

Modulo De Asignación De Agendas Basado En Algoritmos Genéticos

Jefersson Castrillon Giron
Código 201449948
castrillon.jefersson@correounivalle.edu.co

Universidad del Valle
Programa Ingeniería de Sistemas
Mayo de 2015

Modulo De Asignación De Agendas Basado En Algoritmos Genéticos

Autor

Jeferssón Castrillón Girón
Código 201449948

Coodirector

Carlos Delgado Saavedra
carlos.andres.delgado@correounivalle.edu.co

Director

Ángel García Baños Ph.D.
angel.garcia@correounivalle.edu.co

Universidad del Valle
Programa Ingeniería de Sistemas
Mayo de 2015

Índice general

Agradecimientos	v
Resumen	vi
1. Introducción	1
1.1. Descripción General	1
1.2. Estructura del Documento	2
1.3. Descripción del Problema	2
1.4. Formulación del Problema	3
1.5. Objetivo General	3
1.6. Objetivos Específicos	3
1.7. Resultados Esperados	4
2. Marco Referencial	5
2.1. Marco Histórico	5
2.2. Marco Teórico	6
2.2.1. Teoría de la Evolución	6
2.2.2. Algoritmos Evolutivos	6
2.2.3. Algoritmos Genéticos	7
2.2.4. Cromosoma	7
2.2.5. Cruce Uniforme	8
2.2.6. Cruce Por Emparejamiento Parcial (PMX)	8
2.2.7. Mutación	10
2.2.8. Selección Por Torneo	10
2.2.9. Inteligencia Artificial	11
2.2.10. Ruby on Rails	11
2.2.11. Algoritmos Genéticos y Método Enumerativo	11
2.2.12. Algoritmos Genéticos y Sistemas Expertos	12
2.3. Antecedentes	13

3. Diseño Algoritmo Genético	14
3.1. Se define un cromosoma	14
3.2. Definir una función de evaluación	16
3.3. Definir los operadores de reproducción (operadores genéticos).	16
3.4. Definir la forma en que los nuevos cromosomas reemplazarán a los viejos, en la población de N cromosomas.	17
3.5. Desarrollo Algoritmo Genético	17
4. Desarrollo de una interfaz prototipo	19
4.1. Interfaz	19
4.2. Interfaz login	19
4.3. Interfaz Pagina De Inicio	20
4.4. Interfaz Listado De Los Usuarios	21
4.5. Interfaz Detalles Del Usuario	22
4.6. Interfaz Ediccion, Modificacion y Creacion De Usuarios	23
4.7. Interfaz Listado De Las Actividades	24
4.8. Interfaz Listado De Los Consultorios	25
4.9. Interfaz, Panel Del Medico	26
5. Análisis y Pruebas	28
5.0.1. Entrada de pruebas	29
5.0.2. Entrada de pruebas	32
6. Conclusiones	36
7. Trabajos Futuros	37
Referencias	38
anexos	40

Índice de cuadros

3.1. Informacion del medicos con sus dias laborales.	15
3.2. Cromosoma. Mejor posible solucion al problema.	16
5.1. Entradas usadas para pruebas, 10 medicos.	29
5.2. Salida AG con PMX, Tiempo de ejecucion 4.107818029 segundos. .	29
5.3. Salida AG con Cruce Uniforme, Tiempo de ejecucion 4.075858705 segundos.	30
5.4. Entradas usadas para pruebas, 20 medicos.	32
5.5. Salida AG con Cruce Uniforme, Tiempo de ejecucion 5.464118657 segundos.	33
5.6. Salida AG con PMX, Tiempo de ejecucion 6.140112786 segundos. .	34

Índice de figuras

3.1. Funcionamiento de un Algoritmo Genético (Imagen tomada de las diapositivas de las clases del profesor Ángel García)	17
4.1. Funcion de registro al sistema.	20
4.2. Pantalla de inicio del administrador.	21
4.3. Listado de todos los usuarios del sistema.	22
4.4. Detalle, informacion de cada usuario.	23
4.5. Formulario de ediccion, modificacion y creacion de usuarios.	24
4.6. Listado de actividades.	25
4.7. Listado de todos los consultorios del sistema.	26
4.8. Listado de las acciones del medico.	27
5.1. Lomgitud de la poblacion VS Tiempo ejecusion (seg) operador genetico (CU vs PMX).	30
5.2. Lomgitud de la poblacion VS Tiempo ejecusion (seg) operador genetico (CU vs PMX).	35
7.1. Modelo de la base de datos que utiliza la interfaz	41
7.2. Diagrama de flujo de la aplicaci3n.	42

Agradecimientos

A los replicadores por su evolución al darme la vida y la evolución de los genes por darme la salud.

A mi familia por la guía moral y por la gran confianza depositada en mi persona.

A mis compañeros de estudio por los buenos momentos que se pasaron durante la carrera.

A mis profesores por su guía en mi aprendizaje. Al profesor Ángel García Baños y Víctor Bucheli Guerrero por la motivación en las ciencias de la computación.

Resumen

En este proyecto de trabajo de grado se diseña y desarrolla una aplicación con una interfaz prototipo para la asignación de agendas, mediante el empleo de técnicas de computación evolutiva.

Mediante el uso de las técnicas de computación evolutiva y la colaboración del grupo EVALAB de la Universidad del Valle, se realiza un Algoritmo Genético que pueda dar solución o acercarse a este al problema de asignación de horarios que presenta el Centro Cardiológico de Valle. En este documento se encuentran descritos los pasos como se abordó el problema y como se implementó el Algoritmo Genético desarrollado en el lenguaje de programación Ruby.

Finalmente, con el desarrollo de la interfaz prototipo se busca crear un diseño práctico para ingresar la información que el Algoritmo Genético necesita. La interfaz fue desarrollada con el Framework (marco de trabajo) Ruby on Rails.

Capítulo 1

Introducción

1.1. Descripción General

El problema de asignación de horarios que se aborda en este documento corresponde a una versión simplificada, donde se consideran algunas restricciones de factibilidad, las cuales son la disponibilidad de médicos y consultorios. La solución a este problema consiste en establecer una secuencia, en un periodo determinado de tiempo (específicamente en una semana), de secciones y consultas ofrecidas a los pacientes de forma que cumpla un conjunto de restricciones de diferentes tipos. Es considerado un problema combinatorial multidimensional NP-completo [1], de difícil solución.

En el presente documento se realiza un proceso de solución para el problema de asignación de agendas en el Centro Cardiológico del Valle (CCV), utilizando un algoritmo evolutivo (CE), haciendo uso de los operadores genéticos (selección, cruce y mutación). Consiste en buscar la mejor solución para la asignación de agendas, determinada por una función de evaluación.

El problema de asignación de horarios es muy común por ello es normal encontrar diferentes tipos de soluciones, como el algoritmo de búsqueda tabú que describe Hertz [2], para resolver los problemas de programación de horarios y asignación de estudiantes a grupos, en donde estos problemas se integran de forma iterativa y encontrar la asignación de profesores y salón. Otras soluciones que se pueden encontrar son las que dividen el problema para posteriormente resolverlo de forma combinada e iterativa [3, 4, 5, 6, 7]. Generalmente los subproblemas resultantes de esta división corresponden a una programación de horarios y a otro de programación de consultorios. Otras estrategias de programación de horarios se han basado en la aplicación de técnicas propias del campo de la inteligencia artificial.

1.2. Estructura del Documento

- Cap 1. Introducción. Descripción general de este trabajo de grado.
- Cap 2. Marco Referencial. Se explican los conceptos necesarios para comprender el problema y su solución
- Cap 3. Diseño Algoritmo Genético. En este capítulo se informa cómo se llevo acabo el desarrollo del algoritmo genético.
- Cap 4. Desarrollo Interfaz Prototipo. En este capítulo se informa cómo se diseño, que herramientas se implementaron y cómo funciona la interfaz gráfica.
- Cap 5. Análisis y Pruebas. En este capítulo se muestran los diferentes resultados comparativos obtenidos al realizar diferentes pruebas con la aplicación.
- Cap 6. Conclusiones. Información final de los resultados académicos que dejó el desarrollo de este trabajo de grado.
- Cap 7. Trabajos Futuros. En este capítulo se muestran los trabajos futuros que deben realizarse para mejorar la aplicación propuesta en este trabajo de grado.

1.3. Descripción del Problema

Los procesos que se realizan en el proyecto KNX (Aplicación Web Desarrollada por KENOVIX), están conformados de una serie de módulos, los cuales se dividen en sus respectivas áreas como: facturación, parametrizar, agendamiento y administración. Estos procesos se llevan a cabo por los organismos de división técnica y personal administrativo.

En general, en un modelo de asignación de agendas en un centro de atención en particular y de acuerdo a las condiciones de los pacientes que se atienden, se establecen diferentes citas dependiendo al tipo de agenda que se haya generado. Además, se debe tener en cuenta que los pacientes son tratados por uno o varios profesionales. Actualmente KNX cuenta con un sistema de asignación de agendas, el cual no se ajusta a este modelo, puesto que fue diseñado siguiendo un modelo de adjudicación de citas sistematizado, donde el paciente solicita una cita con un profesional determinado.

Para empezar el módulo de agendas de KNX el usuario Administrador debe consultar la información de la disponibilidad de los profesionales, para así poder

generar las agendas necesarias dependiendo de su tiempo laboral, dónde una vez creadas las agendas se adjudican las citas correspondientes a cada paciente. Para asignar una cita el paciente debe llamar o ir hasta la institución. El paciente será atendido por un usuario Auxiliar el cual recibirá la solicitud de la cita y procederá a registrar. Si la agenda está llena, el paciente pasa a un listado de citas encoladas al cual se concede la cita; una vez se encuentre una agenda disponible para el tipo de cita solicitada, o el paciente debe solicitar la cita otro día que se encuentren agendas disponibles. Si una agenda es cancelada por el usuario Administrador ya sea porque el profesional no puede asistir o por alguna otra razón, el usuario Auxiliar debe llamar a cada paciente para informar que su cita fue cancelada y comunicarle que espere una pronta llamada para adjudicar la cita.

El proyecto KNX, no presenta un procedimiento que asegure la distribución uniforme de pacientes entre los diferentes profesionales que atienden en una misma especialidad, lo que obliga a las secretarías a realizar este proceso manualmente con el fin de evitar la sobrecarga de trabajo entre los profesionales de la institución.

1.4. Formulación del Problema

¿Cómo asignar agendas en el sistema KNX del CCV (Centro Cardiológico del Valle) basado en Algoritmos Genéticos. ?

1.5. Objetivo General

Desarrollar una aplicación para la asignación de agendas del proyecto KNX basado en los Algoritmos Genéticos.

