRATO, el cual tiene su arquitectura definida, por lo que debe poseer los requerimientos no funcionales definidos en ella:

1. Requisitos de usabilidad

1.1 La interfaz de usuario final debe permitir ser utilizada por usuarios con pocos o ningún conocimiento informático.

2. Requisitos de software

Capítulo II

2.1 Máquina virtual de Java 1.6 o superior.

2.2 Se debe disponer del sistema operativo Linux (Ubuntu).

2.3 PC con instalación del programa estadístico R.

3. Requisitos de hardware

3.1 Se requiere un mínimo de 512 Mb de RAM, 1Gb recomendado.

3.2 PC con Microprocesador Pentium IV a 2.41 GHz o superior.

3.3 PC con 160 GB de disco duro.

4. Requisitos de apariencia e interfaz externa

4.1 La interfaz de usuario final debe ser amigable y sencilla.

5. Requisitos de portabilidad

5.1 La aplicación debe correr en el sistema operativo Linux.

6. Requisitos de seguridad

6.1 Confidencialidad:

Se requiere que los módulos que se le adicionen a la plataforma sean reconocidos por el servidor encargado de ofrecer los servicios, para que los mismos puedan funcionar. De ocurrir lo contrario el usuario podrá utilizar el módulo, pero no los servicios que brinda.

6.2 Integridad:

Cada módulo debe ser adicionado y eliminado de la plataforma en tiempo de ejecución. Asimismo, deberá garantizarse la funcionalidad independiente de estos, aunque exista dependencia entre ellos.

7. Requisitos de soporte

El sistema debe propiciar su mejoramiento y la incorporación de otras opciones en un futuro.

2.3 Modelo de Caso de uso del sistema

El modelo de caso de uso del sistema contiene actores, caso de uso y sus relaciones.

2.3.1 Actores del sistema

Un actor es la idealización de una persona externa, un proceso o cosa que interactúa con un sistema, subsistema, o clase. Caracteriza a las interacciones que pueden tener los usuarios con el mismo. Se puede incluir que un usuario puede llegar a desempeñarse como varios actores en tiempo de ejecución, además que diferentes usuarios pueden estar reflejados en un mismo actor, y por lo tanto,

Capítulo II

representan distintas instancias de la misma definición de actor. Cada actor participa en uno o más casos de uso. [26]

Actor del sistema	Descripción
Especialista	Representa al usuario que va a hacer uso del sistema, y quien tiene la posibilidad de interactuar con todas las funcionalidades de este.

Tabla 1: Actor del Sistema

2.3.2 Diagrama de caso de uso del sistema

Un diagrama de casos de uso del sistema contiene actores, casos de uso del sistema y las relaciones existentes entre los mismos. Este diagrama representa de forma gráfica como será concebido el sistema para una mejor comprensión. Refleja cómo interactúan los usuarios con el sistema. [27]

Los casos de uso definidos son:

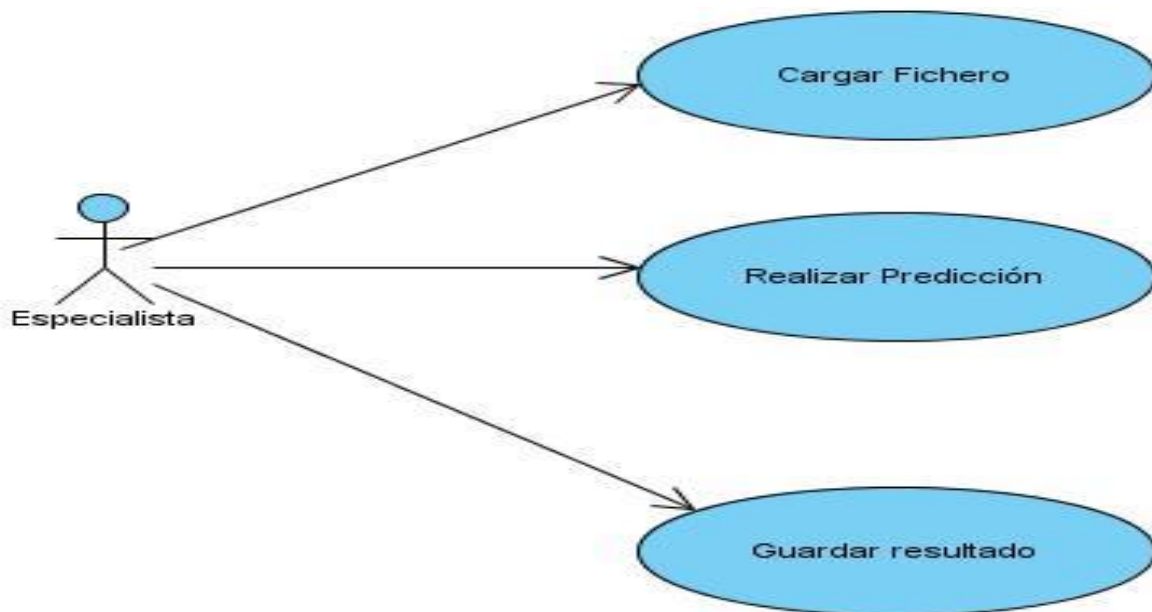


Figura 2.2: Diagrama de caso de uso del sistema.

Capítulo II

2.3.3 Descripción textual de los Casos de Uso del sistema

2.3.3.1 Caso de Uso Cargar Fichero

Caso de Uso:	Cargar Fichero
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el especialista va a realizar la opción cargar fichero.
Referencia:	RF1
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. El especialista selecciona la opción "open".	1.1 El sistema muestra una ventana de diálogo donde permite seleccionar el fichero que desea abrir desde un directorio específico.
2. El especialista selecciona el fichero que desea cargar y luego presiona el botón Abrir.	2.1 Si la estructura del fichero arff que se desea cargar es correcta, se visualiza el fichero en el área de visualización.
Flujo alternativo de eventos	
Acción del actor	Respuesta del sistema
	2.2 Si la estructura del fichero arff que se desea cargar es incorrecta, se muestra el siguiente mensaje: "error loading arff file".

Tabla 2: Descripción textual del Caso de uso: Cargar Fichero

2.3.3.2 Caso de Uso Realizar Predicción

Caso de Uso:	Realizar Predicción
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el especialista desea realizar la predicción por el método PLS.
Referencia:	RF2

Capítulo II

Precondiciones:	Debe de existir un fichero cargado en la aplicación.
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. El especialista selecciona la opción "PLS".	1.1 El sistema muestra una ventana con nuevas opciones. <ul style="list-style-type: none"> • "Variables" • "Kenel PLS" • "Nipals Algorithm" • "SIM PLS"
2. El especialista selecciona la opción "Variables"	2.1 El sistema muestra una ventana permitiendo realizar la selección de las variables.
3. El especialista selecciona las variables y la opción OK.	3.1 El sistema selecciona las variables satisfactoriamente.
4. El especialista realiza una de las opciones que brinda el sistema.	
	6. El sistema muestra las opciones de OK y cancel.
7. El especialista selecciona la opción OK.	7.1 El sistema emite un resultado.
Flujos alternos	
Acción del actor	Respuesta del sistema
7.1 El especialista selecciona la opción Cancel.	7.2 El sistema cancela lo anteriormente realizado.

2.3.3.4 Guardar Resultado

Caso de Uso:	Guardar Resultado
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el especialista desea guardar el resultado obtenido.
Referencia:	RF3

Capítulo II

Precondiciones:	Debe haberse realizado el caso de uso realizar predicción.
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. El especialista accede a la opción Save.	1.1 El sistema muestra una ventana de diálogo que permite seleccionar el lugar donde se quiere guardar el resultado obtenido.
2. El especialista selecciona donde desea guardar el resultado, escribe el nombre y presiona el botón Guardar.	2.1 El sistema guarda el resultado en el destino indicado con el nombre especificado.
Flujos alternos	
Acción del actor	Respuesta del sistema
2.1 El especialista selecciona la opción Cancel.	2.2 El sistema cancela lo anteriormente realizado.

2.4 Conclusiones

En el presente capítulo resultó concluida la fase de descripción del negocio, realizando una modelación del dominio por no poseer un negocio claramente definido. Se definieron tres requisitos funcionales los cuales se agruparon en tres casos de uso y siete requisitos no funcionales, necesarios para obtener un sistema eficiente, se definió el diagrama de caso de uso del sistema y se realizaron las descripciones de los casos de uso involucrados en el desarrollo de la aplicación.

Capítulo III

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

En el presente capítulo se desarrolla el flujo de trabajo del diseño, a través del cual se modela la aplicación que debe ser capaz de cumplir los requisitos antes analizados. Para ello se desarrollarán diferentes artefactos como los diagramas de clases del diseño y los diagramas de interacción, además se realizará la descripción de estilos arquitectónicos y patrones de diseño. Se define también la distribución del sistema mediante el modelo de despliegue.

3.1 Objetivos del diseño

El diseño tiene como principales objetivos comprender detalladamente los requisitos funcionales y no funcionales, sistemas operativos, tecnologías de distribución, restricciones relacionadas con el lenguaje de programación, tecnologías de interfaz de usuario. Además, crea un punto de partida para las actividades de implementación que siguen.

3.2 Patrones utilizados en la solución del problema

3.2.1 Patrones de diseño

Una solución a un problema que se usa repetidamente en contextos similares con algunas variantes en la implementación es a lo que se define como un patrón de diseño. Un patrón de este tipo identifica, abstrae y nombra los aspectos elementales de una estructura de diseño, donde los componentes, son las clases y objetos, y sus mecanismos de interacción son mensajes. Cada patrón prescribe una estructura de clases, sus roles y colaboraciones, y una adecuada asignación de métodos para resolver un problema de diseño en una manera flexible y adaptable. Ayudan a elegir diseños alternativos que hacen un sistema reutilizable y evitan alternativas que comprometan la reutilización. [28]

Los patrones empleados en la solución del problema fueron:

Singleton (Instancia única)

El patrón de diseño *Singleton* está pensado para restringir la creación de objetos de una clase determinada garantizando que sólo exista una instancia de dicha clase y proporcionar un punto de acceso global a ella. [29]

A continuación, en la figura 3.1 se muestran las clases que evidencian el empleo del patrón en la aplicación desarrollada.

Capítulo III

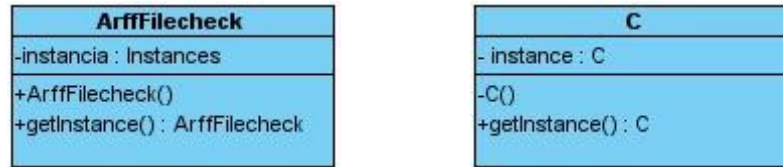


Figura 3.1: Ejemplos del patrón *Singleton* en la implementación de la herramienta computacional

GRASP

GRASP son patrones generales de *software* para asignación general de responsabilidades, es el acrónimo de “Asignación de Responsabilidad General Patrones de Software” (GRASP). De los cuales se utilizan en la solución del problema los siguientes:

Creador

Identifica quién debe ser el responsable de la creación o instanciación de nuevos objetos o clases. Se brinda un soporte al bajo acoplamiento.

