

UNIVERSIDAD ANDRÉS BELLO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INFORMÁTICA  
INGENIERÍA EN COMPUTACIÓN E INFORMÁTICA



“ModelFitness: Sintonizador Semiautomático  
de Parámetros del Modelo *C-C-O*”

DIANA BISCAY MANTILLA

PROYECTO DE TÍTULO PARA OPTAR AL TÍTULO  
DE INGENIERO EN COMPUTACIÓN E INFORMÁTICA

VIÑA DEL MAR - CHILE

Diciembre, 2013

*A mi esposo y mis padres  
por el apoyo y amor infinitos.*

# Agradecimientos

Quiero agradecer a mi familia. En especial a mi esposo y a mis padres que han sido pieza fundamental en esta etapa, contribuyendo a que yo pudiera cumplir con las metas que tenía previstas en este proyecto. Su apoyo, confianza y ánimo han sido mi fuerza. También a mis amigos, a los que tengo la suerte de tener cerca, y a los que en la distancia han estado pendientes de mí y me han enviado sus buenos deseos.

Agradezco también a los profesores de la facultad de Ingeniería de la Universidad Andrés Bello de la sede Viña del Mar; a todos los que de una forma u otra han puesto de sí durante el proceso. Al profesor y tutor Oscar Pinto, por su tiempo y la confianza en mí. A la profesora Romina Torres por sus revisiones, su criterio y las tardes compartidas. Al profesor Gonzalo Guajardo por transmitirme ánimo y seguridad en todo este tiempo. Y a los profesores Claudio Paredes, Cinthya Acosta y Carlos Laguna, por sus revisiones, acalaraciones y comentarios.

Mis agradecimientos también para los investigadores del Centro Interdisciplinario de Neurociencias de Valparaíso por abrirme las puertas con el mejor de los recibimientos, lugar donde tuve la oportunidad de conocer a magníficas personas. Entre ellas quiero agradecer especialmente a los Doctores Carlos González y Alan Neely, excelentes investigadores que me permitieron formar parte de su equipo durante este año.

# Índice general

Agradecimientos	III
Lista de tablas	VIII
Lista de figuras	X
Resumen	XIII
<b>1. Introducción</b>	<b>1</b>
1.1. Contextualización del proyecto . . . . .	2
1.2. Identificación del problema . . . . .	3
1.3. Solución propuesta . . . . .	3
1.4. Objetivos . . . . .	6
1.4.1. Objetivo general . . . . .	6
1.4.2. Objetivos específicos . . . . .	6
<b>2. Fundamentación del tema</b>	<b>7</b>
2.1. Alcance del proyecto . . . . .	8
2.1.1. Supuestos del alcance . . . . .	11
2.1.2. Limitaciones al alcance . . . . .	11

2.2.	Enfoque de solución . . . . .	11
2.2.1.	Enfoque técnico . . . . .	11
2.2.2.	Enfoque funcional . . . . .	13
2.3.	Validación . . . . .	14
2.4.	Riesgos . . . . .	14
<b>3.</b>	<b>Materiales y Métodos</b>	<b>16</b>
3.1.	Organización . . . . .	17
3.1.1.	Equipo de proyecto . . . . .	17
3.1.2.	Roles y responsabilidades . . . . .	19
3.2.	Metodología de desarrollo . . . . .	20
3.2.1.	Subproyecto 1: pre-investigación . . . . .	20
3.2.2.	Suproyecto 2: proyecto informático . . . . .	23
3.3.	Cierre del proyecto . . . . .	30
3.4.	Estimaciones y planificación . . . . .	33
3.5.	Planes de proceso en el desarrollo del proyecto . . . . .	38
3.5.1.	Plan de gestión de riesgos . . . . .	38
3.5.2.	Plan de control de la configuración . . . . .	40
3.5.3.	Plan de gestión de requerimientos . . . . .	41
3.5.4.	Plan de pruebas . . . . .	42
<b>4.</b>	<b>Resultados y discusión: subproyecto de investigación</b>	<b>44</b>
4.1.	Idea . . . . .	45
4.2.	Planteamiento del problema . . . . .	47
4.3.	Desarrollo del marco teórico . . . . .	50

4.3.1.	Optimización . . . . .	50
4.3.2.	Región factible . . . . .	51
4.3.3.	Convexidad . . . . .	51
4.3.4.	Algoritmos de resolución . . . . .	52
4.4.	Visualización del alcance del estudio . . . . .	55
4.4.1.	Estrategia propuesta . . . . .	55
4.5.	Elaboración de hipótesis . . . . .	58
4.6.	Pruebas iniciales . . . . .	59
<b>5.</b>	<b>Resultados y discusión: subproyecto informático</b>	<b>62</b>
5.1.	Resultados de la fase de Análisis . . . . .	63
5.1.1.	Requisitos . . . . .	63
5.1.2.	Casos de uso . . . . .	65
5.2.	Resultados de la fase de Diseño . . . . .	72
5.3.	Resultados de la fase de Codificación . . . . .	85
5.4.	Resultados de la fase de Pruebas . . . . .	89
5.4.1.	Casos de prueba . . . . .	89
5.4.2.	Pruebas estándar . . . . .	91
5.4.3.	Defectos . . . . .	93
5.5.	Cierre del subproyecto informático . . . . .	101
5.5.1.	Balance temporal . . . . .	101
5.5.2.	Balance económico . . . . .	101
<b>6.</b>	<b>Conclusiones</b>	<b>103</b>
6.1.	Lecciones aprendidas . . . . .	106

<b>Apéndices</b>	<b>110</b>
<b>A. Transporte de electrolitos mediante canales de iones</b>	<b>111</b>
A.0.1. Células excitables . . . . .	113
<b>B. Mecanismos de compuerta: Modelo <i>C-C-O</i></b>	<b>115</b>
<b>C. Recorrido de variables, no convexidad</b>	<b>117</b>

# Índice de cuadros

3.1. Roles y responsabilidades. . . . .	19
3.2. Pruebas funcionales. . . . .	43
4.1. Intervalo de los parámetros del modelo <i>C-C-O</i> . . . . .	50
5.1. Caso de uso 01 (Primera Parte). . . . .	66
5.2. Caso de uso 01 (Segunda Parte). . . . .	67
5.3. Caso de uso 02. . . . .	68
5.4. Caso de uso 03 (Primera Parte). . . . .	69
5.5. Caso de uso 03 (Segunda Parte). . . . .	70
5.6. Caso de uso 04 (Primera Parte). . . . .	70
5.7. Caso de uso 04 (Segunda Parte). . . . .	71
5.8. Trazabilidad (CU x Req). . . . .	72
5.9. Casos de prueba para el CU1 . . . . .	89
5.10. Casos de prueba para el CU2 . . . . .	90
5.11. Casos de prueba para el CU3 . . . . .	90
5.12. Casos de prueba para el CU4 . . . . .	91
5.13. Pruebas estándares de la interfaz gráfica. . . . .	93
5.14. Pruebas estándares de la validación de datos de entrada. . . . .	94



5.15. Pruebas estándares referentes a las pruebas funcionales. . . . .	95
5.16. Pruebas estándares referentes a la verificación de informes. . . . .	95
5.17. Pruebas estándares referentes a la verificación de impresiones. . . . .	96
5.18. Primera parte de la planilla de defectos identificados. . . . .	97
5.19. Segunda parte de la planilla de defectos identificados. . . . .	98
5.20. Tercera parte de la planilla de defectos identificados. . . . .	99
5.21. Cuarta parte de la planilla de defectos identificados. . . . .	100

# Índice de figuras

1.1. Diagrama del proceso actual. . . . .	4
1.2. Diagrama del proceso con el producto ModelFitness. . . . .	5
2.1. Diagrama causa-efecto. . . . .	9
2.2. Diagrama causa-efecto de alcance. . . . .	10
2.3. Diagrama de alto nivel de arquitectura física. . . . .	12
2.4. Enfoque funcional de implementación. . . . .	13
3.1. Cliente, usuarios y equipo de proyecto. . . . .	18
3.2. Dos subproyectos . . . . .	20
3.3. Modelo secuencial para el primer subproyecto. . . . .	22
3.4. Modelo Cascada para el segundo subproyecto. . . . .	24
3.5. Acta de aceptación. . . . .	31
3.6. Acta de informe de situación. . . . .	32
3.7. HH para el primer subproyecto. . . . .	33
3.8. Planificación basada en las HH del primer subproyecto. . . . .	33
3.9. Diagrama casos de uso del producto. . . . .	34
3.10. HH en la implementación del <i>software</i> . . . . .	34
3.11. HH para el segundo subproyecto. . . . .	35

3.12. Planificación basada en las HH del segundo subproyecto. . . . .	35
3.13. Costo de la mano de obra para el proyecto. . . . .	36
3.14. Costo total del proyecto. . . . .	36
3.15. Carta Gantt general del proyecto. . . . .	37
4.1. Modelo de tres niveles . . . . .	46
4.2. Flujo de datos para el problema de optimización. . . . .	48
4.3. Conjuntos convexos y no convexos. . . . .	52
4.4. Recorrido de la variable $\gamma$ . . . . .	53
4.5. Algoritmo genético. . . . .	56
4.6. Resultados con datos exactos. . . . .	60
4.7. Recorrido de uno de los resultados. . . . .	61
5.1. Diagrama casos de uso del producto. . . . .	65
5.2. Diagrama de clases. . . . .	73
5.3. Diagrama de clases con módulo principal. . . . .	74
5.4. Descripción de la clase Model. . . . .	75
5.5. Descripción de la clase ModelBK_3. . . . .	76
5.6. Descripción de la clase Report. . . . .	76
5.7. Descripción de la clase Grapher. . . . .	77
5.8. Descripción de la clase Recommender. . . . .	77
5.9. Diagrama de la estructura de datos simulados. . . . .	78
5.10. Ejemplo de fichero de datos experimentales (carga y tiempo). . . . .	78
5.11. Estructura del fichero que se salva como reporte. . . . .	79
5.12. Diagrama de actividades. . . . .	80

5.13. Diseño de la interfaz: Recomendador. . . . .	81
5.14. Diseño de la interfaz: Refinamiento. . . . .	82
5.15. Diagrama de componentes del sistema. . . . .	83
5.16. Diagrama de despliegue del sistema. . . . .	84
5.17. Primera interfaz del sistema. . . . .	85
5.18. Graficación de las distancias que va obteniendo el GA. . . . .	86
5.19. Graficación de las distancias que va obteniendo el Fminsearch. . . . .	87
5.20. Interfaz de simulación, y de proceso de refinamiento. . . . .	88
5.21. HH estimadas y HH reales en el subproyecto 2. . . . .	101
5.22. Costo estimado y costo real de la mano de obra. . . . .	102
5.23. Costo total estimado y costo total real. . . . .	102
A.1. Membrana citoplasmática. . . . .	112
B.1. Modelo de tres niveles . . . . .	116
C.1. Recorrido de la variable $\beta$ . . . . .	117
C.2. Recorrido de la variable $\sigma$ . . . . .	118
C.3. Recorrido de la variable $\delta$ . . . . .	119
C.4. Recorrido de la variable $\gamma$ . . . . .	120
C.5. Recorrido de la variable $d\_alpha$ . . . . .	121
C.6. Recorrido de la variable $\theta 1$ . . . . .	122

# Resumen

El departamento de investigación de Estructura y Funciones de Sensores Moleculares del Centro Interdisciplinario de Neurociencia (CINV), estudia la respuesta de los canales de iones a las variaciones de voltaje en los canales de Potasio ( $K^+$ ). Este departamento ha realizado estudios empíricos determinando cómo, en laboratorio, las variaciones de voltaje afectan a los mecanismos de compuerta de estos canales; datos que les ha llevado a proponer un modelo teórico que denominaron *C-C-O*.

A pesar de que sus investigadores han podido, por un lado, obtener en laboratorio data empírica, y por otro lado, contar con un modelo que, en teoría, debería ser capaz de acercarse a los datos reales si se encuentran los parámetros adecuados, pierden tiempo crucial en intentar determinar si realmente el modelo que proponen explica o no dichos datos, pues tratan de ajustar los parámetros manualmente en la búsqueda del error mínimo entre datos empíricos y datos simulados desde el modelo. En este proceso, los resultados hasta ahora obtenidos han sido desfavorables.

Para el proceso de encontrar este error mínimo, se propone el desarrollo de un proyecto informático que resulte en una herramienta con la cual los investigadores puedan obtener de manera semiautomática conjuntos de valores de los parámetros del modelo y que se ajusten los más óptimo posible a los datos empíricos. Esta recomendación de valores de los parámetros por parte de la herramienta puede tener un gran impacto en la comunidad, pues se podrá determinar si el modelo propuesto explica o no los datos empíricos obtenidos, lo que se justifica desde las distancias que se obtengan.

Además, la herramienta permitirá modificar manualmente los valores de los parámetros, con el fin de analizar la influencia de cada uno de ellos en el funcionamiento de los mecanismos de compuerta de los canales de iones, proceso que hemos denominado refinamiento. Este

proceso y la obtención de valores de parámetros recomendados, pudiera llevar a la propuesta de nuevos modelos teóricos, con los respectivos avances en conocimientos del área y mayores publicaciones del CINV.

La validación del proyecto como producto se realizó midiendo la satisfacción del cliente, a través de una encuesta, en la tarea de encontrar el conjunto de parámetros que les permita el mejor ajuste a los datos, así como a su refinamiento manual. El cliente es el subdirector del CINV e investigador del departamento de Estructura y Función de Sensores Moleculares.

# Capítulo 1

## Introducción

## 1.1. Contextualización del proyecto

El Centro Interdisciplinario de Neurociencia de Valparaíso (CINV) es un centro de investigación en neurociencia donde se estudia el funcionamiento del sistema nervioso a diferentes niveles, desde el funcionamiento de proteínas presentes en las células hasta la dinámica que siguen las redes neuronales. El centro realiza publicaciones científicas a partir del avance en los conocimientos del área que obtienen en sus investigaciones.

Entre sus departamentos de investigación se encuentra el de Estructura y Funciones de Sensores Moleculares, que estudia la respuesta de los canales de iones a las variaciones de voltaje en los BK<sup>1</sup> de humanos expresados en ovocitos de ranas del género *Xenopus*. Estos canales de iones dependientes del voltaje están presentes en las membranas de células electroexcitables de los organismos vivos participando en el transporte de sustancias entre el interior y exterior celular (para mayor detalle acerca del transporte mediante los canales de iones ver Apéndice A).

Este departamento ha realizado estudios empíricos de laboratorio para determinar cómo las variaciones de voltaje afectan a los mecanismos de compuerta de los canales de potasio ( $K^+$ ). Los canales de iones están compuestos por cuatro dominios protéicos (subunidades  $\alpha$ ) y pueden tener acopladas otras proteínas auxiliares (subunidades  $\beta_1$ ). Para estudiar los efectos de las diferentes subunidades  $\beta$  han medido la corriente de compuerta ( $I_g$ ) y la corriente iónica del canal ( $I_k$ ), en los canales donde subunidades  $\alpha$  se encuentran solas, y donde están asociadas con  $\beta_1$ ,  $\beta_{2IR}$ ,  $\beta_{3b}$  y  $\beta_4$  ( $\alpha/\beta_1$ ,  $\alpha/\beta_{2IR}$ ,  $\alpha/\beta_{3b}$  y  $\alpha/\beta_4$ ).

Los datos empíricos han sido obtenidos mediante la técnica de *patch clamp*, que consiste en colocar una pipeta de punta microscópica en una región (“parche”) de la membrana, que puede ser lo suficientemente fina como para atrapar en su interior a un sólo canal (aunque en estos estudios se utilizaron parches más grandes donde se registró la actividad de miles de canales simultáneamente). Posteriormente, se realiza una succión y mediante distintas maniobras (romper el área seccionada o mantenerla unida a la célula) se fija y aplica un voltaje, que determina un potencial de membrana ( $E_m$ ), a través de un microelectrodo que se encuentra en el interior de la pipeta. Al aplicar el voltaje se registra la actividad eléctrica del canal a través del mismo microelectrodo. Con estos registros lo que se obtiene es la variación de corriente en el tiempo a un  $E_m$  dado.

---

<sup>1</sup>Canales de potasio ( $K^+$ ) dependientes de voltaje y de la presencia de calcio ( $Ca^{++}$ ) intracelular.



## 1.2. Identificación del problema

Los datos empíricos obtenidos por el CINV, de las respuestas de los canales de  $K^+$  a las variaciones de voltaje, no son explicados a partir de modelos teóricos anteriores propuestos en la comunidad científica, por lo que el centro propone un nuevo modelo teórico del funcionamiento de los mecanismos de compuerta de estos canales, que denominaron *C-C-O* (para mayor detalle acerca del mecanismo de compuerta propuesto con el modelo *C-C-O*, ver Apéndice B).

El CINV cuenta entonces con data empírica de las variables fisiológicas de las respuestas de los canales de iones y con un modelo teórico para explicar éstos datos; pero se necesita, para demostrar que los datos son explicados por el modelo, que las distancias entre la simulación de datos fisiológicos desde el modelo y los datos fisiológicos empíricos sea lo suficientemente pequeña, con valores menores a  $1 \cdot 10^{-2}$ . En la búsqueda de esta distancia, los investigadores del centro pierden tiempo crucial tratando de ajustarlo mediante un proceso manual, el cual de encuentra representado en la figura 1.1. Con este proceso, los resultados hasta ahora obtenidos han sido desfavorables, pues no se han obtenido valores menores a los deseados, y se desconce si los valores ya obtenidos se pueden mejorar.

## 1.3. Solución propuesta

Se propone como proyecto el desarrollo de una herramienta que permita a investigadores dado un conjunto de datos obtenidos de manera empírica y el modelo *C-C-O* que en teoría los explica, encontrar de manera semiautomática el conjunto de valores para los parámetros del modelo que permiten ajustar lo más óptimo posible los datos simulados desde el modelo a los datos empíricos, proceso que denominaremos: recomendación de parámetros.

Esta herramienta permitirá además, realizar análisis de sensibilidad de los parámetros a través de su modificación manual de manera de visualizar el impacto que tienen los cambios de éstos en el ajuste final, proceso que denominaremos: refinamiento de parámetros. Por último, permitirá salvar y cargar instancias de los parámetros recomendados por la herramienta e instancias de los parámetros especificados por el usuario en el proceso de refinamiento. En la Figura 1.2 se representa el proceso que se llevaría a cabo con este producto.

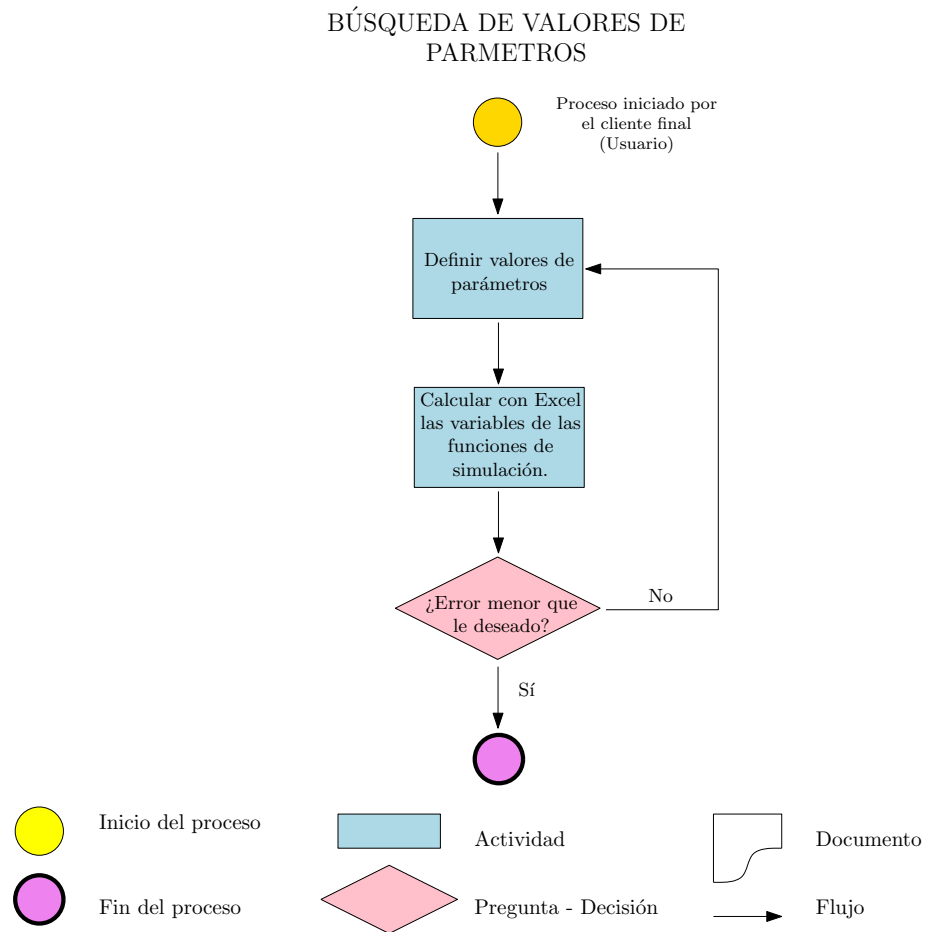


Figura 1.1: Diagrama del proceso actual.

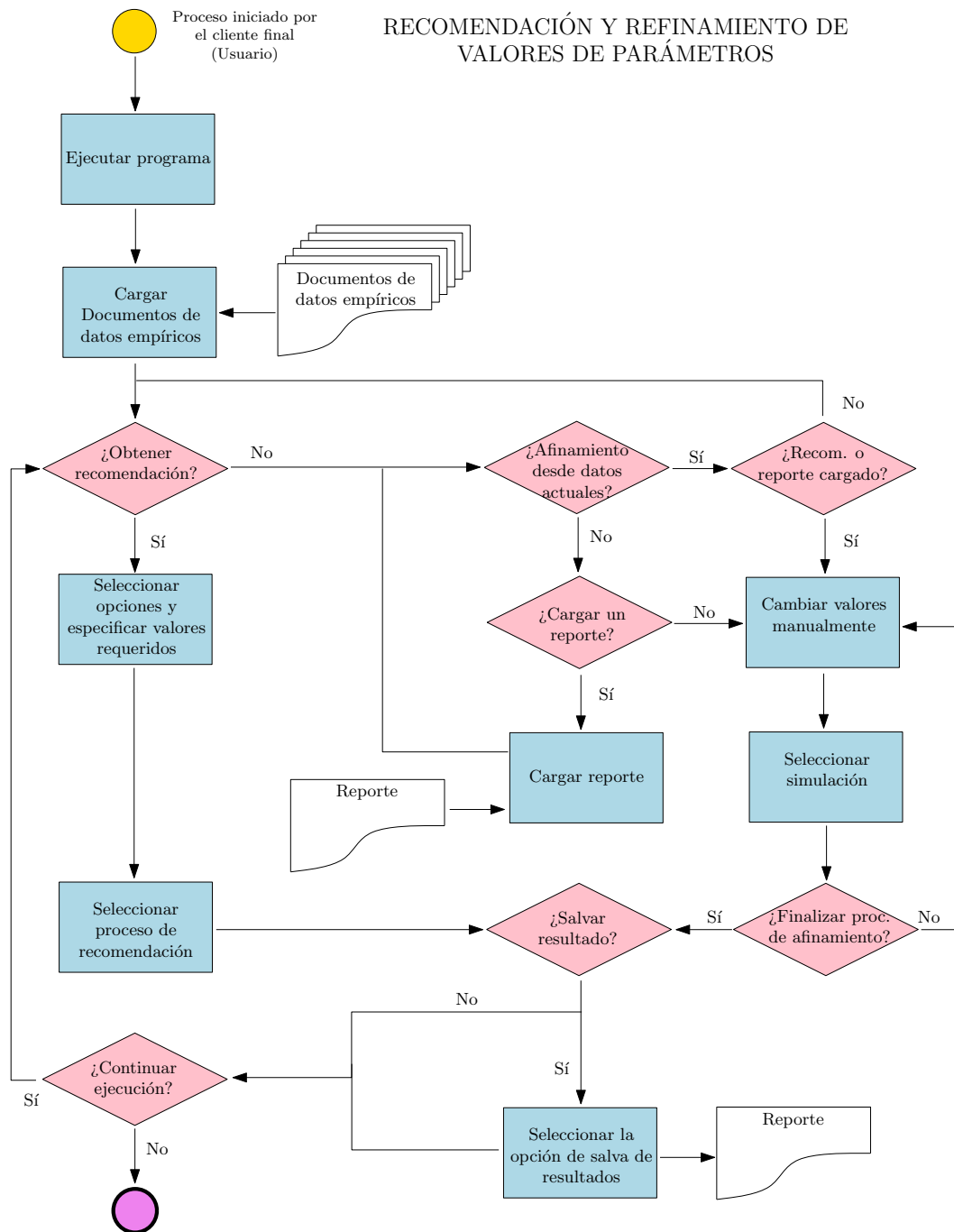


Figura 1.2: Diagrama del proceso con el producto ModelFitness.

## 1.4. Objetivos

### 1.4.1. Objetivo general

Construcción de una herramienta, mediante el desarrollo de un proyecto informático, que recomiende y permita afinar parámetros del modelo teórico *C-C-O* para el departamento de investigación de Estructura y Funciones de Sensores Moleculares del CINV, con el fin de determinar si los datos empíricos son explicados o no desde dicho modelo.

Aunque la solicitud nace de este departamento, se pretende abrir a la comunidad científica para que pueda ser utilizado por investigadores del área, ya sean del CINV o no.

### 1.4.2. Objetivos específicos

- Definir y modelar el problema de optimización multi-objetivo que plantea la recomendación de valores de los parámetros.
- Proponer una estrategia de resolución para el mismo.
- Recomendar valores para los parámetros del modelo que correspondan a la distancia mínima encontrada entre los datos empíricos y los datos simulados desde dicho modelo.
- Afinar los valores de los parámetros permitiendo visualizar la implicancia de cada uno de ellos en el modelo.
- Salvar y cargar instancias (reportes) del resultado de la recomendación de parámetros, y del resultado del refinamiento.

## Capítulo 2

### Fundamentación del tema

## 2.1. Alcance del proyecto

En la Figura 2.1, se representa el Diagrama causa-efecto del problema que presenta este departamento, a saber, la pérdida de tiempo en determinar si existe un conjunto de valores de los parámetros del modelo que permita explicar los datos empíricos. Que expliquen los datos significa que, con esos valores, la distancia entre los datos empíricos y los datos simulados desde el modelo sea menor que  $1 \cdot 10^{-2}$ . Los causales de este problema actual son:

- El proceso manual, mediante prueba y error, que se realiza en la búsqueda de valores para los parámetros que representen la distancia mínima o una cercana a la mínima.
- Los materiales tecnológicos en fisiología celular son deficientes para el esclarecimiento del mecanismo de compuerta de los canales de iones.
- No existe una herramienta que busque la distancia mínima, o una cercana a la mínima, para recomendar valores de parámetros para el modelo específico que se desea estudiar.
- Los datos empíricos actuales no son explicados por modelo teóricos anteriores propuestos en la comunidad científica.

Con esta propuesta de proyecto se resuelve realizar un proceso semiautomático en la búsqueda de valores de los parámetros que sustituya al proceso manual que se realiza actualmente y el que no exista herramienta que recomiende valores a los parámetros para el modelo específico *C-C-O*. Además, se podrá determinar si existe un conjunto de valores de los parámetros del modelo que permita explicar los datos empíricos. Teniendo en cuenta lo anterior, en la figura 2.2 el color verde representa los elementos de los que el proyecto se hará cargo, el color rojo lo que queda fuera del alcance del mismo.

Por lo tanto, la herramienta permitirá a investigadores obtener los valores de los parámetros del modelo *C-C-O* que permiten explicar mejor un conjunto de datos obtenidos de manera empírica. Además, a través del refinamiento de parámetros, permitirá analizar la influencia de cada uno de ellos en el funcionamiento de los mecanismos de compuerta de los canales de iones (determinando, por ejemplo, si algunos parámetros son redundantes), lo que pudiera llevar a la propuesta de nuevos modelos teóricos. Todo lo anterior, podría llevar a avances en conocimientos del área reflejados en mayores publicaciones científicas del CINV.

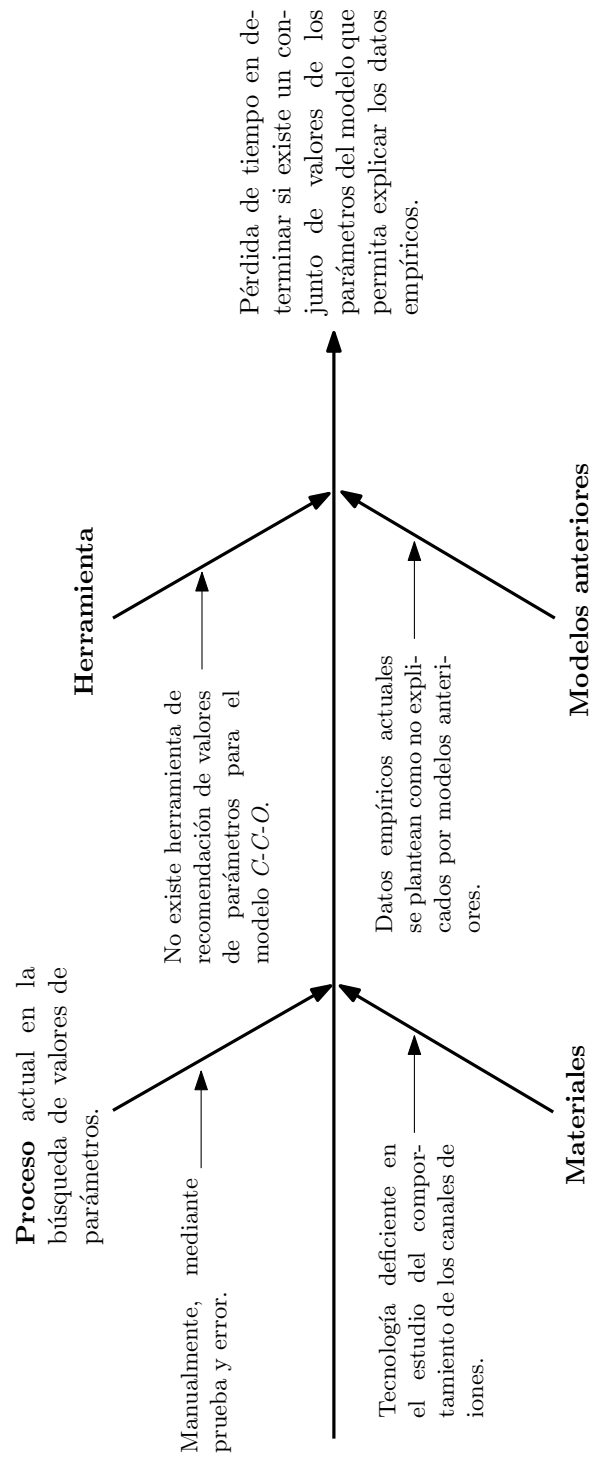


Figura 2.1: Diagrama causa-efecto.

