Universidad de las Ciencias Informáticas

Facultad 6

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

Título: *Componente para análisis estadísticos de coeficientes de endogamia para la solución informática SEEGEN-R.*

Autor: Ernesto Alfredo Molina Suárez

Tutor(es): Ing. Yudiel la Rosa González

Ing. Dayana Joseph Smarth

La Habana, Junio 2015

"Año 57 de la Revolución"

"El futuro de nuestra patria tiene que ser necesariamente un futuro de hombres de ciencia, tiene que ser un futuro de hombres de pensamiento, porque precisamente es lo que más estamos sembrando…"

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los __ días del mes de _____ del año 2015.

_____

Ernesto Alfredo Molina Suárez

Firma del autor

_____

Ing. Yudiel la Rosa González

Firma del tutor

_____

Ing. Dayana Joseph Smarth

Firma del tutor

El presente trabajo está dedicado a:

Mis padres que siempre me han ayudado a cursar el camino que escogí con la mejor de las disposiciones.

Mis abuelos que han sido los que más han tenido que batallar para cumplir mis caprichos.

En especial a la memoria de mi abuelo Ernesto por ser el principal responsable de las decisiones que he tomado en mi vida y gracias a él soy Ingeniero en Ciencias Informáticas.

Deseo agradecer:

A toda mi familia, por apoyarme siempre y gracias a lo cual he vivido mi vida feliz.

A mis compañeros de aula, que aguantaron tanto mi complicada forma de ser.

A todos los amigos que he conocido en estos 5 años de estudios, en especial a aquellos con los cuales he compartido más tiempo, Piedra, Asley, Arlet, Yisell, Rolando, Felipe, Lorgio, Ana, Pepo, El palenque con todos sus integrantes, Fabian, Alex, Bernardo, Argota, Alejandro, Emilio, el Fuhrer, Pepe, Yainel, Ernesto, Kiki, Katty, El Cabo, Marbier, Dayana, Paneque, Glennis, Jessie, Andy, Rolando Whisper, Aryanna, Yailin, Yaima, Yosiel, Osviel, Fabiel, a los que ya no están, y ojalá que no se me quede ninguno por mencionar.

A los que de una manera u otra pusieron su granito de arena para que este día llegara.

A los que han sido mis compañeros de apartamento, por las luchas y las fiestas que hemos vivido.

A mis profesores, por haberme permitido aprender de ellos todo lo que pude.

A mis compañeros del club de los laboratorios de docencia que tantas noches hicimos la misma colita para entrar a estudiar.

A todo el que me aguantó una conversación por el jabber estando yo aburrido.

Y por último, pero no menos importante, a bailar casino y a la programación, porque gracias a que se un poquito de las dos cosas pude bailar una rueda de casino en primer año que nunca voy a olvidar.

Resumen

En el Centro Nacional de Genética Médica (CNGM), se realizan los cálculos estadísticos de los estudios de Epidemiología Genética utilizando el Sistema Estadístico de Epidemiología Genética – basado en R (SEEGEN-R), herramienta desarrollada en la Universidad de las Ciencias Informáticas (UCI) que contiene módulos para la realización de estudios de Genética Poblacional, Epidemiología Tradicional y Epidemiología Genética. Actualmente, en el CNGM, se realizan análisis estadísticos de los coeficientes de endogamia de la población, pero SEEGEN-R no contiene las funciones necesarias para la realización de los mismos. El objetivo de la presente investigación es desarrollar un componente que permita la integración de los estudios de coeficientes de endogamia para su uso por los especialistas del CNGM. Se utiliza Java como lenguaje de programación para manejar la presentación de los análisis a los usuarios y R como motor de cálculo de las funciones estadísticas. En este componente se utilizó RUP como metodología de desarrollo, Netbeans como entorno de desarrollo integrado y la librería de clases RServe para comunicar los lenguajes de programación utilizados. Como resultado se obtuvo un componente, que satisface las necesidades del CNGM, para realizar los estudios de coeficientes de endogamia en la solución informática SEEGEN-R.

PALABRAS CLAVE: análisis estadísticos, coeficientes de endogamia, estadísticos F, genética médica, genética poblacional.

Abstract

The National Centre for Medical Genetic (CNGM), perform statistical calculations of the studies of genetic epidemiology using the Statistical System of Genetic Epidemiology – R based (SEEGEN-R), tool developed at the University of Informatics Sciences (UCI) that contains modules for studies of population genetics, traditional epidemiology and genetic epidemiology. Currently, in the CNGM, statistical analysis of the inbreeding coefficients in the population are made, but SEEGEN-R does not contain the functions required for the realization of the same. The objective of this research is to develop a component that allows the integration of studies of inbreeding coefficients for use by the CNGM specialists. Is used Java as a programming language to manage the presentation of the analyses to the users and R as an engine for calculation of statistical functions. RUP as methodology development, Netbeans as IDE and RServe library was used in this component to communicate the used programming languages. Was obtained as result a component, which meets the needs of the CNGM to make studies of inbreeding coefficients in the software SEEGEN-R.

KEYWORDS: statistical analysis**,** inbreeding coefficients, F-statistics, medical genetics, population genetics

Índice

Índice de figuras

Índice de tablas

Introducción

La Epidemiología Genética, disciplina de la Medicina surgida a mediados de los años 80, trata de comprender la interacción entre los factores genéticos ambientales que dan origen a las enfermedades del ser humano. Está considerada como una ciencia básica de la medicina preventiva, además de una fuente de información para la formulación de políticas de salud pública. En los últimos años ha tenido un desarrollo notable gracias a la biología molecular y el despliegue de sus marcadores genéticos, además de los avances de la tecnología en esta rama de la medicina. (Khoury MJ, 1993)

Actualmente, las iniciativas en este campo de investigación se encuentran en plena expansión, ofreciéndose diversos programas educativos y de investigación por todo el mundo. La Sociedad Internacional de Epidemiología Genética suma cada año nuevos miembros, evidenciando así la evolución de la epidemiología como ciencia.

En Cuba, la principal fortaleza de la genética radica en la introducción de los servicios de genética clínica en la atención primaria de salud, gracias a la organización vigente en la Red Nacional de Genética Médica que va desde el nivel primario (consultorios y policlínicos), hasta el terciario que son las instituciones, como el Centro Nacional de Genética Médica (CNGM), que constituye su institución rectora. Esta red tiene dentro de sus funciones principales las investigaciones básicas y aplicadas en el campo de la Genética Médica, la Inmunología, la Bioquímica y la Epidemiología Genética.

Con los avances tecnológicos y el conocimiento biológico que subyace en la acción de los genes, se puede decir que la Epidemiología Genética es una disciplina que combina el método epidemiológico con el genético, para estudiar la variación genética en poblaciones humanas y su relación con los cambios fenotípicos normales y patológicos. Permitiendo determinar la manera en que los factores de riesgos presentes en el medio ambiente interactúan con la constitución genética de una población determinada.

Un estudio importante dentro de la Epidemiología Genética es el de Genética Poblacional, que se ocupa de predecir las consecuencias que entrañan la estructura de la población y los fenómenos de selección y mutación para los fenotipos constitucionales y las enfermedades.

Dentro de la Genética Poblacional se realizan los estudios de análisis estadísticos F, que examinan los coeficientes de endogamia en una población. Al analizar estos coeficientes se pueden obtener descripciones de los niveles y el grado esperado de reducción de heterocigosidad en una población e igualmente se puede medir la correlación entre los genes extraídos en diferentes niveles de una población dividida jerárquicamente.

En el CNGM, los estudios sobre análisis estadísticos de Genética Poblacional se realizan utilizando diferentes software estadísticos, los cuales no cubren muchas de las funcionalidades demandadas por los especialistas y en varios casos son herramientas propietarias. Lo que hace necesario el pago de licencias en cada una de las instituciones donde se vayan a utilizar, constituyendo un gasto considerable de dinero para el país.

Estas herramientas no son pertinentes para la realización de los estudios ya que no arrojan un resultado completo sobre los mismos, teniendo los especialistas que recurrir a cálculos manuales de fórmulas matemáticas complejas para completar la investigación iniciada. Por lo que, para obtener los resultados, se emplea una cantidad de tiempo considerable y, al depender del factor humano para realizar complejas ecuaciones, se pueden cometer errores de cálculo que surgen al trabajar con juegos de datos grandes como los que se almacenan en el CNGM.

Actualmente se invierte mucho tiempo de la investigación al comprobar los cálculos realizados varias veces para no arribar a conclusiones equívocas sobre los estudios de Genética Poblacional del país. Una investigación con resultados inexactos puede traer consigo una mala inversión de dinero o el desarrollo de tecnologías innecesarias para resolver padecimientos de la población.

El CNGM, en colaboración con la Universidad de las Ciencias Informáticas (UCI), desarrolló la solución informática SEEGEN-R para realizar análisis estadísticos en estudios de Epidemiología Genética. Esta solución está basada en componentes independientes, contando hasta el momento con uno destinado a los estudios de Genética Poblacional, otro a la Epidemiología Genética y un tercero a la Epidemiología Tradicional. Con el desarrollo constante de la Genética Médica y la aparición de nuevos estudios y métodos se hace necesario ir adicionando componentes a la aplicación que permitan aumentar los análisis a realizar, para que la solución informática agrupe y centralice la mayoría de los análisis

sostenidos en la esfera científica y lograr eliminar los problemas que presenta el CNGM, contribuyendo de esta manera a la salud de la sociedad actual y futura.

Después de la problemática planteada anteriormente surge el siguiente **problema de la investigación:** ¿cómo contribuir a la realización de análisis estadísticos de coeficientes de endogamia vinculados a la Genética Médica en Cuba?

En concordancia con el problema anterior surge como **objetivo general** desarrollar un componente para análisis estadísticos de coeficientes de endogamia para la solución informática SEEGEN-R. Se tiene como **objeto de estudio:** desarrollo de aplicaciones informáticas para el análisis estadístico de la disciplina Genética Poblacional y como **campo de acción:** análisis estadístico de la disciplina Genética Poblacional en la aplicación informática SEEGEN-R.

Para cumplir el objetivo general se proponen las siguientes **tareas de investigación:**

1. Elaboración del marco teórico de la investigación sobre los análisis estadísticos de coeficientes de endogamia para definir los conceptos asociados al tema en cuestión.
2. Análisis de las aplicaciones que actualmente se utilizan para la realización de análisis estadísticos en el campo de la Genética Poblacional para identificar las funcionalidades y alcance de las mismas.
3. Descripción de la metodología, herramientas y tecnologías a utilizar para el desarrollo del componente para análisis estadísticos de coeficientes de endogamia para la solución informática SEEGEN-R.
4. Análisis y diseño de las funcionalidades del componente para análisis estadísticos de coeficientes de endogamia para la solución informática SEEGEN-R, para guiar el proceso de desarrollo de software.
5. Implementación del diseño propuesto para satisfacer los requerimientos funcionales del componente para análisis estadísticos de coeficientes de endogamia para la solución informática SEEGEN-R.
6. Realización de pruebas al componente para análisis estadísticos de coeficientes de endogamia para la solución informática SEEGEN-R, para verificar el correcto funcionamiento de los requisitos identificados.

Se definen, además, las siguientes **preguntas científicas** con el objetivo de dirigir correctamente el proceso de investigación:

- ¿Cuáles son los fundamentos teórico metodológicos relacionados a los análisis de coeficientes de endogamia?
- ¿Cuál es el estado actual de las aplicaciones existentes que realizan estudios estadísticos de coeficientes de endogamia?
- ¿Cómo diseñar un componente que incorpore estudios de análisis estadísticos de coeficientes de endogamia a la solución informática SEEGEN-R?
- ¿Cómo implementar un componente para análisis estadísticos de coeficientes de endogamia que se integre a la solución informática SEEGEN-R?
- ¿Qué técnicas de pruebas se pueden utilizar en la validación de la propuesta de solución y cómo se deben aplicar?

En la presente investigación se utilizaron los siguientes **métodos de investigación:**

Métodos teóricos

Analítico-Sintético

Procedimiento en el que se realiza el estudio de realidades complejas. Consiste en separar las partes de dichas realidades hasta llegar a los elementos fundamentales y sus relaciones. Permite componer dichos elementos en nuevas ideas elaboradas a partir de la misma base. Permite analizar la información obtenida de los sistemas que realizan análisis estadísticos en el campo de la Epidemiología Genética e identificar las necesidades actuales que no cumplen dichas aplicaciones.

Modelación

Método de la investigación muy visto en el campo del desarrollo de software que consiste en el modelado de sistemas, sus conceptos y relaciones. En el presente trabajo se evidencia su uso al modelar el proceso de desarrollo de software completo para desarrollar el componente.

CAPÍTULO 1. Fundamentación Teórica de los estudios de coeficientes de endogamia

Introducción

En este capítulo se definen los conceptos básicos asociados al tema en cuestión y se presenta una descripción de los estudios de coeficientes de endogamia que se realizan en el área de la Genética Poblacional. Se describen los juegos de datos utilizados por los genetistas y se definen los estudios realizados con éstos. Además, se caracterizan las herramientas existentes de cálculos estadísticos para estudios de Genética Poblacional, y se presentan la metodología, las tecnologías y las herramientas a utilizar para el desarrollo de la solución propuesta.

1.1 Conceptos asociados

Para un mejor entendimiento de los términos a tratar en este capítulo, se presentan conceptos básicos del campo de la genética.

Un **locus** (plural **loci**) es una posición fija en un cromosoma, como la posición de un gen o un marcador genético. Una variante de la secuencia del ADN en un determinado locus se llama **alelo**. (Mozo, 2011)

1.2 Reseña histórica de los estudios de Genética en Cuba

A inicios de la década de los 70 se creó en Cuba el Centro Nacional de Investigaciones Científicas, a raíz de esto se organizaron cursos entre los que se encontraba uno de Genética y Biofísica. Seguidamente se empezaron a reunir investigadores interesados en el tema y surgieron los primeros grupos de trabajo de genética médica en el sistema nacional de salud, a partir del desarrollo de un programa de formación en la especialidad de genética clínica. Con estas actividades se sentaron las bases para el desarrollo de la genética clínica vinculada a las aplicaciones de la genética a la salud pública.