La clase open contiene un objeto de la clase” ArffFilecheck”, por lo que el patrón creador sugiere que open es idónea para asumir la responsabilidad de crear las instancias de la clase “ArffFilecheck”.

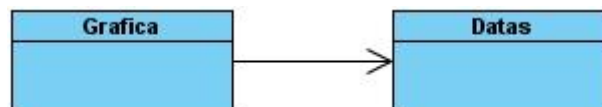


Figura 3.2: Ejemplo del patrón creador en la implementación de la herramienta computacional.

Controlador

Asigna la responsabilidad de controlar el flujo de eventos del sistema a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Le ofrece mayor potencial de los componentes reutilizables, garantizando que la empresa o los procesos de dominio sean manejados por la capa de los objetos de dominio y no por la de la interfaz. Reflexionar sobre el estado del caso de uso. [30]

Bajo acoplamiento

Es la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización y disminuyendo la dependencia entre las clases. Estas clases

Capítulo III

no se afectan por los cambios en otros componentes, fáciles de entender por separado y de fácil reutilización.

Puesto que la entidad “open” tiene la responsabilidad de cargar fichero esta incluye la clase “ArfftToModel”, esta clase terminara acoplándose al conocimiento de un objeto de la clase “C” para almacenar las instancias de las variables cargadas en la aplicación, así para que el experto “Open” pueda conocer la información de las variables cargadas. Este no necesariamente debe acoplarse a la clase “C” pues su acoplamiento con “ArfftToMode” hace que disponga d la información.

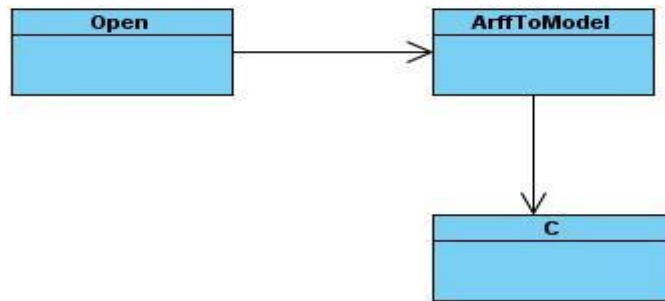


Figura 3.3: Ejemplo del patrón bajo acoplamiento en la implementación de la herramienta computacional.

3.2.2 Patrones de arquitectura

Los patrones de arquitectura expresan el esquema fundamental de organización para sistemas de *software*. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos. Los patrones de arquitectura representan el nivel más alto en el sistema de patrones. Ayudan a especificar la estructura fundamental de una aplicación. Cada actividad de desarrollo es gobernada por esta estructura; por ejemplo: el diseño detallado de los subsistemas, la comunicación y colaboración entre diferentes partes del sistema. Cada patrón de arquitectura ayuda a conseguir una propiedad específica en el sistema global. [30]

El patrón arquitectónico empleado en el presente trabajo de diploma es Modelo Vista Controlador donde:

- **Modelo:** Representa específicamente el dominio de la información sobre la cual funciona la aplicación. El modelo es otra forma de llamar a la capa de dominio. La lógica de dominio añade significado a los datos. El modelo encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.

Capítulo III

- **Vista:** Presenta el modelo en un formato adecuado para interactuar, usualmente con un elemento de interfaz de usuario. Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.
- **Controlador:** Responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. Los eventos son traducidos a solicitudes de servicio para el modelo o la vista.

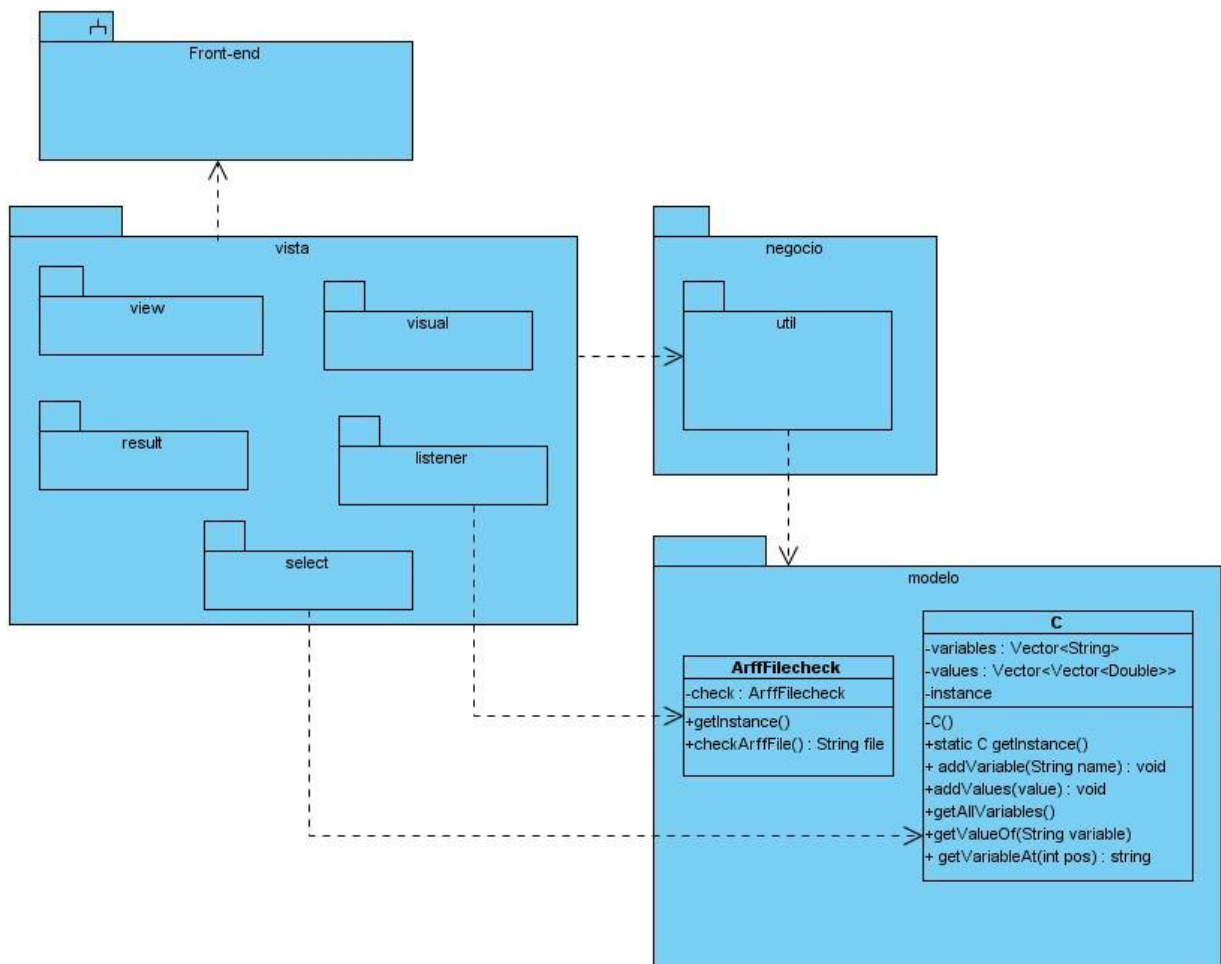


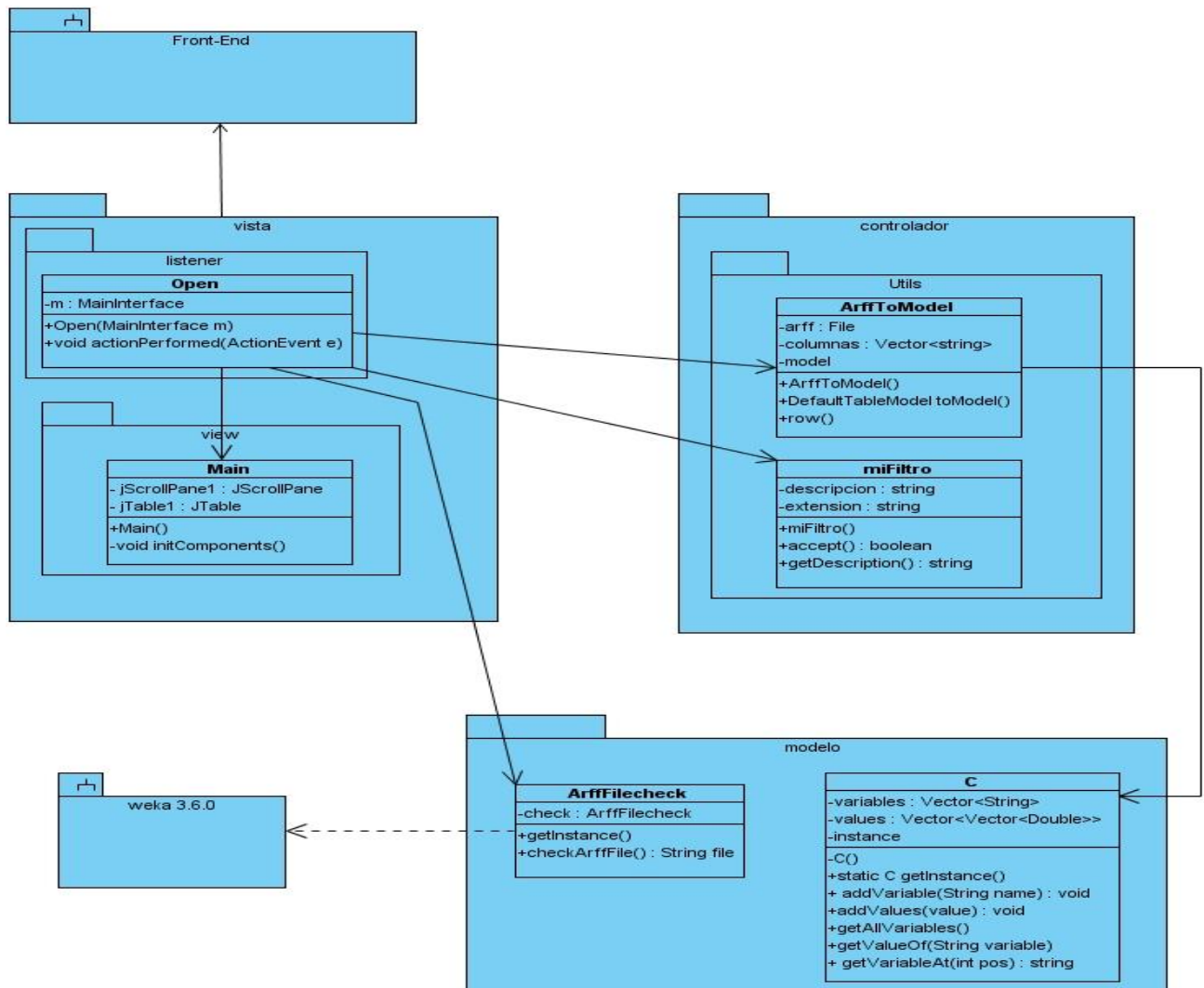
Figura 3.4: Ejemplo de aplicación de patrón MVC

Capítulo III

3.3 Diagrama de clases del diseño

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases y las relaciones entre ellas. Son utilizados durante el proceso de análisis y diseño, donde se describe la realización de los casos de uso y sirve de abstracción al modelo de implementación y al código fuente correspondiente. Constituyen la base para los diagramas de componentes y los diagramas de despliegue. Son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa.

