ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE CIENCIAS NATURALES Y MATEMÁTICAS

DEPARTAMENTO DE MATEMÁTICAS

PROYECTO DE GRADUACIÓN

PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

"MAGÍSTER EN CONTROL DE OPERACIONES Y GESTIÓN LOGÍSTICA"

TEMA

USO DE METAHEURÍSTICA PARA LA ASIGNACIÓN ÓPTIMA DE RECURSOS EN LA PLANIFICACIÓN DE LA ATENCIÓN DE VUELOS EN EL AEROPUERTO INTERNACIONAL JOSÉ JOAQUÍN DE OLMEDO (AIJJO) POR UNA EMPRESA PRIVADA

AUTOR:

Viviana Elizabeth Alcívar Millán

Guayaquil – Ecuador

AÑO 2016

DEDICATORIA

A Dios, mi inspiración, mi guía, mi luz y fuente de sabiduría.

A mis padres que han sido el pilar fundamental desde mi niñez, hasta el día de hoy. A mi madre quien ha sido ejemplo te entereza, ahínco, perseverancia y humildad, ese ser sublime que me da muestras cada día de que una verdadera mujer cultiva su inteligencia y su alma para el servicio de los demás.

A ti mi amor que serás mi esposo y quien ha sido mi compañero y soporte en este tiempo de maestría, dándome día a día las fuerzas para continuar y siendo calma y descanso cuando el cuerpo daba señales de cansancio y mi corazón se encontraba abatido. Te amo.

Viviana Alcívar M.

AGRADECIMIENTOS

A Dios por permitir concluir esta larga travesía con éxito.

A mi familia, por su amor infinito.

A Natacha quien fue soporte fundamental en el desarrollo de este proyecto y mi guía intelectual.

A mis amigas: Paulina, Brenda y Fabiola por su apoyo incondicional.

A Mi Amor, Richard, por sacrificar sus fines de semana, feriado, sus ratos libres, simplemente gracias por amarme.

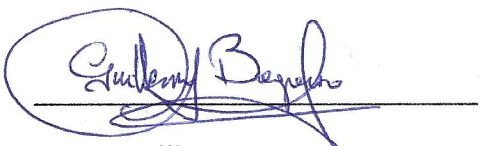
Viviana Alcívar M.

DECLARACIÓN EXPRESA

La responsabilidad por los hechos y doctrinas expuestas en este Proyecto de Graduación, me corresponde exclusivamente; el patrimonio intelectual del mismo, corresponde exclusivamente a la **Facultad de Ciencias Naturales y Matemáticas, Departamento de Matemáticas** de la Escuela Superior Politécnica del Litoral.

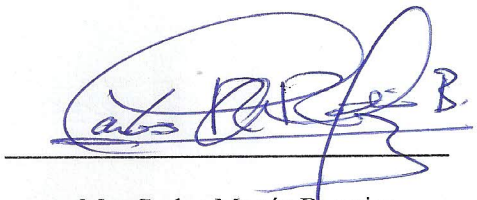
Ing. Viviana Alcívar Millán

TRIBUNAL DE GRADUACIÓN



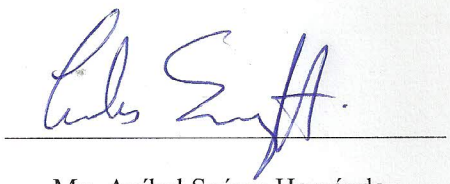
Mg. Guillermo Baquerizo

PRESIDENTE DEL TIBUNAL



Mg. Carlos Martín Barreiro

DIRECTOR DE POYECTO



Mg. Aníbal Suárez Hernández

VOCAL DEL TRIBUNAL

AUTOR DEL PROYECTO

Ing. Viviana Alcívar Millán

GLOSARIO

- SS: Abreviatura Scatter Search
- BD: Abreviatura Búsqueda Dispersa
- OC: Abreviatura Optimización Combinatoria
- CBR: Abreviatura Case Based Reasoning
- AIJJO: Abreviatura Aeropuerto Internacional José Joaquín de Olmedo
- AM: Abreviatura Algoritmo Memético

ÍNDICE DE FIGURAS

ÍNDICE TABLAS

ÍNDICE GENERAL

RESUMEN

Este proyecto está basado en la elaboración de una metaheurística, que ayude a realizar la planificación diaria para el proceso de aprovisionamiento de catering por la empresa privada, por motivos de confidencialidad y ética laboral se la denominará empresa ABC, de forma más simplificada y optimizando el uso de recursos disponibles limitados.

Actualmente en el AIJJO la empresa ABC da aprovisionamiento de servicios de catering aéreo y misceláneos a un promedio de 9 vuelos internacionales y 8 vuelos nacionales diariamente. En el caso de los vuelos internacionales existe un itinerario mensual que se realiza de forma manual con la información confirmada de los distintos clientes (aerolíneas varias/demás), y en el caso de los vuelos nacionales la información es recibida el día anterior por parte del cliente. Con esta información completa, el supervisor de turno procede a realizar manualmente la planificación de la operación tomando en cuenta el personal disponible (choferes y abordadores) y vehículos en uso (highloaders /camionetas). Es decir qué recurso se asignará para atender cada vuelo/ entrega.

El tiempo que utiliza el supervisor encargado es de 45 minutos cada cambio de turno. Existen 3 turnos para los supervisores que cubre la operación diaria:

- Turno E: 05:00 - 14:00
- Turno M: 12:00 -21:00
- Turno W: 21:00 - 05:30

Tomando en consideración esta información, se estaría dedicando 2 horas y 30 minutos a planificación, una planificación que es manual y que se basa en el puro requerimiento de atención al vuelo sin cumplir un objetivo de optimización de uso de recursos; que en este caso es representado por su coste/Km/hora hombre.

Con este proyecto se busca generar una algoritmo que sea aplicable en los distintos aeropuertos con una N cantidad de vuelos y con recursos limitados varios, ya que si tomamos en referencia que para la cantidad de 17 vuelos se necesita 2.50 horas de planificación. En otros aeropuertos como el Aeropuerto Internacional Jorge Chávez (LIMA –PERÚ) el mismo que maneja una media de 180 vuelos diarios, la planificación manual sería una tarea caótica casi imposible de realizarla en el mismo día, ya que el proceso tomaría alrededor de 23 horas sin tomar en consideración la dificultad para la asignación óptima de recursos dada la cantidad de vuelos a atender.

Una vez analizada la naturaleza de los datos se resuelve que se desarrollará un algoritmo de búsqueda dispersa (Scatter Search), el mismo que tomará los datos ingresados y a partir de ello comenzará a generar soluciones óptimas. Si bien es cierto por la misma naturaleza del algoritmo, no se evalúa la totalidad de la población del conjunto de soluciones, pero si analiza un conjunto más pequeño de posibles soluciones óptimas. Dando como resultado la obtención de soluciones óptimas en el rango deseado.

El programa será presentado en una aplicación de C# ya que es un derivado de C optimizado y elaborado para el sistema operativo Windows que es el más usado por las computadoras.

OBJETIVO GENERAL

El objetivo principal de este proyecto es realizar una planificación óptima para el uso de recursos en el proceso de abastecimiento de vuelos. Es decir que los vehículos sean asignados a los distintos vuelos sin que se use el mismo camión para atender dos vuelos en el mismo momento y a su vez que los empleados sean asignados a vuelos que estén dentro de su jornada de trabajo.

Para ello se ha investigado el estado del arte de la programación de planificación basado en las herramientas que brinda la optimización combinatoria.

OBJETIVOS ESPECÍFICOS

- Desarrollar un algoritmo Scatter Search, para el proceso de planificación.
- Lograr la implementación del algoritmo desarrollado a través de una herramienta computacional.
- Usar el lenguaje C+ para el desarrollo de la herramienta computacional.
- Disminuir el tiempo usado para la planificación diaria con el uso de la aplicación computacional del algoritmo.
- Evaluar la eficiencia de los resultados obtenidos contraponiéndolos con la realidad de la operación diaria y las variaciones que pueden sufrir los itinerarios de vuelos planificados.
- Lograr el uso de la herramienta de forma diaria en el proceso de planificación de la empresa ABC, en su tarea de asignación de recursos.
- Proponer mejoras en base a los resultados obtenidos a través del uso del algoritmo, tal que no solo se optimice el uso del recurso, sino que se evalúe la mejora en la productividad de los recursos.

CAPÍTULO 1
REVISIÓN BIBLIOGRÁFICA

1.1. USO DE LA OPTIMIZACIÓN COMBINATORIA PARA LA RESOLUCIÓN DE PROBLEMAS COMPLEJOS

Podemos encontrar un sin número de problemas de optimización en los diferentes campos: industrial, empresarial, computacional, económico, operacional, etc. Cada una de los campos citados anteriormente tiene un sin número de problemas específicos inherentes a la actividad que desarrollan, pero sin embargo todos tienen la misma particularidad, complejidad para su resolución.

La parte matemática que aborda estos problemas es la Programación Matemática, la misma que dependiendo de las características de las variables y linealidad de sus ecuaciones se puede clasificar en: Programación Lineal o Programación No Lineal.

Por otro lado si las variables toman únicamente valores enteros, hablamos de Programación Entera y si por el contrario asumen variables continuas, estamos hablando de Programación Continua.

Mientras que si los objetivos a optimizar son varios, estamos hablando de Programación Multiobjetivo. Así mismo, hablamos de Programación Determinística, siempre y cuando los parámetros que describen el problema asumen valores fijos; en el caso de tener valores aleatorios, se habla de Programación Estocástica.

Otra rama de la Programación matemática, es la Optimización Combinatoria, que trata de resolver problemas de optimización caracterizados por tener un número finito de soluciones.

Tomando en cuenta que la optimización combinatoria tiene como principal objetivo encontrar la mejor solución posible, en muchos casos este proceso de búsqueda se ve acotado por la factibilidad de ejecución de esa solución y el tiempo en que se llega a esa solución, por ello es necesario incursionar en métodos no tradicionales que nos permita hallar no necesariamente la mejor solución, pero sí un conjunto de buenas soluciones, en un periodo de tiempo mucho más corto.

Para ello la OC ha incursionado en otro tipo métodos como lo son: las heurísticas y metaheurísticas. Actualmente además de estos métodos no convencionales se está usando una nueva tecnología de Razonamiento Basado en Casos o CBR (Case Based Reasoning).

Si bien es cierto en primera instancia el desarrollo de la optimización combinatoria se centró en dar solución a problemas informáticos, ha ido ganando terreno en la solución de problemas operacionales dentro de las empresas que día a día han visto la necesidad de aumentar la eficiencia en sus procesos, ya que los recursos se escasean día a día y se encarecen.

"En un problema combinatorio de optimización se desea encontrar un orden específico sobre un conjunto de elementos discretos (Aarts y Lenstra, 2003; Sait y Youssef, 1999)"

1.2. PROGRAMACIÓN ESTOCÁSTICA

La programación estocástica trata problemas en los que algunos de los parámetros son variables aleatorias. Se considera un problema de Programación estocástica el siguiente:

$$\begin{aligned} &\min_{x} \tilde{g}_0\left(x, \tilde{\xi}\right), \\ &\text{sujeto a :} \\ &\tilde{g}_i\left(x, \tilde{\xi}\right) \leq 0,\ i = 1, 2, ..., m, \\ &x \in D, \end{aligned}$$

Fundamentalmente existen dos tipos de modelos en Programación Estocástica:

• Modelos "esperar y ver" ("wait and see") o modelos de programación estocástica pasiva.- estos se basan en la suposición de que el factor de decisión está en la capacidad de esperar a que se produzca la realización de las variables aleatorias y tomar la decisión con información completa de dicha realización. Con esto el problema se convierte en determinístico y es posible encontrar el valor óptimo de las variables de decisión con las técnicas habituales de programación matemática determinística.

En ocasiones puede ser de interés el conocer la distribución de probabilidad del valor objetivo óptimo o algunos de sus momentos (valor esperado o varianza) antes de conocer la realización de sus variables aleatorias. Dichos problemas reciben el nombre problemas de distribución.

• Modelos "aquí y ahora" ("here and now") o modelos de programación estocástica activa.- en estos modelos el factor de decisión toma la decisión sin el conocimiento de la realización de las variables aleatorias, sin que por ello queden afectadas las distribuciones de probabilidad de las mismas.

1.3. ORIGEN DEL USO DE LA HEURISTICA PARA LA OPTIMIZACIÓN DE RECURSOS

En muchas ocasiones los métodos convencionales no permiten la solución a ciertos problemas de optimización, esto se debe al grado de complejidad de los mismos. El uso de los diferentes métodos heurísticos han permitido a lo largo del tiempo obtener soluciones buenas que nos permiten solucionar problemas en un período de tiempo relativamente corto. Estos métodos en su mayoría ofrecen buenas soluciones aplicables a la realidad.

A lo largo de los años se han ido creando un gran número de técnicas que han permitido facilitar el proceso de búsqueda de la "mejor solución", pero en muchas ocasiones está búsqueda ha resultado agobiante y hasta cierto punto sin resultados positivos, he aquí el origen de la implementación de ciertas técnicas que si bien cierto no arrojan la mejor solución nos permiten obtener buenas soluciones.

El término heurístico proviene de proviene del griego heuriskein, que significa hallar ó inventar; la Real Academia Española (2012) lo define como la técnica de la indagación y del descubrimiento; o como la manera de buscar la solución de un problema mediante métodos no rigurosos, es decir, reglas empíricas. En ciencia, la idea más genérica del término heurístico está vinculada con la tarea de resolver inteligentemente problemas reales usando el conocimiento disponible.

1.3.1 CONCEPTO DE HEURISTICA DE ACUERDO A VARIOS AUTORES

Kahneman y Tversky.- Heurística Un grupo de atajos cognitivos, reglas empíricas que reducen el esfuerzo requerido, no necesariamente aumenta la calidad o exactitud de las decisiones tomadas.

Journal of Heuristic.- Un método heurístico (también llamado algoritmo de aproximación, un procedimiento inexacto o simplemente una heurística) es un bien definido conjunto de pasos para una identificación rápida de una muy buena solución para un problema dado, donde una solución es un conjunto de valores para un problema desconocido y la calidad es definida por una métrica de evaluación o criterio. El propósito de un método heurístico es identificar las soluciones de los problemas cuando el tiempo de resolución es más importante que la calidad de la solución o el conocimiento de la calidad.

Maroto et al. (2002).- La define como las técnicas o procedimientos informales para resolver problemas, basados en la creatividad, la intuición, el conocimiento o la experiencia para hallar buenas soluciones o mejorar una existente; además, indican que los mismos surgen como la única alternativa en aquellos casos cuyos modelos son muy complejos o intratables desde el punto de vista de los medios computacionales disponibles.

En este aspecto es muy importante recordar que obtenemos una buena solución, necesariamente no es la mejor solución, ya que la complejidad de los problemas los hace no viables al momento de tratar de resolverlos. Los métodos heurísticos / metaheurísticos tienen como objetivo la relajación de estos problemas analizando la importancia de cada uno de los criterios, asignando un peso ponderado a cada uno de ellos.

1.4. TÉCNICAS METAHEURÍSTICAS

Constituyen procedimientos de búsqueda que tratan de huir de los óptimos locales, orientando la búsqueda dependiendo de la evolución de dicho proceso a cada momento.

La aplicación de estas técnicas es especialmente interesante cuando hablamos de problemas de optimización combinatoria, ya sea que se trate de problemas con variables discretas o continuas.

La lógica de las técnicas metaheurísticas se basa en un punto de inicio o partida en el cual se toman en cuenta una solución o conjunto de soluciones que típicamente no es el óptimo. A partir de ello se obtienen otras similares, y luego entre estas nuevas soluciones se elige la que cumple con el criterio de selección, a partir de ello comienza un nuevo proceso y dicho proceso para cuando cumple con la condición previamente establecida.

1.4.1. CARÁCTERÍSTICAS DE LAS TÉCNICAS METAHEURÍSTICAS

- Desconocen cuándo deben parar, por esta razón se les debe indicar cuándo detenerse.
- Son algoritmos de aproximación, por tal motivo no garantizan la obtención de la solución óptima, pero sí una buena solución.
- En muchas ocasiones aceptan malos movimientos e incluso soluciones no factibles como paso requerido para acceder a nuevas regiones no exploradas.
- Son relativamente sencillas, lo que necesitan es una representación adecuada del espacio de soluciones, una solución inicial y un mecanismo para explorar el espacio de soluciones.

- Son generales, se pueden aplicar en la resolución de cualquier problema de optimización combinatoria.

1.5. ANTECEDENTES HISTÓRICOS METAHEURÍSTICA SCATTER SEARCH (BÚSQUEDA DISPERSA)

Los conceptos fundamentales y principios de este método fueron propuestos en la década de 1970**¡Error! No se encuentra el origen de la referencia.**, basados en la información desarrollada en los años 1960 referente a la combinación reglas de decisión y de restricciones. A diferencia de otros métodos evolutivos como los denominados "***algoritmos genéticos", scatter search*** (búsqueda dispersa) está basado en la premisa de diseños sistemáticos y métodos para la creación de nuevas soluciones que ofrecen beneficios significativos por encima de los obtenidos a través de recursos o procesos de aleatorización.

Es importante tomar en consideración que SS inicialmente era una de los componentes de la búsqueda Tabú (Tabu Search TBE). En inicios la mayoría de los escritos y textos sobre TS y sus aplicaciones no tomaron en cuenta el componente de búsqueda dispersa, esto originó que varios investigadores y autores comenzaran a ver a SS como un método independiente referente a procedimientos evolutivos.

La primera descripción formal del método fue publicada en 1977 por Fred Glover, en dicho artículo, Glover establece los principios de SS y determina la forma de funcionamiento del método, determinando que SS realiza una exploración sistemática sobre un conjunto de referencia (serie de buenas soluciones).

En 1998 Glover, publica una versión mucho más específica del método en donde se recopilan y simplifican muchas de las ideas de las publicaciones anteriores [2]. Esta versión es considerada actualmente el estándar de la búsqueda dispersa como método evolutivo.

Posteriormente el mismo Glover, y otros investigadores como Laguna y Martí indagan las implementaciones más recientes del método en la resolución de problemas de

Optimización combinatoria, y muestran las conexiones entre SS y "Path Relinking" (Re-encadenamiento de trayectorias).

Dado que este método se basa en realizar combinaciones y aplicar métodos de búsqueda local, se puede concluir que está incluido en los conocidos algoritmos meméticos. Los AM son metaheurísticas basadas en población. Esto significa que mantienen un conjunto de soluciones candidatas para el problema considerado. En la actualidad, existen implementaciones comerciales del método cómo:

- OptQuest. ***OptTeck Systems, Inc***.
- IBM ILOG CPLEX Optimizer. ***International Business Machines Corp. (IBM)***

1.6. FUNCIONAMIENTO SCATTER SEARCH (BÚSQUEDA DISPERSA)

Scatter Search está basado en la combinación de un grupo de soluciones que se encuentran almacenadas en un conjunto de referencia, denominado ***RefSet***. El objetivo de las combinaciones es generar centroides, centros geométricos, a fin de obtener nuevas soluciones de mayor calidad.

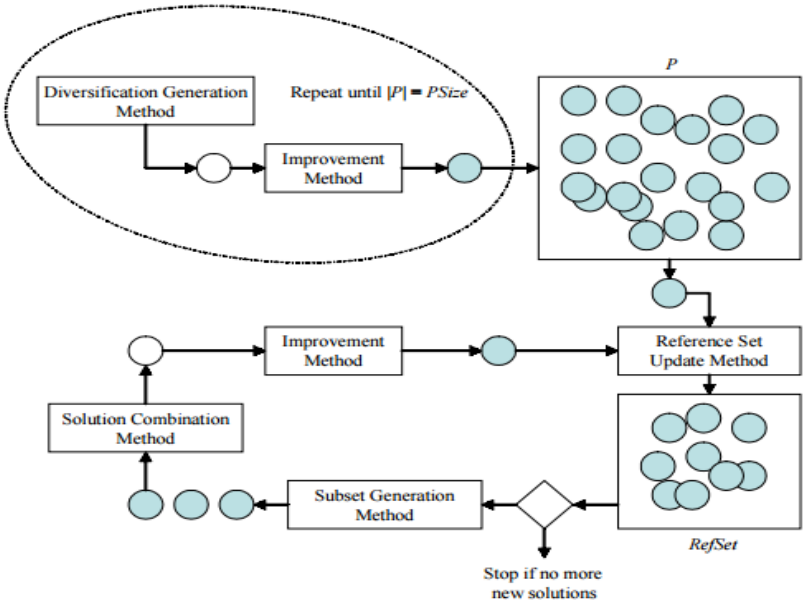


Figura 1.1: Esquema del Método Scatter Search

El tamaño del conjunto denominado ***RefSet*** puede variar durante la ejecución del algoritmo hasta llegar a un máximo de "***b***" elementos. Glover menciona que el tamaño del ***RefSet*** puede estar entre 20 y 40 soluciones, mientras que Martí y Laguna establecen que el tamaño adecuado es 10 soluciones. Para ambos escenarios el conjunto referencia debe constar de ***b/2*** soluciones de alta calidad y las sobrantes deben estar lo suficientemente alejadas para asegurar la dispersión.

