

**Diseño de un prototipo software que permita modelar los procesos de información entre
empresas distribuidoras de energía eléctrica y clientes**

**Fidel Herney Palacios Cuacialpud
Luis Carlos Arcila Soto**

**Universidad del Valle
Facultad de ingeniería
Escuela de ingeniería de sistemas y computación
Santiago de Cali
2014**

**Diseño de un prototipo software que permita modelar los procesos de información entre
empresas distribuidoras de energía eléctrica y clientes**

Fidel Herney Palacios Cuacialpud

Código 083261

herney400@gmail.com

Luis Carlos Arcila Soto

Código 0811111

tenchbunker@gmail.com

**Documento presentado como requisito parcial para la obtención de
grado de Ingeniero de Sistemas**

Director

Eduardo Francisco Caicedo Bravo, Ph.D.

eduardo.caicedo@correounivalle.edu.co

Codirector

Oswaldo Solarte Pabon , M.Sc.

oswaldo.solarte@correounivalle.edu.co

Universidad del Valle

Facultad de ingeniería

Escuela de ingeniería de sistemas y computación

Santiago de Cali

2014

Trabajo de grado presentado por
Fidel Herney Palacios Cuacialpud

Y

Luis Carlos Arcila Soto

Como requisito parcial para la obtención del título de Ingeniero de Sistemas

Eduardo Francisco Caicedo Bravo , Ph.D.
Director

Oswaldo Solarte Pabon, M.Sc.
Codirector

xxxx xxxx xxx, Ph.D.
Jurado

xxx xxxxx xxxx, Ph.D
Jurado

A Dios por darme la vida y la salud.

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

A mis profesores por su guía en mi aprendizaje.

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

Tabla de Contenido

1. Contexto y objetivos	1
1.1. Planteamiento del problema	1
1.2. Objetivos	1
1.2.1. Objetivo general	1
1.2.2. Objetivos específicos	1
1.3. Antecedentes	1
1.4. Justificación	3
1.4.1. Justificación teórica	3
1.4.2. Justificación metodológica	3
1.4.3. Justificación práctica	3
1.5. Información sobre los capítulos	3
2. Marco teórico	5
2.1. Modelo de agentes basado en redes neuronales artificiales (RNA)	6
2.2. Glosario	7
2.3. Redes neuronales	8
2.3.1. Introducción	8
2.3.2. Topologías de las redes neuronales	9
2.3.3. Mecanismos de aprendizaje	9
2.3.4. Algoritmo Backpropagation	9
2.3.5. Normalización de datos	10
2.4. Inteligencia de negocio	10
2.4.1. Problemas de satisfacción de restricciones.	10
2.4.2. Satisfacción de restricciones	10
2.4.3. Propagación de restricciones	11
2.4.4. Estrategias de búsqueda	11
2.4.5. Estrategias de distribución	13
3. Alcances de la propuesta	14
3.1. Alcances metodológicos	14
3.2. Alcances prácticos	14
4. Modelo de agente inteligente para predecir precio y consumo de energía eléctrica basado en redes neuronales	16
4.1. Anotaciones sobre el modelo	16
4.2. Datos de entrada	16
4.3. Variables de decisión	17

4.4.	Variables usadas en restricciones y estrategias de búsqueda	18
4.5.	Restricciones	18
4.5.1.	Restricciones triviales	18
4.5.2.	Restricciones co-canal	19
4.5.3.	Restricciones legales	19
4.6.	Cálculo de costos	19
4.6.1.	Número de cambios de asignación de canales para un operador	19
4.6.2.	Diferencia entre el número de canales y el mayor bloque de canales libres	20
4.6.3.	Número de canales inutilizables	20
4.6.4.	Costo total	20
4.7.	Justificación del modelo	20
4.7.1.	Uso de matrices y codificación binaria de dominios	20
4.7.2.	Tipos de restricciones	21
4.7.3.	Limitaciones de encontrar soluciones óptimas	21
5.	Consideraciones para implementación del modelo	22
5.1.	Formato de entradas	22
5.1.1.	Formato general	22
5.1.2.	Especificación de llaves	23
5.2.	Formato de salidas	23
5.2.1.	Formato general	24
5.2.2.	Campos de una solución	25
5.3.	Parámetros de las aplicaciones	26
5.4.	Información del cuadro nacional de atribución de frecuencias.	26
5.4.1.	Abstracción.	26
5.4.2.	Diseño de la base de datos.	27
5.4.3.	Datos de prueba en la aplicación prototipo.	27
6.	Implementación del modelo usando programación por restricciones	28
6.1.	Estrategias de distribución	28
6.2.	Estrategias de búsqueda	30
6.3.	Debilidad y flexibilidad de restricciones	30
6.3.1.	Restricciones flexibles	30
6.3.2.	Debilidad de restricciones	31
6.4.	Diseño e implementación del aplicativo	31
6.4.1.	Estructura de la aplicación	31
6.4.2.	Parámetros de la aplicación.	31
6.4.3.	Módulos del aplicativo	32
6.5.	Anotaciones sobre la implementación	34
6.5.1.	Número de variables de dominios finitos y propagadores	34
6.5.2.	Tamaño de entradas	34
6.5.3.	Errores conocidos del aplicativo	34
6.5.4.	Consideraciones para implementación usando programación por restricciones	34
7.	Implementación del modelo usando un algoritmo genético	35
7.1.	Diseño del algoritmo genético	35
7.1.1.	El algoritmo	35
7.1.2.	Problemas de modelado	37

7.1.3. Valores recomendados	37
7.2. Diseño e implementación de la aplicación	37
7.2.1. Parámetros de la aplicación	37
7.2.2. Formato entradas	38
7.2.3. Módulos	38
8. Interfaces Web de la aplicación	39
8.1. Estructura de la aplicación	39
8.1.1. Módulos de la aplicación	40
8.1.2. Integración con el portal de Avispa	42
8.2. Despliegue de información al usuario	44
8.2.1. Consultas	44
8.2.2. Archivos de entrada y salida	45
8.2.3. Resultados	46
9. Experimentación y pruebas	47
9.1. Descripción pruebas	47
9.1.1. Entradas de prueba	47
9.1.2. Procedimiento de pruebas	49
9.2. Pruebas sobre aplicativo basado en programación por restricciones	50
9.2.1. Pruebas sobre tamaño de entrada	50
9.2.2. Pruebas de acuerdo a la distribución de la asignación en la banda	52
9.2.3. Pruebas sobre estructura requerimientos	54
9.2.4. Pruebas sobre tamaño de requerimientos	56
9.2.5. Pruebas sobre tiempo de exploración en entradas grandes	57
9.2.6. Pruebas de acuerdo al nivel de recomputación	58
9.2.7. Pruebas de flexibilidad y debilidad de restricciones	59
9.3. Pruebas sobre el aplicativo basado en algoritmos genéticos	60
9.3.1. Pruebas sobre entradas varias	60
9.3.2. Pruebas sobre entradas que no cumplen restricciones	61
9.4. Resultados	61
9.4.1. Solución basada en programación por restricciones	61
9.4.2. Solución basada en algoritmos genéticos	62
9.4.3. Análisis comparativo entre los métodos de restricciones y evolutivo	63
9.4.4. Análisis de extensibilidad del modelo	63
9.4.5. Aplicación del modelo en situaciones reales	63
10. Conclusiones y trabajos futuros	64
10.1. Conclusiones	64
10.2. Trabajos futuros	66
11. Bibliografía	67
Anexos	69
Ejemplo entrada XML	69
Ejemplo salida XML	71
Diagrama de flujo aplicativo basado en programación por restricciones	72
Modelo entidad relación	73

Ejemplo de entrada algoritmo genético	74
Manuales de usuario	75

Lista de Figuras

2.1. Red neuronal biologica.	9
2.2. Estructura de un árbol de búsqueda	12
4.1. Perceptron multicapa.	16
8.1. Estructura de aplicación	40
8.2. Alerta sobre usuario no autenticado.	44
8.3. Alerta sobre usuario sin rol de administrador.	44
8.4. Ejemplo de consulta a base de datos.	45
8.5. Despliegue información de la entrada.	45
8.6. Ejemplo de consulta a base de datos.	46
8.7. Ejemplo de consulta a base de datos.	46
11.1. Diagrama de flujo del funcionamiento del aplicativo basado en programación por restricciones.	72
11.2. Modelo entidad relación del proyecto	73
11.3. Filtro en la consulta por zona geográfica.	75
11.4. Filtro en la consulta por operador.	75
11.5. Filtro en la consulta por banda.	75
11.6. Registrar operador	75
11.7. Editar operador.	75
11.8. Agregar servicio.	76
11.9. Editar servicio.	76
11.10 Selección banda de frecuencia.	76
11.11 Agregar banda de frecuencia.	76
11.12 Lista de rangos de frecuencia registrados en una banda.	77
11.13 Editar rango de frecuencia.	77
11.14 Lista de canales asociados a un rango de frecuencia.	77
11.15 Editar canal.	77
11.16 Lista de entradas XML almacenadas en el sistema.	77
11.17 Lista de salidas XML almacenadas en el sistema.	77
11.18 Especificación de requerimientos.	78
11.19 Selección salida XML para actualizar base de datos.	78
11.20 Selección entrada XML para actualizar base de datos.	78
11.21 Aplicativo por programación por restricciones.	78
11.22 Flexibilidad y debilidad de restricciones.	78
11.23 Aplicativo por algoritmo genético.	79
11.24 Información general acerca de una salida XML.	79
11.25 Información de costos y reporte de una solución obtenida de una salida XML.	79

Lista de tablas

1.1. Antecedentes	2
1.2. Estructura de capítulos	3
5.1. Estructura de llaves en las entradas	23
5.2. Campos de información de ejecución en general.	24
5.3. Campos de información de ejecución usando programación por restricciones.	25
5.4. Campos de costos de una solución.	25
5.5. Campos de información de un reporte de una solución específica.	25
5.6. Parámetros generales para la aplicación	26
5.7. Datos de prueba de la aplicación prototipo	27
6.1. Estrategias de distribución	29
6.2. Estrategias de búsqueda	30
6.3. Parámetros aplicación por restricciones	32
6.4. Funciones auxiliares	33
7.1. Valores recomendados para la aplicación basada el algoritmos genéticos	37
7.2. Parámetros para la aplicación basada el algoritmos genéticos	37
8.1. Consultas en el aplicativo prototipo de gestión del espectro	40
8.2. Operaciones de edición en la base de datos gestión del espectro	41
9.1. Entradas usadas para pruebas	47
9.2. Pruebas para E1 con 3seg de ejecución variando estrategia de distribución	50
9.3. Pruebas para E7 con 10 seg de ejecución variando estrategia de distribución	51
9.4. Pruebas para E11 con 10 seg de ejecución variando estrategia de distribución	51
9.5. Pruebas para E1 con 3 seg de ejecución variando el motor de búsqueda	52
9.6. Pruebas para E7 con 10 seg de ejecución variando el motor de búsqueda	52
9.7. Pruebas para E11 con 10 seg de ejecución variando el motor de búsqueda	52
9.8. Pruebas para E3 con 10 seg de ejecución variando estrategia de distribución	53
9.9. Pruebas para E4 con 10 seg de ejecución variando estrategia de distribución	53
9.10. Pruebas para E5 con 10 seg de ejecución variando estrategia de distribución	53
9.11. Pruebas para E6 con 10 seg de ejecución variando estrategia de distribución	53
9.12. Pruebas para E7 con 10 seg de ejecución variando estrategia de distribución	54
9.13. Pruebas para E8 con 10 seg de ejecución variando estrategia de distribución	54
9.14. Pruebas para E9 con 10 seg de ejecución variando estrategia de distribución	54
9.15. Pruebas para E10 con 10 seg de ejecución variando estrategia de distribución	55
9.16. Pruebas para E11 con 10 seg de ejecución variando estrategia de distribución	55

9.17. Pruebas para E12 con 10 seg de ejecución variando estrategia de distribución	55
9.18. Pruebas para E2 con 10 seg de ejecución variando estrategia de distribución	56
9.19. Pruebas para E12 con 10 seg de ejecución variando estrategia de distribución	56
9.20. Pruebas para E15 con 10 seg de ejecución variando estrategia de distribución	56
9.21. Pruebas para E13 con 5 seg de ejecución variando estrategia de distribución	57
9.22. Pruebas para E13 con 10 seg de ejecución variando estrategia de distribución	57
9.23. Pruebas para E13 con 20 seg de ejecución variando estrategia de distribución	57
9.24. Pruebas para E11 con 10 seg de ejecución con nivel de recomputación 1	58
9.25. Pruebas para E11 con 10 seg de ejecución con nivel de recomputación 2	58
9.26. Pruebas para E11 con 10 seg de ejecución con nivel de recomputación 4	58
9.27. Pruebas para E11 con 10 seg de ejecución con nivel de recomputación 6	58
9.28. Pruebas para E11 con 10 seg de ejecución con nivel de recomputación 8	59
9.29. Pruebas flexibilidad y debilidad para F1	59
9.30. Pruebas flexibilidad y debilidad para F2	59
9.31. Pruebas flexibilidad y debilidad para F3	60
9.32. Pruebas algoritmo genético para E6	60
9.33. Pruebas algoritmo genético para E8	60
9.34. Pruebas algoritmo genético para E15	60
9.35. Pruebas algoritmo genético para F1	61
9.36. Pruebas algoritmo genético para F2	61
9.37. Pruebas algoritmo genético para F3	61

Resumen

En el siguiente trabajo de grado se diseña, modela e implementa un prototipo de software para la predicción de precio y consumo de energía eléctrica en el mercado al retail, empleando técnicas de inteligencia artificial y minería de datos.

El diseño y modelado del prototipo de software se crea utilizando técnicas propias de la ingeniería de software como el **Lenguaje Unificado de Modelado** (UML) y la implementación se elabora en lenguaje de programación java.

Con el desarrollo del prototipo software se busca mejorar la toma de decisiones en el momento de asignar precio de energía eléctrica en el mercado al retail y así tanto usuarios como empresas, tenga un mejor soporte a la hora de adquirir y brindar respectivamente el servicio.

Introducción

El problema de la gestión del espectro consiste en buscar la mejor asignación posible de canales en una banda, para un grupo de operadores satisfaciendo restricciones que son impuestas por legislaciones, condiciones ambientales, estándares de tecnologías y regularizaciones existentes. Para determinar la mejor solución en éste trabajo de grado se ha definido algunos parámetros de costos por solución, los cuales son extrapolados de los análisis económicos que se han realizado a la gestión del espectro.

La fundamentación del problema, los objetivos del proyectos y los antecedentes, se describen en el capítulo 1. En el capítulo 2 se realiza una introducción a los conceptos teóricos necesarios para soportar la solución implementada en este proyecto de grado.

La implementación de la solución está basada en un modelo matemático lineal que se describe en el capítulo 4, el cuál se consigue al analizar las diferentes restricciones que se han elegido dentro de los alcances del proyecto que son definidos en el capítulo 3.

El proceso de implementación es descrito en los capítulos 5 y 6, donde el aplicativo del proyecto es implementado en lenguaje Mozart, debido a la facilidad de creación de estrategias de búsqueda, diseño de estrategias de distribución y algoritmos de búsqueda que posee de forma nativa. También se realiza un proceso de implementación utilizando un algoritmo genético que se describe en el capítulo 7, en este proceso se aprovechan algunos aspectos descritos en los capítulos anteriores.

En el capítulo 8 se explica el proceso creación del servicio Web, en el cuál se utiliza la metodología definida por el grupo Avispa[1] en donde se indica los pasos necesarios para convertir una aplicación de programación por restricciones en un servicio Web.

Para mostrar las ventajas de la programación por restricciones como método de solución del problema se realizan análisis sobre distintos escenarios que se pueden presentar en la gestión del espectro, esto se puede consultar en el capítulo 9.

Capítulo 1

Contexto y objetivos

1.1. Planteamiento del problema

¿Es posible a través de Sistemas Multi-Agentes caracterizar los procesos de información que se presentan en un mercado eléctrico competitivo al retail, para la construcción del precio de un Kwh, en el marco de las tecnologías Smart Grids?

1.2. Objetivos

1.2.1. Objetivo general

Modelar en un prototipo de software los procesos de información entre empresas distribuidoras de energía eléctrica y clientes, con el fin de caracterizar los diferentes factores que intervienen en la conformación de un precio competitivo en el mercado al retail.

1.2.2. Objetivos específicos

1. Establecer el nivel de desarrollo y utilización de los multi-agentes en el mercado de energía eléctrica al retail.
2. Identificar la información que se requiere en los procesos que se llevan a cabo entre las empresas y los usuarios para la creación del modelo.
3. Plantear un modelo que describa el funcionamiento del mercado de la energía eléctrica al retail utilizando multi-agentes.
4. Implementar un prototipo de software mediante multi-agentes, que represente el modelo del sistema.
5. Evaluar el prototipo con pruebas realizadas utilizando una base de datos de libre acceso.

1.3. Antecedentes

Tabla 1.1: Antecedentes

Aplicación	Tiempo Real	AG	Microredes	Gestión de redes	Fallos en redes	Reconfiguración de redes	Tipo

1.4. Justificación

1.4.1. Justificación teórica

El manejar información de los clientes de cualquier empresa, es una necesidad que con el tiempo acumula información que puede ser relevante para la toma de decisiones, dicha información puede ser tratada con técnicas de las ciencias de la computación como la minería de datos.

La predicción, es un problema que para el caso del mercado eléctrico es de gran importancia, ya que le permite adelantarse a sucesos, que si bien pudieron haber sucedido antes se puede volver a repetir en forma de ventaja o desventaja y es por ello que con técnicas de inteligencia artificial como agentes inteligentes basado en redes neuronales se pueden predecir basados en hechos históricos.

1.4.2. Justificación metodológica

La toma de decisiones que afecta a cualquier mercado, es la misma que afecta la oferta y la demanda en el mercado eléctrico al retail, siendo este un problema de inteligencia de negocio donde se requiere realizar un estudio de la información histórica para encontrar patrones de comportamiento invisibles a simple vista y que se puede analizar con técnicas de minería de datos para obtener información relevante, tanto para usuarios como para las empresas comercializadoras de energía eléctrica.

La predicción es también una herramienta importante para la toma de decisiones, para lo cual una buena técnica es crear agentes inteligentes basada en redes neuronales, capaces de aprender, generalizar y tomar decisiones a partir de datos históricos fusionando así la minería de datos y la inteligencia artificial.

1.4.3. Justificación práctica

En la actualidad en cualquier tipo de industria, es de vital importancia saber manejar la información tanto histórica como la que se vaya generando para estar un paso adelante ante hechos fortuitos, que afecten el normal funcionamiento de una producción.

Una correcta gestión y manejo de la información, permite tanto a usuarios como empresas comercializadoras de energía tomar decisiones basadas en hechos reales, obtenidos a través de técnicas que garanticen su integridad.

Una tecnología que avanza es la inteligencia artificial, si ha esta le encontramos utilidad con el uso de agentes inteligentes basado en redes neuronales para realizar predicciones en el mercado eléctrico, obtendremos una herramienta capaz de disminuir pérdidas que se verán reflejada en dinero y tiempo.

1.5. Información sobre los capítulos

En la siguiente tabla se muestra una corta descripción de cada capítulo del documento y se explica qué objetivos específicos aborda.

Tabla 1.2: Estructura de capítulos

Capítulo	Descripción	Objetivo específico
----------	-------------	---------------------

Capítulo 2: Marco teórico	Se explican los conceptos necesarios para comprender el problema y su solución, se espera que el lector tenga formación básica en física y matemáticas	Ninguno
Capítulo 3: Alcances de la propuesta	En este capítulo se definen los alcances metodológicos, teóricos y prácticos de la propuesta	Ninguno
Capítulo 4: Modelo de asignación de canales	Se presenta la propuesta de modelo para solucionar el problema de asignación de frecuencias y la estrategia para calcular costo de cada solución del problema.	1 y 4
Capítulo 5: Consideraciones para implementación	En este capítulo se muestran algunos aspectos que se han tenido en cuenta para el diseño e implementación de la aplicación.	2
Capítulo 6: Implementación del modelo usando programación por restricciones	El capítulo revela los aspectos que se deben tener en cuenta para la implementación del modelo.	3
Capítulo 7: Implementación del modelo usando un algoritmo genético	Se presenta una propuesta para la implementación del modelo usando un algoritmo genético y el modelo propuesto en el capítulo 5 a partir de un trabajo existente del problema de <i>channel assignment</i> .	Ninguno.
Capítulo 8: Interfaces Web de la aplicación	Se muestran los diferentes módulos de la aplicación final y como se despliegan los datos al usuario.	4
Capítulo 9: Análisis y resultados	En este capítulo se muestran los diferentes resultados comparativos obtenidos al realizar diferentes pruebas con la aplicación.	5
Capítulo 10: Conclusiones y trabajos futuros y pruebas	Las conclusiones finales del proyecto y los trabajos futuros que deben realizarse para mejorar la aplicación propuesta en éste trabajo de grado.	Ninguno.

Capítulo 2

Marco teórico

Para establecer los lineamientos del prototipo de software a realizar, es necesario definir los elementos teóricos con el fin de dar soporte al usuario respecto a conceptos y etapas de desarrollo. A continuación se da a conocer las definiciones y principales características que se tuvieron en cuenta para el desarrollo del proyecto de grado:

Es recomendado que el lector tenga formación básica en:

- Conceptos de minería de datos:
 - Descubrimiento de conocimiento en bases de datos.
 - Inteligencia de negocio.
 - extracción, transformación y carga de datos.
- Conceptos de redes neuronales:
 - Tipos de redes neuronales.
 - Tipos de aprendizaje (supervisado y no supervisado) y algoritmos de Entrenamiento.
 - Función de transferencia.
 - Reglas de aprendizaje.
 - Conceptos biológicos de lo qué es una red neuronal.
- Conocimiento sobre ingeniería de software.
- Conocimiento sobre diagramas Lenguaje Unificado de Modelado (**UML**) .
- Claridad en el concepto de modelo matemático.
- Información acerca de la diferencia entre aplicaciones de escritorio y Web.