A continuación se presenta en la figura 3.5 el diagrama de clase de diseño que representa al caso de uso cargar fichero.



Capítulo III

Figura 3.5: Ejemplo del diagrama de clase del diseño que representa al caso de uso cargar fichero.

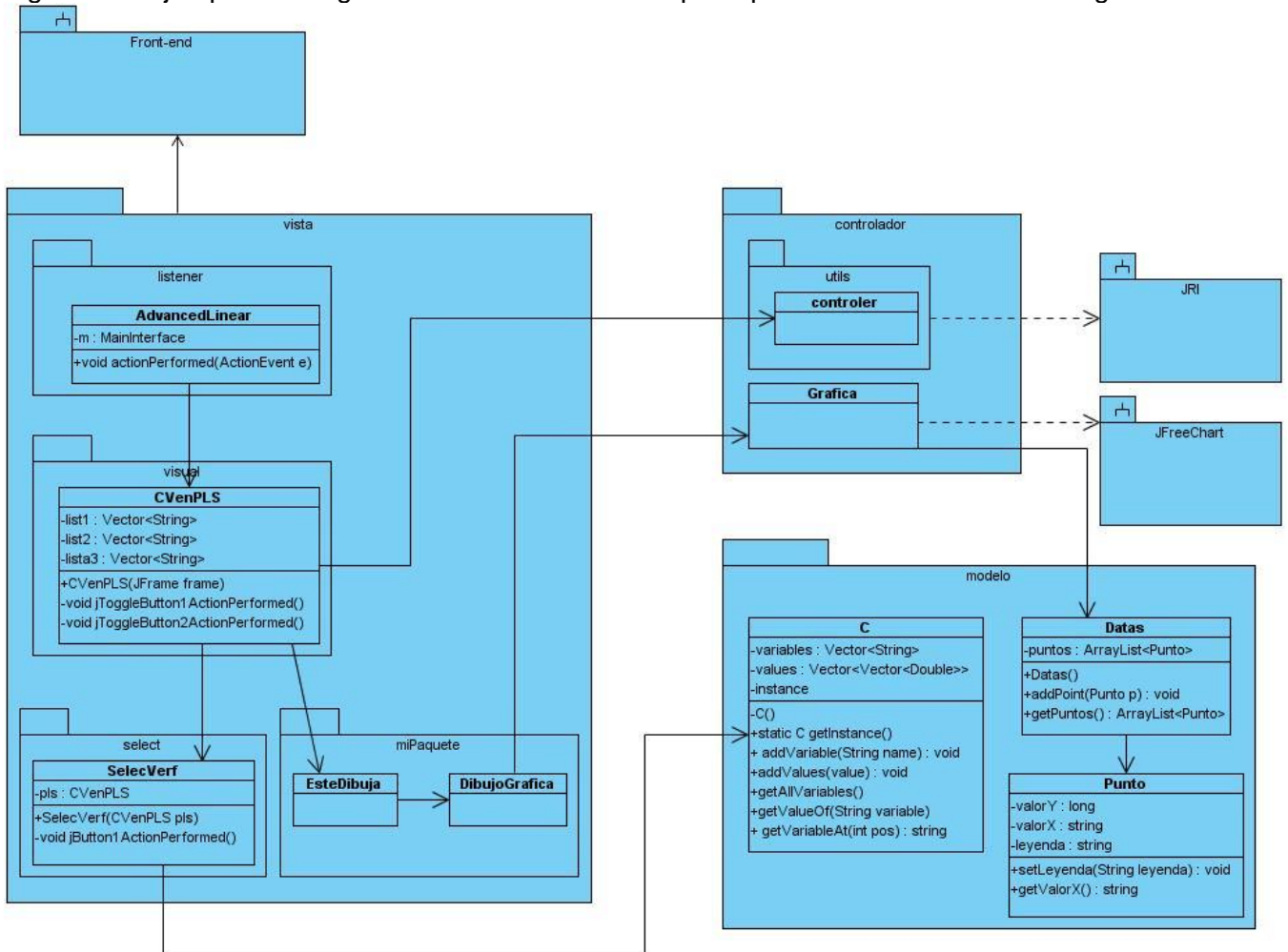


Figura 3.6: Ejemplo del diagrama de clase del diseño que representa al caso de uso realizar predicción.

3.4 Diagramas de Interacción: secuencia

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema. Un diagrama de interacción consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. Los diagramas de secuencia destacan el orden temporal de los mensajes. [31]

3.4.1 Diagramas de secuencia

Diagrama de secuencia que representa al caso de uso Cargar Fichero.

Capítulo III

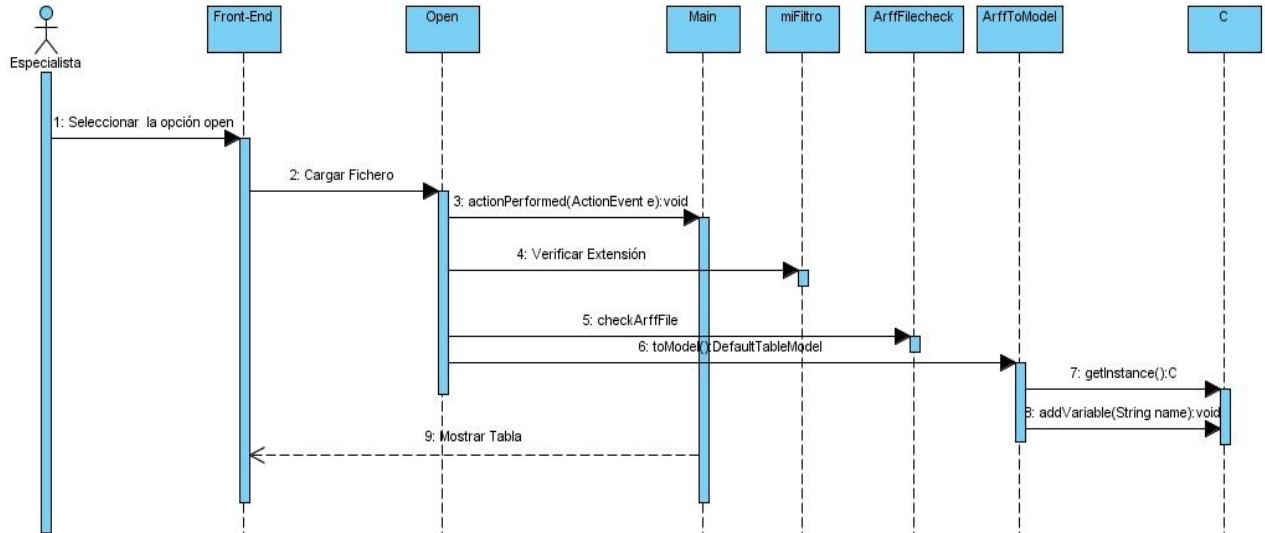


Figura 3.7: Ejemplo del diagrama de secuencia que representa al caso de uso Cargar Fichero.

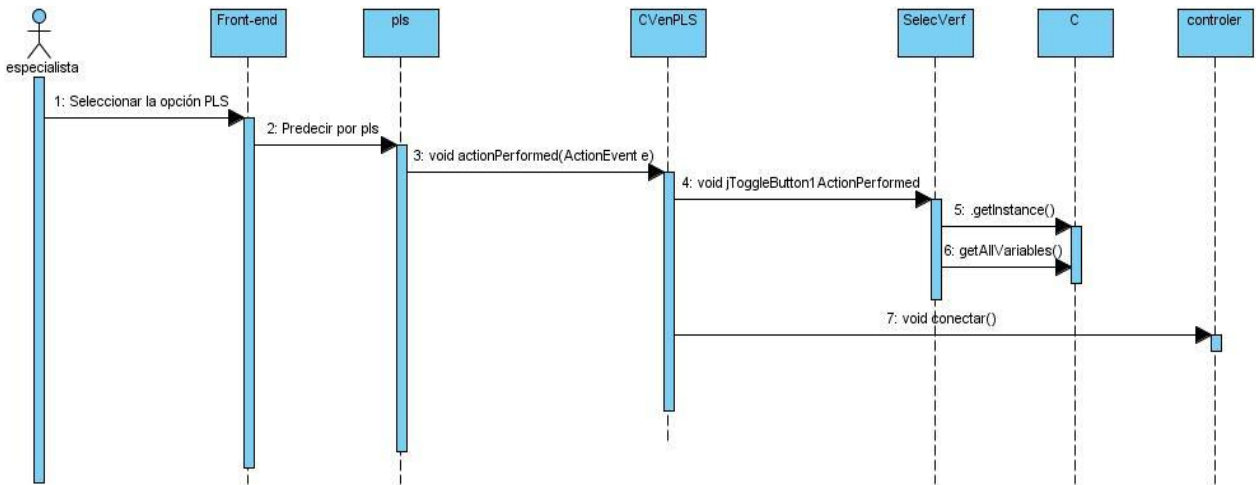
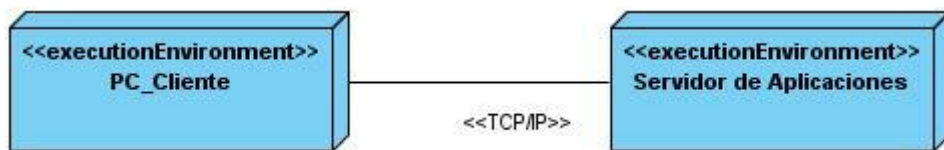


Figura 3.8: Ejemplo del diagrama de secuencia que representa al caso de uso Realizar Predicción.

3.5 Diagrama de despliegue

Configuración de nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos, se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño.



Capítulo III

Figura 3.9: Ejemplo del diagrama de despliegue en el diseño de la herramienta computacional.

3.6 Conclusiones

En el presente capítulo para la realización del diseño de la aplicación se utilizó el patrón de diseño *Singleton* y los patrones GRASP y como patrón arquitectónico Modelo Vista Controlador. Además, se realizó el diagrama de clases del diseño para los casos de uso así como los diagramas de interacción. Se mostró la posible distribución del sistema mediante el diagrama de despliegue.

Capítulo IV

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

Este capítulo está enfocado en la implementación del sistema con el objetivo de darle solución a los requisitos funcionales. Se representan los diagramas de componentes de los CU. Se muestran algunas pantallas de la aplicación con sus descripciones y se representa todo lo relacionado con las pruebas realizadas al sistema.

4.1 Diagrama de componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software, sean estos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del *software*, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo.

Por la complejidad del sistema, se consideró que no era factible una representación del diagrama de componentes de forma explícita. Por lo que a continuación se muestra el diagrama de componentes con el *plug-in* en forma de paquetes.

En la figura 4.1 se muestra el diagrama de componentes de la herramienta desarrollada.