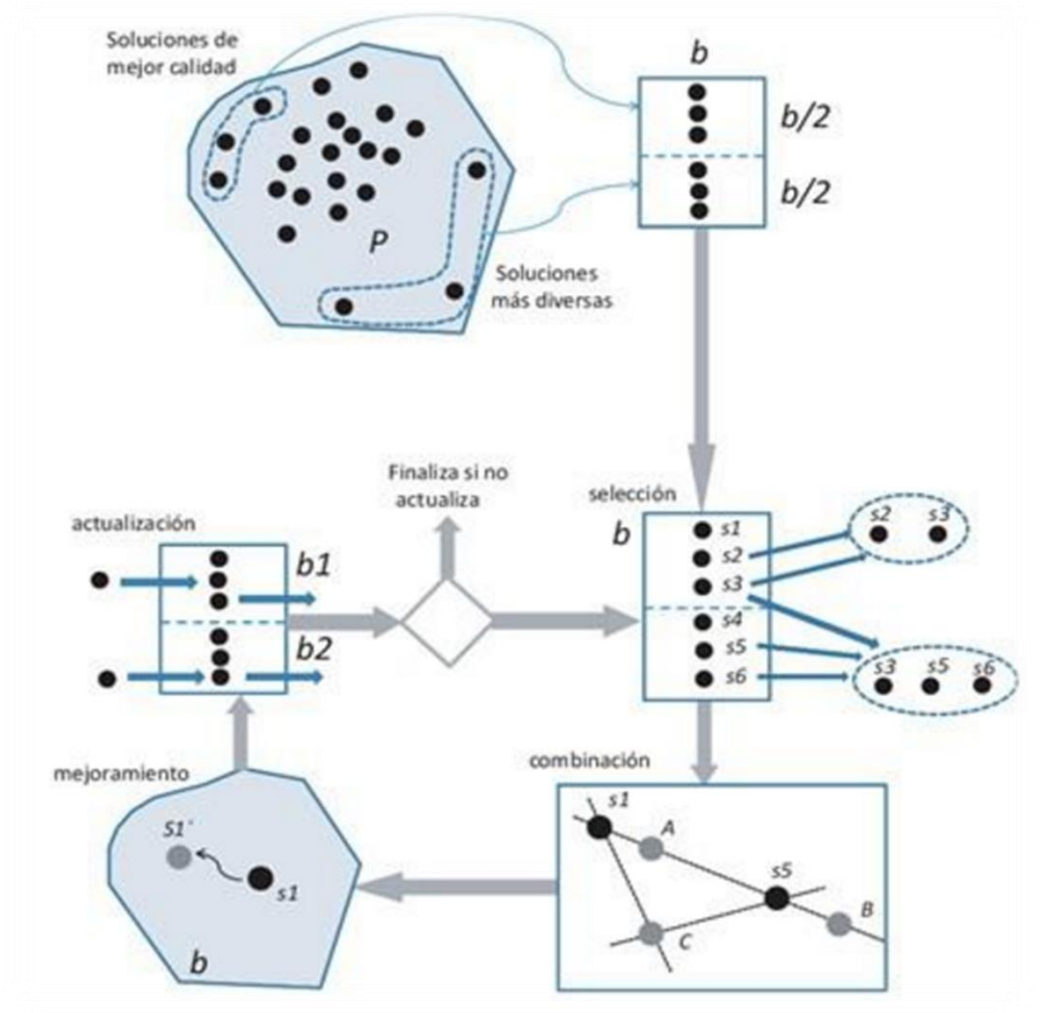


Figura 1.2: Componentes del algoritmo Scatter Search.

Por unanimidad de autores se usa la nomenclatura ***bNow***, para especificar el tamaño actual del conjunto referencia **(*RefSet*).** SS usa estrategias para la diversificación búsqueda e intensificación que han demostrado su eficacia en entornos variados.

Uno de los principales preceptos de SS consiste en cómo realiza la combinación de soluciones y cómo empiezan a explorar estas combinaciones. Un fundamento para entender la lógica detrás de estos procesos de combinación, consiste en analizar el desarrollo temprano que envuelve las estrategias de solución basado en la combinación de reglas de decisión o combinación de restricciones. La combinación de soluciones debe ser sistemática, por ello se utilizan pesos para ponderar las soluciones existentes en el conjunto referencia. SS usa combinaciones convexas y no convexas para asegurar de esta manera soluciones dispersas (nuevamente se enfatiza el principio de dispersión).

En necesario considerar que SS es un método evolutivo que integra heurísticas con procesos basados en la población para la combinación de soluciones en el espacio euclidiano[1]. Estos diseños que fueron reconocidos como casos especiales, representan los mecanismos fundamentales " cruzados (crossover) " que se introdujeron en la literatura de algoritmo genético aproximadamente una década después (incluyendo cruce uniforme, cruce de Bernoulli y cruce aritmético). Los componentes adicionales de SS y su generalización, vuelven a vincular el camino que reemplaza el espacio euclidiano por un nuevo espacio de vecindad, estos componentes a su vez han proporcionado nuevos métodos eficaces para la optimización no lineal y para la integración de optimización con simulación.

Scatter Search consta de 5 subrutinas usadas de manera repetitiva:

1. **Generación de diversificación.-** Se genera un conjunto inicial ***P*** que contiene p soluciones dispersas, pero no necesariamente factibles. Se usa una solución de prueba arbitraria (solución semilla) como entrada. Las demás soluciones se generan respetando las reglas definidas durante la etapa de diseño para que el proceso sea sistemático.

[1] ***Espacio Euclidiano:*** *El espacio euclidiano, denotado por R n, está definido por el conjunto (1.1) R n = {x = (x 1 , x2 , . . . , xn) : x i ∈ R}.*

2. **Método de mejora.-** Consiste en un método de búsqueda local usado para mejorar los elementos del ***RefSet*** y las soluciones combinadas previo a agregarlas a dicho conjunto. De existir soluciones no factibles, el método debe tener la capacidad de hacerlas factibles y posteriormente intentar mejorar su calidad.
3. **Método de actualización del RefSet.-** Este método cumple dos funciones esenciales: crear y actualizar el conjunto referencia.

   Fase de Creación: Debe buscar las b/2 soluciones de mayor calidad del conjunto inicial (P). Las restantes deben ser las soluciones de ***P*** que más se alejen de las mejores. Para la implementación de este método se debe definir una función de distancia. Glover usa la función de ***Hamming***[2]. Recordar que la lejanía entre las soluciones asegura de manera eficiente la dispersión.

   Fase de Actualización.- Aquí se actualiza el conjunto de referencia luego de haber combinado soluciones. Recordar que se no se debe superar el tamaño máximo del mismo. Es decir que si alguna solución es mejor a alguna existente en el conjunto referencia, se sustituye. A su vez recordar que para realizar la combinación de soluciones se puede trabajar bajo los siguientes criterios: comparar solo la calidad o comparar calidad y dispersión.
4. **Método de Generación de Subconjuntos**.- Aquí se generan subconjuntos a partir del conjunto referencia, dichos subconjuntos pueden estar compuestos por 2 o más elementos que luego se combinaran.

   Tipos de Subconjuntos:

   Tipo 1: Lo conforman 2 soluciones y el algoritmo de generación es el siguiente, tomando en consideración que X es el subconjunto.

[2] ***La distancia mínima de Hamming:*** *está dada por la siguiente ecuación: Dm= 2X+1. Para un código binario se refiere al número de bits que diferencian 2 palabras. Es decir ente 0000 y 1001 la distancia Hamming es 2. (Practical Data Communications for Instrumentation and Control) Pág. 110*

Tipo 2: Lo conforman 3 elementos y se deriva un subconjunto de tipo 1, más una solución combinada de mejor calidad.

Tipo 3: Lo conforman 4 elementos y se deriva de un subconjunto de tipo 3 más una solución combinada de mejor calidad.

Tipo 4: Contiene los mejores i elementos del ***RefSet*** desde i = 5 hasta b.

5. **Método de Combinación de Soluciones.-** Este método tiene la función de transformar un subconjunto dado de soluciones producidas, a través del método de generación de subconjuntos, en uno o más vectores de soluciones combinadas.

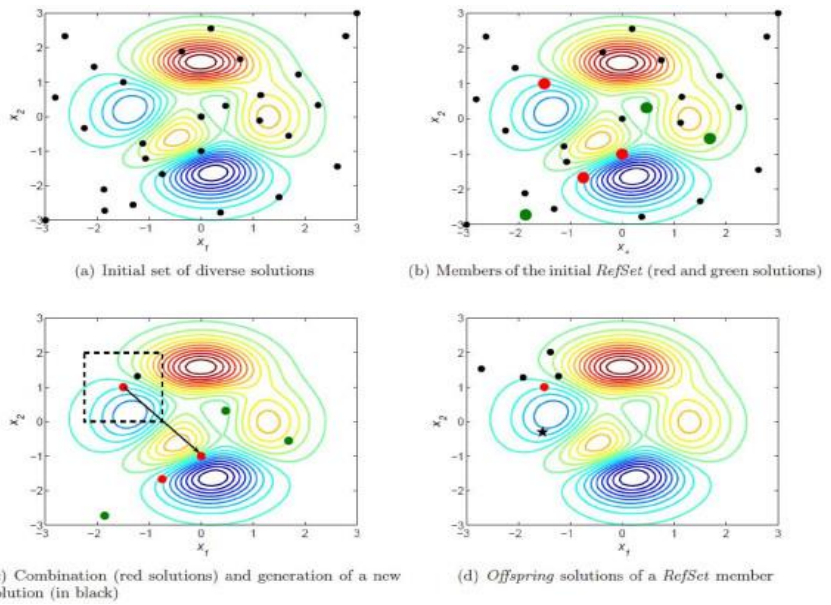


Figura 1.3: Representación Gráfica del funcionamiento de SS

Es muy importante notar que algoritmo para cuando al intentar combinar, vemos que no hay nuevos elementos en ***Refset*** (conjunto referencia), es decir la variable ***NuevaSolución*** está en "0".

Habitualmente se suele completar el proceso mediante un procedimiento de búsqueda local a fin de mejorar las soluciones.

1.6.1. ALGORITMO SCATTER SEARCH "ESTÁTICO"

1. Generar un conjunto inicial de ***P*** soluciones con un método Generador Diversificador

2. Mejorar estas soluciones con un método de mejora

3. Con estas soluciones construir el conjunto de referencia inicial

4. Repetir

    4.1. Obtener todos los subconjuntos de pares del conjunto de referencia

    4.2. Combinar estos subconjuntos para obtener nuevas soluciones

    4.3. Mejorar estas nuevas soluciones con el método de mejora

    4.4. Actualizar el conjunto de referencia con estas nuevas soluciones hasta que se estabilice (i.e. no se incluyan nuevas soluciones)

5. Si han trascurrido max_iter iteraciones (pasos 1-4) sin mejora, finalizar, caso contrario regresar al paso 1 (Reinicializar)

Para formar el conjunto de Referencia, (paso 3) se comienza seleccionando las b/2 soluciones de mayor calidad según la función objetivo.

Luego para añadir la otra mitad de soluciones se usa la siguiente función o criterio que mide la 'diversidad' de una solución candidata X a entrar con respecto a las que ya están en ***Refset***:

$$Difmin(X, Refset) = min \{dif(X, X') / X' \in Refset\};$$

Donde dif(X, X') = |X-X'|, es decir el número de elementos (localizaciones) de la solución X que no están en X'.

La actualización de ***Refset*** (paso 4.4.) se realiza tomando en cuenta la calidad de las soluciones. Esto quiere decir que se incorporan aquellas nuevas soluciones que mejoren la función objetivo de alguna de las soluciones existentes en ***Refset.***

El esquema detallado a continuación muestra cómo actúan las subrutinas o métodos descritos dentro de un esquema básico del algoritmo SS.

Algoritmo de Búsqueda Dispersa

1. *Comenzar con P = Ø. Utilizar el método de generación para construir una solución y el método de mejora para tratar de mejorarla; sea x la solución obtenida. Si* $x \notin P$ *entonces añadir x a P. (i.e.,* $P = P \cup x$*), en otro caso, rechazar x. Repetir esta etapa hasta que P tenga un tamaño prefijado.*
2. *Construir el conjunto de referencia* $R = \{ x^1, ..., x^b \}$ *con las b/2 mejores soluciones de P y las b/2 soluciones de P más diversas a las ya incluidas.*
3. *Evaluar las soluciones en R y ordenarlas de mejor a peor respecto a la función objetivo.*
4. *Hacer NuevaSolución = TRUE*

Mientras (NuevaSolución)

5. *NuevaSolucion = FALSE*
6. *Generar los subconjuntos de R en los que haya al menos una nueva solución.*
   Mientras (Queden subconjuntos sin examinar)
7. *Seleccionar un subconjunto y etiquetarlo como examinado.*
8. *Aplicar el método de combinación a las soluciones del subconjunto.*
9. *Aplicar el método de mejora a cada solución obtenida por combinación. Sea x la solución mejorada:*
   Si f(x) mejora a $f(x^b)$ y x no está en R)
10. *Hacer* $x^b = x$ *y reordenar R*
11. *Hacer NuevaSolucion = TRUE*

Figura 1.4: Método Algoritmo Búsqueda Dispersa

Se puede producir un segmento de línea, y de allí seleccionar a uno. De la misma forma se generan las nuevas soluciones 2, 3 y 4.

Dependiendo de la implementación el número de combinaciones ***Refset*** puede ser muy grande.

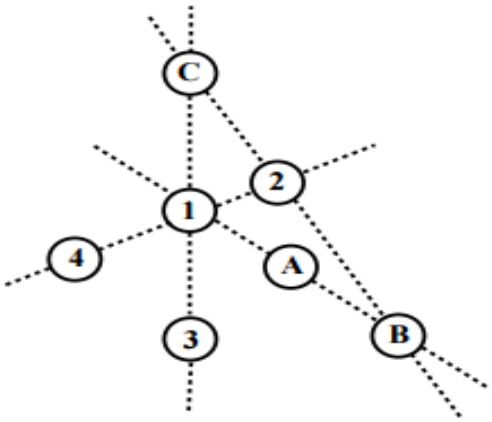


Figura 1.5: Ejemplo de generación de nuevos puntos con Scatter Search

1.7. PATH-RELINKING

Fue originalmente sugerido como un integrador de estrategias de intensificación y diversificación en el concepto de búsqueda TABÚ. Este enfoque constituye un generador de nuevas soluciones explorando trayectorias que conectan soluciones de alta calidad. Comienza en una de las soluciones y luego genera un camino en el espacio de vecindades que llevan hacia otras soluciones. Esto se logra seleccionando movimientos que incorporan atributos de las soluciones destino. En muchas ocasiones se introducen ideas de Búsqueda Tabú para poder realizar estos caminos y no repetir caminos anteriores.

Para la generación de los caminos deseados, únicamente es necesario llevar a cabo movimientos que cumplan con:

- Introducción progresiva de atributos de la solución final, empezando desde la solución inicial.
- Los roles de solución inicial y solución final son intercambiables. Cada solución puede ser inducida a moverse simultáneamente hacia la otra con el objetivo de generar combinaciones.

Path Relinking puede ser considerado una extensión del método de combinación de Scatter Search. En lugar de producir una nueva solución al combinar dos o más soluciones originales, PR genera caminos entre y hacia las soluciones seleccionadas en un espacio de vecindad. Esto debería dar a notar que el método de combinación usado en la búsqueda dispersa es un elemento problema dependiente, el cual es personalizado dependiendo del problema y de la representación de la solución.

La combinación de Scatter Search con Path Relinking, es muy usada para la resolución de Job Shop Schudeling Problems con duraciones de tareas inciertas.

Algoritmo 5 Algoritmo de Path Relinking

Obtener un $RefSet$ de b soluciones élite.
Evaluar las soluciones de $RefSet$ y ordenarlas de acuerdo con su función de fitness de tal manera que x^1 sea la mejor y x^b la peor.
$NuevasSoluciones = TRUE$.
mientras $NuevasSoluciones$ **hacer**
  Generar $NuevoSubconjunto$ con el *Método de generación de subconjuntos.*
  Hacer $NuevasSoluciones = FALSE$ y $Pool=\emptyset$.
  mientras $NuevoSubconjunto \neq \emptyset$ **hacer**
    Seleccionar el siguiente par $(x', x")$ de $NuevoSubconjunto$.
    Aplicar el método de enlazado para construir la secuencia $x' = x'(1), x'(2), ..., x'(r) = x"$ y añadir las soluciones a $Pool$.
    para i=1 hasta $i < r/NumImp$ **hacer**
      Aplicar el *Método de Mejora* a $x'(i * NumImp)$ y añadir soluciones a $Pool$
    fin para
    Aplicar el método de enlazado para construir la secuencia $x" = x"(1), x"(2), ..., x"(s) = x'$ y añadir las soluciones a $Pool$.
    para i=1 to $i < s/NumImp$ **hacer**
      Aplicar el *Método de Mejora* a $x"(i * NumImp)$ y añadir soluciones a $Pool$
    fin para
    para cada solución $x \in Pool$ **hacer**
      si $x \notin RefSet$ y $f(x) < f(x^b)$ **entonces**
        Hacer $x^b = x$ y reordenar $RefSet$.
        $NuevasSoluciones = TRUE$.
      fin si
    fin para
    Eliminar $(x', x")$ de $NuevoSubconjunto$.
  fin mientras
fin mientras

Figura 1.6: Método Algoritmo Path Relinking

CAPÍTULO 2
DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA

2.1 Descripción del problema

Actualmente en el AIJJO (Aeropuerto Internacional José Joaquín de Olmedo) la empresa ABC da aprovisionamiento de servicios de catering aéreo y misceláneos a un promedio de 9 vuelos internacionales y 8 vuelos nacionales diariamente. En el caso de los vuelos internacionales existe un itinerario mensual que se realiza de forma manual con la información confirmada de los distintos clientes (aerolíneas varias/demás), y en el caso de los vuelos nacionales la información es recibida el día anterior por parte del cliente. Con esta información completa, el supervisor de turno procede a realizar manualmente la planificación de la operación, tomando en cuenta: el personal disponible (choferes y abordadores) y vehículos en uso (highloaders /camionetas).

El tiempo que utiliza el supervisor encargado para realizar la planificación de la operación es de 45 minutos cada cambio de turno.

Existen 3 turnos para los supervisores que cubre la operación diaria:

- Turno E: 05:00 - 14:00 o Turno D: 04:00 - 13:00
- Turno O: 14:00 -22:30
- Turno W: 21:00 - 05:30

Tomando en consideración esta información, se estaría dedicando 2 horas y 30 minutos a planificación, una planificación que es manual y que se basa en el puro

requerimiento de atención al vuelo sin cumplir un objetivo de optimización de uso de recursos; que en este caso es representado por su coste/km/hora hombre.

Con este proyecto también se busca diseñar un algoritmo aplicable en los distintos aeropuertos con una N cantidad de vuelos y con recursos limitados varios, ya que si tomamos en referencia que para la cantidad de 17 vuelos se necesita 2.50 horas de planificación. En otros aeropuertos como el Aeropuerto Internacional Jorge Chávez (LIMA –PERÚ) el mismo que maneja una media de 180 vuelos diarios, la planificación manual sería una tarea caótica casi imposible de realizarla en el mismo día, ya que el proceso tomaría alrededor de 23 horas sin tomar en consideración la dificultad para la asignación óptima de recursos dada la cantidad de vuelos a atender.

2.2 Descripción de la operación y recursos usados

La franja horaria es de 24 horas. Se incluyen los 7 días de la semana, cubriendo la operación de la semana entera.

2.2.1. Recurso Vehículos

Se cuenta con 7 camiones

HL Nº	Modelo	Año	Largo (m)	*Operativo*	*Elevador Hidráulico*	*Refrigerado*
1	FTR-34M	2012	7	Y	Yes	N
2	FTR-34M	2012	7	Y	Yes	N
3	FTR-32M	2010	7	Y	Yes	N
4	F-700	1993	6	Y	Yes	N
5	F-700	1985	6	Y	Yes	N
6	F-600	1971	6	Y	Yes	N
7	FTR-34M	2015	6	Y	Yes	N

Tabla 2.1 Características de los camiones disponibles.

Nota: La planta se encuentra a 1.5km de la plataforma de aviones, por lo que el promedio de recorrido por operación es 4km. (1.5km de ida, 1.5km de vuelta y el

recorrido en plataforma a lo que se atiende la aeronave). Camiones 1, 2, 3, 7 usan combustible diésel; camiones 4, 5, 6 usan gasolina.

Adicional se cuenta con 2 camionetas: camioneta Mazda usa gasolina y camioneta Chevrolet usa diésel.

2.2.2. Recurso Humano

En la actualidad se cuenta con 6 asistentes (guías/abordadores) y 10 choferes (operadores de galley), Distribuidos en diferentes turnos. Los turnos pueden variar dependiendo de la necesidad de la operación.

	Last Minute Transport										
	Nombre del Supervisor										
	octubre, 2015	jue 1	vie 2	sáb 3	dom 4	lun 5	mar 6	mié 7	jue 8	vie 9	sáb 10	dom 11
	Name											
ASISTENTES	TOBAR ANTONIO	D	F	FD	D	D	O	P	P	F	FD	D
	JORGE BRAVO	P	P	P	P	FD	F	O	O	O	O	O
	VELIZ JONATHAN	F	D	D	D	P	P	F	F	D	D	D
	JIMENEZ JORGE	O	O	F	F	C	D	D	D	P	F	F
	DOMINGUEZ NELSON	F	F	V	V	V	V	V	V	V	V	V
	FERNANDEZ MARC.	D	C	D	FD	F	D	C	D	C	P	FP
CHOFERES	BURGOS OMAR	O	W	F	F	D	D	D	E	W	F	F
	CARRIEL LUIS	F	D	D	D	D	F	F	D	D	O	O
	CHAGUAY TITO	V	V	V	V	V	D	D	D	E	E	FD
	HERRERA LUIS	O	F	W	W	W	W	W	F	F	D	E
	LLERENA CHRISTIAN	D	O	O	FO	F	O	O	O	O	W	F
	MENDOZA ANGEL	W	F	P	P	P	P	F	F	D	D	W
	ORTEGA JOEL	P	P	F	D	E	E	E	F	F	F	D
	PALOMINO LEONCIO	E	E	E	E	F	F	L	O	O	O	O
	RIVAS RODOLFO	D	D	D	F	O	O	O	W	F	F	F
	VARGAS GERMAN	F	O	O	O	FO	F	P	P	P	P	P

Tabla 2.2 Características de los empleados disponibles

TURNO	ENTRADA	SALIDA	OBSERVACIONES DE ALIMENTACIÓN
C	3:00	11:30	(INCLUYE 30 MINUTOS DESAYUNO)
FC	3:00	11:30	(INCLUYE 30 MINUTOS DESAYUNO)
D	4:00	13:00	(INCLUYE 30 MINUTOS DESAYUNO Y 30 MINUTOS DE ALMUERZO)
FD	4:00	13:00	(INCLUYE 30 MINUTOS DESAYUNO Y 30 MINUTOS DE ALMUERZO)
O	14:00	22:30	(INCLUYE 30 MINUTOS CENA)
FO	14:00	22:30	(INCLUYE 30 MINUTOS CENA)
P	15:00	23:30	(INCLUYE 30 MINUTOS CENA)
FP	15:00	23:30	(INCLUYE 30 MINUTOS CENA)
Q	16:00	00:30	(INCLUYE 30 MINUTOS CENA)
FQ	16:00	00:30	(INCLUYE 30 MINUTOS CENA)
W	21:00	05:30	(INCLUYE 30 MINUTOS CENA)
FW	21:00	05:30	(INCLUYE 30 MINUTOS CENA)

Tabla 2.3 Descripción de turnos de trabajo actuales

2.2.3. Legislación laboral vigente a considerar

De acuerdo a la legislación laboral:

Art. 47.- De la jornada máxima.- La jornada máxima de trabajo será de ocho horas diarias, de manera que no exceda de cuarenta horas semanales, salvo disposición de la ley en contrario. El tiempo máximo de trabajo efectivo en el subsuelo será de seis horas diarias y solamente por concepto de horas suplementarias, extraordinarias o de recuperación, podrá prolongarse por una hora más, con la remuneración y los recargos correspondientes.