1.6. Objetivos Específicos

- Analizar los Algoritmos Genéticos para establecer la asignación de agendas.
- Determinar las variables relevantes en la solución del problema, así como las restricciones existentes.
- Desarrollar funciones de codificación y evaluación de los individuos, en función de las variables relevantes y teniendo en cuenta las restricciones existentes.
- Desarrollar una interfaz prototipo donde se visualizarán los individuos propuestos por los Algoritmos Genéticos.

1.7. Resultados Esperados

- Establecimiento de la relación que se le puede dar a los Algoritmos Genéticos con la asignación de agendas. Primer acercamiento de este problema por medio de estos métodos.
- Estructura del Algoritmo Genético a implantar desde las variables y restricciones del problema.
- Algoritmo Genético que permitirá asignar las agendas de KNX.
- Interfaz de usuario que hará posible visualizar los resultados obtenidos por el Algoritmo Genético.

Capítulo 2

Marco Referencial

2.1. Marco Histórico

KENOVIX es una empresa de desarrollo de software conformada por un grupo de jóvenes. En el 2011, estos jóvenes se reunían periódicamente para plantear ideas de negocios y ofrecer servicios como mantenimiento de equipos, instalación de cámaras y desarrollo de software. En el 2011 KENOVIX logra iniciar contrato con el Centro Cardiológico del Valle (CCV) para el desarrollo de un sistema de información orientado a la parte de cardiología el cual es culminado con éxito a finales del mismo año. A inicios del 2012 se registra oficialmente KENOVIX en cámara y comercio en la ciudad de Buga con NIT 1.116.156.124-7. A partir de allí inicia el desarrollo de una aplicación Web que permita prestar sus servicios a cualquier institución de salud y así nace el proyecto KNX (Aplicación Web), esté actualmente se encuentra en constantes actualizaciones y agregando nuevas funcionalidades.

El grupo de desarrollo de software está conformado por:

- Hernando Bocanegra Grajales. Tecnólogo en Sistemas, egresado de la Universidad del Valle, actualmente es el encargado del desarrollo y mantenimiento de los módulos: Facturación, Parametrización y Administración del proyecto KNX.
- Jefersson Castrillón Girón. Estudiante de Ingeniería en Sistemas de la Universidad del Valle. Actualmente es el encargado del desarrollo y mantenimiento de los módulos de Agenda miento de consultas, Informes, Historia Clínica y Usuarios del proyecto KNX.
- Aldemar Latorre Bejarano. Tecnólogo en Sistemas, egresado de la Universidad

del Valle. Actualmente es el encargado de ofrecer los servicios y realizar las contrataciones para KENOVIX y se encarga de prestar apoyo si estas lo ameritan en el desarrollo de las aplicaciones.

El módulo de Agendas se encuentra en actualización y por eso en este documento se establece extender el proyecto KNX para implementar un proceso en la asignación de agendas implementando AG.

2.2. Marco Teórico

2.2.1. Teoría de la Evolución

Los algoritmos evolutivos (AE) parten de las ideas del modelo de evolución natural que fue propuesto por Charles Darwin. De acuerdo con la teoría de Darwin, la evolución de las especies se debe al principio de selección natural, que favorece la supervivencia y multiplicación de aquellas especies que están mejor adaptadas a las condiciones de su entorno. Otro elemento que Darwin señaló como relevante para la evolución son las mutaciones, pequeñas variaciones que introducen diferencias en las características físicas y tipos de respuesta de los padres y los hijos. El mecanismo que fuerza la actuación de la selección es la producción de descendencia. Mientras hay abundancia de recursos, la población crece exponencialmente. Este proceso lleva a situaciones de escasez de recursos en el entorno, en la que los individuos "mejor adaptados" al medio tienen mayor probabilidad de sobrevivir y de dejar descendencias. [8].

Estas ideas están en la base del diseño de los AE. Sin embargo, los algoritmos evolutivos no tratan de ser un reflejo fiel de la evolución biológica. Se debe tener en cuenta que la naturaleza evoluciona a lo largo de millones de años, mientras que para los Algoritmos es de interés que estos proporcionen una solución en un tiempo algo mas corto.

2.2.2. Algoritmos Evolutivos

Los Algoritmos Evolutivos (AEs) son una técnica de resolución de problemas de búsqueda y optimización inspirada en la teoría de la evolución de las especies y la selección natural. Estos algoritmos reúnen características de búsqueda aleatoria con características de búsqueda dirigida que provienen del mecanismo de selección de los individuos más adaptados. La unión de ambas características les permite abordar los problemas de una forma muy particular, ya que tienen capacidad para acceder a cualquier región del espacio de búsqueda del problema, capacidad de

la que carecen otros métodos de búsqueda exhaustiva, a la vez que exploran el espacio de soluciones de una forma mucho más eficiente que los métodos puramente aleatorios. Indudablemente, un algoritmo diseñado de forma específica para la resolución de un problema concreto será más eficiente que un algoritmo evolutivo, que es una técnica general de resolución. Pero existen muchas situaciones en las que no es posible contar con tales algoritmos[8]

2.2.3. Algoritmos Genéticos

Los Algoritmos Genéticos (AGs) son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos [9].

Los algoritmos genéticos son: [10] Estrategias adaptativas para la solución de diversos problemas de búsqueda y de optimización usando conceptos de evolución. Un algoritmo genético esta definido por:

- Una población inicial que se genera de manera aleatoria, cada individuo es una solución validada a un problema que se busca solucionar.
- Un método de selección que clasifica de una población a los mejores individuos con mayor probabilidad.
- Cruce que es tomar los individuos que han sido seleccionados y combinarlos para generar mas individuos que recojan sus características.
- Mutación, que es alterar características a unos pocos individuos generados en el cruce.
- Los pasos 2, 3 y 4 se repiten un numero determinado de veces o cuando los mejores individuos cumplan ciertos requisitos definidos por el programador.

La idea principal es que el algoritmo a medida que itera se va acercando a la solución optima del problema que se intentan solucionar, aunque este no garantice el optima.

2.2.4. Cromosoma

Es un conjunto de parámetros que definen una solución propuesta al problema que el Algoritmo Genético está intentando resolver. El cromosoma se representa a menudo como una serie de bits, aunque también se utilizan una variedad amplia de otras estructuras de datos [10].

2.2.5. Cruce Uniforme

Consiste en elegir un subsegmento de uno de los progenitores, cruzarlos y agregar los genes de los progenitores al hijo. La descendencia contiene una mezcla de cada uno de los padres [8].

Primero se establecen los puntos de corte, segundo se realiza el cruce sin tener en cuenta los datos repetidos, tercero se da a luz el nuevo hijo. El proceso es el siguiente. Se tienen tres arreglos Madre, Padre y Mascara, este ultimo arreglo especifica los genes de los padres que se van a cruzar para dar a luz el hijo. los genes con valor uno (1) del arreglo Mascara, especifican los genes del Padre que se van a cruzar y los genes de valor cero (0) especifican los genes de Madre que serán cruzados. Se presentan los dos siguientes ejemplos.

Ejemplo 1.

"[1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0]"Mascara

"[1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9]"Padre

"[1 | 2 | 3 | 1 | 8 | 7 | 6 | 9 | 3]"Hijo

"[4 | 5 | 2 | 1 | 8 | 7 | 6 | 9 | 3]"Madre

Ejemplo 2.

"[1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1]"Mascara

"[1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9]"Padre

"[1 | 2 | 3 | 1 | 8 | 7 | 7 | 8 | 9]"Hijo

"[4 | 5 | 2 | 1 | 8 | 7 | 6 | 9 | 3]"Madre

2.2.6. Cruce Por Emparejamiento Parcial (PMX)

Consiste en elegir un tramo de uno de los progenitores y cruzar preservando el orden y la posición de la mayor cantidad posible [8].

Algoritmo:

- Elegir aleatoriamente dos puntos de corte.
- Intercambiar las dos subcadenas comprendidas entre dichos puntos en los hijos que se generan.
- Para los valores que faltan en los hijos se copian los valores de los padres:
- Si un valor no está en la subcadena intercambiada, se copia igual.

- Si está en la subcadena intercambiada, entonces se sustituye por el valor que tenga dicha subcadena en el otro padre.

Ejemplo.

- Se tienen los siguientes padres. y un arreglo de poscisiones.

"[a | b | c | d | e | f | g | h | i]"poscisiones

"[1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9]"Padre

"[4 | 5 | 2 | 1 | 8 | 7 | 6 | 9 | 3]"Madre

- Se selecciona un recorrido parcial eligiendo al azar dos puntos de corte (Ejemplo D y G).

"[a | b | c | x | x | x | x | h | i]"Poscisiones

"[1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9]"Padre

"[4 | 5 | 2 | 1 | 8 | 7 | 6 | 9 | 3]"Madre

- Se intercambian los segmentos situados entre los puntos de corte y se agregan a los hijos.

"[a | b | c | x | x | x | x | h | i]"Poscisiones

"[1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9]"Padre

"[4 | 5 | 2 | 1 | 8 | 7 | 6 | 9 | 3]"Madre

A esta descendencia se le agregan los genes de la Madre previamente seleccionados

"[x | x | x | 1 | 8 | 7 | 6 | x | x]"Hijo 1

A esta descendencia se le agregan los genes del Padre previamente seleccionados.

"[x | x | x | 4 | 5 | 6 | 7 | x | x]"Hijo 2

- Se agregan los genes faltantes que no generan conflictos de los padres a cada hijo.

"[x | x | x | d | e | f | g | x | x]"Poscisiones

"[1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9]"Padre

"[4 | 5 | 2 | 1 | 8 | 7 | 6 | 9 | 3]"Madre

Los siguientes genes provenientes del Padre no generan conflictos son ingresados a su descendencia.

"[x | 2 | 3 | 1 | 8 | 7 | 6 | x | 9]"Hijo 1

Los siguientes genes provenientes de la Madre no generan conflictos son ingresados a su descendencia.

"[x | x | 2 | 4 | 5 | 6 | 7 | 9 | x]"Hijo 2

Se especifican las x en los hijos que plantean conflicto.

- Las x que plantean conflicto se reemplazan por su pareja, es decir se busca el gen faltante en los primeros genes de su hermano que le fueron suministrados por su primer progenitor. Ejemplo, Hijo 1 presenta en la primera posición un conflicto, para agregar el gen correspondiente se busca el gen que le hace falta en los primeros genes que le suministro el progenitor a Hijo 2 (4,5,6,7) entonces el gen faltante es (4). Se sustituye por 4.

Primeros genes suministrados por sus progenitores.