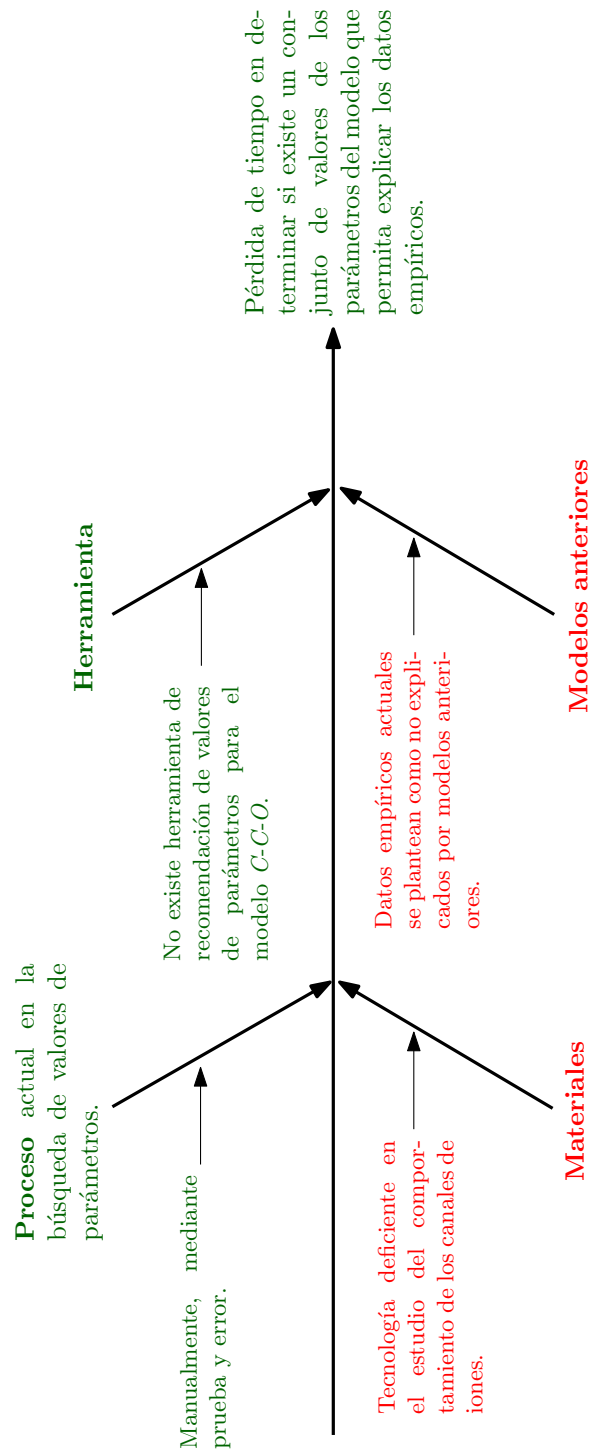


Figura 2.2: Diagrama causa-efecto de alcance. En verde: de lo que el proyecto se hará cargo. En rojo: lo que queda fuera del alcance del proyecto.



### 2.1.1. Supuestos del alcance

Se tienen presente los siguientes supuestos:

- **Conocimientos:** que el equipo tiene los conocimientos para la realización del proyecto, o la capacidad para integrar estos conocimientos.
- **Cumplimiento:** que el Jefe de Proyecto garantizará el cumplimiento de los plazos y el ajuste al presupuesto previsto para el proyecto, y en general el cumplimiento de todas las responsabilidades que se le han asignado.
- **Recursos:** que se contarán con los recursos necesarios para la realización y presentación de todos los entregables y para el desarrollo del proyecto.

### 2.1.2. Limitaciones al alcance

El proyecto es específico para un modelo teórico, el modelo *C-C-O*. En este sentido, si en un futuro los investigadores del centro tienen interés en hacer los ajustes y refinamientos para otros modelos, ya sean modelos anteriores o futuros, será necesario primero la implementación del mismo y luego los cambios requeridos en la herramienta para la inclusión del nuevo modelo. No obstante, el producto permitirá la inclusión de estos módulos con el mínimo de cambios necesarios en el resto de los módulos ya funcionales.

Por otro lado, el proyecto no resuelve el que los materiales (en cuanto a tecnología) no permitan esclarecer el funcionamiento de los mecanismos de compuerta de los canales, ni resuelve el que los datos no sean explicado por modelo teóricos anteriores.

## 2.2. Enfoque de solución

### 2.2.1. Enfoque técnico

El sistema será un programa ejecutable en el computador sin necesidad de acceso a internet. El programa cargará los datos desde archivos con extensión *.txt*, y devolverá documentos de

reporte con extensión .mat que luego podrán ser cargados. Como se muestra en la figura 2.3, desde el conjunto de ficheros y desde un modelo teórico implementado, el sistema realizará un proceso cuyo resultado es la recomendación de un conjunto de valores de parámetros, que representan los valores cuya simulación mantiene la menor distancia encontrada respecto a los datos empíricos. Además, los valores recomendados se pueden modificar manualmente y realizarse su posterior simulación.

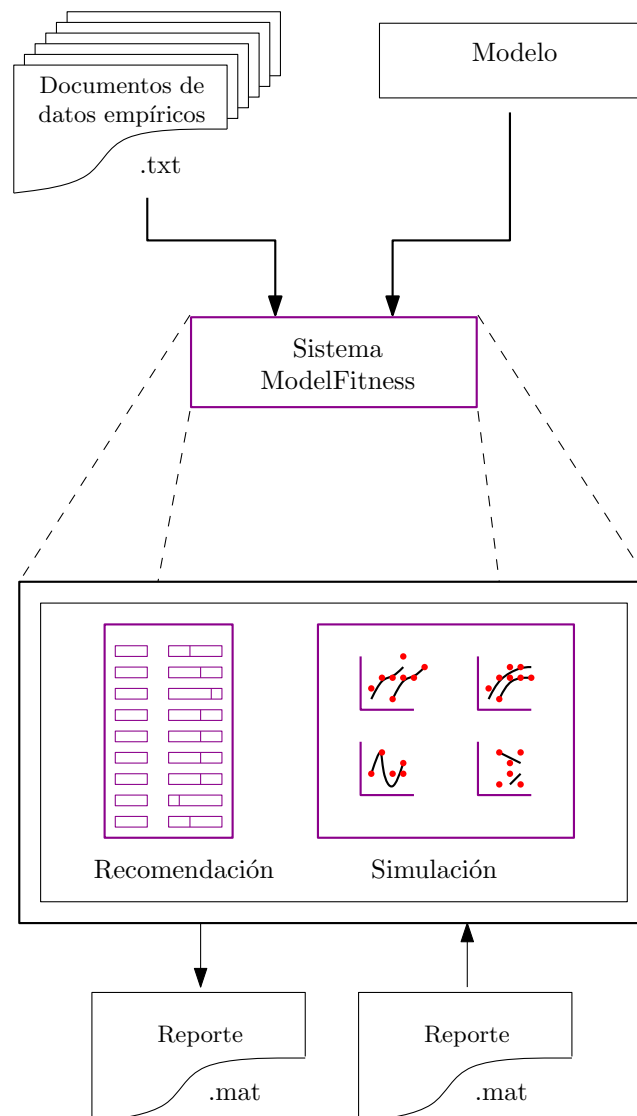


Figura 2.3: Diagrama de alto nivel de arquitectura física.

## 2.2.2. Enfoque funcional

La interfaz con el usuario permitirá que, inicialmente, se pueda seleccionar si se desea una recomendación de parámetros del modelo teniendo en cuenta datos empíricos, o si se desea hacer un refinamiento de parámetros. En ambos casos, se solicitará la selección de diferentes opciones y la carga de los ficheros de datos empíricos.

La recomendación conlleva a un proceso de ajuste donde se le va mostrando al usuario los errores que se van encontrando entre los datos simulados a partir del modelo y los datos empíricos. El proceso de refinamiento se puede comenzar sin valores recomendados, siendo el usuario quien los define, o realizarse al finalizar el proceso de recomendación partiendo de los valores recomendados, o con los valores cargados desde un fichero de reporte que fue salvado anteriormente. Este proceso de refinamiento, como se fue mencionado anteriormente, se realizaría con el fin de analizar la influencia de los diferentes parámetros en el comportamiento de los canales.

Además de graficar el error durante el proceso de ajuste, al finalizar el proceso se grafican las diferentes funciones de datos fisiológicos simulados para un conjunto de valores de parámetros y los datos fisiológicos empíricos. Por último, el usuario podrá guardar y cargar reportes con los resultados del proceso de ajuste, y guardar y cargar reportes en el proceso de refinamiento, cada vez que lo requiera. Todas las funcionalidades anteriores quedan resumidas en los módulos presentados en la figura 2.4.

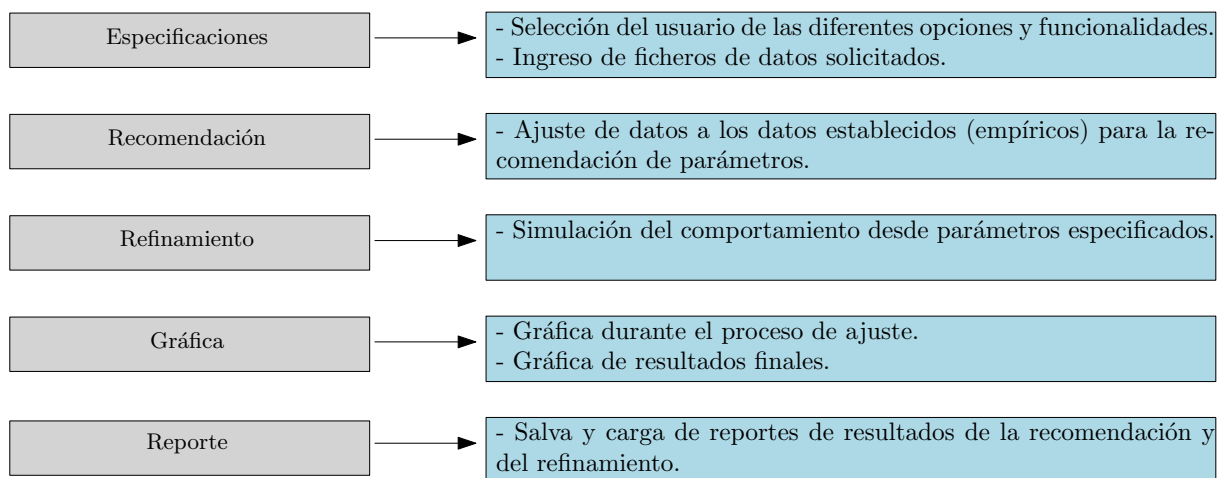


Figura 2.4: Enfoque funcional de implementación.

## 2.3. Validación

El proyecto será validado por diferentes clientes como proceso y como producto. La validación del proyecto como proceso la realizarán, de manera sistemática durante la realización del proyecto, docentes de la carrera de Ingeniería en Computación e Informática de la Facultad de Ingeniería de la Universidad Nacional Andrés Bello (UNAB), sede Viña del Mar.

La validación del proyecto como producto se hará midiendo la satisfacción del cliente final a través de una encuesta en la tarea de encontrar el conjunto de parámetros que les permita el mejor ajuste a los datos, así como a su refinamiento manual, e incluyendo su valoración de otros factores como el cumplimiento, seriedad, puntualidad en el proceso, etc., que tiene el cliente sobre el equipo desarrollador.

## 2.4. Riesgos

Los riesgos iniciales que se perciben en el proyecto son los siguientes:

- **Experiencia en proyectos:** referida a la experiencia en proyectos anteriores por parte del equipo. El equipo no ha realizado anteriormente un proyecto de ésta índole, lo cual es un importante reto pero a la vez un riesgo.
- **Horario:** además de llevar a cabo el proyecto, los participantes del mismo tiene otras tareas (trabajo y estudio, por ejemplo) que pueden limitar el horario dedicado al mismo, e incluso puede producir choques de horarios en las reuniones previstas.
- **Conocimientos en Fisiología Celular:** por las características del proyecto, a la hora del diseño e implementación, resulta importante la comprensión por parte del equipo de las bases fisiológicas con las que se trabajan, el significado de los datos que se manejan y de los modelos investigativos propuestos en el estudio de los mecanismos de compuerta de canales de iones. Esto requiere un estudio en temas de Fisiología Celular relacionados directamente con los objetivos del proyecto.

- **Ajuste de parámetros:** uno de los objetivos del proyecto, el ajuste de los datos a partir del modelo, se plantea como un problema complejo de optimización multi-objetivo de varias variables, con valores de las mismas en los Reales. Para lograr este objetivo, un método exacto no es viable y, será necesario la utilización de una heurística para resolverlo. En este sentido, se requerirá una investigación de los algoritmos que mejor resultado han producido en este tipo de problemas de optimización, y plantear una estrategia para la resolución del mismo.

## Capítulo 3

### Materiales y Métodos

## 3.1. Organización

### 3.1.1. Equipo de proyecto

El equipo de proyecto está formado por la empresa desarrolladora y el cliente, éste último es el subdirector del CINV. Los organigramas de la empresa desarrolladora y del CINV se encuentran a la izquierda y a la derecha de la figura 3.1, respectivamente.

En la misma figura se observa que los usuarios finales son los investigadores del departamento de Estructuras y Funciones de Sensores Moleculares. El cliente, además de su cargo de subdirector, es miembro de este departamento, y por tanto, junto con los miembros del mismo, es futuro usuario del sistema.

#### **Roles:**

- J. de Proyecto, Analista, Diseñador, Programador y *Tester*: Estudiante de pregrado de la Facultad de Ingeniería, Ingeniería en Computación e Informática, Universidad Nacional Andrés Bello.
- Cliente: subdirector e investigador del Centro de Neurociencia de Valparaíso (CINV), profesor de la Universidad de Valparaíso.

#### **Usuarios:**

- Investigadores del Departamento de Estructura y Función de Sensores Moleculares del CINV, y comunidad científica en general.

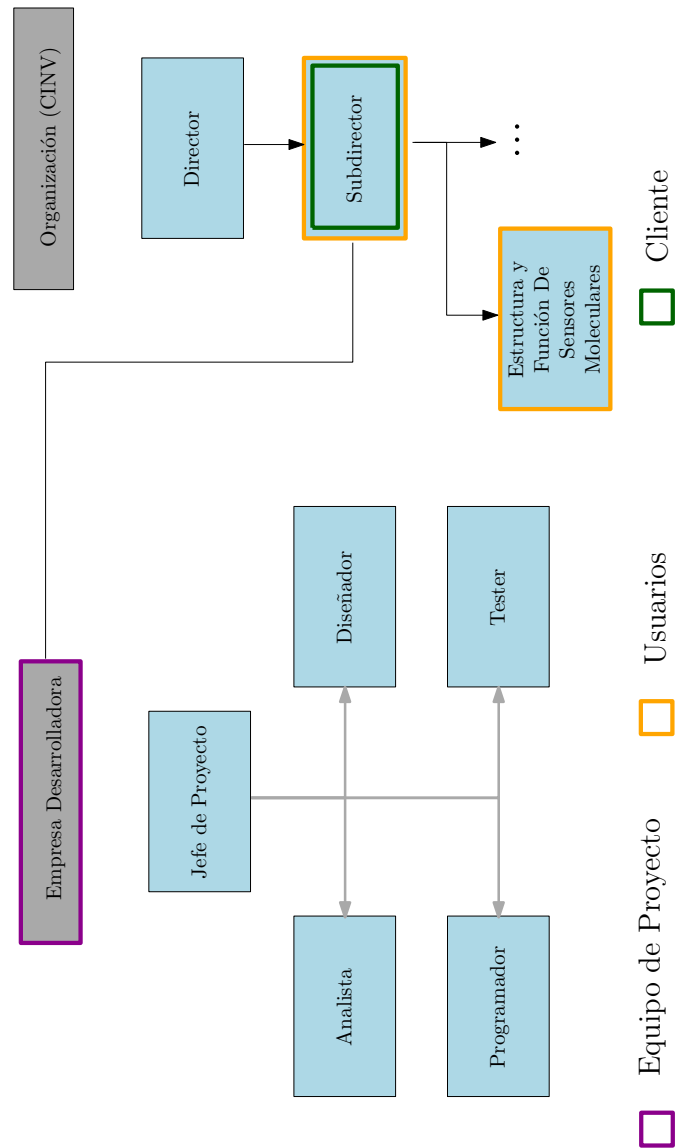


Figura 3.1: Cliente, usuarios y equipo de proyecto.



### 3.1.2. Roles y responsabilidades

Rol	Responsabilidades
Jefe de Proyecto	<ul style="list-style-type: none"> <li>- Controlar de manera continua los tiempos, el cumplimiento de objetivos y los costos.</li> <li>- Facilitar y coordinar la labor del equipo de proyecto.</li> <li>- Interactuar con el cliente.</li> <li>- Planificar y administrar el proyecto de <i>software</i>.</li> </ul>
Analista	<ul style="list-style-type: none"> <li>- Estudiar el problema de optimización y proponer una estrategia de resolución para el mismo (lo que posteriormente será definido como el primer subproyecto).</li> <li>- Realizar la recolección de datos.</li> <li>- Recopilar y analizar los requisitos del sistema.</li> <li>- Modelar y describir los casos de uso.</li> <li>- Analizar trazabilidad entre casos de uso y requisitos.</li> <li>- Describir las funcionalidades del sistema y su comportamiento.</li> </ul>
Diseñador	<ul style="list-style-type: none"> <li>- Diseñar la arquitectura y el sistema.</li> <li>- Asegurarse de que el producto final se ajuste al diseño.</li> <li>- Diseñar <i>software</i> que responda a los requerimientos especificados.</li> </ul>
Tester	<ul style="list-style-type: none"> <li>- Construir y aplicar los planes de prueba de módulo y de sistema.</li> </ul>
Cliente	<ul style="list-style-type: none"> <li>- Proveer información sobre los requisitos del sistema a desarrollar.</li> <li>- Realizar las pruebas de aceptación del sistema.</li> <li>- Validar el sistema una vez terminadas las pruebas.</li> </ul>

Cuadro 3.1: Roles y responsabilidades.

## 3.2. Metodología de desarrollo

El proyecto fue dividido en dos subproyectos, con metodologías y fines diferentes, donde el segundo depende de los resultados del primero. Un primer subproyecto donde se llevaron a cabo las primeras fases del proceso de investigación cuantitativa relacionado con el problema de optimización que surge desde la recomendación de valores de parámetros, con el fin de obtener la propuesta de una estrategia adecuada para el problema. Y un segundo subproyecto, informático, que utiliza la estrategia propuesta en el subproyecto anterior y que tiene como fin la construcción del *software* recomendador y afinador. La consecución de estos subproyectos se observa gráficamente en la figura 3.2.

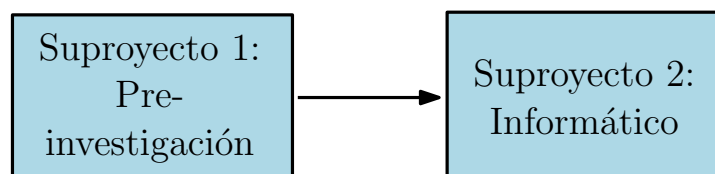


Figura 3.2: Dos subproyectos (el segundo depende del primero).

### 3.2.1. Subproyecto 1: pre-investigación

#### Fases definidas en la pre-investigación

Este subproyecto tiene como fin la propuesta de una estrategia para el problema de optimización y se hace cargo del cumplimiento de dos de los objetivos específicos enunciados en el capítulo Introducción, éstos son:

- Definir y modelar el problema de optimización multi-objetivo que plantea la recomendación de valores de los parámetros.
- Proponer una estrategia de resolución para el mismo.

La investigación no fue concluida antes de la continuación del proyecto informático por cuestiones de tiempo, debido a que el proyecto en su totalidad es un proyecto para optar a un

título profesional, el cual tiene una fecha establecida, y los objetivos de esta investigación se escapan del marco solicitado por la institución. Debido a que no incluye todas las fases de un proyecto de investigación, lo denominaremos pre-investigación.

Las primeras fases que se llevaron a cabo, definidas por Hernández y cols en el 2010, fueron desarrolladas con una metodología secuencial y son:

- La idea,
- el planteamiento del problema,
- la revisión de la literatura y desarrollo del marco teórico,
- la visualización del alcance del estudio,
- y la elaboración de hipótesis y definición de variables.

La consecución de las mismas de manera secuencial se muestra gráficamente en la Figura 3.3. A estas fases, se decidió agregar una última fase de pruebas iniciales, para contar con resultados a utilizar en el siguiente subproyecto, el subproyecto informático.

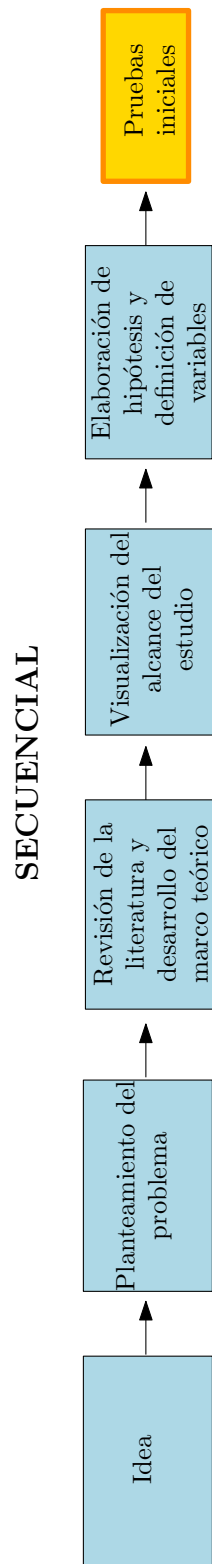


Figura 3.3: Modelo secuencial para el primer subproyecto.

### 3.2.2. Suproyecto 2: proyecto informático

El segundo suproyecto, el informático, tiene como fin la obtención del sistema, y se hace cargo de los restantes objetivos específicos de los que no se hace cargo la fase anterior, a saber:

- Recomendar valores para los parámetros del modelo que correspondan a la distancia mínima encontrada entre los datos empíricos y los datos simulados desde dicho modelo.
- Afinar los valores de los parámetros permitiendo visualizar la implicancia de cada uno de ellos en el modelo.
- Salvar y cargar instancias (reportes) del resultado de la recomendación de parámetros, y del resultado del refinamiento.

Este subproyecto se llevó a cabo con una metodología Cascada, la cual en la literatura es también nombrada como Modelo Lineal Secuencial o Ciclo de Vida Básico. Se seleccionó esta metodología porque los requisitos estaban claros desde un inicio, al igual que las herramientas a utilizar; permitiendo que de manera rigurosa se pudiera pasar de una fase a otra sin retrocesos (al menos para esta primera versión del producto). El estar las fases establecidas de manera secuencial, delimita las mismas dando al proceso mayor organización, producto de no permitir el inicio de una sin estar concluida la anterior.

Siguiendo la definición de Pressman en el 2001 (Pressman, 2001, p.24), las cuatro fases que se llevaron a cabo con esta metodología se han graficado en la figura 3.4 y son:

- Análisis: donde se recopilaron y analizaron los requisitos y se modelaron y describieron los casos de uso, enfocándose en el *software* a construir.
- Diseño: donde se diseñaron y modelaron las estructuras de datos, la arquitectura del *software*, la interfaz de usuario, las clases a codificar y el procedimiento.
- Codificación: donde se tradujo el resultado de la fase anterior a código compilable y se realizaron las pruebas unitarias.
- Pruebas: por último, se realizaron las pruebas para detectar errores externos referentes a los resultados reales en comparación con los resultados deseados.

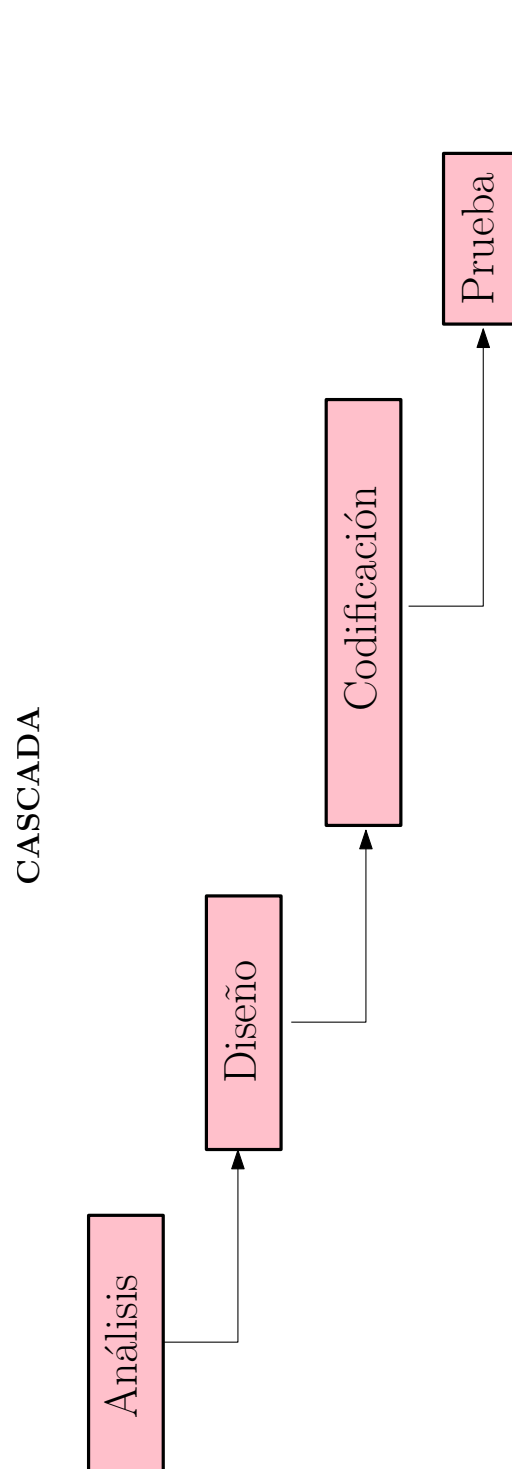


Figura 3.4: Modelo Cascada para el segundo subproyecto. Los tamaños están en proporción con el porcentaje de HH que se destinaría a cada fase según la estimación inicial.

## Análisis

El Análisis se realizó en 4 fases: reconocimiento del problema, evaluación y síntesis, modelado, revisión y corrección; las cuales fueron propuestas por Pressman (Pressman, 2001, p.182). Los requisitos de datos, funciones y comportamiento se identificaron mediante la obtención de información facilitada por el cliente. En esta fase se refinaron y analizaron estos requisitos para verificar su claridad, completitud y consistencia. La especificación resultante fue incorporada al modelo del *software* y validada tanto por el ingeniero del *software* como por el cliente.

En el reconocimiento del problema se estudió el plan de proyecto del *software*, con el objetivo de entenderlo en el contexto de un sistema y además se revisó el ámbito del *software* que se empleó para generar las estimaciones de la planificación.

En la segunda fase, la evaluación del problema y la síntesis de la solución, se definieron todos los objetos de datos observables externamente, se evaluó el flujo y contenido de la información, y se definieron y elaboraron todas las funciones del *software*. Al conjunto de estas tareas las denominamos: determinar y analizar los requisitos. Por último, en esta fase se establecieron las características de la interfaz del sistema. Todo lo anterior fue establecido siguiendo las preguntas que según Pressman esta fase debe responder: “¿*Qué datos produce y consume el sistema, qué funciones debe realizar el sistema, qué interfaces se definen y qué restricciones son aplicables?*” (Pressman, 2001, p.182).

En el modelado, se realizó el modelado de los casos de uso con sus respectivas descripciones; así se representaron los requisitos para ser comprobado con la percepción del cliente sobre el sistema. La correcta relación entre los casos de uso y los requisitos sistematizados, se comprobó mediante la matriz de trazabilidad de los casos de uso respecto a los requisitos funcionales del sistema.

En esta etapa se generó un documento de especificación y la revisión consistió en la aceptación por parte del cliente y del analista del *software*, del mismo. Por último, producto de la revisión, se llevaron a cabo las respectivas correcciones finalizando la fase de Análisis de requisitos.

La descomposición del trabajo en la fase de Análisis quedó definida entonces con las siguientes tareas:

### Reconocimiento del problema:

- Estudiar el plan de proyecto del *software*.

- Establecer la comunicación por medio de reuniones con el cliente.

Evaluación y síntesis:

- Determinar y organizar los requisitos.
- Establecer las características de interfaz del sistema.

Modelado:

- Modelar los casos de uso.
- Describir los casos de uso.

Revisión:

- Verificar trazabilidad.
- Enviar resultados al cliente.
- Revisar resultados y analizar correcciones con el cliente.

Corrección:

- Corregir los errores/cambios.

## Diseño

El diseño se realizó siguiendo el modelo de 4+1 vistas propuesto por Philippe Kruchten en 1995, el cual permite llegar a una descripción de la arquitectura del *software* desde cinco vistas, las cuales se refieren a los intereses de los diferentes *stakeholders* del sistema. El autor define estas vistas como:

- *“La vista lógica describe el modelo de objetos del diseño cuando se usa un métodos de diseño orientado a objetos.*
- *... La vista de procesos describe los aspectos de concurrencia y sincronización del diseño.*



- *La vista física describe el mapeo del software en el hardware y refleja los aspectos de distribución.*
- *La vista de desarrollo describe la organización estática del software en su ambiente de desarrollo.” (Kruncheten, 1995, p.1)*

La quinta vista es la vista de escenarios o casos de uso, que es una ilustración de las vistas anteriores.

El *software* que se realizó en este proyecto, tiene la característica de ser orientado a objetos (OO), por lo que el diseño se realizó orientado a objetos mediante la realización de diagramas que siguieran el estándar UML 2.2. En estos casos el modelado contiene las clases (objetos) que representan el problema a resolver, la relación entre esas clases, el funcionamiento interno de las mismas (atributos y operaciones) y los mecanismos de comunicación (mensajes) en la interacción.

Así, desde el modelo de Kruncheten y el estándar de modelado UML 2.2, se modelaron las diferentes vistas propuestas. La vista de escenarios se modeló mediante los diagramas de casos de uso, la vista de procesos mediante el diagrama de actividades, la vista de arquitectura lógica mediante el diagrama de clases, la vista de desarrollo mediante el diagrama de componentes del sistema, y por último, la vista física mediante el diagrama de despliegue.

La descomposición del trabajo en la fase de Diseño quedó establecida con las siguientes tareas:

- Estudiar la modelación desde el modelo de 4+1 vistas de Philippe Kruncheten.
- Estudiar el modelado mediante diagramas en UML.
- Analizar y modelar las actividades del sistema.
- Analizar y modelar las clases y la interacción entre ellas.
- Analizar y modelar los componentes del sistema.
- Analizar y modelar la arquitectura física del sistema.
- Revisar los resultados de las tareas anteriores.

## Codificación

### Ambiente de desarrollo

El ambiente de desarrollo para el proyecto fue MATLAB (Abreviatura de MATrix LABoratory, Laboratorio de matrices). MATLAB es una herramienta de *software* matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M).

### Características:

- Herramienta de *software* matemático.
- Disponible para las plataformas Unix, Windows y Mac OS X.a.
- Prestaciones principales: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes.
- Ampliamente utilizado en universidades y centros de investigación y desarrollo.