Ya en la década de los 80, con más experiencia en el área de Genética Médica en el país, se llevaron a cabo cursos intensivos de la disciplina para preparar a especialistas en el área. Gracias a lo cual se

crearon departamentos de genética en todas las provincias para el desarrollo de servicios de genética médica de base comunitaria, universal y gratuita para la población cubana.

En julio de 2001 y hasta junio de 2003, a propuesta del Comandante en Jefe Fidel Castro Ruz, se impulsan aún más los estudios de Genética Médica extendiendo su utilidad a todas las áreas de salud del país. Se establece entonces un programa dirigido a la formación de asesores genéticos y a la creación de servicios de genética, quedando constituido el Centro Nacional de Genética Médica para desarrollar íntegramente la especialidad y controlar la red investigativa de todos los centros del país.

Por los resultados obtenidos en los estudios de Genética Médica, nuestro país se hace merecedor del reconocimiento mundial. Lo que ha permitido un amplio desarrollo de tratamientos para la cura de enfermedades detectadas gracias al análisis genético de las poblaciones. (Marcheco, 2009)

1.3 Genética Poblacional y estadísticos F

La Genética de Poblaciones humanas consiste en estudiar y describir la variabilidad existente en la composición genética de los diferentes grupos humanos desde una perspectiva evolutiva. Además, se interesa por conocer las causas determinantes de dicha variabilidad, a través del análisis de los procesos microevolutivos que operan sobre la estructura genética de una población a lo largo de las generaciones (Mesa 2010).

Para analizar y comparar las poblaciones se precisa de una metodología compleja de análisis de datos que hoy en día resulta accesible debido a la aparición de programas estadísticos cada vez más completos. Los cuales permiten medir identidad o distancia entre grupos humanos, las relaciones genéticas entre ellos, tanto numérica como gráficamente, establecer árboles de alelos/haplotipos. Como parte especial de este conjunto de métodos está el análisis filogenético, es decir, el estudio de las relaciones antecesor-descendiente en una escala de tiempo y la medición de los niveles de endogamia que ocurren en una población subdividida por subniveles (Mesa, 2010).

Para detectar diferencias significativas en las frecuencias alélicas entre las poblaciones se pueden realizar diferentes pruebas estadísticas. El genetista americano Sewall Wright introdujo un método para dividir el coeficiente de endogamia en una población subdividida en un componente dado por los apareamientos no aleatorios dentro de poblaciones y otro componente dado por la subdivisión entre poblaciones. De manera

tal que los estadísticos F pueden ser vistos como la correlación entre genes homólogos tomados de un nivel de la subdivisión, en relación con cualquier otro nivel superior. Para calcular los estadísticos F son necesarios los datos genotípicos de las poblaciones de varias generaciones (Eguiarte, y otros, 2010).

1.4 Datos genotípicos

Gracias al nivel actual de las investigaciones en el CNGM, se cuenta con una amplia variedad de datos genotípicos acumulados a lo largo de los años que permiten realizar estudios de Genética Poblacional. Esta información contiene poblaciones divididas por niveles y los locus asociados a cada individuo. Los datos están formados por individuos en cada fila y las columnas son representadas por el lugar o lugares de origen de la población y el valor de cada locus estudiado dentro de los individuos, un ejemplo se puede observar en la figura 1 donde las columnas *Locality* y *Patch* representan el origen de los individuos y L21.V y L37.J representan la información de los locus. Estos datos están recopilados por los genetistas y pueden ser cargados en la aplicación SEEGEN-R.

Locality	Patch	L21.V	L37.J
1	1	22	11
1	1	77	11
1	1	22	11
1	1	77	11
1	1	27	23
1	1	22	11
1	1	22	11
1	1	77	22
1	1	22	11
1	1	77	22
1	1	22	22
1	1	27	23
1	1	77	11
1	2	22	22
1	2	22	22
1	2	22	11
1	2	22	11
1	2	22	22
1	2	22	22
1	2	27	11

Fig. 1 Ejemplo de datos genotípicos

1.5 Estudios sobre datos genotípicos

Los estudios de datos genotípicos brindan la posibilidad de calcular diferentes factores de coeficientes de endogamia que se describen a continuación, utilizando los datos genotípicos definidos con anterioridad.

Conteos de alelos, los cuales cuentan el número de copias de alelos diferentes en cada locus y población.

Riqueza alélica, el cual estima la riqueza alélica y los conteos de alelos enrarecidos por locus y por población.

Estadísticas básicas, realiza conteos individuales, frecuencias alélicas, muestra las heterocigosidades observadas y diversidades genéticas por locus y población. Estima la media de las heterocigosidades observadas y de las diversidades genéticas en subpoblaciones (H_s), diversidades genéticas sobre las heterocigosidades observadas de toda la especie (H_t y H_t'). También calcula la cantidad de diversidad genética entre muestras (D_{st} y D_{st}') y la proporción del total de la varianza genética contenida en una subpoblación relativa a la varianza genética total también conocida como subdivisión geográfica (F_{st} y F_{st}') y por último la proporción de la varianza en la subpoblación contenida en un individuo también conocido como coeficiente de endogamia (F_{is}).

Beta (β), estima Fst (subdivisión geográfica) por poblaciones, realiza estimaciones de población sobre loci y promedios de poblaciones.

Bootstrap sobre loci de los coeficientes de endogamia (Fis), realiza la prueba bootstrap para determinar intervalos de confianza de los coeficientes de endogamia.

Bootstrap sobre loci de comparaciones por parejas de las subdivisiones geográficas (Fst), determina el límite más bajo y más alto de los intervalos de confianza y muestra componentes de varianza por locus para cada par de población.

Bootstrap sobre los intervalos de confianza para componentes de varianza, proporciona un intervalo de confianza bootstrap, sobre loci, para sumas de los diferentes componentes de varianza y estadísticos F derivados de estos.

Distancia de Cavalli-Sforza & Edwards Chord, estima la matriz de distancias genéticas entre pares de población.

Distancia euclidiana, determina la distancia euclidiana entre pares de muestras de un juego de datos genéticos (genotípicos).

Distancia euclidiana para un rasgo, determina la distancia euclidiana entre una muestra de la población y un rasgo genético.

Distancia genética de Nei, determina la distancia genética imparcial definida por Masatoshi Nei.

Estadísticas G de coeficientes de riesgo, calcula las estadísticas G para una columna de muestra y una de locus en una tabla de contingencia así como el número esperado de observaciones alélicas, las estadísticas chi-cuadradas y las estadísticas de coeficientes de riesgo.

Estadísticas G de coeficientes de riesgo sobre loci, calcula las estadísticas de coeficientes de riesgos por locus y para todos los loci de un juego de datos con varias columnas de muestras y varias columnas de locus.

Separar genotipos diploides, separa genotipos diploides de una columna de locus, en los alelos que los conforman.

Convertir genotipos diploides, convierte datos de genotipos diploides de varias columnas de locus en datos de sus alelos constituyentes, permitiendo su representación en un juego de datos de columnas o en un juego de datos de una matriz tridimensional donde la tercera dimensión contiene los alelos constituyentes.

Conteos individuales, determina la cantidad de individuos en el genotipo por locus y por población.

Frecuencias alélicas, se encarga de estimar las frecuencias alélicas para cada población y locus.

Subdivisiones geográficas (Fst) por pares, determina Fst por pares de poblaciones y los componentes de varianza.

Probar la importancia del efecto de nivel sobre la diferenciación genética, genera estadísticas G a partir de un vector que contenga la asignación de cada observación en correspondencia con el nivel que presente.

Probar la significancia del efecto de nivel de prueba en diferenciación genética, construye tablas de contingencia a partir de un vector que contenga las unidades para calcular las estadísticas G asociadas y otro vector que contenga las permutaciones a realizarse.

Probar la significancia del efecto de nivel interior en diferenciación genética entre bloque definidos por nivel exterior, construye tablas de contingencia a partir de un vector en el cual mantener las observaciones de los cambios y otro vector mediante el cual se puedan construir las estadísticas G.

Probar la significancia de las tablas de contingencia del alelo X, usando permutaciones de una unidad aleatoria dentro de unidades definidas por un vector, construye tablas de contingencia a partir de un vector mediante el cual mantener las observaciones, un vector que contenga las unidades para construir las tablas de contingencia y un vector que contenga las permutaciones a realizarse para calcular las estadísticas G.

Componentes de varianza por alelo, estima los componentes de varianza para cada alelo en un diseño aleatorio totalmente jerárquico permitiendo determinar grados de libertad para cada alelo, los coeficientes asociados a los componentes de varianza, los componentes de varianza para cada alelo, la suma de dichos componentes para cada alelo y las estadísticos F de tipos coeficientes.

Componentes de varianza general, estima los componentes de varianza para cada locus y la suma de dichos componentes sobre todos los loci.

Estimados de estadísticos F, calcula los estimados de estadísticos F de Weir y Cockerham que permiten determinar componentes de varianza de frecuencias alélicas para cada alelo entre poblaciones e individuos, componentes de varianza por locus y F_{st} y F_{is} por alelos, por locus y sobre todos los loci.

Juegos de datos de ejemplo, el sistema contiene 3 juegos de datos de ejemplo predefinidos que pueden ser utilizados con fines académicos o de aprendizaje:

- Genotipos de microsatélite de Galba truncatula 370: una colección de datos recogida del oeste de Suiza de diferentes localidades y varios parches.
- Datos de ejemplo presentado en los anexos del informe de Yang: un juego de datos compuesto de varias columnas, una de localidad y dos niveles más de subpoblaciones y una última columna de genotipos.
- Datos de 4 niveles: datos de ejemplo con 4 niveles de origen, un locus diploide y otro haploide.

Simular genotipos, el sistema contiene una función de simulación de genotipos que permite crear un juego de datos con información aleatoria sobre los locus y los orígenes de las poblaciones.

1.6 Aplicaciones que permiten realizar cálculos estadísticos

Las aplicaciones que realizan cálculos estadísticos son un conjunto de programas y subprogramas que funcionan de manera integral; es decir, para pasar de uno a otro no se necesita salir del programa y volver a él. Un paquete estadístico permite aplicar a un mismo fichero de datos un conjunto ilimitado de procedimientos estadísticos de manera sincronizada, sin salir del programa. De esta forma, la utilidad del conjunto integrado es mayor que la suma de las partes. A continuación se describen algunos de los principales software, en aras de seleccionar el más factible para darle solución al problema planteado.

1.6.1 IBM SPSS Statistics

Es una familia de software estadístico integrada que se centra en el completo proceso analítico, desde la planificación a la colección de datos y a su análisis con más de una docena de módulos plenamente integrados. El programa contiene un módulo base que contiene las capacidades claves necesarias para el proceso analítico de principio a fin y varios módulos que se anexan luego a este, dichos módulos se adquieren por separado mediante la compra. Es una aplicación muy completa que goza de muy buena opinión a nivel mundial. (IBM, 2014).

Se pueden identificar como deficiencias su alto costo, y el requerimiento de conocimientos propios de un especialista en estadísticas. Esto trae consigo la inversión de tiempo por parte de los genetistas en adquirir preparación en un campo de la ciencia distinto al que laboran y gasto de dinero para adquirir el software.

1.6.2 InfoStat

Es un software para análisis estadístico de aplicación general. Presenta una interfaz muy sencilla combinada con capacidades profesionales para el análisis estadístico y el manejo de datos. Es de carácter universitario, presentando muchas facilidades para la enseñanza de la disciplina de estadísticas. Su característica más sobresaliente es que permite conectarse con el lenguaje R mediante un intérprete integrado que permite ejecutar cualquier script del lenguaje sin dejar el ambiente de trabajo de InfoStat así como utilizar el motor de cálculo de R pero con la interfaz deseada por el programa principal (Córdoba, 2010).

El programa requiere de una licencia pagada para poder adquirirlo lo cual supone una desventaja monetaria para el país. Las funcionalidades que posee no son suficientes para realizar un estudio completo de coeficientes de endogamia.

1.6.3 PSPP

Es un programa para el análisis estadístico de muestras de datos. Para su uso son necesarios dos ficheros: uno de sintaxis y otro de datos. Realiza un análisis de los datos y devuelve los resultados por la salida estándar o en varios ficheros. Produce salidas de dos formas: tablas y diagramas. La aplicación posee integración con OpenCalc[1] y Gnumeric[2], permitiendo manejar los cálculos directamente desde hojas de cálculo, funcionalidad que permite ahorrar tiempo y evitar errores (Abuin, 2012).

El programa está orientado a desarrollar los análisis de la manera más rápida posible, cualidad por la cual destaca además de su licencia libre de pagos. Sería una alternativa a considerar si no fuera porque dispone de menos funcionalidades que SPSS y que su implementación sigue incompleta estando todavía en etapa de desarrollo (IBM, 2009).

---

[1] Apache OpenOffice Calc (OpenCalc), es el programa de hojas de cálculo de código abierto y de carácter gratuito que es parte de la suite ofimática Apache OpenOffice, utilizada en GNU/Linux y compatible con la suite ofimática Microsoft Office.

[2] Gnumeric es un programa de hojas de cálculo libre que forma parte del entorno de escritorio GNOME el cual es utilizado en algunas versiones de GNU/Linux.

1.6.4 EPIDAT

Es un programa de libre distribución que permite el manejo de forma sencilla de análisis estadísticos y epidemiológicos de datos. Resulta muy útil para el aprendizaje universitario y fortalecer la capacidad analítica de los profesionales de salud. Está programado en Java por lo que permite su utilización en varias plataformas como Windows, Linux y Macintosh (Galicia, 2014).

Se compone de 19 módulos en su última versión estable pero de éstos se han liberado muy pocos. Solamente es necesario tener conocimientos básicos de epidemiología o estadísticas para el manejo con la aplicación. No permite hacer estadística descriptiva por lo que todavía es necesario el uso de otras aplicaciones por los especialistas para completar los estudios.

1.6.5 SEEGEN-R

Es una aplicación desarrollada con el objetivo de agrupar los estudios de Epidemiología Genética que se realizan en el CNGM, brindando una mayor fluidez en la realización de sus análisis. Utiliza el lenguaje de programación R para el desarrollo de la capa de análisis estadístico y Java como lenguaje de programación para la capa de presentación.