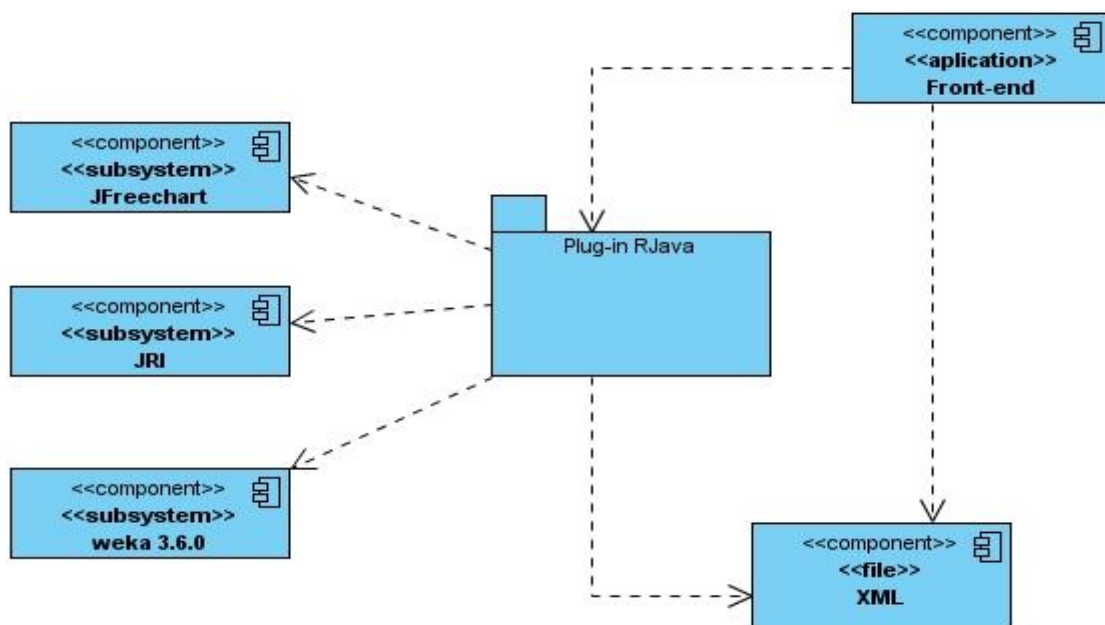


Figura 4.1: Ejemplo del diagrama de componentes.

Capítulo IV

4.2 Solución en la implementación del problema

Para dar cumplimiento al objetivo referente a la implementación de la herramienta diseñada se usó la librería JRI la cual permite la comunicación entre java y R.

R es un programa estadístico que cuenta con diversos paquetes que se agrupan en vistas, es decir, según la función que realizan. Uno de los paquetes que permite incluir el R es el pls, este paquete es de gran importancia para el desarrollo del trabajo ya que posee funciones necesarias que se utilizaron en la implementación.

Para la realización de las llamadas de las funciones del R desde java se crea un objeto de la clase *Reengine* de la librería JRI, para acceder a las funciones implementadas en el R.

A continuación se muestra un ejemplo de la llamada de funciones desde java al R.

```
if (!Reengine.versionCheck()) {
    System.err.println("*** Version mismatch - Java files don't match library version.");
    System.exit(1);
}
Reengine re=new Reengine(args, false, null);
if (!re.waitForR()) {
    System.out.println("Cannot load R");
    return;
}
String command;
String cbind = "cbind(";
String[] varNames = {jComboBox1.getSelectedItem().toString(),
jComboBox2.getSelectedItem().toString(), jComboBox3.getSelectedItem().toString()};

cbind += varNames[0];
for(int i = 0; i < 3; i++)
{
    if(i != 0)
        cbind += "," + varNames[i];
```


Capítulo IV

```
command = varNames[i] + "<-c(";
Vector<Double> valores = C.getInstance().getValueOf(varNames[i]);

command += valores.get(0);
for(int j = 1; j < valores.size(); j++)
{
    command += "," + valores.get(j);
}
command += ")";
re.eval(command);

System.out.println(command);
}
cbind += ")";

System.out.println(cbind);
re.eval("Matriz<-\" + cbind);
REXP g = re.eval("Matriz");
System.out.println(g);

re.eval("library(pls)");
REXP x = re.eval("NIPALS<-pls(Matriz[,1]~Matriz[,2:3], ncomp = 2, method = \"oscorespls\")");
System.out.println(x);
re.eval("plot(RMSEP(NIPALS), legendpos = \"topright\")");
re.end();
} else {
JOptionPane.showMessageDialog(null, " should select variables ");
}
}
```

En el ejemplo mostrado se crea un objeto de la clase *Rengine* llamado *re*, luego se crean los vectores con las variables elegidas del fichero cargado en la aplicación. Después se forma la matriz con los

Capítulo IV

valores de cada vector y posteriormente mediante la función `eval()` la cual analiza y evalúa una expresión R, devuelve el resultado (ver figura 4.2).

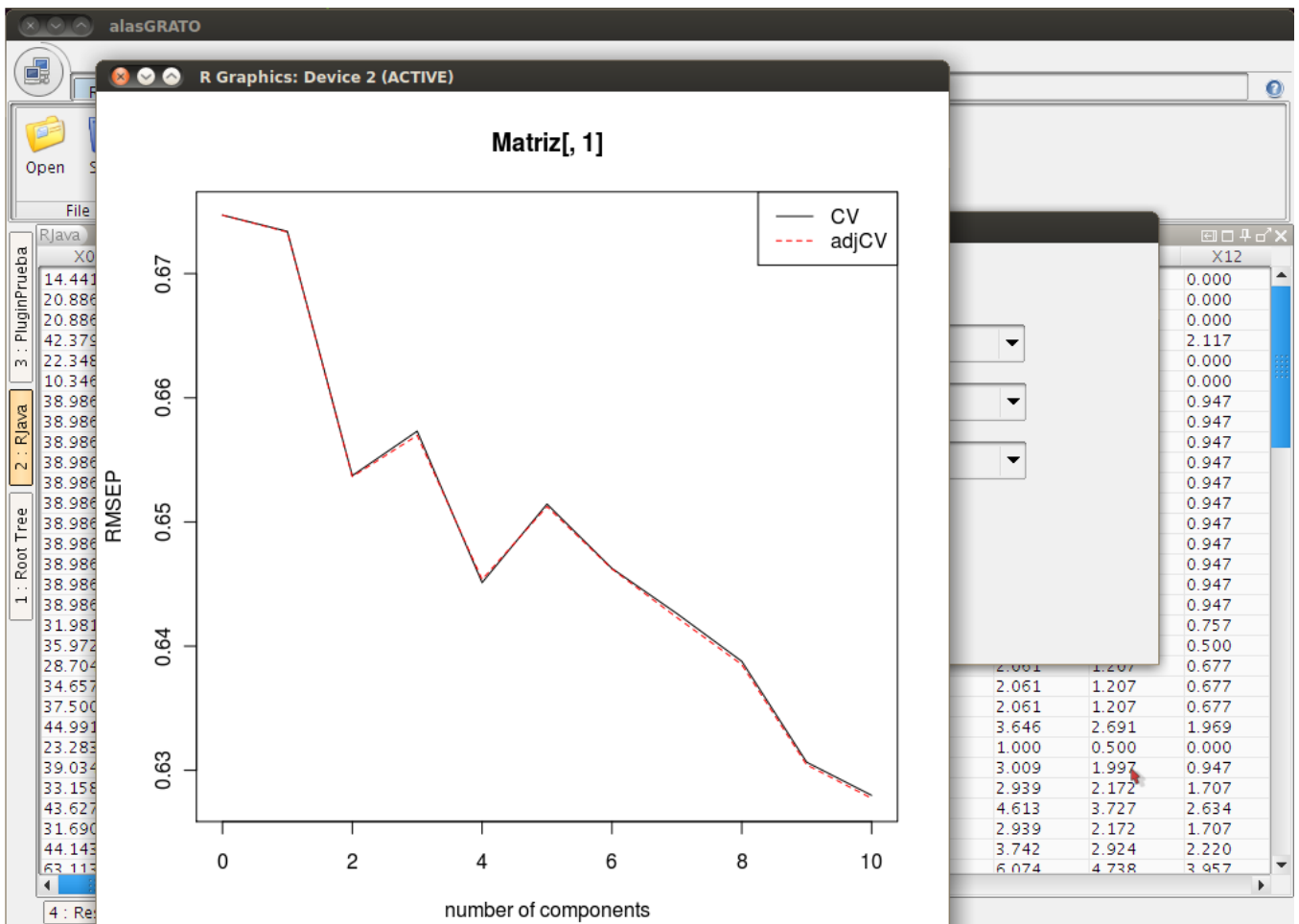


Figura 4.2: Ejemplo de resultados de la aplicación.

El resultado obtenido se muestra en un componente del R lo cual no permite obtener los datos para mostrarlo en un componente java en la aplicación, ya que el R evalúa las funciones y no hay forma de obtener los parámetros. Debido a esto no se pudo representar el resultado gráfico haciendo uso de la librería `jfreechart`, esta permitiría al usuario poder cambiarle los parámetros del gráfico tales como: título, color, fuentes y nombre de los ejes.

Capítulo IV

4.3 Pruebas del sistema

Las pruebas de *software*, son los procesos que permiten verificar la calidad de un producto de *software*. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un programa.

Se integran dentro de las diferentes fases del ciclo del *software* dentro de la Ingeniería de software. Así se ejecuta un programa y mediante técnicas experimentales se trata de descubrir que errores tienen. Para determinar el nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema. [32]

4.3.1 Configuración del entorno de pruebas

La configuración del entorno donde se vayan a ejecutar las distintas pruebas que se realizan a un *software*, es un aspecto significativo dentro del proceso de pruebas, pues si no se analizan bien los recursos que necesita el producto que se está construyendo. A la hora de probarlo se prescindirá de los elementos necesarios para la ejecución de un proceso de pruebas exitoso.

Por ello se tuvo en cuenta, a la hora de llevar a cabo todo el proceso de pruebas, algunos requerimientos de *hardware* y de *software* que hicieron posible un mejor desarrollo de las mismas, con el objetivo de minimizar los errores de la aplicación en desarrollo. Los requerimientos que se creyeron necesarios para las pruebas son los mostrados a continuación.

➤ **Requerimientos de software:**

- PC con Linux (Ubuntu).
- Máquina Virtual de Java 1.6.
- PC con instalación del programa estadístico R.

➤ **Requerimientos de hardware:**

- PC con Microprocesador Pentium IV a 2.41 GHz o superior.
- PC con 160 GB de disco duro.
- 512 MB de memoria RAM o superior.

Fecha de inicio	Fecha de fin	Actividades	Personal implicado
30/04/2010	30/04/2010	Aceptación y firma del Plan de Prueba.	Marisel Valdés Arango Elisa Fonseca Torrente

Capítulo IV

21/05/2010	3/05/2010	Verificación de las condiciones previas para el inicio de las pruebas.	Marisel Valdés Arango Elisa Fonseca Torrente
Primera Iteración del plug-in Rjava.			
3/05/2010	6/05/2010	Ejecución de las pruebas de Caja Negra al caso de uso Cargar Fichero.	Marisel Valdés Arango Elisa Fonseca Torrente
3/05/2010	6/05/2010	Ejecución de las pruebas de Caja Negra al caso de uso Realizar predicción	Marisel Valdés Arango Elisa Fonseca Torrente
3/05/2010	6/05/2010	Ejecución de las pruebas de Caja Negra al caso de uso Guardar resultado.	Marisel Valdés Arango Elisa Fonseca Torrente

4.3.2 Diseño de las pruebas de unidad (caja negra)

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del *software*. O sea, los casos de prueba pretenden demostrar que las funciones del *software* son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

4.3.2.1 Casos de prueba

Un caso de prueba es un conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados desarrollados para cumplir un objetivo en particular o una función esperada. Es la entidad más simple que siempre se ejecuta como una unidad, desde el comienzo hasta el final.