A continuación se ilustran los términos más relevantes para el proyecto de grado respecto de agentes basado en redes neuronales artificiales (**RNA**).

2.1. Modelo de agentes basado en redes neuronales artificiales (RNA)

El modelado de agentes basado en Redes Neuronales constituye una metodología especialmente interesante. Debido a que la tarea a realizar tiene manejo de grandes cantidades de datos, se tuvo en cuenta para el diseño del agente, la metodología de redes neuronales, ya que estas pueden utilizarse por sus cualidades naturales para tratar con presencia de ruidos e incompletitud en los datos de entrada” Kohonen

Una red neuronal es capaz de particionar los datos de entrada en un conjunto de cluster basado en un criterio de similitud [Freeman4, Rasmussen5, Skapura6, Wasserman7], también han mostrado proveer modelos útiles para problemas que requieren clasificación en dos o más categorías, en el trabajo que nos compete la idea de utilizar redes neuronales como una serie de agentes inteligentes que en conjunto conforman un sistema integral de apoyo a la toma de decisiones.

- Entre las técnicas de minería de datos se encuentran las siguientes:
 - Redes neuronales artificiales.
 - Árboles de decisión.
 - Algoritmos genéticos.
 - Modelos lineales.
 - Vecino más cercano.

Dado que el núcleo del prototipo a implementar, necesita agentes inteligentes capaces de tomar decisiones partiendo de un conocimiento previo obtenido a través de datos históricos, se escogió de las anteriores técnicas las Redes Neuronales Artificiales (RNA) por su capacidad de detectar y aprender patrones y características de los datos, una vez entrenadas pueden hacer previsiones, clasificaciones y segmentación.

Si bien un conjunto de redes neuronales cumple con ciertas características, a continuación se expresan algunas que cumplen como agente inteligente:

- **Información contextual:** dado que el conocimiento está representado en la red neuronal. Cada neurona en la red está afectada por cualquier actividad global del resto de las neuronas.
- **Inferencia:** una red neuronal, a diferencia de los modelos conocidos, utiliza los datos para crear mapas entre las entradas y las salidas de la red.
- **Generalización:** el conjunto de datos de entrenamiento hace que las redes neuronales bien entrenadas pronostiquen de manera satisfactoria desde nuevos conjuntos de datos.
- **Paralelismo:** cada neurona conectada a través de la red tiene capacidad de procesamiento distribuido de la información contando también con memoria local en cada neurona.

El funcionamiento de un algoritmo de Red Neuronal Artificial básicamente consta de tres niveles:

- **Nivel de entrada:** las neuronas de entrada definen todos los valores de atributos de entrada para el modelo de minería de datos, así como sus probabilidades. [8]

- ● **Nivel oculto:** las neuronas ocultas reciben entradas de las neuronas de entrada y proporcionan salidas a las neuronas de salida. El nivel oculto es donde se asignan pesos a las distintas probabilidades de las entradas. Un peso describe la relevancia o importancia de una entrada determinada para la neurona oculta. Cuanto mayor sea el peso asignado a una entrada, más importante será el valor de dicha entrada. Los pesos pueden ser negativos, lo que significa que la entrada puede desactivar, en lugar de activar, un resultado concreto. [8]
- ● **Nivel de salida:** las neuronas de salida representan valores de atributo de predicción para el modelo de minería de datos. [8]

A continuación se muestra algunos ejemplos de agentes basado en redes neuronales.

QUERANDO! Un Agente de Filtrado de Documentos Web: dicho artículo “describe un agente de filtrado de páginas web llamado Querando! Capaz de aprender perfiles representativos de las preferencias del usuario, a través de una red neuronal basada en Resonancia Adaptativa Difusa”. [Gómez, Lanzarini 9].

GRAIL: A Multi-Agent Neural Network System for Gene Identification: sistema multiagente basado en redes neuronales como múltiples sensores para la identificación de genes **An artificial neural network based dynamic controller for a robot in a multi-agent system:** este artículo propone el modelamiento y simulación de una red neuronal computarizada para controlar la trayectoria de un robot en un sistema de fútbol de robots multiagente.

Control de un agente inteligente mediante Redes Neuronales en el entorno del videojuego UT2004: el objetivo de este trabajo es obtener agentes sintéticos para videojuegos de acción en primera persona de forma que su comportamiento no sea definido directamente por el programador, sino que estos sean capaces de adquirirlo mediante aprendizaje automático. Para ello se ha optado por una estrategia de aprendizaje basada en redes neuronales recurrentes de tiempo continuo. [Moreno 12] Multi-agent market modeling based on neural networks

2.2. Glosario

- **Sistema de Información** Un sistema de información se define como un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control de una organización:
 1. **Entrada de información:** Es el proceso mediante el cual el sistema toma los datos que requiere para procesar la información.
 2. **Almacenamiento de información:** El almacenamiento es una de las actividades o capacidades más importantes que tiene un computador, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior
 3. **Procesamiento de información:** Es la capacidad que tiene el sistema de información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida.
- **Lenguaje UML:** Lenguaje Unificado de Modelado, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group).
- **Base de Datos:** Uno de los objetivos de los sistemas de información es contar no solo con los recursos de información, sino también con los mecanismos para poder encontrar y recuperar estos

recursos. De esta manera, las bases de datos se han convertido en un elemento indispensable no solo para el funcionamiento de los grandes motores de búsqueda y la recuperación de información, sino también para la creación de sistemas de información en los que se precisa manejar grandes o pequeños volúmenes de información.

- **Web 2.0:** [8] La Web dos (punto) cero podría definirse como la promesa de una visión realizada: la Red convertida en un espacio social, con cabida para todos los agentes sociales, capaz de dar soporte y formar parte de una verdadera sociedad de la información, la comunicación y/o el conocimiento (Fumero y Roca, 2007).
- **Minería de Datos:** La minería de datos es un campo de las ciencias de la computación referido al proceso que intenta descubrir patrones en grandes volúmenes de conjuntos de datos.

2.3. Redes neuronales

2.3.1. Introducción

Las redes neuronales se pueden definir de distintas maneras:

- Como una forma de computación inspirada en modelos biológicos.
- Como modelos matemáticos que se componen de un gran número de elementos procesables organizados en niveles, elementos de procesos muy interconectados que procesan información por medio de su estado dinámico como respuesta a entradas externas [.
- Modelos matemáticos para el procesamiento de información. Redes interconectadas masivamente en paralelo de elementos simples (usualmente adaptativos) y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace un sistema nervioso [Eduardo Caicedo].
- Según Haykin(1994), una red neuronal es un procesador paralelo masivamente distribuido que tiene una facilidad natural para almacenar el conocimientos obtenido de la experiencia para luego hacerlos utilizable.

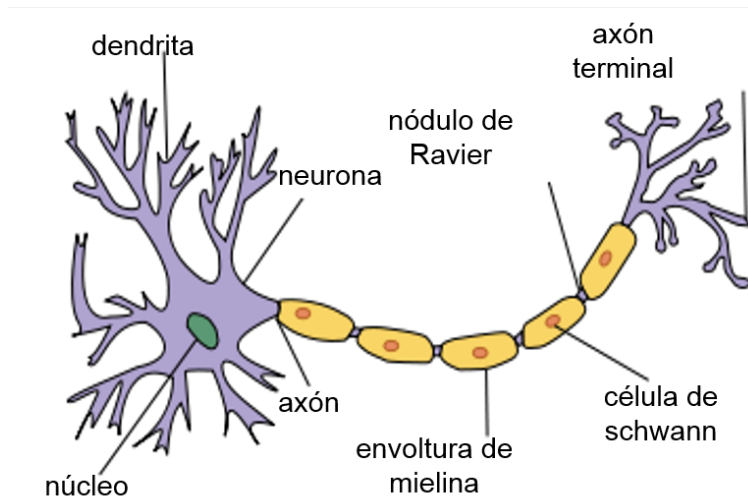


Figura 2.1: Red neuronal biológica.

Tomado de <http://slideplayer.es/slide/92022/>.

2.3.2. Topologías de las redes neuronales

la topología o arquitectura de una red neuronal consiste de la organización y disposición de la neuronas en la red neuronal, la cual esta íntimamente ligada con el algoritmo (reglas) de aprendizaje utilizado, entre ellas tenemos.

- **Redes monocapa:** En este tipo de red tenemos una capa de nodos de entrada desde el origen, que se proyecta en una capa de salida.
- **Redes multicapa:** la segunda clase de red neuronal se distingue por la presencia de una o mas capas ocultas, donde la neuronas son llamadas neuronas ocultas o unidades ocultas, siendo estas las que intervienen entre la entrada externa y la salida de la red de alguna manera útil.

2.3.3. Mecanismos de aprendizaje

- **Aprendizaje supervisado:** Este tipo de aprendizaje es conocido así debido a que en el proceso de aprendizaje se realiza mediante un entrenamiento, controlado por un supervisor, quien define la salida para una entrada específica.
- **Aprendizaje no supervisado:** Este tipo de aprendizaje esta diseñado para aquellas redes neuronales que no requieren influencia externa, así que la red no recibe ninguna información por parte del entorno para decidir si la respuesta es la correcta o no.

2.3.4. Algoritmo Backpropagation

Algoritmos mas popular para el entrenamiento de redes neuronales, ya que muestra un desempeño optimo.

2.3.5. Normalización de datos

Para que la red neuronal pueda calcular la salida se hace necesario tener datos numéricos, por lo tanto para aquellos datos que no lo son se hace necesario aplicar ciertas técnicas de normalización donde se encuentren valores apropiados para representar las características simbólicas, ejemplo (madrugada, mañana, tarde, noche).

2.4. Inteligencia de negocio

La programación por restricciones [20] es una herramienta poderosa para solucionar problemas combinatorios. La programación por restricciones utiliza técnicas de diferentes áreas, por ejemplo de inteligencia artificial, ciencias de la computación y lenguajes de programación. La programación por restricciones es actualmente aplicada para problemas de muchos dominios, como son la programación de tareas, planeación, búsqueda de caminos o rutas para llegar a un punto, redes de comunicación y bioinformática.

La idea básica en programación por restricciones es que el usuario escriba las restricciones y un solucionador de propósito general halle una solución. Las Restricciones son relaciones y un problema de satisfacción de restricciones (CSP) contiene las relaciones entre las variables de decisión. Los modelos de restricciones, son representados en términos de variables de decisión y restricciones, a partir de ellos, se debe encontrar una asignación de todas las variables que satisfaga las restricciones. Los solucionadores para programación por restricciones buscan sistemáticamente la solución en un espacio de búsqueda, con algoritmos de *Backtracking* o *Branch and Bound*. El método de búsqueda sistemático busca la interpolación, se conoce como la propagación de la información contenida en una restricción a otras restricciones, lo que reduce el espacio de búsqueda de valores para un conjunto de variables.

2.4.1. Problemas de satisfacción de restricciones.

Un problema de satisfacción de restricciones [21] se define así:

- Variables y_1, y_2, \dots, y_n
- Dominios D_1, D_2, \dots, D_n
- Restricciones $C : y_1 \in D_1, y_2 \in D_2, \dots, y_n \in D_n$

Una solución es:

$$(d_1, d_2, \dots, d_n) \in D_1 * D_2 * \dots * D_n$$

Donde se cumplen las restricciones C, es decir:

$$(d_1, d_2, \dots, d_n) \in C$$

2.4.2. Satisfacción de restricciones

La satisfacción de restricciones [20], es básicamente encontrar un valor para cada una de las variables del problema. Las restricciones podan los posibles valores de las variables; para conseguir una solución al problema se utilizan estrategias de búsqueda.

La estrategia de búsqueda fundamental es el *Backtracking* o vuelta hacia atrás, que consiste en realizar un recorrido en un árbol o grafo dirigido que no contiene ciclos. El objetivo es recorrer la estructura hasta encontrar soluciones, lo que se consigue obteniendo soluciones parciales a medida que progresa el recorrido; las soluciones parciales limitan las regiones donde se puede encontrar una solución completa. La búsqueda tiene éxito, si se puede encontrar una solución completa, el algoritmo puede detenerse o buscar soluciones alternativas. Si no encuentra una solución en un recorrido por una parte de la estructura del grafo dirigido o árbol, vuelve hacia atrás de forma similar a la de un recorrido por profundidad, eliminando los elementos que se hubieren añadido en cada fase.

2.4.3. Propagación de restricciones

La propagación de restricciones [21] es un concepto general, que consiste en encontrar relaciones entre restricciones usando algoritmos de filtrado, interferencia de restricciones, reglas de interacción e interacciones caóticas.

La propagación de restricciones embebe cualquier razonamiento que consista en prohibir cualquier valor o combinación de valores para algunas variables de un problema a causa de que un subgrupo de restricciones no pueden ser satisfechas.

Arco consistencia:

La arco consistencia es la más conocida de todas las formas de propagación de restricciones. Para que un CSP sea arco consistente se requiere que:

Para un conjunto de variables X e Y , se requiere que $\forall X \in \text{dom}(X) \exists Y \in \text{dom}(Y)$ tal que $r(X, Y)$ esta satisfecha y sea recíproca, es decir para todo $\forall Y \in \text{dom}(Y) \exists x \in \text{dom}(X)$ tal que $r(X, Y)$ esta satisfecha.

2.4.4. Estrategias de búsqueda

Una estrategia de búsqueda [21] consiste en un árbol de búsqueda y un algoritmo de búsqueda el cual explorar el árbol nodo a nodo para encontrar soluciones a un problema de programación por restricciones específico. Existen dos grandes estrategias de búsqueda la *Branch and Bound* y el *Backtrack*.

Árbol de búsqueda

Un árbol de búsqueda está representado de la siguiente manera

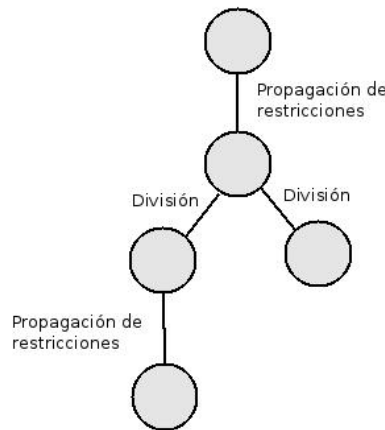


Figura 2.2: Estructura de un árbol de búsqueda

En la figura 2.2 las operaciones de propagación de restricciones indican que al aplicar una restricción a un dominio finito D éste se conserva o se reduce para obtener un dominio D' . El momento en que se realizan las operaciones de división, es cuando no se puede propagar más las restricciones y el resultado es la división del dominio D en n dominios D_1, \dots, D_n .

Algoritmo Backtrack

Este algoritmo construye una solución iniciando con una instancia vacía y sucesivamente intenta extender hasta obtener una instancia consistente definida por las siguientes variables. Si el procedimiento obtiene que la variable final obtiene una instanciación se encuentra una solución.

Bajo esta premisa se han elaborado muchas estrategias de búsqueda como son *Backtrack-free search with constraint propagation* y *Backtrack-free Search*.

Estas no son objeto de estudio, ya que en el proyecto se utiliza la estrategia *Branch and Bound* programada en *Mozart OZ*.

Branch and Bound

En este algoritmo existen dos grandes procedimientos:

- **Ramificación:** La expansión del árbol de búsqueda está condicionado por la búsqueda de la mejor solución, con esto se exploran todos los nodos de una rama hasta explorar en un nuevo nivel.
- **Poda:** El objetivo es eliminar aquellos nodos que no lleven a soluciones buenas, con esto se busca evitar expandirlos ya que sus hijos tampoco serán buenas soluciones.

Esta es una de las mejores estrategias para realizar búsquedas ya que evita expandir todo el árbol, para obtener ganancias en velocidad de procesamiento y uso de memoria.

Motores de búsqueda

Son algoritmos que permiten filtrar las soluciones encontradas por un algoritmo de búsqueda, por ejemplo seleccionar las soluciones que mejoran un costo determinado. Con éstos motores se busca encontrar la mejor solución a un problema de acuerdo a criterios de costo propios del problema.

Recomputación

Durante el proceso de expansión de un árbol de búsqueda[22] se almacena en memoria la información de cada nodo expandido; con esto se busca evitar que se calcule de nuevo un nodo si el algoritmo de búsqueda va hacia atrás. Un nivel de recomputación n indica que se almacenan los nodos que quedan a una distancia n de un nodo que se encuentra guardado. La recomputación es útil para árboles de búsqueda muy grandes donde se requiere una gran cantidad de memoria para su almacenamiento pero tiene un costo adicional de procesamiento al tener que calcular de nuevo los nodos no almacenados en memoria.

2.4.5. Estrategias de distribución

Una estrategia de distribución[23] define la forma y el contenido del árbol de búsqueda, cuántas alternativas existen en un nodo y qué restricción se agrega por cada alternativa. Las estrategias de distribución se utilizan para mejorar el rendimiento de los algoritmos de búsqueda, ya que una definición apropiada de un árbol de búsqueda permite guiar la exploración para encontrar más rápido la solución óptima.

Capítulo 3

Alcances de la propuesta

En este proyecto de grado se busca obtener una propuesta de solución al problema de gestión de los clientes que hacen uso de servicios de energía eléctrica usando agentes inteligentes basado en redes neuronales y minería de datos, con una aplicación prototipo en la que se puedan realizar desarrollos a futuro, por ello no se busca obtener un producto final.

3.1. Alcances metodológicos

El desarrollo de un prototipo software que permita modelar los procesos de información entre empresas distribuidoras de energía eléctrica y clientes para la construcción del precio de un **kwh**, en el marco de las tecnologías Smart Grid. En la aplicación prototipo se podrá tener un informe del consumo, precio y pagos históricos basado en información de la cual se obtendrá información relevante para la toma de decisiones en la construcción del precio de un **Kwh**.

La gestión de información se elaborara únicamente utilizando técnicas de minería de datos, entre las cuales se tiene en cuenta clientes industriales, residenciales.

En la practica se establecen normas que regulan el precio de la energía eléctrica, el cual depende de varios factores como la altitud, el estrato, el tipo de cliente (industrial, residencial), los cuales se han tenido en cuenta para el diseño del modelo que se implementara.

3.2. Alcances prácticos

Con respecto al proceso de modelar el proceso de información entre empresas distribuidoras de energía eléctrica y clientes se toma las siguientes consideraciones.

- Se trabaja con datos históricos tanto para la aplicación de técnicas de minería de datos.
- Las variables que intervienen en el precio de energía se ven reflejadas en el modelo de datos.
- Los clientes se dividen en residenciales, industriales y comerciales, para los clientes residenciales se pueden dividir por estrato.
- Las zonas de la ciudad se divide en comunas, barrios.
- el día esta dividido en franjas horarias (madrugada, mañana, tarde y noche).

Para modelar la predicción del precio de energía eléctrica mediante agentes basado en redes neuronales se tiene en cuenta las siguientes alcances prácticos.

- Se trabaja con datos históricos para el entrenamiento de las redes neuronales.
- Se trabaja con algoritmos de aprendizaje como **Backpropagation**.
- Las variables para modelar las entradas de la red neuronal son (clima, día ,altitud, estrato, franja horaria).

Capítulo 4

Modelo de agente inteligente para predecir precio y consumo de energía eléctrica basado en redes neuronales

4.1. Anotaciones sobre el modelo

La arquitectura de red utilizada en el diseño del agente inteligente es un **Perceptron Multicapa (MLP)** como se puede ver en la figura 4.1

- 0 si está libre.
- 1 si está asignado a un operador.

En la figura ?? se puede observar un ejemplo de una asignación para una banda con 9 canales y 3 operadores.

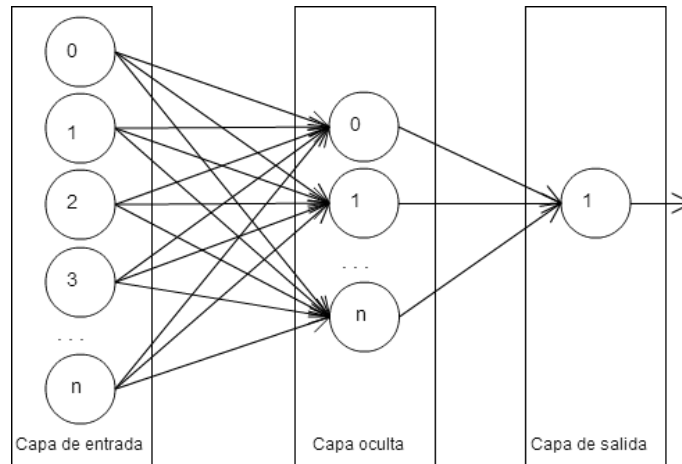


Figura 4.1: Perceptron multicapa.

4.2. Datos de entrada

Las variables de entrada representan el estado actual del modelo, en otras palabras la asignación actual del espectro en una banda de interés.