SEEGEN-R es un sistema multiplataforma y sigue una arquitectura basada en componentes, permitiéndole ser extendido con nuevas funcionalidades. Las extensiones pueden ser desarrolladas de forma independiente y luego integradas al sistema, las más importantes son las de tipo estudio y paquete, que responden a un grupo de estudios específicos de un campo. La incorporación de una nueva extensión es relativamente fácil y se cuenta con una guía completa, así como un ejemplo para introducir el desarrollo de nuevos componentes. Pudiéndose desarrollar extensiones que se adapten a problemas específicos de diferentes campos, personalizándolas tanto como se necesite.

La aplicación está internacionalizada para usarse en los idiomas español e inglés. Está orientada a los especialistas en genética por lo que su interfaz es amigable y no requiere de conocimientos estadísticos para ser utilizada. Permite además guardar informes de los resultados de los estudios en archivos para ser almacenados.

Luego de analizar las aplicaciones actuales que realizan cálculos estadísticos, se puede concluir que ningún software existente satisface las necesidades actuales del CNGM para realizar análisis estadísticos de coeficientes de endogamia. Además se pone en evidencia la facilidad de la solución informática SEEGEN-R para agregar estudios que permitan la solución del problema planteado, por lo que se decide expandir sus funcionalidades a través de un componente que, al integrarse, incluirá en el software la realización de estudios de coeficientes de endogamia.

1.7 Metodología de Desarrollo de Software

Las metodologías de desarrollo de software son un conjunto de técnicas, procedimientos y herramientas que ayudan a los desarrolladores a realizar un nuevo producto. Indican paso a paso todas las actividades a realizar para lograr el producto informático deseado, señalan además el papel que tendrán las personas que estarán presentes en el desarrollo de las actividades. Existen dos tipos de metodologías, ágiles y tradicionales (Erazo, 2013).

Estas últimas imponen una disciplina de trabajo sobre el proceso de desarrollo del software. Para ello, se hace énfasis en la planificación total del trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar.

Por lo anteriormente planteado, se decidió escoger como metodología de desarrollo **RUP** (Rational Unified Process), que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de alta calidad de software que satisfaga las necesidades de sus usuarios finales, dentro de un horario y presupuesto predecible (Erazo, 2013).

Esta metodología crea como base los casos de uso, los cuales describen los requerimientos de la aplicación desde el punto de vista del usuario. Además define en cada momento del ciclo de vida del

proyecto, qué artefactos, con qué nivel de detalle, y por cuál rol, se deben crear. Con RUP se presentarán al cliente los artefactos al final de una fase y se valorarán las precondiciones para la siguiente.

1.8 Lenguaje de modelado y Herramienta CASE

En la presente investigación se utilizan UML como lenguaje de modelado y la Herramienta CASE Visual Paradigm, los cuales se describen a continuación:

UML (Unified Modeling Language): proporciona una forma estándar de escribir los planos de un sistema, cubriendo tanto las formas conceptuales, tales como procesos del negocio y funciones del sistema, como las concretas, tales como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes de software reutilizables (UML, 2015).

Herramienta CASE Visual Paradigm 8.0: es una herramientas CASE (Computer Aided Software Engineering) muy utilizada en la actualidad para aumentar la productividad en el proceso de desarrollo de software. Se caracteriza principalmente por su disponibilidad en múltiples plataformas (Windows, Linux), presenta licencia gratuita y comercial, es fácil de instalar y actualizar, permite la ingeniería inversa con diferentes lenguajes de programación y con sistemas gestores de bases de datos, además soporta UML como lenguaje de modelado (Venete, 2011).

1.9 Lenguajes de programación

Un lenguaje de programación es diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar; estos se componen de un conjunto de reglas sintácticas y semánticas que permiten expresar instrucciones que luego serán interpretadas. El programador es el encargado de utilizarlo para crear un conjunto de instrucciones que constituirán un programa informático (Saavedra, 2008).

Java 1.6

Es el lenguaje de programación creado por la empresa Sun Microsystems, se ha consolidado firmemente como uno de los más utilizados en la actualidad y ha demostrado ser un lenguaje muy efectivo en

programación general (García, 2001). Examinando la arquitectura Java se puede decir que sus principales características son:

Orientado a objetos: Java fomenta los diseños que conlleven a componentes reutilizables, extensibles y sostenibles. Estos componentes son lo bastante flexibles como para controlar los cambios que se puedan producir con el tiempo. Soporta las características propias de la programación orientada a objetos: clase, objeto, herencia, encapsulamiento y polimorfismo.

Interpretado: los programas de Java en lugar de ser compilados en ejecutables nativos, su código es interpretado en una Máquina Virtual de Java (JVM por sus siglas en inglés, Java Virtual Machine) y de este modo pueden ejecutarse sin tener que volver a compilarlos.

Robusto: Java es un lenguaje basado en tipos lo que evita las diferencias implícitas entre tipos, hay referencias en lugar de punteros por lo que no se puede hacer referencia a un puntero en memoria corrompiendo accidentalmente la memoria.

Seguro: garantiza la seguridad del código que se está ejecutando y evita que el código no seguro realice operaciones seguras.

De arquitectura neutral: si una empresa desarrolla un nuevo sistema operativo o un hardware completamente nuevo, no tiene que empezar desde cero sin ningún software. Con tan solo agregar la JVM en la plataforma recién diseñada, se pueden ejecutar todos los programas de Java existentes.

R 3.2.0

Proporciona un amplio abanico de herramientas estadísticas (modelos lineales y no lineales, test estadísticos, análisis de series temporales y algoritmos de clasificación y agrupamiento) y gráficas (Foundation, 2015).

Además, R puede integrarse con distintas bases de datos y existen bibliotecas que facilitan su utilización desde lenguajes de programación interpretados como Java, Perl y Python. Otra de las características de R es su capacidad para generar gráficos con alta calidad. También puede usarse como herramienta de

cálculo numérico, campo en el que llega a ser tan eficaz como otras herramientas específicas tales como GNU Octave y su equivalente comercial, MATLAB.5.

Esta investigación utiliza R por su fácil integración con otros lenguajes y su capacidad principal de realizar análisis estadísticos con rapidez y precisión con cualquier tipo de dato, además el lenguaje tiene su fundamentación en las siguientes características:

- Robustez del lenguaje.
- Constante actualización y amplia literatura disponible.
- Amplias facilidades de manipulación de bases de datos.
- Obtención de informes con un formato predeterminado y con la información que se desea.
- Facilidades gráficas.
- Facilidades para la documentación de todo el proceso de manipulación de los datos y procesamiento estadístico.

1.10 Entorno de Desarrollo Integrado

Un entorno de desarrollo integrado (IDE por sus siglas en inglés, Integrated Development Environment) es una aplicación que proporciona servicios integrales que facilitan al programador el desarrollo de software y ha sido empaquetado como tal, es decir, consiste en un editor, un compilador, un depurador y un constructor de interfaz gráfica (GUI) (EcuRed, 2012).

NetBeans 8.0

Es un entorno de desarrollo integrado, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. NetBeans IDE v8.0 se utiliza para el desarrollo del sistema en cuestión ya que es un producto libre y gratuito sin restricciones de uso, posee un potente *debugger* integrado y además viene con soporte para Java v1.6. (Carreño, 2013)

Conclusiones parciales

En el presente capítulo, luego de una revisión bibliográfica referente al estudio de la genética en Cuba, se identificaron los análisis de coeficientes de endogamia que se realizan en el CNGM. A partir de un análisis de lo escrito en la literatura científica, para identificar las aplicaciones que permiten realizar los cálculos estadísticos, se selecciona la solución informática SEEGEN-R para desarrollar el componente propuesto.

Sobre las tendencias actuales en el desarrollo de software se seleccionó RUP como metodología para guiar el proceso de desarrollo. Del estudio de las herramientas y tecnologías se definió la línea base de la arquitectura utilizando: UML como lenguaje de modelado, Visual Paradigm 8.0 como herramienta CASE, Java 1.6 y R 3.2.0 como lenguajes de programación; siendo NetBeans 8.0 el entorno de desarrollo integrado a utilizar.

CAPÍTULO 2. Análisis y diseño del componente para análisis estadísticos de coeficientes de endogamia para la solución informática SEEGEN-R

Introducción

En el presente capítulo se realiza una descripción sobre el componente propuesto, a través de la caracterización del negocio donde se definen los procesos que son objeto de informatización. Se identifican los requisitos funcionales y no funcionales, así como los actores y casos de uso del sistema a desarrollar. Además, se especifica la arquitectura que tendrá el componente, y se confeccionan los diagramas de clases del diseño y diagramas de secuencia necesarios para garantizar la correcta implementación del mismo.

2.1 Modelo del negocio

El modelo del negocio se basa en comprender la estructura y dinámica de una organización, identificar sus problemas, mejoras potenciales y derivar los requerimientos del sistema. Consiste en tener un conocimiento preciso de lo que actualmente se hace en los procesos que están considerados en el nuevo sistema (Barrios, 2010.).

2.1.1 Actores del negocio

Tabla. 1 Actores del negocio

Actor	Descripción
Consejo Científico	Indica el inicio de las investigaciones y recibe los resultados de las mismas.

2.1.2 Trabajadores del negocio

Tabla. 2 Trabajadores del negocio

Trabajador	Descripción
Especialista en Genética	Realiza los estudios de estadísticos F.

2.1.3 Diagrama de Casos de Uso del Negocio

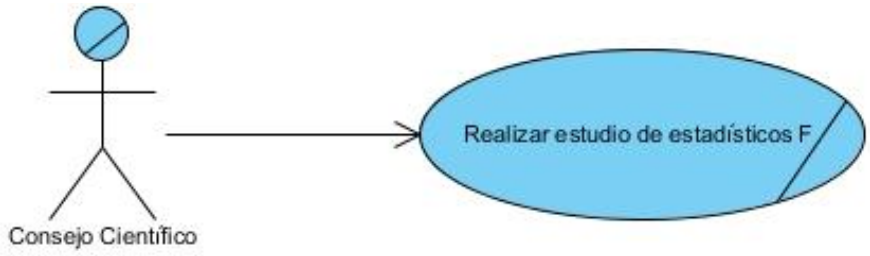


Fig. 2 Diagrama de Casos de Uso del Negocio

2.1.4 Descripción del caso de uso del negocio

Tabla. 3 Descripción textual del caso de uso del negocio

Caso de uso del negocio	**Realizar estudio de estadísticos F**
Actores	Consejo Científico (inicia)
Resumen	El caso de uso inicia cuando el Consejo Científico orienta el inicio de una investigación de estadísticos F. Seguidamente el especialista en genética realiza los estudios pertinentes y comunica los resultados al Consejo Científico, terminando de esta forma el caso de uso.
Acción del actor	**Respuesta del proceso de negocio**
1: El Consejo Científico orienta el inicio de la investigación.	
	2: El especialista en genética realiza la recolección de datos.
	3: El especialista en genética realiza los estudios de estadísticos F.
	4: El especialista en genética obtiene los resultados.
	5: El especialista en genética comunica al Consejo Científico los resultados.
6: El Consejo Científico recibe los resultados del estudio de estadísticos F.	

2.1.5 Diagrama de actividades

Los diagramas de actividades muestran el flujo de trabajo desde el punto de inicio hasta el punto final detallando muchas de las rutas de decisiones que existen en el progreso de eventos contenidos en la actividad. Estos también pueden usarse para detallar situaciones donde el proceso paralelo puede ocurrir en la ejecución de algunas actividades (Systems, 2007).

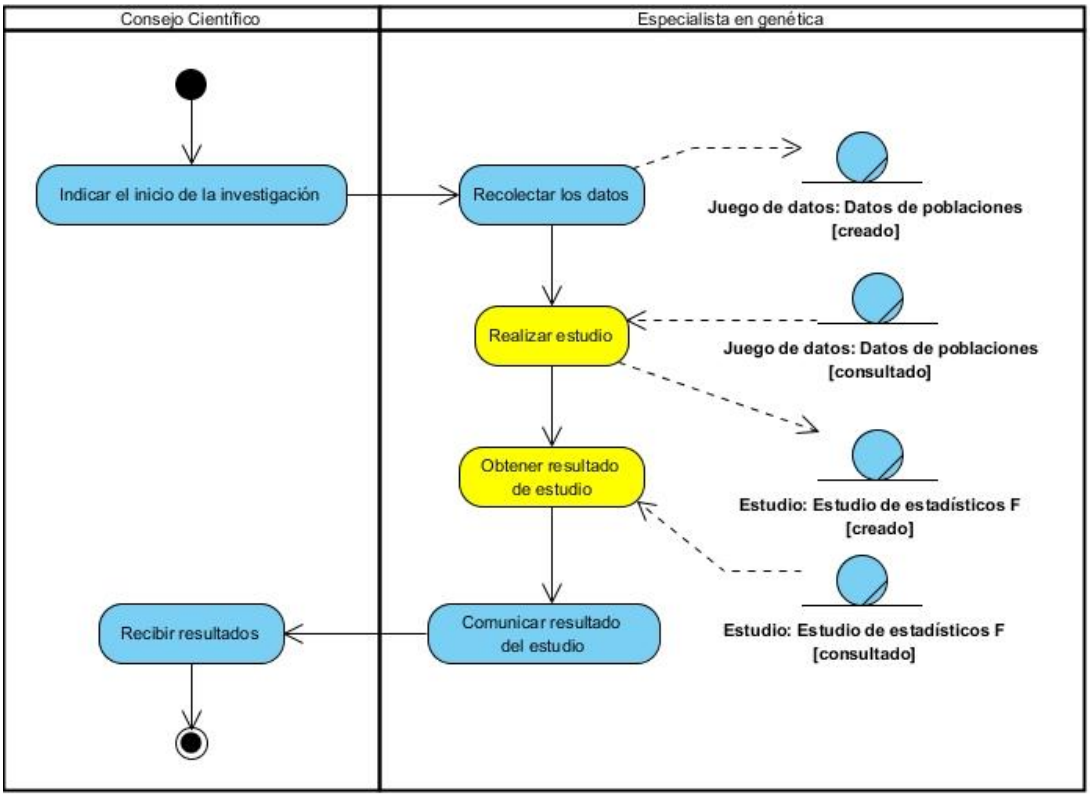


Fig. 3 Diagrama de actividades

2.1.6 Modelo de objetos del negocio

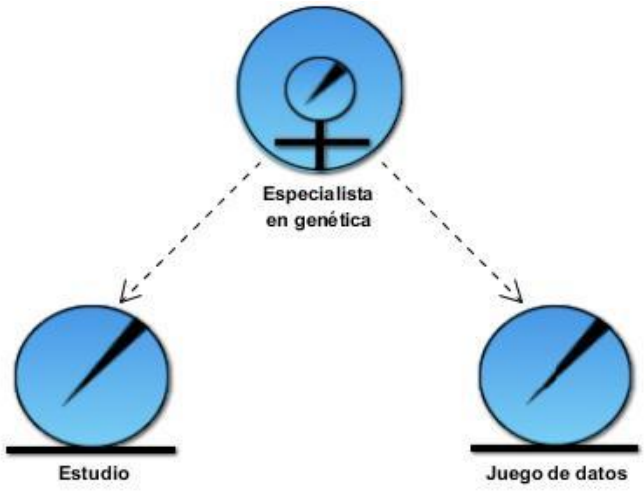


Fig. 4 Diagrama de objetos del negocio

2.2 Especificación de los requisitos del sistema

Los requisitos funcionales son una condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado, éstos definen las funciones que el sistema será capaz de realizar o sea, las transformaciones que el sistema realiza sobre las entradas para producir salidas (Pressman, 2002).