Los casos de prueba deben verificar si el producto satisface los requerimientos del usuario, tal y como se describe en las especificaciones de los requerimientos. Y si se comporta como se desea, tal y como se describe en las especificaciones funcionales del diseño. [33]

4.3.2.2 Pruebas de unidad (caja negra) para el caso de uso cargar fichero

Caso de uso “Cargar Fichero”

Sección “Cargar Fichero”

Capítulo IV

Variable (Var.)

1. Fichero Arff

Escenario	Var 1	Respuesta del sistema	Resultado de la prueba	Flujo central
Cargar Fichero correctamente	V	Se carga el fichero arff y se visualiza en pantalla.	Se carga el fichero arff y se visualiza en pantalla.	1. Pulsar el botón Open. 2. Se carga el fichero de tipo arff y se presiona el botón Open.
Cargar Fichero incorrectamente	F	No se carga el fichero arff y se muestra el siguiente mensaje: "error loading arff file".	No muestra el fichero.	1. Pulsar el botón Open. 2. Se presiona el botón OK y no se carga el fichero.

[Las celdas de la tabla que contienen V y F significan que, V indica válido, F indica inválido.]

Capítulo IV

4.4 Pantalla principal

RJava	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12
14.441	2.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.414	0.707	0.000	0.000
20.886	2.707	1.414	0.707	0.000	0.000	0.000	0.000	0.000	0.000	1.914	1.000	0.500	0.000
20.886	2.707	1.414	0.707	0.000	0.000	0.000	0.000	0.000	0.000	1.914	1.000	0.500	0.000
42.379	6.768	4.893	3.480	2.353	0.371	0.830	1.056	0.950	4.843	3.811	2.968	2.117	
22.348	2.707	1.414	0.707	0.000	0.000	0.000	0.000	0.000	0.000	1.914	1.000	0.500	0.000
10.346	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000
38.986	5.983	3.843	2.408	1.164	0.236	0.402	0.285	0.201	4.308	3.009	1.997	0.947	
38.986	5.983	3.843	2.408	1.164	0.236	0.402	0.285	0.201	4.308	3.009	1.997	0.947	
38.986	5.983	3.843	2.408	1.164	0.236	0.402	0.285	0.201	4.308	3.009	1.997	0.947	
38.986	5.983	3.843	2.408	1.164	0.236	0.402	0.285	0.201	4.308	3.009	1.997	0.947	
38.986	5.983	3.843	2.408	1.164	0.236	0.402	0.285	0.201	4.308	3.009	1.997	0.947	
38.986	5.983	3.843	2.408	1.164	0.236	0.402	0.285	0.201	4.308	3.009	1.997	0.947	
38.986	5.983	3.843	2.408	1.164	0.236	0.402	0.285	0.201	4.308	3.009	1.997	0.947	
38.986	5.983	3.843	2.408	1.164	0.236	0.402	0.285	0.201	4.308	3.009	1.997	0.947	
38.986	5.983	3.843	2.408	1.164	0.236	0.402	0.285	0.201	4.308	3.009	1.997	0.947	
38.986	5.983	3.843	2.408	1.164	0.236	0.402	0.285	0.201	4.308	3.009	1.997	0.947	
38.986	5.983	3.843	2.408	1.164	0.236	0.402	0.285	0.201	4.308	3.009	1.997	0.947	
38.986	5.983	3.843	2.408	1.164	0.236	0.402	0.285	0.201	4.308	3.009	1.997	0.947	
38.986	5.983	3.843	2.408	1.164	0.236	0.402	0.285	0.201	4.308	3.009	1.997	0.947	
31.981	5.276	3.343	2.055	0.895	0.236	0.402	0.285	0.236	3.808	2.656	1.747	0.757	
35.972	4.121	2.414	1.354	0.707	0.000	0.000	0.000	0.000	2.914	1.707	0.957	0.500	
28.704	4.828	2.914	1.707	0.957	0.000	0.000	0.000	0.000	3.414	2.061	1.207	0.677	
34.657	4.828	2.914	1.707	0.957	0.000	0.000	0.000	0.000	3.414	2.061	1.207	0.677	
37.500	4.828	2.914	1.707	0.957	0.000	0.000	0.000	0.000	3.414	2.061	1.207	0.677	
44.991	6.975	4.932	3.487	2.447	0.192	0.408	0.577	0.709	4.932	3.646	2.691	1.969	
23.283	2.707	1.414	0.707	0.000	0.000	0.000	0.000	0.000	1.914	1.000	0.500	0.000	
39.034	5.983	3.843	2.408	1.164	0.236	0.402	0.285	0.201	4.308	3.009	1.997	0.947	
33.158	5.560	3.932	2.753	2.046	0.192	0.408	0.617	0.465	3.932	2.939	2.172	1.707	
43.627	8.085	5.804	4.209	2.797	0.563	1.251	1.593	1.630	5.774	4.613	3.727	2.634	
31.690	5.560	3.932	2.753	2.046	0.192	0.408	0.617	0.465	3.932	2.939	2.172	1.707	
44.143	6.878	4.843	3.495	2.463	0.385	0.823	1.129	1.081	4.864	3.742	2.924	2.220	
63.113	10.439	7.915	5.825	4.490	0.577	1.125	1.727	1.928	7.381	6.074	4.738	3.957	
37.967	8.085	5.804	4.209	2.797	0.563	1.251	1.593	1.630	5.774	4.613	3.727	2.634	

Figura 4.3: Ejemplo de la pantalla principal de la aplicación.

En la figura 4.2 se muestra la pantalla principal de la aplicación donde el especialista carga el fichero para poder efectuar la selección de las variables y posteriormente realizar la predicción mediante el método de mínimos cuadrados parciales (PLS).

4.5 Conclusiones

Como resultado de este capítulo se obtuvo el diagrama de componentes. Asimismo, se identificaron seis funcionalidades agrupadas en tres requisitos funcionales, de los cuales sólo se lograron implementar tres de ellos representando el 50 por ciento del total. En este capítulo también se diseñaron tres casos de prueba en correspondencia con los casos de usos identificados logrando

Capítulo IV

aplicar uno con el objetivo de probar las funcionalidades del caso de uso cargar fichero obteniéndose resultados positivos. Por último, se muestra la interfaz principal de la aplicación con una breve descripción de esta.

Conclusiones

CONCLUSIONES

El estudio de la regresión de mínimos cuadrados parciales aplicada al diseño de fármacos como parte de las técnicas de estadística clásica y avanzada, proporcionó la elaboración del presente trabajo, cuyo resultado fue el “Diseño e implementación del módulo de estadística avanzada para la plataforma alasGRATO”, alcanzando los siguientes resultados:

- Se diseñó una herramienta que contempla la regresión de mínimos cuadrados parciales (PLS).
- Se implementó parcialmente la herramienta diseñada.
- Se implementó un *plug-in* para el módulo de estadística avanzada de la plataforma alasGRATO.

Recomendaciones

RECOMENDACIONES

- Utilizar la presente investigación como base de referencia cuando se quieran insertar nuevas técnicas de estadística clásica y avanzada en la Plataforma alasGRATO.
- Investigar cómo se pueden obtener los parámetros del R con el objetivo de poder utilizar la librería *jfreechart*, la cual permitirá al usuario poder cambiarle los parámetros del gráfico tales como: título, color, fuentes y nombre de los ejes.
- Utilizar como base el diseño realizado y lo investigado en la recomendación anterior para concluir la implementación de la herramienta.

Referencias Bibliográficas

REFERENCIAS BIBLIOGRÁFICAS

- [1]. PRIETO, J. O. -Entrenza, RUIZ, J. M. -Godoy, RAMÍREZ, Y -Hernández, I. Martí-Pérez y CARRASCO R. -Velar. Modelos difusos para la predicción de actividad biológica de cefalosporinas. Implementación y evaluación de un algoritmo. [En línea] 2010. [Citado el: 5 de Marzo de 2010.] Disponible en: http://www.google.com/#hl=es&source=hp&q=La+bioinform%C3%A1tica+es+una+nueva+disciplina+que+surge+como+respuesta+al+creciente+aumento+de+los+vol%C3%BAmenes+de+datos+biol%C3%B3gicos+y+estructurales&aq=f&aqi=&aql=&oq=&gs_rfai=&fp=5b3584be4cca4dda.
- [2]. MAROVAC, Jacqueline. Investigación y desarrollo de nuevos medicamentos: de la molécula al fármaco. [En línea] 2010. [Citado el: 5 de Marzo de 2010.] Disponible en: http://www.scielo.cl/scielo.php?pid=S0034-98872001000100015&script=sci_arttext#a1.
- [3]. ESTADISTICA. *Estadística*. [En línea] 2010. [Citado el: 5 de Marzo de 2010.] Disponible en: <http://www.scribd.com/doc/23664108/La-estadistica-es-una-ciencia-con-base-matematica-referente-a-la-recoleccion>.
- [4]. MOLINA, Luis M. Historia del razonamiento estadístico. [En línea] 2004. [Citado el: 18 de Marzo de 2010.] Disponible en: <http://www.seh-lelha.org/historiastat.htm>
- [5]. HISTORIA DEL RAZONAMIENTO ESTADISTICO. *Historia del razonamiento estadístico*. [En línea] 2010. [Citado el: 12 de junio de 2010.] Disponible en: <http://www.seh-lelha.org/historiastat.htm>.
- [6]. GLEASON, Federico Augusto. Técnicas estadísticas. [En línea] 2003. [Citado el: 18 de Marzo de 2010.] Disponible en: <http://www.liderempresarial.com/num101/13.php>
- [7]. FERNÁNDEZ, Rigoberto. Técnicas estadísticas multivariadas y sus aplicaciones a indicadores e índices económicos financieros de la actividad turística. Comentarios acerca de las Técnicas Multivariadas. [En línea] 2006. [Citado el: 18 de Marzo de 2010.] Disponible en: <http://www.gestiopolis.com/canales7/fin/matematicas-aplicacion-de-estadisticas-multivariables-e-indicadores-financieros.htm>
- [8]. FERNÁNDEZ, Rigoberto. Técnicas estadísticas multivariadas y sus aplicaciones a indicadores e índices económicos financieros de la actividad turística. Objetivos de las técnicas multivariadas. [En línea] 2006. [Citado el: 18 de Marzo de 2010.] Disponible en: <http://www.gestiopolis.com/canales7/fin/matematicas-aplicacion-de-estadisticas-multivariables-e-indicadores-financieros.htm>