Art. 48.- Jornada especial.- Las comisiones sectoriales y las comisiones de trabajo determinarán las industrias en que no sea permitido el trabajo durante la jornada completa, y fijarán el número de horas de labor. La jornada de trabajo para los adolescentes, no podrá exceder de seis horas diarias durante un período máximo de cinco días a la semana.

Art. 49.- Jornada nocturna.- La jornada nocturna, entendiéndose por tal la que se realiza entre las 19H00 y las 06H00 del día siguiente, podrá tener la misma duración y dará derecho a igual remuneración que la diurna, aumentada en un 25%

1. Las horas suplementarias no podrán exceder de cuatro en un día, ni de doce en la semana; 2. Si tuvieren lugar durante el día o hasta las 24H00, el empleador pagará la remuneración correspondiente a cada una de las horas suplementarias con más un 50% de recargo. Si dichas horas estuvieren comprendidas entre las 24H00 y las 06H00, el

trabajador tendrá derecho a un 100% de recargo. Para calcularlo se tomará como base la remuneración que corresponda a la hora de trabajo diurno;

Además tomar en consideración SBU 2015 $354.00 y SBU 2015 $366.00

2.3 Procedimiento de entrega de servicios

El Supervisor de Operaciones supervisa y controla el cumplimiento de los respectivos procedimientos aplicados en la elaboración y preparación de los servicios solicitados, antes de que el producto ya procesado salga de la planta.

1. El Supervisor de operaciones, autoriza el abordaje de los servicios a los vehículos de transporte, (camiones o camionetas), y se procede a colocar los sellos de seguridad de la aerolínea contratante en las puertas posteriores del vehículo. En la planilla de Seguridad, el guardia registra el número del sello de seguridad antes de la salida del vehículo.
2. Los servicios son transportados a la plataforma. Deben ubicarse en la parte posterior o delantera de la aeronave según sea el caso. El vehículo estacionado se asegura con las patas y trancas.
3. Un representante de Seguridad de la aerolínea procede a romper el sello de seguridad de la puerta y autoriza la subida del cajón del camión. Además se encarga de revisar el contenido de trolleys, gabinetes y todo lo que se encuentre dentro del vehículo y sea parte del servicio.
4. Se abren las puertas del avión para la entrega de los servicios, a su vez se retiran los trolleys del servicio ya despachado. El personal responsable de la aerolínea y el personal de operaciones revisan y realizan un conteo de los servicios.
5. El vehículo es reubicado en la parte posterior o delantera de la aeronave siguiendo el procedimiento de seguridad del paso c) para el desembarque de los servicios restantes.
6. El personal responsable de la aerolínea firma la remisión como acuerdo de los servicios recibidos.
7. Se realiza el cierre final cuando Seguridad firma el registro con el número de sello de Seguridad empleado para el retorno del camión a la planta.
8. El vehículo retorna a la planta, donde el Guardia de Seguridad revisa y registra el número del Sello de Seguridad.
9. Los servicios se descargan y se procede a limpiarlos.

Los vehículos empleados para la entrega de servicios deben someterse a mantenimientos preventivos y correctivos para su uso.

2.3.1. Diagrama de flujo

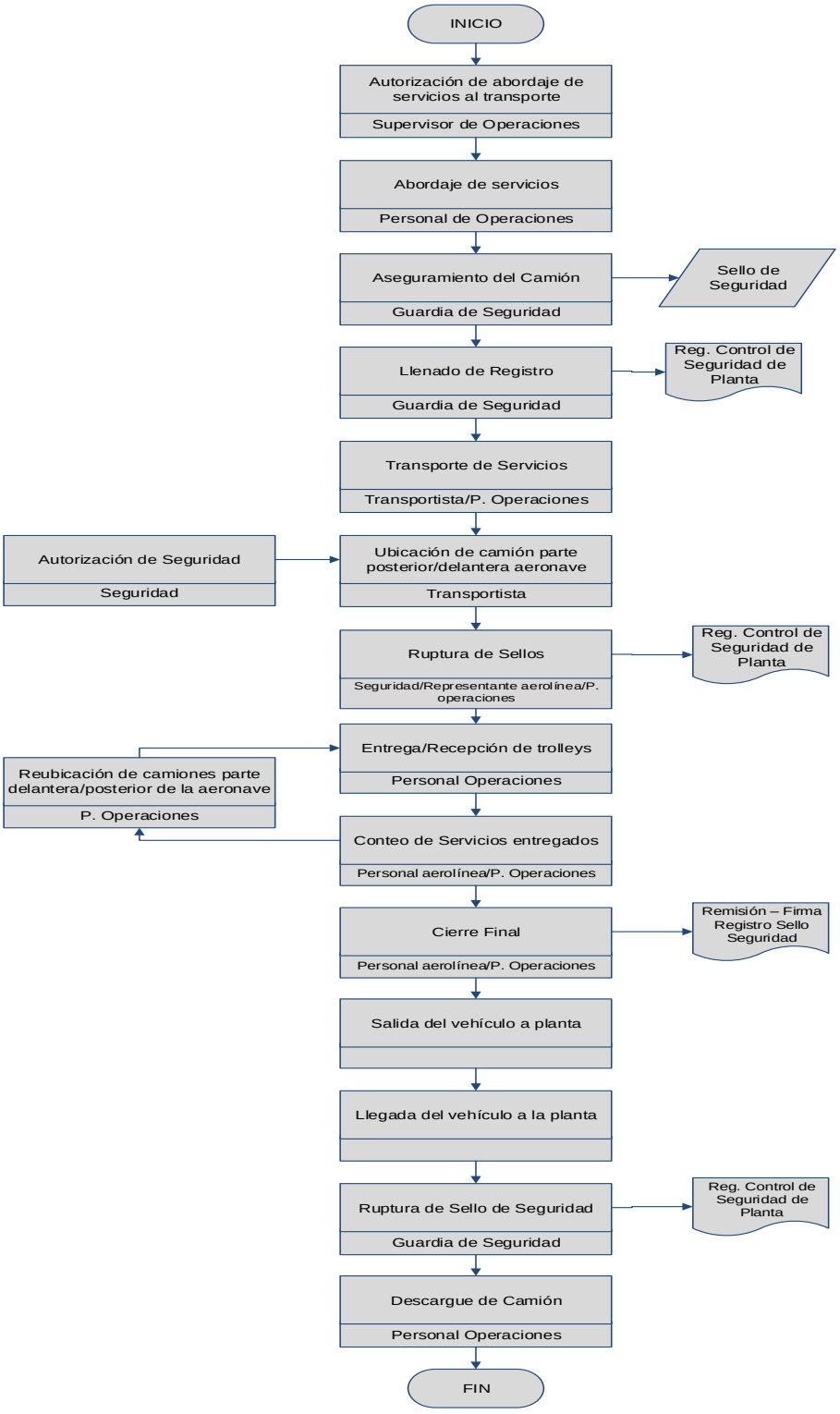


Figura 2.1 Flujo entrega de Servicio

MAPEO DE OPERACIÓN DIARIA

LUNES

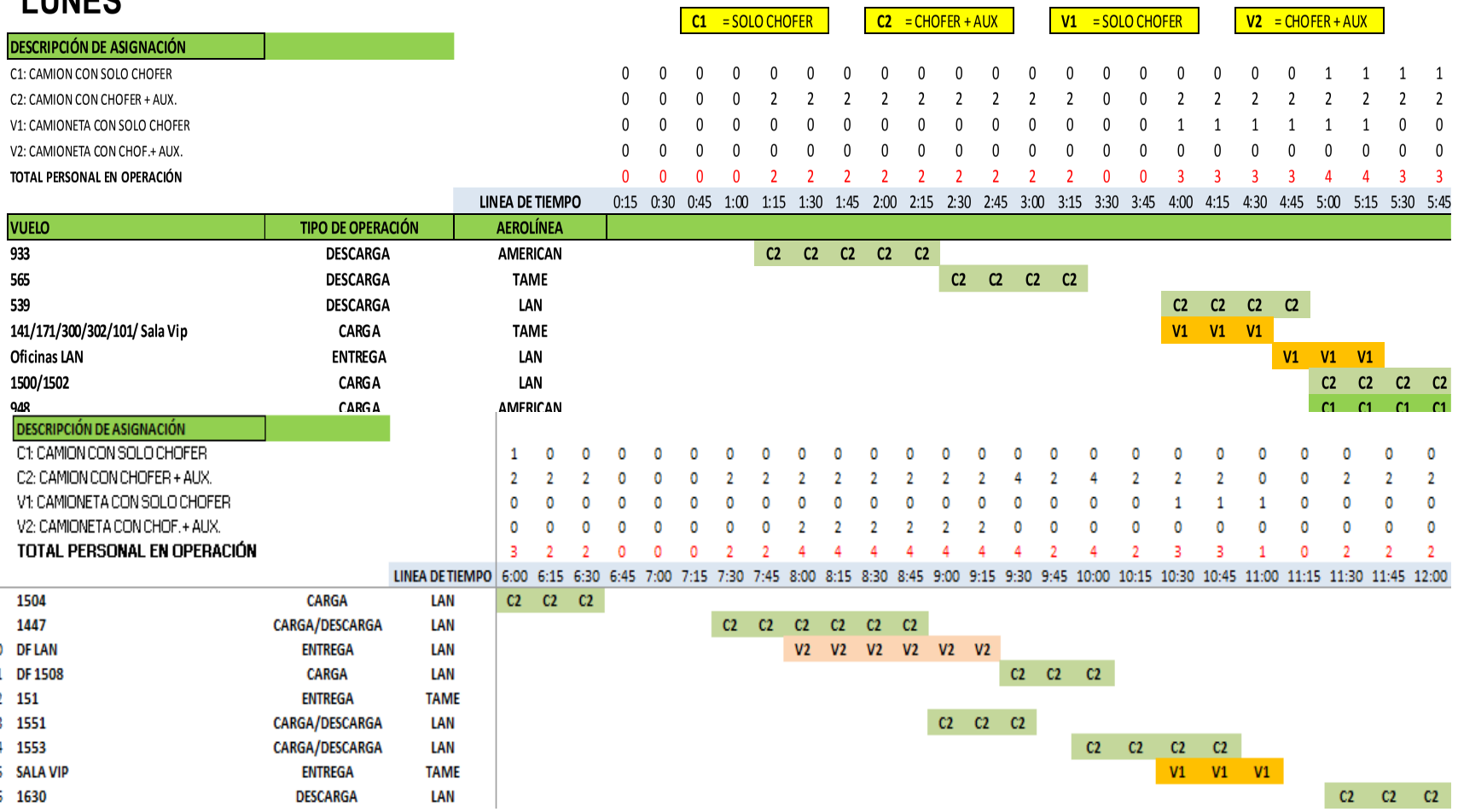


Tabla 2.4 Mapeo Operación Diaria Lunes

MAPEO DE OPERACIÓN DIARIA

LUNES

DESCRIPCIÓN DE ASIGNACIÓN	12:15	12:30	12:45	13:00	13:15	13:30	13:45	14:00	14:15	14:30	14:45	15:00	15:15	15:30	15:45	16:00	16:15	16:30	16:45	17:00	17:15	17:30	17:45	18:00
C1: CAMION CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.	4	2	2	2	0	0	0	2	2	2	2	0	2	4	4	4	2	4	4	4	4	4	4	2
V1: CAMIONETA CON SOLO CHOFER	0	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V2: CAMIONETA CON CHOF.+ AUX.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN	4	3	3	2	0	0	0	3	3	3	2	0	2	4	4	4	2	4	4	4	4	4	4	2
LINEA DE TIEMPO																								
17 538 CARGA LAN	C2	C2	C2	C2																				
18 1550 CARGA/DESCARGA LAN								C2	C2	C2	C2													
19 SALA VIP ENTREGA TAME								V1	V1	V1														
20 1552 CARGA/DESCARGA TAME													C2	C2	C2	C2								
21 1730 CARGA LAN														C2	C2	C2	C2	C2						
22 755 CARGA/DESCARGA KLM																		C2	C2	C2	C2	C2	C2	
23 755 CARGA/DESCARGA KLM																			C2	C2	C2	C2	C2	C2

DESCRIPCIÓN DE ASIGNACIÓN	18:15	18:30	18:45	19:00	19:15	19:30	19:45	20:00	20:15	20:30	20:45	21:00	21:15	21:30	21:45	22:00	22:15	22:30	22:45	23:00	23:15	23:30	23:45
C1: CAMION CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.	0	2	2	2	2	4	4	4	4	4	2	2	2	2	4	4	2	2	2	0	0	0	0
V1: CAMIONETA CON SOLO CHOFER	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V2: CAMIONETA CON CHOF.+ AUX.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN	0	3	3	3	3	4	4	4	4	4	2	2	2	4	4	4	2	2	2	0	0	0	0
LINEA DE TIEMPO																							
24 1514 ENTREGA LAN		C2	C2	C2	C2																		
25 177 ENTREGA TAME		V1	V1																				
26 109 ENTREGA TAME				V1	V1																		
27 6464 CARGA/DESCARGA IBERIA						C2	C2	C2	C2	C2													
28 6464 CARGA/DESCARGA IBERIA						C2	C2	C2	C2														
29 1446 DESCARGA LAN										C2	C2	C2	C2	C1									
30 927 DESCARGA AMERICAN														C1	C2	C2							
31 902 CARGA AMERICAN														C2	C2	C2							
32 1523 CARGA/DESCARGA LAN																	C2	C2	C2				

Tabla 2.5 Mapeo Operación Diaria Lunes

MAPEO DE OPERACIÓN DIARIA

MARTES

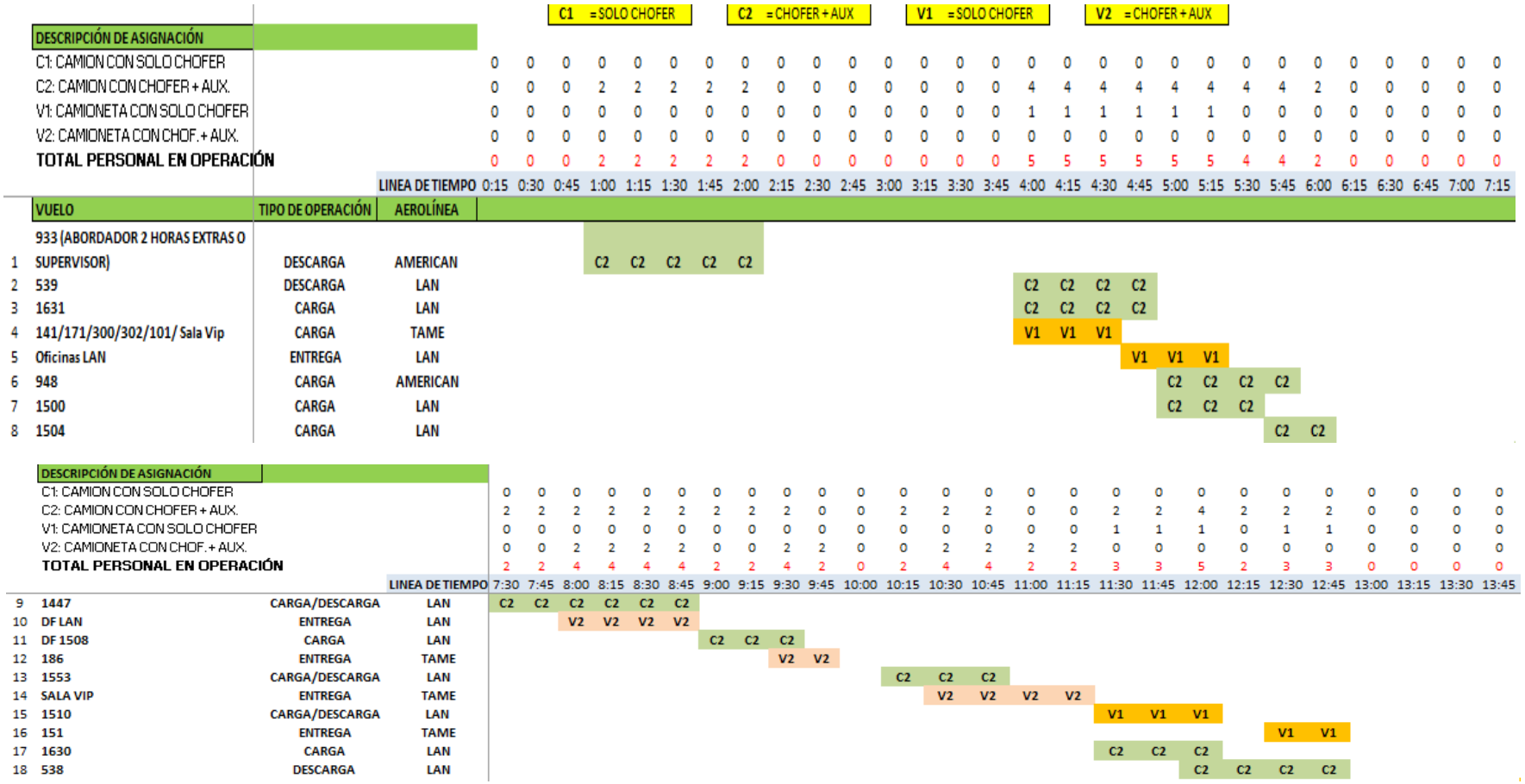


Tabla 2.6 Mapeo Operación Diaria Martes

MARTES

MAPEO DE OPERACIÓN DIARIA

DESCRIPCIÓN DE ASIGNACIÓN	14:00	14:15	14:30	14:45	15:00	15:15	15:30	15:45	16:00	16:15	16:30	16:45	17:00	17:15	17:30	17:45	18:00	18:15	18:30	18:45	19:00
C1: CAMION CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.	0	0	0	2	2	2	2	2	2	2	4	6	6	4	4	0	0	0	0	0	0
V1: CAMIONETA CON SOLO CHOFER	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	1	1
V2: CAMIONETA CON CHOF.+ AUX.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN	1	1	1	2	2	2	2	2	2	2	4	6	6	4	5	1	1	1	2	1	1
LINEA DE TIEMPO																					
19 SALA VIP ENTREGA TAME	V1	V1	V1																		
20 566 CARGA/DESCARGA TAME				C2	C2	C2	C2														
21 1552 CARGA/DESCARGA LAN								C2	C2	C2											
22 1560 CARGA/DESCARGA LAN											C2	C2	C2								
23 755 CARGA/DESCARGA KLM											C2	C2	C2	C2	C2						
24 755 CARGA/DESCARGA KLM												C2	C2	C2	C2						
25 177 ENTREGA TAME															V1	V1					
26 1516 ENTREGA LAN																	V1	V1	V1		
27 1518 ENTREGA LAN																			V1	V1	V1

DESCRIPCIÓN DE ASIGNACIÓN	19:15	19:30	19:45	20:00	20:15	20:30	20:45	21:00	21:15	21:30	21:45	22:00	22:15	22:30	22:45	23:00	23:15	23:30	23:45
C1: CAMION CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.	0	0	0	0	2	2	2	6	6	4	0	4	4	4	2	0	0	0	0
V1: CAMIONETA CON SOLO CHOFER	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V2: CAMIONETA CON CHOF.+ AUX.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN	0	0	1	1	2	2	2	6	6	4	0	4	4	4	2	0	0	0	0
LINEA DE TIEMPO																			
28 109 ENTREGA TAME			V1	V1															
29 1446 DESCARGA LAN					C2	C2	C2												
30 927 DESCARGA AMERICAN								C2	C2										
31 902 CARGA AMERICAN								C2	C2	C2									
32 1521 CARGA/DESCARGA LAN								C2	C2	C2									
33 1523 CARGA/DESCARGA LAN												C2	C2	C2					
34 1733 DESCARGA LAN												C2	C2	C2	C2				

Tabla 2.7 Mapeo Operación Diaria Martes

MIÉRCOLES

MAPEO DE OPERACIÓN DIARIA

C1 = SOLO CHOFER | C2 = CHOFER + AUX | V1 = SOLO CHOFER | V2 = CHOFER + AUX

DESCRIPCIÓN DE ASIGNACIÓN	0:15	0:30	0:45	1:00	1:15	1:30	1:45	2:00	2:15	2:30	2:45	3:00	3:15	3:30	3:45	4:00	4:15	4:30	4:45	5:00	5:15	5:30	5:45	6:00	6:15	6:30	6:45
C1: CAMION CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.	0	2	2	2	2	2	2	4	4	4	4	2	0	0	2	4	4	6	4	6	4	4	2	0	0	0	0
V1: CAMIONETA CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0
V2: CAMIONETA CON CHOF.+ AUX.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN	0	2	2	2	2	2	2	4	4	4	4	2	0	0	2	5	5	7	5	7	5	4	2	0	0	0	0

	VUELO	TIPO DE OPERACIÓN	AEROLÍNEA	0:15	0:30	0:45	1:00	1:15	1:30	1:45	2:00	2:15	2:30	2:45	3:00	3:15	3:30	3:45	4:00	4:15	4:30	4:45	5:00	5:15	5:30	5:45	6:00	6:15	6:30	6:45
1	933	DESCARGA	AMERICAN		C2	C2	C2	C2																					
2	565	DESCARGA	TAME										C2	C2	C2															
3	948	CARGA	AMERICAN																		C2	C2	C2							
4	539	DESCARGA	LAN															C2	C2	C2	C2									
5	141/171/300/302/101/ Sala V	ENTREGA	TAME															V1	V1	V1										
6	Oficinas LAN	ENTREGA	LAN																			V1	V1	V1						
7	1500	CARGA	LAN																				C2	C2	C2	C2				
8	546	CARGA	LAN								C2	C2	C2	C2																
9	1631	CARGA/DESCARGA	LAN															C2	C2	C2	C2	C2	C2	C2						
10	547	CARGA	LAN						C2	C2	C2	C2																		