- C : Cardinalidad del conjunto de canales definidos para un servicio particular.
- O : Número de operadores en la banda que requieren asignación.
- N : Número de operadores presentes en la banda.
- $OPp = \{o_k : 1 \leq k \leq N\}$: Conjunto de etiquetas que representan a los operadores presentes.
- $OPi = \{o_k : 1 \leq k \leq O\}$: Conjunto de etiquetas que representan a los operadores que solicitan canales. Si $o_k \in OPp$, es un operador presente o antiguo; sino, es un operador nuevo.
- $OPt = OPp \cup OPi$: Son las etiquetas de los operadores presentes en la banda y de los que requieren asignación.
- $CI_c \in \{0, 1\}$: Canales marcados como inutilizables en la banda, si $CI_c = 1$ el canal c es inutilizable, en caso contrario se puede usar, donde $1 \leq c \leq C$.
- $CR_c \in \{0, 1\}$: Canales marcados como reservados en la banda, si $CR_c = 1$ el canal c está reservado, en caso contrario se puede usar, donde $1 \leq c \leq C$.
- $CP_c \in \{0, 1\}$: Canales asignados en las subdivisiones de la banda, si $CP_c = 1$ el canal c está asignado, en caso contrario se encuentra libre, donde $1 \leq c \leq C$.
- $B_{co}^I \in \{0, 1\}$: Indica si en el canal c de la banda se encuentra asignado el operador o donde $1 \leq c \leq C, o \in OPp$.
- $Req = [(o_1, nr_1), (o_2, nr_2), \dots, (o_O, nr_O)]$: (o, nr) indica que el operador etiquetado $o \in OPi$ requiere nr canales.
- Sep : Valor de separación mínima de canales entre canales asignados a operadores distintos.
- $Tope$: Valor máximo de canales que puede tener asignado un operador en una banda específica.
- AP_o : Indica el número máximo de canales que tiene el operador o en una subdivisión de la región de trabajo, donde $1 \leq o \leq O$.
- R : Número máximo de operadores por canal. Por defecto su valor es 1.
- $CAC \in \{0, 1\}$: Indica si se conserva asignación para operadores que solicitan nuevos canales y ya tienen asignación actualmente.

4.3. Variables de decisión

Las variables de decisión son aquellas que se desean encontrar en el problema, en este caso la asignación de canales a nivel nacional, regional, departamental o municipal.

- $B_{co}^O \in \{0, 1\}$ Asignación de los operadores en la banda, donde $1 \leq c \leq C, o \in OPt$

4.4. Variables usadas en restricciones y estrategias de búsqueda

Son aquellas variables que se utilizan para ayudar a definir algunas restricciones y las estrategias de búsqueda.

- $EC_c \in \{0, 1\}$ es una variable reificada que define si el canal c esta libre u ocupado.

$$(\forall c \in \{1, 2, \dots, C\}) \sum_{o \in OPt} B_{co}^O + CP_c < 1 \leftrightarrow EC_c = 1$$

El canal c se encuentra libre si y sólo si $EC_c = 1$

- $CAO_{co} \in \{0, 1\}$: Permite conocer el número de bloques contiguos asignados a un operador o , de la siguiente forma:

$$CAO_{co} = \begin{cases} 1 & \text{Si un bloque de canales asignados al operador etiquetado } o \text{ termina en el canal } c - 1 \\ 1 & \text{Si un bloque de canales asignados al operador etiquetado } o \text{ inicia en el canal } c \\ 0 & \text{En otro caso} \end{cases}$$

Si para un operador o se cumple $B_{co}^O \neq B_{(c+1)o}^O$ y $c < C$ entonces $CAO_{co} = 1$ en caso contrario $CAO_{co} = 0$, para el caso donde $c = C$ y $B_{co}^O = 1$ entonces $CAO_{co} = 1$ en caso contrario $CAO_{co} = 0$, donde: $1 \leq c \leq C, o \in OPt$.

- CLM_c : Define el tamaño de los bloques libres, si $c = 1$ entonces $CLM_c = EC_c$, si $c > 1$ y $EC_c = 1$ entonces $CLM_c = 1 + CLM_{c-1}$, si $c > 1$ y $EC_c = 0$ entonces $CLM_c = 0$, donde $1 \leq c \leq C$.
- $CLMmax = \max(CLM_c : 1 \leq c \leq C)$
- $CAI_c \in \{0, 1\}$: Permite conocer que canales se encuentran marcados como inutilizados debido a la separación mínima exigida entre dos operadores diferentes.

$$CAI_c = \begin{cases} 1 & \text{Si el canal se encuentra a una distancia entre 1 y } Sep \text{ de un canal asignado} \\ 0 & \text{En otro caso} \end{cases}$$

Para un canal c y un operador o si $B_{co}^O = 1$ y $B_{(c+1)o}^O = 0$, entonces $\forall c, 1 \leq i \leq Sep, CAI_{c+i} = 1$, en caso de que $B_{co}^O = 1$ y $B_{(c-1)o}^O = 0$ entonces $\forall c, 1 \leq i \leq Sep, CAI_{c-i} = 1$, si $B_{co}^O = B_{(c+1)o}^O$ entonces $CAI_c = 0$. En caso de que $R > 1$ o $Sep = 0$, entonces $\forall c \in \{1, 2, \dots, C\}, CAI_c = 0$

4.5. Restricciones

4.5.1. Restricciones triviales

Todo operador que solicita canales y actualmente tiene canales asignados los conserva a menos que $CAC \neq 1$

$$CAC = 1 \rightarrow \forall c \in \{1, 2, \dots, C\}, o \in OPp \cap OPi, B_{co}^I = 1 \rightarrow B_{co}^O = 1 \quad (4.1)$$

Todo operador que no solicita canales tendrá la misma asignación de canales en la salida

$$\forall c \in \{1, 2, \dots, C\}, o \in OPp \setminus OPi \quad B_{co}^I = B_{co}^O \quad (4.2)$$

Máximo R operadores por canal

$$\forall c \in \{1, 2, \dots, C\} \sum_{o \in OPt} B_{co}^O + CP_c \leq R \quad (4.3)$$

En el caso de que un canal esté reservado o marcado como inutilizable, no puede ser asignado a ningún operador que requiere asignación

$$\forall c \in \{1, 2, \dots, C\} CI_c + CR_c \geq 1 \rightarrow \sum_{o \in OPi} B_{co}^O = 0 \quad (4.4)$$

Todos los operadores que solicitan canales y no se encuentran actualmente con asignación, deben tener asignados un número de canales igual al que requieren.

$$\forall o \in OPi \setminus OPp, \sum_{c=1}^C B_{co}^O = Req_o \quad (4.5)$$

Todos los operadores que solicitan canales y actualmente tienen asignación, deben tener asignados un número de canales igual al que requieren más los canales que actualmente poseen en la banda.

$$\forall o \in OPi \cap OPp, \sum_{c=1}^C B_{co}^O = \sum_{c=1}^C B_{co}^I + Req_o \quad (4.6)$$

4.5.2. Restricciones co-canal

Debe existir una distancia de Sep entre dos canales asignados a operadores diferentes que solicitan asignación.

$$R = 1 \rightarrow (\forall o \in OPi, \forall o' \in OPt, \forall s \in \{1, 2, \dots, Sep\}, \forall c \in \{1, 2, \dots, C\}, o \neq o') \\ (B_{co}^O + B_{(c \pm s)o'}^O \leq 1) \quad (4.7)$$

4.5.3. Restricciones legales

Un operador no puede superar el tope de canales establecido por el gobierno en una banda de la región ni en sus divisiones territoriales

$$(\forall o \in OPi) \sum_{c=1}^C B_{co}^O + AP_{o_o} \leq Tope \quad (4.8)$$

4.6. Cálculo de costos

4.6.1. Número de cambios de asignación de canales para un operador

Los cambios en los canales asignados a un operador $Cost_1$, se calcula de acuerdo a la ecuación 4.9

$$Cost_1 = \lceil \frac{1}{2} * \sum_{o \in OPi} \sum_{c=1}^C CAO_{co} \rceil \quad (4.9)$$

Se toma como el número de bloques la mitad del cálculo realizado en CAO_{co} ya que este cuenta los límites de cada bloque, es decir que para un bloque se cuenta su inicio y su final, por lo que si hay n bloques se contarán $2n$ en CAO_{co} ; sin embargo en el caso de que el primer bloque inicia en la posición 1 de la banda, no se cuenta, por lo que se toma el techo de este cálculo, ya que en ese caso da un número impar.

Justificación

El número de cambios en la asignación de un operador representa costos adicionales en la práctica ya que debe adquirir equipos adicionales para trabajar en canales que se encuentran separados. Por lo tanto, es ideal dar a un operador un bloque contiguo en la asignación para reducir costos.

4.6.2. Diferencia entre el número de canales y el mayor bloque de canales libres

La diferencia entre el número de canales de una banda y el tamaño del mayor bloque libre $Cost_2$ se calcula de acuerdo a la ecuación 4.10.

$$Cost_2 = C - CLMmax \quad (4.10)$$

Justificación

Se busca que la asignación en una banda considere dejar un bloque de canales libres, para facilitar la asignación en futuros requerimientos, debido a que no se conoce cuantos canales se van a necesitar y se desea realizar la mejor asignación posible para esos requerimientos.

4.6.3. Número de canales inutilizables

Un canal libre es marcado como inutilizable cuando no puede ser asignado a ningún operador debido a la separación mínima exigida. En la ecuación 4.11 se muestra cómo calcular este costo.

$$Cost_3 = \sum_{c=1}^C CAI_c \quad (4.11)$$

Justificación

En la asignación se debe buscar que el número de canales inutilizados por la separación sea la mínima posible debido a que esto representa un desperdicio de canales que no pueden ser asignados.

4.6.4. Costo total

El costo total se calcula a partir de la suma de pesos entre 1 y 100 especificados por el usuario conocidos como $peso_1, peso_2, peso_3$ multiplicado por cada uno de los costos.

$$Cost_T = peso_1 * Cost_1 + peso_2 * Cost_2 + peso_3 * Cost_3 \quad (4.12)$$

4.7. Justificación del modelo

4.7.1. Uso de matrices y codificación binaria de dominios

El uso de matrices permite modelar con más precisión algunas restricciones, como es el caso de la separación y posibilita que el canal este asignado a más de un operador a diferencia del caso de una lista de canales.

El uso de dominios binarios permite modelar con mayor facilidad las restricciones de tope, separación entre operadores diferentes y de no asignar en canales marcados como asignados y reservados. Además hace posible el uso de marcadores para los canales como reservado, inutilizado y asignado en división sin hacer uso de artilugios propios de la implementación, como codificarlos como operadores virtuales.

4.7.2. Tipos de restricciones

Los tipos de restricciones han sido extraídos de la solución al problema de *channel assignment* [4] y del uso de topes por operador por parte del gobierno nacional colombiano [24].

Se definen los tipos de restricciones así:

- **Restricciones co-canal:** Son aquellas restricciones aplicadas a transmisores ubicados dentro de una pequeña zona, se asume en el modelo que un operador tiene transmisores en toda una zona geográfica, por lo tanto se debe garantizar una separación entre dos operadores diferentes.
- **Restricciones triviales:** Garantizan que se respete la asignación actual de los operadores, las asignaciones no se pueden mover en principio por los enormes costos que se generan en ese proceso.
- **Restricciones legales:** La única restricción que se toma es la de topes por operador, no se evalúa calidad de servicio ni potencia de emisiones, eso es objeto de estudio futuro.

4.7.3. Limitaciones de encontrar soluciones óptimas

Con el modelo se pueden obtener soluciones óptimas para la asignación en una zona determinada, pero no en cada una de las divisiones de esa zona. Por ejemplo, para el caso de los departamentos la asignación obtenida a nivel departamental cumplirá todas las restricciones y tendrá el mejor costo, pero no necesariamente es así para los municipios que componen el departamento ya que al no considerar el detalle, un operador puede quedar con una asignación con un peor costo que la solución óptima considerando cada uno de los municipios.

Esta decisión fue tomada debido al gran tamaño que tomaban las entradas al considerar el detalle de las divisiones territoriales, dificultaban su lectura y manejo en el aplicativo, además complica enormemente el modelado del problema ya que las restricciones deben considerar las asignaciones en cada una de las divisiones de la zona geográfica. En su lugar se ha optado por marcar los canales como asignados si alguna de las divisiones del territorio está asignado.

Capítulo 5

Consideraciones para implementación del modelo

Para realizar el proceso de implementación del proyecto se deben tomar en cuenta los pasos especificados en la metodología para aplicaciones Web del grupo Avispa [1].

En este capítulo se dan las pautas más relevantes para la implementación de la aplicación que son:

- Formatos de entradas y salidas.
- Parámetros de las aplicaciones.

Para facilitar la divulgación del proyecto, la aplicación se nombra cómo: **CaFeSA** que es la abreviación de *Constraints Application For Enhanced Spectrum Allocation*, en español, aplicación por restricciones para asignación de espectro aumentado.

5.1. Formato de entradas

A continuación se va definir el formato de las entradas para los aplicativos del proyecto, un ejemplo de entrada puede ser consultado en el anexo entradas de ejemplo ubicado en la página 69.

5.1.1. Formato general

Para las entradas se utiliza el formato XML usando la especificación *dict* de XCSP[25].

Para definir una entrada de acuerdo al formato *dict*, se debe asociar un orden convencional a un diccionario. Este orden convencional especifica un orden de llaves que pueden ser usadas para acortar notaciones. Mientras el grupo de llaves pueda ser conocido desde el contexto, los valores de cada llave pueden ser escritos en una notación abreviada, listando los valores de cada llave que son conocidas en el diccionario.

Por ejemplo, las coordenadas de un punto pueden ser representadas por un diccionario que contiene dos llaves *x* e *y* con valores (2,5) respectivamente.

```
<dict>
<entry key="x"><i>2</i></entry>
<entry key="y"><i>5</i></entry>
</dict>
```

5.1.2. Especificación de llaves

Para el proyecto se han definido las siguientes llaves:

Tabla 5.1: Estructura de llaves en las entradas

Llave	Descripción
GeograficAssignmentType	Tipo de asignación geográfica, 0 indica nacional, 1 territorial o regional, 2 departamental y 3 municipal.
GeograficAssignmentID	ID o llave de la entidad territorial, en el caso de asignación nacional este valor es 0.
FrequencyBand	ID o llave banda de frecuencia de trabajo.
FrequencyRank	ID o llave del rango de frecuencia o banda específica de trabajo.
NumberChannels	Número de canales que tiene la canalización del rango de frecuencia.
NumberPresentOperators	Número de operadores que tienen asignación actualmente en el rango de frecuencia.
NumberOfOperatorWithRequirements	Número de operadores que requieren asignación.
ChannelSeparation	Separación mínima de canales entre dos operadores diferentes.
PresentOperators	Lista de operadores que tienen asignación en la banda actualmente.
OperatorsWithRequeriments	Lista de operadores que requieren asignación.
Requeriments	Lista de los requerimientos en canales que tienen los operadores que requieren asignación.
MaxAssignationsSubDivision	Lista que indica el máximo número de canales que tiene asignado en una división del área geográfica cada operador que solicita asignación.
ChannelAssignInDivisions	Indica que canales están asignados en las divisiones del área geográfica.
ReservedChannels	Indica que canales están reservados en las divisiones del área geográfica.
DisabledChannels	Indica que canales están deshabilitados en las divisiones del área geográfica.
ChannelAssignment	Matriz que especifica la asignación actual para los operadores presentes.
MaxChannelAssignmentByOperator	Indica el máximo número de canales que puede tener asignado un operador en la banda.

5.2. Formato de salidas

A continuación se definen la estructura de salidas que se va a utilizar en el aplicativo, un ejemplo puede ser consultado en el anexo salida de ejemplo ubicado en la página 71.

5.2.1. Formato general

A partir de la metodología la estructura de la salida XML es la siguiente:

1. Raíz del documento:

```
<solutions>
</solutions>
```

2. Se establece un encabezado donde se encuentra la información de la ejecución de la aplicación:

```
<head>
</head>
```

3. Se estructura cada solución encontrada de la siguiente forma:

```
<solution id="n">
</solution>
```

n indica el número de la solución, estas se encuentran ordenadas de acuerdo al mejor costo.

4. Se colocan los costos de cada solución y el reporte de la salida.

```
<solution>
  <costs> ...
</costs>
  <report> ...
</report>
</solution>
```

Los campos de información de ejecución permiten evaluar el desempeño de la aplicación.

Tabla 5.2: Campos de información de ejecución en general.

Campo	Descripción
numSolutions	Número de soluciones encontradas.
memoryUsage	Uso del memoria del aplicativo.
executionTime	Tiempo de ejecución.

Los campos propios de la aplicación de programación por restricciones describen algunos aspectos de la ejecución usando programación por restricciones.

Tabla 5.3: Campos de información de ejecución usando programación por restricciones.

Campo	Descripción
considerTop	Indica si se considera la restricción del tope de canales en la ejecución.
staticAssignment	Especifica si la asignación de canales de los operadores que requieren asignación se mantiene.
considerSeparation	Muestra si en la ejecución se ha tomado en cuenta la separación de canales de operadores diferentes.
spacesCreated	Número de espacios computacionales creados en la ejecución.
spacesSucceeded	Número de espacios computacionales exitosos en la ejecución.
FDVariables	Número de variables de estado finito creadas en la ejecución.
propagators	Número de propagadores creados durante la ejecución.

5.2.2. Campos de una solución

En el reporte de una solución tiene dos campos de información; el primero que son los costos de la solución y el segundo que es el reporte de la solución.

Campos de los costos de una solución

Tabla 5.4: Campos de costos de una solución.

Campo	Descripción
blocksNumber	Número de bloques existentes en la solución.
difChannelNumberMaxBlockFree	Diferencia entre el tamaño del bloque libre más grande y el número de canales.
channelNumberUseless	Número de canales inutilizados por separación.
totalCost	Costo total de la solución, que consiste en sumar los anteriores costos multiplicados cada uno por un peso especificado por el usuario.
violations	Número de violaciones a las restricciones. Sólo aplica en algoritmo genético.

Campos de reporte de una solución

Tabla 5.5: Campos de información de un reporte de una solución específica.

Campo	Descripción
operator	Indica cual es la asignación del operador.
channels	Asignación final para un operador dado.

5.3. Parámetros de las aplicaciones

Los parámetros de las aplicaciones se especifican así:

Tabla 5.6: Parámetros generales para la aplicación

Parámetro	Descripción
– – <i>file</i>	Ruta del archivo de entrada.
– – <i>o</i>	Ruta archivo de salida.
– – <i>tm</i>	Tiempo de ejecución máximo de la aplicación.
– – <i>pnb</i>	Peso del costo número de bloques.
– – <i>psm</i>	Peso del costo diferencia entre número de canales y el tamaño del bloque más grande libre.
– – <i>pnc</i>	Peso del costo de número de canales inutilizados.
– – <i>ns</i>	número máximo de soluciones que se escriben en la salida

5.4. Información del cuadro nacional de atribución de frecuencias.

Para realizar la gestión del espectro se debe sistematizar el cuadro nacional de atribución de frecuencias [8]; en este proyecto se realizan los siguientes tareas para realizar éste proceso.

1. Abstracción del cuadro nacional de atribución de frecuencias tomando en cuenta los alcances del proyecto definidos en el capítulo 3.
2. Diseño de la base de datos de acuerdo a la abstracción del cuadro nacional de atribución de frecuencia.
3. Creación de datos de prueba para la aplicación prototipo.

5.4.1. Abstracción.

De acuerdo a los alcances del proyecto, se abstrae el cuadro nacional de atribución de frecuencias [8] bajo las siguientes consideraciones:

- Se utilizan los rangos de frecuencia definidos por servicios en el cuadro nacional de atribución de frecuencias, excepto para servicios marítimo móvil y fijo.
- Se emplean las grandes bandas de frecuencia HF, VHF, UHF, SHF, EHF para agrupar los diferentes rangos de frecuencia de acuerdo a su frecuencia inicial.
- La asignación es jerárquica territorialmente, es decir que una asignación nacional aplica en todo el territorio colombiano, una departamental en sus municipios y así sucesivamente.
- Para la división territorial se toma en cuenta las estaciones monitoras fijas del cuadro nacional de atribución de frecuencias y la organización territorial de Colombia [26] tomando la estructura de departamentos y municipios.

5.4.2. Diseño de la base de datos.

Para el diseño de la base de datos se toma en cuenta lo siguiente:

- Los rangos de frecuencias son elementos de los grandes bandas de frecuencia. Cada rango tiene autorizado uno o más servicios.
- Los canales son elementos de los rangos de frecuencia.
- Una asignación es una entidad especializada por tipo: nacional, territorial, departamental y municipal.
- Para determinar una asignación en una entidad territorial debe tomarse en cuenta las asignaciones en las entidades a las cuales ésta pertenece hasta llegar a las asignaciones nacionales.
- Los operadores prestan uno o más servicios.

De acuerdo a lo anterior, se define el modelo entidad relación de la base de datos, el cual puede se consultando en el anexo modelo entidad relación en la página 73.

5.4.3. Datos de prueba en la aplicación prototipo.

En la tabla 5.7 se muestran los diferentes datos que se encuentran la base de datos y su fuente, es de anotar que algunos son aleatorios, por la imposibilidad de obtenerlos de la ANE.