Referencias Bibliográficas

- [9]. GELADI, P. & KOWALSKI, B. (1986): Partial Least Squares Regression: A tutorial, *Analytica Chmica Acta*, 185, pp. 1-17
- [10]. LÓPEZ, Yadira Marrero y GARCÍA, Adonis R. Rosales. Propuesta del diseño arquitectónico de la Plataforma bioGRATO. Ciudad de La Habana: s. n., 2008.
- [11]. JORDANA, Garcilaso. Introducción a OpenUp. [PPT] 2008. [En línea] 2006. [Citado el: 18 de Marzo de 2010.] Disponible en: www.mug.org.ar/Descargas/Jornadas/Downloads/GetFile.aspx?id.
- [12]. INTRODUCCIÓN TO OPENUP. Introducción to OpenUP. [En línea] 2009. [Citado el: 5 de Marzo de 2010.] Disponible en: <http://epf.eclipse.org/wikis/openup/>.
- [13]. LÓPEZ, Yadira Marrero y GARCÍA, Adonis R. Rosales. Propuesta del diseño arquitectónico de la Plataforma bioGRATO. Ciudad de La Habana: s. n., 2008.
- [14]. DEFINICIÓN DEL LENGUAJE DE PROGRAMACIÓN. Definición de lenguaje de programación. [En línea] 2010. [Citado el: 10 marzo de 2010.] Disponible en: <http://www.definicion.org/lenguaje-de-programacion>.
- [15]. DEFINICIÓN DEL LENGUAJE DE PROGRAMACIÓN. Lenguaje de programación Java [En línea] 2009. [Citado el: 15 de marzo de 2010.] Disponible en: <http://www.scribd.com/doc/3208978/INVESTIGACION-UNIDAD5>.
- [16]. ESCUELA DE C/ C++. Escuela de C/ C++. [En línea] 2004. [Citado el: 10 de Marzo de 2010.] Disponible en: <http://ldc.mx/foro/index.php?topic=1261.0;wap2>.
- [17]. NETBEANS. NetBeans [En línea] 2009. [Citado el: 15 de marzo de 2010.] Disponible en: http://netbeans.org/index_es.html.
- [18]. PROGRAMACIÓN ARCHIVOS. Aplicación para desarrollo en Java [En línea] 2009. [Citado el: 15 de marzo de 2010.] Disponible en: <http://www.deprogramas.com/?cat=18&paged=2>.
- [19]. RATIONAL SOFTWARE CORPORATION. 2003. Rational Unified Process. s. l.: Rational Software Corporation, 2003. Version 2003. 06.00. 65.
- [20]. VISUAL PARADIGM. Visual Paradigm. [En línea] 2009. [Citado el: 5 de Marzo de 2010.] Disponible en: <http://www.visual-paradigm.com/product/vpuml/modeleredition.jsp>.
- [21]. DAVID. Evaluación de Software: GNU R. [En línea] 2008. [Citado el 16 de Marzo 2010] Disponible en: <http://crisol.uc3m.es/index.php/gnur>.
- [22]. LENGUAJE DE PROGRAMACIÓN R. Lenguaje de programación R. [En línea]. [Citado el 16 de Marzo 2010] Disponible en: [http://www.worldlingo.com/ma/enwiki/es/R_\(programming_language\)](http://www.worldlingo.com/ma/enwiki/es/R_(programming_language)).

Referencias Bibliográficas

- [23]. DEFINICIÓN DE PLUG-IN. Definición de plug-in. [En línea] [Citado el 16 de Marzo 2010] Disponible en:<http://www.masadelante.com/faqs/plugin>.
- [24]. MARTINEZYS, Julio Villaverde. *Un nuevo front-end para la plataforma alasgrato*. Cuba : s. n., 2009. 1:4_21.
- [25]. UML y UP. Modelo de dominio. [En línea]. [Citado 20 mayo 2010] Disponible en: www.thevirtualworld.com.ar/files/megaresumen.doc
- [26]. TOROSI, Gustavo. Modelado de Objetos con UML: Actor. [En línea] [Citado el 16 de Marzo 2010] Disponible en: <http://www.chaco.gov.ar/utn/disenodesistemas/apuntes/oo/ApunteUML.pdf>.
- [27]. Diagrama de caso de uso del sistema. *Fase de Inicio. Disciplina de Requisitos* [En línea] [Citado el 16 de Marzo 2010] Disponible en: <http://eva.uci.cu/mod/resource/view.php?id=22044>.
- [28]. LARMAN, Craig. UML y Patrones. Introducción al análisis y diseño orientado a objetos. s. l. Prentice Hall, 2004.
- [29]. MAYO, Carlos. Patrones de diseño: Singleton. [En línea] [Citado el: 19 de 04 de 2010.] Disponible en: <http://blog.carlosmayo.net/2008/04/patrones-de-diseo-singleton.html>.
- [30]. WELICKI, León. Patrones y antipatrones. [En línea] [Citado el: 19 de 04 de 2010.] Disponible en: <http://msdn.microsoft.com/es-es/library/bb972251.aspx>.
- [31]. GUERRERO, Luis A. UML - Diagramas de interacción. [En línea] [Citado el: 19 de 04 de 2010.] Disponible en: http://www.google.com/url?sa=t&source=web&cd=1&ved=0CBQQFjAA&url=http%3A%2F%2Fwww.dcc.uchile.cl%2F~luguerre%2Fcc61j%2Frecursos%2Finteraccion.ppt&rct=j&q=Los+diagramas+de+interacci%C3%B3n+se+utilizan+para+modelar+los+aspectos+din%C3%A1micos+de+un+sistema.+Un+diagrama+de+interacci%C3%B3n+consiste+en+un+conjunto+de+objetos+y+sus+relaciones%2C+incluyendo+los+mensajes+que+se+pueden+enviar+entre+ellos.+Los+diagramas+de+secuencia+destacan+el+orden+temporal+de+los+mensajes.&ei=pz0WTMTyJIG0IQeWpdj7Cw&usg=AFQjCNGkusOBQ8A-mKO_ZVO8VFiF5BdqRA.
- [32]. INGENIERIA DE SOFTWARE. Ingeniería de Software. Un enfoque práctico. Capítulo 18 (Estrategias de Pruebas de Software). página 305-322.
- [33]. L/T El Proceso Unificado de Desarrollo de Software. Capítulo 11 (Prueba) de la página 281 a la 299.

Bibliografía

BIBLIOGRAFÍA

- BLALOCK, H. M. *Nonlinear iterative partial least squares (NIPALS) Modelling*. Vols. Quantitative Sociology: International Perspectives on Mathematical and Statistical Model Building, Academic Press. Nueva York: s. n., 1975.
- BENZÉCRI, J. *L'analyse des correspondances*. s. l.: Dunod, 1973.
- *El Proceso Unificado de Desarrollo de Software*. págs. 281 - 299. Capítulo 11 (Prueba).
- ESPOSITO-VINZI, V. & TENENHAUS, M. *PLS Logistic Regression, Proceedings of the PLS'01 International Symposium*. s. l.: CISIA-CERESTA, 2010.
- GELADI, P. & KOWALSKI, B. *Partial Least Squares Regression*. s. l.: Analytica Chimica Acta, 1986. págs. 1-17. 185.
- *Ingeniería de Software*. Capítulo 18 (Estrategias de Pruebas de Software). págs. 305-322. Un enfoque práctico.
- JORDANA, G. *Introducción a OpenUp*. 2008. ppt.
- JORESÖG, K.G. & WOLD, H. *The Basic Design and some Extensions, Systems under indirect observation*. Ámsterdam: s. n. Vol. Part II.
- KUTNER, M.H., NACHTSHEIM, C.J., NETER, J. & LI, W. *Applied Linear Statistical Models*. 5ta. s. l. : McGraw-Hill International Edition, 2005.
- LARMAN, Craig. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. s. l. : Prentice Hall, 2004.
- SJÖSTRÖM, M., WOLD, S., LINDBERG, W., PERSSON, J.-A. & MARTENS, H. *A Multivariate Calibration Problem in Analytical Chemistry Solved by Partial Least-Squares Models in Latent Variables*, *Analytica Chimica Acta*. 1983. págs. 61-70.
- MARDIA, K.V., KENT, J. T. & BIBBY, J.M. *Multivariate Analysis*, Academic Press. Londres: s. n., 1979.
- MARTENS, H. & NAES, T. *Multivariate Calibration*, Wiley. Nueva Cork: s. n., 1989.
- PERISSE, Marcelo Claudio. *Una Metodología Simplificada*. Buenos Aires: s. n., 2001. ISBN: 987-43-2947-5.

Bibliografía

- Rational Software Corporation. *Rational Software Corporation*. 2003. Version 2003.06.00.65.
- WOLD, H. *Estimation of principal components and related models by iterative least squares*. Nueva York: Multivariate Analysis, Academic Press, 1966. Vols. KRISHNAIAH, P.R.
- BOULESTEIX, A.-L. & STRIMMER, K. *Partial Least Squares: A Versatile Tool for the Analysis of High-dimensional Genomic Data*. 1997, 8 (1): 32-44.
- BETZIN, J. *PLS-Pfadanalyse und KFA-Modelle*. s. I.: Psychologische Beiträge, 2000. Traducción de: *Globale und lokale Entdeckungen*. 42 (3): 469-493.
- DAYAL, B. S. & MACGREGOR, J.F. *Improved PLS Algorithms*, *Journal of Chemometrics*. 1997, 11: 73-85.
- DE JONG, S. & TER BRAAK, C.J.F. *Comments on the PLS kernel algorithm*, *Journal of Chemometrics*. 1994, 8:169-174.
- DE JONG, S. *SIMPLS: An alternative approach to partial least squares regression*. 1993, 18: 251-263..
- DING, B. & GENTLEMAN, R. *Classification using generalized partial least squares*, *Journal of Computational and Graphical Statistics*. 2005, 14: 280-298.
- JORESÖG, K G. *A general method for analysis of covariance structures*. s. I.: *Biométrica*, 1970, 57: 239-251.
- LOPEZ, Yadira Marrero y GARCIA, Adonis R. Rosales. *Propuesta del diseño arquitectónico de la Plataforma bioGRATO*. Ciudad de La Habana, 2008.
- MARTINEZ VILLAVARDE, Julio. *Un nuevo front-end para la plataforma alasgrato*. Habana, 2008.
- MIRANDA, J. *Comparación de métodos discriminantes para la búsqueda de predictores a partir de perfiles de expresión de genes*. Tesis presentada en opción al grado de Máster en Ciencias Matemáticas. Universidad de La Habana. Habana.
- MIRANDA, J. *Comparación de métodos discriminantes para la búsqueda de predictores a partir de perfiles de expresión de genes*. Tesis presentada en opción al grado de Máster en Ciencias Matemáticas. Universidad de La Habana Habana, 2008.