"[x | x | x | 1 | 8 | 7 | 6 | x | x]"Hijo 1

"[x | x | x | 4 | 5 | 6 | 7 | x | x]"Hijo 2

Resultado del cruce. Descendencia de los progenitores.

"[4 | 2 | 3 | 1 | 8 | 7 | 6 | 5 | 9]"Hijo 1

"[1 | 8 | 2 | 4 | 5 | 6 | 7 | 9 | 3]"Hijo 2

- Se obtienen soluciones factibles y sin datos repetidos.

2.2.7. Mutación

En este tipo de mutación se seleccionan dos puntos al azar (dos genes) del individuo y se intercambian los valores. Se tienen los dos siguientes valores (genes) 6 y 3, estos valores se cambian entre si en el individuo, vease a continuación:

"[1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1]"Selección de puntos al azar.

"[1 | 2 | 6 | 9 | 4 | 7 | 3 | 8 | 5]"Individuo sin mutacion.

"[1 | 2 | 3 | 9 | 4 | 7 | 6 | 8 | 5]"Individuo despues de ser mutado.

2.2.8. Selección Por Torneo

Esta selección que se establece en el AG es la selección por torneo determinista, donde se selecciona al azar un número p de individuos, donde p=3. De entre los individuos seleccionados se selecciona el más apto para pasarlo a la siguiente generación. La selección del individuo mas apto se establece por medio de una

funcion de evaluacion (fitness), esta funcion es de maximizacion, es decir que el individuo con el valor mas alto (puntaje) es el mas apto.

2.2.9. Inteligencia Artificial

Se define la Inteligencia Artificial (IA) como aquella inteligencia exhibida por artefactos creados por humanos (es decir, artificial). A menudo se aplica hipotéticamente a los computadores. El nombre también se usa para referirse al campo de la investigación científica que intenta acercarse a la creación de tales sistemas [11].

La IA tuvo muchos padres, como: Alan Turing su principal progenitor, en 1937 publicó un artículo de bastante repercusión sobre los "Números Calculables", un artículo que estableció las bases teóricas para todas las ciencias de computación, y que puede considerarse el origen oficial de la informática teórica. No hay un consenso para definir el concepto de la IA, pero se puede decir que se encarga de modelar muchas formas inteligentes como la inteligencia humana en sistemas computacionales. Puede decirse que la IA es una de las áreas más fascinantes y con más retos de las ciencias de la computación, en su área de ciencias cognitivas. Nació como mero estudio filosófico y razonístico de la inteligencia humana, mezclada con la inquietud del hombre de imitar la naturaleza circundante (como volar y nadar), hasta inclusive querer imitarse a sí mismo. Sencillamente, la IA busca el imitar la inteligencia humana. Obviamente, no se ha logrado, imitar la inteligencia humana, almenos no completamente [11]. Un ejemplo de este es el Test de Turing (o prueba de turing) es una prueba propuesta por Alan Turing para demostrar la existencia de inteligencia en una máquina. Se fundamenta en la hipótesis positivista (se afirma que el unico conocimiento autentico es el conocimiento científico) de que, si una máquina se comporta en todos los aspectos como inteligente, entonces debe ser inteligente

2.2.10. Ruby on Rails

Es un framework (marco de herramientas) para el desarrollo web, basado en el lenguaje de programación ruby.

2.2.11. Algoritmos Genéticos y Método Enumerativo

Existe la posibilidad teórica de encontrar soluciones a problemas de optimización enumerando todas las soluciones posibles para todos los casos y posteriormente buscando la misma en la base de datos resultante. Los problemas se limitan entonces a un sistema de búsqueda eficiente del caso concreto. Por ejemplo los libros con tablas de logaritmos tradicionales constan de una larga serie de cálculos para todos

los valores usuales. La solución consiste simplemente en buscar en la lista el número decimal y retorna el logaritmo dado. Este método es factible siempre que el número de valores sea manejable. De otra manera el simple cálculo de los mismos se vuelve imposible. Ejemplo: Generar una tabla que contenga todas las movidas de todos los partidos posibles de un juego de damas resultaría imposible de hacer en la práctica [12]. La “memorización” de una serie de datos no es otra cosa que la construcción en la memoria del equivalente a una base de datos en donde se busca la pregunta y se encuentra automáticamente la respuesta. Los algoritmos genéticos usan heurística para la resolución de problemas, lo cual limita drásticamente el número de datos a utilizar

2.2.12. Algoritmos Genéticos y Sistemas Expertos

Un Sistema Experto (SE) es un conjunto de programas que, sobre una base de conocimientos, posee información de uno o más expertos en un área específica. Se puede entender como una rama de la IA, donde el poder de resolución de un problema en un programa de computadora viene del conocimiento de un dominio específico. Estos sistemas imitan las actividades de un humano para resolver problemas de distinta índole, no necesariamente tiene que ser de IA. También se dice que un SE se basa en el conocimiento declarativo como hechos sobre objetos, situaciones y el conocimiento de control de información sobre el seguimiento de una acción [13].

Para que un sistema experto sea herramienta efectiva, los usuarios deben interactuar de una forma fácil, reuniendo dos capacidades para poder cumplirlo:

Explicar sus razonamientos o base del conocimiento: los sistemas expertos se deben realizar siguiendo ciertas reglas o pasos comprensibles de manera que se pueda generar la explicación para cada una de estas reglas, que a la vez se basan en hechos.

Adquisición de nuevos conocimientos o integrador del sistema: son mecanismos de razonamiento que sirven para modificar los conocimientos anteriores. Sobre la base de lo anterior se puede decir que los sistemas expertos son el producto de investigaciones en el campo de la inteligencia artificial ya que ésta no intenta sustituir a los expertos humanos, sino que se desea ayudarlos a realizar con más rapidez y eficacia todas las tareas que realiza

Los algoritmos genéticos se revalorizaron ya que poseen las siguientes ventajas competitivas:

- Solo necesitan asesoramiento del experto cuando se agregan o suprimen variables al modelo. Los Sistemas Expertos requieren la presencia del mismo ante cada modificación del entorno.

- Los algoritmos genéticos sólo requieren el asesoramiento del experto para identificar las variables pertinentes, aunque no es necesario que éstos definan sus valores ni sus relaciones (las reglas) iniciales o finales. Los Sistemas Expertos solo trabajan con las reglas y valores que les dictan los seres humanos.

2.3. Antecedentes

Actualmente en el proyecto del Centro Cardiológico Del Valle se ha implementado una forma muy directa de dar solución al problema de la asignación de agendas, el usuario se encarga de hacer todo el procedimiento de forma casi manual, se ayuda del proyecto KNZ, aunque este no ha tenido éxito porque este proceso no presenta un procedimiento que asegure la distribución uniforme de pacientes entre los diferentes profesionales que atienden en una misma especialidad, lo que obliga a las secretarías a realizar este proceso manualmente.

Con el fin de evitar la sobrecarga de trabajo entre los profesionales de la institución. Debido a esta situación (problema que se encuentra descrito en la descripción del problema) se ha optado por implementar los AG para resolver el problema y la solución propuesta de los AG es encontrar la mejor distribución de la asignación de agendas.

En semestres anteriores se desarrolló un proyecto que busca el desarrollo de un modelo de satisfacción de restricciones para la asignación de citas terapéuticas en la Clínica de Rehabilitación del Valle, aunque el proyecto se culminó con éxito este fue realizado a la medida y solo satisface la distribución de citas. Los pacientes cuentan con un número de citas por determinado tiempo. Este modelo no se ajusta al proyecto Centro Cardiológico Del Valle, porque los pacientes piden una única cita con un profesional, es decir cuando se asignan las agendas a los profesionales el paciente puede pedir una cita con ese profesional para ese día y cuando el paciente quiera atender a otra cita, este debe volver a solicitarla.

Capítulo 3

Diseño Algoritmo Genético

3.1. Se define un cromosoma

Antes de generar el cromosoma se genera la población inicial. Este parámetro nos indica el número de cromosomas que se tiene en la población para una generación determinada. En caso de que esta medida sea insuficiente, el AG tiene pocas posibilidades de realizar reproducciones, con lo que se realizaría una búsqueda de soluciones escasa y sin resultados correctos. Por otro lado si la población es excesiva, el AG será excesivamente lento. Se puede concluir que existe un límite a partir del cual es ineficiente elevar el tamaño de la población; puesto que no se consigue una mayor velocidad en la resolución del problema. En este caso el límite de la población máxima tiene como cota un valor de cien (100) individuos para el AG. El límite de la población se determina porque se verifica que el AG no va muy lento y encuentra resultados correctos.

En la construcción del cromosoma los genes almacenan un símbolo. En este caso cada gen almacena un valor numérico entero, este hace referencia a los consultorios, además toma los diferentes alelos que van desde el consultorio 1 hasta el consultorio N. Los consultorios son ingresados al sistema por medio de la interfaz prototipo, esta se explica mas adelante.

A continuación se presenta la Tabla:3.1 con la información que se debe ingresar al sistema y del que se alimenta el AG para crear un cromosoma. Los datos que se tienen en cuenta para generar este, son los que se presenta en la columna (Días de la semana "Lunes a Viernes"), con la información de esta columna se genera la población inicial y los cromosomas. La columna seleccionada presenta la siguiente información y el significado de estos datos se definen a continuación. (x: no trabajar, f: trabajar tiempo completo, m: trabajar medio tiempo en la mañana, a: trabajar medio tiempo en la tarde).

Descripción de las columnas que se presenta en la Tabla:3.1 :

- Nombre y Apellido del médico.
- Tipo de procedimiento que realiza un médico al paciente (generalmente en la clínica por medio del nombre de éste, se genera el nombre de la agenda).
- Días de la semana, estos representan los días laborales, en este caso los días laborales están predefinidos por la clínica que van de Lunes a Viernes.

Nombre	Apellido	Tipo de procedimiento	Días de la semana "Lunes a Viernes"
Medico 1	Apellido 1	Procedimiento 6	x f m a f
Medico 2	Apellido 2	Procedimiento 5	f f x m f
Medico 3	Apellido 3	Procedimiento 4	a x m x x
Medico 4	Apellido 4	Procedimiento 3	m a x f x
Medico 5	Apellido 5	Procedimiento 2	a m a x x
Medico 6	Apellido 6	Procedimiento 1	m x f f f

Cuadro 3.1: Informacion del medicos con sus dias laborales.

Se toma la Tabla:3.1 como ejemplo de un problema a resolver mediante los AGs. Se alimenta el AG con la información requerida (Médicos, Consultorios y Días laborales); para los consultorios se supone que se tienen tres (3) disponibles. Médicos y Días laborales se toman los de la Tabla:3.1. El objetivo es generar un cromosoma que represente la solución correcta de agendar las actividades de la Tabla:3.1 con su respectivo médico y los días laborales de éste. Ahora que el AG cuenta con la información necesaria para realizar su búsqueda se procede a ejecutarlo.