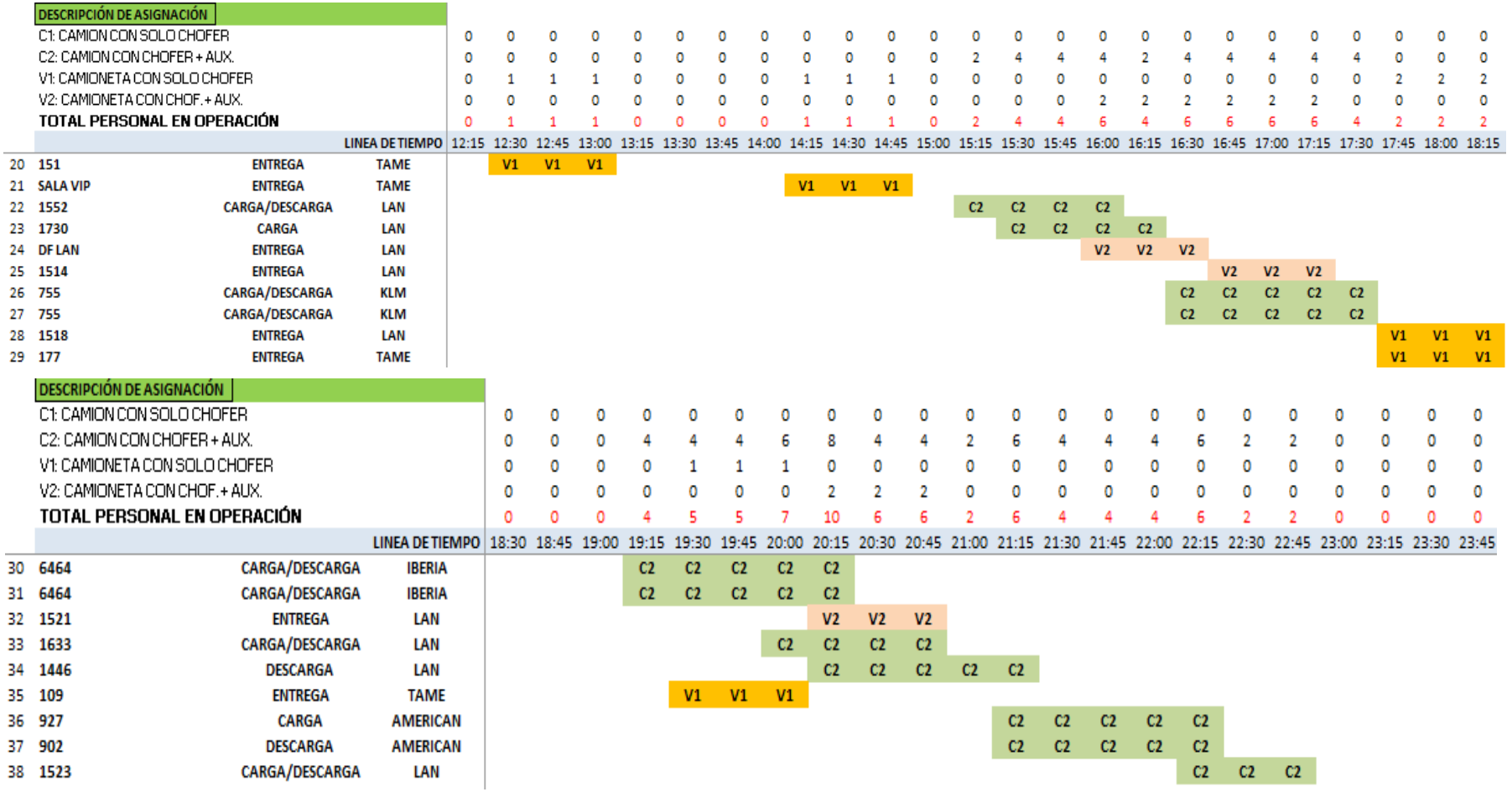
DESCRIPCIÓN DE ASIGNACIÓN	7:00	7:15	7:30	7:45	8:00	8:15	8:30	8:45	9:00	9:15	9:30	9:45	10:00	10:15	10:30	10:45	11:00	11:15	11:30	11:45	12:00
C1: CAMION CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.	0	0	2	2	2	2	2	0	0	0	0	2	2	4	2	2	0	4	4	4	4
V1: CAMIONETA CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	1	0	0	0	0
V2: CAMIONETA CON CHOF.+ AUX.	2	2	2	0	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN	2	2	4	2	4	4	4	2	0	1	1	3	2	4	3	3	1	4	4	4	4

	VUELO	TIPO DE OPERACIÓN	AEROLÍNEA	7:00	7:15	7:30	7:45	8:00	8:15	8:30	8:45	9:00	9:15	9:30	9:45	10:00	10:15	10:30	10:45	11:00	11:15	11:30	11:45	12:00
11	1506	CARGA/DESCARGA	LAN	V2	V2	V2																		
12	1447	CARGA/DESCARGA	LAN			C2	C2	C2	C2	C2														
13	DF LAN	ENTREGA	LAN					V2	V2	V2	V2													
14	DF 1508	CARGA	LAN												C2	C2	C2							
15	186	ENTREGA	TAME										V1	V1	V1									
16	1553	CARGA/DESCARGA	LAN													C2	C2	C2						
17	SALA VIP	ENTREGA	TAME														V1	V1	V1					
18	1631	DESCARGA	LAN																	C2	C2	C2	C2	
19	538	CARGA	LAN																	C2	C2	C2	C2	

Tabla 2.8 Mapeo Operación Diaria Miércoles

MIÉRCOLES

MAPEO DE OPERACIÓN DIARIA



DESCRIPCIÓN DE ASIGNACIÓN																											
C1: CAMION CON SOLO CHOFER			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.			0	0	0	0	0	0	0	0	0	0	0	0	2	4	4	4	2	4	4	4	4	4	0	0	0
V1: CAMIONETA CON SOLO CHOFER			0	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	2	2	2
V2: CAMIONETA CON CHOF.+ AUX.			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN			0	1	1	1	0	0	0	0	1	1	1	0	2	4	4	6	4	6	6	6	6	4	2	2	2
		LINEA DE TIEMPO	12:15	12:30	12:45	13:00	13:15	13:30	13:45	14:00	14:15	14:30	14:45	15:00	15:15	15:30	15:45	16:00	16:15	16:30	16:45	17:00	17:15	17:30	17:45	18:00	18:15
20	151	ENTREGA	TAME	V1	V1	V1																					
21	SALA VIP	ENTREGA	TAME								V1	V1	V1														
22	1552	CARGA/DESCARGA	LAN												C2	C2	C2	C2									
23	1730	CARGA	LAN													C2	C2	C2	C2								
24	DF LAN	ENTREGA	LAN															V2	V2	V2							
25	1514	ENTREGA	LAN																		V2	V2	V2				
26	755	CARGA/DESCARGA	KLM																	C2	C2	C2	C2	C2			
27	755	CARGA/DESCARGA	KLM																	C2	C2	C2	C2	C2			
28	1518	ENTREGA	LAN																						V1	V1	V1
29	177	ENTREGA	TAME																						V1	V1	V1

DESCRIPCIÓN DE ASIGNACIÓN																											
C1: CAMION CON SOLO CHOFER			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.			0	0	0	4	4	4	6	8	4	4	2	6	4	4	4	6	2	2	0	0	0	0	0	0	0
V1: CAMIONETA CON SOLO CHOFER			0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V2: CAMIONETA CON CHOF.+ AUX.			0	0	0	0	0	0	0	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN			0	0	0	4	5	5	7	10	6	6	2	6	4	4	4	6	2	2	0	0	0	0	0	0	0
		LINEA DE TIEMPO	18:30	18:45	19:00	19:15	19:30	19:45	20:00	20:15	20:30	20:45	21:00	21:15	21:30	21:45	22:00	22:15	22:30	22:45	23:00	23:15	23:30	23:45			
30	6464	CARGA/DESCARGA	IBERIA				C2	C2	C2	C2	C2																	
31	6464	CARGA/DESCARGA	IBERIA				C2	C2	C2	C2	C2																	
32	1521	ENTREGA	LAN								V2	V2	V2															
33	1633	CARGA/DESCARGA	LAN							C2	C2	C2	C2															
34	1446	DESCARGA	LAN								C2	C2	C2	C2	C2													
35	109	ENTREGA	TAME					V1	V1	V1																		
36	927	CARGA	AMERICAN											C2	C2	C2	C2	C2										
37	902	DESCARGA	AMERICAN											C2	C2	C2	C2	C2										
38	1523	CARGA/DESCARGA	LAN															C2	C2	C2								

Tabla 2.9 Mapeo Operación Diaria Miércoles

JUEVES

MAPEO DE OPERACIÓN DIARIA

C1 = SOLO CHOFER | C2 = CHOFER + AUX | V1 = SOLO CHOFER | V2 = CHOFER + AUX

DESCRIPCIÓN DE ASIGNACIÓN	0:15	0:30	0:45	1:00	1:15	1:30	1:45	2:00	2:15	2:30	2:45	3:00	3:15	3:30	3:45	4:00	4:15	4:30	4:45	5:00	5:15	5:30	5:45	6:00	6:15	6:30	6:45
C1: CAMION CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.	0	0	0	2	2	2	2	2	2	2	2	0	0	0	0	4	4	4	4	4	4	4	4	2	0	0	0
V1: CAMIONETA CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0
V2: CAMIONETA CON CHOF. + AUX.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN	0	0	0	2	2	2	2	2	0	0	0	0	0	0	0	5	5	5	5	5	5	4	4	2	0	0	0

	VUELO	TIPO DE OPERACIÓN	AEROLÍNEA	LINEA DE TIEMPO	7:00	7:15	7:30	7:45	8:00	8:15	8:30	8:45	9:00	9:15	9:30	9:45	10:00	10:15	10:30	10:45	11:00	11:15	11:30	11:45	12:00	12:15	12:30	12:45	13:00
	933 (ABORDADOR 2								C2	C2	C2	C2																	
1	HORAS EXTRAS O SUPERVISOR)	DESCARGA	AMERICAN																										
2	539	DESCARGA	LAN														C2	C2	C2	C2									
3	1631	CARGA	LAN														C2	C2	C2	C2									
4	141/171/300/302/101/Sala v	CARGA	TAME														V1	V1	V1										
5	Oficinas LAN	ENTREGA	LAN																		V1	V1	V1						
6	948	CARGA	AMERICAN																			C2	C2	C2	C2				
7	1500	CARGA	LAN																			C2	C2	C2					
8	1504	CARGA	LAN																						C2	C2			
	DESCRIPCIÓN DE ASIGNACIÓN																												
	C1: CAMION CON SOLO CHOFER				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	C2: CAMION CON CHOFER + AUX.				0	0	2	2	2	2	2	2	2	2	2	0	0	2	2	2	0	0	2	2	4	2	2	2	0
	V1: CAMIONETA CON SOLO CHOFER				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	1	1	1	0	1	1	0
	V2: CAMIONETA CON CHOF. + AUX.				0	0	0	0	0	2	2	2	2	0	0	2	2	2	4	4	2	2	2	3	5	2	3	3	0
	TOTAL PERSONAL EN OPERACIÓN				0	0	2	2	4	4	4	4	2	2	4	2	0	2	4	4	2	2	3	3	5	2	3	3	0
9	1447	CARGA/DESCARGA	LAN				C2	C2	C2	C2	C2	C2																	
10	DF LAN	ENTREGA	LAN						V2	V2	V2	V2																	
11	DF 1508	CARGA	LAN										C2	C2	C2	C2													
12	186	ENTREGA	TAME												V2	V2													
13	1553	CARGA/DESCARGA	LAN																C2	C2	C2								
14	SALA VIP	ENTREGA	TAME																V2	V2	V2	V2							
15	1510	CARGA/DESCARGA	LAN																				V1	V1	V1				
16	151	ENTREGA	TAME																										V1
17	1630	CARGA	LAN																				C2	C2	C2				
18	538	DESCARGA	LAN																							C2	C2	C2	C2

Tabla 2.10 Mapeo Operación Diaria Jueves

JUEVES

MAPEO DE OPERACIÓN DIARIA

DESCRIPCIÓN DE ASIGNACIÓN	13:15	13:30	13:45	14:00	14:15	14:30	14:45	15:00	15:15	15:30	15:45	16:00	16:15	16:30	16:45	17:00	17:15	17:30	17:45	18:00	18:15	18:30	18:45	19:00
C1: CAMION CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.	0	0	0	0	0	0	2	2	2	2	2	2	2	4	6	6	4	4	0	0	0	0	0	0
V1: CAMIONETA CON SOLO CHOFER	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	1	1
V2: CAMIONETA CON CHOF. + AUX.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN	0	0	0	1	1	1	2	2	2	2	2	2	2	4	6	6	4	5	1	1	1	2	1	1
LINEA DE TIEMPO	13:15	13:30	13:45	14:00	14:15	14:30	14:45	15:00	15:15	15:30	15:45	16:00	16:15	16:30	16:45	17:00	17:15	17:30	17:45	18:00	18:15	18:30	18:45	19:00
19 SALA VIP ENTREGA TAME				V1	V1	V1																		
20 566 CARGA/DESCARGA TAME							C2	C2	C2	C2														
21 1552 CARGA/DESCARGA LAN											C2	C2	C2											
22 1560 CARGA/DESCARGA LAN														C2	C2	C2								
23 755 CARGA/DESCARGA KLM														C2	C2	C2	C2	C2						
24 755 CARGA/DESCARGA KLM															C2	C2	C2	C2						
25 177 ENTREGA TAME																		V1	V1					

DESCRIPCIÓN DE ASIGNACIÓN	19:15	19:30	19:45	20:00	20:15	20:30	20:45	21:00	21:15	21:30	21:45	22:00	22:15	22:30	22:45	23:00	23:15	23:30	23:45
C1: CAMION CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.	4	4	4	4	6	2	2	6	6	4	0	4	4	4	2	0	0	0	0
V1: CAMIONETA CON SOLO CHOFER	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V2: CAMIONETA CON CHOF. + AUX.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN	4	4	5	5	6	2	2	6	6	4	0	4	4	4	2	0	0	0	0
LINEA DE TIEMPO	19:15	19:30	19:45	20:00	20:15	20:30	20:45	21:00	21:15	21:30	21:45	22:00	22:15	22:30	22:45	23:00	23:15	23:30	23:45
26 1516 ENTREGA LAN																			
27 1518 ENTREGA LAN																			
28 6464 CARGA/DESCARGA IBERIA	C2	C2	C2	C2	C2														
29 6464 CARGA/DESCARGA IBERIA	C2	C2	C2	C2	C2														
30 109 ENTREGA TAME			V1	V1															
31 1446 DESCARGA LAN					C2	C2	C2												
32 927 DESCARGA AMERICAN								C2	C2										
33 902 CARGA AMERICAN								C2	C2	C2									
34 1521 CARGA/DESCARGA LAN								C2	C2	C2									
35 1523 CARGA/DESCARGA LAN												C2	C2	C2					
36 1733 DESCARGA LAN												C2	C2	C2	C2				

Tabla 2.11 Mapeo Operación Diaria Jueves

MAPEO DE OPERACIÓN DIARIA

VIERNES

C1 = SOLO CHOFER | C2 = CHOFER + AUX | V1 = SOLO CHOFER | V2 = CHOFER + AUX

DESCRIPCIÓN DE ASIGNACIÓN	0:15	0:30	0:45	1:00	1:15	1:30	1:45	2:00	2:15	2:30	2:45	3:00	3:15	3:30	3:45	4:00	4:15	4:30	4:45	5:00	5:15	5:30	5:45	6:00	6:15	6:30	6:45
C1: CAMION CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.	0	0	2	2	2	2	0	0	0	2	2	2	0	0	0	2	2	2	2	2	2	2	0	0	0	2	2
V1: CAMIONETA CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0
V2: CAMIONETA CON CHOF.+ AUX.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN	0	0	2	2	2	2	0	0	0	2	2	2	0	0	0	3	3	3	3	3	3	2	0	0	0	2	2

	VUELO	TIPO DE OPERACIÓN	AEROLÍNEA	0:15	0:30	0:45	1:00	1:15	1:30	1:45	2:00	2:15	2:30	2:45	3:00	3:15	3:30	3:45	4:00	4:15	4:30	4:45	5:00	5:15	5:30	5:45	6:00	6:15	6:30	6:45
1	933	DESCARGA	AMERICAN			C2	C2	C2	C2																					
2	565	DESCARGA	TAME										C2	C2	C2															
3	539	DESCARGA	LAN																C2	C2	C2	C2								
4	141/171/300/302/101/ Sala V	CARGA	TAME																V1	V1	V1									
5	Oficinas LAN	ENTREGA	LAN																			V1	V1	V1						
6	948	CARGA	AMERICAN																				C2	C2	C2					
7	1504	CARGA	LAN																										C2	C2
8	1447	CARGA/DESCARGA	LAN																											

DESCRIPCIÓN DE ASIGNACIÓN	7:00	7:15	7:30	7:45	8:00	8:15	8:30	8:45	9:00	9:15	9:30	9:45	10:00	10:15	10:30	10:45	11:00	11:15	11:30	11:45	12:00	12:15	12:30	12:45	13:00
C1: CAMION CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.	2	0	0	2	2	2	2	4	4	2	0	0	0	0	2	2	2	0	2	2	4	4	4	0	0
V1: CAMIONETA CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	0	0	0	0	0	1	1	0
V2: CAMIONETA CON CHOF.+ AUX.	0	0	0	0	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN	2	0	0	2	4	4	4	6	6	3	1	1	0	1	3	3	2	0	2	2	4	4	5	1	0

	VUELO	TIPO DE OPERACIÓN	AEROLÍNEA	7:00	7:15	7:30	7:45	8:00	8:15	8:30	8:45	9:00	9:15	9:30	9:45	10:00	10:15	10:30	10:45	11:00	11:15	11:30	11:45	12:00	12:15	12:30	12:45	13:00
8	1447	CARGA/DESCARGA	LAN				C2	C2	C2	C2	C2	C2																
9	DF LAN	CARGA/DESCARGA	LAN					V2	V2	V2	V2	V2																
10	1551	CARGA/DESCARGA	LAN								C2	C2	C2															
11	186	ENTREGA	TAME										V1	V1	V1													
12	SALA VIP	ENTREGA	TAME													V1	V1	V1										
13	1561	CARGA/DESCARGA	LAN														C2	C2	C2									
14	1631	DESCARGA	LAN																	C2	C2	C2	C2	C2				
15	538	CARGA	LAN																			C2	C2	C2				
16	151	ENTREGA	TAME																						V1	V1		

Tabla 2.12 Mapeo Operación Diaria Viernes

VIERNES

MAPEO DE OPERACIÓN DIARIA

DESCRIPCIÓN DE ASIGNACIÓN	13:15	13:30	13:45	14:00	14:15	14:30	14:45	15:00	15:15	15:30	15:45	16:00	16:15	16:30	16:45	17:00	17:15	17:30	17:45	18:00	18:15	18:30	18:45	19:00
C1: CAMION CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.	0	0	0	0	0	2	2	2	0	0	2	4	4	4	4	2	2	2	0	0	2	2	2	2
V1: CAMIONETA CON SOLO CHOFER	0	0	0	0	1	1	1	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
V2: CAMIONETA CON CHOF. + AUX.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN	0	0	0	0	1	3	3	2	0	0	2	4	4	5	5	3	3	3	0	0	2	2	2	2

				LINEA DE TIEMPO	13:15	13:30	13:45	14:00	14:15	14:30	14:45	15:00	15:15	15:30	15:45	16:00	16:15	16:30	16:45	17:00	17:15	17:30	17:45	18:00	18:15	18:30	18:45	19:00
17	SALA VIP	ENTREGA	TAME						V1	V1	V1																	
18	1550	CARGA/DESCARGA	LAN							C2	C2	C2																
19	1560	CARGA/DESCARGA	LAN												C2	C2	C2											
20	1730	CARGA/DESCARGA	LAN													C2	C2	C2	C2									
21	755	CARGA/DESCARGA	KLM															C2	C2	C2	C2	C2						
22	755	CARGA/DESCARGA	KLM																									
23	1514	ENTREGA	LAN															V1	V1	V1								
24	177	ENTREGA	TAME																		V1	V1						
25	1633	CARGA/DESCARGA	LAN																						C2	C2	C2	C2