Los requisitos no funcionales tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares (Oliva, 2010).

2.2.1 Requisitos funcionales

RF1. Mostrar datos de ejemplo.

- Mostrar genotipos de microsatélite de Galba truncatula 370.
- Mostrar datos genéticos del informe de Yang.
- Mostrar datos de ejemplo de 4 niveles.

RF2. Contar cantidad de copias de alelos diferentes.

RF3. Estimar riqueza alélica.

RF4. Calcular estadísticas básicas.

RF5. Estimar el valor de β por población.

RF6. Realizar bootstrap de los coeficientes de endogamia sobre loci.

RF7. Realizar bootstrap de comparaciones por parejas de las subdivisiones geográficas sobre loci.

RF8. Realizar bootstrap sobre los intervalos de confianza para componentes de varianza.

RF9. Estimar distancia de Cavalli-Sforza & Edwards Chord.

RF10. Estimar distancia genética de Nei.

RF11. Estimar distancia euclidiana.

RF12. Estimar distancia euclidiana para un rasgo.

RF13. Calcular las estadísticas G de coeficientes de riesgo.

RF14. Calcular las estadísticas G de coeficientes de riesgo sobre loci.

RF15. Separar genotipos diploides.

RF16. Convertir genotipos diploides en datos alélicos.

RF17. Convertir genotipos diploides en datos alélicos en una matriz.

RF18. Realizar conteos individuales.

RF19. Estimar frecuencias alélicas.

RF20. Determinar subdivisiones geográficas por pares.

RF21. Simular genotipos en un modelo de islas en equilibrio.

RF22. Probar la significancia del efecto de nivel de prueba en diferenciación genética.

RF23. Probar la significancia del efecto de nivel interior en diferenciación genética entre bloque definidos por nivel exterior.

RF24. Probar la significancia de las tablas de contingencia del alelo X, usando permutaciones de una unidad aleatoria dentro de unidades definidas por un vector.

RF25. Estimar componentes de varianza por alelo.

RF26. Estimar componentes de varianza general.

RF27. Probar la importancia del efecto de nivel sobre la diferenciación genética.

RF28. Calcular estimaciones de Weir y Cockerham de estadísticos F.

RF29. Estimar distancia genética DA de Nei.

RF30. Contar cantidad de alelos diferentes.

2.2.2 Requisitos no funcionales

Apariencia o interfaz

RNF1. Los resultados de los estudios los debe mostrar siempre en la misma posición del escritorio, posibilitando mayor rapidez en la ubicación de los resultados.

RNF2. Debe existir uniformidad entre las interfaces del componente, en aras de que las vistas tengan organizado el contenido de la misma forma en cada una de ellas para reducir el tiempo de aprendizaje y familiarización del usuario.

Usabilidad

RNF3. La aplicación informática debe garantizar un acceso fácil y rápido, contando con un menú que satisfaga las necesidades de los usuarios.

RNF4. La aplicación podrá ser usada sólo por especialistas que posean conocimientos en el dominio de la especialidad de genética poblacional.

RNF5. Los botones de las ventanas secundarias deben estar ubicados en la parte inferior, en el caso de los botones Aceptar se ubicarán hacia la izquierda y los Cancelar hacia la derecha.

RNF6. Cada opción a escoger o a llenar debe tener un ToolTip con la descripción de la misma.

RNF7. Ante la ocurrencia de algún error, la aplicación informática debe mostrar al usuario el origen de este, además de otras informaciones significativas a modo de ayuda que lo guíen a su solución. También debe garantizar que el usuario pueda recuperarse lo más rápido posible de algún error cometido durante la realización de un determinado estudio de estadísticos F.

Software

RNF8. Para que la aplicación funcione es necesario tener instalado los Sistemas Operativos Linux o Windows y la máquina virtual de Java en su versión 1.6 como mínimo y el lenguaje de programación R en su versión 3.2.0.

Hardware

RNF9. Para poder utilizar la aplicación junto con la máquina virtual de Java es necesario como mínimo 256MB de memoria RAM y 200 MB de capacidad disponible en el disco duro.

Requisitos de licencia

RNF10. Las herramientas y tecnologías en que estará basada la aplicación informática deben cumplir con las licencias de software libre.

Ayuda y documentación

RNF11. El sistema deberá brindar una ayuda para el mejor entendimiento del sistema, así como para utilizar correctamente sus funcionalidades, donde se explique cada una de las acciones que se puedan realizar y cómo deben hacerse.

2.3 Definición de los casos de uso del sistema.

Un caso de uso (CU) del sistema es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento iniciado por un actor principal. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas (Landacay, 2008).

2.3.1 Actores del sistema

Tabla. 4 Actores del sistema

Actor	Descripción
Especialista en Genética	Interactúa con el sistema y es el encargado de realizar los análisis y estudios de estadísticos F.

2.3.2 Descripción de casos de uso del sistema

Teniendo en cuenta los 30 requisitos funcionales definidos, se agruparon en 8 casos de uso del sistema con el objetivo de ofrecer un mayor entendimiento del problema. Cada CU responde a una interfaz en el componente.

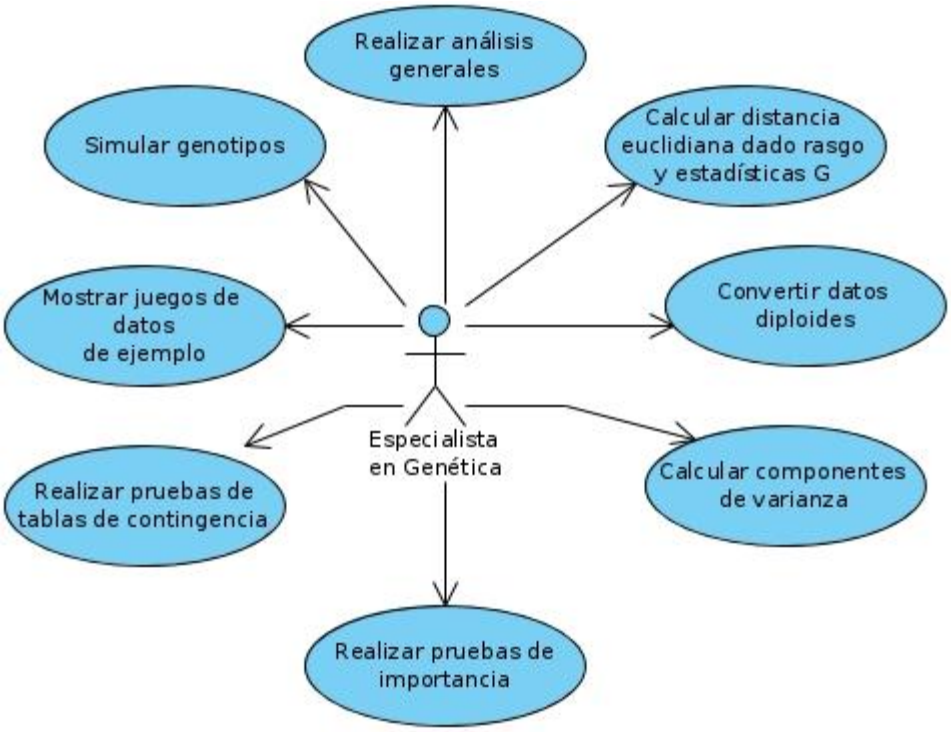


Fig. 5 Diagrama de casos de uso del sistema

Tabla. 5 Descripción de los casos de uso del sistema

Caso de Uso	Realizar análisis generales
Actores	Especialista en Genética (inicia)
Propósito	Este caso de uso se lleva a cabo con el objetivo de realizar análisis generales de coeficientes de endogamia de datos almacenados por el especialista en genética.
Resumen	El caso de uso inicia cuando el especialista en genética desea realizar algunos de los siguientes estudios: contar cantidad de copias de alelos diferentes, estimar riqueza alélica, calcular estadísticas básicas, calcular estimaciones de Weir y Cockerham de estadísticos f, estimar distancia genética de Nei, estimar distancia de Cavalli-Sforza & Edwards Chord, estimar el valor de β por población, realizar bootstrap de comparaciones por parejas de las subdivisiones geográficas sobre loci, separar genotipos diploides, calcular las estadísticas g de coeficientes de riesgo sobre loci, estimar distancia genética da de Nei, estimar distancia euclidiana, realizar conteos individuales, contar cantidad de alelos diferentes, estimar frecuencias alélicas, convertir genotipos diploides en datos alélicos,

	realizar bootstrap de los coeficientes de endogamia sobre loci, determinar subdivisiones geográficas por pares y probar la importancia del efecto de nivel sobre la diferenciación genética.
Complejidad	Media
Precondiciones	Los datos deben estar cargados en el módulo base de SEEGEN-R.
Referencia	RF4, RF2, RF3, RF5, RF6, RF7, RF9, RF10, RF11, RF14, RF15, RF16, RF18, RF19, RF20, RF27, RF28, RF29, RF30
Caso de Uso	Calcular distancia euclidiana dado rasgo y estadísticas G
Actores	Especialista en Genética (inicia)
Propósito	Este caso de uso se lleva a cabo con el objetivo de calcular distancia euclidiana dado una columna de rasgo y calcular estadísticas G de coeficientes de riesgo.
Resumen	El caso de uso inicia cuando el especialista en genética desea realizar algunos de los siguientes estudios: calcular distancia euclidiana para un rasgo y calcular estadísticas G de coeficientes de riesgo.
Complejidad	Media
Precondiciones	Los datos deben estar cargados en el módulo base de SEEGEN-R.
Referencia	RF12, RF13
Caso de Uso	Convertir datos diploides
Actores	Especialista en Genética (inicia)
Propósito	Este caso de uso se lleva a cabo con el objetivo de convertir datos diploides en alélicos.
Resumen	El caso de uso inicia cuando el especialista en genética desea convertir genotipos diploides en datos alélicos en una matriz.
Complejidad	Media
Precondiciones	Los datos deben estar cargados en el módulo base de SEEGEN-R.
Referencia	RF17
Caso de Uso	Realizar pruebas de importancia
Actores	Especialista en Genética (inicia)
Propósito	Este caso de uso se lleva a cabo con el objetivo de realizar pruebas de importancia de los efectos de nivel de prueba y de nivel interior en diferenciación genética.

Resumen	El caso de uso inicia cuando el especialista en genética desea realizar algunos de los siguientes estudios: probar la significancia del efecto de nivel de prueba en diferenciación genética y probar la significancia del efecto de nivel interior en diferenciación genética entre bloque definidos por nivel exterior.
Complejidad	Media
Precondiciones	Los datos deben estar cargados en el módulo base de SEEGEN-R.
Referencia	RF22, RF23
Caso de Uso	Realizar pruebas de tablas de contingencia
Actores	Especialista en Genética (inicia)
Propósito	Este caso de uso se lleva a cabo con el objetivo de probar la significancia de las tablas de contingencia.
Resumen	El caso de uso inicia cuando el especialista en genética desea probar la significancia de las tablas de contingencia del alelo X, usando permutaciones de una unidad aleatoria dentro de unidades definidas por un vector.
Complejidad	Media
Precondiciones	Los datos deben estar cargados en el módulo base de SEEGEN-R.
Referencia	RF24
Caso de Uso	Mostrar juegos de datos de ejemplo
Actores	Especialista en Genética (inicia)
Propósito	Este caso de uso se lleva a cabo con el objetivo de mostrar los juegos de datos de ejemplo que posee el componente.
Resumen	El caso de uso inicia cuando el especialista en genética desea mostrar algunos de los siguientes juegos de datos de ejemplo: mostrar genotipos de microsatélite de galba truncatula 370, mostrar datos genéticos del informe de yang, mostrar datos de ejemplo de 4 niveles.
Complejidad	Media
Precondiciones	Los datos deben estar cargados en el módulo base de SEEGEN-R.
Referencia	RF1
Caso de Uso	Simular genotipos
Actores	Especialista en Genética (inicia)
Propósito	Este caso de uso se lleva a cabo con el objetivo de simular un juego de datos de genotipos.
Resumen	El caso de uso inicia cuando el especialista en genética desea simular un juego

	de datos de genotipos en un modelo de isla en equilibrio.
Complejidad	Media
Precondiciones	Los datos deben estar cargados en el módulo base de SEEGEN-R.
Referencia	RF21

2.3.3 Descripción extendida del caso de uso Calcular componentes de varianza

Tabla. 6 Descripción extendida del CU Calcular componentes de varianza

Caso de Uso	Calcular componentes de varianza
Actores	Especialista en Genética (inicia)
Propósito	Este caso de uso se lleva a cabo con el objetivo de estimar los componentes de varianza de datos almacenados por el especialista en genética.
Resumen	El caso de uso inicia cuando el especialista en genética desea realizar algunos de los siguientes estudios: estimar componentes de varianza, estimar componentes de varianza general y realizar bootstrap sobre los intervalos de confianza para componentes de varianza de datos genéticos.
Complejidad	Media
Precondiciones	Los datos deben estar cargados en el módulo base de SEEGEN-R
Relaciones	RF8, RF25, RF26

Flujo Normal de Eventos		
No	**Acciones del Actor**	**Respuesta del Sistema**
1.	El especialista en genética accede a la opción “Componentes de varianza” del menú principal.	
2.		El sistema muestra una interfaz para seleccionar los estudios a realizar y llenar los datos de los mismos (ver Interfaz 1.1).
3.	El especialista en genética introduce los datos solicitados y da clic en el botón Aceptar.	
4.		El sistema valida que los datos introducidos sean correctos y los campos obligatorios no estén vacíos, si hay algún campo incorrecto o vacío ver flujo alternativo 4.a.