Tabla 5.7: Datos de prueba de la aplicación prototipo

Dato	Fuente
Bandas de frecuencia	Cuadro nacional de atribución de bandas de frecuencias página 47.
Rangos de frecuencia	Cuadro nacional de atribución de bandas de frecuencias páginas 246-253,272-275 y 279-418.
Servicios	Cuadro nacional de atribución de bandas de frecuencias páginas 23-27.
Servicios por rango de frecuencia	Cuadro nacional de atribución de bandas de frecuencias páginas 51-141.
Operadores	Operadores concodios que utilizan el espectro conocidos en Colombia que prestan servicios en el espectro.
Servicios por operador	Ninguna.
Tope por operador en rango de frecuencia	Ninguna.
Separación en rango de frecuencia	Ninguna.
División territorial o regiones	Para las regiones se utiliza la definición del cuadro nacional de atribución de bandas de frecuencias en el anexo II estaciones monitoras fijas.
Lista de departamentos y municipios	Biblioteca virtual del Banco de la República de Colombia [27], actualizada al 2005.
Asignación actual	Ninguna.

Capítulo 6

Implementación del modelo usando programación por restricciones

En este capítulo se muestran los aspectos más relevantes del proceso de implementación. Se explican qué estrategias de búsqueda y distribución se usaron, la estructura del aplicativo y se realizan algunos apuntes con respecto al proceso de implementación que deben ser tenidos en cuenta al momento de utilizar la aplicación.

Las estrategias de búsqueda y distribución son propias del proceso de programación por restricciones, donde se busca mejorar el proceso de ejecución del aplicativo para encontrar las soluciones óptimas más rápidamente.

La flexibilidad y debilidad en las restricciones permite encontrar soluciones en instancias del problema que no cumplen las restricciones. A lo largo de éste capítulo se describe cómo se implementaron.

6.1. Estrategias de distribución

Las estrategias de distribución están enfocadas a mejorar el rendimiento del algoritmo de búsqueda y están diseñadas para agilizar el proceso de gestión del espectro a partir de conocer las características propias de una banda específica. Por ejemplo, si esta se encuentra ocupada al inicio, lo más recomendable es intentar asignar al inicio, ya que la mejor solución de acuerdo a los costos definidos en el modelo buscan que los bloques libres sean lo más grandes posibles.

Para definir las estrategias de distribución se crean las siguientes variables auxiliares:

- *Req* es la lista de requerimientos, es una lista de tuplas que contiene los elementos (o_i, nr_i) donde o_i es el operador i y nr_i es el número de canales requeridos por el operador i .
- *ReqA*: Es una sublista de *req* que contiene los requerimientos correspondientes a los operadores que ya tienen asignación en la banda. Es decir que $o_i \in OP_i \cap OPp$.
- *NReqA*: Es una sublista de *req* que contiene los requerimientos correspondientes a los operadores que no tienen asignación en la banda. Es decir que $o_i \in OP_i \setminus OPp$.
- *MayorReq*: es la lista de requerimientos ordenada de mayor a menor de acuerdo al número de canales solicitados por cada operador.

- *MenorReq*: es la lista de requerimientos ordenada de menor a mayor de acuerdo al número de canales solicitados por cada operador.

En la tabla 6.1 se especifican los diferentes estrategias de distribución usados en el proyecto.

Tabla 6.1: Estrategias de distribución

Estrategia	Descripción
Asignar primero al inicio de la banda	Se toman los requerimientos <i>Req</i> en orden de llegada, se toma el primer requerimiento y se intenta asignar en el primer canal de la banda, si no es posible en el segundo y así sucesivamente; una vez se ha cumplido el primer requerimiento se prosigue con el segundo hasta tomar todos los requerimientos.
Asignar primero al final de la banda	Se toman los requerimientos <i>Req</i> en orden de llegada, se toma el primer requerimiento y se intenta asignar en el último canal de la banda, si no es posible en el penúltimo y así sucesivamente; una vez se ha cumplido el primer requerimiento se prosigue con el segundo hasta tomar todos los requerimientos.
Asignar primero a operadores en orden de llegada con asignación al inicio de la banda	Se intenta asignar canales al inicio de la banda a la lista de requerimientos <i>ReqA</i> y luego a <i>NReqA</i> .
Asignar primero a operadores en orden de llegada con asignación al final de la banda	En este caso se asignar canales al final de la banda a la lista de requerimientos <i>ReqA</i> y luego a <i>NReqA</i> .
Asignar primero a operadores sin asignación al inicio de la banda	En este caso se asignar canales al inicio de la banda a la lista de requerimientos <i>NReqA</i> y luego a <i>ReqA</i> .
Asignar primero a operadores sin asignación al final de la banda	En este caso se asignar canales al final de la banda a la lista de requerimientos <i>NReqA</i> y luego a <i>ReqA</i> .
Asignar primero a requerimientos más grandes al inicio de la banda	A partir de <i>MayorReq</i> se realiza el proceso de asignación intentado asignar a partir del inicio de la banda.
Asignar primero a requerimientos más grandes al final de la banda	A partir de <i>MayorReq</i> se realiza el proceso de asignación intentado asignar a partir del final de la banda.
Asignar primero a requerimientos más pequeños al inicio de la banda	A partir de <i>MenorReq</i> se realiza el proceso de asignación intentado asignar a partir del inicio de la banda.

Asignar primero a requerimientos más pequeños al final de la banda	A partir de <i>MenorReq</i> se realiza el proceso de asignación intentado asignar a partir del final de la banda.
Estrategia genérica: ff	Intenta asignar la primera variable de la lista, donde el domino es más pequeño, por el menor valor del dominio es decir 0.
Estrategia genérica: naive	Intenta asignar la primera variable de la lista, por el menor valor del dominio es decir 0.
Estrategia genérica: split	Intenta asignar la primera variable de la lista, por el valor medio del dominio es decir 1.

6.2. Estrategias de búsqueda

Las estrategias de búsqueda utilizadas en el aplicativo se muestran en la tabla 6.2. La función objetivo en los costos consiste en encontrar la solución con el costo mínimo.

Tabla 6.2: Estrategias de búsqueda

Estrategia	Función
Buscar mejor solución minimizando el número de cambios de asignación de canales para un operador	$\min(cost_1)$
Buscar mejor solución minimizando la diferencia entre el número de canales y el mayor bloque de canales libres	$\min(cost_2)$
Buscar mejor solución minimizando el número de canales inútiles	$\min(cost_3)$
Buscar por mejor costo total	$\min(cost_T)$
Ninguno	No hay criterio de selección para la mejor solución

6.3. Debilidad y flexibilidad de restricciones

Existen algunas instancias del problema que no cumplen una o más restricciones, por lo tanto, se diseñan estrategias de flexibilidad y debilidad de restricciones para realizar su procesamiento.

6.3.1. Restricciones flexibles

La flexibilidad en las restricciones consiste en que algunas entradas del problema toman valores que deshabilitan el efecto de una o más restricciones del problema.

No considerar tope

En este caso $Tope = C$, se fuerza a que el tope máximo de cada operador sea el número de canales de la banda.

No considerar separación entre canales

En este caso $Sep = 0$, es decir que la separación entre canales de operadores diferentes es cero.

Frecuencias asignadas a operadores con requerimientos se pueden mover

Se toma $CAC = 0$, lo que evita que se conserva la asignación de los operadores que solicitan canales.

Más de un operador por canal

Se cumple $R > 1$, en este caso también $Sep = 0$, ya que no tiene sentido considerar separación en un caso donde varios operadores pueden compartir un canal.

6.3.2. Debilidad de restricciones

La estrategia utilizada en el proyecto para manejar fortaleza y debilidad de las restricciones, consiste en permitir al usuario elegir que restricciones son flexibles, en seis posibles iteraciones.

Cada iteración es independiente de las otras. El usuario define qué restricciones flexibiliza en cada una de ellas y a partir de los resultados de cada una elige cuál tomar como solución final.

Las iteraciones en el aplicativo se ejecutan al mismo tiempo debido a que son independientes entre sí y los resultados de cada una son desplegados al usuario.

6.4. Diseño e implementación del aplicativo

6.4.1. Estructura de la aplicación

La aplicación se diseña de acuerdo a la siguiente arquitectura:

- **Vista:** Son las interfaces Web que tiene la aplicación. Para mayor información consultar capítulo 8. Esta en el proyecto es provista por el sistema de gestión de contenido Drupal[28].
- **Comunicación:** Provee las funciones necesarias para comunicar las capas vista y ejecución. En el proyecto son las funciones escritas en diferentes archivos auxiliares usando lenguajes *PHP* y *JavaScript* que permiten tomar la información provista por el usuario y procesarla.
- **Ejecución:** Son los algoritmos y funciones que toman los datos provistos por la capa de comunicación y envía información para ser desplegada al usuario. En el proyecto es básicamente los programas escritos en *Mozart OZ*[29] para la aplicación por programación por restricciones y *C++* para el caso de la aplicación por algoritmos genéticos.
- **Persistencia:** Está compuesta por la base de datos y la estructura de archivos en el servidor donde se almacena información de forma consistente. En el proyecto está compuesta por los archivos de entrada en formato XML, archivos de salida en formato XML y la base de datos que abstrae el cuadro nacional de atribución de frecuencias.

6.4.2. Parámetros de la aplicación.

Los parámetros propios de la aplicación usando programación por restricciones están definidos en la tabla 6.3.

Tabla 6.3: Parámetros aplicación por restricciones

Parámetro	Función
– – <i>motor</i>	Seleccionar motor de búsqueda, número entero entre 1 y 10
– – <i>es</i>	Elegir estrategia de distribución.
– – <i>rc</i>	Nivel de recomputación.
– – <i>nmas</i>	Campo binario, si es 1 no se mantienen asignaciones actuales de operadores que solicitan asignación, en caso contrario las mantiene.
– – <i>ntope</i>	Campo binario, si es 1 se flexibiliza la restricción de tope por operador.
– – <i>nsep</i>	Campo binario, si es 1 no se considera la separación mínima de canales entre operadores diferente.
– – <i>nopc</i>	Campo entero positivo mayor que cero, indica el número de operadores por canal.

6.4.3. Módulos del aplicativo

La aplicación cuenta con diferentes módulos con funciones específicas para realizar los procesos de:

- Lectura de entrada XML.
- Imposición de restricciones.
- Implementación de estrategias de búsqueda.
- Implementación de estrategias de distribución.
- Generación de salidas XML.

Aplicación.

En la figura ubicada en el anexo diagrama de flujo en la página 72 se describe el proceso de ejecución del aplicativo del proyecto basado en programación por restricciones y escrito en lenguaje *Mozart OZ*.

Conversor de entradas.

Para la conversión de entradas se hace uso del *Parser* conocido como *XML Parser*, que permite tomar una estructura de archivo en formato XML y transformarlo en un árbol de etiquetas que facilita la extracción de la información.

Restricciones.

Las restricciones son escritas usando el módulo de dominios finitos de *Mozart OZ* conocido como *FD*. Se hace uso de condicionales para imponer la flexibilidad de restricciones.

Cálculo de costos.

Para el cálculo de costos se hace uso de las funciones reificadas del módulo *FD*. Es importante aclarar que una restricción reificada presenta una salida binaria: 0 si no se cumple o 1 si se cumple; además no es una restricción que se imponga en la ejecución, es decir sí ésta falla la ejecución continúa sin problema.

En el proyecto se encontraron estas restricciones especialmente útiles para calcular los costos de cada solución.

Funciones auxiliares.

Existen algunas funciones auxiliares que se mencionan brevemente en la tabla 6.4.

Tabla 6.4: Funciones auxiliares

Función	Descripción
Trasponer Salida	Permite trasponer la matriz de la variable de decisión para efectos de facilitar la escritura de algunas restricciones
Cálculo de variables auxiliares	Permite el cálculo de variables como es el caso de <i>ECC</i> .
Sumar estructura	Permite calcular la suma de una fila de una matriz o bien de una lista. Muy útil para el cálculo de costos.

Estrategias de búsqueda.

Las diferentes estrategias de búsqueda hacen uso de la posibilidad que provee *Mozart* se comparar dos soluciones *A* y *B*, eligiendo cual es la mejor.

En el caso del aplicativo, se encuentra inicialmente una solución y se compara ésta hasta que se encuentre una mejor; una vez se ha encontrado una mejor ésta se compara con las que se encuentren más adelante.

Distribuidores

Los distribuidores hacen uso del entorno $\{Space.waitStable\}$ que permite esperar la ejecución hasta que el espacio de búsqueda es estable, es decir en el momento que se puede empezar a asignar valores o dominios a las diferentes variables.

Para realizar el proceso de cada uno de los distribuidores que han sido especificadas en la tabla 6.1 se hace uso de las restricciones reflejadas, que permiten imponer restricciones con respecto a la propagación de las variables. Por ejemplo, si el dominio tiene cierto tamaño o qué tantas variables han sido asignadas en una estructura de datos; estas restricciones presentan una salida binaria, 0 si se cumple o 1 si no se cumple. Gracias al uso de restricciones reificadas se puede hacer uso de ciclos, filtros y condicionales sin problema.

Generador de salidas XML

La salida se construye a partir de la información provista por la salida del proyecto que se encuentra en estructuras de datos propias de *Mozart OZ* usando una librería conocida como *VirtualString*. La construcción de la salida se realiza de acuerdo a lo especificado en la sección 5.2 del presente documento.

6.5. Anotaciones sobre la implementación

6.5.1. Número de variables de dominios finitos y propagadores

En la ejecución del aplicativo se da un gran número de variables de dominios finitos por:

- Estructura de matriz de la variable de decisión.
- Operaciones de trasponer ésta matriz.
- Variables auxiliares para el cálculo de costos.
- Variables auxiliares para estrategias de búsqueda.
- Variables auxiliares para estrategias de distribución.
- Tamaño del árbol generado.

Se ha encontrado que se genera un gran número de variables de estado finitos y propagadores, no es posible realizar un cálculo estimado de cuantas se generan en una instancia dada del problema, pero se espera un gran número debido a la forma en que se ha implementado el aplicativo.

6.5.2. Tamaño de entradas

La mayor entrada corresponde a una asignación de 831 canales, que corresponde a la banda de 800MHz, sin embargo el *storage* de Mozart OZ, presenta ciertas limitaciones que al ser desbordadas generan en un problema de memoria que produce un error en el aplicativo y lo cierra de inmediato. Este problema fue encontrado al inicio del proyecto cuando se intentó manejar una entrada con los detalles de las divisiones geográficas, por lo que se debe tener presente para futuros desarrollos usando como base la aplicación presentada en este proyecto de grado.

6.5.3. Errores conocidos del aplicativo

Los errores conocidos del aplicativo son:

- Si se usa una versión antigua de un navegador, los reportes no se verán por la incompatibilidad de las funciones de JavaScript.
- Si una entrada no es válida no se generará archivo de salida, mostrándose un mensaje de error. Para éste efecto se ha creado un generador de entradas, debido a la complejidad de la estructura de la misma.

Es posible existan más errores, sin embargo estos son los más relevantes y que no pueden ser resueltos desde el proceso de implementación de la aplicación prototipo.

6.5.4. Consideraciones para implementación usando programación por restricciones

Es necesario contar con la versión *1.4.0* de *Mozart OZ* debido a que el *Parser XML* ha sido implementado para ésta versión, durante el proyecto se encontró el problema que el servidor de AVISPA, contaba con la versión *1.3.6* que no tiene implementado el *Parser XML*, la solución fue instalar en una carpeta aparte la versión *1.4.0* y usar las funciones *putenv()* y *setenv()* ¹ que permiten configurar las variables de entorno, en específico la variable *PATH* para direccionar la ejecución hacia la versión actualizada.

¹Para mayor información consultar documentación de PHP disponible en <http://www.php.net/manual/es>, consultado Noviembre 2012'

Capítulo 7

Implementación del modelo usando un algoritmo genético

En el proyecto se ha desarrollado un método alternativo de solución usando el modelo y la metodología para aplicaciones Web del grupo Avispa, esto es para mostrar que el modelo desarrollado en el proyecto puede ser aplicado en otras metodologías que permiten solucionar el problema de asignación de frecuencias.

El algoritmo propuesto en éste capítulo se basa en un método genético para solucionar el problema de *channel allocation* [30].

7.1. Diseño del algoritmo genético

7.1.1. El algoritmo

El diseño del algoritmo genético tiene los siguientes pasos:

- Definición del individuo.
- Función de aptitud.
- Criterio de selección.
- Criterio de cruce.
- Criterio de mutación.
- Selección de individuos para la próxima generación.
- Criterio de parada.

El objetivo del algoritmo es minimizar el costo de la solución.

El individuo

El individuo es una cadena de caracteres con ceros y unos de tamaño N por $T = \text{card}(\text{Opt})$, que representa la asignación final.

Función de aptitud

El costo total se define tomando en cuenta:

- $Cost_T$ de la ecuación 4.12.
- C número de canales.
- NR_e número de restricciones que la solución infringe.

$$CostGen_T = 10000 * T * C * NR_e + Cost_T \quad (7.1)$$

En la ecuación se busca darle un mayor costo a una solución que presente infracciones a las restricciones del modelo, pero se presentan algunos problemas que se explican en la sección 7.1.2.

Para definir la función de aptitud se normalizan los costos de la población de acuerdo a una curva Gaussiana con desviación estándar 1.5. Se hace un corte de acuerdo al método de *sigma truncation*[31].¹

Criterio de selección

La selección se hace por ruleta², para seleccionar un número de individuos igual al de la población inicial.

Criterio de cruce

Los individuos se cruzan con una probabilidad especificada por el usuario. Se utiliza el criterio de reemplazo por inserción³ para generar dos individuos hijos por pareja de individuos padres.

Criterio de mutación

Se selecciona un individuo de acuerdo a una probabilidad especificada por el usuario y se cambia el valor de uno de sus campos elegido con una función pseudoaleatoria, si el campo contiene un cero se cambia por uno y viceversa.

Selección individuos próxima generación

Se seleccionan los individuos generados con el criterio de cruce y un número de los mejores padres para completar el tamaño inicial de la población, así generación tras generación el tamaño de la población es estable.

Criterio de parada

Se realiza de acuerdo a un tiempo de ejecución especificado por el usuario.

¹El método de sigma *sigma truncation* consiste en descartar los individuos cuya función de aptitud sea peor que cierto valor de desviación estándar.

²La selección por ruleta utiliza un número aleatorio r y el tamaño de la población inicial P para generar $r_j = \frac{r+j-1}{j} \forall j = 1, \dots, P$

³El reemplazo por inserción, significa tomar cierto porcentaje de características de cada individuo y mezclarlas para generar uno o más individuos hijos

7.1.2. Problemas de modelado

Debido a la formula de la función de aptitud, el algoritmo teóricamente debe presentar el problema de mínimos locales, en este caso se presentaría en el momento que no se tenga infracciones a las restricciones y se tenga un costo que no es el óptimo, el algoritmo puede quedar atrapado allí debido a que para obtener un mejor costo debe pasar por un proceso que implica tener costos muy por encima del obtenido en ese punto.

Para enfrentar ese problema el usuario puede decidir cuantas veces ejecutar el algoritmo para aumentar su precisión.

7.1.3. Valores recomendados

En la siguiente tabla se muestran los parámetros recomendados de acuerdo al método de solución propuesto [30].

Tabla 7.1: Valores recomendados para la aplicación basada el algoritmos genéticos

Parámetro	Valor
Número de individuos.	20-40.
Probabilidad de cruce individuo seleccionado	90 % (0.9).
Probabilidad mutación	2 % (0.02).
Valor de sigma	1.5 desviación estándar.

7.2. Diseño e implementación de la aplicación

De acuerdo a lo especificado en el capítulo 5, se realizan algunos ajustes para permitir la implementación bajo el criterio de un algoritmo genético.

7.2.1. Parámetros de la aplicación

A partir de la metodología para aplicaciones Web del grupo Avispa se han definido los siguientes parámetros para la aplicación:

Tabla 7.2: Parámetros para la aplicación basada el algoritmos genéticos

Parámetro	Función
— n	Número de veces que se ejecuta el algoritmo.
— p	Población inicial.
— ec	Probabilidad de cruce, entre 0 y 100.
— em	Probabilidad de mutación, entre 0 y 100.

7.2.2. Formato entradas

Debido a que en el lenguaje *C++* no es fácil de forma nativa realizar la lectura de entradas en formato XML, se ha definido una entrada propia para la aplicación, que se obtiene a partir de un proceso de conversión que realiza el módulo de conversor de entradas que se explica en la sección 7.2.3. El formato de la entrada es el siguiente:

```
Total de canales
Numero de operadores presentes
Numero de operadores que requieren asignacion
Numero total de operadores
Numero de operadores presentes que no solicitan asignacion
Numero de operadores presentes que solicitan asignacion
Operadores que solicitan asignacion
Requerimientos de operadores que solicitan asignacion
Numero de operadores que no están presentes y solicitan asignacion
Lista de operadores presentes en la banda
Lista de operadores actuales que no solicitan asignacion
Lista de operadores actuales que si solicitan asignacion
Lista de operadores que entran y no solicitan asignacion
Total de operadores que van a ir en la banda
Lista de asignaciones actuales
Lista de asignaciones en divisiones
Lista de CanalesReservados
Lista de CanalesInutilizados
ID banda de frecuencia
ID rango de frecuencia
Separacion
Tope
Tipo asignacion geografica
ID asignacion geografica
```

En el anexo ejemplo de entrada para algoritmo genético de la página 74 se puede consultar una instancia para el problema de asignación de canales para el algoritmo genético.

7.2.3. Módulos

El aplicativo consta de dos módulos principales que son explicados a continuación.

Conversor XML a entradas

Este módulo es una pequeña modificación al módulo de entradas de la aplicación basada en programación por restricciones explicado en la sección 6.4.3. La modificación consiste en que en lugar de almacenar en memoria los datos extraídos del XML de entrada genera un archivo de entrada para el algoritmo genético, de acuerdo al formato que se ha establecido anteriormente.