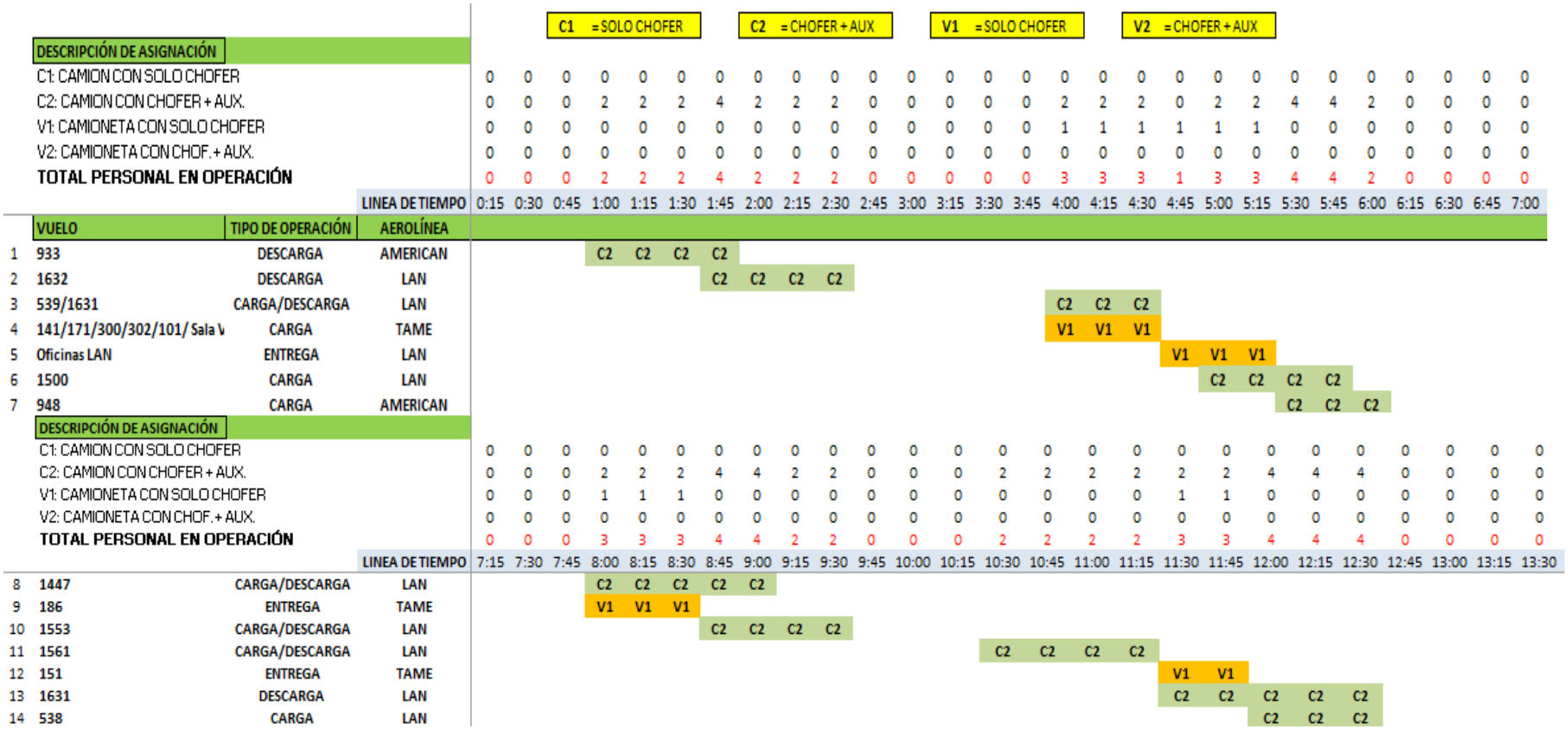
DESCRIPCIÓN DE ASIGNACIÓN	19:15	19:30	19:45	20:00	20:15	20:30	20:45	21:00	21:15	21:30	21:45	22:00	22:15	22:30	22:45	23:00	23:15	23:30	23:45
C1: CAMION CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.	0	2	2	2	2	2	4	2	6	4	4	4	0	0	0	0	0	0	0
V1: CAMIONETA CON SOLO CHOFER	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V2: CAMIONETA CON CHOF. + AUX.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN	0	3	3	2	2	2	4	2	6	4	4	4	0	0	0	0	0	0	0

				LINEA DE TIEMPO	19:15	19:30	19:45	20:00	20:15	20:30	20:45	21:00	21:15	21:30	21:45	22:00	22:15	22:30	22:45	23:00	23:15	23:30	23:45
26	109	ENTREGA	TAME			V1	V1																
27	1520	CARGA/DESCARGA	LAN			C2	C2	C2															
28	1446	DESCARGA	LAN						C2	C2	C2												
29	1521	CARGA/DESCARGA	LAN								C2	C2	C2										
30	927	DESCARGA	AMERICAN										C2	C2	C2	C2							
31	902	CARGA	AMERICAN										C2	C2	C2	C2							

Tabla 2.13 Mapeo Operación Diaria Viernes

MAPEO DE OPERACIÓN DIARIA

SÁBADO



DESCRIPCIÓN DE ASIGNACIÓN	0:15	0:30	0:45	1:00	1:15	1:30	1:45	2:00	2:15	2:30	2:45	3:00	3:15	3:30	3:45	4:00	4:15	4:30	4:45	5:00	5:15	5:30	5:45	6:00	6:15	6:30	6:45	7:00
C1: CAMION CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.	0	0	0	2	2	2	4	2	2	2	0	0	0	0	0	2	2	2	0	2	2	4	4	2	0	0	0	0
V1: CAMIONETA CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0
V2: CAMIONETA CON CHOF.+ AUX.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN	0	0	0	2	2	2	4	2	2	2	0	0	0	0	0	3	3	3	1	3	3	4	4	2	0	0	0	0

	VUELO	TIPO DE OPERACIÓN	AEROLÍNEA	LINEA DE TIEMPO	0:15	0:30	0:45	1:00	1:15	1:30	1:45	2:00	2:15	2:30	2:45	3:00	3:15	3:30	3:45	4:00	4:15	4:30	4:45	5:00	5:15	5:30	5:45	6:00	6:15	6:30	6:45	7:00
1	933	DESCARGA	AMERICAN					C2	C2	C2	C2																				
2	1632	DESCARGA	LAN								C2	C2	C2	C2																	
3	539/1631	CARGA/DESCARGA	LAN																	C2	C2	C2									
4	141/171/300/302/101/ Sala V	CARGA	TAME																	V1	V1	V1									
5	Oficinas LAN	ENTREGA	LAN																				V1	V1	V1						
6	1500	CARGA	LAN																					C2	C2	C2	C2				
7	948	CARGA	AMERICAN																							C2	C2	C2			

DESCRIPCIÓN DE ASIGNACIÓN	7:15	7:30	7:45	8:00	8:15	8:30	8:45	9:00	9:15	9:30	9:45	10:00	10:15	10:30	10:45	11:00	11:15	11:30	11:45	12:00	12:15	12:30	12:45	13:00	13:15	13:30
C1: CAMION CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.	0	0	0	2	2	2	4	4	2	2	0	0	0	2	2	2	2	2	2	4	4	4	0	0	0	0
V1: CAMIONETA CON SOLO CHOFER	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
V2: CAMIONETA CON CHOF.+ AUX.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN	0	0	0	3	3	3	4	4	2	2	0	0	0	2	2	2	2	3	3	4	4	4	0	0	0	0

	VUELO	TIPO DE OPERACIÓN	AEROLÍNEA	LINEA DE TIEMPO	7:15	7:30	7:45	8:00	8:15	8:30	8:45	9:00	9:15	9:30	9:45	10:00	10:15	10:30	10:45	11:00	11:15	11:30	11:45	12:00	12:15	12:30	12:45	13:00	13:15	13:30
8	1447	CARGA/DESCARGA	LAN					C2	C2	C2	C2	C2																		
9	186	ENTREGA	TAME					V1	V1	V1																					
10	1553	CARGA/DESCARGA	LAN								C2	C2	C2	C2																	
11	1561	CARGA/DESCARGA	LAN														C2	C2	C2	C2											
12	151	ENTREGA	TAME																		V1	V1									
13	1631	DESCARGA	LAN																		C2	C2	C2	C2	C2						
14	538	CARGA	LAN																				C2	C2	C2						

Tabla 2.14 Mapeo Operación Diaria Sábado

MAPEO DE OPERACIÓN DIARIA

SÁBADO

DESCRIPCIÓN DE ASIGNACIÓN	13:45	14:00	14:15	14:30	14:45	15:00	15:15	15:30	15:45	16:00	16:15	16:30	16:45	17:00	17:15	17:30	17:45	18:00	18:15	18:30	18:45	19:00
C1: CAMION CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.	0	0	0	0	0	0	0	0	2	2	2	2	6	6	4	4	0	0	0	0	0	0
V1: CAMIONETA CON SOLO CHOFER	0	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1	1
V2: CAMIONETA CON CHOF. + AUX.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN	0	1	1	1	0	0	0	0	2	2	2	2	7	7	5	5	1	0	0	1	1	1
LINEA DE TIEMPO	13:45	14:00	14:15	14:30	14:45	15:00	15:15	15:30	15:45	16:00	16:15	16:30	16:45	17:00	17:15	17:30	17:45	18:00	18:15	18:30	18:45	19:00
15 SALA VIP — ENTREGA — TAME		V1	V1	V1																		
16 1552 — CARGA/DESCARGA — LAN									C2	C2	C2											
17 1560 — CARGA/DESCARGA — LAN												C2	C2	C2								
18 755 — CARGA/DESCARGA — KLM													C2	C2	C2	C2						
19 755 — CARGA/DESCARGA — KLM													C2	C2	C2	C2						
20 177 — ENTREGA — TAME																V1	V1					
21 1514 — ENTREGA — LAN													V1	V1	V1							
22 1518 — ENTREGA — LAN																				V1	V1	V1

DESCRIPCIÓN DE ASIGNACIÓN	19:15	19:30	19:45	20:00	20:15	20:30	20:45	21:00	21:15	21:30	21:45	22:00	22:15	22:30	22:45	23:00	23:15	23:30	23:45
C1: CAMION CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.	0	0	0	0	2	6	6	6	4	4	4	4	2	2	0	0	0	0	0
V1: CAMIONETA CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V2: CAMIONETA CON CHOF. + AUX.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN	0	0	0	0	2	6	6	6	4	4	4	4	2	2	0	0	0	0	0
LINEA DE TIEMPO	19:15	19:30	19:45	20:00	20:15	20:30	20:45	21:00	21:15	21:30	21:45	22:00	22:15	22:30	22:45	23:00	23:15	23:30	23:45
23 1446 — DESCARGA — LAN					C2	C2	C2	C2											
24 927 — DESCARGA — AMERICAN						C2	C2	C2	C2										
25 902 — CARGA — AMERICAN						C2	C2	C2	C2										
26 1523 — CARGA/DESCARGA — LAN										C2	C2	C2							
27 1733 — DESCARGA — LAN										C2	C2	C2	C2	C2					

Tabla 2.15 Mapeo Operación Diaria Sábado

DOMINGO

MAPEO DE OPERACIÓN DIARIA

C1 = SOLO CHOFER | C2 = CHOFER + AUX | V1 = SOLO CHOFER | V2 = CHOFER + AUX

DESCRIPCIÓN DE ASIGNACIÓN	0:15	0:30	0:45	1:00	1:15	1:30	1:45	2:00	2:15	2:30	2:45	3:00	3:15	3:30	3:45	4:00	4:15	4:30	4:45	5:00	5:15	5:30	5:45	6:00	6:15	6:30	6:45	7:00
C1: CAMION CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0	2	2	2	0	0	0	2	2	2	2	2	2	2
V1: CAMIONETA CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0
V2: CAMIONETA CON CHOF.+ AUX.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0	3	3	3	1	1	1	2	2	2	2	2	2	2

	VUELO	TIPO DE OPERACIÓN	AEROLÍNEA	LINEA DE TIEMPO
1	933	DESCARGA	AMERICAN	C2 C2 C2 C2
2	539	DESCARGA	LAN	C2 C2 C2
3	141/171/300/302/101/ Sala V	CARGA	TAME	V1 V1 V1
4	Oficinas LAN	CARGA	LAN	V1 V1 V1
5	948	CARGA	AMERICAN	C2 C2 C2
6	1506	CARGA	LAN	C2 C2 C2 C2

DESCRIPCIÓN DE ASIGNACIÓN	7:15	7:30	7:45	8:00	8:15	8:30	8:45	9:00	9:15	9:30	9:45	10:00	10:15	10:30	10:45	11:00	11:15	11:30	11:45	12:00	12:15	12:30	12:45	13:00	13:15	13:30
C1: CAMION CON SOLO CHOFER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.	0	0	0	2	2	2	4	4	2	2	0	0	0	2	2	2	2	2	2	4	4	4	0	0	0	0
V1: CAMIONETA CON SOLO CHOFER	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
V2: CAMIONETA CON CHOF.+ AUX.	0	0	0	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN	0	0	0	5	5	5	4	4	2	2	0	0	0	2	2	2	2	2	3	5	4	4	0	0	0	0

	VUELO	TIPO DE OPERACIÓN	AEROLÍNEA	LINEA DE TIEMPO
7	1447	CARGA/DESCARGA	LAN	C2 C2 C2 C2 C2
8	DF LAN	ENTREGA	LAN	V2 V2 V2
9	186	ENTREGA	TAME	V1 V1 V1
10	1553	CARGA/DESCARGA	LAN	C2 C2 C2 C2
11	1561	CARGA/DESCARGA	LAN	C2 C2 C2 C2
12	151	ENTREGA	TAME	V1 V1
13	1631	DESCARGA	LAN	C2 C2 C2 C2 C2
14	538	CARGA	LAN	C2 C2 C2

Tabla 2.16 Mapeo Operación Diaria Domingo

DOMINGO

MAPEO DE OPERACIÓN DIARIA

DESCRIPCIÓN DE ASIGNACIÓN				13:45	14:00	14:15	14:30	14:45	15:00	15:15	15:30	15:45	16:00	16:15	16:30	16:45	17:00	17:15	17:30	17:45	18:00	18:15	18:30	18:45	19:00	19:15	19:30	19:45	20:00
C1: CAMION CON SOLO CHOFER				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.				0	0	0	0	2	2	2	2	2	2	2	4	6	6	6	4	0	0	0	0	0	0	4	4	4	4
V1: CAMIONETA CON SOLO CHOFER				0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	1	1	0	0	1	1
V2: CAMIONETA CON CHOF. + AUX.				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN				0	1	1	1	3	2	2	2	2	2	2	4	6	6	6	5	1	1	1	2	1	1	4	4	5	5
			LINEA DE TIEMPO	13:45	14:00	14:15	14:30	14:45	15:00	15:15	15:30	15:45	16:00	16:15	16:30	16:45	17:00	17:15	17:30	17:45	18:00	18:15	18:30	18:45	19:00	19:15	19:30	19:45	20:00
15	SALA VIP	ENTREGA	TAME		V1	V1	V1	V1																					
16	566	CARGA	TAME					C2	C2	C2	C2																		
17	1552	CARGA/DESCARGA	LAN									C2	C2	C2	C2														
18	1560	CARGA/DESCARGA	TAME												C2	C2	C2	C2											
19	755	CARGA/DESCARGA	KLM													C2	C2	C2	C2										
20	755	CARGA/DESCARGA	KLM													C2	C2	C2	C2										
21	177	ENTREGA	TAME																V1	V1									
22	1516	ENTREGA	LAN																		V1	V1	V1						
23	1518	ENTREGA	LAN																				V1	V1	V1				
24	6464	CARGA/DESCARGA	IBERIA																							C2	C2	C2	C2
25	6464	CARGA/DESCARGA	IBERIA																							C2	C2	C2	C2

DESCRIPCIÓN DE ASIGNACIÓN				20:15	20:30	20:45	21:00	21:15	21:30	21:45	22:00	22:15	22:30	22:45	23:00	23:15	23:30	23:45
C1: CAMION CON SOLO CHOFER				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2: CAMION CON CHOFER + AUX.				2	6	6	8	6	4	2	2	0	0	0	0	0	0	0
V1: CAMIONETA CON SOLO CHOFER				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V2: CAMIONETA CON CHOF. + AUX.				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL PERSONAL EN OPERACIÓN				2	6	6	8	6	4	2	2	0	0	0	0	0	0	0
			LINEA DE TIEMPO	20:15	20:30	20:45	21:00	21:15	21:30	21:45	22:00	22:15	22:30	22:45	23:00	23:15	23:30	23:45
26	109	ENTREGA	TAME															
27	1446	DESCARGA	LAN	C2	C2	C2	C2											
28	927	DESCARGA	AMERICAN		C2	C2	C2	C2										
29	902	CARGA	AMERICAN		C2	C2	C2	C2										
30	1521	CARGA/DESCARGA	LAN				C2	C2	C2									
31	1523	CARGA/DESCARGA	LAN						C2	C2	C2							

Tabla 2.17 Mapeo Operación Diaria Domingo

CAPÍTULO 3
DESARROLLO DEL ALGORITMO

3.1. Objetivo

Se buscará desarrollar un algoritmo SS para optimizar la asignación de recursos y sean usados de la mejor manera, es decir que los vehículos sean asignados a los vuelos sin que se tenga que enviar el mismo camión a atender dos vuelos en el mismo momento y que los empleados sean asignados al momento de estar en su hora de trabajo.

3.2. Notación Matemática

El método del algoritmo SS es un proceso estocástico no lineal, es decir se genera un espacio solución y el algoritmo selecciona de manera aleatoria, haciendo distintos elementos y los combina para encontrar las soluciones más optimas, esto se logra mediante la selección de los genes (vehículos, empleados) y hace distintas combinaciones que pertenecen al espacio solución, algunas de las fórmulas utilizadas para hacer estas combinaciones son las de la teoría combinatoria básica, es decir el espacio que contiene todas las soluciones se encuentra de la siguiente manera:

$$\boldsymbol{soluciones = vehiculos^{N°de\ vuelos} x\ choferes^{N°vehiculos} x\ guias^{N°de\ vehiculos}}$$

Claramente la ecuación anterior representa todas las soluciones posibles incluyendo las óptimas, no óptimas y las no viables. El algoritmo, valiéndose de métodos computacionales como: condicionales, ciclos, variables booleanas, y las fórmulas de teoría combinatoria como: la combinación, la variación y la permutación (todas con repetición en algunos casos y en otros no), además de las restricciones impuestas, logrará seleccionar las soluciones más acordes.

No utiliza función de optimización lineal ya que como se dijo anteriormente es un ***proceso estocástico***, lo que significa que el algoritmo va calculando soluciones y las evalúa según las restricciones. A continuación las distintas fórmulas estadísticas:

Combinación y combinación con repetición:

$$C_k^n = \frac{n!}{k!\,(n-k)!} \qquad rC_k^n = \frac{(n+k-1)!}{k!\,(n-1)!}$$

Permutación y Permutación con repetición:

$$nPr = \frac{n!}{(n-r)!} \qquad P_n = n!$$

Variación con Repetición:

$$VR_n^p = n^p$$

De esta manera se seleccionaran las distintas combinaciones logrando así la solución de asignación más óptima y de menor costo.

Se ingresa el número de vuelos en formato Excel, así como su respectiva hora de llegada y salida y mediante el algoritmo de búsqueda dispersa, el modelo arrojará una planificación asignando los trabajadores y vehículos a cada vuelo.

Vale acotar que ciertas consideraciones fueron hechas previamente, éstas fueron:

- Se presume de antemano que todos los vuelos requerirán de dos vehículos (carga/descarga). Ya que es imposible determinar la operación que se necesitará antes de que el vuelo sea atendido, por lo que la mejor opción es considerar situaciones en las que el uso de los recursos sea el más extenso.
- Se considerará que los vehículos podrán estar encendidos de manera indefinida.
- Los casos que sean entregas a salas VIP se tomará en cuenta con el uso de una camioneta con un conductor.

Los datos serán ingresados bajo el siguiente esquema que fue proporcionado como muestra:

LUNES		
REQUERIMIENTO		
VUELOS	SALIDA PLANTA	LLEGADA PLANTA
539LAN	4:05	5:40
VIP,SAB	4:15	5:55
948 AA	4:55	6:00
NAC.LAN	5:00	6:36
1447LAN	7:50	8:40
1551LAN	8:50	9:20
1553LAN	10:00	10:45
538LAN	11:35	12:45
1633LAN	13:30	14:53
1550LAN	14:04	14:51
1730LAN	15:06	15:57
550TAM	15:25	16:27
1552LAN	15:37	16:12
755KLM	17:00	18:00
927AA	19:56	21:00
1632LAN	20:25	21:00
551TAM	21:36	22:06
NAC.LAN	21:50	22:05
1446LAN	22:40	23:40
1441LAN	22:40	23:40
902AA	22:30	23:25

Tabla 3.1 Ejemplo formato ingreso de datos horarios vuelos

Y posteriormente el programa arrojará un archivo Excel con los vehículos y trabajadores que irán a atenderlos.

3.2.1. Definición De Variables

Este algoritmo SS para la planificación se centra en el uso de algoritmo de búsqueda dispersa para optimizar el uso de los recursos que son limitados, estas serán nuestras variables:

- "Tipo de operación" Es una variable booleana/binaria (carga/descarga).
- "Tipo de vuelo" Variable booleana/binaria (servicio regular / sala VIP)
- "Disponibilidad de trabajadores" Variable booleana/binaria (Disponibilidad/ No Disponibilidad)
- "Cantidad de Trabajadores" Constante.
- "Cantidad de Vehículos" Constante.
- "Cantidad de vuelos a evaluar" Constante.
- "Cantidad de soluciones posibles". Constante
- "Soluciones Óptimas". Constante.

3.2.2. Datos De Entrada

- Se tiene un conjunto de 8 vehículos: 6 camiones y dos camionetas.
- Se tiene un conjunto de 16 trabajadores: 6 guías y 10 choferes.
- Se ingresa el número de vuelos.

3.2.3. Restricciones

$$Vehículo_i \text{^} Vehículo_j \xrightarrow[se\ asigna]{} Vuelo_k \qquad con\ i \neq j \neq k$$

$$Guia_i \text{^}\ Chofer_j \rightarrow Vehículo$$

La primera expresión se refiere a las combinaciones de distinta índole a seguir para que los vehículos ejecuten la operación de carga/descarga de cada vuelo.

La segunda expresión se centra en la combinación de guía y chofer para ser asignada a un vehículo, en el caso de la camioneta y la sala VIP el problema se reduce a cual chofer está disponible y cuál camioneta está disponible, es un problema trivial ya que hay muy pocas operaciones en las que se necesitará de una camioneta.

Los subíndices deben ser distintos en todo momento ya que de no ser así, se asignaría por ejemplo un mismo camión a dos operaciones (carga/ descarga de un vuelo k) o peor aún a dos vuelos al mismo tiempo.

Vuelo	Camión	Chofer	Guía
1	1	1	1
2	2	2	2
3	3	3	1
4	4	1	2
5	3	2	2

Tabla 3.2 Ejemplo de posible combinación con 5 vuelos

En la tabla anterior se expresa una posible solución del problema con 5 vuelos enumerados del 1 al 5 y se dispone de 4 camiones enumerados del 1 al 4, 3 choferes y 2 guías enumerados del uno al tres y del uno al 2 respectivamente, como se muestra es una combinación en la asignación de los recursos garantizando que ninguno coincida y con una asignación prudente para que haya oportunidad por parte de cada recurso de cumplir su función a tiempo para cumplir la siguiente tarea en caso de ser requerido.

En el ejemplo anterior se considera que los vuelos llegan a horas distintas por temas de simplicidad.

3.3. Resumen de tiempos en los que incurre la atención de un vuelo

A continuación se detalla la compilación de operación por día se ha tomado como referencia el máximo de vuelos a atender por día acorde a itinerarios nacionales e internacionales.