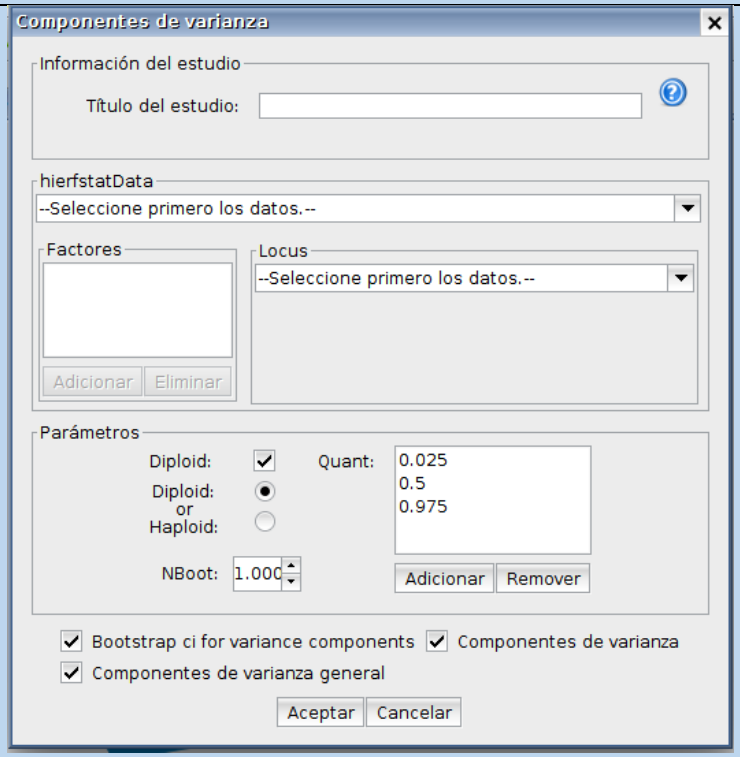
5.		El sistema muestra los resultados del estudio.
6.		Termina el caso de uso.

Flujos alternos

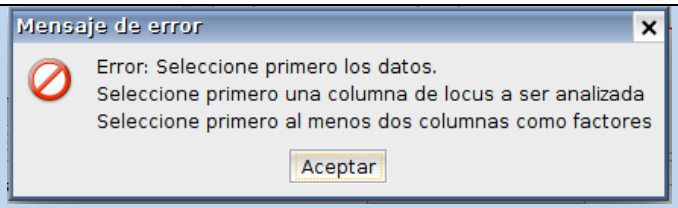
4.a Datos incorrectos y/o campos obligatorios vacíos

No	Acciones del Actor	Respuesta del Sistema
4.1		El sistema señala en rojo los campos incorrectos y muestra un mensaje de error (ver Interfaz 1.2).

Prototipo de interfaz



Interfaz 1.1 Calcular componentes de varianza



Interfaz 1.2 Mensaje de error

2.4 Arquitectura de Software

Una arquitectura de software se utiliza para estructurar y guiar el desarrollo de un sistema, con lo cual se puede satisfacer los atributos de calidad. Este proceso se refiere a las estructuras compuestas de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos. Juega un papel fundamental en las etapas del desarrollo de un software permitiendo tomar las decisiones críticas del sistema al principio de su concepción. (Addison Wesley, 2003)

N-capas

La programación por capas (N-capas) es una arquitectura cuyo objetivo primordial es la separación entre la lógica de negocio y la lógica de diseño. Al separar los componentes de la aplicación en niveles independientes, se aumenta la facilidad de mantenimiento y la escalabilidad de la aplicación (Microsoft, 2013). En dichas arquitecturas a cada nivel se le confía una misión simple.

Dos capas

La arquitectura N-capas posee una variante que presenta solamente dos capas que quedan definidas de la siguiente manera:

- Capa de Presentación (o interfaz del usuario): presenta el sistema al usuario, comunica la información y captura la información del usuario en un mínimo proceso. Esta capa se comunica únicamente con la capa de negocio (Universidad Nacional Abierta y a Distancia, 2008).
- Capa Lógica de Funcionalidad / Negocio: es donde residen los programas que se ejecutan, se reciben peticiones del usuario y se envían las respuestas tras el proceso. Es aquí donde se

establecen todas las reglas que deben cumplirse, se comunica con la capa de presentación, para recibir solicitudes y presentar los resultados.

2.5 Patrones arquitectónicos

La aplicación SEEGEN-R usa una arquitectura basada en componentes, debido a esto le permite el desarrollo de componentes de forma independiente y luego integrarlos. Dentro de sus componentes presenta:

Módulo Base: es el encargado de cargar los datos genéticos para realizar cada uno de los estudios ya sea desde un fichero guardado o creados en la misma aplicación por el genetista, también cumple con la función de mostrar en el menú correspondiente cada una de las vistas a mostrar para los especialistas y por último recibe los datos de los resultados y los muestra en pantalla.

SeegenrApiPlugin: es la biblioteca de clases encargada de conectar el módulo base con los componentes desarrollados, brindando una interfaz de conexión entre estos que permite el intercambio de información. Comunica al módulo base el menú que debe crear para acceder a las vistas del componente y los resultados de los estudios realizados. Comunica a los componentes integrados el idioma del sistema para visualizar los textos correspondientes y brinda los datos cargados previamente en el módulo base.

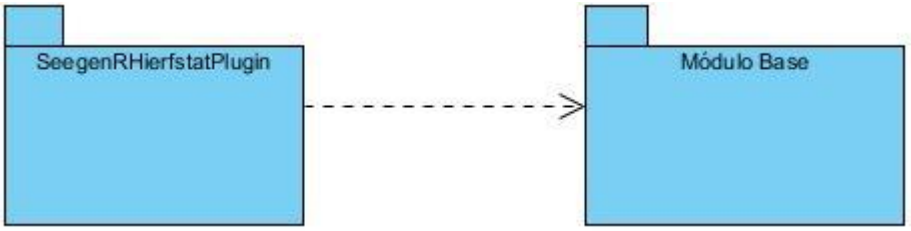


Fig. 6 Diagrama de paquetes

2.5.1 Vista lógica del sistema

La vista lógica se centra principalmente en los requerimientos funcionales, por ejemplo, los servicios que el sistema debe proporcionar a sus usuarios. El sistema se descompone en una serie de abstracciones claves y sobre estas se aplican los principios de abstracción, encapsulamiento y herencia. Esta

descomposición se realiza para potenciar el análisis funcional e identificar mecanismos y elementos de diseño comunes a diversas partes del sistema. Los diagramas se usan para representar los distintos componentes del sistema y la interacción que existe entre estos (Figueroa, 2007)**.**

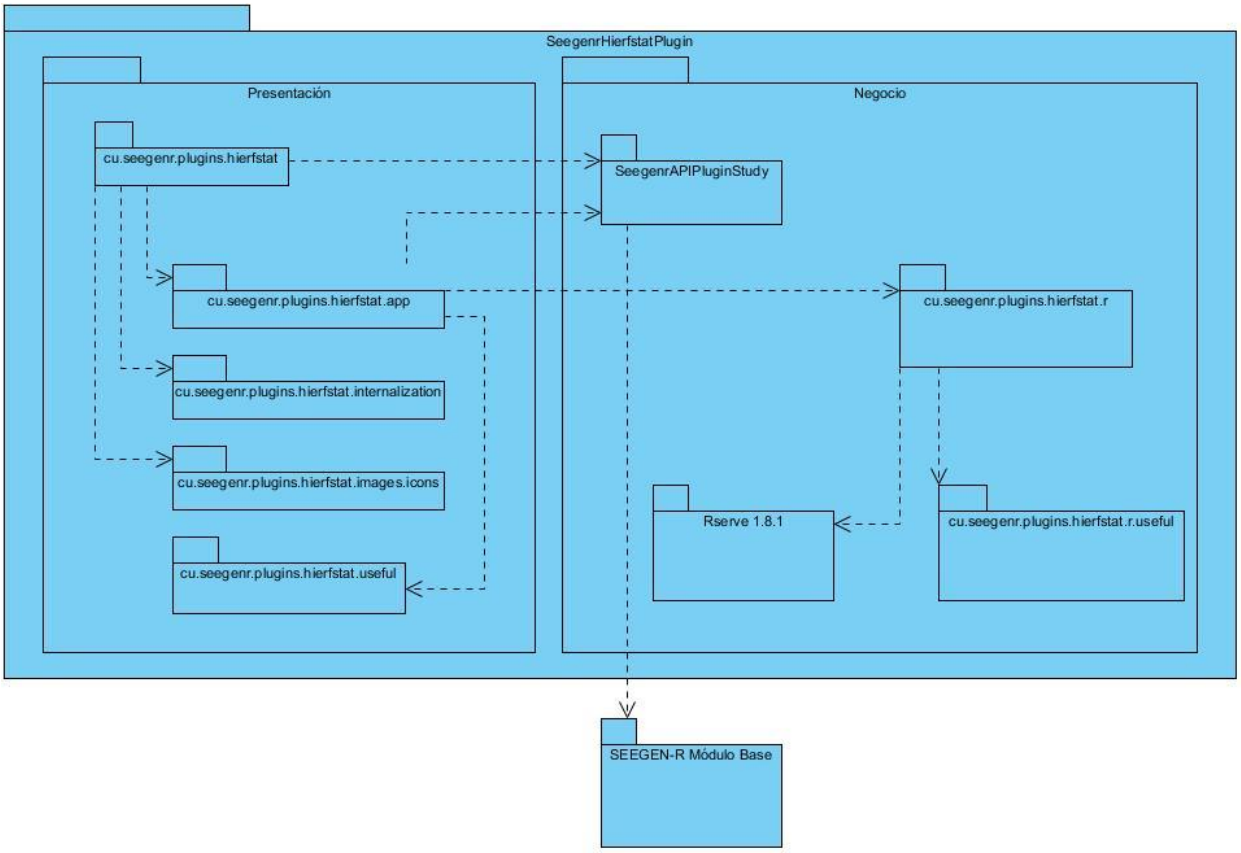


Fig. 7 Vista lógica del sistema

En la vista lógica del sistema se representan los componentes de la capa de Presentación y Negocio del patrón de arquitectura Dos-Capas. Estos componentes son:

Presentación: contiene las vistas necesarias para que el especialista en genética interactúe con la aplicación.

- cu.seegenr.plugin.hierfstat: contiene la clase encargada de crear el menú que tiene la vista de cada uno de los casos de uso descritos.

- cu.seegenr.plugins.hierfstat.app: contiene una vista por cada CU representando los estudios que se realizan.
- cu.seegenr.plugins.hierfstat.useful: contiene clases auxiliares para la implementación de las vistas.
- cu.seegenr.plugins.hierfstat.internalization: contiene las clases que definen los textos a visualizar en la aplicación en los dos idiomas soportados (español e inglés).
- cu.seegenr.plugins.hierfstat.images.icons: contiene los íconos utilizados en las vistas.

Negocio: en la capa de negocio se ejecuta la lógica de la aplicación según la orden emitida desde la capa de Presentación. Está dividida en los siguientes paquetes:

- cu.seegenr.plugins.hierfstat.r: contiene las clases que conforman los datos necesarios para hacer las llamadas a las funciones R y realiza las mismas.
- cu.seegenr.plugins.hierfstat.r.useful: contiene clases que implementan métodos auxiliares para realizar una correcta ejecución de los datos y las funciones de R.
- Rserve: es una biblioteca de clases que permite conectar R con Java mediante un servicio en ejecución en segundo plano.
- SeegenrApiPluginStudy: es una biblioteca de clases que comunica la extensión desarrollada con el módulo base del sistema, cumpliendo las funciones de entregarle los datos para realizar los estudios al componente desarrollado y recibir los resultados obtenidos y posteriormente devolverlos en el módulo base.

2.6 Diagrama de clases del diseño

Los diagramas de clase describen los tipos de objetos de un sistema, así como los distintos tipos de relaciones que pueden existir entre ellos. El diagrama de clase es creado y refinado durante las fases de análisis y diseño, estando presente como guía para el sistema (Pardo Aguilar, 1998) .