El AG establece la población inicial y realiza los siguientes procesos (Cruce, Mutación y Selección) que se explican mas adelante.

Una vez finalizado el AG se presenta un cromosoma con la solución correcta. Cabe resaltar que los AG no garantizan una solución óptima pero si un acercamiento a esta.

La Tabla:3.2 representa a un cromosoma (solución del problema de asignación de la Tabla:3.1) correcto generado por el AG. Para entender este cromosoma de longitud $7 \times n$, $n=6$ (días de la semana por la cantidad de médicos disponibles para laborar) y representa la mejor solución posible para asignar las agendas correspondiente. Los valores enteros en las columnas (Lu, Ma, Mi, Ju, Vi) del cromosoma representan

los consultorios (Consultorio1 = 1, Consultorio2 = 2, Consultorio3 = 3). Ejemplo de interpretación del cromosoma: El medico1 labora el dia martes en el consultorio 1 tiempo completo. De esta forma se puede interpretar la información generada por el AG.

Medico	Procedimientos	Ln	Ma	Mi	Ju	Vi
Medico1	procedimiento6	1	1	2	3	3
Medico2	procedimiento5	1	2	2	3	1
Medico3	procedimiento4	2	1	3	3	2
Medico4	procedimiento3	2	3	2	2	2
Medico5	procedimiento2	3	3	2	1	3
Medico6	procedimiento1	3	1	1	1	2

Cuadro 3.2: Cromosoma. Mejor posible solucion al problema.

3.2. Definir una función de evaluación

buscar la forma de normalizar la función de evaluación para conseguir la función de aptitud (fitness). La función aplicada a un cromosoma, informa lo buena que es la solución representada por el.

Para formalizar la función, tomamos los cromosomas y evaluamos las condiciones necesarias, en este caso se maximiza la función donde se evalúa cada gen con la información del médico, si cumple o no con la restricción, es decir si los horarios de atención que establece el médico coinciden con los genes del cromosoma, si es así el cromosoma que más cumpla la restricción será el más apto, sin descartar el resto de cromosomas.

3.3. Definir los operadores de reproducción (operadores genéticos).

Los operadores de reproducción que fueron seleccionados para el AG son:

- Selección por torneo, es decir que el AG toma tres (3) individuos que serian dos posibles soluciones al problema de asignación y entre estos se toman los dos mas aptos donde posteriormente se les realiza un cruce y una mutación.
- Mutación aleatoria. La mutación que presenta este AG, es básicamente tomar un gen de forma aleatoria y cambiarlo por otro del mismo cromosoma. Se

tiene una arreglo $[1,2,3,4,5,6,7]$ y se necesita aplicar una mutacion, despues de realizar la mutacion su estructura es la siguiente $[1,2,3,5,6,7,4]$.

- Cruce por emparejamiento parcial PMX. Consiste en elegir un subsegmento de uno de los progenitores y cruzarlos preservando el orden y la posición de la mayor cantidad de genes posible del otro manteniendo la coherencia.
- Cruce uniforme. Consiste en elegir un subsegmento de uno de los progenitores, cruzarlos y agregar los genes de los progenitores que no se encuentran en el hijo.

3.4. Definir la forma en que los nuevos cromosomas reemplazarán a los viejos, en la población de N cromosomas.

La forma en que los nuevos cromosomas reemplazarán a los viejos, será realizado mediante las nuevas generación; es decir la nueva población reemplaza la anterior.

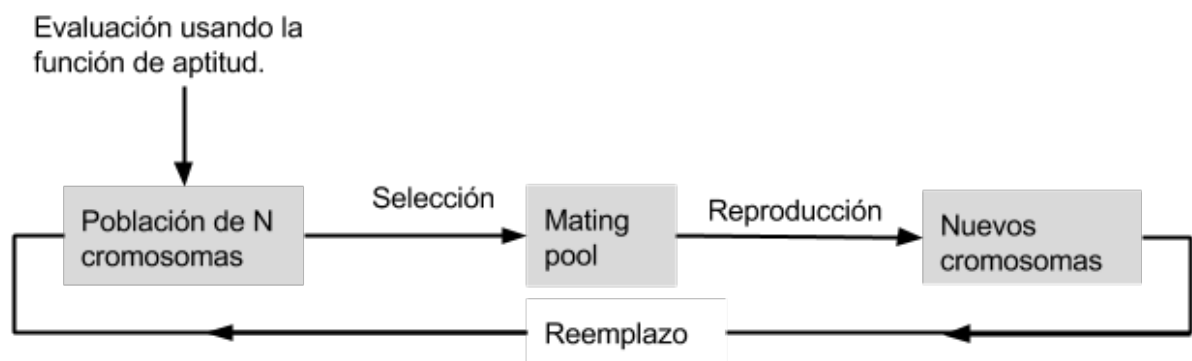


Figura 3.1: Funcionamiento de un Algoritmo Genético (Imagen tomada de las diapositivas de las clases del profesor Ángel Garcia)

3.5. Desarrollo Algoritmo Genético

Para el desarrollo del AG primero se establece la población inicial. La población inicial se genera de forma aleatoria y está compuesta por n cromosomas que representan las posibles soluciones al problema de asignación de horarios. Para este caso $n = 100$.

El AG se encarga de crear los cromosomas (individuos) y cada uno de estos esta compuesto por una serie de genes. Una vez se establece la población inicial, es necesario hacer evolucionar la población hasta que encuentre una respuesta correcta al problema o una que se aproxime a ella. De esta forma se puede interpretar como una condición de parada de la evolución, se puede asignar k cantidad de iteraciones en la evolución. Para este problema la condicion de parada depende de los resultados obtenidos, es decir, si se presenta el mismo cromosoma después de cada evolución por un periodo de iteraciones z , se retorna ese cromosoma como salida. Esto se puede concluir partiendo que la población no esta evolucionando y ha llegado a un tope de su desarrollo.

El AG se encarga de hacer evolucionar la población como se mencionó en el diseño de este, se realiza una selección por torneo, esta selección retorna dos cromosomas con buena capacidad para ser progenitores o individuos aptos para reproducción, los cromosomas deben pasar por la función de evaluación (fitness) para ser establecidos como aptos de reproducirse. Después de realizar la selección se realiza un cruce, utilizando el cruce PMX o el cruce Uniforme. El cruce depende del problema que identifique el AG, esto se debe a que pueden existir casos donde el cruce PMX sea mejor que el cruce uniforme. ¿Por que dos tipos de cruce? Existe el caso donde los genes no pueden repetirse si la cantidad de consultorios es igual a la cantidad de médicos, entonces se usa el cruce PMX, de lo contrario se utiliza el cruce uniforme (por motivos académicos con los AG se utiliza el cruce PMX pero este no es necesario, si la cantidad de consultorios es igual a la de los médicos, de hecho no se necesita aplicar el AG).

Después que se ha realizado el cruce correctamente, estos nuevos cromosomas se van reemplazando a los cromosomas viejos hasta pasar por toda la población. Luego de generar la nueva población se realiza una mutación a los nuevos cromosomas (nueva generación).

Capítulo 4

Desarrollo de una interfaz prototipo

4.1. Interfaz

El desarrollo de la interfaz se lleva a cabo para brindar al usuario un buen manejo de la aplicación y permita ingresar de forma cómoda los datos, cuando el AG sea ejecutado tomara esta información que se le suministro para generar la mejor respuesta. El desarrollo de la interfaz se realizo con el framework Ruby on Rails y tomo los estilos del bootstrap y CSS3.

En las siguientes secciones se explica la funcionalidad de la interfaz. Hay que tener en cuenta que el diseño de la interfaz es un prototipo sin muchos efectos visuales. Los suficientes efectos y colores para que este sea agradable a la vista.

4.2. Interfaz login

El inicio de la aplicación consta de una interfaz de login, que recibe dos parametros como el usuario y la contraseña. La interfaz fue maquetada en HTML5, este brinda herramientas para realizar buenos estilos. La ventana del login cuenta con un contenedor donde se encuentra el formulario centrado, los bordes utilizan el efecto del (border-radius) dando un efecto redonde y un efecto de sombreado con (box-shadow) estos efectos son realizados gracias al motor de plantillas CSS3, Figura:4.1.

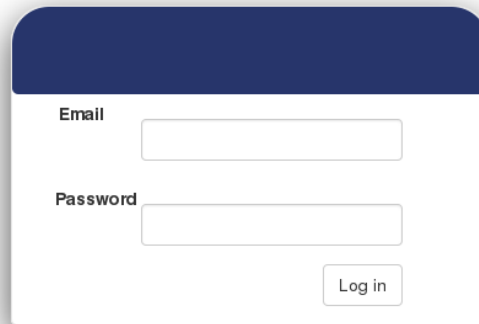
A login form with a dark blue header bar. Below the header, there are two input fields: one labeled 'Email' and one labeled 'Password'. To the right of the 'Password' field is a 'Log in' button.

Figura 4.1: Funcion de registro al sistema.

4.3. Interfaz Pagina De Inicio

El sistema presenta al administrador la información de los médicos que ya generaron sus tiempos laborales, con esta información el administrador puede ejecutar el AG para generar la mejor opción de agendamiento. Esta ventana es el inicio de administrador Figura:4.2.



The screenshot shows a web browser at localhost:3000/admin_panel. The navigation bar includes links for Tesis, Agenda, Usuarios, Cargos, and Consultorios. The user is logged in as Hernando Bocanegra Grajales. The main content area displays a table titled 'Listado De Cargos Relacionados Al Usuario'.