TIEMPO PROMEDIO TRANSCURRIDO TOTAL	0:53
TIEMPO ESTIMADO QUE TARDA EL VEHÍCULO EN LLEGAR DE LA PLANTA AL AVIÓN (MIN)	0:10
TIEMPO ESTIMADO QUE TARDA EL VEHÍCULO EN LLEGAR DEL AVION A PLANTA (MIN)	0:08
TIEMPO ESTIMADO DEL VEHÍCULO FUERA DE PLANTA (HH/MM)	1:11
Número DE TRABAJADORES POR CAMIÓN A CONSIDERAR	2

Tabla 3.3 Resumen de tiempos a considerar

3.4. Tiempo promedio transcurrido entre la salida y llegada de los camiones a la planta.

LUNES

VUELOS	SALIDA GG	LLEGADA GG	TIEMPO TRANSCURRIDO HH/MM
539LAN	4:05	5:40	1:35
VIP,SAB	4:15	5:55	1:40
948 AA	4:55	6:00	1:05
NAC.LAN	5:00	6:36	1:36
1447LAN	7:50	8:40	0:50
1551LAN	8:50	9:20	0:30
1553LAN	10:00	10:45	0:45
538LAN	11:35	12:45	1:10
1633LAN	13:30	14:53	1:23
1550LAN	14:04	14:51	0:47
1730LAN	15:06	15:57	0:51
550TAM	15:25	16:27	1:02
1552LAN	15:37	16:12	0:35
755KLM	17:00	18:00	1:00
927AA	19:56	21:00	1:04
1632LAN	20:25	21:00	0:35
551TAM	21:36	22:06	0:30
NAC.LAN	21:50	22:05	0:15
1446LAN	22:40	23:40	1:00
1441LAN	22:40	23:40	1:00
902AA	22:30	23:25	0:55
PROMEDIO	0:57		

Tabla 3.4 Resumen de tiempos Lunes

MARTES

VUELOS	SALIDA GG	LLEGADA GG	TIEMPO TRANSCURRIDO HH/MM
539LAN	4:40	5:25	0:45
VIP,SAB	4:30	6:40	2:10
948AA	4:55	6:08	1:13
NAC.LAN	5:05	6:05	1:00
1447LAN	7:55	8:45	0:50
1553LAN	9:55	10:50	0:55
1561LAN	11:25	12:05	0:40
538LAN	11:40	12:30	0:50
1633LAN	13:05	13:55	0:50
1442LAN	13:05	13:55	0:50
1512LAN	15:30	16:11	0:41
1552LAN	15:30	16:11	0:41
1560LAN	16:35	17:12	0:37
550TAM	16:05	17:00	0:55
755KLM	17:00	18:03	1:03
6464IB	18:48	19:01	0:13
927AA	19:55	20:46	0:51
NAC.LAN	20:36	21:20	0:44
1632LAN	20:36	21:20	0:44
NAC.LAN	22:00	22:21	0:21
1446LAN	22:21	23:46	1:25
1441LAN	22:21	23:46	1:25
902AA	22:18	23:00	0:42
1733LAN	22:30	0:30	2:00
551TAM	0:00	1:08	1:08
PROMEDIO	0:56		

Tabla 3.5 Resumen de tiempos Martes

MIÉRCOLES

VUELOS	SALIDA GG	LLEGADA GG	TIEMPO TRANSCURRIDO HH/MM
1631LAN	4:30	5:36	1:06
539LAN	4:30	5:36	1:06
VIP,SAB	4:38	5:50	1:12
948AA	5:00	5:36	0:36
NAC.LAN	5:00	6:20	1:20
1447LAN	7:40	8:40	1:00
1553LAN	10:06	10:43	0:37
1551LAN	11:30	12:03	0:33
1630LAN	11:20	12:18	0:58
538LAN	11:20	12:18	0:58
1512LAN	12:46	13:41	0:55
1442LAN	12:46	13:41	0:55
1552LAN	15:30	15:55	0:25
1730LAN	15:20	16:10	0:50
1550LAN	16:20	17:00	0:40
755KLM	16:45	18:00	1:15
6464IB	18:50	19:45	0:55
551TAM	19:36	20:05	0:29
927AA	20:35	21:25	0:50
1446LAN	22:40	23:12	0:32
550TAM	21:40	22:20	0:40
NAC.LAN	21:35	21:55	0:20
902AA	23:00	23:55	0:55
PROMEDIO	0:49		

Tabla 3.6 Resumen de tiempos Miércoles

JUEVES

VUELOS	SALIDA GG	LLEGADA GG	TIEMPO TRANSCURRIDO HH/MM
1631LAN	4:10	5:25	1:15
539LAN	4:10	5:25	1:15
VIP,SAB	4:05	6:00	1:55
948 AA	4:40	5:30	0:50
NAC.LAN	5:00	6:15	1:15
1447LAN	7:40	8:35	0:55
1551LAN	8:40	9:20	0:40
1553LAN	10:05	10:45	0:40
1630LAN	11:20	12:05	0:45
538LAN	11:20	12:05	0:45
1510LAN	12:50	13:18	0:28
1550LAN	14:00	14:45	0:45
1552LAN	15:20	15:50	0:30
550TAM	15:30	16:45	1:15
755KLM	16:40	17:40	1:00
927AA	20:25	20:52	0:27
551TAM	21:47	22:31	0:44
902AA	22:00	23:03	1:03
1441LAN	22:44	0:03	1:19
1446LAN	23:14	23:50	0:36
1733LAN	1:56	2:50	0:54
PROMEDIO	0:55		

Tabla 3.7 Resumen de tiempos Jueves

VIERNES

VUELOS	SALIDA GG	LLEGADA GG	TIEMPO TRANSCURRIDO HH/MM
1631LAN	4:22	5:22	1:00
539LAN	4:22	5:22	1:00
VIP,SAB	4:01	4:50	0:49
948AA	4:44	5:36	0:52
NAC.LAN	5:30	5:52	0:22
1447LAN	7:35	8:40	1:05
1553LAN	8:40	9:15	0:35
1561LAN	10:30	11:00	0:30
1630LAN	11:20	12:10	0:50
538LAN	11:35	12:15	0:40
1512LAN	13:10	14:10	1:00
1442LAN	13:10	14:10	1:00
1552LAN	14:15	14:45	0:30
550TAM	14:55	16:30	1:35
1730LAN	15:20	16:10	0:50
1560LAN	15:35	16:10	0:35
755KLM	16:35	17:30	0:55
927AA	19:42	20:00	0:18
NAC.LAN	21:25	22:06	0:41
551TAME	21:20	21:54	0:34
902AA	22:00	22:47	0:47
1441LAN	22:14	0:10	1:56
1446LAN	22:14	0:10	1:56
PROMEDIO	0:52		

Tabla 3.8 Resumen de tiempos Viernes

SÁBADO

VUELOS	SALIDA GG	LLEGADA GG	TIEMPO TRANSCURRIDO HH/MM
1631LAN	4:26	6:01	1:35
539LAN	4:26	6:01	1:35
VIP,SAB	4:50	5:35	0:45
948AA	4:50	5:35	0:45
NAC.LAN	5:30	6:54	1:24
1447LAN	7:40	8:30	0:50
1553LAN	8:50	10:00	1:10
1551LAN	10:15	11:00	0:45
1630LAN	11:40	12:30	0:50
538LAN	11:40	12:30	0:50
1560LAN	12:45	13:30	0:45
1510LAN	13:05	13:35	0:30
1931LAN	14:18	14:40	0:22
1552LAN	15:26	16:00	0:34
550TAM	15:31	16:37	1:06
755KLM	16:51	17:35	0:44
1930LAN	19:23	19:48	0:25
927AA	19:34	20:25	0:51
551 LAN	21:25	22:00	0:35
1521LAN	20:51	21:18	0:27
902AA	22:04	23:15	1:11
1733LAN	22:15	23:38	1:23
547LAN	22:15	23:38	1:23
1743LAN	23:17	23:50	0:33
1446LAN	23:17	23:50	0:33
PROMEDIO	0:52		

Tabla 3.9 Resumen de tiempos Sábado

DOMINGO

VUELOS	SALIDA GG	LLEGADA GG	TIEMPO TRANSCURRIDO HH/MM
1631LAN	4:23	5:26	1:03
539LAN	4:23	5:26	1:03
VIP,SAB	4:24	5:28	1:04
948AA	4:28	5:09	0:41
NAC.LAN	4:33	5:46	1:13
1447LAN	5:12	7:25	2:13
1553LAN	7:55	8:40	0:45
1551LAN	8:45	9:15	0:30
1630LAN	10:15	10:50	0:35
538LAN	11:30	12:00	0:30
1560LAN	11:35	12:20	0:45
1510LAN	11:35	12:20	0:45
1931LAN	14:00	14:25	0:25
1552LAN	15:20	16:15	0:55
550TAM	15:35	16:00	0:25
755KLM	16:38	17:05	0:27
1930LAN	16:50	17:40	0:50
927AA	19:05	20:10	1:05
551 LAN	19:05	20:10	1:05
1521LAN	21:30	22:10	0:40
902AA	22:00	22:48	0:48
1733LAN	22:30	22:48	0:18
547LAN	22:30	23:42	1:12
1743LAN	0:10	0:40	0:30
PROMEDIO	0:49		

Tabla 3.10 Resumen de tiempos Domingo

3.5. Justificación del uso de algoritmo Scatter Search

Se ha utilizado el algoritmo genético de búsqueda dispersa para encontrar una planificación lo más óptima posible, aprovechando al máximo los recursos limitados y costo, que en este caso son los camiones con combustible diésel y los de gasolina, además de los distintos turnos de asignación de los chóferes de vehículos con sus respectivos asistentes.

El algoritmo muestra las mejores parejas (chofer y asistente) según la hora de llegada y salida de cada vuelo y el vehículo a utilizar. Es lógico que los camiones diésel tienen mayor uso debido a que usan un combustible más económico, las camionetas no se tomaron en cuenta debido a que son muy pocos los vuelos VIP y la asignación será bastante obvia.

Este algoritmo genético utiliza distintas combinaciones y evalúa dichas combinaciones, si es satisfactorio el resultado, entonces la solución es tomada en cuenta, de lo contrario se utilizará otra combinación, es clara la utilización de métodos iterativos y combinaciones aleatorias para la resolución del problema.

Básicamente en nuestro caso en particular, nuestro algoritmo toma las distintas combinaciones de los recursos disponibles para un momento dado, y posteriormente decide si es viable o no, y si hay soluciones que faltan en la planificación, significa que no hay recursos disponibles para atender el vuelo.

3.6. Estructura del algoritmo de solución propuesto

El programa será presentado en una aplicación de C# ya que es un derivado de C optimizado y elaborado para el sistema operativo Windows que es el más usado por las computadoras.

3.7. Funcionamiento del algoritmo

Para entender el funcionamiento del algoritmo con fines didácticos, se utilizará un ejemplo de 2 camiones, 1 diésel y 1 a gasolina, 2 vuelos de llegada a hora distinta, los camiones llevarán un operador y un chofer, tendremos como personal 2 choferes y 2 operadores de grúa, el tiempo que tarda el proceso de atención del vuelo es de aproximadamente 45 minutos y el tiempo de llegada de planta al avión es de 4 minutos y viceversa. Adicionalmente, supondremos que los empleados están cumpliendo turnos de 24 horas. Se está usando el horario militar para leer las horas de llegada de los vuelos

Las variables serán:

- CAMION (C1 Y PARA DIESEL , C2 PARA GASOLINA)
- CHOFERES (CH1-CH2, DONDE AMBOS VALEN UN NUMERO ENTRE 100 Y 2400)
- OPERADORES (OP1-OP2)
- TC (TIEMPO EN LLEGAR A ATENDER Y EN REGRESAR A L APLANTA (MINUTOS = 4+4 =8)
- HORA DE LLEGADA (HLLEGADA1-HLLEGADA2)
- TIEMPO ATENDIENDO AL VUELO (TA)

```
HLLEGADA1 = 100
HLLEGADA2 = 140
C1 = 0
C2 = 0
TA =45

        IF  C1 <=HLLEGADA1

            C1= HLLEGADA1 + TA + TC

                IF  CH1<=HLLEGADA1

                    CH1 = HLLEGADA1 + TA + TC

                    IF HLLEGADA1 <= OP1

                    OP1 = CH1

        ELSE IF C2 <=HLLEGADA1

            C2= HLLEGADA1 + TA + TC

                IF  CH1<=HLLEGADA1

                    CH1 = HLLEGADA1 + TA + TC
```

IF HLLEGADA1 <= OP1

OP1 = CH1

ELSE NO HAY CAMIONES DISPONIBLES PARA ATENDER ESTE VUELO
IF C1 <=HLLEGADA1

C1= HLLEGADA1 + TA + TC

IF CH1<=HLLEGADA1

CH1 = HLLEGADA1 + TA + TC

ELSE IF CH2 <= HLLEGADA1

CH2 = HLLEGADA1 + TA + TC

IF HLLEGADA1 <= OP1

OP1 = CH1

ELSE NO HAY CHOFERES PARA ATENDER ESTE VUELO

IF C1 <=HLLEGADA1

C1= HLLEGADA1 + TA + TC

IF CH1<=HLLEGADA1

CH1 = HLLEGADA1 + TA + TC

IF HLLEGADA1 <= OP1

OP1 = CH1

ELSE IF HLLEGADA1 <= OP2

OP2 = CH1

ELSE "NO HAY OPERADORES PARA ATENDER ESTE VUELO"

VUELO 2:

IF C1 <=HLLEGADA2

C1= HLLEGADA2 + TA + TC

IF CH1<=HLLEGADA2

CH1 = HLLEGADA2 + TA + TC

IF HLLEGADA2 <= OP1

OP1 = CH1

ELSE IF C2 <=HLLEGADA2

C2= HLLEGADA2 + TA + TC

IF CH1<=HLLEGADA2

CH1 = HLLEGADA2 + TA + TC

IF HLLEGADA2 <= OP1

OP1 = CH1

ELSE NO HAY CAMIONES DISPONIBLES PARA ATENDER ESTE VUELO
IF C1 <=HLLEGADA1

C1= HLLEGADA2 + TA + TC

IF CH1<=HLLEGADA2

CH1 = HLLEGADA2 + TA + TC

ELSE IF CH2 <= HLLEGADA2

CH2 = HLLEGADA2 + TA + TC

IF HLLEGADA2 <= OP1

OP1 = CH1

"ELSE NO HAY CHOFERES PARA ATENDER ESTE VUELO"

IF C1 <=HLLEGADA2

C1= HLLEGADA2 + TA + TC

IF CH1<=HLLEGADA2

CH1 = HLLEGADA2 + TA + TC

IF HLLEGADA2 <= OP1

OP1 = CH1

ELSE IF HLLEGADA2 <= OP2

OP2 = CH1

ELSE "NO HAY OPERADORES PARA ATENDER ESTE VUELO"

Nótese que siempre se pregunta primero por el camión que funciona a diésel, ya que es el que debe tener mayor uso para minimizar costos.

CAPÍTULO 4

APLICACIÓN COMPUTACIONAL DEL ALGORITMO

4.1. Lenguaje de Programación

El programa será presentado en una aplicación de C#, que es un derivado de C optimizado y elaborado para el sistema operativo Windows que es el más usado por las computadoras.

4.2. Interfaz del Usuario

1. Se deberá ingresar nombre de compañía.

Figura 4.1 Cuadro de diálogo para ingreso nombre de empresa

2. Aparece la pantalla de inicio del programa.

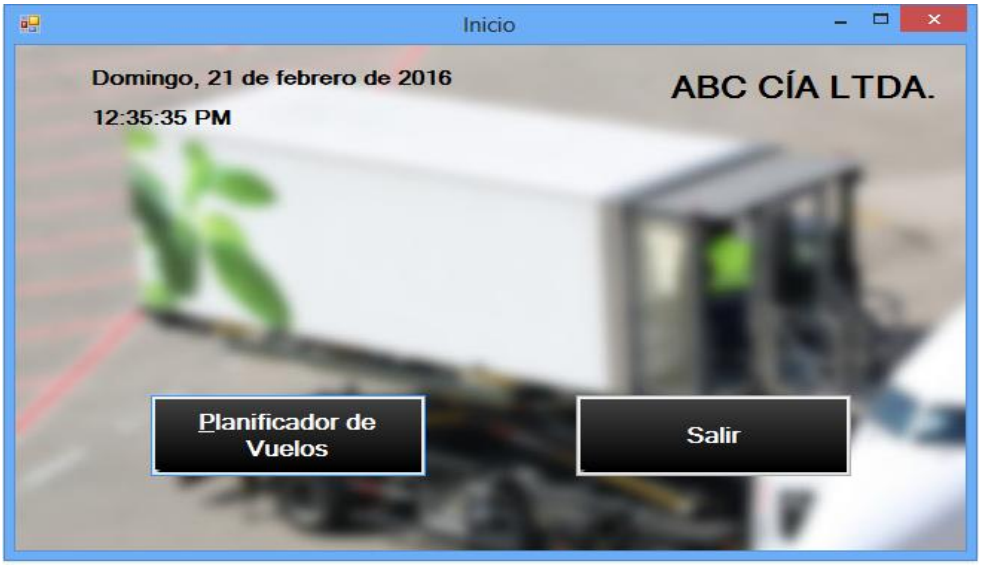


Figura 4.2 Pantalla inicio del programa

3. Presionar botón planificador de vuelos.

Figura 4.3 Cuadro de diálogo para cargar archivo vuelos

Nota: Se deberá seleccionar el tipo de limitador de los archivos planos a ingresar, dependiendo de la configuración de Excel.

4. Seleccionar el archivo plano Vuelos, el mismo contiene el intinerario de vuelos planificados. Se debe mantener el formato de ingreso de información para evitar conflictos de lectura.

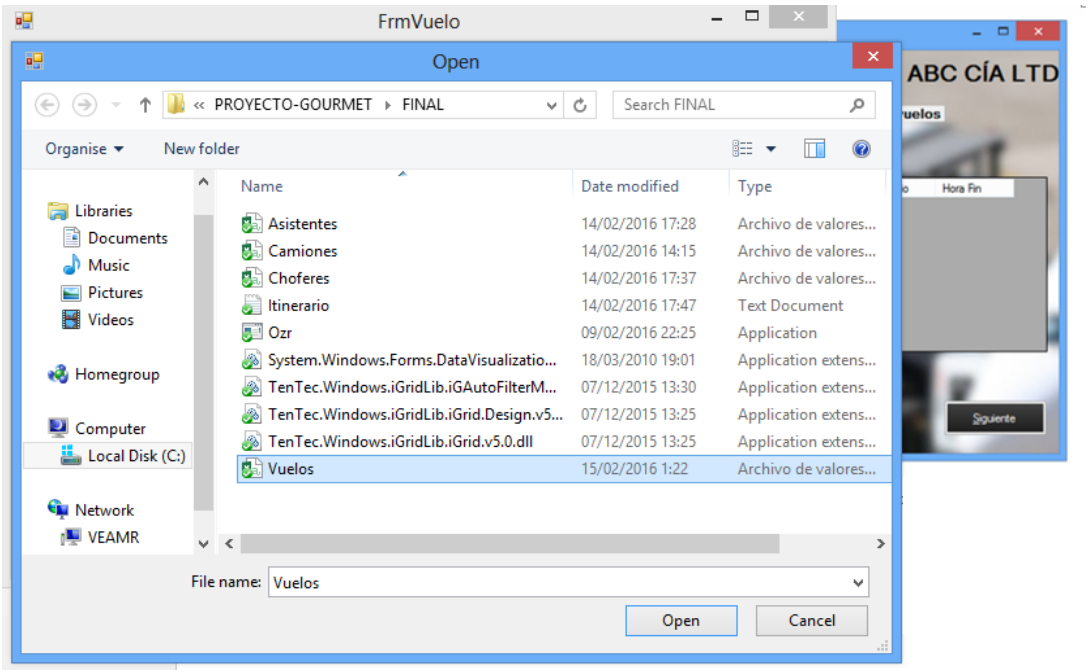


Figura 4.4 Cuadro de diálogo para seleccionar archivo vuelos

5. Luego aparecerá cargado en la pantalla los vuelos como se muestra a continuación. Presionar siguiente para continuar cargando la siguiente librería.

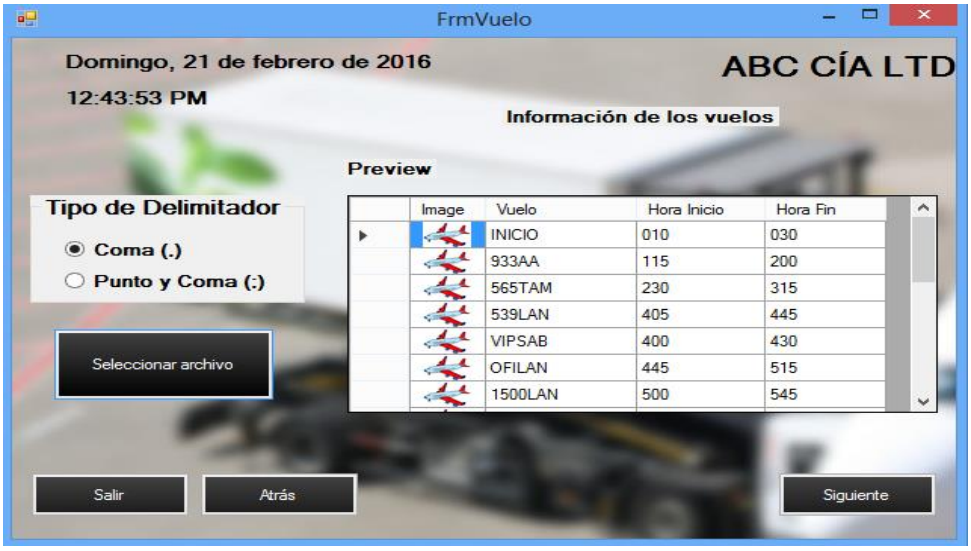


Figura 4.5 Preview archivo Vuelos

6. Seleccionar el archivo plano Asistentes, el mismo contiene horarios planificados previamente del personal. Se debe mantener el formato de ingreso de información para evitar conflictos de lectura.