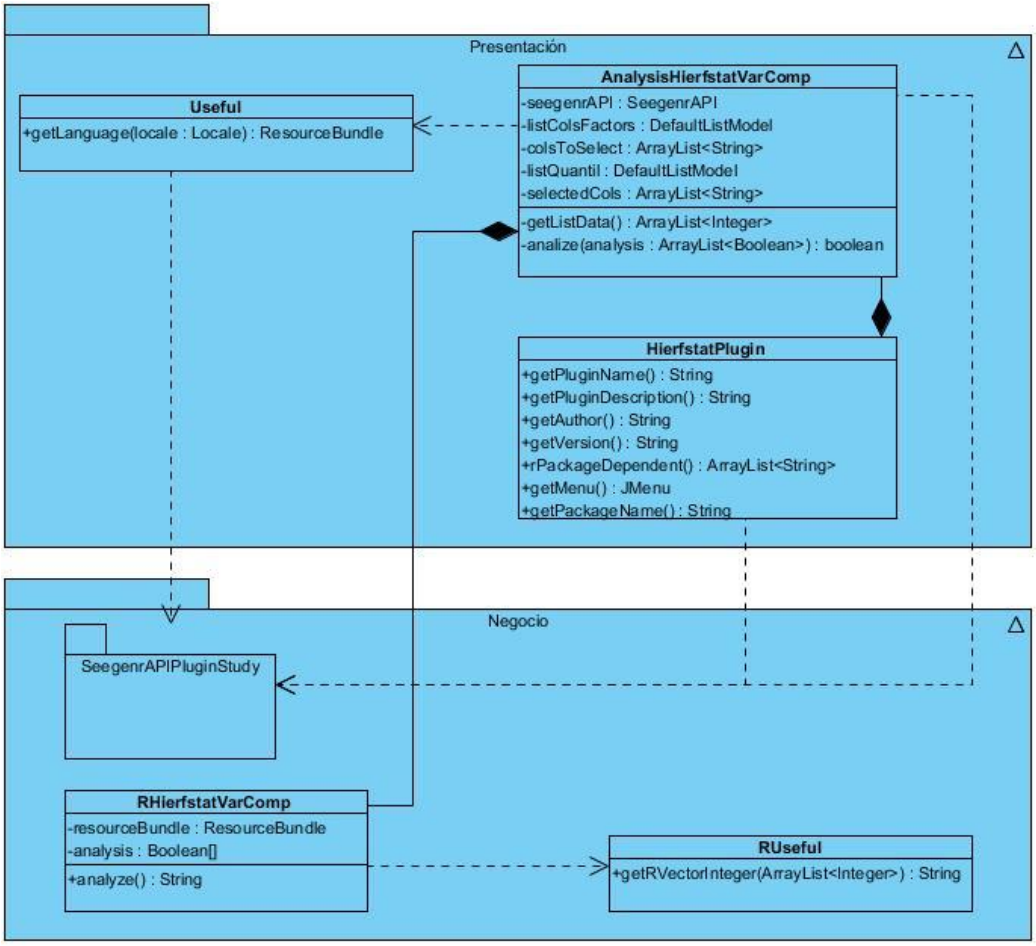


Fig. 8 Diagrama de clases del CU Calcular componentes de varianza

2.7 Descripción de las clases del diseño

HierfstatPlugin: encargada de indicar el orden del menú que se muestra en el módulo base el cual contiene la opción de iniciar el CU Calcular componentes de varianza.

AnalysisHierfstatVarComp: clase de presentación al usuario donde permite escoger el juego de datos a utilizar, los análisis a realizar y los parámetros necesarios para los mismos.

RHierfstatVarComp: crea un juego de datos en R en correspondencia con los escogidos por el usuario en la clase de presentación, realiza los estudios seleccionados con los parámetros especificados y guarda los resultados.

RUseful: contiene funciones auxiliares para construir los juegos de datos en R a partir de los obtenidos del módulo base.

Useful: contiene los métodos necesarios para obtener el idioma brindado por el módulo base.

SeegenrAPIPluginStudy: librería de clases que brinda el idioma del módulo base, los datos cargados y muestra los resultados de los análisis realizados.

2.8 Diagrama de secuencia

Este diagrama representa la secuencia de mensajes entre las instancias de clases, componentes, subsistemas o actores. El tiempo fluye hacia abajo en el diagrama y muestra el flujo de control de un participante a otro. Puede dar detalle de los casos de uso, aclarándolos al nivel de mensajes de los objetos existentes, como también muestra el uso de los mensajes de las clases diseñadas (Valencia, 2009).

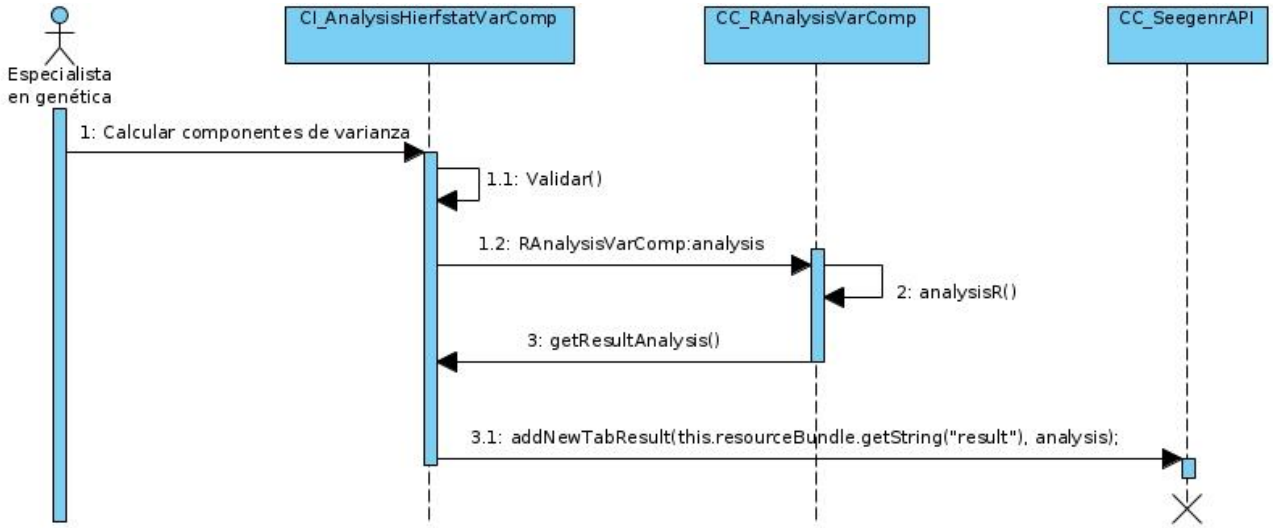


Fig. 9 Diagrama de Secuencia del CU Calcular componentes de varianza

La secuencia inicia cuando el especialista en genética selecciona la opción Calcular componentes de varianza del menú principal de la aplicación, se muestra el visual correspondiente a la clase AnalysisHierfstatVarComp, la cual obtiene el idioma del sistema mediante la clase Useful y muestra los textos en el idioma correspondiente. Después de seleccionarse los estudios a realizar y llenar todos los parámetros necesarios para éstos, se procede a validar los datos introducidos y se hace una llamada a la clase RHierfstatVarComp la cual prepara el juego de datos, los parámetros y realiza las funciones en R seleccionadas y almacena los resultados. Al concluir los análisis en R los resultados almacenados son accedidos por SeegenrAPIPluginStudy, el cual se encarga de entregárselos al módulo base y este los muestra concluyendo así la secuencia del CU.

2.9 Patrones GRASP

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (Guerrero, y otros, 2013).

Experto

Este patrón evidencia la asignación de responsabilidades a las clases, de manera tal que se pueda comprender la lógica de trabajo del sistema y la función específica de cada clase. En la solución propuesta se evidencia al crear una clase visual encargada solamente de brindarle al usuario las funciones a realizar, así como escoger sus parámetros, y otra clase para ejecutar el código en R y guardar el resultado.

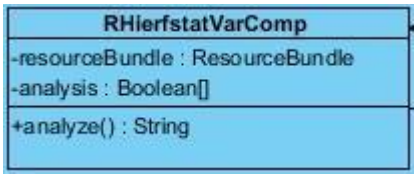


Fig. 10 Patrón Experto en la clase RHierfstatVarComp

Creador

Asigna la responsabilidad a una clase de crear otra con la cual deba interactuar. En la propuesta de solución se evidencia en cada clase visual de la capa de presentación, ya que cada una de estas es responsable de crear la clase del negocio que contiene el código en R que realiza únicamente los estudios de la clase visual que la creó.

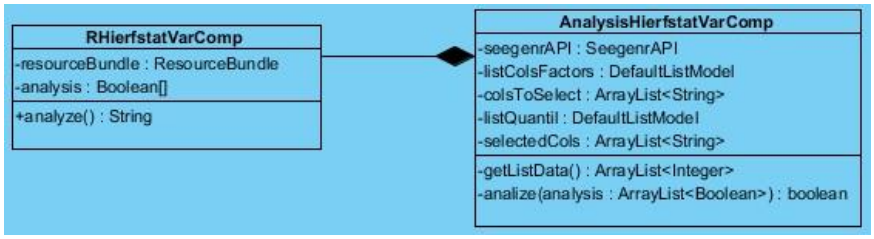


Fig. 11 Patrón Creador en el CU Calcular componentes de varianza

Bajo acoplamiento

El patrón plantea la necesidad de que cada clase dependa lo menos posible de las demás, permitiendo menos instancias de clases y llamadas necesarias para ejecutar un código. En la solución propuesta se pone en práctica en cada clase del negocio que ejecuta el código de R, que depende lo menos posible de otra clase o método externo para ejecutar el código interno.

Alta cohesión

Plantea la posibilidad de delegar código a otra clase que no sea la que lo va a emplear directamente, permitiendo la reutilización del código por varias clases que lo necesiten y evitar que una misma clase contenga muchas funciones. Se evidencia en la propuesta de solución al crear la clase r.Useful que contiene funciones necesarias para crear los datos del lenguaje R que utilizan todas las clases del negocio que implementan funcionalidades de R.

2.10 Patrones GoF

Los patrones Banda de los Cuatro (GoF por sus siglas en inglés, Gang of Four) son la base para la búsqueda de soluciones concretas a problemas comunes en el desarrollo de software y otros ámbitos

referentes al diseño de interacción o interfaces. Indican resoluciones técnicas basadas en la Programación Orientada a Objetos (POO) y ayudan a construir software basado en la reutilización de código (Kuchana, 2004).

En la propuesta de solución se evidencia el patrón Solitario (Singleton) el cual garantiza que una clase solo tiene una única instancia, proporcionando de esta forma un punto de acceso global a la misma. Este patrón se evidencia al emplear la internacionalización en la propuesta de solución, ya que para cada clase que visualice texto es necesario brindarle acceso al objeto que contiene el idioma del Sistema Operativo en el que se ejecuta la aplicación y al mismo tiempo acceso a los textos en el idioma soportado.

Conclusiones parciales

En este capítulo se realizó el análisis del negocio en cuestión, a partir del cual se identificaron 30 RF agrupados en ocho casos de uso del sistema y once requisitos no funcionales; que el componente debe cumplir para su correcto funcionamiento. Una vez realizado el diseño de la propuesta de solución se obtuvieron los diagramas de clases del diseño y de secuencia. Con el propósito de identificar la mejor forma de manejar los objetos del diseño se identificaron los patrones del diseño: Experto, Creador, Bajo Acoplamiento y Alta cohesión de la familia GRASP y Solitario de los GoF.

CAPÍTULO 3. Implementación y prueba del componente para análisis estadísticos de coeficientes de endogamia para la solución informática SEEGEN-R

Introducción

En este capítulo se especifican los estándares de codificación utilizados en la propuesta de solución, se realiza la representación de los diagramas de componentes del componente para el análisis estadístico de coeficientes de endogamia. Además se brinda una descripción de los principales métodos implementados y se reflejan las pruebas de software para comprobar la correcta implementación de cada una de las funcionalidades definidas.

3.1 Estándares de codificación

Un estándar de codificación comprende todos los aspectos a tener en cuenta en la generación de código para que este sea legible y asegurarse que todos los programadores del proyecto trabajen de forma coordinada. Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del software y para obtener un buen rendimiento (Microsoft, 2003).

En el desarrollo del componente los estándares a seguir están regidos por los definidos para el desarrollo de la solución informática SEEGEN-R, algunos de los cuales se presentan a continuación:

- Los nombres de las clases deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúsculas, manteniendo los nombres de las clases simples y descriptivos.
- En los comentarios de las clases debe aparecer el autor de esta y el objetivo de la misma.
- Inicializar las variables locales donde se declaran. La única razón para no inicializar una variable donde se declara es si el valor inicial depende de algunos cálculos que deben ocurrir.
- Utilizar nombres en plural para arreglos, tipos de datos abstractos TDA o matrices de objetos.

- Todos los métodos y clases deben estar comentariadas. Los comentarios deben ser añadidos de forma que resulten prácticos, para explicar el flujo del código y el propósito de las funciones o variables.
- Las líneas en blanco mejoran la facilidad de lectura separando secciones de código que están lógicamente relacionadas por lo que se debe usar siempre una línea en blanco entre métodos y bloques de código.

3.2 Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Representan todos los tipos de elementos de software que intervienen en la realización de componente informático (Pressman, 2002.).