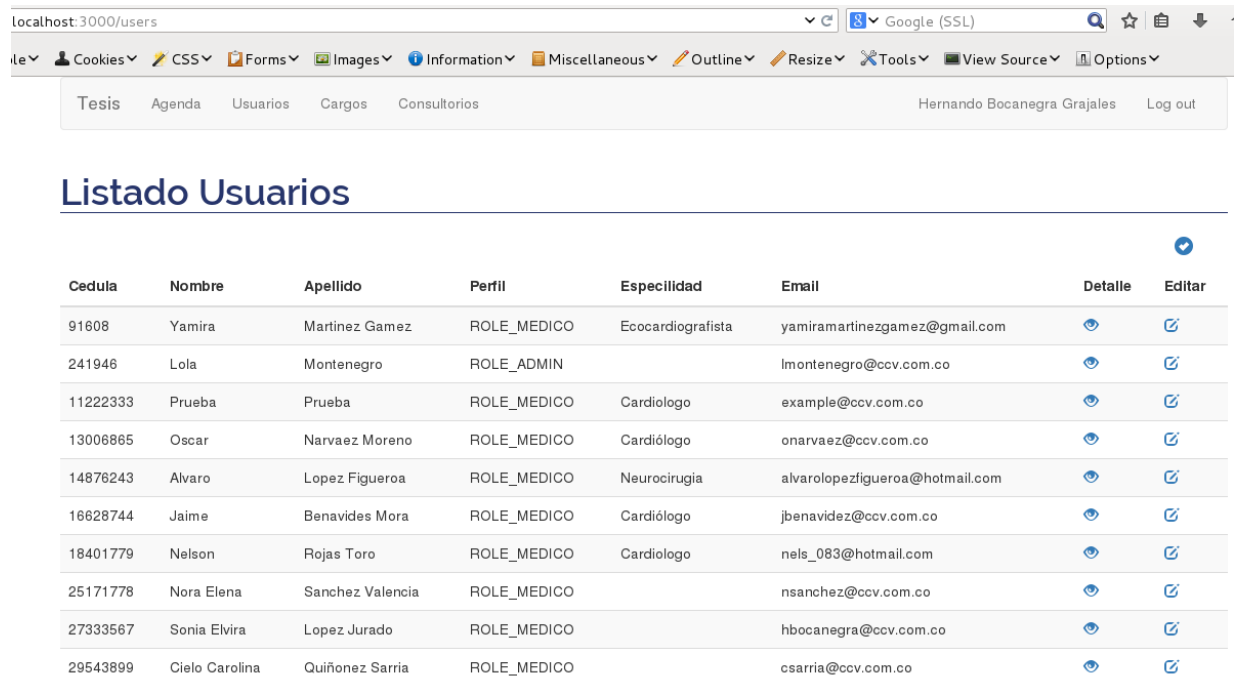
Index	Usuario	Perfil	Cargo	Lu	Ma	Mi	Ju	Vi
1	Oscar Narvaez Moreno	ROLE_MEDICO	DIPIRIDAMOLes	x	x	x	f	f
2	Oscar Narvaez Moreno	ROLE_MEDICO	CONSULTA CARDIOLOGÍA PEDIATRICA	f	f	f	x	x
3	Yamira Martinez Gamez	ROLE_MEDICO	ELECTROCARDIOGRAFIA DINAMICA (HOLTER)	f	x	a	m	x
4	Yamira Martinez Gamez	ROLE_MEDICO	ESPIROMETRIA O CURVA DE FLUJO VOLUMEN PRE Y POST BRONCODILADORES	x	m	x	a	m
5	Yamira Martinez Gamez	ROLE_MEDICO	TERAPIA CARDIO PULMONAR	x	a	m	x	a
6	Alvaro Lopez Figueroa	ROLE_MEDICO	DIPIRIDAMOLes	f	f	f	f	f
7	Prueba Prueba	ROLE_MEDICO	CONSULTA CARDIOLOGÍA PEDIATRICA	m	a	m	x	f
8	Prueba Prueba	ROLE_MEDICO	ECOCARDIOGRAMA DE STRESS CON PRUEBA DE ESFUERZO O CON PRUEBA FARMACOLOGICA	x	m	a	f	x
9	Jaime Benavides Mora	ROLE_MEDICO	CONSULTA CARDIOLOGÍA PEDIATRICA	f	f	f	a	a

Figura 4.2: Pantalla de inicio del administrador.

4.4. Interfaz Listado De Los Usuarios

En el menú de navegación del administrador se puede encontrar otras opciones como usuarios, cargos y consultorios. En la opción de usuarios se puede encontrar la siguiente información.

Como se puede observar en la Figura:4.3 presenta el listado de los usuarios, se muestran todos los usuarios del sistema con dos opciones para cada uno de ellos, la primera opción es detalle, esta se encarga de mostrar la información básica del usuario y permite asignar al usuario la actividad que realiza, la siguiente opción es editar, se encarga de modificar la información personal de usuario que se ha seleccionado.



Cedula	Nombre	Apellido	Perfil	Especialidad	Email	Detalle	Editar
91608	Yamira	Martinez Gamez	ROLE_MEDICO	Ecocardiografista	yamiramartinezgamez@gmail.com		
241946	Lola	Montenegro	ROLE_ADMIN		lmontenegro@ccv.com.co		
11222333	Prueba	Prueba	ROLE_MEDICO	Cardiologo	example@ccv.com.co		
13006865	Oscar	Narvaez Moreno	ROLE_MEDICO	Cardiólogo	onarvaez@ccv.com.co		
14876243	Alvaro	Lopez Figueroa	ROLE_MEDICO	Neurocirugia	alvarolopezfigueroa@hotmail.com		
16628744	Jaime	Benavides Mora	ROLE_MEDICO	Cardiólogo	jbenavidez@ccv.com.co		
18401779	Nelson	Rojas Toro	ROLE_MEDICO	Cardiologo	nels_083@hotmail.com		
25171778	Nora Elena	Sanchez Valencia	ROLE_MEDICO		nsanchez@ccv.com.co		
27333567	Sonia Elvira	Lopez Jurado	ROLE_MEDICO		hbocanegra@ccv.com.co		
29543899	Cielo Carolina	Quiñonez Sarria	ROLE_MEDICO		csarria@ccv.com.co		

Figura 4.3: Listado de todos los usuarios del sistema.

4.5. Interfaz Detalles Del Usuario

En la Figura:4.4 se presenta toda la informacion detallada del usuario. donde se asignan las actividades (cargos) que este va a realizar en en la semana, tambien se brinda la obcion de eliminar las actividades del usuario. Estas acciones son permitidas solo para los administradores, ellos son los que se encargan de realizar las agendas.

[Tesis](#) [Agenda](#) [Usuarios](#) [Cargos](#) [Consultorios](#) Hernando Bocanegra Grajales [Log out](#)

Detalles Del Usuario

Nombre	Cielo Carolina
Apellido	Quiñonez Sarria
Cedula	29543899
Perfil	ROLE_MEDICO

Especialidad	
Email	csarria@ccv.com.co
Direccion	
Telefono	3156105921

Listado De Cargos Disponibles Para El Usuario

Codigo	Nombre	Opcion
890202	CONSULTA CARDIOLOGÍA PEDIATRICA	<input checked="" type="checkbox"/>
890203	CONSULTA INTERDICCIÓN	<input checked="" type="checkbox"/>
890202	CONSULTA PRIMERA VEZ MEDICINA ESPECIALIZADA	<input checked="" type="checkbox"/>
920408	DIPIRIDAMOLes	<input checked="" type="checkbox"/>
881236	ECOCARDIOGRAMA DE STRESS CON PRUEBA DE ESFUERZO O CON PRUEBA FARMACOLOGICA	<input checked="" type="checkbox"/>

Listado De Cargos Relacionados Al Usuario

Codigo	Nombre	Opcion
010101	TERAPIA CARDIO PULMONAR	<input type="checkbox"/>

[← Previous](#) [1](#) [2](#) [3](#) [4](#) [Next →](#)

Figura 4.4: Detalle, informacion de cada usuario.

4.6. Interfaz Ediccion, Modificacion y Creacion De Usuarios

En la siguiente Figura: 4.4 se visualiza la opción de edición de cada usuario, cabe resaltar que esta misma pantalla hace referencia para la creación de nuevos usuario con la diferencia que el formulario esta en blanco para agregar la información del nuevo usuario.

localhost:3000/users/19/edit

Tesis Agenda Usuarios Cargos Consultorios

Hernando Bocanegra Grajales Log out

Editar Usuario

Cedula

Nombre

Apellido

Perfil

Telefono

Direccion

Especialidad

Email

Password

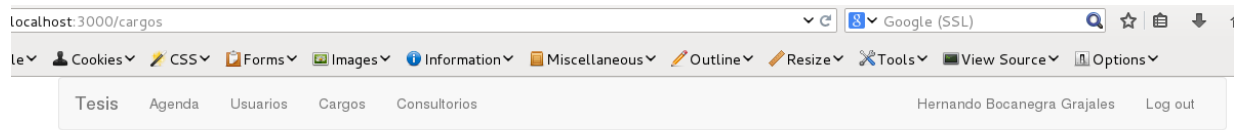
Password confirmation

Aceptar

Figura 4.5: Formulario de ediccion, modificacion y creacion de usuarios.

4.7. Interfaz Listado De Las Actividades

Siguiente opcion del menú de navegacion que son los cargos. En la figura: 4.6 se visualiza un listado de todos los cargos y cada uno con su respectiva opción (detalle y editar), este listado es similar al de los usuarios y básicamente tienen la misma funcionalidad que se explicó con el listado de los usuarios, solo que esta información es explícitamente de los cargos, por eso no vamos a explicar en detalle.