En esta fase de codificación el programador realizó, además de la codificación de los diferentes casos de uso, las pruebas unitarias y las pruebas de integración de los diferentes módulos.

## Pruebas

En la fase de pruebas se identificaron los casos de pruebas a partir de los casos de uso, y se realizaron las pruebas definidas en estos casos de pruebas, y se verificó el cumplimiento, o no, de las pruebas estándar; lo anterior mediante la técnica de caja negra para las pruebas del sistema. Además, se llevó un registro de defectos, los cuales fueron siendo corregidos por el codificador y se dió término a la etapa cuando todas las pruebas y defectos encontrados se reportaron y corrigieron, respectivamente.

Las planillas para las especificaciones de las pruebas, el análisis de resultados, las pruebas estándares y el registro de los defectos, son extraídas del proyecto chileno Tutekan como guía para llevar a cabo una buena planificación de la etapa, un buen seguimiento y un buen registro (Tutekan, 2012).

La descomposición del trabajo en esta fase de Pruebas se organizó de la siguiente manera:

- Identificar los requerimientos para el *testing*.
- Desarrollar la estrategia de pruebas.
- Identificar los recursos para las pruebas.
- Generar Plan de Pruebas detallado.
- Identificar y describir los casos de prueba (pruebas funcionales).
- Revisar y acceder la cobertura de las pruebas.
- Implementar las pruebas.
- Identificar las funcionalidades a probar.
- Establecer el conjunto de datos externos.
- Verificar las pruebas estándar.
- Ejecutar las pruebas.
- Evaluar la ejecución de las pruebas.
- Comprobar los resultados.
- Investigar los resultados inesperados.
- Registrar los defectos.
- Analizar defectos.
- Corregir defectos.

### 3.3. Cierre del proyecto

Los objetivos principales identificados en el cierre del proyecto son:

- Analizar, desde la perspectiva económica, el balance de los recursos.
- Analizar, desde la perspectiva temporal, el balance de lo que se logró hacer en el tiempo estimado.
- Diagnosticar el funcionamiento, tratando de analizar las desviaciones entre las previsiones iniciales y el resultado.
- Analizar correcciones para versiones futuras y las actuaciones que dieron pie a tales desviaciones.

Las tareas a realizar en esta etapa serán:

- Entregar el producto al cliente.
- Realizar una reunión con el cliente para la revisión conjunta del producto.
- Resumir resultados de las pruebas de aceptación realizadas por el cliente.
- Aplicarle al cliente una encuesta de evaluación (validación) del proceso y del producto llevado a cabo por el equipo de proyecto.
- Firmar junto con el cliente el Acta de Aceptación.
- Completar el Acta de Informe de la Situación.
- Realizar el balance económico y temporal.

El Acta de Aceptación se muestra en la figura 3.5, y el Acta de Informe de la Situación al cierre del proyecto se observa en la figura 3.6.

## ACTA DE ACEPTACIÓN

Referencias: \_\_\_\_\_

Proyecto: \_\_\_\_\_

Cliente: \_\_\_\_\_

Entrega de: \_\_\_\_\_

Fecha: \_\_\_\_\_

☐ Total
 ☐ Parcial

Elementos entregados: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Pruebas realizadas: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

El cliente certifica que la totalidad de los suministros o servicios reseñados en la presente acta de aceptación han sido entregados/terminados, y que, habiendo sido sometidos a las pruebas de validación y aceptación indicadas, están de acuerdo con las especificaciones formales y demás requisitos contractualmente convenidos y establecidos entre las partes con las siguientes OBSERVACIONES:

Por el cliente:	Por la empresa desarrolladora:
Fdo.: Fecha:	Fdo.: Fecha:

Figura 3.5: Acta de aceptación.

## INFORME DE SITUACIÓN

### Model Fitness

☐ Intermedio  
☐ Final

Proyecto: \_\_\_\_\_ Cliente: \_\_\_\_\_  
 Título: \_\_\_\_\_  
 Responsable: \_\_\_\_\_  
 Fecha de comienzo: \_\_\_\_\_ Terminado: \_\_\_\_\_

**TRABAJO REALIZADO, ALTERACIONES AL ALCANCE PREVISTO**

**DIFICULTADES ENCONTRADAS**

**RELACIONES CON TERCEROS**

**ACCIONES FUTURAS**

**OTROS COMENTARIOS**

**RESUMEN DEL ESTADO**

		Sí		No	Descripción
Modificaciones al alcance	<input type="checkbox"/>		<input type="checkbox"/>		_____
Retrasos	<input type="checkbox"/>		<input type="checkbox"/>		_____
Incremento del riesgo	<input type="checkbox"/>		<input type="checkbox"/>		_____
Sobrecoste	<input type="checkbox"/>		<input type="checkbox"/>		_____
Insatisfacción del cliente	<input type="checkbox"/>		<input type="checkbox"/>		_____
Carencia de recursos	<input type="checkbox"/>		<input type="checkbox"/>		_____
Conflictos interpersonales	<input type="checkbox"/>		<input type="checkbox"/>		_____
Falta de formación y/o experiencia	<input type="checkbox"/>		<input type="checkbox"/>		_____

Figura 3.6: Acta de informe de situación.

### 3.4. Estimaciones y planificación

Para el cálculo de las Horas Hombre (HH) referentes al primer subproyecto, se utilizó la técnica de Jucio Experto, donde fueron estimadas las HH para cada fase del proyecto de pre-investigación. Los resultados se encuentran en la Figura 3.7. Basado en estos resultados, se realizó una planificación en el tiempo, medido en HH, de las diferentes fases (Figura 3.8).

Fase	%	HH
Gestión del proyecto	5	8
Planteamiento del problema	10	16
Revisión de la literatura y desarrollo del MT	50	80
Visualización del alcance del estudio	15	24
Elaboración de H y definición de variables	10	16
Pruebas iniciales	10	16
<b>Total de esfuerzo</b>	100	160

Figura 3.7: HH para el primer subproyecto.

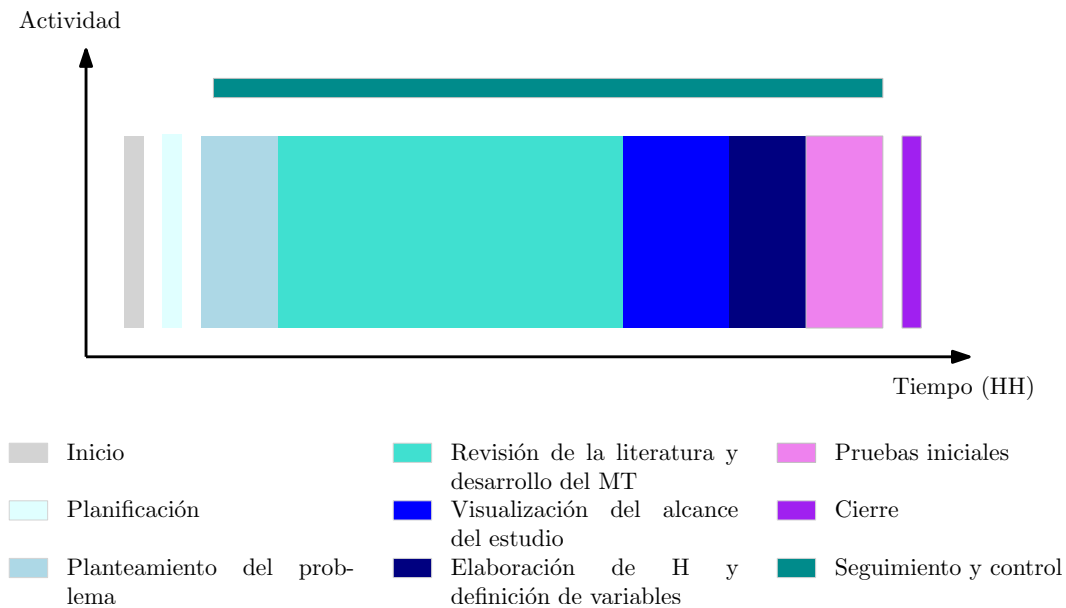


Figura 3.8: Planificación basada en las HH del primer subproyecto.

En la figura 3.9 se presenta el diagrama de casos de uso del producto final, mediante los requerimientos funcionales del mismo. Este análisis se realizó mediante la utilización de los puntos de función para la estimación del tamaño del proyecto. Con estos datos se calculó la cantidad de HH en la codificación del *software* la cual queda especificada en la figura 3.10.

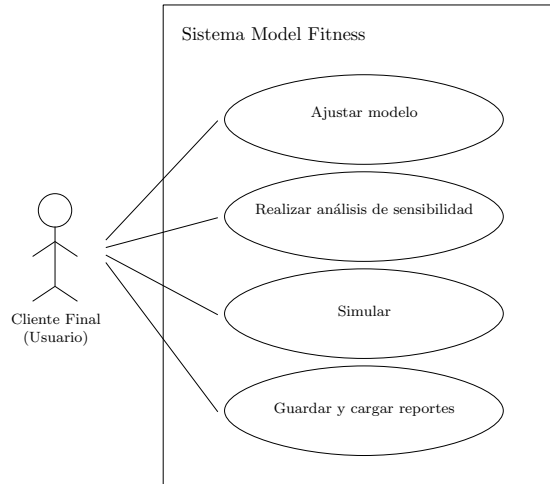


Figura 3.9: Diagrama casos de uso del producto.

Ajuste: contar los factores de entorno entre R1 y R6 cuya influencia es inferior a 3 (influencia promedio) y los factores de entorno entre R7 y R8 que son superiores a 3.

Entonces:

- 20 horas hombre por UCP si el valores es  $\leq 2$ .
- 28 horas hombre por UCP si el valor es  $\leq 4$ .
- 36 horas por UCP si el valor es  $\geq 5$ , en este caso se debería replantear el proyecto.

R1 a R6 $\leq 3$	2
R1 a R6 $\geq 3$	1
Factor entorno	3

HH por UCP	28
------------	----

<b>Total de horas de desarrollo de proyecto</b>	<b>297</b>
---	------------

Figura 3.10: HH en la implementación del *software*.



Con las HH de la codificación, se calculó el porcentaje del resto de las fases implicadas en este segundo subproyecto, las cuales están especificadas en la figura 3.11. Basado en estos resultados de HH se realizó una planificación en el tiempo, medido en HH, de las diferentes fases, como queda reflejado en la figura 3.12.

Actividades	%	HH
Gestión del proyecto	10	99
Análisis	15	148,5
Otras actividades del analista	3	29,7
Diseño	18	178,2
Otras actividades del diseñador	3	29,7
Codificación	30	297
Otras actividades del programador	3	29,7
Pruebas	15	148,5
Otras actividades del tester	3	29,7
<b>Total de esfuerzo</b>	100	990

Figura 3.11: HH para el segundo subproyecto.

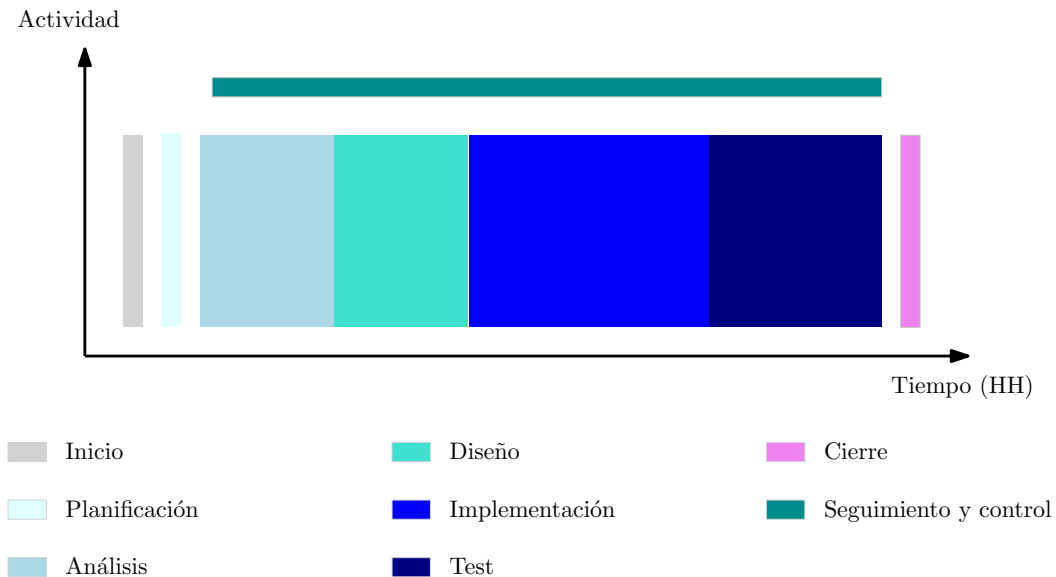


Figura 3.12: Planificación basada en las HH del segundo subproyecto.

Con los datos de salarios (obtenidos de [www.mifuturo.cl](http://www.mifuturo.cl)) se obtuvo el costo de personal del proyecto, el cual fue sumado al costo de *software* y *hardware* requerido, llegando a costo total del proyecto de 4.597.545 pesos chilenos como se observa en la figura 3.13 y en la figura 3.14.

Rol	C/H	HH	CT
Jefe de proyecto (Ingeniero)	5.584	107	597.488
Analista (Ingeniero)	5.584	330,2	1.843.837
Diseñador (Técnico)	3.025	207,9	628.897,5
Programador (Técnico)	3.025	326,7	988.267,5
Tester (Técnico)	3.025	178,2	539.055
<b>Costo total de la mano de obra</b>	<b>20.243</b>	<b>1.150</b>	<b>4.597.545</b>

Figura 3.13: Costo de la mano de obra para el proyecto.

Recurso	Costo
Mano de obra	4.597.545
Software (licencias)	0
Hardware (equipos)	0
Otras (libros, impresiones, ect.)	30.000
<b>Costo total del proyecto</b>	<b>4.897.545</b>

Figura 3.14: Costo total del proyecto.

Por último, teniendo en cuenta las HH estimadas en cada etapa y asumiendo que se contará con 5 horas diarias dedicadas al proyecto, la estimación temporal queda reflejada en la Carta Gantt que se muestra en la figura 3.15, donde se estima el cierre del proyecto para el 06 de enero de 2014.

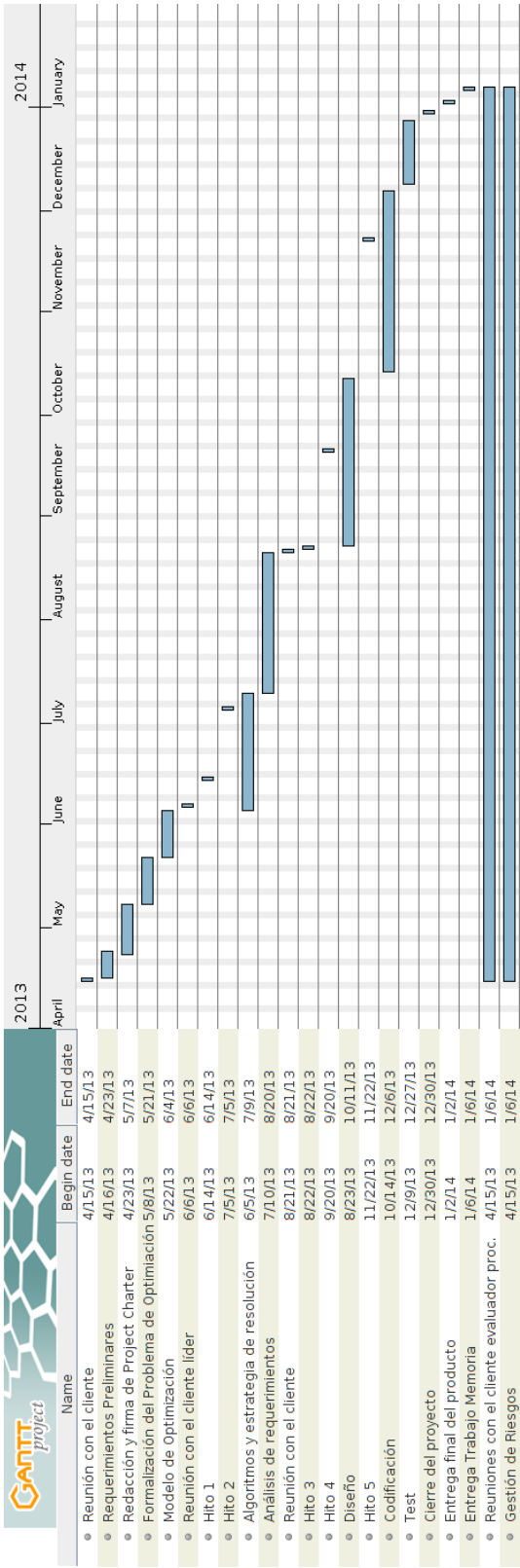


Figura 3.15: Carta Gantt general del proyecto.

## 3.5. Planes de proceso en el desarrollo del proyecto

### 3.5.1. Plan de gestión de riesgos

La primera fuente de riesgos proviene de la escasa experiencia en proyectos anteriores. Esto puede llevar a no saber manejar el proyecto como proceso, sobre todo desde la gestión de la configuración y el seguimiento y control, al no cumplimiento de los plazos establecidos, y a la no finalización del proyecto.

Una segunda fuente es producto de los horarios. Puede llevar al no cumplimiento de las reuniones previstas debido a choques de horario entre cliente y proveedor, a la no realización de las tareas asignadas por falta de tiempo, y a una mala planificación en las horas estimadas semanalmente para la realización del mismo.

Un tercer riesgo es la no utilización de una estrategia adecuada en el problema de optimización en el ajuste de parámetros. Esto puede llevar a malos resultados en dicho ajuste y a un mayor tiempo en la consecución de buenos resultados.

#### **Escasa experiencia**

##### Indicadores

El indicador utilizado para este riesgo será, en cada una de las fases del proyecto, el estado del proyecto. Es decir, si el proyecto se mantiene consistentemente en atraso en su proceso, será indicador de la baja experiencia en esa fase del proyecto, o de una mala planificación y estimación del proyecto; que igualmente refiere a la experiencia en proyectos anteriores.

##### Plan de mitigación

Teniendo en cuenta este riesgo, se ha planificado el proyecto aumentando las horas destinadas a cada fase respecto a las que se recomienda en la literatura.

##### Plan de contingencia

En caso de un atraso significativo en la planificación respecto a lo real, teniendo en cuenta sobre todo las primeras fases de Análisis y Diseño definidas en el ciclo de vida del proyecto, el curso a seguir será la dismunición de las funcionalidades; es decir, no serían implementados todos los casos de uso identificados, acotando las funcionalidades del sistema.

## Horario

Indicadores El indicador será el porcentaje de reuniones planificadas que realmente se llevaron a cabo, el cual debe ser superior al 80 %.

### Plan de mitigación

Desde el inicio del proyecto se deberán planificar las reuniones fuera de horarios conflictivos; es decir, luego de la culminación del horario de trabajo-estudio, y en fechas en las que no se tienen planificado viajes, eventos, congresos, etc.

### Plan de contingencia

Las alternativas, en caso de contingencias, son:

- Utilización de otro medio de comunicación visual como skype.
- Utilización de un medio de comunicación escrito (pj: correos).
- O la utilización la reducción de reuniones mediante la aglomeración de varias de ellas en unas menos. Es decir, aprovechar los tiempos de una reunión para aclarar puntos que se hayan destinado a otras reuniones.

## Estrategia inadecuada

### Indicadores

No aplica.

### Plan de mitigación

Intenso estudio bibliográfico de las estrategias de optimización, especialmente en problemas con características similares.

### Plan de contingencia

Las alternativas, en caso de contingencias, son:

- Utilización de una estrategia menos adecuada siempre que siga los objetivos destinados de minimización.

### 3.5.2. Plan de control de la configuración

#### Control de versiones: Bazaar

La herramienta utilizada para el control de versiones en el proyecto es Bazaar-ng, la cual es un sistema de control de versiones distribuido diseñado para facilitar la contribución en proyectos de *software* libre y opensource. Como el resto de las herramientas del control de versiones para proyectos, y siendo su fundamental funcionalidad, permite guardar progresivamente los cambios sobre un conjunto de archivos de texto (habitualmente código fuente). Además, permite recuperar versiones anteriores, mostrar diferencias, integrar el trabajo de diversos programadores, etc.

Por otro lado, si bien está diseñada para trabajar perfectamente de modo distribuido, también se adapta a un flujo de trabajo centralizado, en cuyo esquema adopta un modo de trabajo muy similar a Suversion pero manteniendo ciertas ventajas propias de los sistemas distribuidos, como:

- Contar con una copia local del repositorio.
- Acceso local al historial de cambios (y poder consultar rápidamente el registro de cambios).
- Poder realizar commits locales al no tener conexión a internet.
- Volver rápidamente a versiones anteriores.

Puede ser usado por un usuario único trabajando en múltiples ramas de un contenido local (formato que se adecúa al presente proyecto), o por un equipo colaborando a través de la red. Está escrito en lenguaje de programación Python y tiene versiones empaquetadas para la mayoría de distribuciones GNU/Linux, así como Mac OS X y MS Windows.

#### Selección de los Ítems de Configuración

La selección de los ítems de configuración será responsabilidad del Jefe de Proyecto.

### Determinación de líneas bases

Al dar inicio al proyecto luego de la firma del Project Charter, se fijará la primera línea base. El objetivo es fijar la línea base de alcance para el proyecto, su estimación y planificación.

Además, se generará una línea base por cada fase, una al concluir el primer subproyecto, y una por cada fase del subproyecto informático, y se modificará ante cada cambio crítico en la planificación del proyecto acordado con el cliente.

También, por cada entrega al cliente, se generará una línea base; y una vez que el Producto esté terminado, probado y validado con el cliente, se tendrá la línea base final, que servirá de entrada para futuras versiones del *software*.

### 3.5.3. Plan de gestión de requerimientos

Los requerimientos serán todos acordados entre el cliente y el analista. En la fase de Análisis se realizarán 3 reuniones con el cliente a fin de sistematizar los requerimientos que solicite, y el analista, con previo acuerdo con la empresa desarrolladora, propondrá requerimientos al cliente como valor agregado del proyecto, o requerimientos que tal vez el cliente asuma implícitos pero que deben registrarse para su posterior comprobación.

Una vez concluida la fase de Análisis de requisitos, no se permitirán bajo ninguna circunstancias modificaciones en los mismos, sólo eliminaciones de requisitos solicitados, y requisitos visuales de la interfaz (como tamaño de ventanas, color, nombres o textos dentro de las mismas, etc.) que luego de la codificación el cliente desee cambiar.

Dada la naturaleza del proyecto, se optará por usar la matrices de trazabilidad del sistema de los requisitos con los casos de uso. Esta matriz debe ser generada al final de la fase de Análisis con el fin de comprobar que no queden requisitos sin tener en cuenta, y que no hayan casos de uso que no estén asociados a ningún requisito.

### 3.5.4. Plan de pruebas

#### Pruebas unitarias y de integración

Estas pruebas serán llevadas a cabo durante la fase de Codificación y estarán a cargo del programador.

#### Pruebas funcionales

Estas pruebas serán diseñadas, especificadas y llevadas a cabo por el *tester* en la fase de Pruebas del proyecto. El *testing* funcional se realizará sobre los requerimientos funcionales antes descritos y sus casos de uso. Estas pruebas tienen por finalidad comprobar la funcionalidad de la aplicación a partir de datos válidamente seleccionados sobre las transacciones del sistema. Este tipo de comprobación se basa en las técnicas de caja negra, que permiten probar la aplicación (y sus procesos internos) vía GUI.

#### Pruebas estándares

Consiste en la verificación por parte del *tester* en la fase de Pruebas, del cumplimiento del producto de las pruebas estándares propuestas en el proyecto Tutelkan, diferenciando las que aplican, las que no aplican y las que no se han tenido en cuenta para el producto.

#### Pruebas de aceptación del cliente

Son pruebas que diseñará y llevará a cabo el cliente a fin de aceptar el producto final. Se realizarán durante el cierre del proyecto.

#### Preparación del ambiente de pruebas

Las pruebas deberán realizarse en un computador con Sistema Operativo Windows, no necesita de acceso a internet ni conexión en red con otros computadores. Se requiere solamente el espacio necesario para la ubicación de un computador.



Objetivo de la prueba:	Asegurar la funcionalidad del conjunto de casos, incluyendo la navegación en la aplicación, el ingreso de datos, el proceso y la recuperación (resultados). Que los objetos de las ventanas y sus características, tales como menús, tamaño, posición, estados y foco, estén de acuerdo a los estándares.
Técnica a utilizar:	Ejecutar cada caso de uso, su flujo y funcionalidad usando tanto datos válidos como inválidos para comprobar lo siguiente: - Que los resultados esperados ocurren cuando los datos válidos son utilizados. - Que el mensaje de error es apropiado cuando se utilizan datos inválidos.
Criterio de validación:	- Todas las pruebas planificadas se ejecutaron correctamente. - Todos los defectos identificados han sido asignados. - Cada ventana debe ser verificada para mantener la consistencia con la versión maestra y comprobar que esté dentro de los estándares aceptables.

Cuadro 3.2: Pruebas funcionales.

Las pruebas de aceptación del usuario y del sistema se ejecutarán en este ambiente de pruebas. Este ambiente representará una configuración idéntica al ambiente de producción. Esto se requiere debido a que se debe replicar el rendimiento de la línea base y las medidas de mejoramiento relacionadas.

## Capítulo 4

### Resultados y discusión: subproyecto de investigación

## 4.1. Idea

Como se mencionó en el capítulo Introducción, en el estudio del funcionamiento de los mecanismos de los canales de iones, el departamento de Estructuras y Funciones Moleculares del Centro Interdisciplinario de Neurociencia de Valparaíso (CINV) ha recopilado un conjunto de datos empíricos sobre las corrientes presentes en los canales de iones de  $K^+$  dependientes del voltaje. Por otro lado, el mismo departamento ha propuesto un modelo teórico ( $C-C-O$ ) sobre cómo funcionan estos mecanismos de compuerta. Resulta importante determinar la distancia en la brecha que pueda existir entre esta teoría y los datos empíricos, con el fin de determinar si los datos son explicados por el modelo propuesto.

El modelo  $C-C-O$  describe las transiciones entre estados **abiertos** ( $O$ , hay corriente iónica) y **cerrados** ( $C$ , no hay corriente iónica) de los canales de iones con un carácter probabilístico; es decir, la activación del canal significa que aumenta la probabilidad de que éste se abra, y no que el canal se mantenga abierto durante todo el estímulo. De esta manera, considera a los canales estructuras dinámicas con cambios en las probabilidades de transiciones entre los estados  $O$  y  $C$ .

Propone además, que cada uno de los cuatro sensores de voltaje presentes en un canal de  $K^+$  (llamada subunidad  $\alpha$ ) pueden estar en forma activa o en reposo de manera independiente. El conjunto de los sensores y la forma en la que pueden estar determinan los diferentes estados, donde el cambio de uno a otro ocurre con cierta probabilidad.

Como se observa en la Figura 4.1 (a), en el modelo  $C-C-O$  cada fila representa las diferentes combinaciones de los cuatro sensores de voltaje, y cada columna describe un canal que se abre en dos pasos: activación del sensor de voltaje, paso de  $C_0$  a  $C_1$ , y apertura del canal, paso de  $C_1$  a  $O$ .

Como se observa en la primera fila de la Figura 4.1 (a), si se asume  $\alpha$  como la probabilidad de que se active uno de los sensores, el pasar de un estado donde todos los sensores están en reposo a un estado donde hay uno activo sería  $4\alpha$ . De igual manera, pasar de un estado de un solo sensor activo y tres en reposo a uno con dos activos y dos en reposo sería  $3\alpha$ , ya que puede ser activado cualquiera de los que se encuentra en reposo. Así, luego se transitaría a un estado de 3 activos con una probabilidad de  $2\alpha$  y a uno con todos activos con una probabilidad  $\alpha$ . Siguiendo la misma lógica, las transiciones inversas quedarían determinadas por otras probabilidades, denominadas  $\beta$ , desde  $4\beta$  hasta  $\beta$ . A modo de ejemplo, en la Figura 4.1 (b)

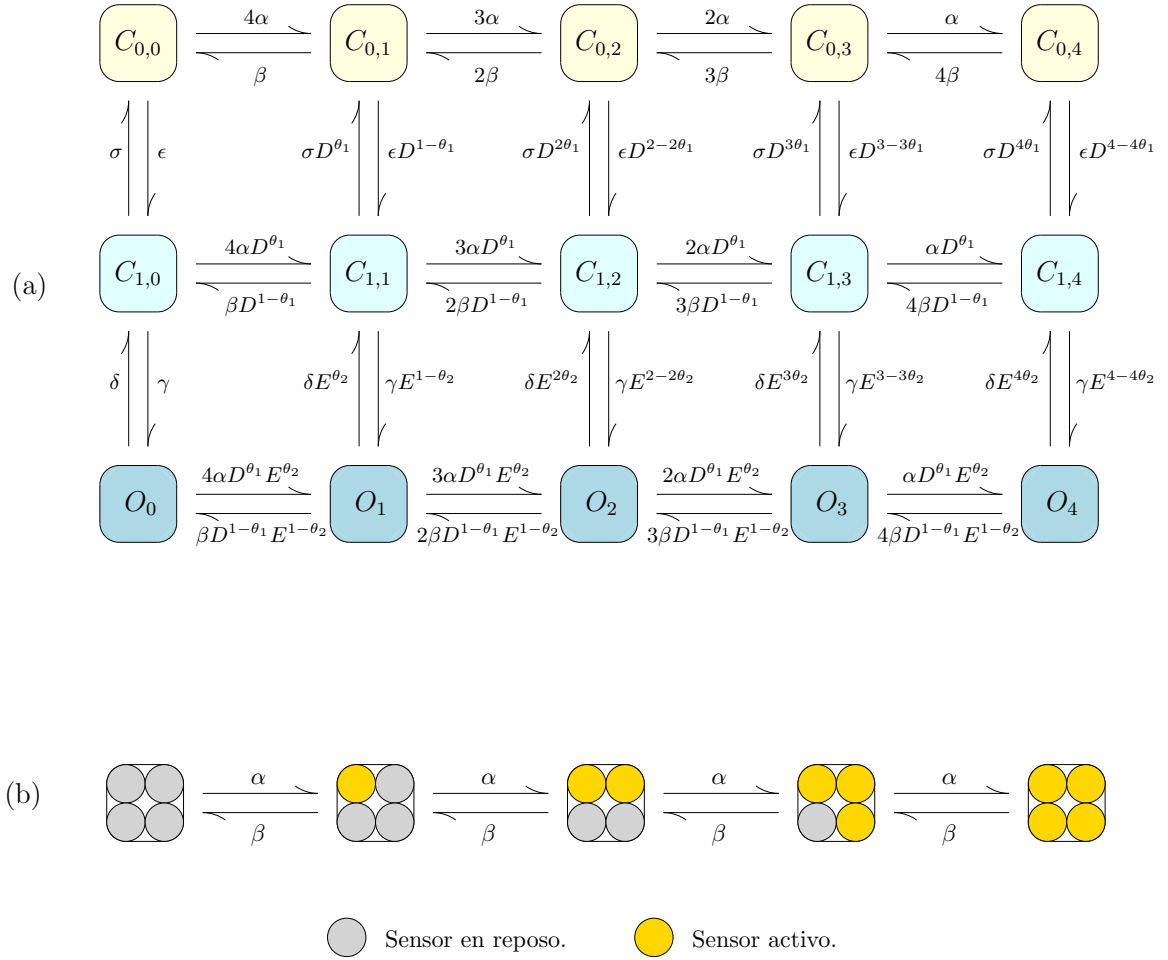


Figura 4.1: Modelo de tres niveles. (a) Los 15 estados con sus probabilidades de transiciones entre ellos. (b) Reposo y activación de los 4 sensores que forman el canal en un recorrido entre estados cerrados del primer nivel.

se muestra un posible recorrido entre estados cerrados de la primera fila.