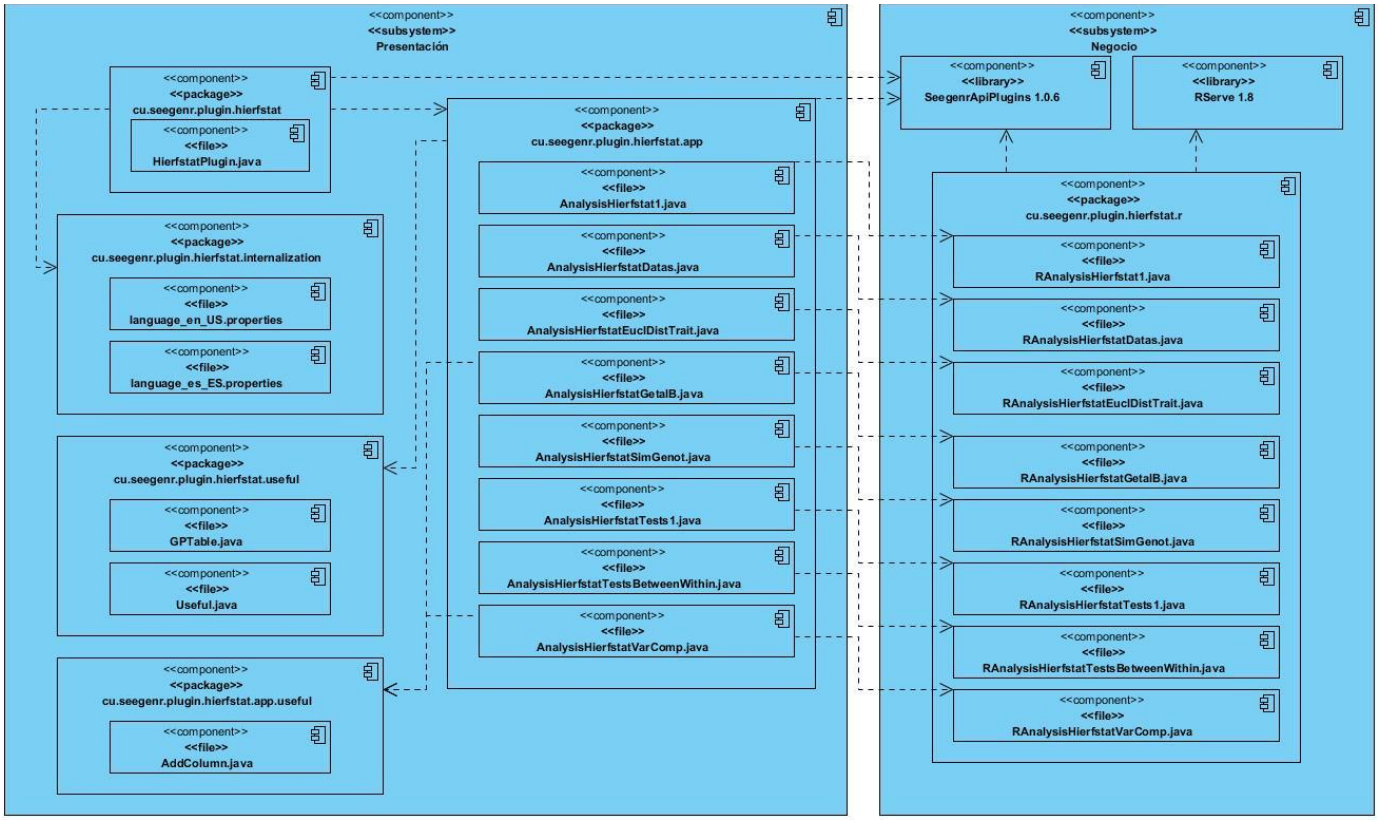


Fig. 12 Diagrama de componentes del componente para análisis estadísticos de coeficientes de endogamia para la solución informática SEEGEN-R

- cu.seegenr.plugin.hierfstat: Este paquete de componentes solo contiene la clase HierfstatPlugin.java, que se encarga de crear el menú principal de la extensión que luego se visualiza en el Módulo Base.
- cu.seegenr.plugin.hierfstat.app: Contiene las clases encargadas de mostrar las interfaces de los estudios junto con sus parámetros, una por cada CU.
- cu.seegenr.plugin.hierfstat.internalization: Contiene las clases de idiomas que contienen cada uno de los textos a visualizar en la aplicación en los idiomas soportados.
- cu.seegenr.plugin.hierfstat.useful: Contiene clases auxiliares que son utilizadas en los visuales para poder manejar los datos de genotipos y el idioma del sistema.
- cu.seegenr.plugin.hierfstat.app.useful: Este paquete contiene clases visuales auxiliares que contienen métodos necesarios por las clases visuales principales y son llamadas por las mismas.
- cu.seegenr.plugin.hierfstat.r: Contiene las clases con el código R necesario para ejecutar los estudios que se muestran en cada uno de los visuales y guardar sus resultados.
- cu.seegenr.plugin.hierfstat.r.useful: Contiene clases auxiliares con métodos para imprimir los resultados de los estudios y organizar los tipos de datos para que sean válidos en R.
- RServe: Es un conjunto de librerías que contienen funciones para conectar el lenguaje Java con R.
- SeegenrApiPlugins: Librería que garantiza la comunicación entre el módulo base de la aplicación y la solución propuesta.

3.3 Fragmentos de código fuente

A continuación se muestran imágenes del código fuente de las clases en Java que ejecutan código en R.

Un juego de datos en R se construye a partir de los datos introducidos en el lenguaje java y la utilización de los métodos auxiliares de la clase RUseful que preparan la cadena necesaria para que R acepte correctamente los datos.

```
c.eval("library(hierfstat)");
String vector = RUseful.getRVectorInteger(hierfstats).replace("-1", "NA");
String quant = RUseful.getRVectorDouble(quantiles);
c.eval("datos<-" + vector);
c.eval("nombcols<-" + RUseful.getRVectorString(colnames));
c.eval("ma<-matrix(data=datos, ncol="+cols+", nrow="+filas+", byrow=TRUE,dimnames=list(NULL,nombcols))");
c.eval("d<-data.frame(ma)");
```

Fig. 13 Fragmento de código: Juego de datos guardados en R

La función ind.count realiza conteos de individuos por locus y por población a partir de un juego de datos previamente almacenado en el lenguaje R. En el siguiente código se muestra la llamada a la función desde Java utilizando la librería de clases RServe.

```
/**
 * # Counts the number of individual genotyped per locus and population. #
 *
 * @return
 * @throws RserveException
 * @throws REXPMismatchException
 */
private String ind_count() throws RserveException, REXPMismatchException {
    String[] s = c.eval("capture.output(ind.count(d))").asStrings();
    String salida = "";
    salida+=RUseful.getAllAdo(resourceBundle.getString("indCount"))+"\n\n";
    for(String s1 : s){
        salida+= s1+"\n";
    }
    return salida+"\n";
}
```

Fig. 14 Fragmento de código: Función para conteos individuales

3.4 Pruebas

Las aplicaciones, en general cualquier mecanismo diseñado e implementado por un humano, son propensas a tener fallos, surge por tanto la necesidad de asegurar en lo posible, la calidad del producto. El único instrumento adecuado para determinar el status de la calidad del mismo es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos y presentar información sobre la calidad del producto a las personas responsables de éste. Los diferentes niveles de prueba se presentan a continuación:

- **Prueba de Sistema:** Se hace para verificar el programa final, cuando el sistema funciona como un todo y cada uno de los componentes de hardware y software están integrados en su totalidad.
- **Prueba de Integración:** Se realiza para asegurar que los componentes que son combinados para ejecutar un CU funcionen correctamente.

Pruebas de integración descendente

Para comprobar el correcto funcionamiento del componente es necesario realizar las pruebas de integración con el módulo base de la solución informática SEEGEN-R, para esto se realizan las pruebas de integración descendente en la cual se integran los módulos moviéndose hacia abajo por la jerarquía de control del sistema, comenzando por el módulo de control principal (módulo base). Los módulos subordinados al módulo de control principal se van incorporando en la estructura.

El proceso de integración se realiza en cinco pasos:

1. Se usa el módulo de control principal como controlador de la prueba, disponiendo de resguardos para todos los módulos directamente subordinados al módulo de control principal.
2. Se van sustituyendo uno a uno los resguardos subordinados por los módulos reales.
3. Se llevan a cabo pruebas cada vez que se integra un nuevo módulo.
4. Tras terminar cada conjunto de prueba, se reemplaza otro resguardo con el módulo real.
5. Se hace la prueba de regresión para asegurarse de que no se han introducido errores nuevos.

El proceso continúa desde el paso dos hasta que se haya construido la estructura del programa entero.

Para ejecutar las pruebas de integración descendente se parte del módulo base de la solución informática SEEGEN-R y se van agregando los componentes a integrar, luego se ejecuta la aplicación principal, la cual identifica el nuevo componente y lo integra al menú principal del sistema. Se accede a las opciones del componente que se desean probar, de ocurrir algún fallo en la integración de una funcionalidad, se accede al registro de errores de la aplicación y se pueden observar los mensajes ocurridos durante la ejecución del sistema.

Para la generación de los casos de prueba de sistema y de pruebas de integración descendente se utilizará la técnica de partición equivalente del método de caja negra.

Método de caja negra

Son aquellas pruebas que se realizan a la interfaz del software, sus casos de prueba están orientados a que las funciones del software sean operativas y las entradas y salidas sean correctas. Estas pruebas se realizan sin tener mucho en cuenta la estructura interna del software por lo que se centran principalmente en los requisitos de software con un conjunto de condiciones de entrada que comprueben completamente todos los requisitos funcionales.

Estas pruebas permiten encontrar:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores de rendimiento.
- Errores de inicialización y terminación.

Para desarrollar la prueba de caja negra se utiliza la **técnica de partición equivalente**, la cual divide los campos de entrada en clases de datos que tienden a ejercitar determinadas funciones y permite examinar los valores válidos e inválidos de las entradas existentes en el software.

3.4.1 Diseño de casos de prueba

Tabla. 7 Caso de prueba: Calcular distancia euclidiana dado rasgo y estadísticas G

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Calcular distancia euclidiana dado rasgo y estadísticas G	EC 1.1: El especialista en genética inserta los datos correctamente.	En este escenario el especialista en genética inserta los datos.
	EC 1.2: El especialista inserta los datos incorrectamente	Este escenario sigue la misma funcionalidad que el anterior pero verifica que todos los datos estén insertados correctamente.
	EC 1.3: El especialista en genética no inserta todos los datos necesarios	Este escenario valida que todos los datos estén insertados.

Tabla. 8 Descripción de variables

No	Nombre del campo	Clasificación	Valor nulo	Descripción
1	Nombre del estudio	texto	no	Se introduce el nombre del estudio a realizar.
2	Datos genéticos	lista desplegable	no	Se escoge el juego de datos a utilizar.
3	Muestreo	lista desplegable	no	Se escoge la columna a utilizar como muestreo de la población.
4	Rasgo	lista desplegable	no	Se escoge una columna a utilizar como rasgo de locus.
5	Diploid	booleano	N/A	Se introduce un valor de tipo lógico (Verdadero, Falso)

Tabla. 9 Matriz de datos. Calcular distancia euclidiana dado rasgo y estadísticas G

ID del escenario	Escenario	Título del estudio	Datos	Muestreo	Rasgo	Diploid	Respuesta del sistema	Resultado de la prueba
EC 1.1	El especialista en genética inserta los datos correctamente.	V[3]/ (Distancia 1)	V/ (gtrunchier)	V/ (Locality)	V/ (L21.V)	V/ (verdadero)	El sistema realiza el cálculo de la distancia euclidiana dado rasgo y las estadísticas G	Satisfactorio

[3] V: válido

EC 1.2	El especialista en genética inserta los datos incorrectamente.	V/ (Distancia 1)	I[4]/ (“no selecciona”)	V/ (Locality)	V/ (L21.V)	V/ (verdadero)	El sistema muestra un mensaje de error indicando el dato introducido incorrectamente	Satisfactorio
EC 1.3	El especialista en genética no inserta datos necesarios.	I/ (“vacío”)	V/ (gtrunchier)	V/ (Locality)	V/ (L21.V)	V/ (verdadero)	El sistema muestra un error indicando los campos que se dejaron vacíos	Satisfactorio

3.4.2 Resultado de la aplicación de las pruebas

Como parte de la ejecución de las pruebas de caja negra se realizaron 3 iteraciones de pruebas, representadas en la figura 15. En la primera iteración se detectaron 15 no conformidades, clasificadas en 12 no significativas y 3 significativas. Una vez corregidas, se procedió a realizar la segunda iteración donde se detectaron 5 no conformidades, de las cuales 4 fueron significativas y 1 no significativa. Por último, se realizó una tercera iteración en la cual se detectaron 0 no conformidades.

[4] I: inválido

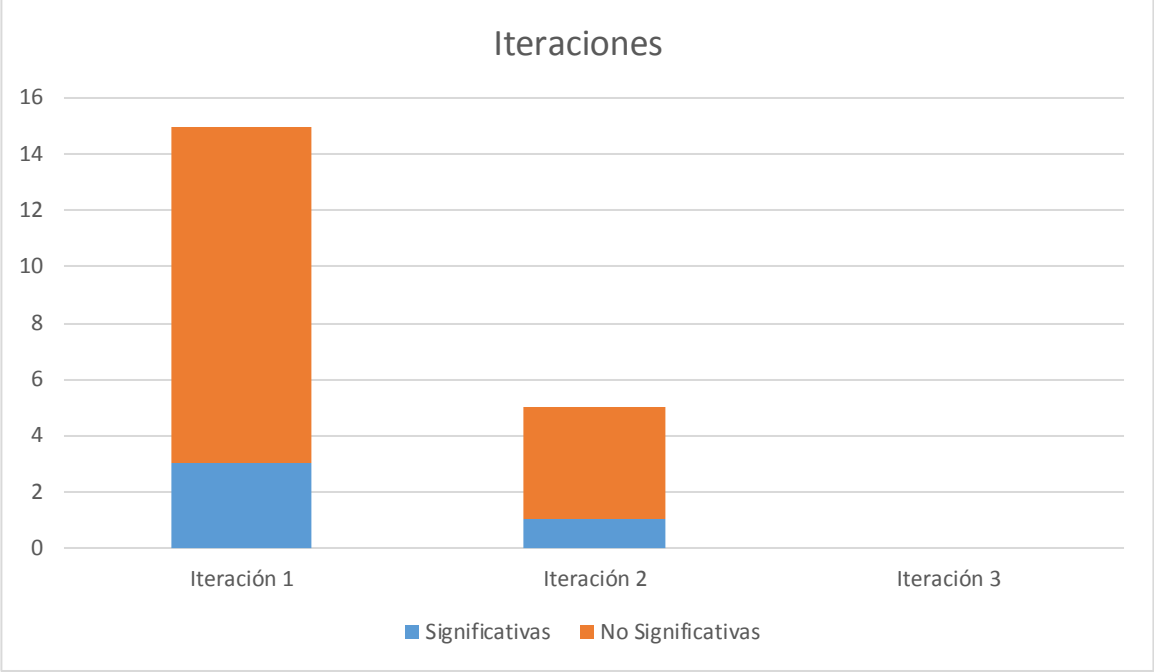


Fig. 15 Resultado de las iteraciones de las pruebas

Conclusiones parciales

En este capítulo se definieron los estándares de codificación que permiten organizar el código a lo largo de la implementación. Con el diagrama de componentes se culminó la fase de implementación, donde se especificaron las relaciones y dependencias entre cada uno de los componentes de la aplicación. Además se realizó el proceso de prueba, para el cual se desarrollaron pruebas del sistema y de integración. Para éstas se utilizó el método de caja negra mediante la técnica partición equivalente, que arrojó 15 no conformidades en la primera iteración, 5 en la segunda y 0 en la tercera; de esta manera se consigue integrar el componente a la solución informática SEEGEN-R.