Aplicativo

El aplicativo está escrito en lenguaje *C++* y realiza todas las tareas que se han definido en la sección 7.1.1, como salida se genera un archivo XML con el formato especificado en la sección 5.2.

Capítulo 8

Interfaces Web de la aplicación

Las interfaces Web son aquellas que interactúan directamente con el usuario, recibiendo las entradas y parámetros de las aplicaciones que son enviados a los diferentes aplicativos del proyecto y desplegando los resultados de forma tal sean de fácil lectura e interpretación.

A continuación se describe el proceso de diseño e implementación de las interfaces Web del proyecto que se han integrado al portal de programación por restricciones del grupo Avispa¹

8.1. Estructura de la aplicación

La aplicación ha sido diseñada bajo la siguiente estructura:

- Vistas de usuario que son explicadas en la sección 8.1.2.
- Scripts en lenguaje *PHP* y *Javascript* para dar soporte de comunicación de datos.
- Archivos ejecutables del aplicativo.
- Estructura de carpetas para dar persistencia a los datos de usuario.

La aplicación se estructura de acuerdo a la figura 8.3.

¹El portal se puede encontrar en <http://avispa.univalle.edu.co>, consultado en Noviembre de 2012



Figura 8.1: Estructura de aplicación

Como se observa en la anterior figura el aplicativo se divide en dos grandes partes: la primera que es la sección de gestión de la base de datos de gestión del espectro, donde se puede gestionar la información sobre el estado de las asignaciones o realizar y la segunda son los aplicativos que permiten encontrar soluciones y desplegar soluciones a instancias del problema.

Para el uso de la aplicación se recomienda consultar los manuales de usuario en la página 75

8.1.1. Módulos de la aplicación

La aplicación se encuentra organizada por módulos, cada uno de los cuales cumple una función específica. Estos módulos han sido diseñados de acuerdo a la especificación de estructura de la aplicación y para facilitar el uso a los usuarios finales.

Gestión base de datos

El módulo de gestión de base de datos presenta dos grandes funciones, la primera de ellas es la consulta de información a la base de datos y la segunda es de realizar operaciones de edición sobre parte de la información almacenada.

En la tabla 8.1 se explican las operaciones de consulta de información almacenada en la base de datos:

Tabla 8.1: Consultas en el aplicativo prototipo de gestión del espectro

Consulta	Descripción
Consultar por entidad territorial por banda	Permite consultar en una zona geográfica determinada la asignación en una banda de frecuencia seleccionada.
Consultar por entidad territorial por operador	Despliega la información sobre la asignación en un territorio de un operador en específico.

En la tabla 8.2 se muestran las operaciones de edición de información que se encuentra en la base de datos:

Tabla 8.2: Operaciones de edición en la base de datos gestión del espectro

Operación	Descripción
Consultar por entidad territorial por banda	Permite consultar en una zona geográfica determinada la asignación en una banda de frecuencia seleccionada.
Gestión Operadores	Permite agregar o editar operadores, no es posible eliminar un operador para garantizar integridad de los datos. También permite agregar o quitar servicios que presta un operador.
Gestión Servicios	Permite agregar o editar servicios, no es posible eliminar debido a la gran dependencia de los datos de asignaciones, operadores y rangos de frecuencia asociados a un servicio.
Gestión Rangos de Frecuencia	Permite editar o crear rangos de frecuencia, no es posible eliminar un rango de frecuencia para garantizar integridad de los datos. La creación de rangos de frecuencia permite especificar las formulas de transmisión o recepción de los canales, una vez creado un rango de frecuencia es posible editar sus canales individualmente.
Gestión Entradas XML	Permite visualizar, descargar y eliminar entradas XML que tenga el usuario almacenadas en el servidor.
Gestión Salidas XML	Permite visualizar, descargar y eliminar salidas XML que tenga el usuario almacenadas en el servidor.

Generador de entradas XML

La generador de entradas permite a partir de la información almacenada en la base de datos y los requerimientos del usuario generar una entrada válida para la aplicación, éste módulo evita que el usuario tenga que hacer entradas manualmente, que debido a la especificación de formatos son difíciles de hacer. El generador de entradas funciona de la siguiente manera:

1. El usuario selecciona la zona geográfica y el rango de frecuencia de interés donde desea realizar asignación
2. Se crean los requerimientos, los operadores que se listan son aquellos que prestan los servicios que acepta la banda seleccionada.
3. Se genera el archivo XML y el usuario decide si lo guarda o no.

Aplicación basada en programación por restricciones

Esta se ha construido de acuerdo a lo definido en el capítulo 6, el aplicativo está construido con las siguientes funciones:

- Un módulo que permite seleccionar los parámetros de aplicación y un archivo de entrada.
- Un módulo que permite seleccionar cuantas iteraciones se desean realizar (entre 1 y 6) y decidir que restricciones no se toman en cuenta en cada una.
- Un módulo para seleccionar que salida dentro de los resultados obtenidos y decidir si se guarda en el sistema.

La aplicación realiza un proceso de recolección de datos suministrados por el usuario, los envía al aplicativo y despliega la información.

Aplicación basada en algoritmos evolutivos

Esta se ha construido de acuerdo a lo definido en el capítulo 7, el aplicativo está construido con las siguientes funciones:

- Un módulo que permite seleccionar los parámetros de aplicación y un archivo de entrada en formato XML.
- Un módulo para seleccionar que salida dentro de los resultados obtenidos y decidir si se guarda en el sistema.

La aplicación realiza un proceso de recolección de datos suministrados por el usuario, los envía al conversor de entradas, ejecuta el aplicativo y despliega la información.

Insertar XML de salida en base de datos

El proceso de inserción de la salidas XML a la base de datos es un proceso que permite tomar los datos de una salida e insertarlos en la base de datos. Éste proceso presenta los siguientes pasos:

1. Lectura del archivo XML de salida, tomando la solución que ha elegido el usuario.
2. Se verifica que tipo de asignación geográfica es: nacional, territorial, departamental o municipal.
3. Se verifica la banda seleccionada.
4. Se borra la asignación actual en la banda correspondiente al área geográfica seleccionada.
5. La asignación se realiza jerárquicamente, se miran las asignaciones de entidades geográficas superiores y se descartan, escribiendo solamente las asignaciones correspondientes a esa zona. Por ejemplo si se mira un municipio, se descartan las que se encuentran asignadas a nivel nacional, territorial y departamental.

Es importante aclarar, que en este proceso la salida XML debe corresponder fielmente a la configuración de la banda, especialmente en el número de canales, de otra forma se perderá la integridad en la base de datos.

8.1.2. Integración con el portal de Avispa

La integración con el portal de Avispa que se encuentra construido usando el gestor de contenidos Drupal[28], se realiza en base a la experiencia adquirida en la metodología de aplicaciones Web del grupo Avispa, se aprovecha la gestión de usuarios y vistas que provee el gestor de contenidos para interactuar con el usuario, se construye una estructura de archivos que proveen la comunicación con los aplicativos construidos para éste proyecto.

Vistas de usuario

Las vistas de usuario son:

- **Formularios Web:** Para enviar parámetros y archivos a los aplicativos del proyecto.
- **Estilos CSS y Javascript:** Para proveer las vistas en consultas, archivos XML de entrada y archivos XML de salida, se utiliza fuertemente la librería *JQuery* de Javascript para hacer vistas interactivas.

Las diferentes vistas pueden ser consultadas en el manual de usuario anexo al final de éste documento.

Los resultados de los aplicativos se despliegan así:

- Una información sobre la ejecución del aplicativo, se incluye información sobre los propagadores, variables de dominios finitos, uso de memoria y espacios de computación en el anexo manual de usuario.
- Un reporte acerca de los costos de cada solución y un reporte en el que se muestra la asignación que se ha calculado para la entrada.

Un ejemplo de los reportes puede observar en el anexo de los manuales de usuario en la página 78 en el anexo manual de usuario en éste documento.

Datos persistentes de usuario

Los datos de usuario están compuestos por:

- *Entradas no persistentes:* se generan automáticamente en el generador de entrada. No son visibles en los aplicativos, si se desean convertir en permanentes el usuario debe indicarlo en el generador de entradas al guardar la información. Es importante aclarar que existe un *Script* que borra estas entradas a las 00:00h de cada día.
- *Entradas:* Se encuentran almacenadas en el sistema, son permanentes, el usuario puede borrarlas en el módulo gestión de entradas XML.
- *Salidas no persistentes:* se generan automáticamente en los aplicativos. No son visibles en el módulo de insertar información en base de datos, si se desean convertir en permanentes el usuario debe indicarlo en el aplicativo al guardar la información. Al igual que las entradas no persistentes son borradas a las 00:00h de cada día.
- *Salidas:* Son permanentes, el usuario puede borrarlas en el módulo gestión de salidas XML.

Se hace uso del atributo *uuid* de la variable global *user* de Drupal, que provee un identificador único por usuario, la estructura de carpetas es la siguiente:

```
/entradasTemporales/$user->uuid  
/entradas/$user->uuid  
/salidasTemporales/$user->uuid  
/salidas/$user->uuid
```

Cada aplicativo del proyecto y el generador de entradas verifica que exista ésta estructura, si no existe procede a crearla, esto para garantizar la persistencia de datos de cada usuario del sistema.

Roles de usuario

Existen dos roles de usuario:

- **Usuario normal:** Sólo puede realizar consultas a la base de datos y utilizar los aplicativos disponibles en el proyecto.
- **Usuario administrador del espectro:** Puede realizar consultas, utilizar los aplicativos y hacer operaciones de edición sobre la base de datos.

Esto se implementa gracias a la administración de roles de usuario que posee *Drupal* con el atributo *Roles* de la variable global *user*, que permite discriminar que roles están asignados a un usuario.

Seguridad.

El gestor de contenidos Drupal almacena un registro de todas las acciones realizadas por un usuarios, por lo que se exige que todos los usuarios tengan cuentas, en caso de no tenerla el aplicativo mostrará el siguiente mensaje.

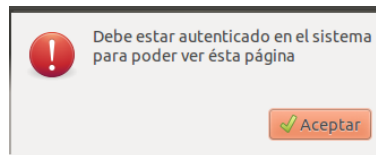


Figura 8.2: Alerta sobre usuario no autenticado.

Posteriormente será direccionado a la página inicial del sitio. En caso de que el usuario no sea administrador e intente ingresar alguna de las páginas donde se realizan modificaciones a la base de datos, se le mostrará el siguiente mensaje:

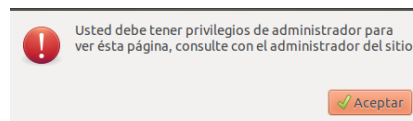


Figura 8.3: Alerta sobre usuario sin rol de administrador.

Posteriormente será direccionado a la página inicial del sitio.

Es de anotar que las restricciones están impuestas en lenguaje *PHP* por lo que no es posible infringir estas restricciones deshabilitando el lenguaje *Javascript* en el navegador Web.

8.2. Despliegue de información al usuario

8.2.1. Consultas

Las consultas son desplegadas al usuario utilizando una librería conocida como *DataTables*² que es un complemento para *JQuery* que permite filtrar información y realizar búsquedas para facilitar su lectura.

²Para mayor información consultar: <http://datatables.net/>, Noviembre 2012

A continuación en la figura 8.4 se muestra un ejemplo de como se despliega una consulta a un usuario.

Mostrar 10 registros

Buscar:

Rango de frecuencia	Canal	Descripción canal	Operador	Tipo de asignación
BANDA DE 800 MHz	2	Telefonia móvil celular 800Mhz Canal 2	Movistar	Nacional
BANDA DE 800 MHz	3	Telefonia móvil celular 800Mhz Canal 3	Movistar	Nacional
BANDA DE 800 MHz	4	Telefonia móvil celular 800Mhz Canal 4	Movistar	Nacional
BANDA DE 800 MHz	5	Telefonia móvil celular 800Mhz Canal 5	Movistar	Nacional
BANDA DE 800 MHz	6	Telefonia móvil celular 800Mhz Canal 6	Movistar	Nacional
BANDA DE 800 MHz	7	Telefonia móvil celular 800Mhz Canal 7	Movistar	Nacional
BANDA DE 800 MHz	8	Telefonia móvil celular 800Mhz Canal 8	Movistar	Nacional
BANDA DE 800 MHz	9	Telefonia móvil celular 800Mhz Canal 9	Movistar	Nacional
BANDA DE 800 MHz	10	Telefonia móvil celular 800Mhz Canal 10	Movistar	Nacional
BANDA DE 800 MHz	11	Telefonia móvil celular 800Mhz Canal 11	Movistar	Nacional
Rango de frecuencia	Canal	Descripción canal	Operador	Tipo de asignación

Mostrando desde 1 hasta 10 de 36 registros

Primero Anterior 1 2 3 4 Siguiente Último

Figura 8.4: Ejemplo de consulta a base de datos.

8.2.2. Archivos de entrada y salida

El contenido de los archivos de entrada y salida pueden ser consultados en los módulos de Gestión de entradas y salidas respectivamente, a continuación en la figura 8.5 se muestra un despliegue de una entrada XML.

Información

Concepto	Valor
Tipo de asignación geográfica	Territorial/Regional
Lugar de asignación geográfica	
Banda de frecuencia	High Frequency HF/3 - 30 MHz
Rango de frecuencia	4357-4355 kHz
Número de canales de la banda	29
Separación mínima requerida	1
Tope de canales por operador en la banda	10
Operadores presentes en la banda	<ul style="list-style-type: none"> Municipio de Cali Ministerio de minas Radioaficionados Agencia de vigilancia Titan Radioastronomía
Operadores con requerimientos	<ul style="list-style-type: none"> Agencia de vigilancia Lost Municipio de Cali
Requerimientos	<ul style="list-style-type: none"> Agencia de vigilancia Lost: 2 canales Municipio de Cali: 3 canales
Maxima asignación en subdivisiones	<ul style="list-style-type: none"> Agencia de vigilancia Lost: 0 canales Municipio de Cali: 0 canales

Estado actual de la banda

Operador	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23
Canales reservados																							
Canales deshabilitados																							
Asignados en subdivisiones																							
Municipio de Cali																							
Ministerio de minas			X	X																			
Radioaficionados	X																						
Agencia de vigilancia Titan											X	X	X	X									
Radioastronomía						X	X	X	X														

Figura 8.5: Despliegue información de la entrada.

Para el caso de las salidas el despliegue es el mismo al de los resultados en los aplicativos, la diferencia es que se muestran todas las soluciones y no una en particular.

8.2.3. Resultados

Cada archivo de salida contiene una o más soluciones encontradas de una instancia del problema, estas contienen información acerca de los parámetros de ejecución y los costos de cada solución, en la figura 8.6 se puede observar como se despliega el usuario la información general de la aplicación y en la figura 8.7 se muestra la información en particular para una solución encontrada para una instancia.

Información	
Concepto	Valor
Tipo de solución	Óptima
Tipo de asignación	Territorial/Regional
Lugar de asignación	DIRECCION TERRITORIAL BARRANQUILLA
Banda	High Frequency HF /3 - 30 MHz
Rango de frecuencia	4357-4355 kHz
Número de canales	29
Tope por operador en la banda	10
Número de soluciones	2
Espacios de computación creados	1
Número de variables de dominios finitos	144909
Número de propagadores	148028
Uso de memoria (Bytes)	339992
Separación de canales	1
Número de operadores por canal	1
Considerar tope de la banda	Si
Se conserva la asignación de los operadores que requieren asignación	Si
Se considera la separación	Si

Soluciones

Figura 8.6: Ejemplo de consulta a base de datos.

Solución: 0	
Costos	
Concepto	Valor
Número de bloques	6
Diferencia entre el mayor bloque libre y el total de canales	23
Número de canales inútiles por separación en operadores que requieren asignación	6
Costo total	1155

Operador	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21
Ministerio de minas			X	X																	
Radioaficionados	X																				
Radioastronomía						X	X	X	X												
Agencia de vigilancia Titan											X	X	X	X							
Agencia de vigilancia Lost																X	X				
Municipio de Cali																					

Figura 8.7: Ejemplo de consulta a base de datos.

Capítulo 9

Experimentación y pruebas

En este capítulo se describe un proceso de pruebas a las aplicaciones desarrolladas en el proyecto, para realizar un estudio con respecto a qué ventajas presentan ciertas características de la aplicación basada en programación por restricciones, como son el motor de búsqueda y las estrategias de distribución. También se analiza la eficiencia del algoritmo evolutivo.

A partir de las pruebas realizadas, se hace un análisis que permite comprobar en qué aspectos difiere cada característica de las soluciones implementadas en el proyecto.

9.1. Descripción pruebas

9.1.1. Entradas de prueba

Para probar diferentes escenarios en la gestión del espectro, se tienen los tipos de entradas que son definidos en la tabla 9.1

Tabla 9.1: Entradas usadas para pruebas

Código	Número canales	Descripción
E1	53	Banda 22000-22853 kHz, Sin asignación, Tope:10, requerimientos: 3 operadores 2 canales cada uno.
E2	53	Banda 22000-22853 kHz, Sin asignación, Tope:10, requerimientos, 3 operadores: 10 canales cada uno.
E3	53	Banda 22000-22853 kHz, 10 canales asignados al inicio de la banda, Tope:10, requerimientos, 3 operadores: 10 canales cada uno.
E4	53	Banda 22000-22853 kHz, 10 canales asignados al final de la banda, Tope:10, requerimientos, 3 operadores: 10 canales cada uno.
E5	53	Banda 22000-22853 kHz, 10 canales asignados en la mitad de la banda, Tope:10, requerimientos, 3 operadores: 10 canales cada uno.
E6	53	Banda 22000-22853 kHz, 10 canales asignados aleatoriamente, Tope:10, requerimientos, 3 operadores: 10 canales cada uno.
E7	480	Banda 254-260 MHz, Sin asignación, Tope:10, requerimientos, 2 operadores: 8 canales cada uno.

E8	480	Banda 254-260 MHz, Sin asignación, Tope:10, requerimientos, 4 operadores: 8 canales cada uno.
E9	480	Banda 254-260 MHz, Sin asignación, Tope:10, requerimientos, 6 operadores: 8 canales cada uno.
E10	480	Banda 254-260 MHz, Sin asignación, Tope:10, requerimientos, 8 operadores: 8 canales cada uno.
E11	480	Banda 254-260 MHz, Sin asignación, Tope:10, 10 operadores: requerimientos (8,4,1,3,2,10,8,7,4,1).
E12	480	Banda 254-260 MHz, Sin asignación, Tope:10, 20 operadores: requerimientos (8,4,1,3,2,10,8,7,4,1,3,8,8,4,6,2,8,9,10).
E13	813	Banda800Mhz, Asignación variada, en divisiones y otros operadores, Tope:80, requerimientos, 10 operadores 8 canales cada uno
E14	813	Banda800Mhz, Asignación variada, en divisiones y otros operadores, Tope:80, requerimientos, 10 operadores (13,24,15,6,14,29,1,12,3,23)
E15	813	Banda800Mhz, Asignación variada, en divisiones y otros operadores, Tope:80, requerimientos, 20 operadores 20 canales cada uno
F1	53	Banda 22000-22853 kHz, Sin asignación, Tope:10, requerimientos: 3 operadores 13 canales cada uno.
F2	53	Banda 22000-22853 kHz, Sin asignación, Tope:10, requerimientos: 6 operadores 10 canales cada uno.
F3	53	Banda 22000-22853 kHz, Sin asignación, Tope:10, requerimientos: 6 operadores 13 canales cada uno.

Las entradas han sido creadas para estudiar varios casos que se pueden presentar en la asignación:

- En las entradas E1 a E5 se estudia las implicaciones de asignación del problema en una banda pequeña, tomando en cuenta la forma en que está asignada.
- Las entradas E7 a E12 permiten estudiar el comportamiento ante requerimientos, de tamaño y si son homogéneos (todos piden lo mismo) y heterogéneos en una banda no asignada.
- Las entradas E1, E7 y E13 permiten comparar el funcionamiento del aplicativo ante diferentes tamaños de entrada en condiciones similares.
- Para las entradas E13 a E15 las asignaciones son: operadores en la banda 140 canales y operadores en divisiones 11 canales
- Las entradas F1, F2 y F3 no cumplen requerimientos. F1 no cumple tope, F2 los requerimientos sobrepasa la capacidad de la banda y F3 los requerimientos superan tope y capacidad de banda.

Las entradas, se clasifican así:

- **Pequeñas:** Si son generadas a partir de rangos de frecuencia con menos de 20 canales.
- **Medianas:** Si son generadas a partir de rangos de frecuencia entre 20 y 100 canales.
- **Grandes:** Si son generadas a partir de rangos de frecuencia con más de 100 canales.

9.1.2. Procedimiento de pruebas

Para realizar las pruebas en los aplicativos se establecen algunas pautas para permitir la recolección de datos y los análisis posteriores.

Codificación parámetros de pruebas

Los parámetros de desempeño y su respectiva codificación son los siguientes:

- **Ns**: Número de soluciones.
- **Ec**: Espacios de computación creados.
- **Nv**: Número de variables de dominios finitos.
- **Np**: Número de propagadores.
- **Um**: Uso de memoria (Bytes).
- **Mc**: Mejor costo solución encontrada.

Importante: Para el caso de las pruebas con el algoritmo genético sólo se toma en cuenta el costo de la mejor solución encontrada, debido a que los otros parámetros por las diferencias de ambos métodos no se pueden comparar.

Los diferentes motores de búsqueda se codifican así:

- **CT**: Mejor costo total.
- **C1**: Mejor por tamaño de bloque libre.
- **C2**: Mejor por número de bloques de asignados.
- **C3**: Mejor por número de canales inutilizables.