Así, este modelo consta en total de 15 estados, donde sus probabilidades de transiciones están determinadas por 14 parámetros:  $\alpha$  ( $\alpha$ ),  $\beta$  ( $\beta$ ),  $\sigma$  ( $\sigma$ ),  $\varepsilon$  ( $\varepsilon$ ),  $\delta$  ( $\delta$ ),  $\gamma$  ( $\gamma$ ),  $z\_alpha$ ,  $d\_alpha$ ,  $z\_delta$ ,  $d\_delta$ ,  $AllD$ ,  $theta1$  ( $\theta1$ ),  $AllE$  y  $theta2$ .

Estos 14 parámetros, definen un conjunto de variables que determinan el comportamiento de los canales, a saber: la carga de compuerta, la probabilidad de apertura del canal, las constantes de tiempo de los componentes lentos (transiciones verticales en el modelo) y rápidos (transiciones horizontales en el modelo), la corriente del canal, la corriente de compuerta, y la amplitud de los componentes lentos. Desde estas variables y los diferentes voltajes aplicados en los experimentos, se definen 6 funciones relacionadas con los datos fisiológicos que se consideran importantes en la explicación de este mecanismo.

Por tanto, se tiene un modelo definido por 14 parámetros, donde se pueden calcular un conjunto de datos fisiológicos teóricos sobre el mecanismo de compuerta de estos canales, a los que llamaremos datos simulados; y se tiene, por otro lado, un conjunto de datos fisiológicos reales obtenidos empíricamente, que llamaremos datos empíricos.

Lo que se pretende entonces es conocer si existen valores de los parámetros que definan un conjunto de datos simulados que tengan una distancia numérica menor a  $1 \cdot 10^{-2}$  con los datos empíricos con los que se cuenta (proceso graficado en la Figura 4.2), lo que llevaría a determinar si los datos son explicados por el modelo. Es decir, encontrar la mínima distancia entre ambos conjuntos de datos, que resulta en un problema de optimización multi-objetivo y complejo, debido a que los datos representan varias funciones y sus valores están definidos en los Reales, respectivamente. Encontrar una estrategia adecuada para la resolución del mismo, teniendo en cuenta sus características, es el objetivo de esta investigación, que se llevará a cabo desde un enfoque cuantitativo.

## 4.2. Planteamiento del problema

Encontrar valores de los diferentes parámetros propuestos en el modelo  $C-C-O$ , que minimicen las distancias entre los datos simulados y los datos empíricos, entendido como un conjunto de funciones a minimizar, definen un problema multi-objetivo de mínimos cuadrados.

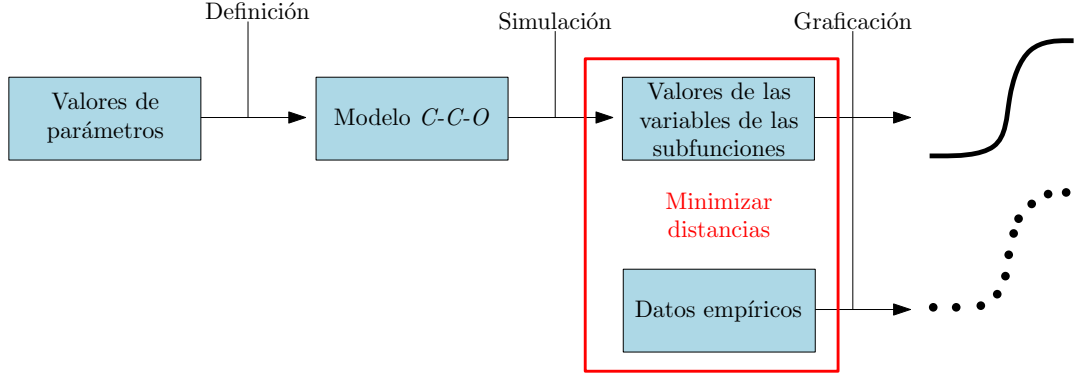


Figura 4.2: Flujo de datos para el problema de optimización.

Teniendo en cuenta que en su resolución un “buen”<sup>1</sup> error (distancia) obtenido, podría significar un buen resultado en sólo una o algunas de las funciones y no en todas, y con el objetivo de poder definir diferentes importancias a las funciones, se ha transformado el problema multi-objetivo en un problema mono-objetivo a través de la ponderación de dichas funciones:

Función objetivo a minimizar:

$$SSQ(p_k) = \sum_{i=1}^n A_i \left[ \sum_{j=1}^S \left( \frac{f_{ij} - x_{ij}}{n_j \cdot \text{máx } x_j} \right)^2 \right] \quad (4.1)$$

donde  $A_i = 1/S$  es una ponderación (pesos) asignada a cada función,

$S$  es la cantidad de conjuntos de datos (funciones, y por tanto, conjuntos que utilizan la misma escala),

$f_{ij}$  son los datos simulados a partir de los parámetros,

$x_{ij}$  son los datos experimentales, y

$n_j$  es la cantidad de datos experimentales en el conjunto  $j$ .

Los conjuntos de datos  $S$  son los datos de las 6 funciones que serán tomadas en cuenta, definidas a partir de las variables fisiológicas:

- $Q_{OFF}/Q_{MAX-Time(ms)}$ , donde  $Q$  es la carga de compuerta de activación.

<sup>1</sup>Que sea bueno o malo el error obtenido depende del criterio que se siga, es un valor subjetivo que depende de los objetivos del estudio. En este caso, como ya se ha mencionado, se refiere a valores menores a  $1 \cdot 10^{-2}$

- $Q/Q_{MAX}$ -Voltage(mV).
- $P_0$ -Voltage(mV), donde  $P_0$  es la probabilidad de apertura del canal.
- $t_{fast}$ -Voltage(mV), donde  $t_{fast}$  es la constante de tiempo de los componentes rápidos de la corriente de compuerta ( $I_g$ ).
- $t_{slow}$ -Voltage(mV), donde  $t_{slow}$  es la constante de tiempo de los componentes lentos de la corriente de compuerta ( $I_g$ ).
- $t_{(I_k)}$ -Voltage(mV), donde  $t_{(I_k)}$  es la constante de tiempo de la corriente del canal.
- $A_{slow}^*$ -Voltage(mV), donde  $A_{slow}^*$  es la amplitud fraccional de los componentes lentos ( $A_{slow}^* = A_{slow} / (A_{fast} + A_{slow})$ )

La diferencia entre datos simulados y empíricos de cada conjunto  $S$  es normalizado mediante la división por la cantidad de datos del conjunto multiplicado por el máximo de ese conjunto ( $n_j \cdot \max x_j$ ).

En este contexto, las preguntas que nos podemos hacer son:

- ¿Se puede encontrar una estrategia de resolución para el problema planteado?
- ¿Esta estrategia devuelve como resultado el valor óptimo o un valor cercano al óptimo?
- ¿Qué tan cercano del óptimo?

Preguntas que definen a su vez el **objetivo de investigación**:

- Investigar la existencia de una estrategia de resolución para el problema de optimización planteado, que devuelva el valor óptimo o un valor muy cercano al óptimo.

Una adecuada estrategia de resolución permitirá conocer la distancia de entre los datos simulados que parten del modelo y los datos empíricos, y esta distancia encontrada llevaría a concluir si los datos empíricos son explicados, o no, por el modelo  $C-C-O$ .

## 4.3. Desarrollo del marco teórico

### 4.3.1. Optimización

No existe ningún método que resuelva todo tipo de problemas de optimización, por tal motivo se han desarrollado diferentes métodos que resuelven las diferentes clases del problemas.

Un **problema de optimización** consisten en encontrar los valores de las  $n$  variables

$$[x_1, x_2, \dots, x_n] = \vec{x} \quad (4.2)$$

que representan las variables de decisión, que satisfacen  $m$  condiciones de desigualdad y  $p$  condiciones de igualdad (restricciones), que optimizan la función objetivo  $f(\vec{x})$ :

Encontrar  $\vec{x}$  que maximice (minimice)  $f(\vec{x})$ , sujeto a:

$$g_i(\vec{x}) \leq 0; \quad i = 1, 2, \dots, m$$

$$h_j(\vec{x}) = 0; \quad j = 1, 2, \dots, p$$

donde  $g_i(\vec{x})$  y  $h_j(\vec{x})$  son las restricciones de desigualdad e igualdad respectivamente.

El anterior problema es denominado problema de optimización con restricciones. Cuando el problema es sin restricciones estas no quedan definidas en su formalización.

Las restricciones en el problema que nos ocupa son referentes a los intervalos en los que están definidas las 14 variables. Éstos son:

Var	$\alpha$	$\beta$	$\sigma$	$\varepsilon$	$\delta$	$\gamma$	z_alpha	d_alpha	z_delta
S	9	12	7	8	3	14	0.75	10	0.5
I	6	9	4	5	-2	3	0.15	0	0.0

Var	d_delta	AllD	$\theta_1$	AllE	$\theta_2$	LAG	FC	
S	10	0.8	10	7	10	5	10	
I	0	0.0	0	1	0	1	0	

Cuadro 4.1: Intervalo de los parámetros del modelo *C-C-O*.

Cuando un problema de optimización involucra a múltiples funciones objetivo es conocido



como problema de programación **multi-objetivo**. En el caso que nos ocupa, como se mencionó anteriormente, se ha transformado el problema multi-objetivo en uno **mono-objetivo** a partir de la ponderación de las diferentes funciones.

El encontrar el mínimo o el máximo de una función se refiere a encontrar el mínimo o máximo global, donde un mínimo  $f(\vec{x}')$  es un **mínimo global** si y sólo si para todo punto factible  $\vec{x}$ ,  $f(\vec{x}') \leq f(\vec{x})$ .

Un mínimo  $f(\vec{x}^l)$  es un **mínimo local** si y sólo si existe una vecindad factible de  $\vec{x}^l$ , definida como  $\|f(\vec{x}^l) - f(\vec{x})\| \leq \gamma$ , siendo  $f(\vec{x})$  factible y  $\gamma$  positivo, tal que para todo  $\vec{x}$  en esa vecindad  $f(\vec{x}^l) \leq f(\vec{x})$ .

### 4.3.2. Región factible

Las variables se pueden representar como un vector  $\vec{x}$ , que sería un punto en el espacio de  $n$  dimensiones. Al considerarse las restricciones de desigualdad  $g_i(\vec{x}) \leq 0$  se forma una superficie en el espacio  $(n-1)$ -dimensional que es llamada superficie de restricción. Al conjunto de todas las superficies de las restricciones se le conoce como **superficie compuesta de restricciones**.

Luego, al subconjunto de puntos que satisfacen las restricciones del problema se le llama **región factible**, y a cualquier de los puntos de esta región se le llama **punto factible**. A la inversa, los puntos que no caen en la intersección de las superficies establecidas por las restricciones, se les denomina **punto infactible**, y al subconjuntos de estos se le llama **región infactible**.

### 4.3.3. Convexidad

Un conjunto  $F$  es convexo si para todo  $\vec{a}, \vec{b} \in F$ , y para toda  $\theta \in [0, 1]$ :

$$(1 - \theta) \cdot \vec{a} + \theta \cdot \vec{b} \in F \quad (4.3)$$

Es decir,  $F$  es convexo si para dos puntos cualesquiera  $\vec{a}$  y  $\vec{b}$  del conjunto, el segmento rectilíneo que une a esos dos puntos pertenece también al conjunto. En la figura 4.3 se

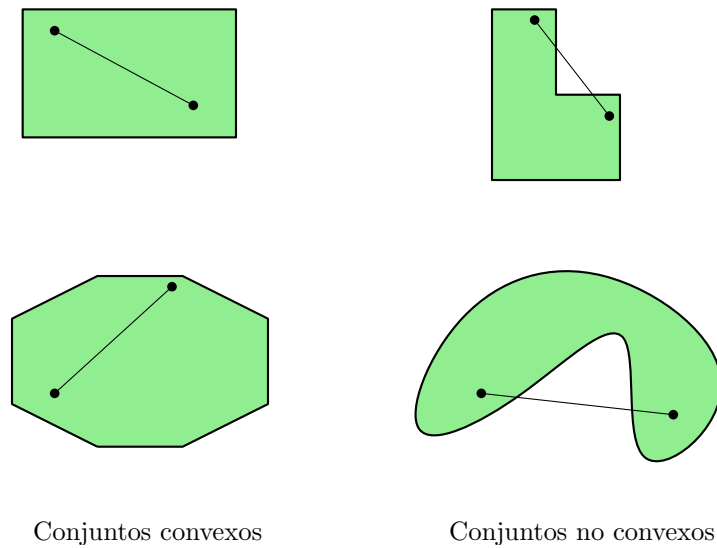


Figura 4.3: Conjuntos convexos y no convexos.

muestran unos ejemplos de conjuntos convexos y no convexos.

Demostrar la no convexidad de un conjunto de datos formados por  $n$  dimensiones se puede hacer iterando una por una las variables que definen la función, manteniendo las restantes  $n-1$  variables fijas, y observando el recorrido de la variable no fija. Si sólo una de las variables de la función tiene un recorrido no convexo, se puede decir que la función es no convexa. Para demostrar la no convexidad de la función que nos ocupa se realizó este proceso. En el apéndice C (Recorrido de variables, no convexidad) se muestran los gráficos de las variables que resultaron con recorridos no convexos. Aquí presentamos sólo una de ellas en la figura 4.4; el lector interesado puede recurrir al apéndice mencionado.

#### 4.3.4. Algoritmos de resolución

##### Métodos clásicos

Existen un conjunto de métodos clásicos para resolver problemas de optimización que pueden ser utilizados para encontrar el valor máximo o mínimo de una función. Son métodos para funciones de varias variables pero sin restricciones. Estos métodos presentan varias ventajas, a saber:

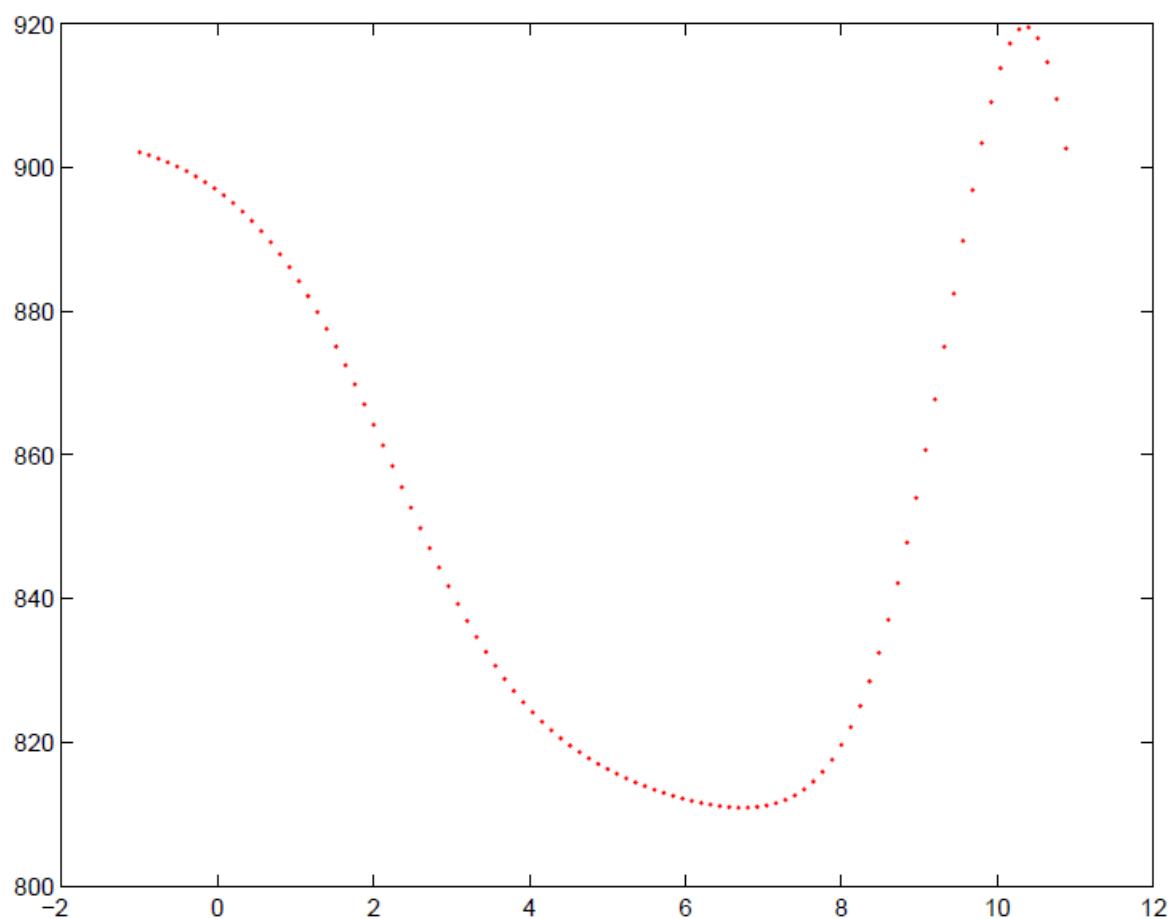


Figura 4.4: Recorrido de la variable  $\gamma$ .

- Asumen que la función es dos veces diferenciable respecto a cada variable.
- Asumen que todas las derivadas de la función son continuas.
- Generalmente requieren de un punto inicial (solución inicial) para comenzar la búsqueda.
- Al depender de un punto inicial no garantizan converger al óptimo global, especialmente en funciones con varios mínimos (o máximos).

Uno de estos métodos es el **Simplex**, propuesto por Spendley, Hext y Himsworth en 1962, y posteriormente mejorado por Nelder y Mead en 1965. Se le llama simplex a la figura geométrica formada por  $n + 1$  puntos en un espacio  $n$ -dimensional. Por ejemplo, en dos dimensiones el simplex es un triángulo, y en tres dimensiones es un tetraedro.

Este método va comparando el valor de la función objetivo en los  $n + 1$  vértices de un simplex general, para entonces mover el simplex de manera gradual mediante los operadores definidos como *reflexión*, *expansión* y *contracción*, convergiendo hacia un mínimo (o máximo) de manera iterativa. En cada iteración se va formando un nuevo simplex que reemplaza al anterior.

### Algoritmos genéticos

Debido fundamentalmente a que los métodos iterativos clásicos dependen del punto inicial de búsqueda y, por lo tanto, tienden a quedar estancados en mínimos globales en dependencia del punto inicial, se han propuesto otros métodos. Un ejemplo de ellos son los algoritmos genéticos (AG), los cuales en cada iteración generan un conjunto de soluciones factibles con el objetivo de analizar posibles soluciones en todo el espacio de búsqueda evitando el estancamiento.

Los AG son una heurística, por lo que pueden llegar a la solución óptima o sólo acercarse a la misma. En ese sentido, al igual que los métodos iterativos, no garantiza obtener la solución óptima. Son recomendados debido a sus buenos resultados en problemas complejos, no lineales, con restricciones, y con espacios de búsqueda continuos.

Se basan en la teoría de la evolución, donde sobreviven y dejan descendencia los mejores individuos; en nuestro caso, las mejores soluciones. Realizan una búsqueda en todo el espacio

y en cada iteración generan un conjunto de soluciones a partir de las anteriores y conservan las mejores. Presentan procesos estocásticos que actúan en las poblaciones y especies. Dichos procesos son la *reproducción*, la *mutación*, la *competencia* y la *selección*.

Así, sus componentes básicos son:

- *Fitness* o función de aptitud.
- Estrategia de la selección de padres.
- Las estrategias de recombinación y mutación en la formación de los hijos.

Cada individuo es una solución del problema, y cada uno de ellos tiene un *fitness*, que puede ser la evaluación de esa solución en la función. El algoritmo genera una población inicial, que comúnmente es un conjunto de soluciones factibles del problema, y teniendo en cuenta su *fitness* se seleccionan los padres de los hijos que formarán parte de la siguiente generación. La nueva población queda constituida por los individuos de mejor *fitness* entre la población actual y la nueva generación formada (la generación de los hijos).

Este proceso se repite, como se muestra en la figura 4.5, hasta llegar a un criterio de parada que puede ser, por ejemplo, el número de generaciones o un valor de *fitness* alcanzado.

## 4.4. Visualización del alcance del estudio

### 4.4.1. Estrategia propuesta

Debido a que nuestro problema es de variables continuas con valores en los Reales, y complejo en el sentido de que presenta 14 dimensiones, no es posible resolverlo de forma exhaustiva con un algoritmo exacto. Fueron estudiados en el marco teórico los algoritmos iterativos y una heurística que analiza un conjunto de soluciones en cada iteración.

Si nuestra función hubiese resultado convexa, un algoritmo iterativo, como es el caso del Simplex, resultaría adecuado para su resolución, debido a su rápida convergencia y a que el mínimo encontrado, al existir sólo uno, fuera el mínimo global. Sin embargo, se demostró que la función objetivo no es convexa por lo que utilizando sólo un algoritmo iterativo se tiene el riesgo de que se estanque en mínimos locales en dependencia del punto de partida inicial.

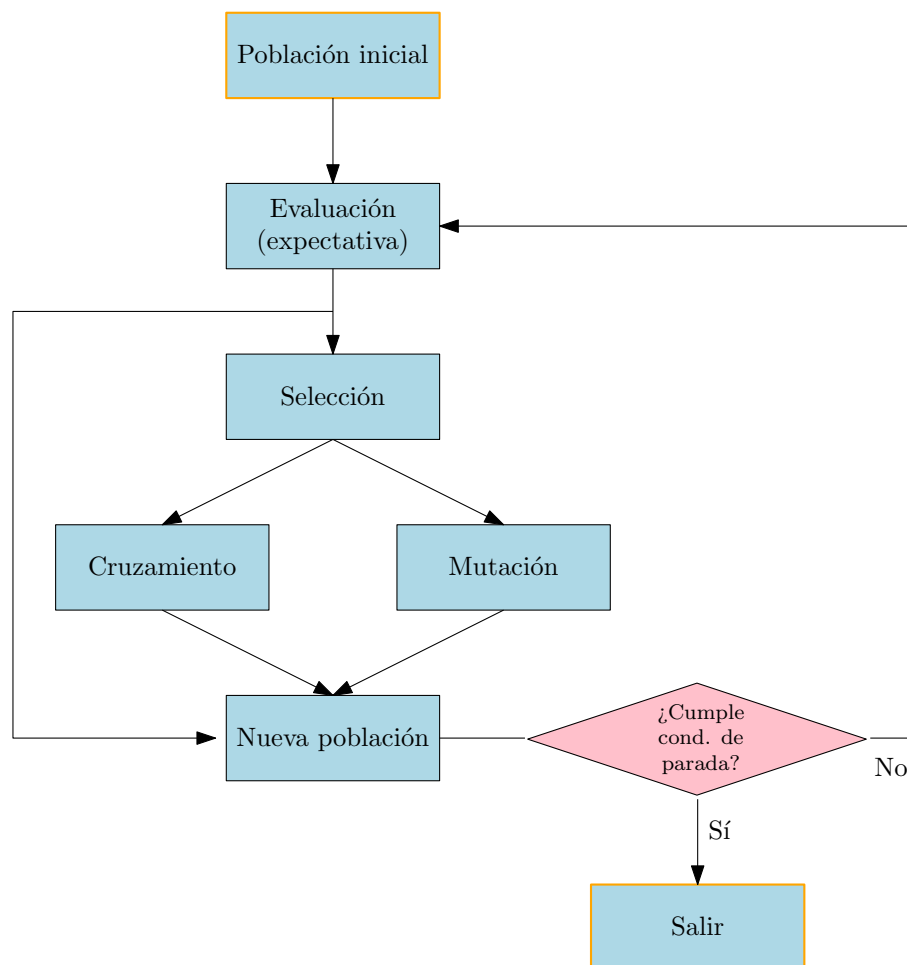


Figura 4.5: Algoritmo genético.

Por otro lado, si se utiliza el AG, a pesar de buscar soluciones factibles en todo el espacio de búsqueda, no convergería tan rápido como un algoritmo iterativo, teniendo una importante demora en soluciones buenas.

Por lo anterior, la estrategia que se propone es:

- a) Comenzar con un Algoritmo Genético, que al realizar una búsqueda global, se espera se acerque a una vecindad del mínimo global.
- b) El resultado del algoritmo anterior, luego de un número determinado de generaciones, utilizarlo como entrada para un algoritmo iterativo, que realiza una búsqueda local y converge más rápido, y el mínimo cercano debe estar en una vecindad del global, o ser el global.

Cada solución (individuo) estaría representada por un conjunto de 14 valores de parámetros, y teniendo en cuenta las restricciones de intervalos se podrían generar individuos factibles. Los datos y operadores propuestos para el AG son:

Tamaño de población: En la literatura existen varias propuestas sobre cómo determinar el tamaño de la población, se cree que está relacionado con la cantidad de variables de la función, pero no hay nada demostrado ni establecido. Lo que sí se ha comprobado es que según va aumentando el tamaño de la población, aunque converge más lento, da soluciones más precisas. En las primeas pruebas iniciales, para 300 generaciones hemos obtenido los mejores valores en poblaciones de 150 individuos.

Creación de la población inicial: predeterminada que cumpla ciertas restricciones, o aleatoria factible.

Escala del fitness (evaluación en la función objetivo), para ser utilizada por la función de selección: Proporcional, calcula unas expectativas (probabilidad de ser escogido para la competencia en la selección de los padres; en nuestro caso, probabilidad de participar en el torneo).

Selección de los padres: Torneo de 4, se selecciona cada padre por competencia de 4 elegidos

al azar según su expectativa.

Fracción de cruce: en el proceso de reproducción, especifica la proporción de la próxima generación que se genera por cruce entre padres, el resto son generados por mutación. Se recomienda una fracción de 0.99, quedando el 0.01 para la mutación.

Mutación: se realiza a través de cambios en individuos, con o sin éxito, de la generación anterior, formando nuevos individuos que sean también factibles (que cumplan los intervalos establecidos).

Cruzamiento: método *Heuristic*, que crea hijos que se encuentran al azar en la línea que contiene los dos padres, a poca distancia del padre con mejor *fitness* y lejos del padre con el peor *fitness*.

## 4.5. Elaboración de hipótesis

Las variables de estudio son los valores de los parámetros que definen los valores de las funciones simuladas y los datos empíricos (ambos conjuntos como variables independientes), y los errores mínimos encontrados para esos datos que devuelve la estrategia propuesta (como datos dependientes). Por lo que consiste en una metodología correlacional entre variables dependientes e independientes.

### Hipótesis:

- a) La estrategia de resolución propuesta es adecuada para el problema, donde para datos simulados exactos y como máximo 4500 iteraciones del AG y 2000 del Simplex, devuelve resultados en una vecindad del óptimo con valor de  $1 \cdot 10^{-2}$ .



## 4.6. Pruebas iniciales

Las pruebas iniciales no se realizaron con los datos experimentales, pues con el fin de validar la estrategia no es recomendable utilizar los mismos. Esto ocurre porque si los resultados no son los esperados, no necesariamente es producto de que la estrategia sea incorrecta, sino que pueden influir otros factores, a saber, que el modelo teórico *C-C-O* no sea válido, o que los datos experimentales contengan errores.

Por tanto, para las pruebas iniciales, partiendo de funciones con las mismas características (mismas variables, misma estructura de la función de mínimos cuadrados, rangos de las variables, etc.), se generaron soluciones factibles que fueron utilizadas en sustitución de los datos empíricos.

Los resultados iniciales parecen indicar la adecuación de la estrategia para la función mono-objetivo, dando resultados de errores en la vecindad esperada. En la figura 4.6 se muestran los resultados de varias corridas de la estrategia con un conjunto de datos exactos. Y en la figura 4.7 se muestra un recorrido, con los valores que devolvió el AG, y luego la mejora que realizó el algoritmo Fminsearch (algoritmo iterativo que es una variante del Simplex).

Sin embargo, debe tenerse en cuenta que ésta no es una investigación terminada, debido a que faltan las fases finales definidas en la investigación cuantitativa, como son la recogida de datos, sistematización de los datos, análisis estadístico, comprobación de hipótesis y resultados.

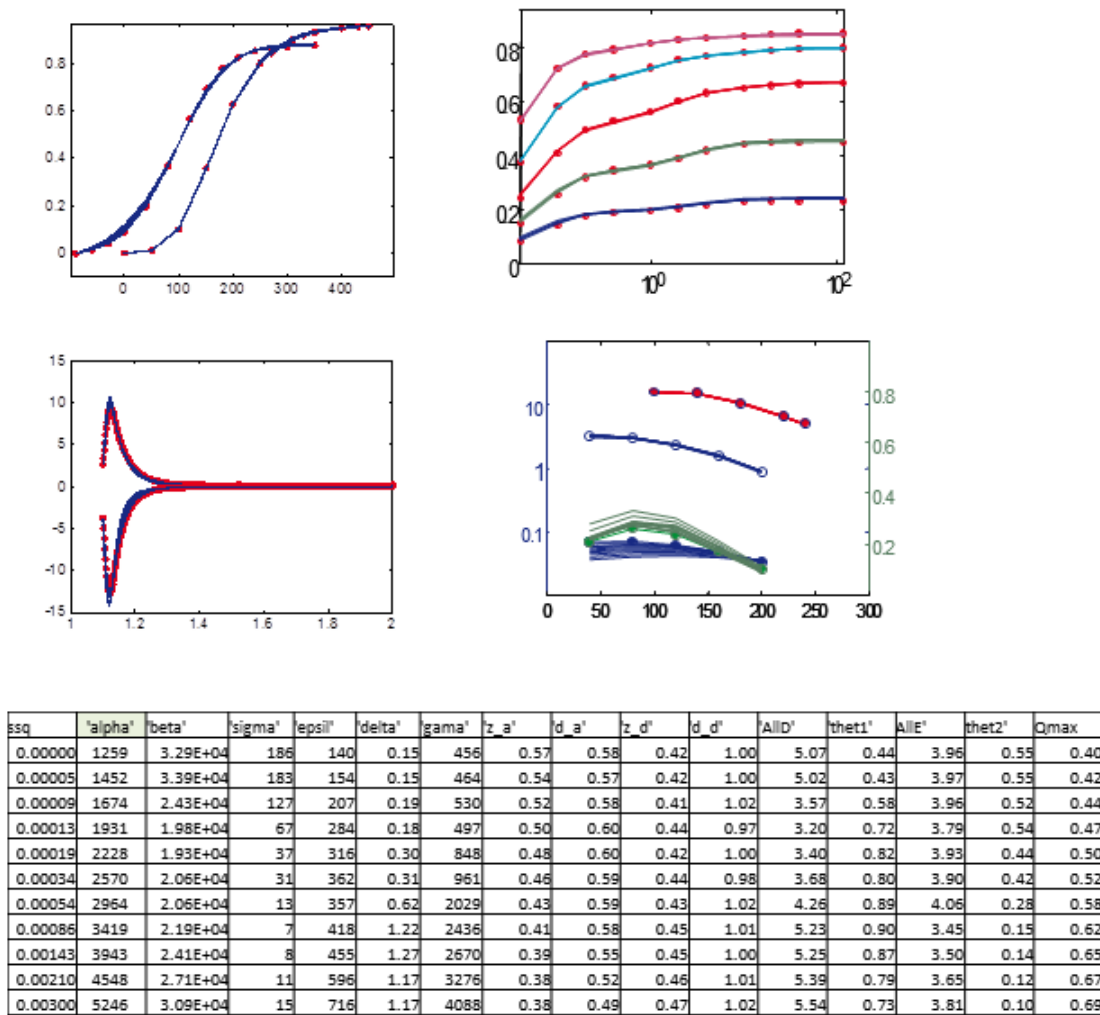


Figura 4.6: Resultados con datos exactos.