Bibliografía

- VEGA VILCA, J. C. *Generalizaciones de mínimos cuadrados parciales con aplicación en clasificación supervisada*. Tesis Presentada en Opción al grado de Dr. en Filosofía. 2004.
- ANONIMO, *Técnicas estadísticas multivariadas y sus aplicaciones a indicadores e índices económicos financieros de la actividad turística. Objetivos de las técnicas multivariadas*. Disponible en: <<http://www.gestiopolis.com/canales7/fin/maticas-aplicacion-de-estadisticas-multivariadas-e-indicadores-financieros.htm>>
- APLICACIÓN PARA DESARROLLO EN JAVA. *Aplicación para desarrollo en Java*. [En línea] 2009. [Citado el: 15 de 3 de 2010.]. Disponible en: <<http://www.deprogramas.com/?cat=18&paged=2>>
- CORRELACIÓN LINEAL Y ANÁLISIS DE REGRESIÓN. *RegresionLineal* [En línea] Disponible en: <www.uoc.edu/in3/emath/docs/RegresionLineal.pdf>
- CORRELACION Y REGRESION LINIAL. *Correlación y regresión lineal*. [En línea] Correlación y regresión lineal. Disponible en: <http://nutriserver.com/Cursos/Bioestadistica/Correlacion_Regresion.html>
- CURSO BASICO R. *Curso básico de R*. [En línea]. Disponible en: <<http://www.ub.edu/stat/docencia/EADB/Curso%20basico%20de%20R.htm>>
- David. *Evaluación de Software: GNU R*. [En línea] 2008. [Citado el: 5 de 3 de 2010.]. Disponible en: <<http://crisol.uc3m.es/index.php/gnur>>
- DEFINICION EN 3 CAPAS. *Definición en 3 capas*. [En línea] 2009. [Citado el: 5 de 3 de 2010.] Disponible en: <http://forge.morfeo-project.org/wiki/index.php/D.3.2_Documento_de_definici%C3%B3n_de_modelos_avanzados_de_comunicaci%C3%B3n_y_composici%C3%B3n_de_recursos#Definici.C3.B3n_de_Capas>
- DISCIPLINA DE REQUISITO. *Diagrama de caso de uso del sistema. Fase de Inicio. Disciplina de Requisitos*. [En línea] [Citado el: 6 de 6 de 2010.] Disponible en: <<http://eva.uci.cu/mod/resource/view.php?id=22044>>

Bibliografía

- ENSAYO DE ESTADISTICA. *Ensayo Estadística*. [En línea]. Disponible en: <<http://www.buenastareas.com/ensayos/Ensayo-Estadistica/93019.html>>
- ESCUELA DE C. *Escuela de C/C++*. [En línea] 2004. [Citado el: 15 de 3 de 2010.] .Disponible en: <<http://ldc.mx/foro/index.php?topic=1261.0;wap2>>
- ESTADISTICA DESCRIPTIVA. *Estadística descriptiva*. [En línea] Disponible en: <<http://sitios.ingenieria-usac.edu.gt/estadistica/estadistica2/estadisticadescriptiva.html>>
- ESTADISTICA DESCRIPTIVA. *Estadística descriptiva*. [En línea] Disponible en: <<http://www.powerpoint-search-engine.com/estad%C3%ADstica-descriptiva-ppt.html>>
- ESTADISTICA DESCRIPTIVA. *Estadística descriptiva*. [En línea]. Disponible en: <<http://eco.unex.es/georgina/La%20Estad%C3%ADstica%20Descriptiva.htm>>
- ESTADISTICA. *estadística*. [En línea]. Disponible en: <<http://aliolibastaya.blogia.com>>
- FERNÁNDEZ, Rigoberto. *Técnicas estadísticas multivariadas y sus aplicaciones a indicadores e índices económicos financieros de la actividad turística. Comentarios acerca de las Técnicas Multivariadas*. [En línea] [Citado el: 18 de 3 de 2010.]. Disponible en: <<http://www.gestiopolis.com/canales7/fin/matematicas-aplicacion-de-estadisticas-multivariadas-e-indicadores-financieros.htm>>
- FOROFILOS. *www.forofilos.com*. [En línea] [Citado el: 18 de 2 de 2010.] Disponible en: <<http://www.forofilos.com/index.php/topic,2444.0.html>>
- GESTIOPOLIS. *Matemáticas aplicación de estadísticas multivariadas e indicadores financieros*. [En línea] Disponible en: <<http://www.gestiopolis.com/canales7/fin/matematicas-aplicacion-de-estadisticas-multivariadas-e-indicadores-financieros.htm>>
- GUERRERO, Luis A. UML - *Diagramas de interacción*. [En línea] [Citado el: 19 de 4 de 2010.]. Disponible en: <<http://www.google.com/url?sa=t&source=web&cd=1&ved=0CBQQFjAA&url=http%3A%2F%2Fwww.dcc.uchile.cl%2F~luguerre%2Fcc61j%2Frecursos%2Finteraccion.ppt&rct=j&q=Los+diagramas+de+interacci%C3%B3n+se+utilizan+para+modelar+los+aspectos+din%C3%A1micos+de+un+sistema.+>>>

Bibliografía

- GUIA ESTADISTICA. *Guía estadística*. [En línea] Universidad de Puerto Rico. Disponible en: <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/guia_estadistica/modulo_12.htm>
- HISTORIASTAT. Historia del razonamiento estadístico. [En línea]. Disponible en: <<http://www.seh-lelha.org/historiastat.htm>>
- JAVA. java. [En línea] java. Disponible en: <mncs.atca.um.es/javaEE/Curso%20Java%20y%20Tecnologías%20J2EE.ppt/java/mncs.atca.um.es/java.>>
- LENGUAJE DE PROGRAMACION JAVA. Lenguaje de programación Java. [En línea] 2009. Disponible en: <<http://www.scribd.com/doc/3208978/INVESTIGACION-UNIDAD5>>
- LENGUAJE DE PROGRAMACION R. *Lenguaje de programación R*. [En línea] [Citado el: 16 de 3 de 2010.] Disponible en: <[http://www.worldlingo.com/ma/enwiki/es/R_\(programming_language\)>](http://www.worldlingo.com/ma/enwiki/es/R_(programming_language)>)>
- LENGUAJE DE PROGRAMACION. *Lenguaje de programación* Definicion.org. *Definicion.org*. [En línea] 2009. [Citado el: 10 de 3 de 2010.] Disponible en: <<http://www.definicion.org/lenguaje-de-programacion>>
- LIBRARY.Library. [En línea] 2009. [Citado el: 15 de 3 de 2010.] Disponible en: <<http://msdn.microsoft.com/es-es/library/bb972251.aspx#M15.>>>
- LIDER EMPRESARIAL. *www.liderempresarial.com*. [En línea] 2010. Disponible en: <<http://www.liderempresarial.com/num101/13.php>>
- MAROVAC, Jacqueline. *Investigación y desarrollo de nuevos medicamentos: de la molécula al fármaco*. [En línea] [Citado el: 5 de 3 de 2010.] .Disponible en: <http://www.scielo.cl/scielo.php?pid=S0034-98872001000100015&script=sci_arttext#a1>
- MAYO, Carlos. *Patrones de diseño: Singleton*. [En línea] [Citado el: 19 de 4 de 2010.]. Disponible en: <<http://blog.carlosmayo.net/2008/04/patrones-de-diseo-singleton.html>>
- MOLINA, Luis M. *Historia del razonamiento estadístico*. [En línea] 2004. [Citado el: 18 de 3 de 2010.]. Disponible en: < <http://www.seh-lelha.org/historiastat.htm>>
- NETBEANS. *Netbeans*. [En línea] 2009. [Citado el: 15 de 3 de 2010.]. Disponible en: <http://netbeans.org/index_es.html>

Bibliografía

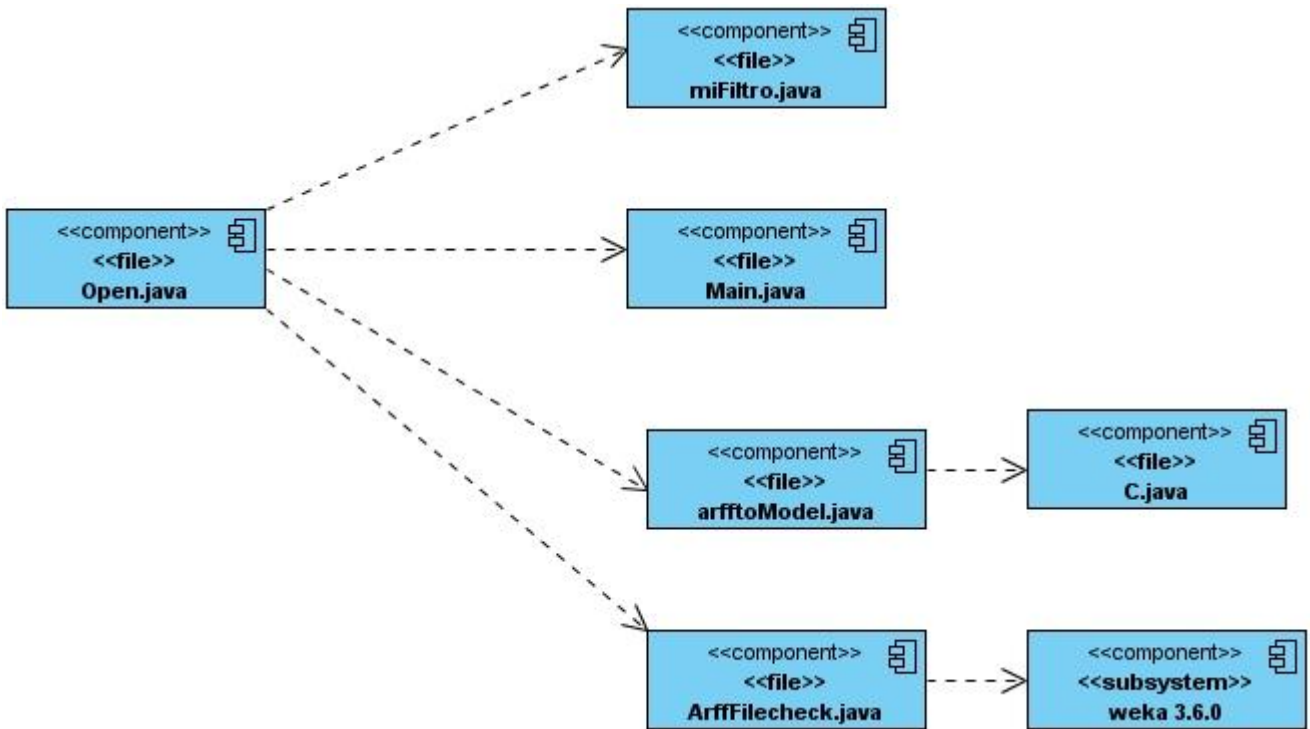
- OPENUP. *Openup*. [En línea] 2009. Disponible en: <<http://epf.eclipse.org/wikis/openup/>>
- PECOS, Daniel. *Mysql postgres*. [En línea] [Citado el: 19 de 02 de 2010.] Disponible en: <http://danielpecos.com/docs/mysql_postgres/x57.html>
- PLUG-IN. *Definición de plug-in*. [En línea] [Citado el: 16 de 3 de 2010.]. Disponible en: <<http://www.masadelante.com/faqs/plug-in>>
- PRIETO, J. O. -Entrenza, RUIZ, J. M. -Godoy, RAMÍREZ, Y -Hernández, I. Martí-Pérez y CARRASCO R. -Velar. *www.google.com*. *www.google.com*. [En línea] [Citado el: 5 de 3 de 2010.] Modelos difusos para la predicción de actividad biológica de cefalosporinas. Implementación y evaluación de un algoritmo. Disponible en: <http://www.google.com/#hl=es&source=hp&q=La+bioinform%C3%A1tica+es+una+nueva+disciplina+que+surge+como+respuesta+al+creciente+aumento+de+los+vol%C3%BAmenes+de+datos+biol%C3%B3gicos+y+estructurales&aq=f&aqi=&aql=&oq=&gs_rfai=&fp=5b3584be4cca4dda>
- R PARA PRINCIPIANTES. *R para Principiantes*. [En línea]. Disponible en: <http://cran.r-project.org/doc/contrib/rdebuts_es.pdf>
- REGRESION LINIAL MULTIPLE. *Regresión lineal múltiple*. [En línea]. Disponible en: <<http://html.rincondelvago.com/regresion-lineal-multiple.html>>
- REGRESION LINIAL. *Regresión lineal*. [En línea]. Disponible en: <<http://www.eumed.net/cursecon/medir/index.htm>>
- REPOSITORIO. *repositorio.uvm.cl*. [En línea] Disponible en: <<http://repositorio.uvm.cl/gsdll/collect/sociol/index/assoc/HASH01e3.dir/doc.pdf>>
- RETAMAR, Ángel. *AsturLinux*. [En línea] 2004. Disponible en: <http://asturlinux.org/archivos/jornadas2004/ponencias/eclipse_ide.pdf>
- SANCHEZ, Juan José. *mncs.atika.um.es*. [En línea] Disponible en: <[EE/Curso%20Java%20y%20Tecnologías%20J2EE.ppt/Universidad de Murcia](EE/Curso%20Java%20y%20Tecnologías%20J2EE.ppt/Universidad%20de%20Murcia)>
- SCRIBD. *La estadística es una ciencia con base matemática referente a la recolección*. [En línea] Disponible en: <<http://www.scribd.com/doc/23664108/La-estadistica-es-una-ciencia-con-base-matematica-referente-a-la-recoleccion>>