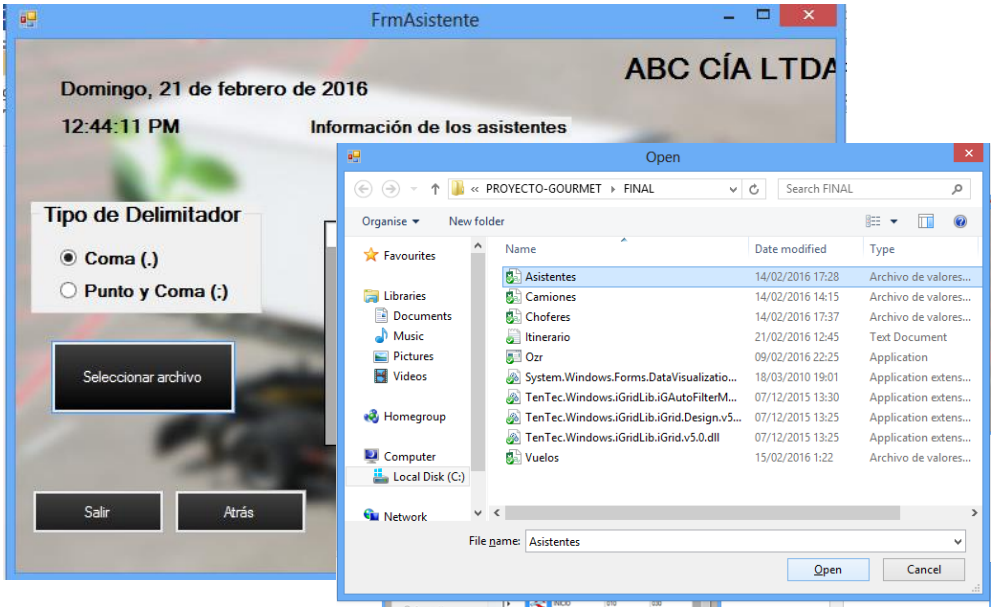


Figura 4.6 Cuadro de diálogo para seleccionar archivo asistentes

7. Luego aparecerá cargado en la pantalla los nombres de los asistentes o guías con el respectivo turno asignado como se muestra a continuación. Presionar siguiente para continuar cargando la siguiente librería.

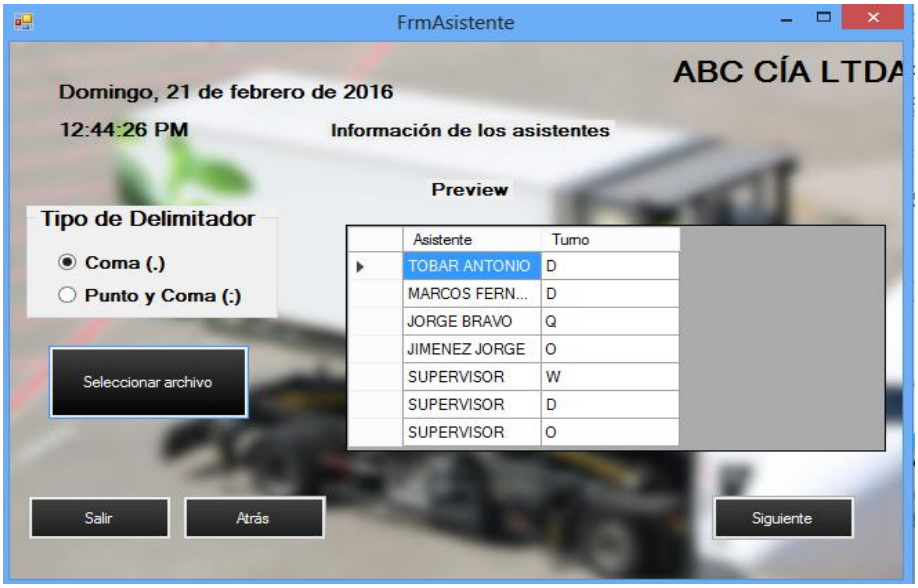


Figura 4.7 Preview Asistentes

8. Seleccionar el archivo plano Camiones, el mismo contiene el número de camiones disponible y tipo de combustible que usa. Se debe mantener el formato de ingreso de información para evitar conflictos de lectura.

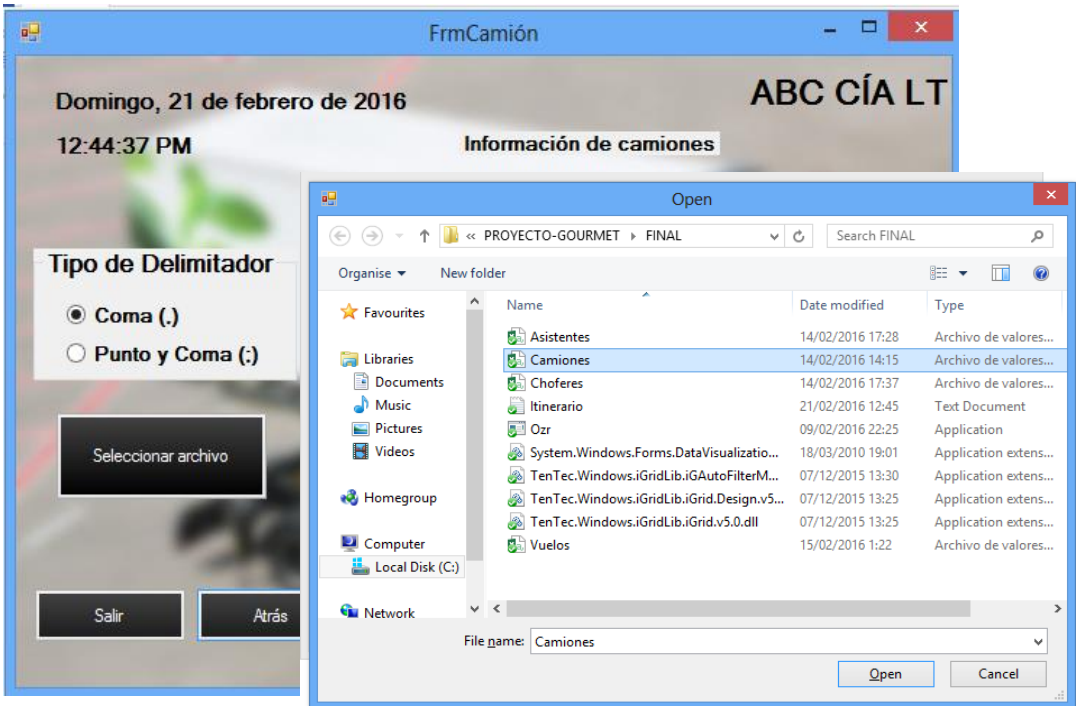


Figura 4.8 Cuadro de diálogo para seleccionar archivo camiones

9. Luego aparecerá cargado en la pantalla los camiones como se muestra a continuación. Presionar siguiente para continuar cargando la siguiente librería.

Figura 4.9 Preview Camiones

10. Seleccionar el archivo plano Choferes, el mismo contiene horarios planificados previamente del personal que realizará esta función. Se debe mantener el formato de ingreso de información para evitar conflictos de lectura.

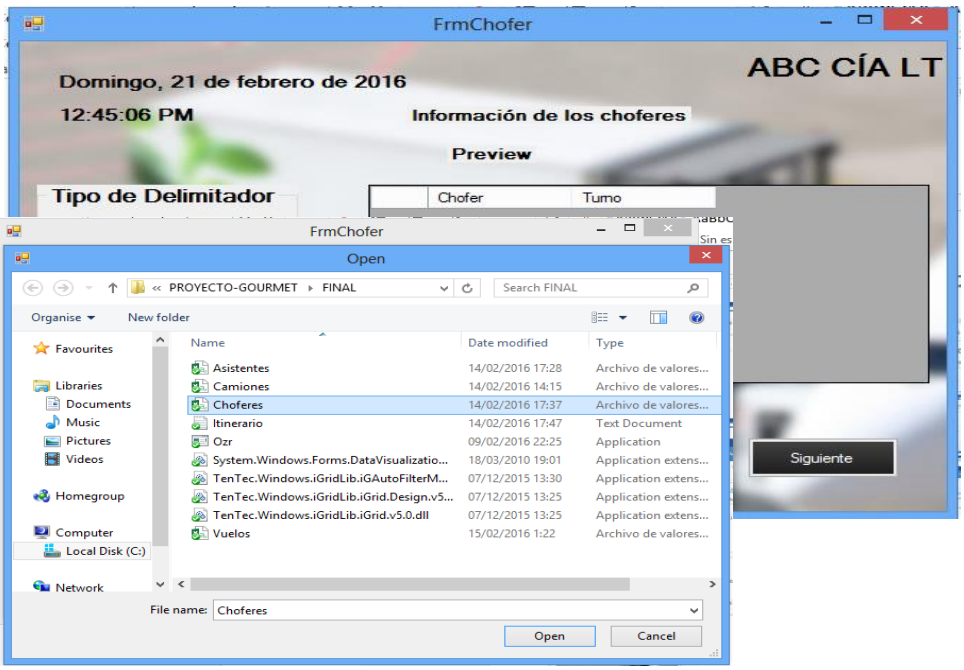


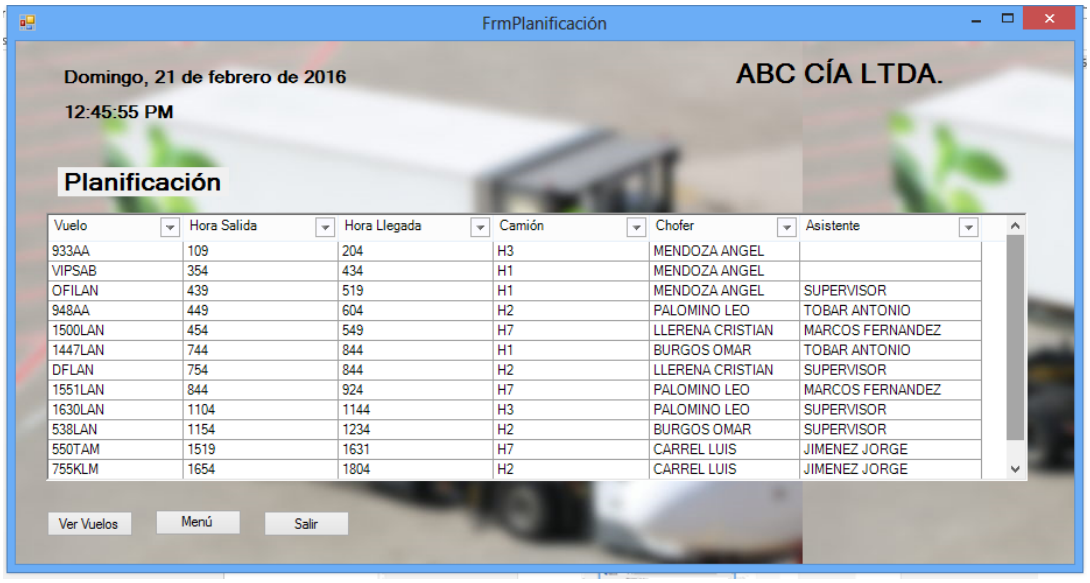
Figura 4.10 Cuadro de diálogo para seleccionar archivo choferes

11. Luego aparecerá cargado en la pantalla los nombres de los choferes con el respectivo turno asignado como se muestra a continuación. Presionar siguiente.

Figura 4.11 Preview Choferes

Se genera la planificación de la operación.

Nota: tomar en cuenta si aparece en blanco el espacio de algún recurso quiere decir que no hay ese recurso disponible en ese momento.



Vuelo	Hora Salida	Hora Llegada	Camión	Chofer	Asistente
933AA	109	204	H3	MENDOZA ANGEL	
VIPSAB	354	434	H1	MENDOZA ANGEL	
OFILAN	439	519	H1	MENDOZA ANGEL	SUPERVISOR
948AA	449	604	H2	PALOMINO LEO	TOBAR ANTONIO
1500LAN	454	549	H7	LLERENA CRISTIAN	MARCOS FERNANDEZ
1447LAN	744	844	H1	BURGOS OMAR	TOBAR ANTONIO
DFLAN	754	844	H2	LLERENA CRISTIAN	SUPERVISOR
1551LAN	844	924	H7	PALOMINO LEO	MARCOS FERNANDEZ
1630LAN	1104	1144	H3	PALOMINO LEO	SUPERVISOR
538LAN	1154	1234	H2	BURGOS OMAR	SUPERVISOR
550TAM	1519	1631	H7	CARREL LUIS	JIMENEZ JORGE
755KLM	1654	1804	H2	CARREL LUIS	JIMENEZ JORGE

Figura 4.12 Planificación Obtenida

Adicionalmente si se presiona el botón ver vuelos se generará automáticamente un gráfico de Gantt que muestra las horas de cada uno de los vuelos de la planificación.

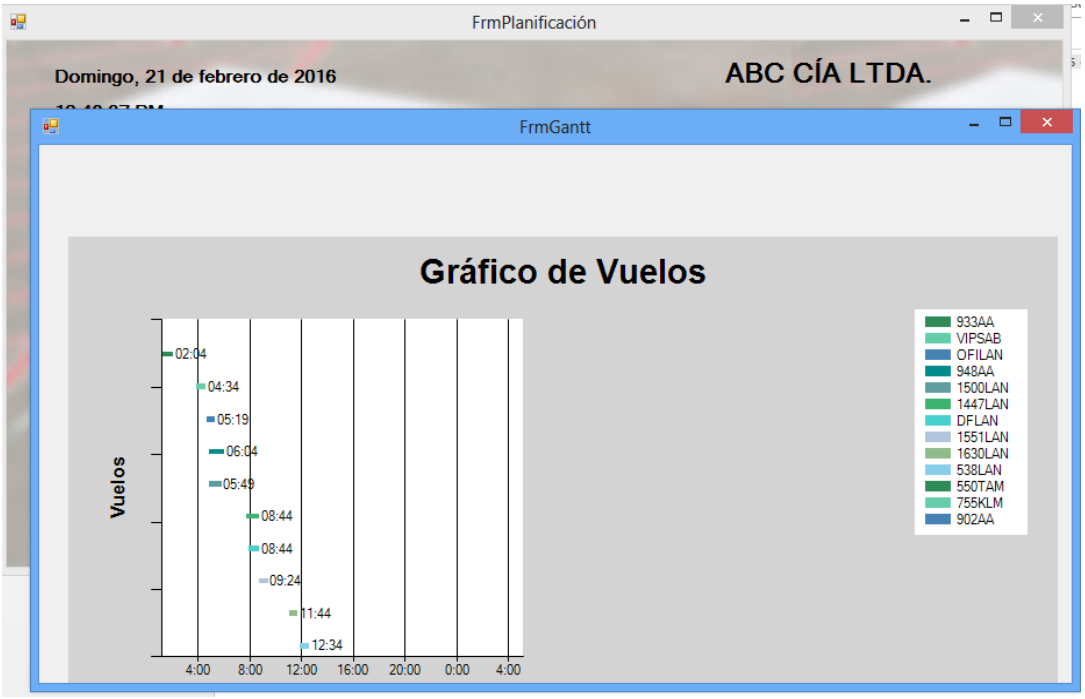


Figura 4.13 Gráfica Gantt

4.3. Resultados Obtenidos

No.	Vuelo	Hora De Salida de Planta	Hora De Llegada a Planta	Camion	Chofer	Asistente
1	NACLAN2	2144	2209	H1	HERRERA LUIS	JIMENEZ JORGE
2	VIPSAB	406	444	H1	PALOMINO LEO	MARCOS FERNANDEZ
3	551TAM	2130	2210	H3	CARREL LUIS	JORGE BRAVO
4	1551LAN	844	924	H3	LLERENA CRISTIAN	MARCOS FERNANDEZ
5	1632LAN	2019	2104	H2	HERRERA LUIS	JORGE BRAVO
6	1553LAN	954	1049	H7	PALOMINO LEO	MARCOS FERNANDEZ
7	539LAN	159	254	H3	MENDOZA ANGEL	VILLACRECES ALEXANDER
8	1447LAN	744	844	H7	BURGOS OMAR	TOBAR ANTONIO
9	1730LAN	1500	1601	H3	CARREL LUIS	JIMENEZ JORGE
10	902AA	2224	2329	H1	MENDOZA ANGEL	JORGE BRAVO
11	1446LAN	2234	2344	H3	RODOLFO RIVAS	VILLACRECES ALEXANDER
12	755KLM	1654	1804	H3	HERRERA LUIS	DOMINGUEZ NELSON
13	1441LAN	2234	2344	H7	HERRERA LUIS	DOMINGUEZ NELSON
14	927AA	1950	2104	H1	RODOLFO RIVAS	JIMENEZ JORGE
15	948AA	449	604	H3	PALOMINO LEO	MARCOS FERNANDEZ
16	538LAN	1129	1249	H3	BURGOS OMAR	MARCOS FERNANDEZ
17	NACLAN	454	640	H2	LLERENA CRISTIAN	TOBAR ANTONIO

Tabla 4.1 Planificación obtenida Corrida 1

No.	Vuelo	Hora De Salida de Planta	Hora De Llegada a Planta	Camion	Chofer	Asistente
1	NACLAN2	2144	2209	H3	MENDOZA ANGEL	JIMENEZ JORGE
2	VIPSAB	406	444	H3	PALOMINO LEO	MARCOS FERNANDEZ
3	551TAM	2130	2210	H7	HERRERA LUIS	JORGE BRAVO
4	1551LAN	844	924	H7	PALOMINO LEO	MARCOS FERNANDEZ
5	1632LAN	2019	2104	H7	RODOLFO RIVAS	JORGE BRAVO
6	1553LAN	954	1049	H1	LLERENA CRISTIAN	TOBAR ANTONIO
7	539LAN	159	254	H7	MENDOZA ANGEL	VILLACRECES ALEXANDER
8	1447LAN	744	844	H1	BURGOS OMAR	TOBAR ANTONIO
9	1730LAN	1500	1601	H1	CARREL LUIS	JIMENEZ JORGE
10	902AA	2224	2329	H2	HERRERA LUIS	JORGE BRAVO
11	1446LAN	2234	2344	H3	RODOLFO RIVAS	DOMINGUEZ NELSON
12	755KLM	1654	1804	H2	RODOLFO RIVAS	JIMENEZ JORGE
13	1441LAN	2234	2344	H7	MENDOZA ANGEL	VILLACRECES ALEXANDER
14	6464IB	1824	1934	H1	RODOLFO RIVAS	DOMINGUEZ NELSON
15	927AA	1950	2104	H2	HERRERA LUIS	DOMINGUEZ NELSON
16	948AA	449	604	H3	PALOMINO LEO	MARCOS FERNANDEZ
17	538LAN	1129	1249	H1	BURGOS OMAR	TOBAR ANTONIO
18	NACLAN	454	640	H7	LLERENA CRISTIAN	TOBAR ANTONIO
19	538LAN	1129	1249	H1	BURGOS OMAR	TOBAR ANTONIO

Tabla 4.2 Planificación obtenida Corrida 2

No.	Vuelo	Hora De Salida de Planta	Hora De Llegada a Planta	Camion	Chofer	Asistente
1	NACLAN2	2144	2209	H2	RODOLFO RIVAS	VILLACRECES ALEXANDER
2	VIPSAB	406	444	H3	MENDOZA ANGEL	MARCOS FERNANDEZ
3	551TAM	2130	2210	H1	MENDOZA ANGEL	JIMENEZ JORGE
4	1551LAN	1124	1207	H3	PALOMINO LEO	TOBAR ANTONIO
5	1632LAN	2019	2104	H2	HERRERA LUIS	DOMINGUEZ NELSON
6	1553LAN	954	1049	H3	PALOMINO LEO	TOBAR ANTONIO
7	539LAN	159	254	H3	MENDOZA ANGEL	VILLACRECES ALEXANDER
8	1550LAN	1358	1455	H3		
9	1447LAN	744	844	H7	LLERENA CRISTIAN	TOBAR ANTONIO
10	1730LAN	1500	1601	H7	CARREL LUIS	JIMENEZ JORGE
11	902AA	2224	2329	H2	MENDOZA ANGEL	VILLACRECES ALEXANDER
12	1630LAN	1114	1222	H1	LLERENA CRISTIAN	MARCOS FERNANDEZ
13	1446LAN	2234	2344	H3	HERRERA LUIS	JORGE BRAVO
14	755KLM	1654	1804	H1	RODOLFO RIVAS	JIMENEZ JORGE
15	1441LAN	2234	2344	H1	RODOLFO RIVAS	DOMINGUEZ NELSON
16	6464IB	1824	1934	H2	CARREL LUIS	DOMINGUEZ NELSON
17	927AA	1950	2104	H3	RODOLFO RIVAS	JORGE BRAVO
18	948AA	449	604	H1	LLERENA CRISTIAN	MARCOS FERNANDEZ
19	1631LAN	424	540	H7	PALOMINO LEO	TOBAR ANTONIO

Tabla 4.3 Planificación obtenida Corrida 3

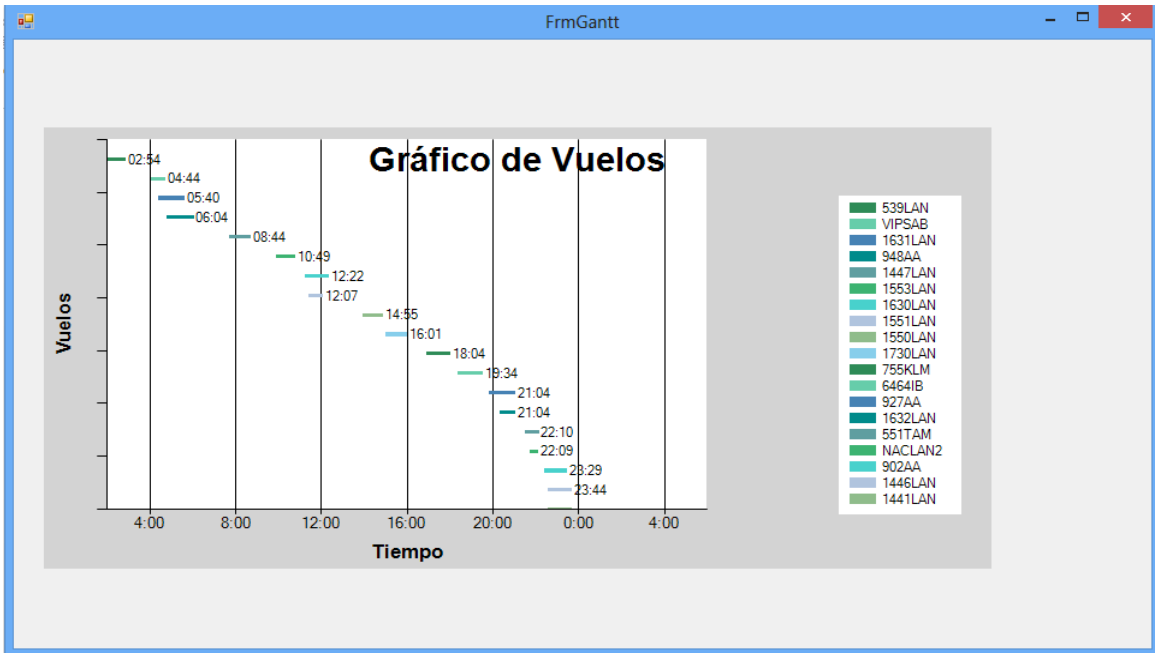


Figura 4.14 Gráfico de Vuelos Corrida 3

Notar que en este corrida en el vuelo 1550LAN, no hay asignado recurso humano, lo que quiere decir que en ese horario no hay ni chofer ni asistente disponibles pero si se

cuenta con un camión disponible para la dicha operación, en esta instancia el supervisor de turno deberá tomar una decisión en base a ello para subsanar falta de personal en ese período de tiempo.