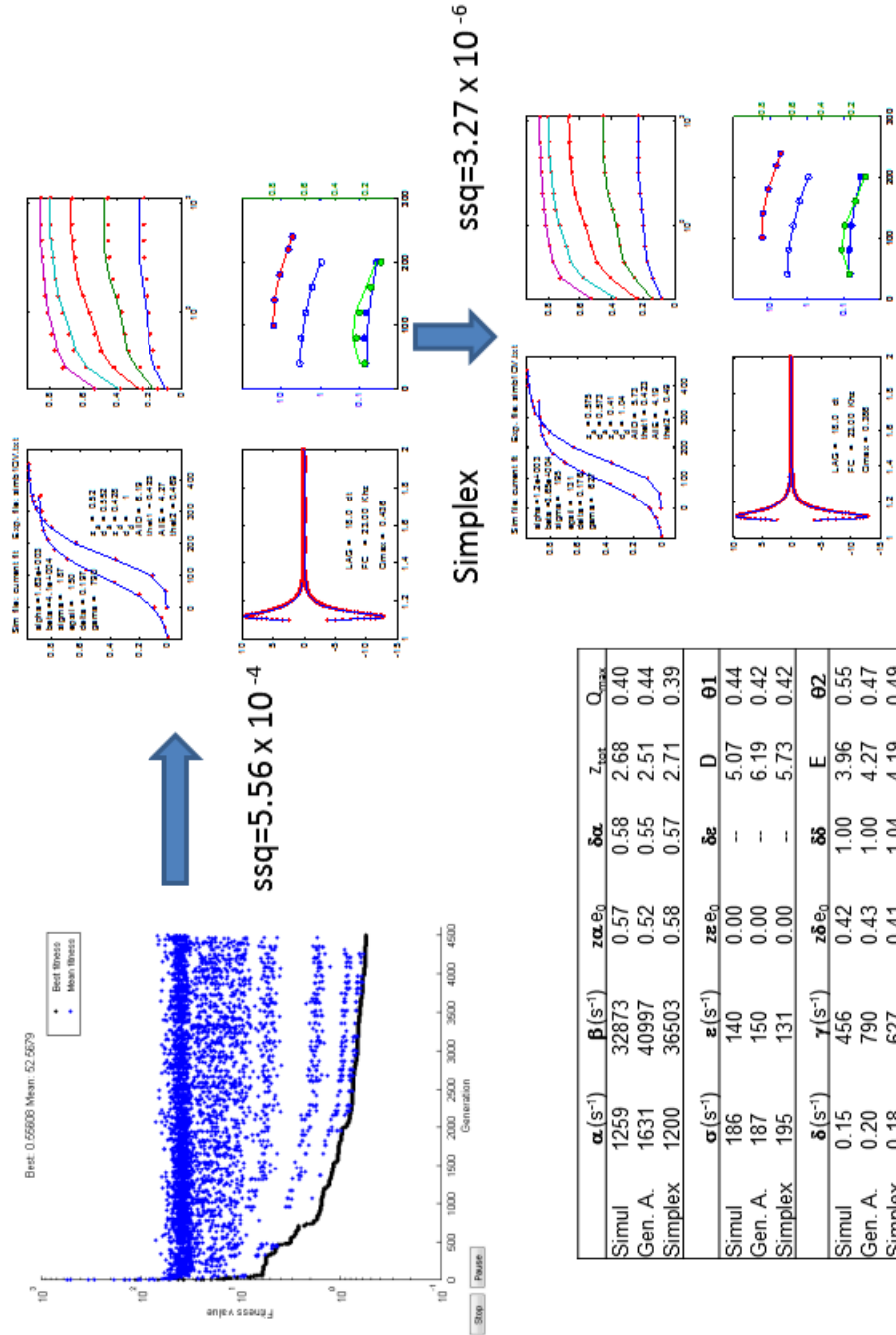


Figura 4.7: Recorrido de uno de los resultados.

## Capítulo 5

### Resultados y discusión: subproyecto informático

## 5.1. Resultados de la fase de Análisis

### 5.1.1. Requisitos

Utilizaremos una clasificación de los requisitos que propone la técnica del Despliegue de la Función de Calidad (DFC) <sup>1</sup>, clasificación que se realiza, según Pressman, para “*entender lo que resulta valioso para el cliente*” (Pressman, 2001, p.186).

Identifica tres tipos de requisitos: los requisitos normales donde se declaran objetivos y metas para el producto durante las reuniones con el cliente, los requisitos esperados que son implícitos pero fundamentales para el correcto funcionamiento del mismo, y los requisitos innovadores que son características del producto que van más allá de las expectativas del cliente. Así, para nuestro producto los requisitos serían:

#### Requisitos normales

- R1: El sistema debe permitir la selección de la recomendación.
- R2: El sistema debe permitir la selección de opciones iniciales para la recomendación, como las:
  - R2.1: Funciones a tener en cuenta.
  - R2.2: Pesos de las funciones.
  - R2.3: Condiciones de parada:
    - Cantidad de iteraciones o generaciones de cada algoritmo.
    - Tamaño de la población.
    - Error de satisfacción.
- R3: Debe permitir la carga de archivos de datos empíricos.
- R4: Debe graficar la variación del error durante el proceso de búsqueda de parámetros para la recomendación.
- R5: Debe devolver el conjunto de valores que representen el menor error encontrado, esos valores serían la recomendación de valores de parámetros.
- R6: Debe graficar las funciones de variables fisiológicas con los valores simulados y empíricos resultado de la recomendación.
- R7: Debe permitir cargar reportes salvados de procesos anteriores o cualquier reporte

---

<sup>1</sup>Técnica de calidad que traduce las necesidades del cliente en requisitos técnicos del *software*, originalmente desarrollado en Japón y usado por primera vez en Kobe Shipyard of Mitsubishi Heavy Industries, Ltd.

que cumpla la sintaxis que se establezca en la fase de diseño.

- R8: Debe permitir visualizar la simulación de los datos que contiene un reporte cargado.
- R9: Debe permitir la salva de reportes del proceso de recomendación.

### Requisitos esperados

- El sistema debe comenzar a ejecutar una acción seleccionada por el usuario (especialmente *Recomendar*, *Simular* y *Salvar*) antes de transcurridos 5 segundos de espera (relacionado con la subcaracterística de *Comportamiento en el tiempo* de la característica de *Eficiencia* en la norma ISO 9126 <sup>2</sup>).
- El sistema debe ejecutar la acción seleccionada por el usuario y no otra, es decir, debe desempeñar las tareas para las que fue creado (relacionado con la subcaracterística de *Idoneidad* de la característica de *Funcionalidad* en la norma ISO 9126).
- El sistema no debe “caerse” mientras esté en ejecución (relacionado con la subcaracterística de *Madurez* de la característica de *Fiabilidad* en la norma ISO 9126).
- Que el sistema esté implementado de manera que permita la inclusión de módulos, que representen otros modelos teóricos, con el menor cambio en los módulos funcionales ya implementados (relacionado con la subcaracterística de *Facilidad de cambio* de la característica de *Mantenibilidad* en la norma ISO 9126).

### Requisitos innovadores

- R10: El sistema debe, al inicio del proceso de recomendación, permitir que el usuario decida que se utilicen los dos algoritmos de la estrategia propuesta, o sólo uno de ellos.
- R11: Debe permitir cambiar manualmente valores de parámetros (proceso de refinamiento).
- R12: Debe permitir que los cambios anteriores sea especificados en la misma ventana de la graficación de las funciones.
- R13: Debe mostrar en la graficación de las funciones simuladas el error global.
- R14: Debe permitir la salva de reportes del proceso de refinamiento cada vez que el usuario lo requiera.

---

<sup>2</sup>ISO 9126 es un estándar internacional para la evaluación de la calidad del *software*

### 5.1.2. Casos de uso

Las tareas seleccionadas en esta fase son la modelación y descripción de casos de uso que representan la perspectiva del usuario. Los diagramas de casos de uso organizan los comportamientos del sistema y representa un conjunto de casos de uso y actores y sus relaciones. Muestran, por tanto, los distintos requisitos funcionales que se esperan de una aplicación o sistema y cómo se relacionan con su entorno (usuarios en este caso). En la figura 5.1 se presenta el diagrama de alto nivel de casos de uso del producto final.

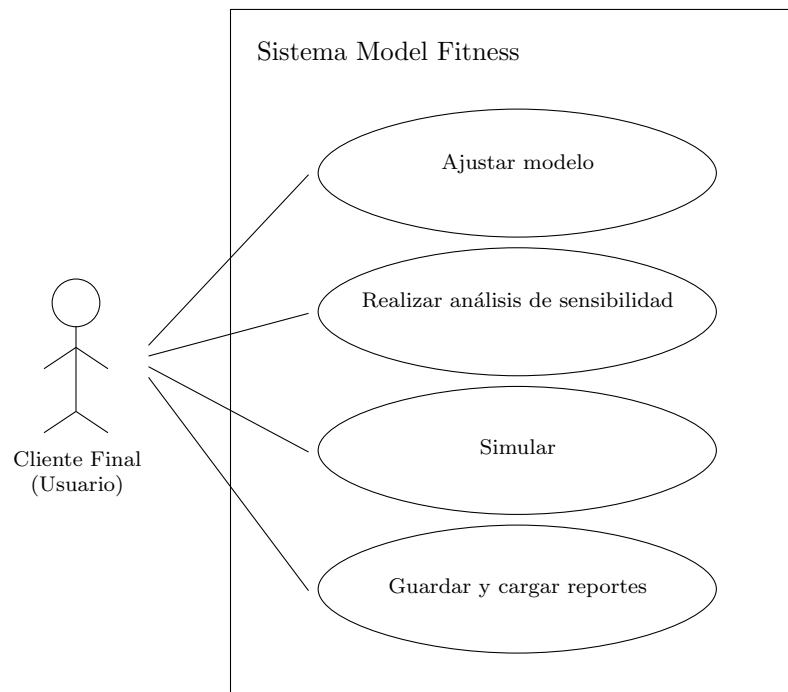


Figura 5.1: Diagrama casos de uso del producto.

### Descripción de los casos de uso

En las siguientes tablas: 5.1.2, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, se muestran las descripciones de los casos de uso identificados para el sistema.

CU01	- Ajustar modelo.
Descripción general	- Permite al usuario obtener una recomendación de valores para los parámetros del modelo que sea la mínima encontrada entre los datos simulados desde modelo y los datos empíricos cargados al sistema.
Actores	- Investigadores del departamento de Estructura y Funciones de Sensores Moleculares del CINV (usuarios).
Gatillo	- El usuario desea obtener valores recomendados.
Precondiciones	- El usuario debe contar con los ficheros de los datos experimentales.
Flujo normal de eventos	<ul style="list-style-type: none"> <li>■ El sistema muestra una ventana al usuario con las diferentes opciones.</li> <li>■ El usuario selecciona el modelo a utilizar.</li> <li>■ El usuario selecciona el proceso de recomendación.</li> <li>■ El usuario selecciona las funciones fisiológicas a tener en cuenta.</li> <li>■ El usuario especifica los valores de pesos de las diferentes funciones.</li> <li>■ El usuario carga el camino de cada uno de los ficheros de los datos empíricos.</li> <li>■ El usuario selecciona la estrategia de resolución.</li> <li>■ El usuario especifica los valores de las variables de la estrategia de resolución.</li> <li>■ El usuario selecciona la opción que inicia el proceso de ajuste.</li> </ul>

Cuadro 5.1: Caso de uso 01 (Primera Parte).



Flujo normal de eventos	<ul style="list-style-type: none"> <li>■ Si el usuario no especifica o selecciona los valores de los pasos indicados anteriormente, el sistema utiliza los valores que están por defecto.</li> <li>■ Si los datos ingresados no cumplen el formato y/o el usuario no ha ingresado los ficheros de datos empíricos, el sistema envía un error.</li> <li>■ El sistema comienza el proceso de ajuste para la recomendación.</li> <li>■ Durante el proceso de ajuste el sistema grafica los valores mínimos encontrados.</li> <li>■ El sistema devuelve los valores de recomendación una vez que se cumplan las condiciones de parada.</li> </ul>
Excepciones	- Si alguno de los datos ingresados no cumple el formato y/o si no ha cargado los ficheros de datos empíricos, el sistema debe enviar error.
Postcondiciones	- Devolución de los valores de los parámetros y la graficación de las funciones con los datos empíricos y los simulados desde el modelo.

Cuadro 5.2: Caso de uso 01 (Segunda Parte).

CU02	- Realizar análisis de sensibilidad.
Descripción general	- Permite al usuario modificar manualmente los diferentes valores de los parámetros, permitiendo luego en la simulación ver la implicancia de cada uno de los parámetros en el resultado global.
Actores	- Investigadores del departamento de Estructura y Funciones de Sensores Moleculares del CINV (usuarios).
Gatillo	- El usuario desea especificar, él mismo, los valores de los parámetros.
Precondiciones	- El usuario debe contar con los ficheros de los datos experimentales.
Flujo normal de eventos	<ul style="list-style-type: none"> <li>■ El sistema muestra una ventana donde el usuario pueda especificar los valores de los parámetros.</li> <li>■ El usuario especifica cada uno de los valores de los diferentes parámetros.</li> <li>■ El usuario selecciona la opción que ejecuta la simulación.</li> <li>■ El sistema realiza la simulación.</li> </ul>
Excepciones	- Si alguno de los valores de parámetros ingresados no cumple el formato, el sistema debe enviar error al usuario.
Postcondiciones	- Los valores de los parámetros quedan cargados en el sistema.

Cuadro 5.3: Caso de uso 02.

CU03	- Simular.
Descripción general	- Permite al usuario graficar las funciones fisiológicas desde los valores de los parámetros que él decida, o desde los valores recomendados por el sistema.
Actores	- Investigadores del departamento de Estructura y Funciones de Sensores Moleculares del CINV (usuarios).
Gatillo	- El usuario desea visualizar los resultados de la simulación desde los valores de parámetros que especificados o recomendados.
Precondiciones	- El usuario debe contar con los ficheros de los datos experimentales y haber, o cargado un reporte salvado, o especificado los valores de los parámetros que desea simular, o ejecutado el proceso de recomendación.
Flujo normal de eventos	<ul style="list-style-type: none"> <li>■ El sistema muestra una ventana con la opción que ejecuta la simulación.</li> <li>■ El usuario selecciona la opción.</li> <li>■ El sistema realiza la simulación.</li> <li>■ El sistema grafica las funciones fisiológicas desde la simulación de los datos desde el modelo y los datos experimentales, en la misma ventana donde se encuentra.</li> </ul>

Cuadro 5.4: Caso de uso 03 (Primera Parte).

Excepciones	- Si la simulación se selecciona desde valores especificados por el usuario y alguno de los valores ingresados no cumple el formato, el sistema debe enviar error al usuario.
Postcondiciones	- Graficación de las funciones con los datos empíricos y los simulados, a partir de los valores de parámetros actuales.

Cuadro 5.5: Caso de uso 03 (Segunda Parte).

CU04	- Salvar y cargar reporte.
Descripción general	- Permite al usuario salvar los resultados hasta ahora obtenido en un fichero .mat, y cargar en el sistema un reporte guardado anteriormente.
Actores	- Investigadores del departamento de Estructura y Funciones de Sensores Moleculares del CINV (usuarios).
Gatillo	- El usuario desea salvar los resultados de la recomendación o la simulación para poderlos cargar posteriormente en el sistema.
Precondiciones	- Para salvar, el usuario debe contar con resultados de recomendación o de refinamiento, y para cargar debe contar con un reporte salvado anteriormente.

Cuadro 5.6: Caso de uso 04 (Primera Parte).

Flujo normal de eventos	<ul style="list-style-type: none"> <li>■ El sistema presentará en la ventana de simulaciones un menú con las opciones que permitan salvar reporte o cargar reporte, y que puedan ser seleccionadas por el usuario.</li> <li>■ El usuario selecciona la opción que desee.</li> <li>■ El sistema realiza la salva con el nombre indicado por el usuario, o muestra una ventana al usuario para que indique el fichero a cargar.</li> <li>■ Si el usuario seleccionó la carga de reporte e indicó el camino del fichero, el usuario selecciona entonces la opción donde acepta el fichero seleccionado.</li> <li>■ Si se aceptó un fichero a cargar, el sistema cargará el fichero con sus datos y simulará los mismos.</li> </ul>
Excepciones	- Si no hay resultados de recomendación o de simulación y el usuario selecciona la salva de reporte, o si el fichero a cargar no cumple el formato, el sistema debe enviar un error.
Postcondiciones	- Salva los datos y permite que el usuario siga haciendo modificaciones en el sistema; o carga datos desde un fichero con su posterior simulación.

Cuadro 5.7: Caso de uso 04 (Segunda Parte).

La trazabilidad de los requisitos respecto a los diferentes casos de uso queda definida en la tabla 5.8.

CU	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13
CU1	X	X	X	X	X				X				
CU2										X	X		
CU3						X		X				X	
CU4							X						X

Cuadro 5.8: Trazabilidad (CU x Req).

## 5.2. Resultados de la fase de Diseño

En la figura 5.2, se muestra el diagrama de clases, resultante de esta fase de Diseño. En éste se muestran las clases con sus atributos y sus métodos, y la relaciones de herencia y utilización entre las diferentes clases.

En la figura 5.3 se muestra el mismo diagrama pero señaladas las clases que representan el módulo principal para las funcionalidades del recomendador y simulador de parámetros, y externamente el modelo que utiliza en este caso, que es el modelo ModelBK\_3, como un módulo externo. Así, se podría agregar todos los modelos que desee el cliente, siempre que se implementen como modelos que heredan de la clase Model y contengan la respectiva implementación de los métodos diseñados en dicha clase.

La descripciones de las clases del diseño se muestran las figuras 5.4, 5.5, 5.8, 5.6 y 5.7.

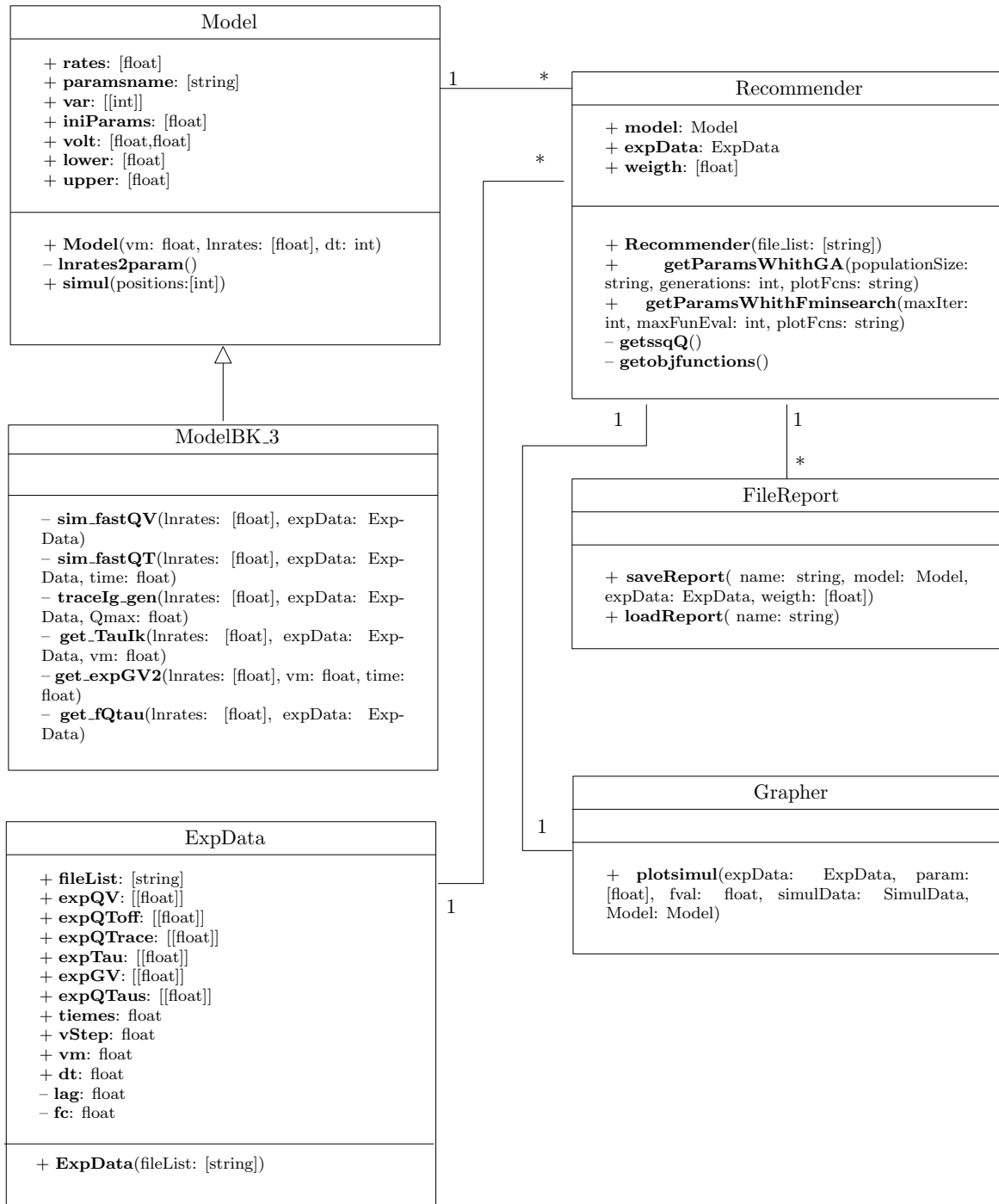


Figura 5.2: Diagrama de clases.

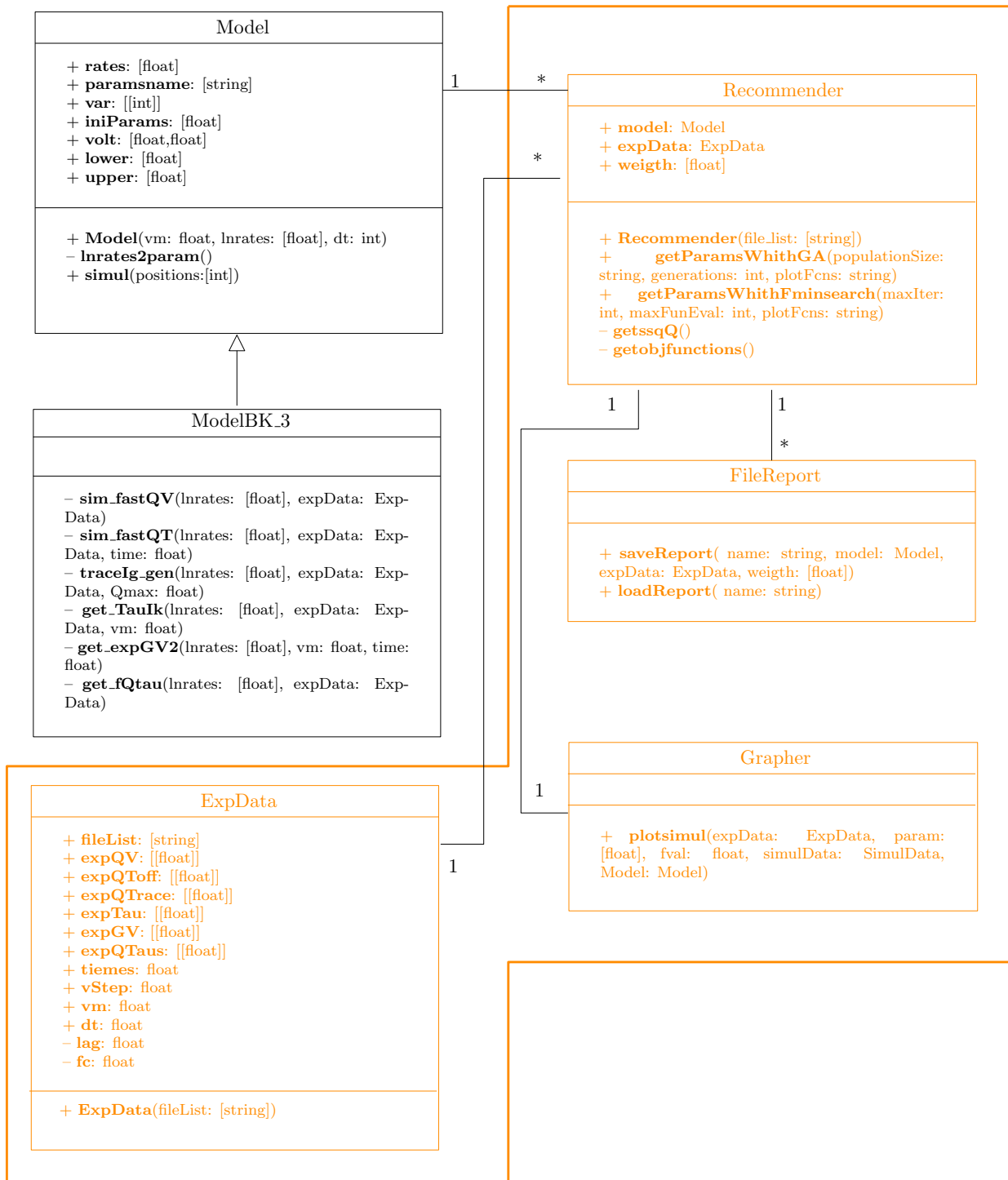


Figura 5.3: Diagrama de clases con módulo principal.



Model
<p>% Valores de las velocidades de transición entre estados del modelo.  <b>rates:</b> [float]</p> <p>%Posiciones de los diferntes parámetros.  <b>var:</b> [[int]]</p> <p>%Nombres de los parámetros.  <b>paramsname:</b> [string]</p> <p>%Valores de los parámetros iniciales.  <b>iniParams:</b> [float]</p> <p>%Voltajes.  <b>volt:</b> [float,float]</p> <p>%Valores mínimos de los parámetros.  <b>lower:</b> [float]</p> <p>%Valores máximos de los parámetros.  <b>upper:</b> [float]</p>
<p>%Constructor, inicializa la clase.  <b>Model</b>(vm: float, lnrates: [float], dt: int)</p> <p>%Actualiza los lnrates.  <b>lnrates2param()</b></p> <p>%Simula valores comparables con los experimentales, con las variables definidas en el modelo.  <b>simul():</b>[int]</p>

Figura 5.4: Descripción de la clase Model.

ModelBK_3
<pre> %Devuelve los valores simulados de QV. <b>sim_fastQV</b>(lnrates: [float], expData: Exp- Data):[float]  %Devuelve los valores simulados de QT. <b>sim_fastQT</b>(lnrates: [float], expData: Exp- Data, time: float):[float]  %Devuelve los valores simulados de Ig. <b>traceIg_gen</b>(lnrates: [float], expData: Exp- Data, Qmax: float):[float]  %Devuelve los valores simulados de Tau. <b>get_TauIk</b>(lnrates: [float], expData: ExpData, vm: float):[float]  %Devuelve los valores simulados de GV. <b>get_expGV2</b>(lnrates: [float], vm: float, time: float):[float]  %Devuelve los valores simulados de Qtau. <b>get_fQtau</b>(lnrates: [float], expData: Exp- Data):[float] </pre>

Figura 5.5: Descripción de la clase ModelBK\_3.

FileReport
<pre> %Salva un fichero .mat que por defecto tiene la fecha actual en su nombre, que contiene Model, la estructura expData, los valores de los parámetros, los pesos de las funciones y el error encontrado. <b>saveReport</b>( name: string, model: Model, expData: ExpData, weigth: [float])  %Carga un fichero .mat creando un Model, un ExpData, una lista de valores de los parámetros, una lista de pesos y un float que contiene el er- ror, desde los datos cargados. <b>loadReport</b>( name: string): model: Model, exp- Data: ExpData, paramsGlob: [float], weigth: [float], fval: float </pre>

Figura 5.6: Descripción de la clase Report.

Grapher
<p>%Grafica las funciones mostrando datos experimentales y simulados.</p> <p><b>plotfit</b>(expData: ExpData, param: [float], fval: float, simulData: SimulData, Model: Model)</p>

Figura 5.7: Descripción de la clase Grapher.

Recommender
<p>%Modelo a partir del cual se har la recomendación.</p> <p><b>model</b>: Model</p> <p>%Estructura de datos experimentales.</p> <p><b>expData</b>: ExpData</p> <p>%Arreglo donde se guardarn los pesos asignados por el usuario a las diferentes funciones.</p> <p><b>weighh</b>: [float]</p>
<p>%Constructor, inicializa la clase.</p> <p><b>Recommender</b>(file_list: [string])</p> <p>%Obtiene recomendación utilizando el algoritmo genético.</p> <p><b>getParamsWhithGA</b>(populationSize: string, generations: int, plotFcns: string)</p> <p>%Obtiene recomendación utilizando el algoritmo Simplex.</p> <p><b>getParamsWhithSimplex</b>(maxIter: int, maxFunEval: int, plotFcns: string)</p> <p>%Calcula la suma total de las diferencias normalizadas al cuadrado entre los datos experimentales y los simulados de todas las funciones, asignado un peso a cada una de ellas.</p> <p><b>getssqQ</b>():float</p> <p>%Calcula las diferencias normalizadas al cuadrado entre los datos experimentales y los simulados de cada una de las funciones.</p> <p><b>getobjectfunctions</b>():[float]</p>

Figura 5.8: Descripción de la clase Recommender.

La estructura de datos que representa la simulación de los datos contenidos en los ficheros de variables fisiológicas (experimentales), se muestra en la figura 5.9.

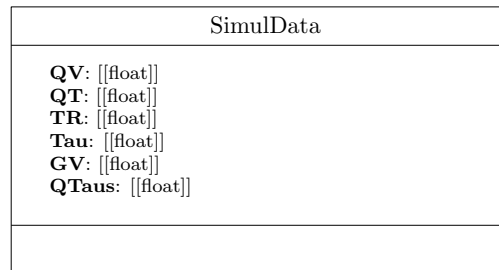


Figura 5.9: Diagrama de la estructura de datos simulados.