La codificación para las estrategias de distribución es:

- **S1**: Asignar primero al inicio de la banda.
- **S2**: Asignar primero al final de la banda.
- **S3**: Asignar primero al inicio de la banda a operadores con asignación.
- **S4**: Asignar primero al final de la banda a operadores con asignación.
- **S5**: Asignar primero al inicio de la banda a operadores sin asignación.
- **S6**: Asignar primero al final de la banda a operadores sin asignación.
- **S7**: Asignar primero al inicio de la banda a operadores con mayores requerimientos.
- **S8**: Asignar primero al final de la banda a operadores con mayores requerimientos.
- **S9**: Asignar primero al inicio de la banda a operadores con menores requerimientos.

- **S10:** Asignar primero al final de la banda a operadores con menores requerimientos.
- **SG1:** Genérica: Naive.
- **SG2:** Genérica: ff.
- **SG3:** Genérica: split.

Procedimiento de pruebas

Para las pruebas se trata de probar los diferentes parámetros de la aplicación, variando cada característica manteniendo las otras estáticas para facilitar el análisis de cada una de ellas y su influencia en el cálculo de la solución a una instancia del problema.

Es importante aclarar que una solución no se marca como **óptima** hasta que se termine de explorar el árbol de búsqueda por completo.

9.2. Pruebas sobre aplicativo basado en programación por restricciones

9.2.1. Pruebas sobre tamaño de entrada

El objetivo de éstas pruebas es estudiar el desempeño de cada uno de los distribuidores con respecto al tamaño de la entrada, para éstas pruebas se establece:

- Motor de búsqueda es mejor costo
- Las entradas seleccionadas son: E1, E7 y E11
- Los tiempos de ejecución son: para E1 3seg, E7 y E11 10 seg.
- En la prueba se varía la estrategia de distribución.
- Los pesos para el costo de la solución total es 33 para cada costo.

Tabla 9.2: Pruebas para E1 con 3seg de ejecución variando estrategia de distribución

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S1	1	1174842	1197956	340336	2	429	No óptima
S2	1	1151598	1174266	340336	2	429	No óptima
S3	1	816354	832574	340336	2	429	No óptima
S4	1	1129914	1152164	340336	2	429	No óptima
S5	1	1205886	1229597	340336	2	429	No óptima
S6	1	1165014	1187940	340336	2	429	No óptima
S7	1	1208226	1231982	340336	2	429	No óptima
S8	1	872202	889496	340336	2	429	No óptima
S9	1	1161738	1184600	340336	2	429	No óptima
S10	1	1184826	1208132	340336	2	429	No óptima
SG1	1	1201674	1209898	340336	2	429	No óptima
SG2	1	1267038	1275681	340336	2	429	No óptima
SG3	1	1147542	1155419	340336	2	429	No óptima

Tabla 9.3: Pruebas para E7 con 10 seg de ejecución variando estrategia de distribución

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S1	1	1602260	1608723	340336	2	693	No óptima
S2	1	1715304	1722121	340336	2	693	No óptima
S3	1	1755540	1762483	340336	2	693	No óptima
S4	1	1668362	1675032	340336	2	693	No óptima
S5	1	1761288	1768249	340336	2	693	No óptima
S6	1	1462392	1468417	340336	2	693	No óptima
S7	1	1690396	1697135	340336	2	693	No óptima
S8	1	1751708	1758639	340336	2	693	No óptima
S9	1	1736380	1743263	340336	2	693	No óptima
S10	1	1977796	1981297	340336	2	693	No óptima
SG1	1	1504544	1507551	340336	2	693	No óptima
SG2	1	2036234	2039796	340336	2	693	No óptima
SG3	1	3435872	3440895	340336	2	693	No óptima

Tabla 9.4: Pruebas para E11 con 10 seg de ejecución variando estrategia de distribución

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S1	1	4834560	4191013	340336	2	2508	No óptima
S2	1	4753130	4838068	340336	1	2508	No óptima
S3	1	4628590	4713450	340336	1	2508	No óptima
S4	1	4901620	4986651	340336	1	2508	No óptima
S5	1	4863300	4948307	340336	1	2508	No óptima
S6	1	3766390	3850710	340336	1	2508	No óptima
S7	1	4398670	4483386	340336	1	2508	No óptima
S8	1	5098010	5183165	340336	1	2508	No óptima
S9	1	5054900	5140027	340336	1	2508	No óptima
S10	1	4798660	5088760	340336	1	2508	No óptima
SG1	1	6240	88160	340336	0	X	No óptima
SG2	1	6240	88160	340336	0	X	No óptima
SG3	1	6240	88160	340336	0	X	No óptima

Para las pruebas por motor de búsqueda se establece:

- En la prueba se utiliza la distribución S1.
- Las entradas seleccionadas son: E1, E7 y E11
- Los tiempos de ejecución son: para E1 3seg, E7 y E11 10 seg.

- Motor de búsqueda se varía.
- Los pesos para el costo de la solución total es 33 para cada costo.

Tabla 9.5: Pruebas para E1 con 3 seg de ejecución variando el motor de búsqueda

Motor búsqueda	Ec	Nv	Np	Um	Ns	Mc	Tipo
CT	1	1180926	1204157	340336	2	429	No óptima
C1	1	926334	944669	340336	2	429	No óptima
C2	1	920094	938309	340336	2	462	No óptima
C3	1	1177962	1201136	340336	2	462	No óptima
Ninguno	1	2034	2584	527024	12	891	No óptima

Tabla 9.6: Pruebas para E7 con 10 seg de ejecución variando el motor de búsqueda

Motor búsqueda	Resultados						
CT	1	2796886	2807090	340336	1	693	No óptima
C1	1	2933880	2944513	340336	1	693	No óptima
C2	1	2338004	2346771	340336	1	693	No óptima
C3	1	2534394	2543776	340336	1	429	No óptima
Ninguno	1	12938	14414	340336	10	1122	No óptima

Tabla 9.7: Pruebas para E11 con 10 seg de ejecución variando el motor de búsqueda

Motor búsqueda	Ec	Nv	Np	Um	Ns	Mc	Tipo
CT	1	4762710	4847654	340336	2	2508	No óptima
C1	1	4930360	5015409	340336	2	2508	No óptima
C2	1	4973470	5058546	340336	2	2541	No óptima
C3	1	4791450	4876412	340336	2	2541	No óptima
Ninguno	1	58930	140918	340336	11	2904	No óptima

9.2.2. Pruebas de acuerdo a la distribución de la asignación en la banda

Estas pruebas se ejecutan de la siguiente manera:

- Motor de búsqueda es mejor costo
- Las entradas seleccionadas son: E3, E4, E5 y E6.

- Los tiempos de ejecución para todas son 10 seg.
- En la prueba se utilizan las siguientes estrategias distribución: S1, S2, SG1, SG2 y SG3.
- Los pesos para el costo de la solución total es 33 para cada costo.

Tabla 9.8: Pruebas para E3 con 10 seg de ejecución variando estrategia de distribución

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S1	1	4563891	4630575	340336	1	1650	No óptima
S2	1	4804131	4874277	340336	1	1683	No óptima
SG1	1	4749843	4773512	340336	1	1683	No óptima
SG2	1	4881715	4906018	340336	1	1683	No óptima
SG3	1	4748803	4772467	340336	1	1683	No óptima

Tabla 9.9: Pruebas para E4 con 10 seg de ejecución variando estrategia de distribución

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S1	1	4812867	4883141	340336	2	1650	No óptima
S2	1	4832003	4902554	340336	1	1683	No óptima
SG1	1	4723011	4746551	340336	1	1683	No óptima
SG2	1	4698259	4721680	340336	1	1683	No óptima
SG3	1	4879635	4903928	340336	1	1683	No óptima

Tabla 9.10: Pruebas para E5 con 10 seg de ejecución variando estrategia de distribución

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S1	1	4814115	4884408	340336	1	1650	No óptima
S2	1	4746931	4816253	340336	1	1650	No óptima
SG1	1	4835539	4859620	340336	1	1650	No óptima
SG2	1	4949315	4973943	340336	1	1650	No óptima
SG3	1	4718019	4741535	340336	1	1650	No óptima

Tabla 9.11: Pruebas para E6 con 10 seg de ejecución variando estrategia de distribución

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S1	1	103747	371539	340336	1	2376	No óptima
S2	1	78995	300207	340336	1	2376	No óptima
SG1	1	90435	91703	340336	1	2376	No óptima
SG2	1	85859	87105	340336	1	2376	No óptima
SG3	1	89811	91076	340336	1	2376	No óptima

9.2.3. Pruebas sobre estructura requerimientos

Los requerimientos pueden estructurarse como homogéneos si todos los operadores solicitan el mismo número de canales o heterogéneos en caso contrario.

- Motor de búsqueda es mejor costo
- Las entradas seleccionadas son: E7, E8, E9, E10, E11, E12
- Los tiempos de ejecución para todas son 10 seg.
- En la prueba se utilizan las siguientes estrategias distribución: S6, S7, S8, S9, SG1, SG2 y SG3.
- Los pesos para el costo de la solución total es 33 para cada costo.

Tabla 9.12: Pruebas para E7 con 10 seg de ejecución variando estrategia de distribución

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S6	1	2765272	2775377	340336	1	693	No óptima
S7	1	2867778	2878204	340336	1	693	No óptima
S8	1	2832332	2842647	340336	1	693	No óptima
S9	1	2931964	2942591	340336	1	693	No óptima
SG1	1	3515386	3520492	340336	1	693	No óptima
SG2	1	3501974	3507066	340336	1	693	No óptima
SG3	1	3552748	3557893	340336	1	693	No óptima

Tabla 9.13: Pruebas para E8 con 10 seg de ejecución variando estrategia de distribución

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S6	1	3350612	3365946	340336	1	1419	No óptima
S7	1	3182004	3197075	340336	1	1419	No óptima
S8	1	3180088	3195155	340336	1	1419	No óptima
S9	1	4117012	4117012	340336	1	1419	No óptima
SG1	1	2900352	2911927	340336	1	1419	No óptima
SG2	1	2946336	2957935	340336	1	1419	No óptima
SG3	1	2938672	2950267	340336	1	1419	No óptima

Tabla 9.14: Pruebas para E9 con 10 seg de ejecución variando estrategia de distribución

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S6	1	4834560	4191013	340336	1	2508	No óptima
S7	1	4753130	4838068	340336	1	2508	No óptima

S8	1	4863300	4948307	340336	1	2508	No óptima
S9	1	3766390	3850710	340336	1	2508	No óptima
SG1	1	6240	88160	340336	0	X	No óptima
SG2	1	6240	88160	340336	0	X	No óptima
SG3	1	6240	88160	340336	0	X	No óptima

Tabla 9.15: Pruebas para E10 con 10 seg de ejecución variando estrategia de distribución

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S6	1	4834560	4191013	340336	1	2508	No óptima
S7	1	4753130	4838068	340336	1	2508	No óptima
S8	1	4863300	4948307	340336	1	2508	No óptima
S9	1	3766390	3850710	340336	1	2508	No óptima
SG1	1	6240	88160	340336	0	X	No óptima
SG2	1	6240	88160	340336	0	X	No óptima
SG3	1	6240	88160	340336	0	X	No óptima

Tabla 9.16: Pruebas para E11 con 10 seg de ejecución variando estrategia de distribución

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S6	1	4834560	4191013	340336	1	2508	No óptima
S7	1	4753130	4838068	340336	1	2508	No óptima
S8	1	4863300	4948307	340336	1	2508	No óptima
S9	1	3766390	3850710	340336	1	2508	No óptima
SG1	1	6240	88160	340336	0	X	No óptima
SG2	1	6240	88160	340336	0	X	No óptima
SG3	1	6240	88160	340336	0	X	No óptima

Tabla 9.17: Pruebas para E12 con 10 seg de ejecución variando estrategia de distribución

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S6	1	4834560	4191013	340336	1	2508	No óptima
S7	1	4753130	4838068	340336	1	2508	No óptima
S8	1	4863300	4948307	340336	1	2508	No óptima
S9	1	3766390	3850710	340336	1	2508	No óptima
SG1	1	6240	88160	340336	0	X	No óptima
SG2	1	6240	88160	340336	0	X	No óptima
SG3	1	6240	88160	340336	0	X	No óptima

9.2.4. Pruebas sobre tamaño de requerimientos

Se busca analizar el impacto del tamaño de los requerimientos en el desempeño de la aplicación.

- Motor de búsqueda es mejor costo
- Las entradas seleccionadas son: E2, E12, E15.
- Los tiempos de ejecución para todas son 10 seg.
- En la prueba se utilizan las siguientes estrategias distribución: S1, S2, SG1, SG2 y SG3.
- Los pesos para el costo de la solución total es 33 para cada costo.

Tabla 9.18: Pruebas para E2 con 10 seg de ejecución variando estrategia de distribución

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S1	1	4754835	4824272	340336	1	1650	No óptima
S2	1	4798099	4868157	340336	1	1650	No óptima
SG1	1	4678915	4702243	340336	1	1650	No óptima
SG2	1	4759411	4783126	340336	1	1650	No óptima
SG3	1	4812867	4836839	340336	1	1650	No óptima

Tabla 9.19: Pruebas para E12 con 10 seg de ejecución variando estrategia de distribución

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S1	1	2291080	2646865	340336	1	5709	No óptima
S2	1	2425200	2781027	340336	1	5709	No óptima
SG1	1	11040	366000	340336	0	X	No óptima
SG2	1	11040	366000	340336	0	X	No óptima
SG3	1	11040	366000	340336	0	X	No óptima

Tabla 9.20: Pruebas para E15 con 10 seg de ejecución variando estrategia de distribución

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S1	1	17451	429219	340336	0	X	No óptima
S2	1	17451	429246	340336	0	X	No óptima
SG1	1	17451	429158	340336	0	X	No óptima
SG2	1	17451	429158	340336	0	X	No óptima
SG3	1	17451	429158	340336	0	X	No óptima

9.2.5. Pruebas sobre tiempo de exploración en entradas grandes

En estas pruebas se busca medir el rendimiento de la aplicación ante entradas grandes, eligiendo diferentes tiempos de ejecución.

- Motor de búsqueda es mejor costo
- Las entrada seleccionada son: E13.
- Los tiempos de ejecución para todas son 5 seg, 10 seg y 20 seg.
- En la prueba se utilizan las siguientes estrategias distribución: S1, S2, SG1, SG2 y SG3.
- Los pesos para el costo de la solución total es 33 para cada costo.

Tabla 9.21: Pruebas para E13 con 5 seg de ejecución variando estrategia de distribución

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S1	1	542006	733982	340336	1	17985	No óptima
S2	1	1232566	1424734	340336	1	18975	No óptima
SG1	1	13296	205048	340336	0	X	No óptima
SG2	1	13296	205048	340336	0	X	No óptima
SG3	1	13296	205048	340336	0	X	No óptima

Tabla 9.22: Pruebas para E13 con 10 seg de ejecución variando estrategia de distribución

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S1	1	3832956	4025847	340336	1	17985	No óptima
S1	1	3908486	4101398	340336	1	18975	No óptima
SG1	1	13296	205048	340336	0	X	No óptima
SG2	1	13296	205048	340336	0	X	No óptima
SG3	1	13296	205048	340336	0	X	No óptima

Tabla 9.23: Pruebas para E13 con 20 seg de ejecución variando estrategia de distribución

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S1	1	9476126	9670586	340336	1	17985	No óptima
S2	1	9314276	9508691	340336	1	18975	No óptima

Para este caso SG1, SG2 y SG3, fallan por memoria virtual excedida.

9.2.6. Pruebas de acuerdo al nivel de recomputación

En estas pruebas se estudia el efecto del nivel de recomputación en el desempeño del proyecto, se escoge una entrada de tamaño mediano para realizar las diferentes pruebas.

- Motor de búsqueda es mejor costo
- Las entrada seleccionada son: E11.
- El tiempo de ejecución es 10 seg.
- En la prueba se utilizan las siguientes estrategias distribución: S1 y SG1.
- Los pesos para el costo de la solución total es 33 para cada costo.
- Los niveles de recomputación son 1, 2, 4 y 8.

Tabla 9.24: Pruebas para E11 con 10 seg de ejecución con nivel de recomputación 1

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S1	1	4753130	4838069	340336	2	2508	No óptima
SG1	1	6240	88160	340336	0	X	No óptima

Tabla 9.25: Pruebas para E11 con 10 seg de ejecución con nivel de recomputación 2

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S1	1	4676490	4761380	4761380	2	2508	No óptima
SG1	1	6240	88160	340336	0	X	No óptima

Tabla 9.26: Pruebas para E11 con 10 seg de ejecución con nivel de recomputación 4

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S1	1	4642960	4727829	340336	1	2508	No óptima
SG1	1	6240	88160	340336	0	X	No óptima

Tabla 9.27: Pruebas para E11 con 10 seg de ejecución con nivel de recomputación 6

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S1	1	3718490	3802780	340336	1	2508	No óptima
SG1	1	6240	88160	340336	0	X	No óptima

Tabla 9.28: Pruebas para E11 con 10 seg de ejecución con nivel de recomputación 8

Estrategia	Ec	Nv	Np	Um	Ns	Mc	Tipo
S1	1	4848930	984496	4933928	1	2508	No óptima
SG1	1	6240	88160	340336	0	X	No óptima

9.2.7. Pruebas de flexibilidad y debilidad de restricciones

Las pruebas correspondientes a la flexibilidad y debilidad de restricciones se realizan así:

- Motor de búsqueda es mejor costo
- Las entradas seleccionadas son: F1, F2, F3
- Los tiempos de ejecución para todas son 10 seg.
- Se establecen dos iteraciones así:
 - **Iteración 1:** No considerar separación y no considerar tope.
 - **Iteración 2:** No considerar tope y Número de operadores por canal es 2.
- En la prueba se utilizan las siguientes estrategias distribución: S1 y SG1.
- Los pesos para el costo de la solución total es 33 para cada costo.

Tabla 9.29: Pruebas flexibilidad y debilidad para F1

Estrategia distribución	Resultados iteración 1	Resultados iteración 2
S1	Ec:1 Nv:5417418 Np:5521525 Um:340336 Ns:1 Mc:1386, No óptima	Ec:0 Nv:5455326 Np:5560788 Um:340336 Ns:1 Mc:1452, No óptima
SG1	Ec:1 Nv:901970 Np:984496 Um:340336 Ns:1 Mc:1386, No óptima	Ec:1 Nv:5638002 Np:5674661 Um:340336 Ns:1 Mc:1452, No óptima

Tabla 9.30: Pruebas flexibilidad y debilidad para F2

Estrategia distribución	Resultados iteración 1	Resultados iteración 2
S1	Ec:1 Nv:477 Np:245128 Um:340336 Ns:0 Mc:X, No óptima	Ec:1 Nv:6572133 Np:6635126 Um:340336 Ns:1 Mc:1155, No óptima
SG1	Ec:1 Nv:477 Np:224 Um:340336 Ns:0 Mc:X, No óptima	Ec:1 Nv:6448269 Np:6468681 Um:340336 Ns:1 Mc:1188, No óptima

Tabla 9.31: Pruebas flexibilidad y debilidad para F3

Estrategia distribución	Resultados iteración 1	Resultados iteración 2
S1	Ec:1 Nv:477 Np:217848 Um:340336 Ns:0 Mc:X, No óptima	Ec:1 Nv:6812997 Np:6878325 Um:340336 Ns:1 Mc:1452, No óptima
SG1	Ec:1 Nv:477 Np:224 Um:340336 Ns:0 Mc:X, No óptima	Ec:1 Nv:6337197 Np:6357253 Um:340336 Ns:1 Mc:1485, No óptima

9.3. Pruebas sobre el aplicativo basado en algoritmos genéticos

9.3.1. Pruebas sobre entradas varias

Se realizan diferentes pruebas sobre el algoritmo genético, los parámetros del algoritmo son los especificados en la tabla 7.1.

Tabla 9.32: Pruebas algoritmo genético para E6

Tiempo (seg)	Resultados
3	Mc:2640, No óptima
30	Mc:2640, No óptima
300	Mc:2475, No óptima

Tabla 9.33: Pruebas algoritmo genético para E8

Tiempo (seg)	Resultados
3	Mc:40194, No óptima
30	Mc:35112, No óptima
300	Mc:31944, No óptima

Tabla 9.34: Pruebas algoritmo genético para E15

Tiempo (seg)	Resultados
3	Mc:161040, No óptima
30	Mc:165891, No óptima
300	Mc:161502, No óptima

9.3.2. Pruebas sobre entradas que no cumplen restricciones

Tabla 9.35: Pruebas algoritmo genético para F1

Tiempo (seg)	Resultados
3	Mc: 2508, No óptima
30	Mc: 2376, No óptima
300	Mc: 1881, No óptima

Tabla 9.36: Pruebas algoritmo genético para F2

Tiempo (seg)	Resultados
3	Mc: 2716, No óptima
30	Mc: 1980, No óptima
300	Mc: 1749, No óptima

Tabla 9.37: Pruebas algoritmo genético para F3

Tiempo (seg)	Resultados
3	Mc: 2739, No óptima
30	Mc: 2343, No óptima
300	Mc: 1551, No óptima

9.4. Resultados

9.4.1. Solución basada en programación por restricciones

Resultados

1. Comparando entre sí las estrategias de distribución creadas en este proyecto no presentan diferencias notables con respecto al desempeño ante entradas sin características.
2. El consumo de memoria de la ejecución depende del motor virtual de *Mozart* y no del tamaño de la entrada.
3. El mejor desempeño en uso de recursos, se encuentra cuando no se utiliza el motor de búsqueda, sin embargo éste presenta las peores soluciones.
4. El uso de los distribuidores que intentan asignar al inicio de la banda presentan un mejor rendimiento de los que intentan asignar inicialmente al final, cuando la banda se encuentra asignada en su inicio o final.