Conclusiones

Como resultado de la investigación se logró implementar un componente para el análisis de coeficientes de endogamia para la solución informática SEEGEN-R; por lo que se llega a las siguientes conclusiones:

1. Con el análisis de los estudios de coeficientes de endogamia que se realizan en el CNGM se logró una correcta identificación de los requisitos funcionales y no funcionales del componente.
2. El estudio de las aplicaciones que permiten realizar los cálculos estadísticos de genética poblacional permitió la selección de la solución informática SEEGEN-R como herramienta para desarrollar la solución propuesta.
3. A partir de la realización del análisis de la solución informática SEEGEN-R se seleccionaron las tecnologías, herramientas y metodología de desarrollo que se ajustan a la situación del negocio. Se realizó un correcto diseño de las clases del sistema donde se obtuvieron los diagramas de clases del diseño y de secuencia, haciendo un buen uso de los patrones de diseño sobre la base del patrón arquitectónico Dos-capas.
4. El componente brinda un conjunto de nuevas funcionalidades a la solución informática SEEGEN-R, permitiendo el cálculo de coeficientes de endogamia y de esta forma se logra una mayor integración de los estudios que se realizan en el CNGM; dando cumplimiento de lo establecido en la especificación de requisitos de software.
5. La realización de pruebas de sistema y de integración así como la resolución de las no conformidades encontradas demostraron el correcto funcionamiento del componente implementado.

Recomendaciones

Introducir la realización de gráficas mediante el ploteo, en el componente que tiene la solución informática SEEGEN-R para visualizar las mismas.

Ampliar los idiomas soportados por el componente.

Referencias Bibliográficas

1. **Abuin, José Manuel Rojo. 2012.** Primeros pasos en SPSS. [En línea] 2012. http://humanidades.cchs.csic.es/cchs/web_UAE/tutoriales/PDF/SPSSIniciacion.pdf.
2. **Addison Wesley, L. Bass, P. Clements, R. Kazman. 2003.** *Software Architecture in Practice, 2nd Edition.* 2003.
3. **Barrios, Marcelo. 2010..** *Modelo de negocio. Universidad Americana. .* 2010.
4. **Carreño, Lohanny. 2013.** Tópicos generales de la ingeniería de software. [En línea] 2013. https://ingsoftwarei2014.wordpress.com/category/rapid-applicationdevelopment-rad-entorno-integrado-de-desarrollo-ide-ingenieria-de-software-asistida-por-computador-case/.
5. **Córdoba, Universidad Nacional de. 2010.** Software InfoStat. [En línea] 2010. http://www.infostat.com.ar/.
6. **EcuRed. 2012.** IDE de Programación. [En línea] 2012. http://www.ecured.cu/index.php/IDE_de_Programaci%C3%B3n.
7. **Eguiarte, Luis, y otros. 2010.** *Flujo génico, diferenciación y estructura genética de las poblaciones, con ejemplos en especies de plantas mexicanas.* México : s.n., 2010.
8. **Erazo, Miguel. 2013.** PREZI. [En línea] 2013. https://prezi.com/0_tqriq_lh_p/untitled-prezi/.
9. **Figueroa, Anthony. 2007.** Descripción de la Arquitectura del Sistema. Entorno de Simulación Robótico. Universidad de la Republica Montevideo. Uruguay : s.n., 2007.
10. **Foundation, The R. 2015.** R-Project. [En línea] 2015. http://www.r-project.org/.
11. **Galicia, Xunta de. 2014.** Epidat. Análisis epidemiológico de datos. [En línea] 2014. http://www.sergas.es/MostrarContidos_N3_T01.aspx?IdPaxina=62713&idioma=es.
12. **García, Angel Franco. 2001.** Programación en el lenguaje java . [En línea] 2001. http://www.sc.ehu.es/sbweb/fisica/cursoJava/Intro.htm.
13. **Guerrero, Grey Leodanys y Medel, Viltres Yamila. 2013.** *Proceso de mejora del Sistema de Gestion de Proyectos para Cuba y Venezuela.* Cuba : s.n., 2013.
14. **IBM. 2014.** IBM SPSS Statistics. [En línea] 2014. [Citado el: 6 de Abril de 2015.] http://www-01.ibm.com/software/es/analytics/spss/products/statistics/.
15. **—. 2009.** SPSS. [En línea] 2009. http://www-03.ibm.com/software/products/es/spss-stats-base.
16. **Khoury MJ, Beaty TH, Cohen BH. 1993.** *Fundamentals of genetic epidemiology.* New York: Oxford University Press : s.n., 1993.

17. **Kuchana, Partha. 2004.** *Software Architecture Desing Patterns in Java.* 2004.
18. **Landacay, Katty. 2008.** UML: Caso de uso. [En línea] 2008. [Citado el: 20 de Enero de 2015.] http://es.slideshare.net/ktyk/uml-casos-de-uso.
19. **Marcheco, Teruel Beatriz. 2009.** El Programa Nacional de Diagnóstico, Manejo y Prevención de Enfermedades Genéticas y Defectos Congénitos de Cuba: 1981-2009. [En línea] 2009. [Citado el: 4 de Abril de 2015.] http://bvs.sld.cu/revistas/rcgc/v3n2_3/rcgc1623010%20esp.htm.
20. **Mesa, Dra. Maria Soledad. 2010.** Departamento de Zoología y Antropología Física – UCM. [En línea] 2010. [Citado el: 4 de Abril de 2015.] http://pendientedemigracion.ucm.es/info/antropo/genetica.htm.
21. **Microsoft. 2013.** Información general sobre aplicaciones de datos con n capas. [En línea] 2013. https://msdn.microsoft.com/es-es/library/bb384398.aspx.
22. **—. 2003.** Revisiones de código y estándares de codificación. [En línea] 2003. https://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx.
23. **Mozo, Maria Victoria. 2011.** *Biology-Online Dictionary.* [En línea] 2011. https://www.biology-online.org/dictionary/Genetic_locus.
24. **Oliva, Ángel. 2010.** Requerimientos funcionales y no funcionales. [En línea] 2010. [Citado el: 20 de Enero de 2015.] http://es.scribd.com/doc/37187866/Requerimientos-funcionales-y-no-funcionales.
25. **Pardo Aguilar, Carlos. Garcia Penalvo, Francisco. 1998.** Diagrama de Clases en UML 1.1. 1998.
26. **Pressman, Roger. 2002..** *Ingeniería de Software. Un enfoque práctico.* 2002.
27. **Saavedra, Jorge. 2008.** Lenguajes de programación. [En línea] 2008. https://jorgesaavedra.wordpress.com/2007/05/05/lenguajes-de-programacion/.
28. **2007.** SG Buzz, Conocimiento para crear software grandioso. [En línea] Abril de 2007. http://sg.com.mx/revista/27/arquitectura-software.
29. **Systems, Sparx. 2007.** Diagrama de actividades. [En línea] 2007. http://www.sparxsystems.com.ar/resources/tutorial/uml2_activitydiagram.html.
30. **UML. 2015.** Lenguaje unificado de Modelado. [En línea] 2015. http://www.uml.org/.
31. **Universidad Nacional Abierta y a Distancia. 2008.** Sistemas Distribuidos. [En línea] 2008. http://datateca.unad.edu.co/contenidos/208017/ContLin/leccin_2_tipos_de_arquitecturas_clienteservidor.html.

32. **Valencia, Maria Eugenia. 2009.** Comportamiento del sistema. [En línea] 2009. http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/DIAGSEC_A6.pdf.
33. **Venete, Adriana. 2011.** Visual Paradigm. [En línea] 2011. http://mahara.uji.es/view/artefact.php?artefact=54800&view=4648.

Bibliografía

Abuin, José Manuel Rojo. 2012. Primeros pasos en SPSS. [En línea] 2012. http://humanidades.cchs.csic.es/cchs/web_UAE/tutoriales/PDF/SPSSIniciacion.pdf.

Addison Wesley, L. Bass, P. Clements, R. Kazman. 2003. *Software Architecture in Practice, 2nd Edition.* 2003.

Barrios, Marcelo. 2010.. *Modelo de negocio. Universidad Americana. .* 2010.

Carreño, Lohanny. 2013. Tópicos generales de la ingeniería de software. [En línea] 2013. https://ingsoftwarei2014.wordpress.com/category/rapid-applicationdevelopment-rad-entorno-integrado-de-desarrollo-ide-ingenieria-de-software-asistida-por-computador-case/.

Córdoba, Universidad Nacional de. 2010. Software InfoStat. [En línea] 2010. http://www.infostat.com.ar/.

EcuRed. 2012. IDE de Programación. [En línea] 2012. http://www.ecured.cu/index.php/IDE_de_Programaci%C3%B3n.

Eguiarte, Luis, y otros. 2010. *Flujo génico, diferenciación y estructura genética de las poblaciones, con ejemplos en especies de plantas mexicanas.* México : s.n., 2010.

Erazo, Miguel. 2013. PREZI. [En línea] 2013. https://prezi.com/0_tqriq_lh_p/untitled-prezi/.

Figueroa, Anthony. 2007. Descripción de la Arquitectura del Sistema. Entorno de Simulación Robótico. Universidad de la Republica Montevideo. Uruguay : s.n., 2007.

Foundation, The R. 2015. R-Project. [En línea] 2015. http://www.r-project.org/.

Galicia, Xunta de. 2014. Epidat. Análisis epidemiológico de datos. [En línea] 2014. http://www.sergas.es/MostrarContidos_N3_T01.aspx?IdPaxina=62713&idioma=es.

García, Angel Franco. 2001. Programación en el lenguaje java . [En línea] 2001. http://www.sc.ehu.es/sbweb/fisica/cursoJava/Intro.htm.

Guerrero, Grey Leodanys y Medel, Viltres Yamila. 2013. *Proceso de mejora del Sistema de Gestion de Proyectos para Cuba y Venezuela.* Cuba : s.n., 2013.

IBM. 2014. IBM SPSS Statistics. [En línea] 2014. [Citado el: 6 de Abril de 2015.] http://www-01.ibm.com/software/es/analytics/spss/products/statistics/.

—. 2009. SPSS. [En línea] 2009. http://www-03.ibm.com/software/products/es/spss-stats-base.

Khoury MJ, Beaty TH, Cohen BH. 1993. *Fundamentals of genetic epidemiology.* New York: Oxford University Press : s.n., 1993.

Kuchana, Partha. 2004. *Software Architecture Desing Patterns in Java.* 2004.

Landacay, Katty. 2008. UML: Caso de uso. [En línea] 2008. [Citado el: 20 de Enero de 2015.] http://es.slideshare.net/ktyk/uml-casos-de-uso.

Marcheco, Teruel Beatriz. 2009. El Programa Nacional de Diagnóstico, Manejo y Prevención de Enfermedades Genéticas y Defectos Congénitos de Cuba: 1981-2009. [En línea] 2009. [Citado el: 4 de Abril de 2015.] http://bvs.sld.cu/revistas/rcgc/v3n2_3/rcgc1623010%20esp.htm.

Mesa, Dra. Maria Soledad. 2010. Departamento de Zoología y Antropología Física – UCM. [En línea] 2010. [Citado el: 4 de Abril de 2015.] http://pendientedemigracion.ucm.es/info/antropo/genetica.htm.

Microsoft. 2013. Información general sobre aplicaciones de datos con n capas. [En línea] 2013. https://msdn.microsoft.com/es-es/library/bb384398.aspx.

—. 2003. Revisiones de código y estándares de codificación. [En línea] 2003. https://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx.

Mozo, Maria Victoria. 2011. *Biology-Online Dictionary.* [En línea] 2011. https://www.biology-online.org/dictionary/Genetic_locus.

Oliva, Ángel. 2010. Requerimientos funcionales y no funcionales. [En línea] 2010. [Citado el: 20 de Enero de 2015.] http://es.scribd.com/doc/37187866/Requerimientos-funcionales-y-no-funcionales.

Pardo Aguilar, Carlos. Garcia Penalvo, Francisco. 1998. Diagrama de Clases en UML 1.1. 1998.

Pressman, Roger. 2002.. *Ingeniería de Software. Un enfoque práctico.* 2002.

Saavedra, Jorge. 2008. Lenguajes de programación. [En línea] 2008. https://jorgesaavedra.wordpress.com/2007/05/05/lenguajes-de-programacion/.

2007. SG Buzz, Conocimiento para crear software grandioso. [En línea] Abril de 2007. http://sg.com.mx/revista/27/arquitectura-software.

Systems, Sparx. 2007. Diagrama de actividades. [En línea] 2007. http://www.sparxsystems.com.ar/resources/tutorial/uml2_activitydiagram.html.

UML. 2015. Lenguaje unificado de Modelado. [En línea] 2015. http://www.uml.org/.

Universidad Nacional Abierta y a Distancia. 2008. Sistemas Distribuidos. [En línea] 2008. http://datateca.unad.edu.co/contenidos/208017/ContLin/leccin_2_tipos_de_arquitecturas_clienteservidor.html.

Valencia, Maria Eugenia. 2009. Comportamiento del sistema. [En línea] 2009. http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/DIAGSEC_A6.pdf.

Venete, Adriana. 2011. Visual Paradigm. [En línea] 2011. http://mahara.uji.es/view/artefact.php?artefact=54800&view=4648.

Glosario de Términos

Endogamia: Se denomina endogamia al matrimonio, unión o reproducción entre individuos de ascendencia común; es decir, de una misma familia, linaje o grupo.

Fenotipos: Se denomina fenotipo a la expresión del genotipo en función de un determinado ambiente. Los rasgos fenotípicos cuentan con rasgos tanto físicos como conductuales.

Heterocigosidad: La células diploides contienen 2 copias del genoma, cada una de ellas procede de un progenitor. La posesión de 2 alelos diferentes se nombra como heterocigosidad.

Alelo: Cada uno de los genes del par que ocupa el mismo lugar en los cromosomas homólogos.

Haplotipo: Es una combinación de alelos de diferentes loci de un cromosoma que son transmitidos juntos. Un haplotipo puede ser un locus, varios loci, o un cromosoma entero dependiendo del número de eventos de recombinación que han ocurrido entre un conjunto dado de loci.

Locus (plural loci): Es una posición fija en un cromosoma, como la posición de un gen o un marcador genético.

Bootstrap: El bootstrapping (o bootstrap) es un método utilizado en estadísticas de remuestreo que se utiliza para aproximar la distribución en el muestreo de un estadístico.