Un ejemplo de fichero de datos a cargar, el cual está constituido por dos columnas de datos float, se muestra en la figura 5.10.

	40	80	120	160	200
0	0.08327969	0.147595438	0.243807408	0.375742648	0.533584307
0.04	0.142756362	0.257000208	0.410297501	0.579779107	0.724537386
0.1	0.176570531	0.320670888	0.495467687	0.658526996	0.773337038
0.2	0.189514283	0.346733944	0.528172136	0.68778384	0.792172157
0.4	0.197454882	0.367801181	0.564328634	0.726404259	0.816224728
1	0.206814063	0.392051879	0.600524477	0.75387613	0.828002673
2	0.218688016	0.419829334	0.631221851	0.768729865	0.834331733
4	0.230515885	0.443063817	0.64970808	0.780618032	0.843453379
10	0.232998886	0.448734842	0.658733244	0.790499131	0.849549939
20	0.23337766	0.451524424	0.665920341	0.797150054	0.851989542
40	0.23341544	0.452380432	0.668732439	0.798939451	0.852242149
120					

Figura 5.10: Ejemplo de fichero de datos experimentales (carga y tiempo).

Los datos del fichero .mat que se guardará como reporte contiene, por defecto, la fecha incluida en el nombre, y en su contenido guarda los datos de los siguientes objetos:

- Model
- ExpData
- ParamsGlob
- Weigth
- Fval

Su estructura queda modelada en la figura 5.11.

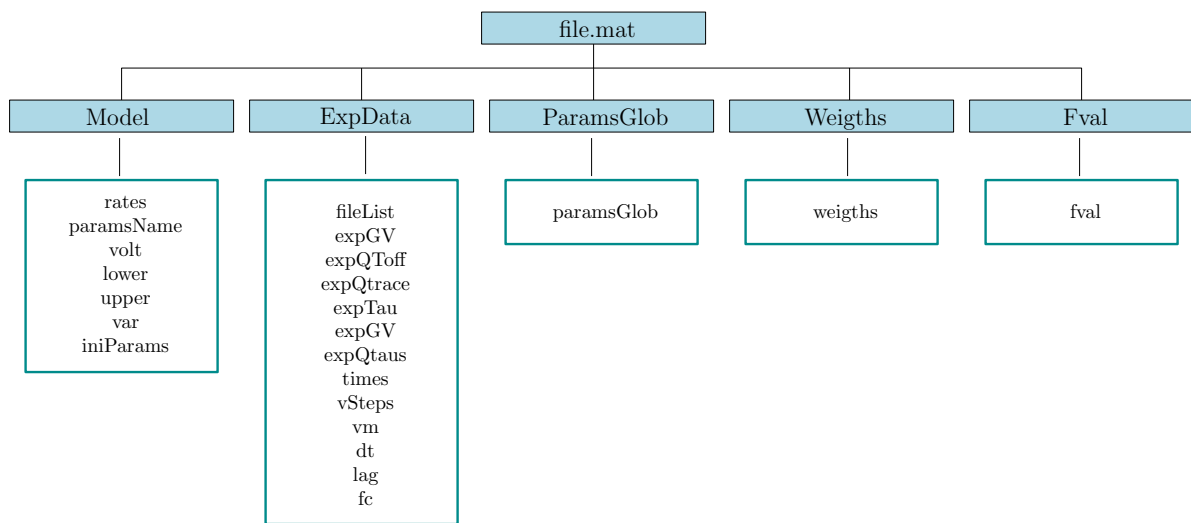


Figura 5.11: Estructura del fichero que se salva como reporte.

Con el objetivo de modelar los aspectos dinámicos del producto, desde una vista del comportamiento, se diseñó un diagrama de actividades. Los diagramas de actividades son importantes para modelar la función del sistema, así como para resaltar el flujo de control entre objetos. Las actividades en la utilización del producto quedan representadas en el diagrama de la figura 5.12.

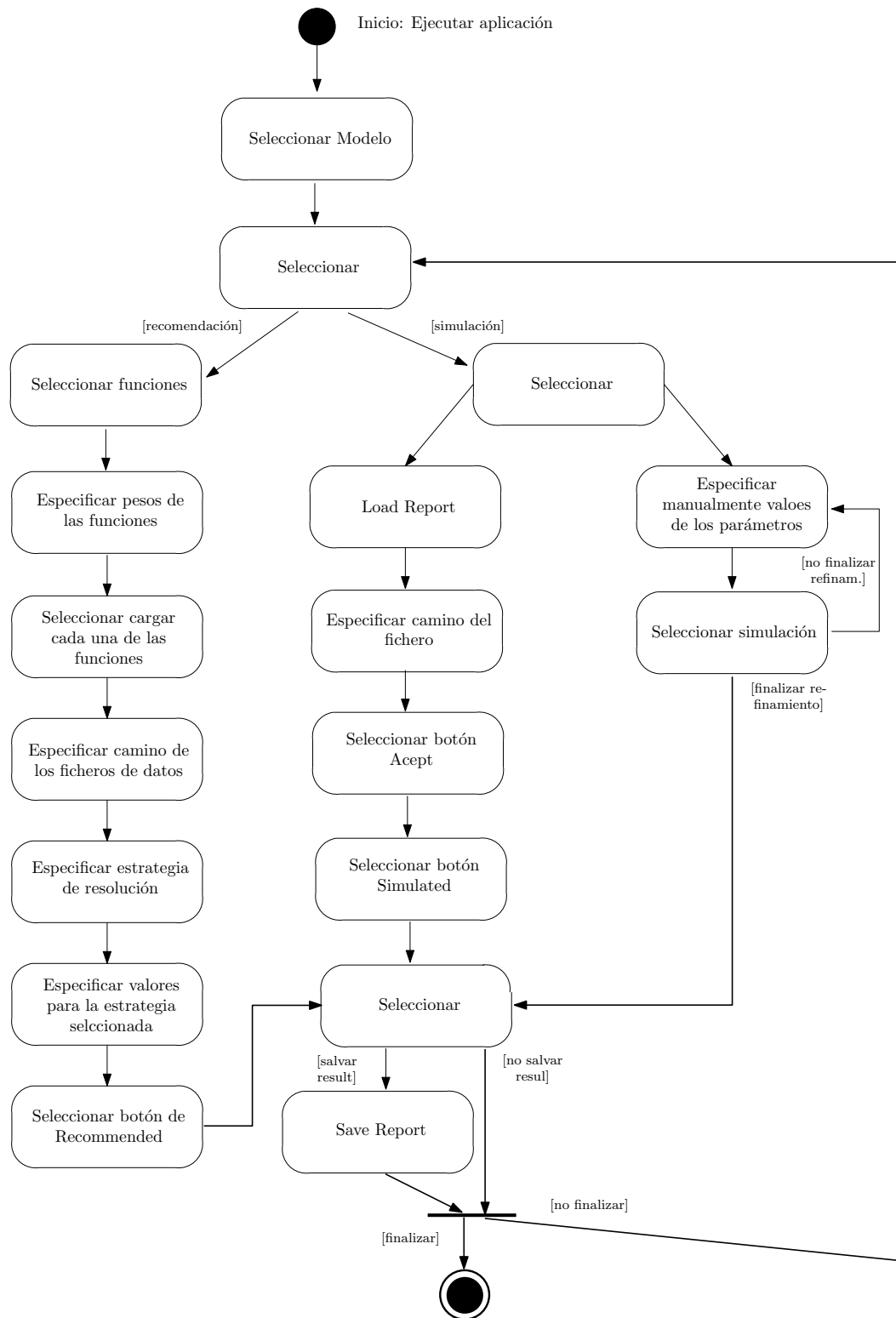
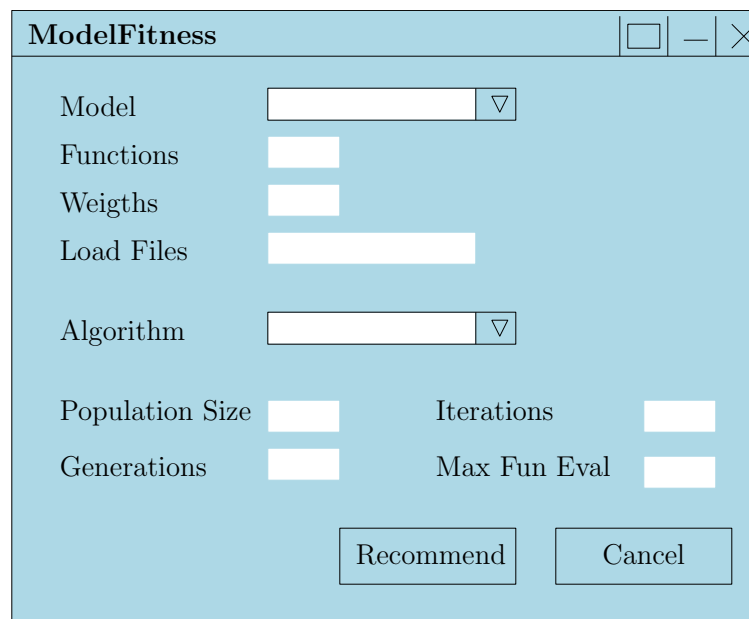


Figura 5.12: Diagrama de actividades.

Para mostrar los aspectos estructurales y de comportamiento desde la vista del entorno, se realizó un diseño inicial de la interfaz hombre-máquina, el cual está formado por 2 ventanas. Una ventana inicial para los datos solicitados en vista a los procesos de recomendación o refinamiento (ver figura 5.13). Y una segunda ventana que muestra los resultados de la recomendación, y permite, mediante la modificación manual de los datos de los parámetros, realizar el proceso de refinamiento y obtener una muestra gráfica de la simulación de los datos en las diferentes funciones fisiológicas (ver figura 5.14).



The image shows a software window titled "ModelFitness" with a light blue background. The window contains several input fields and buttons. At the top right are standard window controls (minimize, maximize, close). The main area has the following elements:

- Model:** A dropdown menu.
- Functions:** A text input field.
- Weighths:** A text input field.
- Load Files:** A text input field.
- Algorithm:** A dropdown menu.
- Population Size:** A text input field.
- Iterations:** A text input field.
- Generations:** A text input field.
- Max Fun Eval:** A text input field.
- Recommend:** A button.
- Cancel:** A button.

Figura 5.13: Diseño de la interfaz: Recomendador.

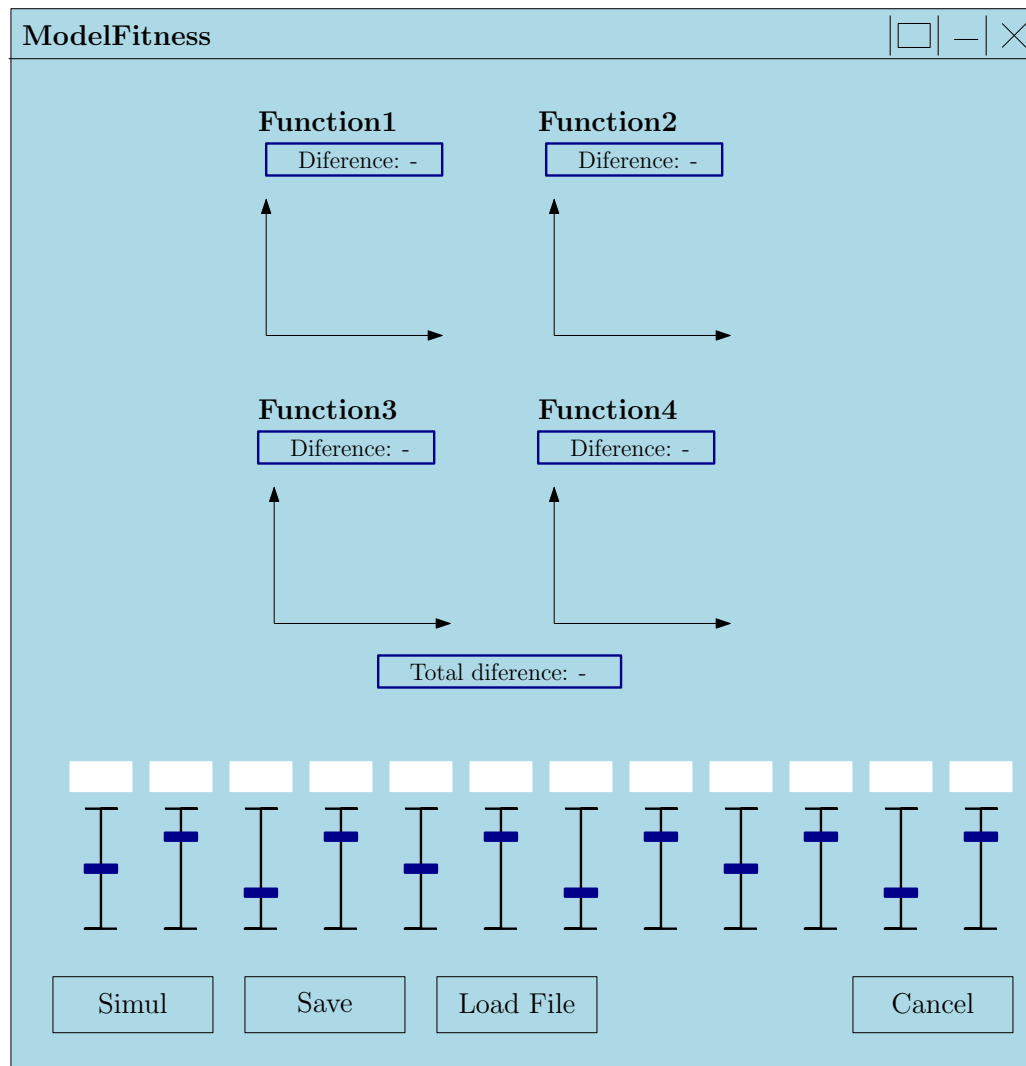


Figura 5.14: Diseño de la interfaz: Refinamiento.



El diagrama de componentes se muestra en la figura 5.15. Como se observa, el modelo es tomado como un componente externo al sistema, ésto significa que podemos agregar tantos modelos como se desee para que el sistema realice la recomendación de parámetros y las simulación desde los mismos.

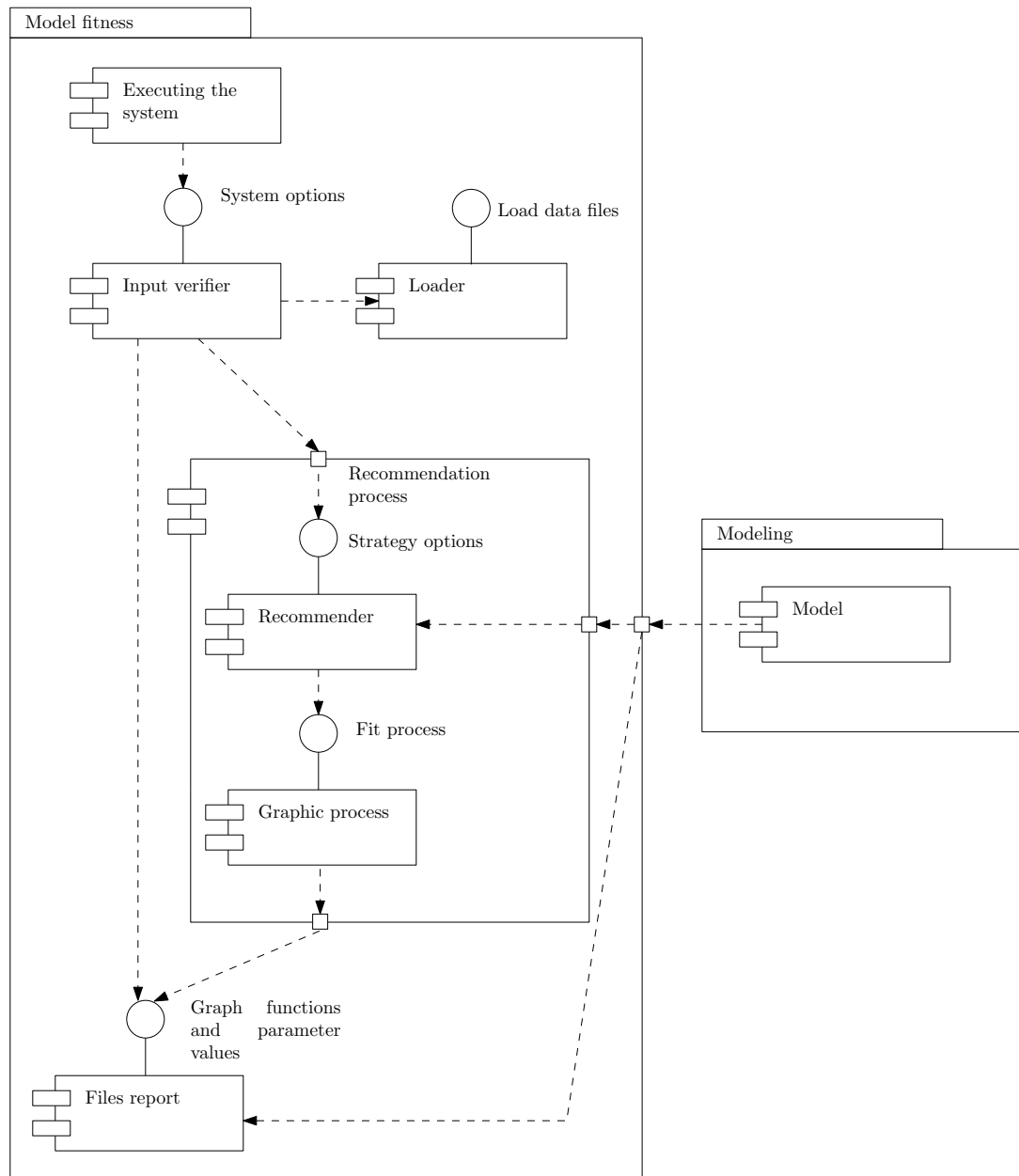


Figura 5.15: Diagrama de componentes del sistema.

Por último, tenemos el diagrama de despliegue representado en la figura 5.16, donde se muestra el sistema montado en un solo computador, y utilizando datos en ese mismo computador. Como se observa, el sistema carga los 6 ficheros de datos empíricos, el modelo y, si el usuario desea, ficheros de reportes cargados anteriormente. Además, los resultados pueden ser guardados en ficheros externos que entrega como salida.

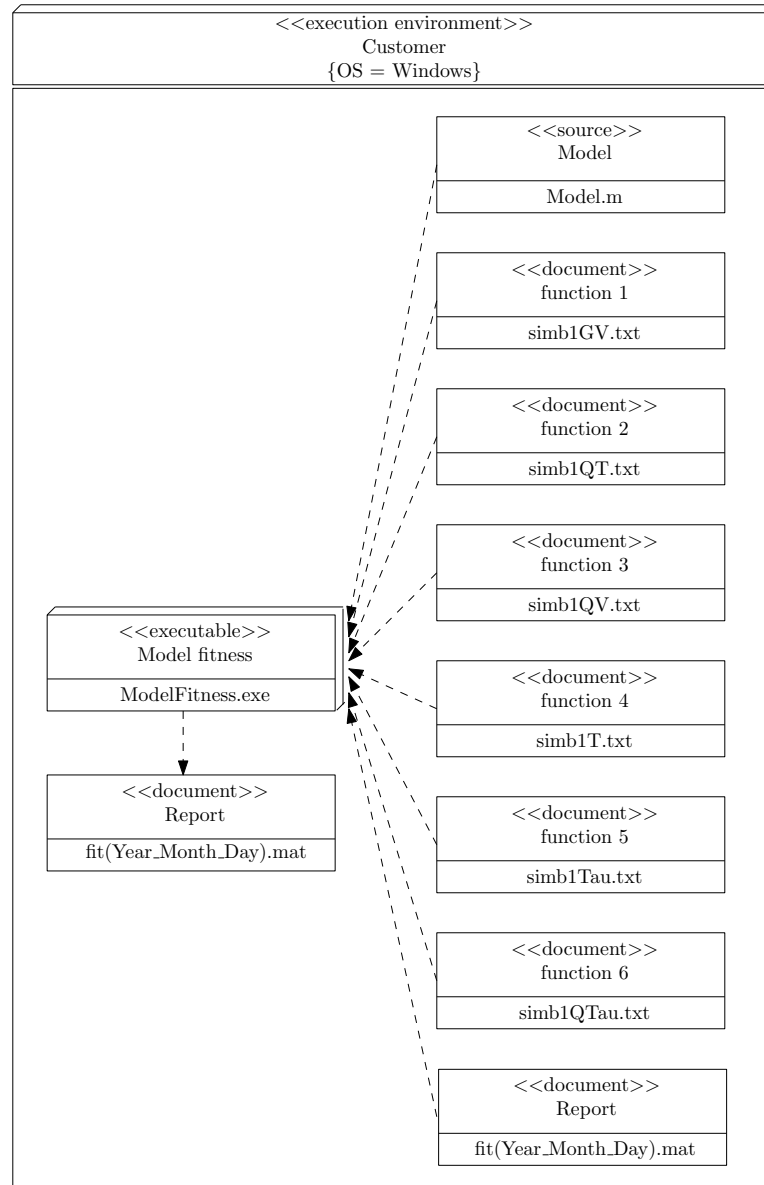


Figura 5.16: Diagrama de despliegue del sistema.

### 5.3. Resultados de la fase de Codificación

Luego del proceso de codificación los resultados se mostrarán a través de las gráficas de la interfaz final, como muestra visual del sistema una vez implementado. En la figura 5.17 se muestra la primera interfaz, donde se permite al usuario la selección de las diferentes opciones: modelo, recomendación o simulación, funciones con sus pesos, los caminos de los ficheros de datos experimentales, Estrategia, y, finalmente, las especificaciones de la estrategia seleccionada.

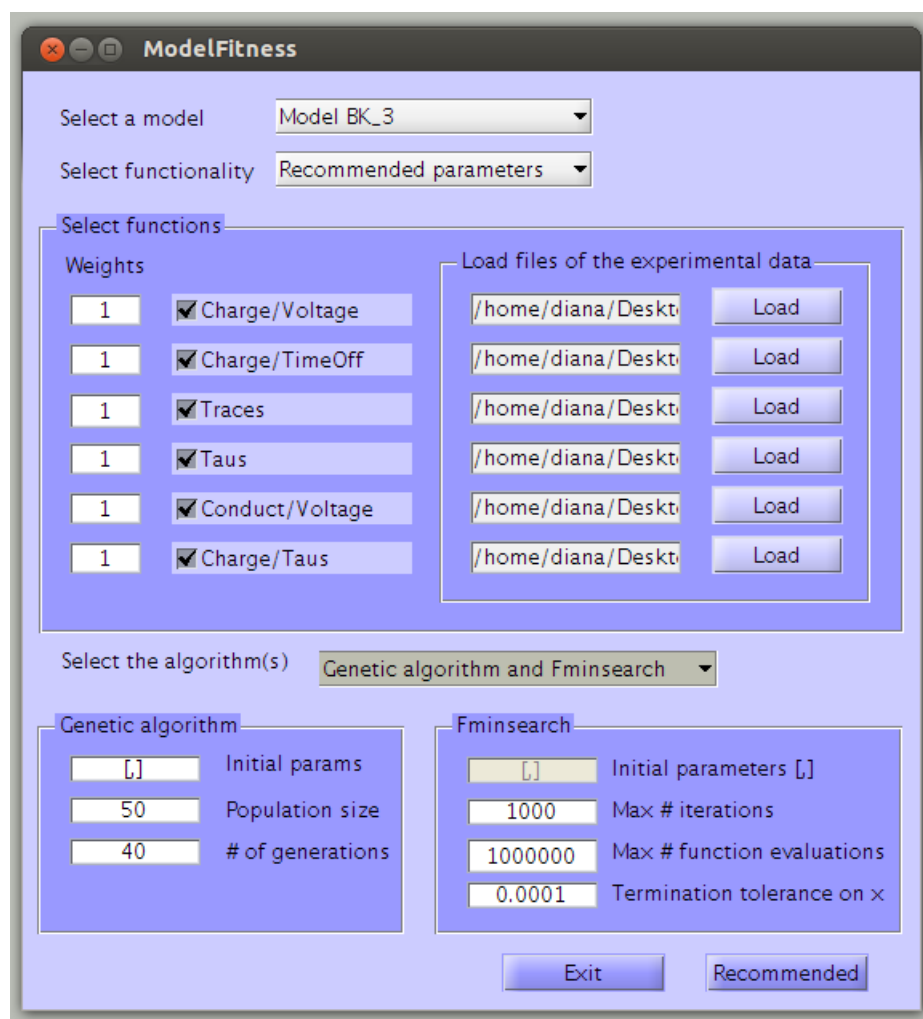


Figura 5.17: Primera interfaz del sistema.

Durante el proceso de Recomendación, se van graficando las distancias que la estrategia va obteniendo. Las distancias del Algoritmo Genético se grafican como se muestra en la figura 5.18, y las distancias del algoritmo Fminsearch (variante del Simplex) se grafican como se muestra en la figura 5.19.

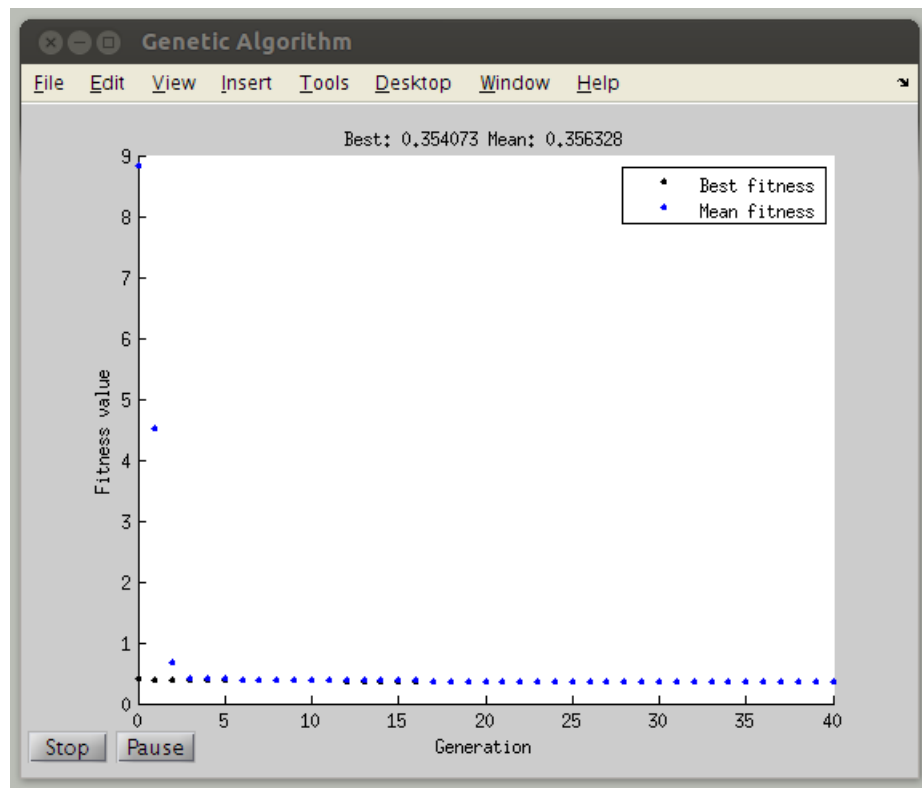


Figura 5.18: Graficación de las distancias que va obteniendo el GA.

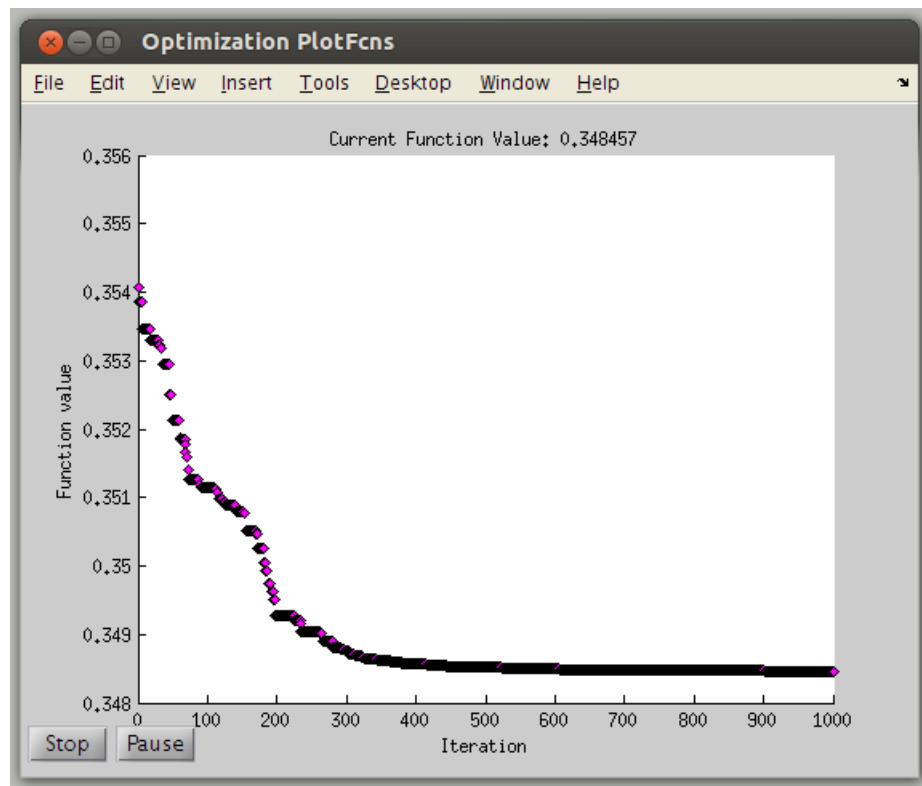


Figura 5.19: Graficación de las distancias que va obteniendo el Fminsearch.

La última interfaz, que se observa en la figura 5.20, muestra la ventana donde se detalla los resultados de la recomendación de parámetros y es a su vez la ventana donde se puede realizar el proceso de refinamiento mediante el cambio manual de los valores y la selección de la simulación. En este proceso, el usuario puede modificar los valores de los parámetros de las barras de la izquierda, y al seleccionar la simulación se grafican los resultados a la derecha de la ventana.

Por último, en el menú Report, el usuario puede seleccionar salvar los resultados actuales de la ventana, o puede cargar un reporte que haya sido salvado anteriormente.