5. No se muestra ventaja de ningún distribuidor, si la banda se encuentra asignada aleatoriamente.
6. A medida que una banda se encuentra con más canales asignados, los parámetros de desempeño mejoran sensiblemente.
7. Con respecto a instancia con requerimientos homogéneos, no es posible obtener alguna mejora sustancial con alguno de los distribuidores implementados en el proyecto.
8. Las ejecuciones con instancias que tiene requerimientos heterogéneos, no muestran mejora frente a la estrategia de búsqueda que se aplique.
9. Al variar el tamaño de los requerimientos, las estrategias creadas en el proyecto tienden a buscar más rápidamente una solución válida que las estrategias genéricas.
10. En las estrategias de distribución propias del proyecto, los parámetros de desempeño presentan un comportamiento proporcional al tamaño de la entrada, los genéricos de *Mozart* no presentan variación en su desempeño frente a la entrada, pero no funcionan si la entrada es muy grande.
11. El nivel de recomputación no afecta sensiblemente el desempeño de la aplicación. Este permite mejorar un poco los parámetros de rendimiento, pero después de cierto valor éstos empeoran al tomar en cuenta la carga adicional al recalcular los nodos.
12. El consumo de memoria depende de la compilación del código.
13. Las restricciones flexibles y las iteraciones permiten probar quitando una o más restricciones, para encontrar posibles soluciones a entradas que no cumplen las restricciones.

Desventajas

La estructura de datos manejada en el proyecto tiene una complejidad de $O(c^2)$ donde c es el número de canales de la banda, por ser una matriz de operadores por número de canales, por lo que presenta un crecimiento considerable al aumentar el tamaño de la banda, así mismo los propagadores que se deben crear para implementar las restricciones dependen en gran medida del tamaño de la entrada, por lo que el aplicativo puede fallar cuando se tienen entradas grandes y se expanden tantos nodos que la máquina virtual de *Mozart OZ* presenta un desborde.

También se ha encontrado que si el archivo es grande ¹, el *Parser* de *Mozart OZ* puede presentar problemas por el gran tamaño del árbol de dependencias creado y puede producir un desborde.

La limitación de memoria viene dada por la infraestructura servidor virtual del grupo AVISPA, por lo que se ve la necesidad de utilizar estrategias de distribución que se deben considerar a partir del modelo, el modelo propuesto en éste proyecto tiene una alta dependencia de las variables de propagación entre sí que dificultan la creación de este tipo de estrategias.

9.4.2. Solución basada en algoritmos genéticos

El algoritmo genético presenta los siguientes resultados:

1. El algoritmo puede encontrar malas soluciones en un corto tiempo, pero se debe dejar un tiempo suficiente para encontrar buenas soluciones.

¹El archivo es de una banda con más de 100 canales

2. A medida que crece la entrada se debe dejar más tiempo el algoritmo para que encuentre soluciones válidas y eventualmente soluciones óptimas.
3. El diseño tiene problemas de mínimos locales que pueden hacer que en ciertas ejecuciones del algoritmo no se encuentre nunca la solución óptima, así se tenga una solución válida.
4. Por el diseño del algoritmo, no es posible saber si se tiene una solución óptima al problema.
5. El algoritmo presenta una convergencia muy lenta ante entradas grandes, por lo que es necesario estudiar técnicas para acelerar la convergencia del algoritmo a una solución válida.

9.4.3. Análisis comparativo entre los métodos de restricciones y evolutivo

Cada uno de los métodos de solución del problema tiene sus ventajas y sus desventajas, para entradas pequeñas ambos son apropiados, pero si la entrada crece el algoritmo genético requiere un tiempo demasiado grande para llegar a una solución válida, a diferencia del método por programación por restricciones.

Sin embargo, ante entradas pequeñas que no cumplen requerimientos el algoritmo genético puede funcionar ya que a pesar de no se cumpla una o más restricciones, se buscará la solución con el menor costo, a diferencia del método de programación por restricciones donde es necesario elegir que restricciones flexibilizar y usar iteraciones para escoger la mejor combinación para obtener el mejor costo posible ante estas instancias.

9.4.4. Análisis de extensibilidad del modelo

Como se pudo observar en las diferentes pruebas el modelo de asignación de canales y la metodología propuesta se pueden expandir sin mayores cambios a métodos de solución del problema diferentes al de programación por restricciones, como fue el caso del algoritmo genético que se implementó en éste proyecto.

Los cambios que se deben tomar en cuenta está principalmente enfocado a los formatos de entrada, ya que el formato XML es complejo y algunos lenguajes no cuentan con librerías para obtener su información directamente, lo que hace necesario crear otros formatos y utilizar módulos intermedios que hagan la conversión de los formatos de entrada y salida.

9.4.5. Aplicación del modelo en situaciones reales

El proceso de pruebas permite establecer que el modelo creado en el proyecto se puede aplicar en casos reales así:

- Instancias pequeñas del problema, en el cual no se tenga un gran número de requerimientos de entrada.
- Casos donde tener una asignación no óptima en las divisiones de un territorio no represente problemas graves en la normal prestación de servicios.
- El prototipo construido, ayuda a obtener una aproximación al comportamiento de una aplicativo más completo para la asignación del espectro utilizando programación por restricciones.

Capítulo 10

Conclusiones y trabajos futuros

10.1. Conclusiones

1. El problema de asignación de canales o frecuencias fue estudiado en base al problema de *channel assignment* y en las consideraciones que se encuentran en el cuadro nacional de atribución de frecuencias de Colombia, quienes permitieron abstraer el problema de tal forma que se pudiera escribir en un modelo lineal, en el cual se pudieran abstraer las restricciones más importantes que se deben tomar en cuenta en éste proceso.
2. El uso de modelo lineal permite realizar un proceso de implementación transparente ya que no se requirieron funciones propias de un lenguaje, lo que facilita su expansión a diferentes métodos de solución, como fue en este caso, que se diseñaron dos aplicativos, uno basado en programación por restricciones y otro basado en algoritmos evolutivos.
3. La metodología del grupo AVISPA para aplicaciones por restricciones permitió seguir una serie de pasos, para convertir en un servicio Web la aplicación creada en lenguaje *Mozart OZ*; así mismo se pudo usar con algunos cambios en otro paradigma de programación que en el proyecto fueron los algoritmos genéticos.
4. El uso del paradigma de programación por restricciones en el análisis comparativo permitió establecer que es el método más apropiado para solucionar el problema en instancias que cumplan las restricciones, debido a que se puede obtener una solución válida y con un costo cercano al óptimo rápidamente a diferencia del método del algoritmo evolutivo que tiene una convergencia bastante lenta.
5. La implementación sobre un 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 posibles soluciones que puede tomar una instancia del problema.
6. El uso de estrategias de distribución diseñadas para el proyecto permite acelerar las búsquedas de soluciones válidas al problema frente a estrategias genéricas, sin embargo son más costosas en recursos, debido a que la implementación y su concepción no considera aspectos de agilidad y de ahorro de recursos.
7. Los resultados permitieron establecer algunas pautas sobre qué estrategias de búsqueda y motores de búsqueda presentan mejor rendimiento frente a los otros tomando como base una forma conocida de la entrada. Sin embargo, en la mayoría de casos no se encuentra mejora en rendimiento comparando

un propagador con otro en entradas cuya forma no corresponde a las formas conocidas que se establecieron a lo largo del proyecto.

8. La gran ventaja del paradigma de programación por restricciones frente a otros métodos es lo rápido que se encuentra una solución válida a una instancia dada, sin ser afectado en gran medida por el tamaño de la entrada, sin embargo la búsqueda de soluciones óptimas es el punto débil del proceso que se ha realizado en éste proyecto, ya que por la alta complejidad que pueden tomar las instancias no es posible encontrar fácilmente un método que permita guiar la búsqueda hacia la solución óptima.
9. Las pruebas realizadas han arrojado que el uso de un motor de búsqueda aumenta sensiblemente el numero de propagadores y dominios finitos requeridos, ya que con éste se busca obtener la mejor solución bajo unos criterios definidos. Para entradas grandes resulta bastante costoso el uso de estos motores, por lo que, buscar la solución óptima en instancias grandes del problema puede ser un proceso bastante complicado, que requiere una gran capacidad de procesamiento.
10. En las pruebas se encontró que la restricción que más efecto produce en el rendimiento es la de asignación pues afecta sensiblemente los parámetros de ejecución, ya que a medida que la banda se encuentra más ocupada el número de propagadores y variables de estado finitos necesarias para realizar la búsqueda de soluciones se reduce considerablemente.
11. Se pudo establecer que la solución implementada en éste proyecto queda sujeta a algunas limitaciones computacionales, como son el tamaño de la entrada y el tamaño de los requerimientos, si son muy grandes el aplicativo puede fallar y no mostrarse soluciones a una instancia.
12. Como resultado final de las pruebas realizadas y del estudio del proceso de diseño e implementación, se puede establecer que la propuesta presentada en éste proyecto es un acercamiento a un proceso basado en programación por restricciones que permita solucionar éste problema. Se debe tomar en cuenta muchos más aspectos que están relacionados en el proceso de gestión del espectro, como son las asignaciones en detalle, el uso de datos reales y poder procesar entradas de gran tamaño sin presentar problemas en el aplicativo. Como aporte de éste proyecto se tiene una experiencia o acercamiento que permite tener una base para construir una propuesta que tenga como producto final un aplicativo para realizar el proceso de gestión del espectro usando el paradigma de programación por restricciones y pueda ser usado como un aplicativo Web.

10.2. Trabajos futuros

1. El modelo implementado en el proyecto no garantiza obtener una solución óptima a nivel de detalle de asignación, ya que no se consideran las asignaciones particulares de cada una de las divisiones territoriales de una zona geográfica donde se desee realizar el proceso de asignación de canales. Se debe a futuro considerar cambios al modelo y a la implementación que posibiliten estudiar la asignación en todo su detalle.
2. Se requiere estudiar un modelo que considere las posiciones y características de los transmisores de un operador en determinada área geográfica, esto para incluir en el modelo aspectos de propagación y calidad de servicio en el proceso de gestión del espectro.
3. Se deben considerar aspectos técnicos del proceso de la gestión del espectro, en especial la recomendación ITU K52, que trata los aspectos sobre niveles de radiación producto de emisiones electromagnéticas que deben respetarse para evitar efectos negativos en los seres humanos.
4. El proyecto fue realizado utilizando datos que fueron diseñados para realizar pruebas en el aplicativo pero que no corresponden a la realidad. Para poder estudiar el problema más a fondo se requieren datos reales, los cuales lamentablemente no se encuentran accesibles al público en general.
5. En las pruebas se encontró que el aplicativo basado en *Mozart OZ* presenta problemas en entradas grandes, por lo que se debe estudiar como superar éste problema desde el diseño de la aplicación o utilizar otras opciones de implementación que no dependen de una máquina virtual, como es el caso de *Gecode* que es una librería de programación por restricciones para C++.
6. Debido a que se necesita una gran capacidad de procesamiento en instancias grandes del problema, se recomienda realizar un estudio posterior sobre qué 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 cómputo 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.

Capítulo 11

Bibliografía

- [1] J. F. Díaz, C. Martinez, C. Delgado, F. Vargas, and M. Cruz, “Metodología para crear aplicaciones de programación por restricciones como servicio web. definición e implementación con diversos casos.” Documento interno del grupo AVISPA.
- [2] ITU, “International telecommunication union.” <http://www.itu.int/>, 2011. Noviembre de 2012.
- [3] TIA, “Broadband spectrum: The engine for innovation, job growth, and advancement of social priorities,” tech. rep., TIA, Advancing Global Communications, 2011.
- [4] Hurley and D. H. Smith, “Fixed spectrum frequency assignment algorithms genetic algorithms in engineering systems: Innovations and applications,” *IEEE*, 1995.
- [5] A. Quellmalz, A. Knalmann, and B. Muller, “Efficient frequency assignment with simulated annealing,” *IEEE*, 1995.
- [6] C. An, L. Zhang, and W. Liu, “A spectrum allocation algorithm based on matching game,” *IEEE*, 2009.
- [7] P. de la República Colombia, “Ley 1341 de 2009.” web.presidencia.gov.co/leyes/2009/julio/ley134130072009.pdf, 2009. Noviembre 2012.
- [8] M. de Tecnologías de Información y Comunicaciones, “Cuadro nacional de atribución de frecuencias,” tech. rep., República de Colombia, 2010.
- [9] Departamento, “Proceso de estudio y subasta para servicios 4g en colombia,” tech. rep., MinTIC, Gobierno de Colombia, 2012.
- [10] T. Roxborough, S. Medidi, and A. Sen, “On channel assignment problem in cellular networks,” in *Signals, Systems and Computers, 1997. Conference Record of the Thirty-First Asilomar Conference on*, vol. 1, pp. 630–634 vol.1, nov. 1997.
- [11] T. Weise, *Global Optimization Algorithms, Theory and Application*. it-weise.de (self-published): Germany, 2009.
- [12] ANE, “Agencia nacional del espectro.” <http://www.ane.gov.co/index.shtml>, 2012. Noviembre 2012.
- [13] R. Barták, “Guide to constraint programming.” <http://ktiml.mff.cuni.cz/~bartak/constraints/intro.html>, 1998. Noviembre 2012.

- [14] M. de Tecnologías de Información y Comunicaciones, “Ministerio de tecnologías de información y comunicaciones, portal web,” tech. rep., Gobierno Colombiano, 2010.
- [15] F. P. C. y. T. d. A. Centro de Apoyo Tecnológico a Emprendedores, “Estudio de los sistemas de gestión de contenidos web,” tech. rep., Creative Commons By-sa, 2012.
- [16] W. W. W. C. (W3C), “Extensible markup language (xml).” <http://www.w3.org/XML/>, 2012. Noviembre 2012.
- [17] S. de la República Colombia, “Ley 252 de 1995.” http://www.secretariasenado.gov.co/senado/basedoc/ley/1995/ley_0252_1995.html, 1995. Noviembre 2012.
- [18] S. de la República Colombia, “Ley 514 de 1999.” http://www.secretariasenado.gov.co/senado/basedoc/ley/1999/ley_0514_1999.html, 1999. Noviembre 2012.
- [19] I. T. Union, “Handbook national spectrum management,” tech. rep., International Telecommunication Union, 2005.
- [20] F. Rossi, P. van Beek, and T. Wals, *Handbook of Constraint Programming*. Oxford University, 2006.
- [21] K. R, *Principles of Constraint Programming*. Cambridge University, 2003.
- [22] P. V. Roy, “Second international conference, moz 2004, charleroi, belgium,” in *Multiparadigm Programming in Mozart/Oz*, 2004.
- [23] P. van Roy, “Conceptos, técnicas y modelos de programación.” Traducido por Juan Francisco Díaz.
- [24] A. N. del Espectro, “Estudio de análisis de toques de espectro radioeléctrico para servicios móviles terrestres,” tech. rep., MinTIC, Gobierno de Colombia, 2010.
- [25] M. Morara, J. Mauro, and M. Gabbrielli, “Solving xcsp problems by using gecode,” *University of Bologna*, 2011.
- [26] D. A. N. de Estadística, “Conceptos básicos de la organización territorial de colombia,” tech. rep., DANE, 2012.
- [27] B. V. del Banco de la República, “Listado de municipios de colombia-ayuda de tareas sobre geografía,” tech. rep., Banco de la República, 2005.
- [28] D. Buytaert., “Drupal open source cms.” <http://drupal.org/>, 2012. Noviembre 2012.
- [29] G. R. C. f. A. I. Saarland University, Swedish Institute of Computer Science, “The mozart programming system.” <http://www.mozart-oz.org/>, 2012. Noviembre 2012.
- [30] S. Pinagapany and A. Kulkarni, “Solving channel allocation problem in cellular radio networks using genetic algorithm,” in *Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 3rd International Conference on*, pp. 239–244, jan. 2008.
- [31] S. K. G de K. F. (Kim F.) Man, K. S. Tang, *Genetic Algorithms: Concepts and Design*. Springer, 1999. Page 26.

Anexos

Ejemplo entrada XML

Ejemplo de una entrada de una banda con 5 canales, dos operadores presentes y dos operadores que solicitan asignación.

```
<?xml version="1.0" encoding="UTF-8"?>
<instance>
  <presentation nbSolutions="?" format="XCSP_2.1">
    Representacion entrada de problema gestion espectro radioelectrico
  </presentation>
  <dict>
    <entry key="GeograficAssignmentType">
      <i>1</i>
    </entry>
    <entry key="GeograficAssignmentID">
      <i>1</i>
    </entry>
    <entry key="FrequencyBand">
      <i>4</i>
    </entry>
    <entry key="FrequencyRank">
      <i>2</i>
    </entry>
    <entry key="NumberChannels">
      <i>5</i>
    </entry>
    <entry key="NumberPresentOperators">
      <i>2</i>
    </entry>
    <entry key="NumberOfOperatorWithRequirements">
      <i>2</i>
    </entry>
    <entry key="ChannelSeparation">
      <i>1</i>
    </entry>
    <entry key="PresentOperators">
      <list>
        <i>35</i>
        <i>16</i>
      </list>
    </entry>
    <entry key="OperatorsWithRequeriments">
      <list>
        <i>33</i>
        <i>35</i>
      </list>
    </entry>
    <entry key="Requeriments">
      <tuple>
        <i>
          <entry key="33">
            <i>2</i>
          </entry>
        </i>
        <i>
          <entry key="35">
            <i>3</i>
          </entry>
        </i>
      </tuple>
    </entry>
    <entry key="MaxAssignationsSubDivision">
      <tuple>
        <i>
          <entry key="33">
            <i>0</i>
          </entry>
        </i>
      </tuple>
    </entry>
  </dict>
</instance>
```

```

        </i>
        <i>
            <entry key="35">
                <i>0</i>
            </entry>
        </i>
    </tuple>
</entry>
<entry key="ChannelAssignInDivisions">
    <list>
        <i>0</i>
        <i>0</i>
        <i>0</i>
        <i>0</i>
        <i>0</i>
    </list>
</entry>
<entry key="ReservedChannels">
    <list>
        <i>0</i>
        <i>0</i>
        <i>0</i>
        <i>0</i>
        <i>0</i>
    </list>
</entry>
<entry key="DisabledChannels">
    <list>
        <i>0</i>
        <i>0</i>
        <i>0</i>
        <i>0</i>
        <i>0</i>
    </list>
</entry>
<entry key="ChannelAssignment">
    <tuple>
        <i>
            <entry key="35">
                <i>
                    <list>
                        <i>0</i>
                        <i>0</i>
                        <i>0</i>
                        <i>0</i>
                        <i>0</i>
                    </list>
                </i>
            </entry>
        </i>
    </tuple>
    <i>
        <entry key="16">
            <i>
                <list>
                    <i>0</i>
                    <i>0</i>
                    <i>1</i>
                    <i>1</i>
                    <i>0</i>
                </list>
            </i>
        </entry>
    </i>
</tuple>
</entry>
<entry key="MaxChannelAssignmentByOperator">
    <i>10</i>
</entry>
</dict>
</instance>

```

Ejemplo salida XML

Ejemplo de una entrada de una banda con 5 canales, dos operadores que han obtenido asignación y dos soluciones.

```
<?xml version="1.0" encoding="UTF-8"?>
<solutions authorXML="Carlos_Andres_Delgado_Saavedra" >
  <head solution="Optima">
    <geograficAssignmentType> 1 </geograficAssignmentType>
    <geograficAssignmentID> 1 </geograficAssignmentID>
    <frequencyBand> 4 </frequencyBand>
    <frequencyRank> 2 </frequencyRank>
    <channelsNumber> 29 </channelsNumber>
    <operatorsNumber> 5 </operatorsNumber>
    <channelSeparation> 1 </channelSeparation>
    <numberOperatorPerChannel> 1 </numberOperatorPerChannel>
    <considerTop> true </considerTop>
    <staticAssignment> true </staticAssignment>
    <considerSeparation> true </considerSeparation>
    <numSolutions> 2 </numSolutions>
    <spacesCreated> 1 </spacesCreated>
    <spacesSucceeded> 2 </spacesSucceeded>
    <FDVariables> 144909 </FDVariables>
    <propagators> 148028 </propagators>
    <memoryUsage> 339992 </memoryUsage>
    <executionTime> 410 </executionTime>
  </head>
  <solution id="0">
    <costs>
      <blocksNumber> 6 </blocksNumber>
      <difChannelNumberMaxBlockFree> 23 </difChannelNumberMaxBlockFree>
      <channelNumberUseless> 6 </channelNumberUseless>
      <totalCost> 1155 </totalCost>
    </costs>
    <report>
      <operator name="16">
        <channels>
          <channel ID="1"> 0 </channel>
          <channel ID="2"> 0 </channel>
          <channel ID="3"> 1 </channel>
          <channel ID="4"> 1 </channel>
          <channel ID="5"> 0 </channel>
        </channels>
      </operator>
      <operator name="18">
        <channels>
          <channel ID="1"> 1 </channel>
          <channel ID="2"> 0 </channel>
          <channel ID="3"> 0 </channel>
          <channel ID="4"> 0 </channel>
          <channel ID="5"> 0 </channel>
        </channels>
      </operator>
    </report>
  </solution>
  <solution id="1">
    <costs>
      <blocksNumber> 7 </blocksNumber>
      <difChannelNumberMaxBlockFree> 24 </difChannelNumberMaxBlockFree>
      <channelNumberUseless> 7 </channelNumberUseless>
      <totalCost> 1254 </totalCost>
    </costs>
    <report>
      <operator name="16">
        <channels>
          <channel ID="1"> 0 </channel>
          <channel ID="2"> 0 </channel>
          <channel ID="3"> 1 </channel>
          <channel ID="4"> 1 </channel>
          <channel ID="5"> 0 </channel>
        </channels>
      </operator>
      <operator name="18">
        <channels>
          <channel ID="1"> 1 </channel>
          <channel ID="2"> 0 </channel>
          <channel ID="3"> 0 </channel>
          <channel ID="4"> 0 </channel>
          <channel ID="5"> 0 </channel>
        </channels>
      </operator>
    </report>
  </solution>
</solutions>
```