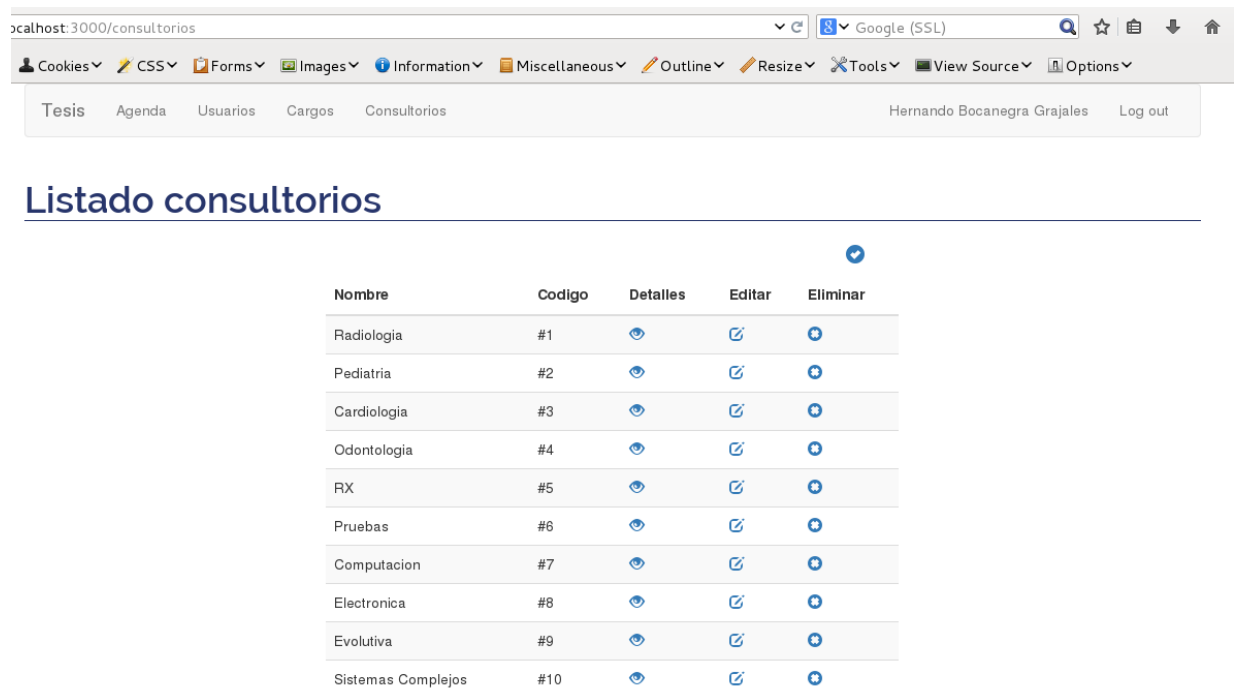
Listado actividades

Codigo	Nombre	Detalles	Editar
890202	CONSULTA CARDIOLOGÍA PEDIATRICA		
890203	CONSULTA INTERDICCIÓN		
890202	CONSULTA PRIMERA VEZ MEDICINA ESPECIALIZADA		
920408	DIPIRIDAMOLes		
881236	ECOCARDIOGRAMA DE STRESS CON PRUEBA DE ESFUERZO O CON PRUEBA FARMACOLOGICA		
881234	ECOCARDIOGRAMA MODO M Y BIDIMENSIONAL CON DOPPLER A COLOR		
881234	ECOCARDIOGRAMA PEDIATRICO		
895001	ELECTROCARDIOGRAFIA DINAMICA (HOLTER)		
895100	ELECTROCARDIOGRAMA DE RITMO O DE		

Figura 4.6: Listado de actividades.

4.8. Interfaz Listado De Los Consultorios

En la siguiente opción de los consultorios es similar a los cargos y usuario así que a continuación se presenta el listado de la vista de este sin ahondar en el figura: 4.7.



Nombre	Codigo	Detalles	Editar	Eliminar
Radiologia	#1			
Pediatria	#2			
Cardiologia	#3			
Odontologia	#4			
RX	#5			
Pruebas	#6			
Computacion	#7			
Electronica	#8			
Evolutiva	#9			
Sistemas Complejos	#10			

Figura 4.7: Listado de todos los consultorios del sistema.

4.9. Interfaz, Panel Del Medico

Ahora que ha finalizado el perfil del administrador, sigue el perfil del médico. En este perfil se encuentra un listado con las actividades que fue asignado por el administrador, como se ve en la siguiente figura: 4.8. Se puede encontrar en este listado las opciones de configuración para los días que el médico va a laborar, cuando este finaliza la asignación de sus días laborales la información queda disponible para ser visualizada. Si el médico quiere modificar la información debe pedirle autorización al administrador, se consulta al administrador porque esta información es relevante para el buen funcionamiento del AG.

localhost:3000/user_cargo_works

Google (SSL)

Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options

Tesis Medico Yamira Martinez Gamez Log out

Actividades pendientes del usuario:
Yamira Martinez Gamez

Cargo	Lu	Ma	Mi	Ju	Vi	Opcion
ELECTROCARDIOGRAFIA DINAMICA (HOLTER)	f	x	a	m	x	
ESPIROMETRIA O CURVA DE FLUJO VOLUMEN PRE Y POST BRONCODILATADORES	x	m	x	a	m	
TERAPIA CARDIO PULMONAR	x	a	m	x	a	
DIPIRIDAMOLes	No Disponible	No Disponible	No Disponible	No Disponible	No Disponible	Aceptar

Figura 4.8: Listado de las acciones del medico.

Capítulo 5

Análisis y Pruebas

En este capítulo el AG va a trabajar con información que se ingresa para realizar pruebas de sus resultados, verificar si son correctos, es decir verificar que AG esta evolucionando y aplicando los operadores genéticos descritos anteriormente también se verifica que la función de aptitud esta evaluando los cromosomas correctamente si es así entonces este arrojará pruebas correctas. En caso que los resultado no sean correctos quiere decir que el AG esta presentando fallos. La información a ingresar se describe a continuación y se presenta en la Tabla: 5.1

Las entradas han sido creadas para verificar los resultados obtenidos del AG, si estos son correctos o se acercan a una posible solución.

En esta entrada se presenta la información de diez (10) médicos con sus horarios laborales y una cantidad de consultorios uniforme a la longitud de médicos, es decir se dispone de diez (10) consultorios.

En esta situación es fácil decir que cada médico le corresponde un consultorio y no es necesario utilizar el AG, pero por motivos académicos se utiliza el AG para ver su posible solución y tiempo de ejecución. En las siguientes pruebas se varía la cantidad los consultorios y de igual forma el AG genera su posible solución.

En las siguientes tablas (Tabla: 5.2 y Tabla: 5.3) se presentan dos tipos de resultados. La primera solución que se obtiene del AG utiliza el operador de cruce de emparejamiento parcial (PMX) y la segunda solución el Cruce Uniforme. Recordar que las matrices presentadas en las soluciones se explicaron en el Capítulo: 3.

5.0.1. Entrada de pruebas

Medico	Procedimientos	Ln	Ma	Mi	Ju	Vi
Medico 1	Procedimiento 1	x	x	x	f	f
Medico 1	Procedimiento 2	f	f	f	x	x
Medico 2	Procedimiento 2	f	x	a	m	x
Medico 2	Procedimiento 3	x	m	x	a	m
Medico 2	Procedimiento 8	x	a	m	x	a
Medico 3	Procedimiento 5	f	f	f	f	f
Medico 4	Procedimiento 3	m	a	m	x	f
Medico 4	Procedimiento 7	x	m	a	f	x
Medico 5	Procedimiento 8	f	f	f	a	a
Medico 5	Procedimiento 9	x	x	x	m	m

Cuadro 5.1: Entradas usadas para pruebas, 10 medicos.

Medico	Procedimientos	Ln	Ma	Mi	Ju	Vi
Medico 1	Procedimiento 1	2	4	1	4	2
Medico 1	Procedimiento 2	1	1	9	1	6
Medico 2	Procedimiento 2	7	10	6	6	8
Medico 2	Procedimiento 3	5	6	10	5	5
Medico 2	Procedimiento 4	3	3	3	10	7
Medico 3	Procedimiento 5	4	9	8	9	10
Medico 4	Procedimiento 6	8	7	4	2	9
Medico 4	Procedimiento 7	10	2	2	7	3
Medico 5	Procedimiento 8	9	8	5	8	4
Medico 5	Procedimiento 9	6	5	7	3	1

Cuadro 5.2: Salida AG con PMX, Tiempo de ejecucion 4.107818029 segundos.

Medico	Procedimientos	Ln	Ma	Mi	Ju	Vi
Medico 1	Procedimiento 1	3	6	2	6	6
Medico 1	Procedimiento 2	2	2	10	8	7
Medico 2	Procedimiento 2	9	7	1	2	1
Medico 2	Procedimiento 3	1	9	10	7	10
Medico 2	Procedimiento 4	4	3	9	10	4
Medico 3	Procedimiento 5	7	10	5	9	5
Medico 4	Procedimiento 6	10	7	7	7	9
Medico 4	Procedimiento 7	5	5	6	4	3
Medico 5	Procedimiento 8	8	8	3	5	8
Medico 6	Procedimiento 9	6	4	8	3	6

Cuadro 5.3: Salida AG con Cruce Uniforme, Tiempo de ejecucion 4.075858705 segundos.

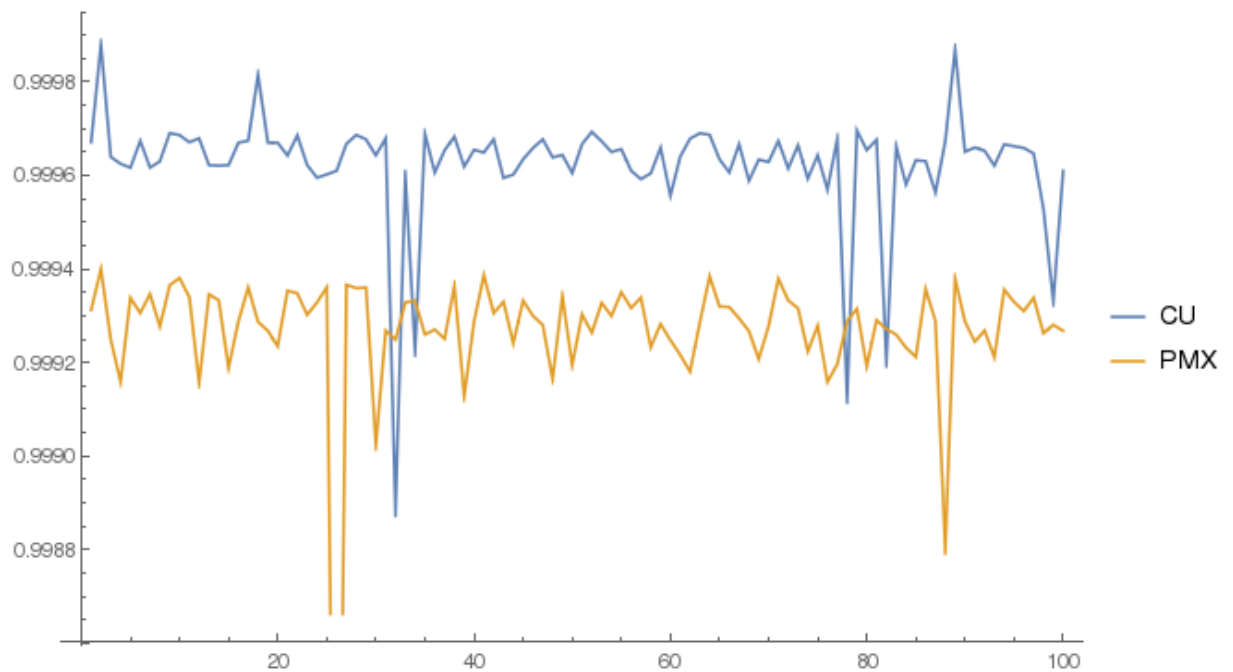


Figura 5.1: Lomgitud de la poblacion VS Tiempo ejecucion (seg) operador genetico (CU vs PMX).