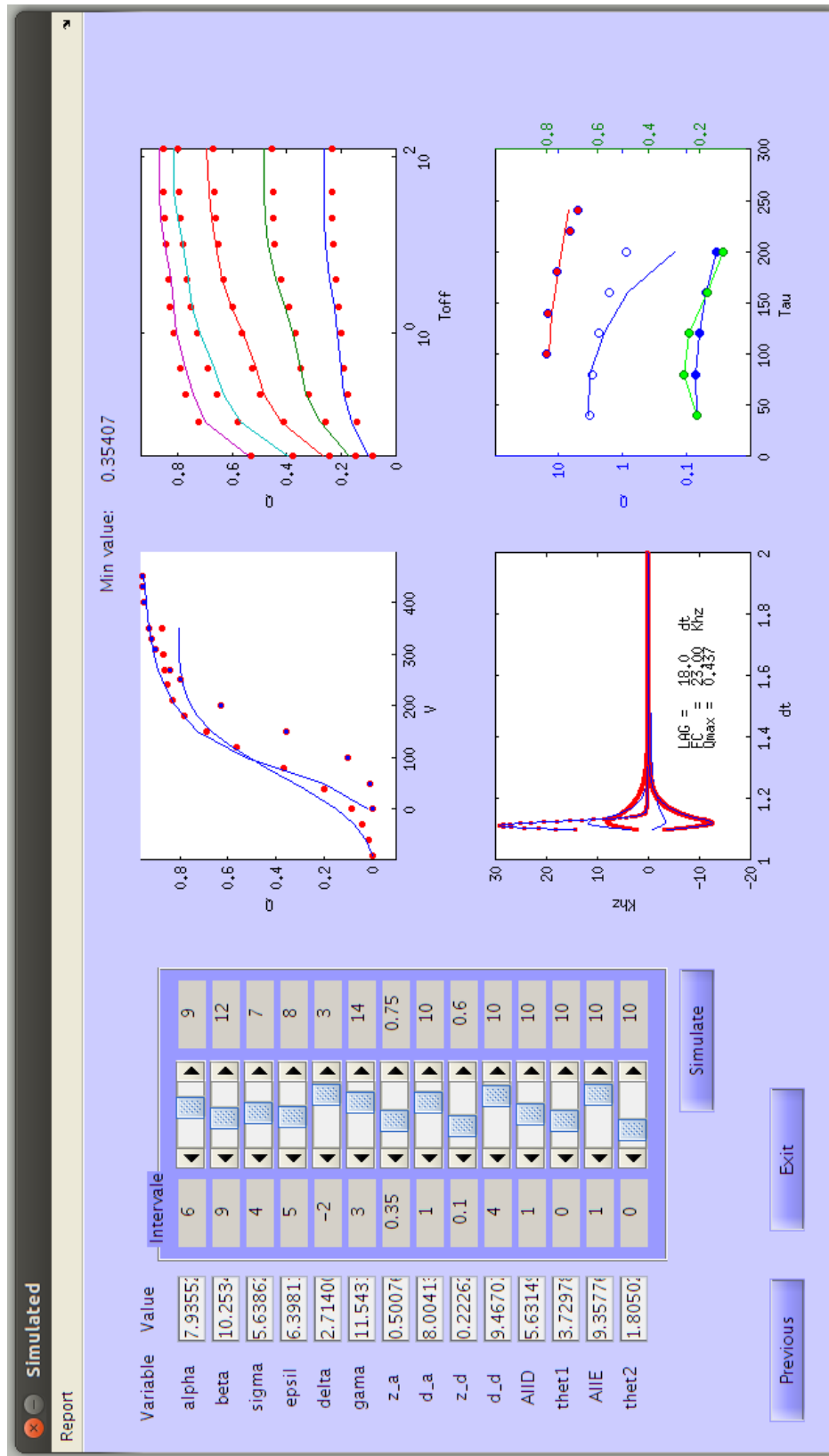


Figura 5.20: Interfaz de simulación, y de proceso de refinamiento.

## 5.4. Resultados de la fase de Pruebas

### 5.4.1. Casos de prueba

Desde los casos de uso que fueron identificados en la fase de Análisis de requisitos se diseñaron los casos de pruebas. Los mismos se muestran con su descripción, los datos que necesitan y los resultados esperados en las tablas 5.9, 5.10, 5.11, 5.12, para el caso de uso de ajustar modelo, el caso de uso de realizar análisis de sensibilidad, el caso de uso de simular y el caso de uso de salvar y cargar reportes, respectivamente.

#	Descripción	Datos de prueba	Resultados esperados
1	Probar el funcionamiento correcto de la validación de los datos de entrada.	Ficheros .txt que contienen los datos experimentales de las 6 funciones fisiológicas y pesos asociados a las mismas.	Mensaje de error en caso de datos con formato incorrecto, de lo contrario, paso a la ventana de opciones de algoritmo.
2	Verificar la validación de los datos de entrada referentes a la estrategia de optimización seleccionada.	Datos especificados en la ventana para la ejecución de la estrategia de optimización.	Mensaje de error en caso de datos con formato incorrecto y ejecución del ajuste en caso de datos correctos.
3	Verificar la muestra gráfica de la ejecución de los algoritmos.		La muestra gráfica de los valores mínimos que van encontrando los algoritmos durante la ejecución.
4	Verificar la correcta culminación y visualización de los resultados del ajuste.		A la culminación de la ejecución en la cantidad de iteraciones solicitadas, deben mostrarse los parámetros recomendados y visualizarse las diferentes funciones.

Cuadro 5.9: Casos de prueba para el CU1 (Ajustar modelo).

#	Descripción	Datos de prueba	Resultados esperados
5	Verificar corrección de la ventana de resultados de parámetros.	Conjunto de valores de los parámetros.	Que los cambios en los datos de texto realicen los respectivos cambios en los datos del <i>slider</i> , y viceversa.
6	Retroceso al inicio.		Que desde esta ventana se pueda regresar al inicio del proceso.
7	Sincronía entre ventanas.	Conjunto de valores de los parámetros.	Que los datos representados en las gráficas en la ventana de resultados de parámetros sean basados en los mismo parámetros.

Cuadro 5.10: Casos de prueba para el CU2 (Realizar análisis de sensibilidad).

#	Descripción	Datos de prueba	Resultados esperados
8	Simulación de los valores fisiológicos.	Conjunto de valores de los parámetros.	Que se ejecute la simulación de los valores fisiológicos desde el modelo una vez que el usuario seleccione la opción simular.
9	Verificar la graficación de resultados.	Conjunto de valores de los parámetros.	Verificar la correcta graficación de las variables fisiológicas en las diferentes funciones, al concluir la estrategia y todas las veces que el usuario desee.
10	Actualización de la ventana.		Que las graficaciones se actualicen en en la misma ventana de las anteriores y no en otra.

Cuadro 5.11: Casos de prueba para el CU3 (Simular).



#	Descripción	Datos de prueba	Resultados esperados
11	Ejecución de guardar reporte.	Resultados de la recomendación y de la simulación.	Que cuando el usuario seleccione la opción de guardar se guarde un reporte con el nombre por defecto o con el indicado por el usuario.
12	Corrección de datos en el reporte.	Resultados de la recomendación y de la simulación.	Que la estructura guardada en el reporte sea la definida en la etapa de diseño (datos en formato establecido).
13	Corrección de la extensión del reporte.		Que el reporte sea salvado con extensión .mat
14	Cargar reporte.	Fichero con extensión .mat que representan los reportes anteriores.	Que el sistema cargue los datos de un reporte en memoria.
15	No sobrescriba.		Que las modificaciones de los cambios de un fichero de reporte cargado no se sobrescriban en el mismo reporte, sino que se guarden como un nuevo fichero, excepto en el caso de que el usuario así lo especifique.

Cuadro 5.12: Casos de prueba para el CU4 (Salvar y cargar reportes).

### 5.4.2. Pruebas estándar

Para la fase de pruebas también fueron verificadas las pruebas estándar, analizando para cada una de ellas si se cumplían en el producto (Estado: OK), si no aplicaba (Estado: NA) o si no se habían tenido en cuenta (Estado: NC). Estas pruebas resultan de interés debido a que son requisitos mínimos que todo sistema debe cumplir.

Resulta importante aclarar en este punto, que la planilla utilizada para el Plan de Pruebas, y las diferentes planillas utilizadas en las especificaciones de las pruebas, el análisis de resultados, las pruebas estándares, y el registro de los defectos; son extraídas de la propuesta del

proyecto chilenos Tutekan, utilizando las mismas como guía para llevar a cabo una buena planificación de la etapa, un buen seguimiento y un buen registro.

En la tabla 5.13 se muestra el chequeo de las pruebas estándares referentes a la interfaz gráfica, en la tabla 5.14 se muestra el chequeo de las pruebas estándares referentes a la validación de los datos de entrada, en la tabla 5.15 se muestra el chequeo de las pruebas estándares referentes a las pruebas funcionales del sistema, y por último en las tablas 5.16 y 5.17 se muestra el chequeo de las pruebas referentes a la verificación de informes y a las impresiones, respectivamente.

ID	1. INTERFAZ GRÁFICA	Estado
1.1	Verificar que se encuentre un ícono en todos los mensajes desplegados por la aplicación.	NA
1.2	Verificar que los botones tengan un ícono o frase adecuado a su función.	OK
1.3	Verificar que las aplicaciones con múltiples pantallas exista la opción retornar a la pantalla anterior.	OK
1.4	Verificar orden y tamaño de los botones.	OK
1.5	Verificar que la etiqueta de los botones no se visualice al perderse el foco.	NA
1.6	Verificar que los botones de ingreso, modificación o eliminación estén habilitados.	NA
1.7	Verificar que el botón cerrar de la ventana, (ubicación superior derecha de la ventana [X]), se encuentre deshabilitado mientras se realiza un proceso.	NA
1.8	Verificar que si un combo-box es sólo de selección, entonces no permita modificar su contenido.	NA
1.9	Verificar la validación del botón Aceptar cuando no se han ingresado los datos.	OK
1.10	Verificar que existe mensaje de Aceptar/Cancelar en botón Salir de la Aplicación.	NA
1.11	Verificar que al minimizar la pantalla, aparezca un ícono estándar del producto.	NC
1.12	Verificar que sólo la Pantalla Principal o Inicial sea minimizable.	NA
1.13	Verificar correcta ortografía en todas las Pantallas.	OK
1.14	Verificar que el TAB esté operando en forma ordenada y secuencialmente bajo un orden lógico de ingreso.	OK
1.15	Verificar que el tamaño de las letras de las pantallas sean uniformes.	OK
1.16	Verificar que la información ingresada sea legible, sin cortes, y que no se exceda el largo del formato de pantalla.	OK

Cuadro 5.13: Pruebas estándares de la interfaz gráfica.

### 5.4.3. Defectos

Los defectos en la etapa de pruebas fueron registrados en planillas de defectos que representan la información que se consideró necesaria para este proyecto. Durante las pruebas se fueron actualizando estas planillas, se fue informando al codificador de los defectos y este fue reparando los mismos desde la línea base en la que siguió trabajando cuando entregó el *software* al *tester*.

ID	2. VALIDACIÓN DE DATOS DE ENTRADA	Estado
2.1	Verificar correcta lectura de la información ingresada.	NC
2.2	Verificar que no se puedan hacer modificaciones en las pantallas de Consultas.	OK
2.3	Verificar la eliminación de la información.	NA
2.4	Verificar formatos de rut, montos, fechas, etc.	NA
2.5	Verificar que la cantidad de caracteres que permita un campo tenga un largo limitado.	OK
2.6	Verificar el ingreso de datos alfanuméricos donde correspondan.	OK
2.7	Verificar el ingreso de datos sólo numéricos donde corresponda.	OK
2.8	Verificar el ingreso sólo de caracteres alfabéticos donde corresponda.	OK
2.9	Verificar validación en el ingreso de los días correspondientes al mes.	NA
2.10	Verificar validación en el ingreso de los meses correspondientes al año.	NA
2.11	Verificar que se esté validando el formato de las fechas.	NA
2.12	Verificar la validación de decimales (con comas).	OK
2.13	Verificar validación de puntos para la separación de miles.	OK
2.14	Verificar el tratamiento de decimales numéricos que utilizan decimales 0.	OK
2.15	Verificar comportamiento del sistema ante cambios en la configuración (si interactúa con otros sistemas).	NA
2.16	Verificar redacción clara y precisa.	OK
2.17	Verificar correcta ortografía.	OK

Cuadro 5.14: Pruebas estándares de la validación de datos de entrada.

ID	3. PRUEBAS FUNCIONALES	Estado
3.1	Verificar que la Aplicación sea compatible con el resto de las Aplicaciones del Ambiente de Pruebas.	OK
3.2	Determinar y validar aspectos técnicos (Tablas, Bases de Datos, perfiles, etc).	NA
3.3	Verificar correcto acceso a sub módulos o Programas.	OK
3.4	Verificar que los botones para grabar, eliminar, modificar, salir, procesar, etc. realmente cumplan su función.	OK
3.5	Detectar tiempos de respuesta frente a un gran número de registros procesados.	OK
3.6	Verificar comportamiento del Sistema o Módulo con gran volumen de datos (pruebas de estrés).	OK
3.7	Validar la existencia de un Manual asociado que permite llevar a cabo el proceso de instalación.	NC
3.8	Verificar el correcto Ingreso y Salida de las Aplicaciones.	OK
3.9	En Pantalla de ingreso de datos, no ingresar un campo y dar Aceptar.	OK
3.10	En Pantalla de ingreso de datos, ingresar un campo y dar Aceptar, verificando la actualización de la formación.	OK
3.11	Borrar el Rut o el Nombre del Cliente y seguir navegando.	NA
3.12	Borrar un dígito al Rut o al Nombre y seguir navegando.	NA
3.13	Ingresar Montos negativos, con el largo máximo permitido y dar aceptar.	NA
3.14	Ingresar en campos puntos, comas y signos ( ¡ · \$ % & ( ) = ? ¿ / * - + ) y dar aceptar.	OK
3.15	VBorrar un dígito al rut o al Nombre y seguir navegando.	NA

Cuadro 5.15: Pruebas estándares referentes a las pruebas funcionales.

ID	4. VERIFICACIÓN DE INFORMES	Estado
4.1	Verificar el correcto formato de los informes generados.	OK
4.2	Verificar que se esté siguiendo una numeración secuencial lógica de acuerdo a un orden establecido.	OK
4.3	Verificar que se realice el proceso de emisión del informe.	OK

Cuadro 5.16: Pruebas estándares referentes a la verificación de informes.

ID	5. IMPRESIONES	Estado
5.1	En listados revisar márgenes, verificando que el ancho del listado no sobrepase la hoja.	NA
5.2	Verificar que el formato de la información que aparezca en las impresiones sea comprensible.	NA
5.3	Verificar que las operaciones de impresión entreguen un aviso que permita Cancelar o Aceptar este mandato.	NA
5.4	Verificar envío de mensajes adecuados cada vez que la impresora este desconectada.	NA
5.5	Verificar correcta lectura y ortografía de la información impresa.	NA

Cuadro 5.17: Pruebas estándares referentes a la verificación de impresiones.

Los datos guardados presentes en las planillas son la fecha en la que se detectó y reportó el defecto, el estado en que se encuentra actualmente (abierto, rechazado, o cerrado) y el tipo de defecto (si era una mejora, una mejora incorporada o un error). Además fue incorporada la información del tipo de error, que son divididos en errores de ambiente, de bases de datos, de códigos, funcionales, de presentación, de configuración o de procedimiento.

Otros datos incorporados en las planillas fueron la severidad (invalidante, grave, media o ligera), la prioridad (inmediata, alta y normal), el caso de prueba al que pertenece, la fecha de resolución, el responsable de la resolución y una pequeña descripción del defecto.

En las tablas 5.18, 5.19, 5.20 y 5.21, se encuentran el listado de defectos encontrados con todos los datos que fueron registrados en referencia a los mismos. Puede notarse que el defecto # 9 se encuentra en estado Abierto. Esto es así debido a que el formato de los ficheros de datos experimentales no ha sido entregado a la empresa desarrolladora por parte del cliente.

#	Fecha	Estado	Tipo de defecto	Tipo de error	Severidad	Prioridad
1	28/10	Cerrado	Error	Funcional	Invalidante	Inmediata
2	28/10	Cerrado	Error	Funcional	Grave	Alta
3	28/10	Cerrado	Error	Presentación	Media	Normal
4	28/10	Cerrado	Error	Presentación	Media	Normal
5	28/10	Cerrado	Error	Presentación	Media	Normal
6	28/10	Cerrado	Error	Presentación	Media	Normal
7	28/10	Cerrado	Error	Presentación	Media	Normal
8	28/10	Cerrado	Error	Presentación	Media	Normal
9	28/10	Abierto	Error	Funcional	Grave	Alta
10	29/10	Cerrado	Error	Funcional	Grave	Alta
11	29/10	Cerrado	Error	Funcional	Invalidante	Inmediata
12	29/10	Cerrado	Error	Funcional	Invalidante	Inmediata
13	29/10	Cerrado	Error	Funcional	Invalidante	Inmediata

Cuadro 5.18: Primera parte de la planilla de defectos identificados.

#	Testeador	CP	Resolución	Responsable	Descripción
1	Diana Biscay	1	29/10	Codificador	Lectura incorrecta del camino de los datos empíricos.
2	Diana Biscay	1	29/10	Codificador	No ejecuta el ajuste cuando se selecciona la opción.
3	Diana Biscay	1	29/10	Codificador	No cuenta con un botón de cancelar la primera ventana.
4	Diana Biscay	1	29/10	Codificador	Los textos explicativos en la ventana de opciones de algoritmo son muy largos.
5	Diana Biscay	1	29/10	Codificador	Agregar botón de regresar en la ventana de opciones de algoritmo.
6	Diana Biscay	1	29/10	Codificador	Ajustar los <i>text</i> en la ventana de parámetros.
7	Diana Biscay	1	29/10	Codificador	La graficación de las funciones de los resultados están muy unidas.
8	Diana Biscay	1	29/10	Codificador	No están incorporadas las variables de funciones en la graficación.
9	Diana Biscay	1	29/10	Codificador	No se está verificando el formato de los datos experimentales.
10	Diana Biscay	2	31/10	Codificador	Pocos decimales en la transformación de los <i>text</i> de los parámetros <i>string</i> .
11	Diana Biscay	2	31/10	Codificador	Visualización incorrecta de la primera función.
12	Diana Biscay	2	31/10	Codificador	No actualización de los datos en el proceso.
13	Diana Biscay	2	31/10	Codificador	Sobregraficación de las funciones incorrecta.

Cuadro 5.19: Segunda parte de la planilla de defectos identificados.



#	Fecha	Estado	Tipo de defecto	Tipo de error	Severidad	Prioridad
14	29/10	Cerrado	Error	Funcional	Invalidante	Inmediata
15	29/10	Cerrado	Error	Funcional	Invalidante	Inmediata
16	29/10	Cerrado	Error	Funcional	Invalidante	Inmediata
16	30/10	Cerrado	Error	Funcional	Grave	Alta
18	30/10	Cerrado	Error	Funcional	Grave	Alta
19	30/10	Cerrado	Error	Presentación	Media	Normal
20	30/10	Cerrado	Mejora	Presentación	Media	Normal
21	31/10	Cerrado	Error	Funcional	Invalidante	Inmediata
22	31/10	Cerrado	Error	Funcionalr	Invalidante	Inmediata
23	31/10	Cerrado	Error	Funcional	Invalidante	Inmediata
24	31/10	Cerrado	Error	Presentación	Grave	Alta

Cuadro 5.20: Tercera parte de la planilla de defectos identificados.

#	Testeador	CP	Resolución	Responsable	Descripción
14	Diana Biscay	2	31/10	Codificador	El <i>text</i> del parámetro 13 no funciona.
15	Diana Biscay	2	31/10	Codificador	El <i>slider</i> del parámetro 2 da valores en rango incorrecto.
16	Diana Biscay	2	31/10	Codificador	Los cambios en los <i>text</i> no se realizan en sintonía con los datos del <i>slider</i> .
17	Diana Biscay	3	2/11	Codificador	Los datos de algunas funciones no se actualizan.
18	Diana Biscay	3	2/11	Codificador	Se cae el sistema cuando lleva varias simulaciones seguidas.
19	Diana Biscay	3	2/11	Codificador	Error ortográfico en el botón Simulate.
20	Diana Biscay	3	2/11	Codificador	La ventana de parámetros está solapada con la de la graficación de funciones.
21	Diana Biscay	4	3/11	Codificador	Reporte guardado con fecha incorrecta.
22	Diana Biscay	4	3/11	Codificador	Sobrescribe el reporte cuando es la misma fecha.
23	Diana Biscay	4	3/11	Codificador	Faltan los datos experimentales en el fichero de reporte.
24	Diana Biscay	4	3/11	Codificador	Luego de cambios en los parámetros no debería dejar guardar sin antes simular

Cuadro 5.21: Cuarta parte de la planilla de defectos identificados.

## 5.5. Cierre del subproyecto informático

### 5.5.1. Balance temporal

Respecto al primer subproyecto (la investigación), debido a que se habían realizado algunas tareas referidas al mismo antes de comenzar el proyecto, se asumirá que las HH estimadas para este subproyecto fueron las reales. Sin embargo, para el segundo subproyecto (el informático) sí fue posible registrar y contabilizar las HH realizadas en las diferentes fases. Esta relación entre las HH estimadas respecto a las HH reales, queda reflejada en la figura 5.21. Como se observa, el proyecto se realizó en 267 HH menos que las estimadas, lo que representa una reducción al 73 % y permitió que el proyecto diera fin con su cierre el 19 de diciembre de 2013.

Actividades	HH e	HH r
Gestión del proyecto	99	80
Análisis	148,5	74
Otras actividades del analista	29,7	29,7
Diseño	178,2	120
Otras actividades del diseñador	29,7	29,7
Codificación	297	200
Otras actividades del programador	29,7	29,7
Pruebas	148,5	130
Otras actividades del tester	29,7	29,7
<b>Total de esfuerzo</b>	<b>990</b>	<b>722,8</b>

Figura 5.21: HH estimadas y HH reales en el subproyecto 2.

### 5.5.2. Balance económico

La reducción en las HH durante el proyecto, se tradujo en reducción en los costos totales del mismo. Como se observa en las figuras 5.22 y 5.23, los costos de mano de obra fueron de 3.605.838,3 pesos chilenos, y el costo total resultó entonces en 3.635.838,3 pesos chilenos, respectivamente.

Rol	C/H	HHr	CTe	CTe
Jefe de proyecto (Ingeniero)	5.584	90	597.488	502.560
Analista (Ingeniero)	5.584	263,7	1.843.837	1.472.500,8
Diseñador (Técnico)	3.025	149,7	628.897,5	452.842,5
Programador (Técnico)	3.025	229,7	988.267,5	694.842,5
Tester (Técnico)	3.025	159,7	539.055	483.092,5
<b>Costo total de la mano de obra</b>	<b>20.243</b>	<b>892,8</b>	<b>4.597.545</b>	<b>3.605.838,3</b>

Figura 5.22: Costo estimado y costo real de la mano de obra.

Recurso	Costo e	Costo r
Mano de obra	4.597.545	3.605.838,3
Software (licencias)	0	0
Hardware (equipos)	0	0
Otras (libros, impresiones, ect.)	30.000	30.000
<b>Costo total del proyecto</b>	<b>4.897.545</b>	<b>3.635.838,3</b>

Figura 5.23: Costo total estimado y costo total real.

## Capítulo 6

### Conclusiones

El proyecto, en la consecución de los objetivos planteados, se dividió en dos subproyectos, un subproyecto de investigación seguido de un subproyecto informático, que concluyó con un *software* que resuelve el problema planteado por el cliente: pues podrán determinar, durante un tiempo de aproximadamente dos días, si existe un conjunto de valores de los parámetros del modelo que permita explicar el conjunto de datos empíricos con los que cuentan y con los que puedan contar a futuro.

El objetivo de la pre-investigación fue encontrar una estrategia adecuada para la resolución de un problema de optimización que nace para la explicación de unos datos empíricos desde un modelo biológico propuesto sobre el funcionamiento de los canales de iones de  $K^+$ . Investigación que se ha iniciado desde un enfoque cuantitativo.

Para el problema de optimización en estudio, debido a las características que presenta, se propuso una estrategia de resolución que consiste en la unión de dos algoritmos de optimización que han resultado tener marcadas ventajas en diferentes problemas. La estrategia consiste en:

- a) Comenzar con un **Algoritmo Genético**, para una búsqueda global.
- b) Continuar con un algoritmo iterativo, específicamente el **Fminsearch**, para una búsqueda local y convergencia rápida.

Las variables de estudio son los valores de los parámetros que definen los valores de las funciones simuladas y los datos empíricos, ambos conjuntos como variables independientes. Por otro lado, tenemos los errores mínimos encontrados para esos datos que devuelve la estrategia propuesta, como variables dependientes. Por lo que consiste en una metodología correlacional entre variables dependientes e independientes.

Los resultados iniciales parecen indicar la adecuación de la estrategia para la función en estudio, dando resultados de errores en la vecindad esperada según la hipótesis planteada. Sin embargo, esta no es una investigación terminada, debido a que faltan las fases finales definidas en la investigación cuantitativa, como son la recogida de datos, sistematización de los datos, análisis estadístico, comprobación de hipótesis y resultados.

Los resultados hasta aquí fueron utilizados en el subproyecto informático que se definió como paso siguiente en el proyecto. No obstante, es de interés del equipo, una vez concluido el

proyecto, completar las fases de la investigación y obtener una base estadística de la validez de la estrategia propuesta para el problema en cuestión.

Por último, referido al subproyecto de investigación, mencionar que parte del mismo estaba realizado antes de comenzar el proyecto que nos ocupa. En este sentido, referente a la planificación estimada, no se cuenta con horas reales medidas con las cuales comparar. Por lo anterior, para este subproyecto, asumiremos que la estimación se acerca a los valores reales con los cuales no se cuenta.

En el subproyecto informático se llevaron a cabo las fases de una metodología Cascada (Análisis de requisitos, Diseño de *software*, Codificación y Pruebas). En la fase de Análisis se registraron y sistematizaron los requisitos y se identificaron, modelaron y describieron los casos de uso; además se verificó la validez mediante el análisis de la trazabilidad entre requisitos y casos de uso.

En la fase de Diseño, se diseñaron las diferentes vistas del modelo de Philippe Kruchten, mediante diagramas con el estándar UML 2.2. Desde el diagrama de casos de uso y los requisitos identificados en la fase anterior, se modeló el diagrama de actividades, desde éste, el diagrama de clases, se realizó el diseño de las estructuras de datos, de los ficheros de reporte, y un diseño inicial de las interfaces. Además, se modeló el diagrama de componentes y el diagrama de despliegue.

La codificación se realizó en lenguaje M con ambiente Matlab, y se realizaron las pruebas unitarias y las pruebas de integración. Una vez concluida la codificación se dió inicio a la fase de Pruebas donde se realizaron las pruebas diseñadas como casos de pruebas desde los casos de uso (pruebas de sistema), se verificaron los estándares y se llevó un registro de todos los defectos encontrados, los que se fueron resolviendo por el codificador.

Terminadas las pruebas, se realizó el Cierre del Proyecto, que culminó con las pruebas de aceptación del cliente, la encuesta de validación y el informe de estado. El informe de estado dió al proyecto como concluido, sin tareas inconclusas ni pendientes, siendo la planificación temporal y monetaria ligeramente por debajo de lo estimado.

En efecto, con una estrategia adecuada para el problema de optimización de la recomendación de parámetros, y una interfaz que de manera semiautomática realiza el proceso, los investigadores del departamento del CINV podrán determinar si la distancia mínima que han obtenido hasta el momento en su proceso manual puede ser mejorada, y si existe un conjunto

de valores de los parámetros del modelo que permita explicar el conjunto de datos empíricos.

## 6.1. Lecciones aprendidas

La principal lección aprendida fue referente a la planificación en cuanto a HH para las diferentes fases del proyecto. Una vez realizada la planificación inicial para el subproyecto informático, la fase de Análisis de requisitos se realizó en la mitad del tiempo que se había estimado. Esta planificación, alejada significativamente de lo real, está relacionada con uno de los riesgos identificados inicialmente, a saber, la escasa experiencia del equipo desarrollador en proyectos anteriores.

Otro problema que surgió, y relacionado también con el riesgo mencionado, fue la planificación de las horas destinadas diariamente al proyecto. Se habían estimado 5 horas diarias para el mismo, pero en la práctica se dedicaron entre 3 y 4 horas diarias. Lo anterior llevó a que, a pesar de lograr realizar las fases con menos HH de las estimadas, el tiempo de culminación de cada una fue cercano al esperado, debido a que fueron menos HH por fase, pero también menos HH dedicadas diariamente.

Por otro lado, sucedió que no se tenían claro los pasos y tareas a seguir en algunas fases, y el cómo hacerlas según las características del proyecto. Lo anterior llevó a un estudio intenso de una amplia bibliografía, por lo que se convirtió en un factor positivo en cuanto al incremento de conocimientos por parte del equipo de proyecto.

Independiente de los contratiempos, el proyecto fue concluido antes de lo planificado, se realizaron todas las etapas, el *software* final fue correctamente probado y se cumplieron todos los requisitos del cliente. Lo más importante es que tanto el objetivo general como los objetivos específicos se cumplieron, y todas las actividades desarrolladas se convirtieron en un fructífero proceso de aprendizaje.



# Bibliografía

- Alonso, Á., F. E. C. M. G. A. L. C. G. J. (2009). *Fundamentos de Psicobiología*. Sanz y Torres, España.
- CINV (2013). *Centro Interdisciplinario de Neurociencia de Valparaíso*. Disponible en: <http://cinv.uv.cl/>.
- Contreras, G.F., N. A. A. O. G. C. and R., L. (2012). *Modulation of BK channel voltage gating by different auxiliary  $\beta$  subunits*. Proceedings of the National Academy of Sciences of the United States of America.
- Coro, F. (1996). *Fisiología Celular: Un enfoque biofísico*. Benemérita Universidad Autónoma de Puebla, México.
- Educación (2013). *Ministerio de Educación: Empleabilidad e Ingresos*. Disponible en: <http://www.mifuturo.cl/index.php/futuro-laboral/buscador-por-carrera-d-institucion>.
- Fontela, C. (2011). *UML: modelado de software para profesionales*. Alfaomega, México.
- Gotshall, S. and Rylander, B. (2000). *Optimal Population Size And The Genetic Algorithm*. Proc On Genetic And Evolutionary Computation Conference.
- Hernández, R., F. C. B. P. (2010). *Metodología de la Investigación*. McGraw-Hill, Perú.
- Hille, B. (2001). *Ion Channels of Excitable Membranes*. SINAUER, Estados Unidos.
- Holtschulte, N. (2005). *Optimal Population Size in Island Model Genetic Algorithms*.
- ISO-9126, N. (2011). *Norma ISO-9126 para análisis de software*. Disponible en: <http://bemuserp.blogspot.com/2011/09/norma-iso-9126-para-analisis-de.html>.

- Kruchten, P. (1995). *Architectural Blueprints- The 4+1 View Model of Software Architecture*. IEEE Software 12(6).
- Larman, C. (2004). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. Prentice Hall PTR.
- Lobo, F., L. C. (2007). *Adaptive Population Sizing Schemes in Genetic Algorithms*. Springer-Verlag, Berlin Heidelberg.
- McConnell, S. (1997). *Desarrollo y Gestión de Proyectos Informáticos*. McGraw-Hill, España.
- Menchada, A. (2008). *Algoritmo híbrido para resolver problemas de optimización con restricciones*. Tesis que presenta para obtener el Grado de Maestra en Ciencias en la Especialidad de Computación, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México, DF.
- Nocedal, J. y Wright, S. (2006). *Numerical Optimization*. Springer, Estados Unidos.
- Pressman, R. S. (2001). *Ingeniería del Software: Un enfoque práctico*. McGraw-Hill, España.
- Purves, D., A. G. F. D. H. W. L. A. M. J. y. W. S. (2004). *Neurociencia*. Editorial Médica Panamericana.
- Randall, D.; Burggren, W. y. F. K. (2002). *Fisiología Animal*. McGraw-Hill.
- Remón, C., T. P. (2010). *Análisis de Estimación de Esfuerzo aplicando Puntos de Caso de Uso*.
- Schmuller, J. (2005). *Aprendiendo UML en 24 horas*. Prentice Hall.
- Sommerville, I. (2005). *Ingeniería del Software*. Pearson, Madrid.
- Tutelkan (2009). *Documentación del proyecto*. Disponible en: <http://www.tutelkan.info/?area=docs>.
- Valdez, S.; Botello, S. y. H. A. (2005). *Optimización multi-objetivo de estructuras, utilizando algoritmos de estimación de distribuciones e información de vecindades*. Comunicación Técnica No I-05-07/09-06-2005, (CC/CIMAT).