Diagrama de flujo aplicativo basado en programación por restricciones

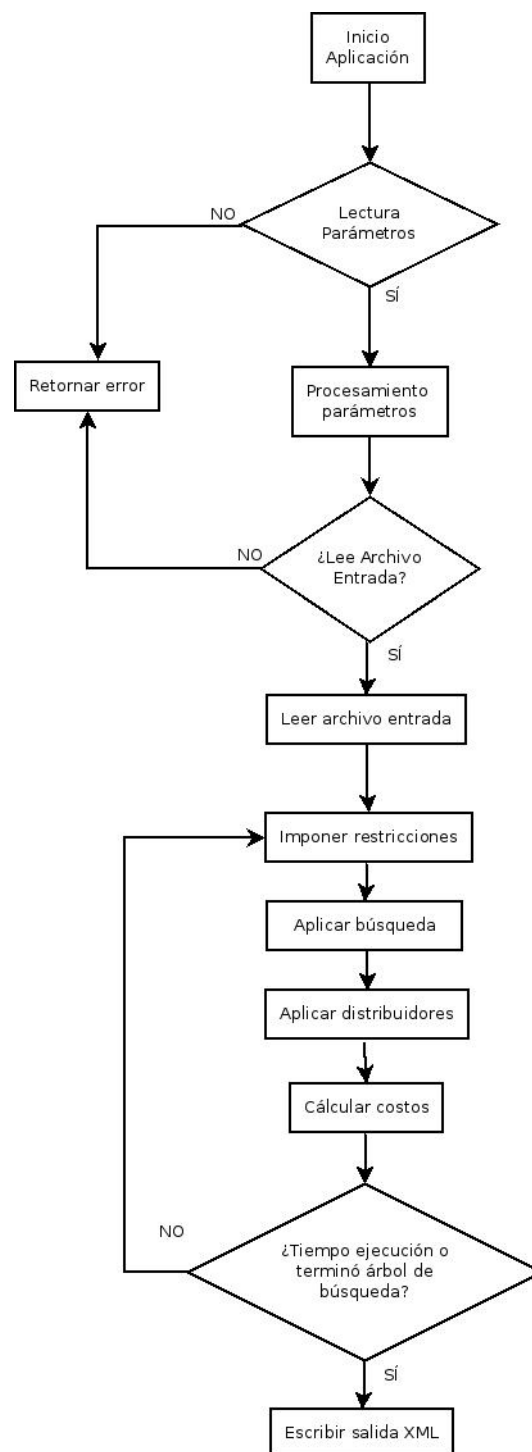


Figura 11.1: Diagrama de flujo del funcionamiento del aplicativo basado en programación por restricciones.

Modelo entidad relación

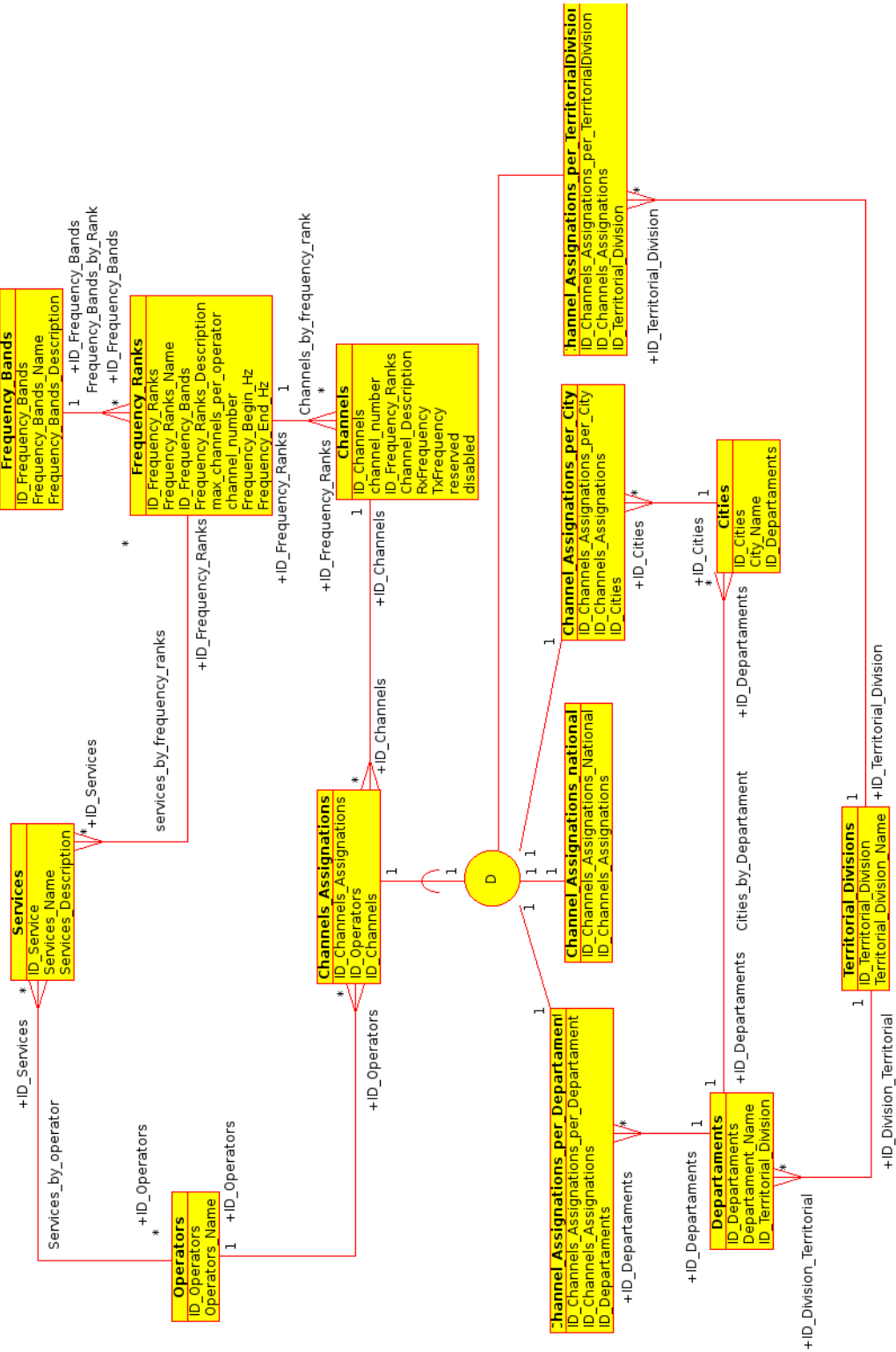


Figura 11.2: Modelo entidad relación del proyecto

Manuales de usuario

Consultas a base de datos

Existen dos tipos de consultas básicas: consultas por zona determinada o por rango de frecuencia. Usted puede elegir consultar a nivel nacional, en una entidad territorial, departamento o municipio. Las consultas requieren un poco de tiempo en desplegarse debido a que presentan información considerable, considere usar una conexión de al menos 516 kbps.

Selección de entidades territoriales

Usted puede seleccionar en que lugar se va realizar el proceso de asignación del espectro:

- Para nivel nacional no debe realizar ninguna selección geográfica
- A nivel regional seleccione la región específica
- A nivel departamental seleccione el departamento dentro de la región específica
- A nivel local seleccione el municipio dentro del departamento específico

Selección nacional

Seleccionar división territorial

Dirección Territorial Barranquilla

Seleccionar departamento

Figura 11.3: Filtro en la consulta por zona geográfica.

Usted puede elegir la zona geográfica que desea consultar.

Por operador

Selección operador

Emcalli

Consultar por operador

Figura 11.4: Filtro en la consulta por operador.

En esta pantalla usted puede elegir el operador al cual se va consultar su asignación.

Por Banda/Rango de frecuencia

Seleccione una banda de frecuencia

High Frequency HF 3 - 30 MHz

Seleccione un rango de frecuencia

Todos

Consultar por frecuencia

Figura 11.5: Filtro en la consulta por banda.

En esta consulta se puede elegir la banda que se desea consultar.

Gestión Operadores

Estas operaciones sólo las puede realizar un usuario con rol administrador de espectro.

Registrar Operador

Nombre operador:

Operador

Ingrese el nombre del operador, recuerde en no insertar un operador con nombre ya existente para evitar confusiones.

Agregar o quitar servicios

Gestión de servicios

Seleccione servicio:

Seleccionar

AGREGAR SERVICIO

Servicios que presta el operador

Nombre servicio	Acción
	GUARDAR

Lista de operadores registrados

Mostrar 10 registros

Buscar:

Acción

Operador	Acción
Agencia de vigilancia los carcinos LTDA	Editar
Agencia de vigilancia Lost	Editar
Agencia de vigilancia Titan	Editar
Amor Sterio	Editar
Bogota Alcaldia Mayor	Editar
Cain Ltda	Editar
Canal Cali	Editar
Canal Caracol	Editar
Canal Irravisión	Editar
Canal Tele antioquia	Editar
Operador	Acción

Mostrando desde 1 hasta 10 de 78 registros

Figura 11.6: Registrar operador

Para registrar un operador, usted debe definir sus parámetros y hacer clic en guardar.

Editar operador: Agencia de vigilancia los carcinos LTDA

Regresar

Nombre operador:

Agencia de vigilancia los carcinos LTDA

Ingrese el nombre del operador, recuerde en no cambiar el nombre por una ya existente.

GUARDAR

Agregar o quitar servicios

Gestión de servicios

Seleccione servicio:

Mi Servicio

AGREGAR

Servicios del operador

Mostrar 10 registros

Buscar:

Acciones

Nombre servicio	Acciones
Servicio de aficionados	Eliminar
Servicio de aficionados por satélite	Eliminar
Servicio de investigación espacial	Eliminar

Figura 11.7: Editar operador.

Para editar un operador, puede editar cualquier de los cambios y servicios asociados a el.

Gestión Servicios.

Estas operaciones sólo las puede realizar un usuario con rol administrador de espectro.

Añadir servicio

Nombre servicio:

Ingrese el nombre del servicio, recuerde en no usar uno ya existente.

Descripción servicio:

Ingrese la descripción del nuevo servicio

Lista de servicios registrados

Mostrar registros Buscar:

Nombre servicio	Descripción	Acciones
Mi Servicio	Este es un servicio de prueba. Para testear las consultas a la BD	Editar
Servicio de aficionados	Servicio de radiocomunicación que tiene por objeto la instrucción individual, la intercomunicación y los estudios técnicos, efectuado por aficionados, esto es, por personas debidamente autorizadas que se interesan en la radiofrecuencia con carácter exclusivamente personal y sin fines de lucro.	Editar

Figura 11.8: Agregar servicio.

Para agregar un servicio, se debe definir un nombre y su descripción, finalmente hacer clic en guardar.

Editar Servicios

Nombre servicio:

Editar el nombre del servicio, recuerde en no usar uno ya existente.

Descripción servicio:

Edite la descripción del nuevo servicio

Figura 11.9: Editar servicio.

Aquí usted podrá editar los parámetros asociados de un servicio.

Gestión Rangos de Frecuencia

Estas operaciones sólo las puede realizar un usuario con rol administrador de espectro.

Seleccione banda

Por favor seleccione una banda de frecuencia y haga clic en seleccionar.

Selección banda:

Seleccione una banda de frecuencia para listar sus rangos asociados

Rangos de frecuencia asociados a banda: High Frequency HF

Añadir rango de frecuencia

Por favor haga clic en Mostrar/Ocultar para ver u ocultar el formulario para agregar un nuevo rango de frecuencia en la banda seleccionada.

Lista de frecuencias registradas

Por favor haga clic en Mostrar/Ocultar para ver u ocultar la lista de rangos de frecuencia asociados a la banda seleccionada.

Figura 11.10: Selección banda de frecuencia.

En esta pantalla usted podrá seleccionar la banda de frecuencia para realizar acciones sobre los rangos de frecuencia asociados a ella.

Rango de frecuencias:

Rango:

Máximo número de canales por operador:

Máximo número de canales permitidos por operador en la banda, debe ser mayor que 0.

Frecuencia inicial (Hz):

Ingrese la frecuencia inicial de la banda en Hz. Debe ser menor que la frecuencia final y mayor que 0.

Frecuencia final (Hz):

Ingrese la frecuencia final de la banda en Hz. Debe ser mayor que la frecuencia inicial.

Número de canales:

Indique el número de canales en el rango de frecuencias, una vez creada la banda podrá editarlos.

Separación mínima:

Indique la separación mínima entre canales de operadores diferentes.

Descripción rango de frecuencia:

Ingrese la descripción del nuevo rango de frecuencia

Descripción de canales rango de frecuencia:

Ingrese la descripción de los canales del rango, esto aplica para todos los canales pero puede ser editado posteriormente.

Ingrese las ecuaciones de frecuencias de transmisión de los canales

Frecuencia base Tx	Multiplicador	Frecuencia corrimiento	Divisor
<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>

Ingrese la fórmula de la frecuencia de transmisión de los canales. En todos los campos debe ingresar enteros mayores o iguales que 0, n es el número consecutivo del canal.

Ingrese las ecuaciones de frecuencias de recepción de los canales

Multiplicador	Divisor	Multiplicador	Frecuencia corrimiento	Divisor
<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>

Ingrese la fórmula de la frecuencia de recepción de los canales. En todos los campos debe ingresar enteros mayores o iguales que 0, n es el número consecutivo del canal.

Figura 11.11: Agregar banda de frecuencia.

En este campo usted podrá crear un nuevo rango de frecuencia, especificando cada uno de los datos del rango y las formulas que se aplican para el cálculo de las frecuencias de transmisión y recepción de cada uno de los canales del nuevo rango de frecuencia.

Lista de frecuencias registradas

Por favor haga clic en **Mostrar/Ocultar** para ver u ocultar la lista de rangos de frecuencia asociados a la banda seleccionada.

Mostrar/Ocultar Lista

Mostrar: 10 registros

Buscar:

Rango de frecuencia	Máximo número de canales por operador	Frecuencia inicial (Hz)	Frecuencia final (Hz)	Número de canales	Separación mínima	Descripción	Acciones
12230-13197 kHz	12	12230002	13197001	41	1	Transmisión para explotación dúplex en banda lateral única (canales de dos frecuencias) Banda de 12-14 MHz	Editar
16360-17408 kHz	10	16360000	17360000	56	1	Transmisión para explotación dúplex en banda lateral única (canales de dos frecuencias) Banda de 16 MHz	Editar
18823-19798 kHz	10	18823000	19798000	15	1	Transmisión para explotación dúplex en banda lateral única (canales de dos frecuencias) Banda de 18/19 MHz	Editar
22000-22853 kHz	10	22000000	22853000	53	1	Transmisión para explotación dúplex en banda lateral única (canales de dos frecuencias) Banda de 22 MHz	Editar

Figura 11.12: Lista de rangos de frecuencia registrados en una banda.

En esta pantalla, usted podrá elegir que rango de frecuencia desea editar.

Regresar

Rango de frecuencias:
12230 13197 kHz
Máximo número de canales permitido por operador en la banda, debe ser mayor que 0.

Máximo número de canales por operador:
12
Ingrese la frecuencia inicial de la banda en Hz. Debe ser menor que la frecuencia final y mayor que 0.

Frecuencia inicial (Hz):
12230002
Ingrese la frecuencia inicial de la banda en Hz. Debe ser menor que la frecuencia final y mayor que 0.

Frecuencia final (Hz):
13197001
Ingrese la frecuencia final de la banda en Hz. Debe ser mayor que la frecuencia inicial.

Separación mínima:
1
Indique la separación mínima entre canales de operadores diferentes.

Descripción rango de frecuencia:
Transmisión para explotación dúplex en banda lateral única (canales de dos frecuencias) Banda de 12-14 MHz

Ingrese la descripción del nuevo rango de frecuencia

Guardar

Figura 11.13: Editar rango de frecuencia.

La operación editar rango de frecuencia permite editar los parámetros asociados a un rango de frecuencia.

Mostrar: 10 registros

Buscar:

Número de canal	Descripción del canal	Frecuencia de transmisión (Hz)	Frecuencia de recepción (Hz)	¿Está reservado?	¿Está deshabilitado?	Acciones
1	Frecuencia estaciones costeras y barco Reservado	13077001	13077001	Si	Si	Editar
2	Frecuencia estaciones costeras y barco	13080000	13080000	No	No	Editar
3	Frecuencia estaciones costeras y barco	13080000	13080000	No	No	Editar
4	Frecuencia estaciones costeras y barco	13080000	13080000	No	No	Editar
5	Frecuencia estaciones costeras y barco	13080000	13080000	No	No	Editar
6	Frecuencia estaciones costeras y barco	13080000	13080000	No	No	Editar
7	Frecuencia estaciones costeras y barco	13080000	13080000	No	No	Editar
8	Frecuencia estaciones costeras y barco	13080000	13080000	No	No	Editar
9	Frecuencia estaciones costeras y barco	13080000	13080000	No	No	Editar
10	Frecuencia estaciones costeras y barco	13080000	13080000	No	No	Editar

Número de canal: Mostrando desde 1 hasta 10 de 10 registros

Figura 11.14: Lista de canales asociados a un rango de frecuencia.

Es la lista de canales asociados a un rango de frecuencia, si usted desea editar alguno de ellos debe hacer clic en *Editar*.

Ayuda

Regresar

Frecuencia transmisión (Hz):
13077001
Indique la frecuencia de transmisión del canal

Frecuencia recepción (Hz):
12230000
Indique la frecuencia de recepción del canal

¿Está reservado?
☒
Indique si el canal está reservado.

¿Está deshabilitado?
☒
Indique si el canal está deshabilitado.

Descripción canal:
Frecuencia estaciones costeras y barco Reservado

Ingrese la descripción del canal

Guardar

Figura 11.15: Editar canal.

Al hacer clic en editar, usted puede editar los parámetros asociados a un canal de un rango de frecuencia.

Gestión Entradas XML

Ingresar XML de entrada:
Choose file | No file chosen
Guardar

Archivos de entrada en el sistema

Mostrar: 10 registros

Buscar:

Archivo	Visualizar	Descargar	Acción
entrada.xml	Ver entrada	Descargar	Borrar
entradaGrande.xml	Ver entrada	Descargar	Borrar
entradaGrande2.xml	Ver entrada	Descargar	Borrar
entradaGrande3.xml	Ver entrada	Descargar	Borrar
entradaMenor.xml	Ver entrada	Descargar	Borrar

Mostrando desde 1 hasta 5 de 5 registros

Figura 11.16: Lista de entradas XML almacenadas en el sistema.

La gestión de entradas XML permite visualizar, descargar y borrar las entradas XML propias de un usuario.

Gestión Salidas XML

Ingresar XML de salida:
Choose file | No file chosen
Guardar

Archivos de salidas en el sistema

Mostrar: 10 registros

Buscar:

Archivo	Descargar	Visualizar	Acción
genetico.xml	Descargar	Ver solución	Borrar
salida30Hz.xml	Descargar	Ver solución	Borrar
salida7.xml	Descargar	Ver solución	Borrar
salidaVariaXML.xml	Descargar	Ver solución	Borrar

Mostrando desde 1 hasta 4 de 4 registros

Figura 11.17: Lista de salidas XML almacenadas en el sistema.

La gestión de salidas XML permite visualizar, descargar y borrar las salidas XML propias de un usuario.

Generador de entradas

En el generador de entradas, usted debe seleccionar la zona geográfica y la banda de frecuencia en donde desea realizar un requerimiento, una vez hecho éste proceso puede proceder a crear los requerimientos de la entrada.

Figura 11.18: Especificación de requerimientos.

Una vez haga la creación de requerimientos podrá generar la entrada XML y almacenarla en el sistema.

Actualizar Base de datos

Estas operaciones sólo las puede realizar un usuario con rol administrador de espectro.

Figura 11.19: Selección salida XML para actualizar base de datos.

Usted debe seleccionar un archivo de salida o ingresar uno, para registrar su información a la base de datos.

Figura 11.20: Selección entrada XML para actualizar base de datos.

Es importante aclarar que una asignación con datos incorrectos, puede comprometer la integridad de los datos de la base de datos.

Aplicativos

Existen dos aplicativos en el proyecto, por programación por restricciones y por algoritmo genético.

Figura 11.21: Aplicativo por programación por restricciones.

Figura 11.22: Flexibilidad y debilidad de restricciones.

En la flexibilidad se pueden definir que restricciones no se toman en cuenta y decidir cuantas iteraciones se van a realizar.

Los resultados se despliegan mostrando la información general de la ejecución, un listado de soluciones encontradas y el detalle de que cada solución.

Figura 11.24: Información general acerca de una salida XML.

Figura 11.24: Información general acerca de una salida XML.

Figura 11.24: Información general acerca de una salida XML.

Figura 11.24: Información general acerca de una salida XML.