Como se puede observar ambas salidas son correctas. En esta prueba no se busca demostrar que operador es mejor, el propósito es plantear que operador es

mas adecuado para el AG. Se puede decir que el operador PMX es apto cuando no se presentan valores repetidos como es en este caso o el problema de las n damas (donde ninguna dama puede colisionar con otra en un cuadrado $n \times n$), pero si, se quiere acercar a una posible solución este operados no es correcto usarlo, como se ha dicho el PMX presenta la restricción de no tener valores repetidos como se puede observar en la salida de las columnas de los días de la semana. Por otro lado se tiene el operador de Cruce Uniforme (CU), este presenta una salida correcta aunque puede llegar a tardar un poco mas o no, todo depende que tan rápido converge. La ventaja del operado CU es cuando se quiere acercar a una posible solución este lo ara. ejemplos.

Esto no garantiza que el AG sea mucho mas rápido. El cruce PMX no es adecuado usarlo si la cantidad de médicos es diferente a la cantidad de consultorios, para este caso es preferible implementar el Cruce Uniforme. En las siguientes pruebas se muestran los resultados de este caso.

Ejemplo 2.

La entrada que se presenta a continuación tiene una longitud de veinte (20) médicos y una longitud de diez (10) consultorios.

5.0.2. Entrada de pruebas

Medico	Procedimientos	Ln	Ma	Mi	Ju	Vi
Medico 1	Procedimiento 1	x	x	x	f	f
Medico 1	Procedimiento 2	f	f	f	x	x
Medico 2	Procedimiento 2	f	x	a	m	x
Medico 2	Procedimiento 3	x	m	x	a	m
Medico 2	Procedimiento 8	x	a	m	x	a
Medico 3	Procedimiento 5	f	f	f	f	f
Medico 4	Procedimiento 3	m	a	m	x	f
Medico 4	Procedimiento 7	x	m	a	f	x
Medico 5	Procedimiento 8	f	f	f	a	a
Medico 5	Procedimiento 9	x	x	x	m	m
Medico 1	Procedimiento 1	f	f	f	f	f
Medico 1	Procedimiento 2	m	m	m	m	m
Medico 2	Procedimiento 2	a	a	a	a	a
Medico 2	Procedimiento 3	m	a	f	x	f
Medico 2	Procedimiento 8	x	f	f	m	a
Medico 3	Procedimiento 5	a	a	f	f	m
Medico 4	Procedimiento 3	x	x	x	x	f
Medico 4	Procedimiento 7	m	m	a	a	a
Medico 5	Procedimiento 8	f	x	x	x	x
Medico 5	Procedimiento 9	m	m	m	m	a

Cuadro 5.4: Entradas usadas para pruebas, 20 medicos.

Las entradas han sido creadas para verificar la solución que mas se acerca a la mejor posible solución y tiempo de ejecución. La solución es la siguiente.

Medico	Procedimientos	Ln	Ma	Mi	Ju	Vi
Medico 1	Procedimiento 1	8	10	5	7	2
Medico 1	Procedimiento 2	10	8	2	5	9
Medico 2	Procedimiento 2	8	4	6	7	9
Medico 2	Procedimiento 3	1	3	4	1	6
Medico 2	Procedimiento 8	2	5	6	5	9
Medico 3	Procedimiento 5	8	5	4	1	7
Medico 4	Procedimiento 3	2	3	1	5	8
Medico 4	Procedimiento 7	3	7	4	2	7
Medico 5	Procedimiento 8	6	2	10	6	6
Medico 5	Procedimiento 9	4	4	7	8	4
Medico 1	Procedimiento 1	7	1	3	5	8
Medico 1	Procedimiento 2	9	3	8	8	9
Medico 2	Procedimiento 2	10	9	8	9	2
Medico 2	Procedimiento 3	9	2	5	8	5
Medico 2	Procedimiento 8	1	9	3	5	3
Medico 3	Procedimiento 5	6	8	1	1	8
Medico 4	Procedimiento 3	2	4	6	6	5
Medico 4	Procedimiento 7	9	1	9	5	2
Medico 5	Procedimiento 8	8	8	3	10	7
Medico 5	Procedimiento 9	4	10	8	10	1

Cuadro 5.5: Salida AG con Cruce Uniforme, Tiempo de ejecucion 5.464118657 segundos.

Medico	Procedimientos	Ln	Ma	Mi	Ju	Vi
Medico 1	Procedimiento 1	1	8	2	10	7
Medico 1	Procedimiento 2	10	3	6	1	6
Medico 2	Procedimiento 2	8	2	7	8	8
Medico 2	Procedimiento 3	9	7	1	9	4
Medico 2	Procedimiento 8	4	1	9	0	0
Medico 3	Procedimiento 5	0	0	0	4	0
Medico 4	Procedimiento 3	3	4	5	0	0
Medico 4	Procedimiento 7	6	0	3	3	0
Medico 5	Procedimiento 8	0	0	4	7	0
Medico 5	Procedimiento 9	7	9	0	0	10
Medico 1	Procedimiento 1	0	6	0	5	3
Medico 1	Procedimiento 2	0	0	10	6	0
Medico 2	Procedimiento 2	5	0	0	5	5
Medico 2	Procedimiento 3	0	5	8	5	1
Medico 2	Procedimiento 8	0	0	0	0	2
Medico 3	Procedimiento 5	2	0	0	0	9
Medico 4	Procedimiento 3	0	0	0	0	0
Medico 4	Procedimiento 7	0	10	0	2	0
Medico 5	Procedimiento 8	0	0	0	0	0
Medico 5	Procedimiento 9	0	0	0	0	0

Cuadro 5.6: Salida AG con PMX, Tiempo de ejecucion 6.140112786 segundos.

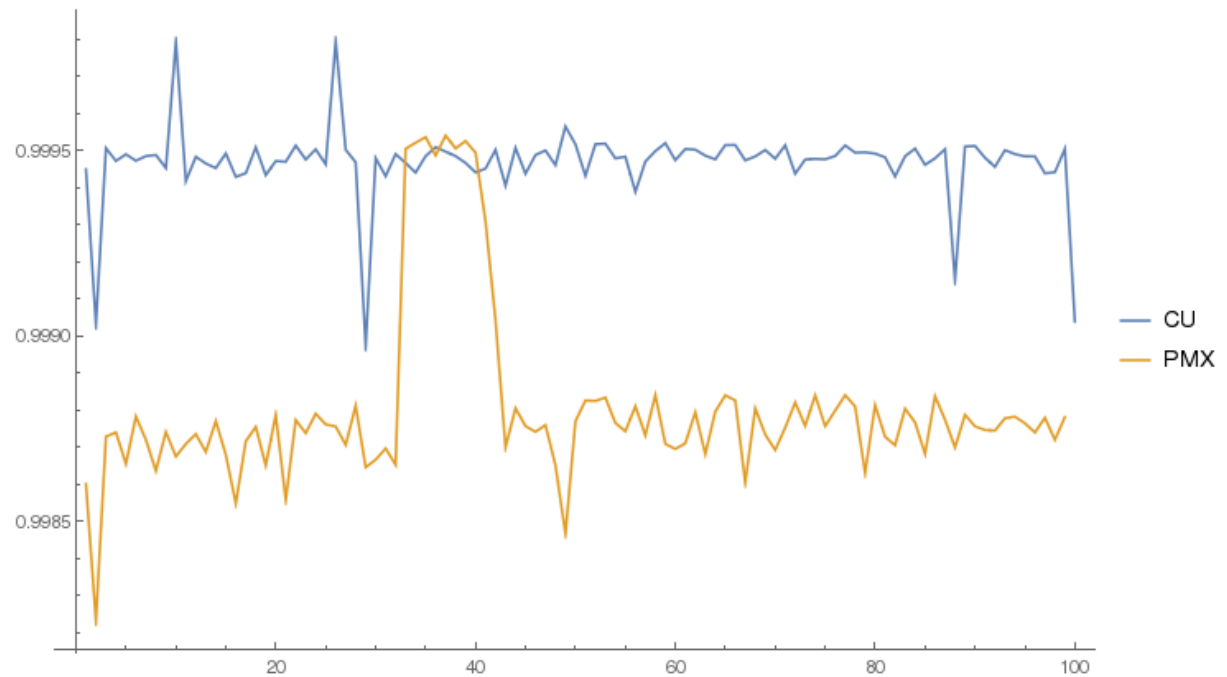


Figura 5.2: Lomgitud de la poblacion VS Tiempo ejecucion (seg) operador genetico (CU vs PMX).

Como se puede observar en las salidas de las tablas 5.5 y 5.6 se presentan los resultados dando uso de los operadores genéticos PMX y Cruce Uniforme. Como se menciono anteriormente se uso el cruce PMX por motivos académicos y se evidencia porque no es correcto el uso de este en los problemas de acercamiento el resultado, por otro lado se muestra como el CU tiene una aproximación al resultado.

En los resultados obtenidos se puede decir que el AG puede encontrar una solución correcta o aproximarse a esta, sin el uso del algoritmo este problema seria bastante tedioso para el usuario si quiere asignar los horarios a los médicos. Los problemas que presenta el AG y como se ha venido mencionando a lo largo del documento es que son bastante lentos y consumen muchos recursos de la maquina, también puede presentar salidas bastante cercanas al óptimo[14] aunque exista el optimo, esto quiere decir que el AG no garantiza un resultado 100 % correcto.

Capítulo 6

Conclusiones

- La implementación del Algoritmo Genético es útil para ver como un problema se puede acercar al resultado o llegar al resultado de ciertos problemas bastantes complejos casi imposibles de solucionar directamente.
- La implementación del Algoritmo Genético demostró ser una buena experiencia para estudiar las ventajas y desventajas que tiene un método de solución de este tipo de problemas frente a otro: además permite tener el usuario un punto de comparación sobre las otras posibles soluciones que pueden tomar una instancia del problema.
- La metodología del grupo EVALAB para aplicaciones evolutivas permitió seguir una serie de pasos, Permite desarrollar la aplicación en lenguaje ruby y su interfaz prototipo con el framework Ruby on Rails.
- El uso de los AG en la asignación de horarios permite acelerar las búsquedas de soluciones validas al problema frente a estrategias genéricas. Sin embargo estos algoritmos son más costosos en recursos, debido a que la implementación no considera aspectos de agilidad y de ahorro de recursos.
- Se pudo establecer que la solución implementada en este proyecto queda sujeta a algunas limitaciones computacionales con el uso del operador PMX, si la cantidad de médicos es superior a los consultorios y se considera una condición de parada donde el PMX encuentre el optimo, puede quedar en un bucle infinito y nunca encontrar la solución.
- Trabajar con la CE ayudo a fortalecer los conocimientos de la evolucion.