Wikipedia (2013). *ISO/IEC 9126*. Disponible en:  
[http://es.wikipedia.org/wiki/ISO/IEC\\_9126](http://es.wikipedia.org/wiki/ISO/IEC_9126).

Zapatero, M<sup>a</sup> J., A. J. P. J. (2005). *Análisis de algunas metaheurísticas creadas a partir de Optimización Gravitatoria*. XIX Jornadas ASEPUMA – VII Encuentro Internacional, Anales de ASEPUMA n° 19: 0902.

# Apéndice

# Apéndice A

## Transporte de electrolitos mediante canales de iones

Todas las células están rodeadas de una membrana, denominada **citoplasmática**, que la separa del medio extracelular y está compuesta por lípidos<sup>1</sup>, proteínas<sup>2</sup> y, generalmente, pequeñas cantidades de carbohidratos<sup>3</sup>.

Los lípidos de la membrana forman una bicapa con “*propiedades de autosellado*” (Coro, 1996) con la que las proteínas interactúan, y las proteínas se encuentran insertadas en la bicapa de manera que pueden atravesarla una o múltiples veces (llamadas **integrales** o **intrínsecas**) o bien no atravesarlas y estar unidas de manera débil a la misma o a las proteínas integrales (llamadas **periféricas** o **extrínsecas**), como se muestra en la Figura A.1.

La membrana citoplasmática desempeña diversas funciones mediante el transporte de sustancias entre el interior y exterior celular, determinando así la composición química en ambos lados. De esta manera, constituye un filtro selectivo que establece el **gradiente de concentración** ( $grad\ c_i$ ), el cual está determinado por la concentración de soluto a cada lado de la membrana.

---

<sup>1</sup>Conjunto de moléculas hidrófobas (insolubles en agua) con funciones en los organismos vivos de reserva energética, estructurales y la reguladoras.

<sup>2</sup>Moléculas imprescindibles para el crecimiento del organismo, con funciones estructurales, reguladoras, inmunológicas (anticuerpos), etc.

<sup>3</sup>Moléculas con funciones en los organismos vivos de reserva energética y estructurales.

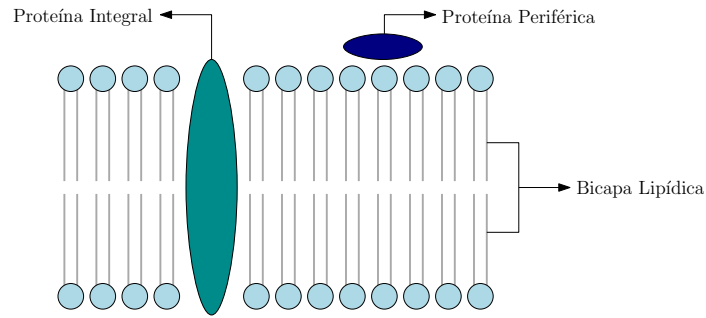


Figura A.1: Membrana citoplasmática.

El transporte de electrolitos<sup>4</sup> a ambos lados de la membrana lo realizan proteínas integrales que forman las **bombas de iones** y los **canales de iones**. Así, debido a la presencia de cargas que presentan los iones, la membrana citoplasmática, además del  $grad\ c_i$ , establece el **potencial eléctrico** ( $grad\ E$ ) que está determinado por la cantidad de cargas positivas y negativas a cada lado de la membrana.

Las bombas de iones tienen la capacidad de mover los iones a través de la membrana en contra de su  $grad\ c_i$ , con la función de crear y mantener gradientes iónicos. Los canales de iones, por su parte, tienen la función de originar cambios en la **permeabilidad** selectiva de la membrana a los diferentes iones. Esta permeabilidad está determinada por el número de canales abiertos que permiten el paso de ese ion a través de ellos.

Los canales iónicos están constituidos por proteínas integrales de membrana que forman poros hidrofílicos llenos de agua a través de los cuales fluyen los iones cuando el canal está abierto. Estos canales pueden tener poca selectividad para los diferentes iones, ser selectivos para aniones<sup>5</sup> y cationes<sup>6</sup>, o ser selectivos para un tipo particular de ion ( $Na^+$ ,  $K^+$ ,  $Ca^{++}$ , o  $Cl^-$ ).

En los no-electrolitos actúa una **fuerza difusional** que está determinada por el  $grad\ c_i$ ; fuerza que impulsa un movimiento del soluto de mayor concentración a menor concentración a través de la membrana. En el caso de los electrolitos, además de la fuerza difusional, actúa una **fuerza eléctrica** determinada por el  $grad\ E$ , la cual ejerce una fuerza de repulsión entre iones con la misma carga eléctrica y una fuerza de atracción entre iones con cargas de distinto signo. Por tanto, esta fuerza eléctrica depende de la carga del ion y del **potencial de membrana** ( $E_m$ ). El  $E_m$  está determinado por la distribución de iones, en cuanto a cantidad

<sup>4</sup>Sustancia que contiene iones libres.

<sup>5</sup>Iones con carga eléctrica negativa.

<sup>6</sup>Iones con carga eléctrica positiva.

de carga positivas y negativas, a ambos lados de la membrana, y representa la carga eléctrica o voltaje que se genera ( $V_m$ ).

Como el movimiento de los iones a través de la membrana se ve afectado por ambas fuerzas (eléctrica y química), se dice que depende del **gradiente electroquímico**. No obstante, en este movimiento no sólo interviene dicho gradiente, también depende de la permeabilidad de la membrana a los diferentes iones que, como se mencionó anteriormente, está regulada por los canales iónicos abiertos que son selectivos a ese ion.

Los canales de iones se encuentran tanto en **células excitables**<sup>7</sup> como no excitables, y presentan un conjunto de características generales descritas por Coro en 1996, de las cuales sólo nos detendremos en las que tienen una relación estrecha con el presente trabajo:

- **Conductancia iónica** ( $g$ ), es una medida de la facilidad con la que los iones se mueven a través del canal. Se mide en pS ( $= 10^{-12}$  S).
- **Selectividad iónica**, indica la naturaleza de los iones que pueden fluir a través del canal abierto.
- **Mecanismos de compuerta** (*gating*, en inglés), muestran los procesos de apertura y cierre del canal bajo la influencia de factores tales como el voltaje de membrana, presencia de sustancias químicas y cambios mecánicos.

### A.0.1. Células excitables

Las células excitables son células que pueden detectar pequeñas variaciones de energía del medio que las rodea; es el caso de las neuronas, las células receptoras sensoriales y las células musculares. Las membranas de estas células responden a variaciones energéticas eléctricas, químicas o mecánicas, lo que ha llevado a clasificarlas en células **electroexcitables**, **químioexcitables** y **mecanoexcitables**.

En condiciones de no-estimulación se mantiene una razón de concentraciones a cada lado de la membrana y un valor de potencial de membrana en reposo ( $PMR$ ) de manera estacionaria. Al estimular las membranas excitables, los canales de iones de la célula son activados aumentando su probabilidad de apertura. Luego, cada canal al abrirse provoca que se muevan los iones a través de la membrana a favor de su gradiente electroquímico, lo que produce

---

<sup>7</sup>Células que responden a variaciones de energía del medio.

**corrientes iónicas** ( $I$ ) a través del mismo. Así, estos canales se relacionan con respuestas rápidas de la células, alterando de manera inmediata y por breve tiempo la permeabilidad iónica de la membrana.

Las **membranas electroexcitables** presentan canales de iones **dependientes del voltaje**, lo que significa que su probabilidad de cierre y apertura depende del  $E_m$ . Por tanto, la permeabilidad de la membrana queda regulada por cambios en este potencial. Estos canales dependientes del voltaje contienen sensores con carga positiva que se mueven en el campo eléctrico de la membrana durante los cambios de potencial, generando un movimiento de cargas llamado **corriente de compuerta** ( $I_g$ )<sup>8</sup>.

Pero, ¿qué realiza la función de sensor en los canales? La estructura de los canales de sodio ( $Na^+$ ) y calcio ( $Ca^{++}$ ) dependientes del voltaje es de cuatro dominios protéicos homólogos (subunidad  $\alpha$ ) que rodean a un poro iónico central. Para los canales de potasio ( $K^+$ ), la estructura es de cuatro subunidades  $\alpha$ , cada una de las cuales está compuesta por un dominio proteico. En ambos casos, cada uno de estos dominios contiene seis segmentos de transmembrana con estructura en  $\alpha$ -hélice, donde el segmento  $S_4$  actúa como sensor de voltaje.

---

<sup>8</sup>Corriente en los sensores de voltaje de los dominios proteicos.



## Apéndice B

### Mecanismos de compuerta: Modelo $C-C-O$

Estudios anteriores con la técnica de *patch-clamp* han descrito las transiciones entre estados **abiertos** ( $O$ , hay corriente iónica) y **cerrados** ( $C$ , no hay corriente) de los canales iónicos con un carácter probabilístico; es decir, la activación del canal significa que aumenta la probabilidad de que éste se abra, y no que el canal se mantenga abierto durante todo el estímulo. De esta manera, los canales se han considerado estructuras dinámicas con cambios en las probabilidades de transiciones entre los estados  $O$  y  $C$ , por tanto, cambios que son los responsables de las variaciones de la conductancia iónica ( $g$ ).

Para los BK en ausencia de  $Ca^{++}$  se ha propuesto que cada uno de los cuatro sensores de voltaje (subunidad  $\alpha$ ) pueden estar en forma activa o en reposo de manera independiente. El conjunto de los sensores y la forma en la que pueden estar determinan los diferentes estados, donde el cambio de uno a otro ocurre con cierta probabilidad.

Basados en lo anterior y en sus recientes datos empíricos obtenidos en laboratorio, Contreras y cols propusieron en el 2012 un modelo del funcionamiento de los mecanismos de compuerta denominado modelo  $C-C-O$ . Como se observa en la Figura B.1 (a), cada fila representa las diferentes combinaciones de los cuatro sensores de voltaje (subunidades  $\alpha$ ), y cada columna describe un canal que se abre en dos pasos: activación del sensor de voltaje, paso de  $C_0$  a  $C_1$ , y apertura del canal, paso de  $C_1$  a  $O$ .

Como se observa en la primera fila de la Figura B.1 (a), si se asume  $\alpha$  como la probabilidad

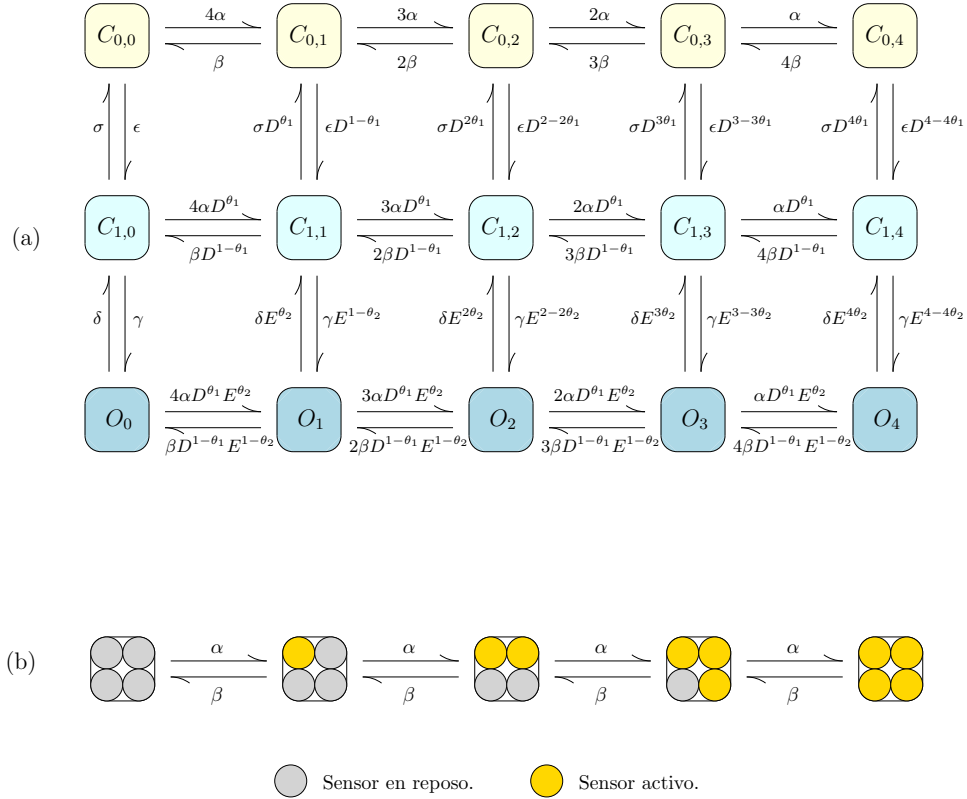


Figura B.1: Modelo de tres niveles. (a) Los 15 estados con sus probabilidades de transiciones entre ellos. (b) Reposo y activación de los 4 sensores que forman el canal en un recorrido entre estados cerrados del primer nivel.

de que se active uno de los sensores, el pasar de un estado donde todos los sensores están en reposo a un estado donde hay uno activo sería  $4\alpha$ . De igual manera, pasar de un estado de un solo sensor activo y tres en reposo a uno con dos activos y dos en reposo sería  $3\alpha$ , ya que puede ser activado cualquiera de los que se encuentra en reposo. Así, luego se transitaría a un estado de 3 activos con una probabilidad de  $2\alpha$  y a uno con todos activos con una probabilidad  $\alpha$ . Siguiendo la misma idea, las transiciones inversas quedarían determinadas por otras probabilidades, denominadas  $\beta$ , desde  $4\beta$  hasta  $\beta$ . A modo de ejemplo, en la Figura B.1 (b) se muestra un posible recorrido entre estados cerrados de la primera fila.

Así, este modelo consta en total de 15 estados, donde sus probabilidades de transiciones están determinadas por 14 parámetros: *alpha* ( $\alpha$ ), *beta* ( $\beta$ ), *sigma* ( $\sigma$ ), *epsilon* ( $\epsilon$ ), *delta* ( $\delta$ ), *gamma* ( $\gamma$ ), *z\_alpha*, *d\_alpha*, *z\_delta*, *d\_delta*, *AllD*, *theta1* ( $\theta_1$ ), *AllE* y *theta2*.

## Apéndice C

### Recorrido de variables, no convexidad

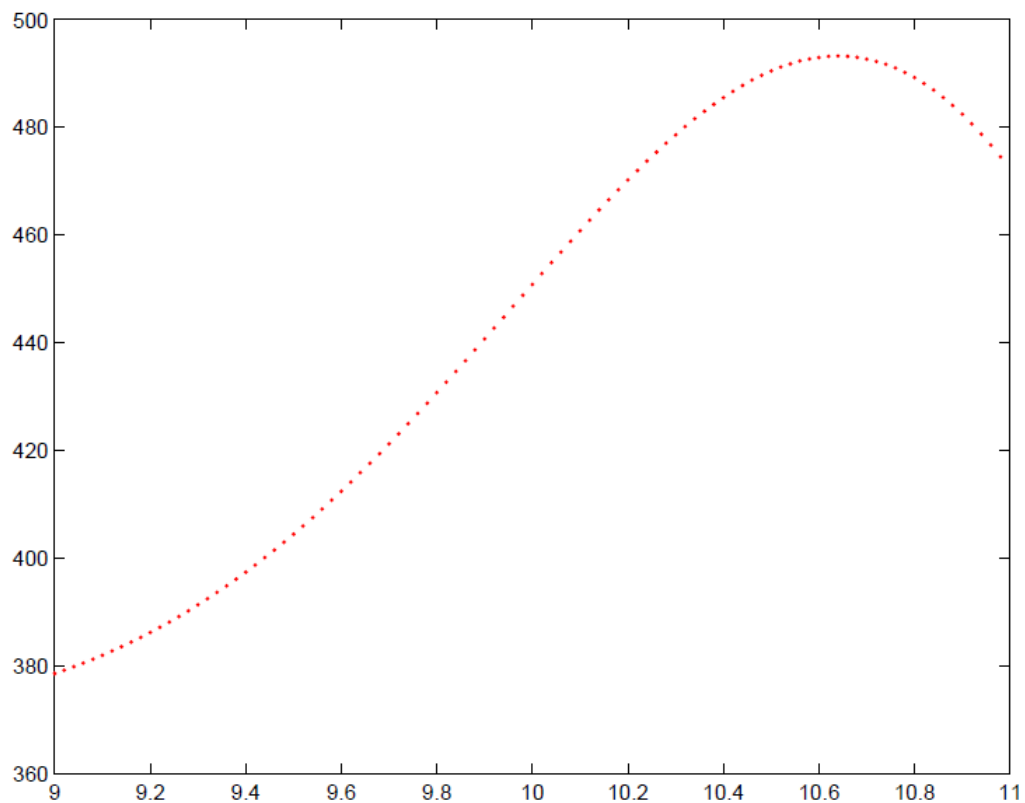


Figura C.1: Recorrido de la variable  $\beta$ .

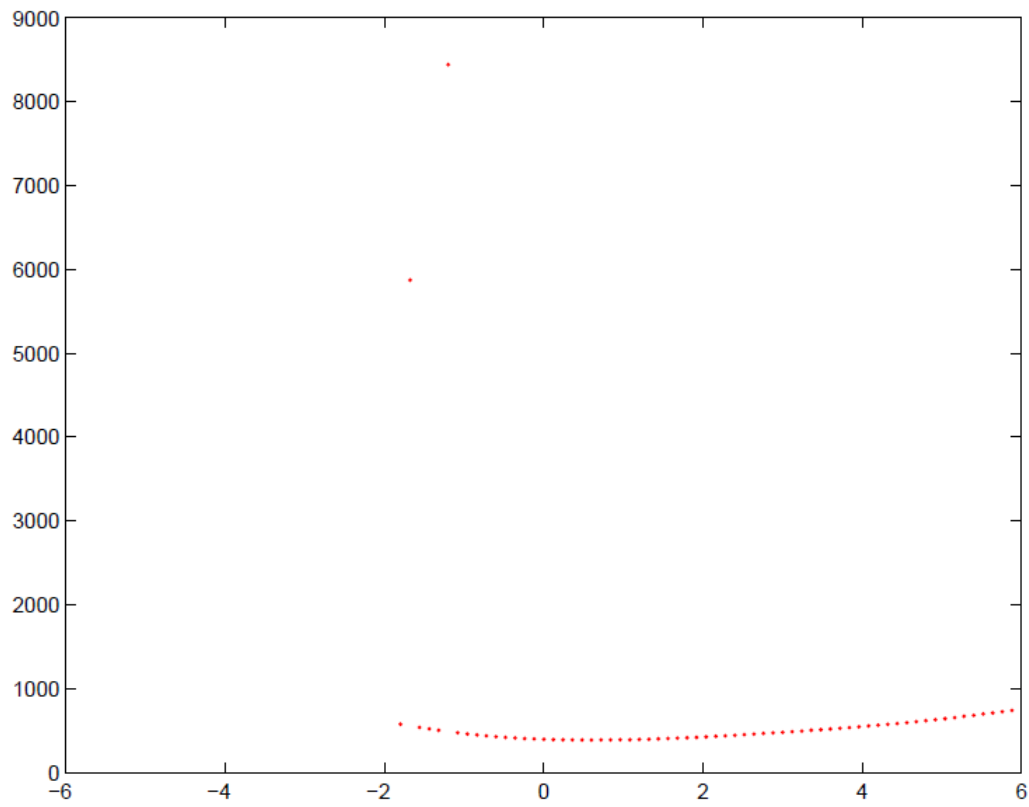


Figura C.2: Recorrido de la variable  $\sigma$ .

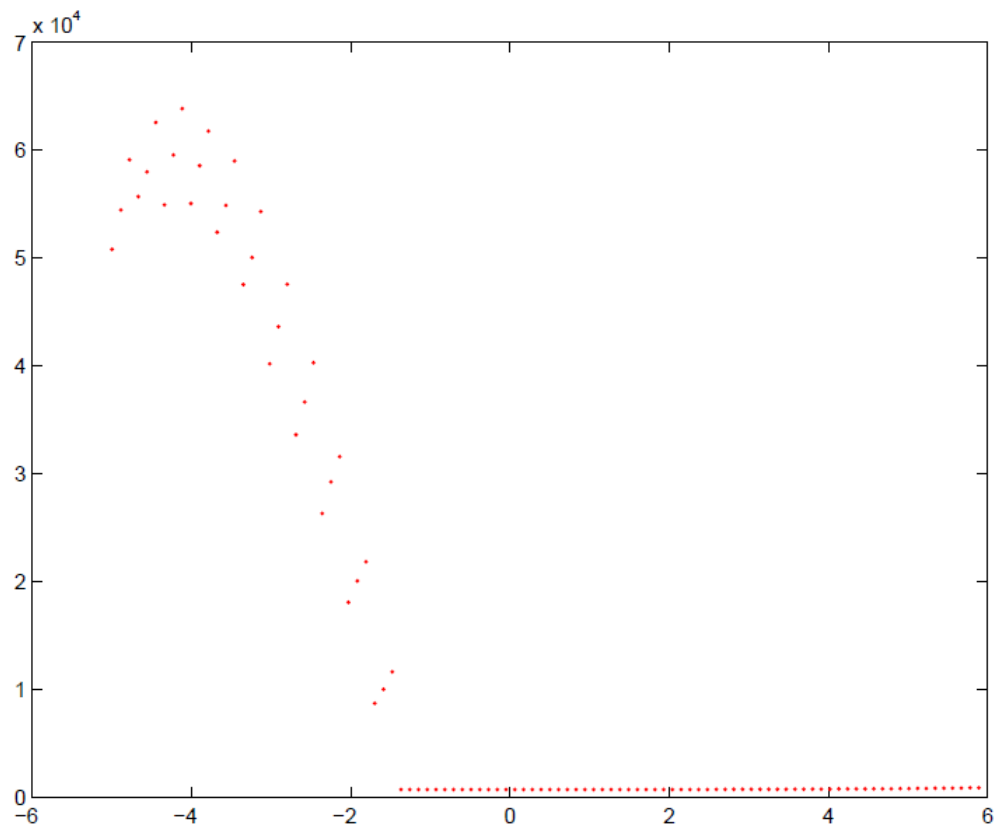


Figura C.3: Recorrido de la variable  $\delta$ .

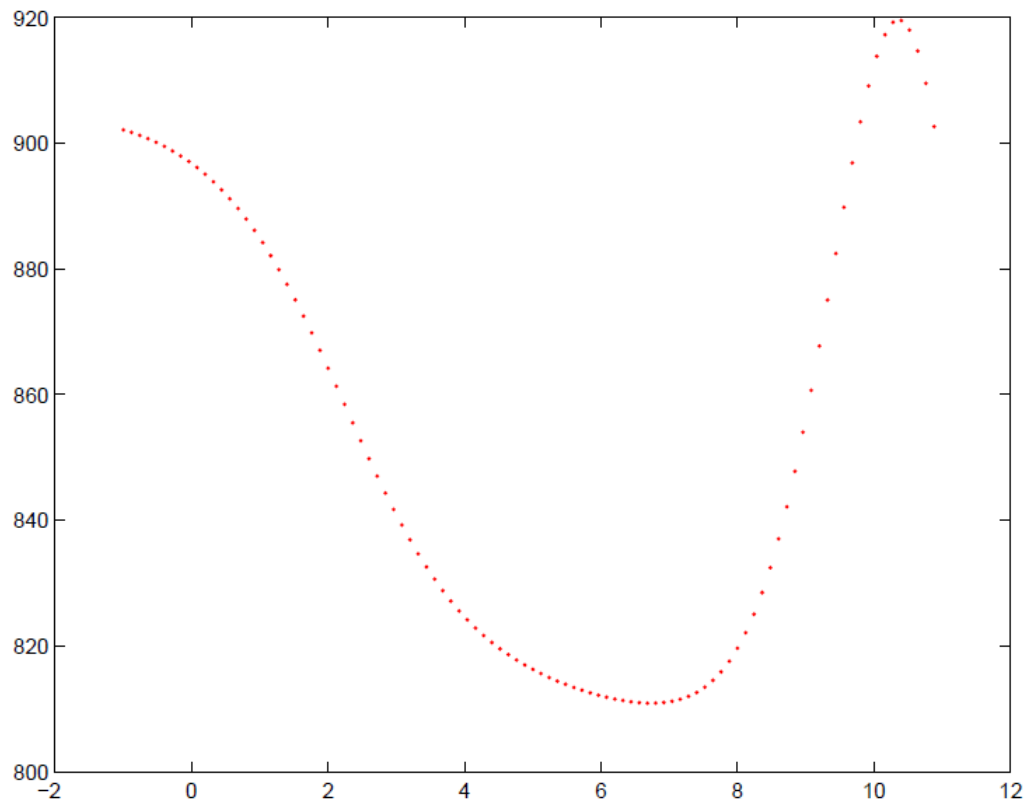


Figura C.4: Recorrido de la variable  $\gamma$ .

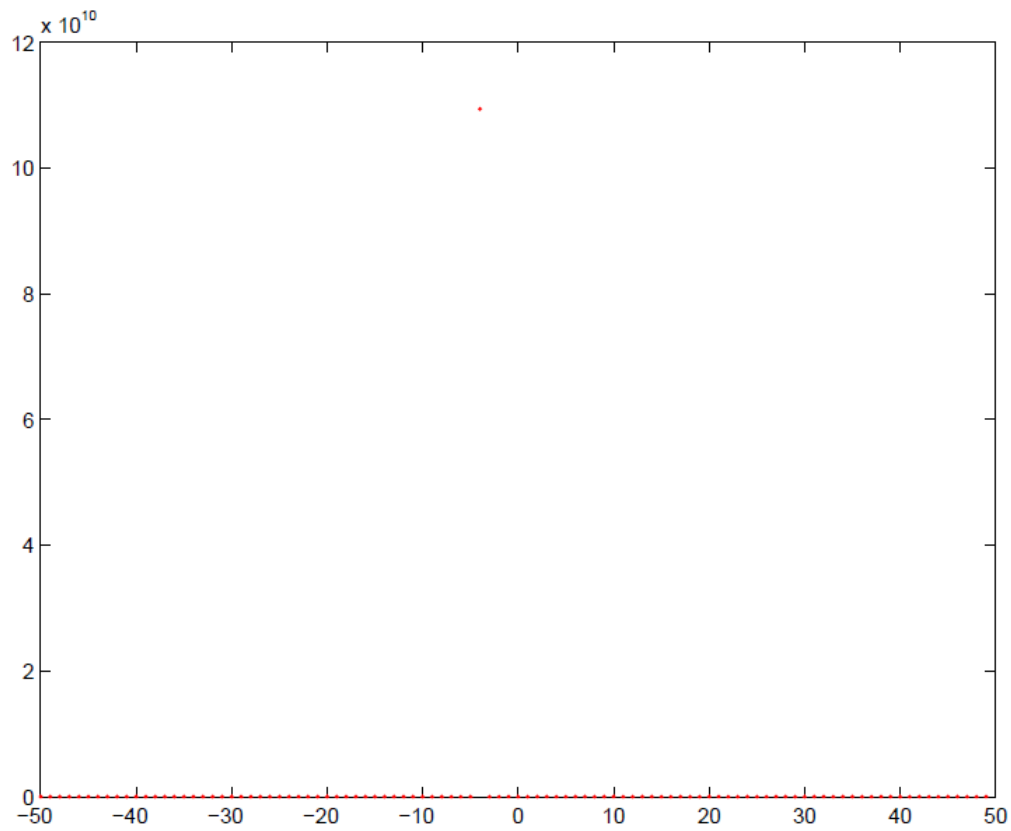


Figura C.5: Recorrido de la variable  $d.\alpha$ .

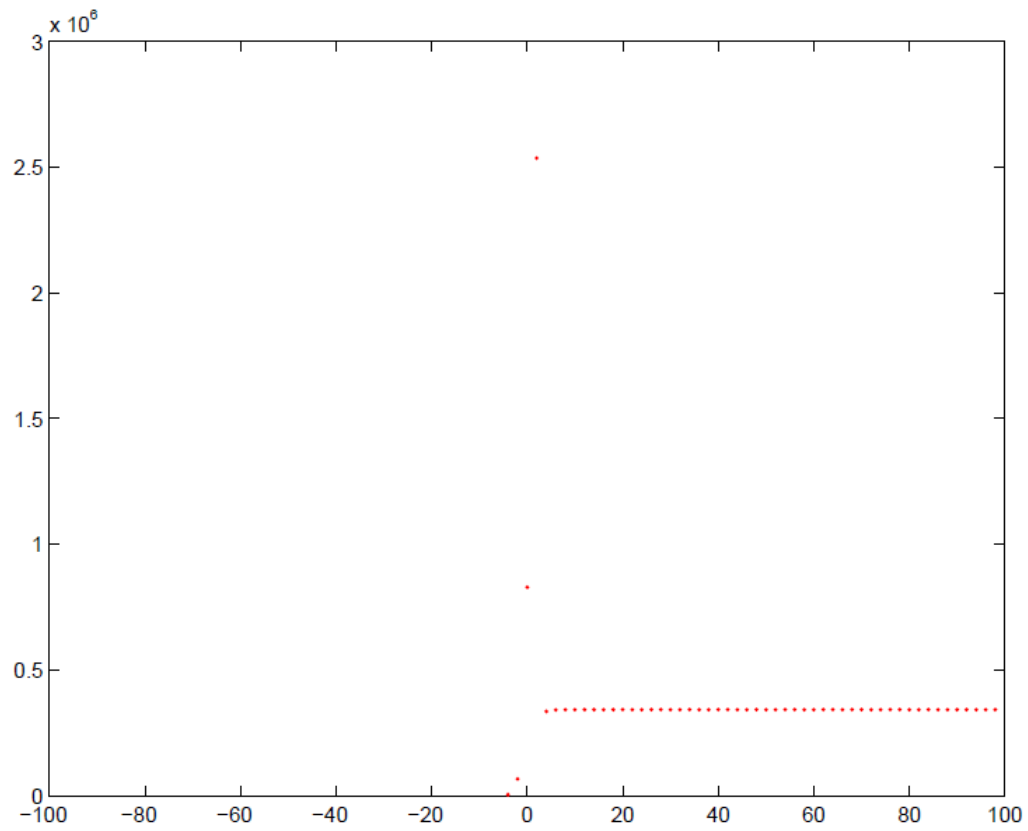


Figura C.6: Recorrido de la variable  $\theta_1$ .