Bibliografía

- SEH-LELHA. *seh-lelha.org*. [En línea] Disponible en: < <http://www.seh-lelha.org/historiastat.htm>>
- SUN DEVELOPER NETWORK. *Sun Developer Network*. [En línea]2009. Disponible en: <<http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>>
- SUN. *Sun Microsystems*. [En línea] [Citado el: 15 de 3 de 2010.] Disponible en: <Sun Microsystems. www.sun.com>
- TORINO, Herman. Resumen del libro de estadísticas de Berenson y Levine. [En línea] Disponible en: <<http://www.monografias.com/trabajos13/beren/beren.shtml>>
- TOROSI, Gustavo. *Modelado de Objetos con UML: Actor*. [En línea] [Citado el: 6 de 6 de 2010.] Disponible en: <<http://www.chaco.gov.ar/utn/disenodesistemas/apuntes/oo/ApunteUML.pdf>>
- UML y UP. *UML y UP. Modelo de dominio*. [En línea] [Citado el: 6 6, 2010.] Disponible en: <www.thevirtualworld.com.ar/files/megaresumen.doc>
- VISUAL PARADIGM. *Visual Paradigm*. [En línea] 2009. Disponible en: <<http://www.visual-paradigm.com/product/vpuml/modeleredition.jsp>>
- W, D. *Masadelante*. [En línea] [Citado el: 22 de 02 de 2010.]. Disponible en: <<http://www.masadelante.com/faqs/plugin-in>>
- WELICKI, León. *Patrones y antipatrones*. [En línea]. Disponible en: <<http://msdn.microsoft.com/es-es/library/bb972251.aspx>>

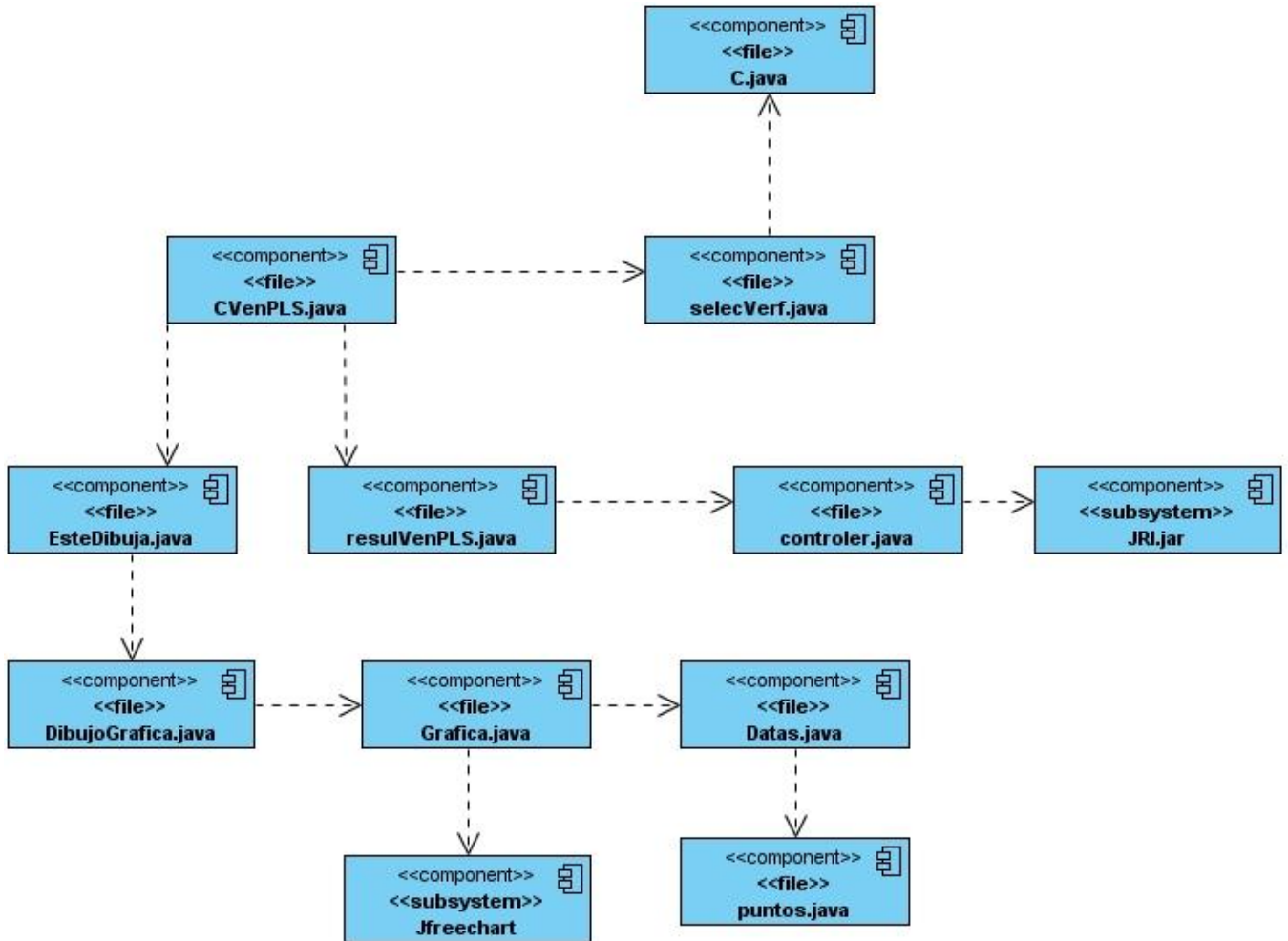
Anexos

ANEXOS



Anexo 1. Diagrama de Componente para el caso de uso Cargar fichero.

Anexos



Anexo 2. Diagrama de Componente para el caso de uso realizar predicción.

Glosario

GLOSARIO DE TERMINOS

Actividad biológica: actividad que caracteriza el comportamiento biológico en compuestos químicos (molécula o fragmento), además es una expresión que describe los efectos benéficos o adversos de una droga sobre la materia viva.

Bioinformática: es la aplicación de los ordenadores y los métodos informáticos en el análisis de datos experimentales y simulación de los sistemas biológicos.

Covarianza: es una medida de la intensidad de cierta asociación estadística entre dos variables. La covarianza es igual al promedio de los productos cruzados menos el producto de las medias.

Datos biológicos: información sobre ciencias de la vida, recogida de experimentos científicos, literatura publicada, tecnología de experimentación de alto rendimiento, y análisis computacional.

Descriptor: número que describe la estructura química o una propiedad de la molécula o fragmento de esta.

Factores demográficos: estructura dinámica de la población, densidad urbana y rural, migración interna y externa, factores genéticos, existencia de una política nacional de planificación familiar.

Formulismo: excesivo apego a las fórmulas en la resolución y ejecución de cualquier asunto.

Fragmento: una pequeña parte de la molécula a la cual se llega a través de un método que es el encargado de fragmentarla.

Hardware: dispositivo electrónico apto para interpretar y ejecutar comandos programados para operaciones de entrada, salida, cálculo y lógica.

Herramientas computacionales: son las que facilitan los trabajos diarios en los ámbitos laborales de cualquier índole.

LaTeX: es un sistema de composición de textos, orientado especialmente a la creación de libros, documentos científicos y técnicos que contengan fórmulas matemáticas.

Máquina virtual: es un software que emula a una computadora y puede ejecutar programas como si fuese una computadora real.

MATLAB (Laboratorio de Matrices): es un lenguaje de alto funcionamiento para computación técnica, este integra computación, visualización, y programación, en un entorno fácil de usar donde los problemas y las soluciones son expresados en la más familiar notación matemática.

Media: es la suma de todos los valores dividida por el número de valores.

Mediana: valor para el cual la mitad o menos de la población, tienen valores inferiores a él y la mitad o menos de la población, tienen valores superiores a él.

Moda: el valor más frecuente de la variable.

Glosario

Moléculas: una molécula es una partícula formada por un conjunto de átomos ligados por enlaces covalentes.

Multicolinealidad: significa que tus variables independientes están relacionadas. Se refiere solamente a relaciones lineales entre variables x . No elimina las relaciones no lineales existentes entre ellas.

Multiplataforma: es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

Perl: se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros.

Plug-in: es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica, generalmente muy específica, como por ejemplo servir como driver en una aplicación, para hacer así funcionar un dispositivo en otro programa.

Poliomielitis: es una enfermedad contagiosa, esta enfermedad afecta al sistema nervioso central.

Predicción: puede referirse tanto a la acción y al efecto de predecir como a las palabras que manifiestan aquello que se predice; en este sentido, predecir algo es anunciar por revelación, ciencia o conjetura algo que ha de suceder.

Python: lenguaje de programación de propósito general, orientado a objetos, que también puede utilizarse para el desarrollo web.

Redes neuronales: es un modelo computacional con un conjunto de propiedades específicas, como son la habilidad de adaptarse o aprender, generalizar u organizar la información, todo ello basado en un procesamiento eminentemente paralelo. Es un procesador masivamente paralelo distribuido que es propenso por naturaleza a almacenar conocimiento experimental y hacerlo disponible para su uso.

Rol: está vinculado a la función o papel que cumple alguien o algo.

RWeka: es un software programado en Java que está orientado a la extracción de conocimientos desde bases de datos con grandes cantidades de información.

Sistema operativo: conjunto de programas que se integran con el hardware para facilitar al usuario, el aprovechamiento de los recursos disponibles.

Software: se refiere al equipamiento lógico o soporte lógico de una computadora digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de tareas específicas.

Vector: es una utilidad matemática que se utiliza en numerosísimos aspectos de las matemáticas así como en las ciencias, y especialmente, en física.