Capítulo 7

Trabajos Futuros

- El aplicativo desarrollado actualmente es muy limitado. Se encuentra funcional solo para el Centro Cardiológico del Valle, la idea es volver este aplicativo mas dinámico, pensando en que sea funcional en cualquier institución médica que necesiten asignar agendas a los médicos.
- La implementación de los Algoritmos Genéticos en el tema de la asignación es interesante para trabajar con las redes y estudiar el comportamiento que tiene la asignación, también se puede verificar el crecimiento que tienen las instituciones médicas con respecto a los pacientes que son atendidos.
- El desarrollo de la interfaz prototipo actualmente no tiene integrado el Algoritmo Genético estos en el momento se encuentran separados y utilizan la misma Base de Datos, la idea es integrar este al framework de Ruby on Rails como un plugin, así permite que el Algoritmo Genético sea más dinámico.
- Debido a que se necesita una gran capacidad de procesamiento en instancias grandes del problema, se recomienda realizar un estudio posterior sobre que cambios se deben realizar al modelo, a la estrategia de implementación y la forma de implementación, para que en la ejecución se pueda realizar una distribución del proceso en varios nodos de computo de tal forma que se pueda procesar una gran carga computacional sin depender de las capacidades de una infraestructura computacional que de soporte al aplicativo.
- Debido a que se necesita una gran capacidad de procesamiento, se recomienda hacer un estudio para implementar los Algoritmos Genéticos paralelos.

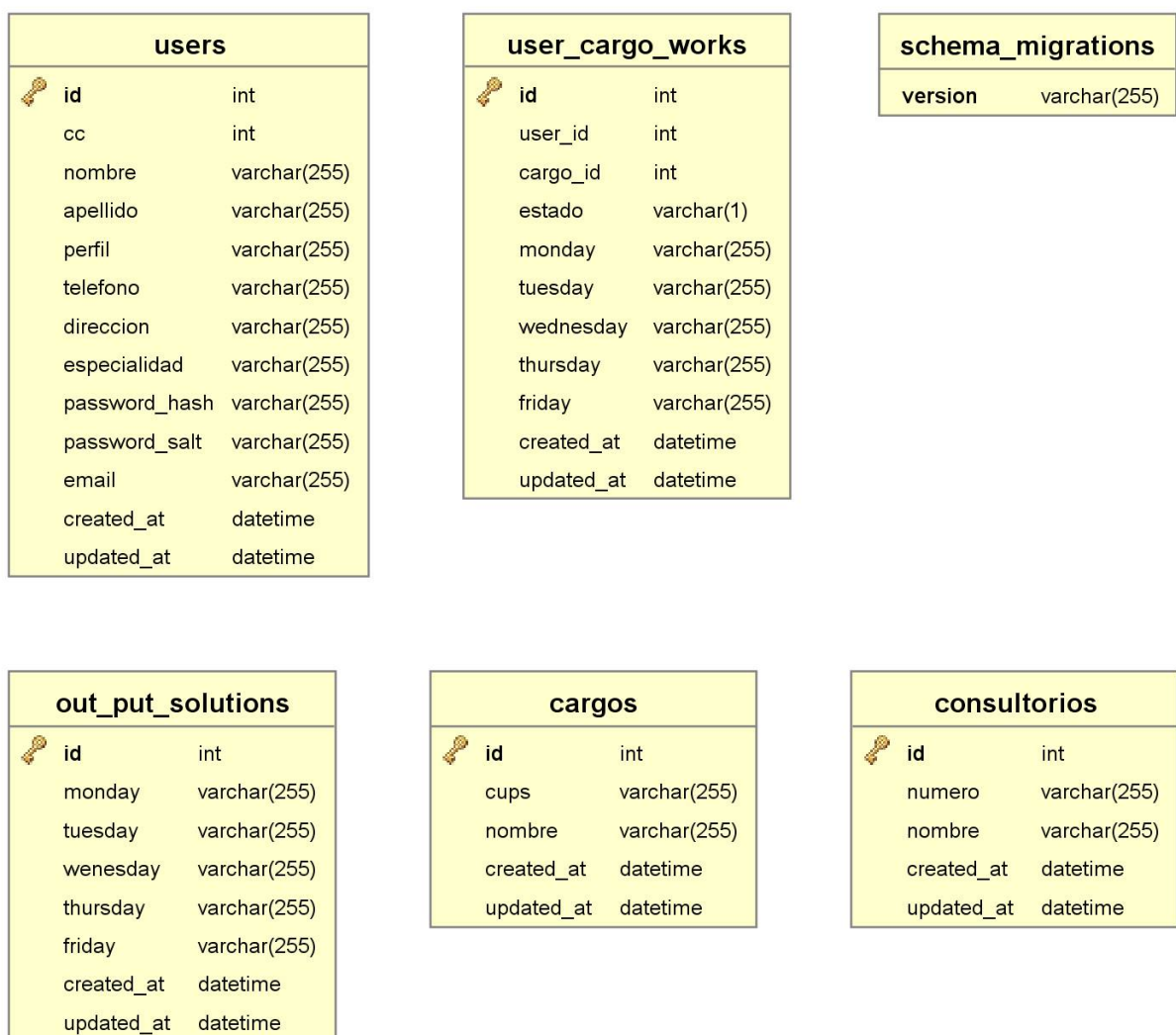
Bibliografía

- [1] Atai A. Even, SI and A Shamir. *On the complexity of timetable and multi-commodity flow problems*. SIAM Journal on Computing, 1976. 1
- [2] M.W. CARTER. *A Lagrangian Relaxation Approach to the Classroom Assignment Problem*. Computers and Operations Research, 1989. 1
- [3] J.A. Ferland and S. Roy. *Timetabling Problem for University as Assignment of Activities to Resources*. Computers and Operations Research, 1985. 1
- [4] J. Aubin and J.A. Ferland. *A large Scale Timetabling Problem*. Computers and Operations Research, 1989. 1
- [5] Piedad Tolmos Rodríguez Piñero. *Introducción a los algoritmos genéticos y sus aplicaciones*. Madrid 2009. 1
- [6] Álvaro Guerrero y Omar D. Castrillón Víctor F. Suárez. *Programación de horarios escolares basados en ritmos cognitivos usando un algoritmo genético de clasificación no-dominada, nsga-ii*. Colombia 2013. 1
- [7] Jazmín A. MONTEVERDE Norman F. SALAZAR José FIGUEROA Eli-seo CADENA Caleb A. LIZÁRRAGA. Pedro FLORES, Ernesto BRAU. *Experimentos con algoritmos genéticos para resolver un problema real de programación maestros-horarios-cursos*. Mexico 2009. 1
- [8] Carlos Cervigon Lourdes Araujo. *Algoritmos Evolutivos un enfoque practico*. Alfaomega, 2009. 6, 7, 8
- [9] Annicchiarico William. Algoritmos Genéticos Cerrolaza, Miguel. *Algoritmos de optimización estructural basados en simulación genética*. CDCH UCV, 1996. 7
- [10] T. Weise. *Global optimization algorithms, theory and application*. it-weise (self-published):. Germany 2009. 7

- [11] Alfonso Galipienso Colomina Pardo Lozano Ortega. Inteligencia Artificial. Escolano Ruiz, Cazorla Quevedo. *Modelos, Técnicas y Áreas de Aplicación*. España, Paraninfo, 2003. 11
- [12] Fernando MEJÍA. Artificial intelligence. 05 de marzo del 2014. 12
- [13] Alberto. Martínez de Abajo Nicolás Diez Pino, Raúl. Gómez Gómez. *Introducción a la IA. Sistemas Expertos, Redes Neuronales y Computación Evolutiva*. Universidad de Oviedo, 2001. 12
- [14] García Ruiz Rubén. Investigación Operativa Álvarez Maroto Concepción, Soria Alcaraz Javier. *Modelos y Técnicas de Optimización*. Editorial de la UPV, 2002. 35

anexos

Modelo Entidad Relacion



Powered by yFiles

Figura 7.1: Modelo de la base de datos que utiliza la interfaz

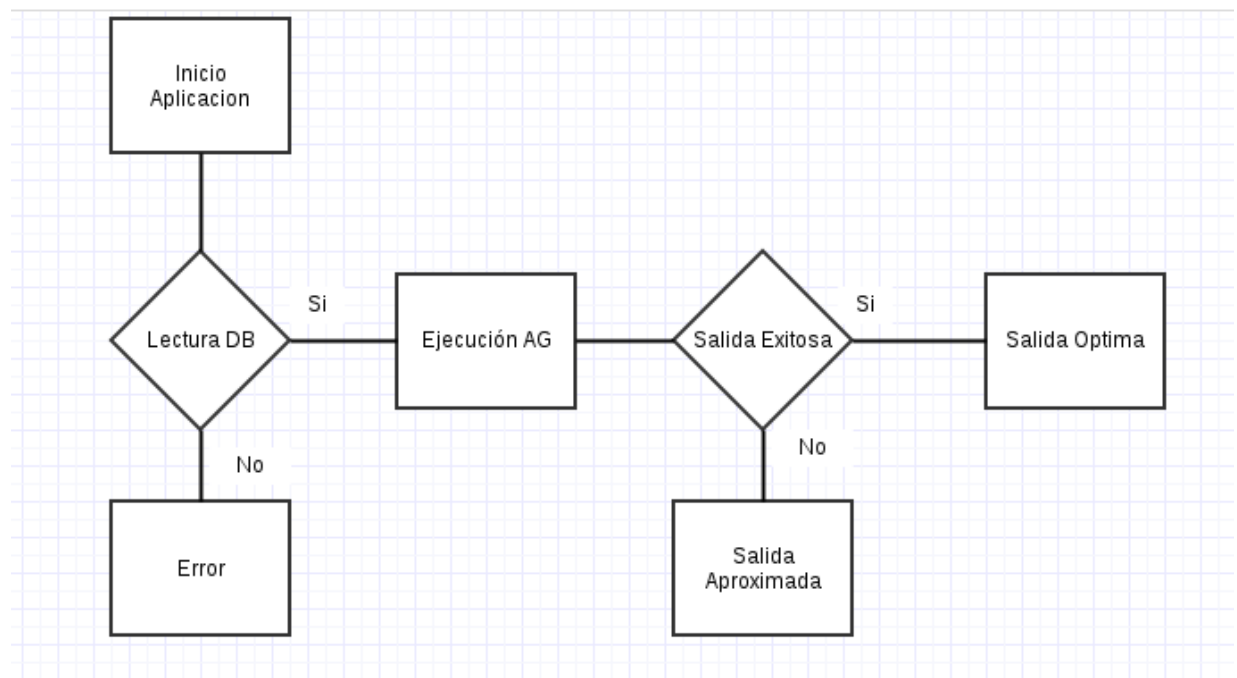
Diagrama De Flujo

Figura 7.2: Diagrama de flujo de la aplicación.