Cuando el vuelo no aparezca dentro del resultado de la planificación obtenida, quiere decir que no hay ningún recurso disponible para la atención de este vuelo.

4.4. Tiempo de Ejecución

El tiempo de ejecución es de 1 minuto 30 segundo, tomando en consideración tiempo necesario para la selección de archivos. Lo que tomaba 45 minutos por turno, con la aplicación de esta herramienta se puede realizar en un tiempo mínimo y por una sola vez al inicio de toda la operación, en caso de existir alguna variación en el itinerario se podrá realizar el debido ajuste y se deberá correr nuevamente la aplicación.

4.5. Conclusiones y Recomendaciones

4.5.1. Conclusiones

En referencia al objetivo principal de este proyecto de graduación, el mismo que está centrado en el desarrollo de una herramienta, que permita obtener una planificación que optimice el uso de recursos para la atención de vuelos de la empresa ABC en el AIJJO, se concluye que:

- Se desarrolló un algoritmo basado en Scatter Search, que resolvió problemas de asignación de recursos basados en ventanas horarias y tiempos promedios que dura cada actividad (vuelos), tomando en consideración tiempos requeridos para sub actividades que están inmersas en dicho proceso, como lo es tiempos de recorridos desde puntos de partida (planta) hasta puntos de llegada (aeronave a ser abastecida)
- Se verificó que efectivamente se hace una asignación correcta de vehículos y recurso humano a los diferentes vuelos.

- Se realizaron pruebas de campo y se concluye que la asignación que realiza el programa es un 90% coincidente con la que realiza el supervisor de turno de forma manual.
- Se concluye que la herramienta presenta una interfaz amigable al usuario, fácil de implementación. El tiempo que tomó a cada supervisor entender el funcionamiento fue de 20 minutos.
- Efectivamente existe caso en los que el supervisor por cambios no programados en la operación tendrá que tomar una decisión fuera de lo planificado. Como por ejemplo cargar 2 vuelos de la misma aerolínea en un solo camión para evitar el retorno a la planta, ya que de hacerlo perdería tiempo valioso en la operación. Recordar que en lo que respecta a los aeropuertos, juega un papel fundamental agentes exógenos como el clima, paros de trabajadores del gremio, daño de aviones por diversos motivos (aves que ingresan a la turbinas, rayos que dañan el fuselaje de la aeronave etc.)
- Se deberá hay tomar en consideración que en este tipo de problemas, en los que las tareas son ejecutadas por el recurso humano, es decir no se trata de una producción en serie, en que los tiempos son fijos de acuerdo a la velocidad de la máquina. Tiene mucha injerencia la agilidad con la que cada par (chofer y asistente) realice su trabajo de manera rápida, respetando las políticas de la empresa y procedimientos de seguridad establecidos por cada uno de los clientes (aerolíneas).

4.5.2. Recomendaciones

- Para mejorar el uso de esta herramienta, se podría incluir gráficos que permitan conocer la cantidad de combustible usado y cuanto representa esto en términos monetarios, de igual manera que se muestre el costo del recurso humano. Notar que el algoritmo como busca la optimización del uso de los recursos, evita la incurrencia en horas extras, es por ello que por cambios de itinerario la decisión como antes lo hemos mencionado, será tomada por el supervisor de turno.

- También se propone extender el algoritmo expuesto en este trabajo, incluyendo sheduling, de tal manera que de manera previa a la planificación de asignación de personal a los vuelos, se obtenga la cantidad de recursos (vehículos/personal) necesarios para una n cantidad de vuelos. Esto se tendría que realizar de manera mensual debido a que de lo investigado las aerolíneas planifican su operación internacional mes a mes, tomando en consideración que cambios mayores son informados con 2-3 meses de anticipación.

- Se recomienda desarrollar una aplicación web y algún tipo de app móvil para dispositivos electrónicos como teléfonos celulares y tabletas, para que los usuarios tengan mayor facilidad de acceso.

- Mejorar los archivos de los distintos elementos, mediante la creación de una base de datos para un fácil acceso.

APÉNDICE

Librerías del algoritmo

Programa

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Planificador
{
   class Program
   {
      static void Main(string[] args)
      {
         Random rnd = new Random();
         StreamReader reader = new StreamReader(File.OpenRead("Vuelos.csv"));
         List<string> listA = new List<string>();
         List<int> listB = new List<int>();
         List<int> listC = new List<int>();
         reader.ReadLine();
         string[] aux = (reader.ReadLine()).Split(',');
         int offset = int.Parse(aux[1]);
         while (!reader.EndOfStream){
            string line = reader.ReadLine();
            if (line.Equals("")){
               break;
            }
            string[] values = line.Split(',');
            listA.Add(values[0]);
```

```
    listB.Add(int.Parse(values[1]));
    listC.Add(int.Parse(values[2]));
}
//cargar intinerario
Intinerario intinerario = new Intinerario();
for (int i = 0; i < listA.Count; ++i){
    intinerario.add(new Vuelo(listA[i], listB[i], listC[i], offset));
}
//fin cargar intinerario
reader = new StreamReader(File.OpenRead("Camiones.csv"));
listA = new List<string>();
List<string> listD = new List<string>();
reader.ReadLine();
while (!reader.EndOfStream){
    string line = reader.ReadLine();
    if (line.Equals("")){
        break;
    }
    string[] values = line.Split(',');
    listA.Add(values[0]);
    listD.Add(values[1]);
}
//cargar recurso camion
PoolCamiones poolCamiones = new PoolCamiones();
for (int i = 0; i < listA.Count; ++i){
    poolCamiones.add(new RecursoCamion(listA[i], listD[i]));
}
//fin cargar recurso camion
reader = new StreamReader(File.OpenRead("Choferes.csv"));
listA = new List<string>();
listD = new List<string>();
reader.ReadLine();
while (!reader.EndOfStream){
    string line = reader.ReadLine();
    if (line.Equals("")){
        break;
    }
    string[] values = line.Split(',');
```

```
    listA.Add(values[0]);
    listD.Add(values[1]);
}
//cargar los recursos choferes
PoolChoferes poolChoferes = new PoolChoferes();
for (int i = 0; i < listA.Count; ++i){
    poolChoferes.add(new RecursoChofer(listA[i], listD[i]));
}
//fin cargar los recursos
reader = new StreamReader(File.OpenRead("Asistentes.csv"));
listA = new List<string>();
listD = new List<string>();
reader.ReadLine();
while (!reader.EndOfStream){
    string line = reader.ReadLine();
    if (line.Equals("")){
        break;
    }
    string[] values = line.Split(',');
    listA.Add(values[0]);
    listD.Add(values[1]);
}
//cargar los recursos asistentes
PoolAsistentes poolAsistentes = new PoolAsistentes();
for (int i = 0; i < listA.Count; ++i){
    poolAsistentes.add(new RecursoAsistente(listA[i], listD[i]));
}
//fin cargar los recursos
listD = null;
listC = null;
listB = null;
listA = null;
reader.Close();
//ordenar los vuelos por mas cortos a mas largos
intinerario.sort("cronograma");
//asignar todos los camiones a los vuelos
PoolCamiones camionesDisponiblesDiesel = new PoolCamiones();
PoolCamiones camionesDisponiblesGasolina = new PoolCamiones();
```

```
        for (int i = 0; i < intinerario.getIntinerario().Count; ++i){

camionesDisponiblesDiesel.setPoolCamiones(poolCamiones.getCamionesDisponiblesDiesel(intin
erario.get(i)));
            if (camionesDisponiblesDiesel.getPoolCamiones().Count != 0) {
               //escoger camion aleatorio
               int camionElegido = rnd.Next(camionesDisponiblesDiesel.getPoolCamiones().Count);
               camionesDisponiblesDiesel.usarCamion(camionElegido, intinerario.get(i));

//camionesDisponibles.setPoolCamiones(poolCamiones.getCamionesDisponibles(intinerario.get(i)
));
            }
            else {

camionesDisponiblesGasolina.setPoolCamiones(poolCamiones.getCamionesDisponiblesGasolina(
intinerario.get(i)));
               if (camionesDisponiblesGasolina.getPoolCamiones().Count != 0){
                  //escoger camion aleatorio
                  int                                camionElegido                                =
rnd.Next(camionesDisponiblesGasolina.getPoolCamiones().Count);
                  camionesDisponiblesGasolina.usarCamion(camionElegido, intinerario.get(i));

//camionesDisponibles.setPoolCamiones(poolCamiones.getCamionesDisponibles(intinerario.get(i)
));
               }
            }
         }
         //si el vuelo no tiene camion asignado se saca del intinerario xq no se puede atender
         for (int i = 0; i < intinerario.getIntinerario().Count; ++i){
            if (intinerario.get(i).getCamion() == null){
               intinerario.remove(intinerario.get(i));
            }
         }
         //asignar todos los choferes a los vuelos
         PoolChoferes choferesDisponibles = new PoolChoferes();
         for (int i = 0; i < intinerario.getIntinerario().Count; ++i){

choferesDisponibles.setPoolChoferes(poolChoferes.getChoferesDisponibles(intinerario.get(i)));
```

```
            if (choferesDisponibles.getPoolChoferes().Count != 0){
               //escoger chofer aleatorio
               int choferElegido = rnd.Next(choferesDisponibles.getPoolChoferes().Count);
               choferesDisponibles.usarChofer(choferElegido, intinerario.get(i));

//choferesDisponibles.setPoolChoferes(poolChoferes.getChoferesDisponibles(intinerario.get(i)));
            }
         }
         //si el vuelo no tiene chofer asignado se saca del intinerario xq no se puede atender
         for (int i = 0; i < intinerario.getIntinerario().Count; ++i){
            if (intinerario.get(i).getChofer() == null){
               intinerario.remove(intinerario.get(i));
            }
         }
         //asignar todos los asistentes a los vuelos
         PoolAsistentes asistentesDisponibles = new PoolAsistentes();
         for (int i = 0; i < intinerario.getIntinerario().Count; ++i){

asistentesDisponibles.setPoolAsistentes(poolAsistentes.getAsistentesDisponibles(intinerario.get(i)
));
            if (asistentesDisponibles.getPoolAsistentes().Count != 0){
               //escoger asistente aleatorio
               int asistenteElegido = rnd.Next(asistentesDisponibles.getPoolAsistentes().Count);
               asistentesDisponibles.usarAsistente(asistenteElegido, intinerario.get(i));

//asistentesDisponibles.setPoolAsistentes(poolAsistentes.getAsistentesDisponibles(intinerario.get(i
)));
            }
         }
         //si el vuelo no tiene asistente asignado se saca del intinerario xq no se puede atender
         for (int i = 0; i < intinerario.getIntinerario().Count; ++i){
            if (intinerario.get(i).getAsistente() == null){
               intinerario.remove(intinerario.get(i));
            }
         }
         intinerario.sort("tiempo");
         File.Delete("Intinerario.csv");
         StreamWriter writer = new StreamWriter(File.OpenWrite("Intinerario.csv"));
```

```
        writer.WriteLine("Vuelo,Hora De Salida de Planta,Hora De Llegada a
Planta,Camion,Chofer,Asistente");
        for (int i = 0; i < intinerario.getIntinerario().Count; ++i){
           writer.WriteLine(intinerario.get(i).getIdentificador() + "," +
intinerario.get(i).getHoraInicio() + "," + intinerario.get(i).getHoraFin() + "," +
intinerario.get(i).getCamion() + "," + intinerario.get(i).getChofer() + "," +
intinerario.get(i).getAsistente());
        }
```

Comparador De Vuelos Por Programa

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Planificador
{
   class ComparadorDeVuelosPorCronograma : IComparer<Vuelo>
   {
      public int Compare(Vuelo x, Vuelo y)
      {
         if (x == null)
         {
            if (y == null)
            {
               // If x is null and y is null, they're
               // equal.
               return 0;
            }
            else
            {
               // If x is null and y is not null, y
               // is greater.
               return -1;
            }
         }
         else
         {
            // If x is not null...
            //
            if (y == null)
            // ...and y is null, x is greater.
            {
               return 1;
            }
            else
            {
               // ...and y is not null, compare the
               // lengths of the two strings.
               //

               int retval = (x.getHoraInicio()).CompareTo(y.getHoraInicio());
```

```
                if (retval != 0)
                {
                   // If the strings are not of equal length,
                   // the longer string is greater.
                   //
                   return retval;
                }
                else
                {
                   // If the strings are of equal length,
                   // sort them with ordinary string comparison.
                   //
                   return 1;
                }
             }
          }
       }
    }
}
```

Comparador De Vuelos Por Tiempo

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Planificador
{
   class ComparadorDeVuelosPorTiempo : IComparer<Vuelo>
   {
      public int Compare(Vuelo x, Vuelo y)
      {
         if (x == null)
         {
            if (y == null)
            {
               // If x is null and y is null, they're
               // equal.
               return 0;
            }
            else
            {
               // If x is null and y is not null, y
               // is greater.
               return -1;
            }
         }
         else
         {
            // If x is not null...
            //
            if (y == null)
```

```
                // ...and y is null, x is greater.
                {
                    return 1;
                }
                else
                {
                    // ...and y is not null, compare the
                    // lengths of the two strings.
                    //
                    int retval = (x.getMinutos().Length).CompareTo(y.getMinutos().Length);

                    if (retval != 0)
                    {
                        // If the strings are not of equal length,
                        // the longer string is greater.
                        //
                        return retval;
                    }
                    else
                    {
                        // If the strings are of equal length,
                        // sort them with ordinary string comparison.
                        //
                        return 1;
                    }
                }
            }
        }
    }
}
```

Itinerario

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Planificador
{
    class Intinerario
    {
        private List<Vuelo> intinerario = new List<Vuelo>();

        public void add(Vuelo vuelo)
        {
            intinerario.Add(vuelo);
        }

        public void remove(Vuelo vuelo)
        {
```

```
            intinerario.Remove(vuelo);
        }

        public List<Vuelo> getIntinerario()
        {
            return intinerario;
        }

        public Vuelo get(int index)
        {
            return intinerario[index];
        }

        public void sort(string tipo)
        {
            if (tipo.Equals("cronograma"))
            {
                ComparadorDeVuelosPorCronograma cc = new ComparadorDeVuelosPorCronograma();
                intinerario.Sort(cc);
            }
            else if (tipo.Equals("tiempo")) {
                ComparadorDeVuelosPorTiempo ct = new ComparadorDeVuelosPorTiempo();
                intinerario.Sort(ct);
            }
        }




    }
}
```

Pool Asistentes

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Planificador
{
    class PoolAsistentes
    {
        private List<RecursoAsistente> poolAsistentes = new List<RecursoAsistente>();


        public void add(RecursoAsistente recursoAsistente)
        {
```

```
            poolAsistentes.Add(recursoAsistente);
        }

        public List<RecursoAsistente> getAsistentesDisponibles(Vuelo vuelo)
        {
            List<RecursoAsistente> disponibles = new List<RecursoAsistente>();

            for (int i = 0; i < poolAsistentes.Count; ++i)
            {
                bool flag = true;
                for (int j = 0; j < vuelo.getMinutos().Length; ++j)
                {
                    if (!poolAsistentes[i].existeMinuto(vuelo.getMinuto(j)))
                    {
                        flag = false;
                        break;
                    }
                }
                if (flag)
                {
                    disponibles.Add(poolAsistentes[i]);
                }
            }
            return disponibles;
        }
        public void usarAsistente(int i, Vuelo vuelo)
        {
            poolAsistentes[i].eraseBracket(vuelo.getMinuto(0),
    vuelo.getMinuto(vuelo.getMinutos().Length - 1));
            vuelo.setAsistente(poolAsistentes[i].getIdentificador());
        }

        public void setPoolAsistentes(List<RecursoAsistente> poolAsistentes)
        {
            this.poolAsistentes = poolAsistentes;
        }

        public List<RecursoAsistente> getPoolAsistentes()
```

```
 {
        return poolAsistentes;
    }
  }
}
```

Pool Camiones

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Planificador
{
   class PoolCamiones
   {
      private List<RecursoCamion> poolCamiones = new List<RecursoCamion>();

      public void add(RecursoCamion recursoCamion)
      {
         poolCamiones.Add(recursoCamion);
      }

      public List<RecursoCamion> getCamionesDisponiblesDiesel(Vuelo vuelo)
      {
         List<RecursoCamion> disponibles = new List<RecursoCamion>();

         for (int i = 0; i < poolCamiones.Count; ++i)
         {
            bool flag = true;
            for (int j = 0; j < vuelo.getMinutos().Length; ++j)
            {
               if (!poolCamiones[i].existeMinuto(vuelo.getMinuto(j)))
               {
                  flag = false;
                  break;
               }
```

```
        }
        if (flag)
        {
            if (poolCamiones[i].getTipo().Equals("D")){
                disponibles.Add(poolCamiones[i]);
            }
        }
    }
    return disponibles;
}

public List<RecursoCamion> getCamionesDisponiblesGasolina(Vuelo vuelo)
{
    List<RecursoCamion> disponibles = new List<RecursoCamion>();

    for (int i = 0; i < poolCamiones.Count; ++i)
    {
        bool flag = true;
        for (int j = 0; j < vuelo.getMinutos().Length; ++j)
        {
            if (!poolCamiones[i].existeMinuto(vuelo.getMinuto(j)))
            {
                flag = false;
                break;
            }
        }
        if (flag)
        {
            if (poolCamiones[i].getTipo().Equals("G"))
            {
                disponibles.Add(poolCamiones[i]);
            }
        }
    }
    return disponibles;
}
```

```
        public void usarCamion(int i,Vuelo vuelo)
        {

poolCamiones[i].eraseBracket(vuelo.getMinuto(0),vuelo.getMinuto(vuelo.getMinutos().Length-
1));
            vuelo.setCamion(poolCamiones[i].getIdentificador());
        }

        public void setPoolCamiones(List<RecursoCamion> poolCamiones)
        {
            this.poolCamiones = poolCamiones;
        }

        public List<RecursoCamion> getPoolCamiones()
        {
            return poolCamiones;
        }
    }
}
```

Pool Choferes

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Planificador
{
    class PoolChoferes
    {
        private List<RecursoChofer> poolChoferes = new List<RecursoChofer>();

        public void add(RecursoChofer recursoChofer)
        {
            poolChoferes.Add(recursoChofer);
        }
```

```
public List<RecursoChofer> getChoferesDisponibles(Vuelo vuelo)
{
   List<RecursoChofer> disponibles = new List<RecursoChofer>();

   for (int i = 0; i < poolChoferes.Count; ++i)
   {
      bool flag = true;
      for (int j = 0; j < vuelo.getMinutos().Length; ++j)
      {
         if (!poolChoferes[i].existeMinuto(vuelo.getMinuto(j)))
         {
            flag = false;
            break;
         }
      }
      if (flag)
      {
         disponibles.Add(poolChoferes[i]);
      }
   }
   return disponibles;
}
public void usarChofer(int i, Vuelo vuelo)
{
   poolChoferes[i].eraseBracket(vuelo.getMinuto(0),
vuelo.getMinuto(vuelo.getMinutos().Length - 1));
   vuelo.setChofer(poolChoferes[i].getIdentificador());
}

public void setPoolChoferes(List<RecursoChofer> poolChoferes)
{
   this.poolChoferes = poolChoferes;
}

public List<RecursoChofer> getPoolChoferes()
{
   return poolChoferes;
}
```

```
  }
}
```

Recurso Asistente

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Planificador
{
   class RecursoAsistente
   {
      string turno;
      string identificador;
      int[] minutos;

      public RecursoAsistente(string identificador, string turno)
      {
         int horaInicio=0;
         int horaFin=0;
         this.identificador = identificador;
         this.turno = turno;
         if (turno.Equals("C") || turno.Equals("FC")){
            horaInicio = 300;
            horaFin = 1130;
         }
         else if (turno.Equals("D") || turno.Equals("FD")){
            horaInicio = 400;
            horaFin = 1300;
         }
         else if (turno.Equals("O") || turno.Equals("FO")){
            horaInicio = 1400;
            horaFin = 2230;
```

```
    }
    else if (turno.Equals("Q") || turno.Equals("FQ")){
       horaInicio = 1600;
       horaFin = 2400;
    }
    else if (turno.Equals("W") || turno.Equals("FW")){
       horaInicio = 2100;
       horaFin = 2400;
    }
    else if (turno.Equals("W1") || turno.Equals("FW1"))
    {
       horaInicio = 0;
       horaFin = 530;
    }
else if (turno.Equals("F"))
    {
       horaInicio = 0;
       horaFin = 2400;
    }
else if (turno.Equals("P")|| turno.Equals("FP"))
    {
       horaInicio = 1500;
       horaFin = 2330;
    }
    int contador = 0;
    for (int i = horaInicio; i < horaFin; ++i)
    {
       ++contador;
       if (((i / 10) % 10) == 6)
       {
          i = i + 40;
       }
    }
    minutos = new int[contador-1];
    contador = horaInicio;
    for (int i = 0; i < minutos.Length; ++i){
       minutos[i] = contador;
       ++contador;
```

```
                if (((contador / 10) % 10) == 6){
                    contador = contador + 40;
                }
            }
        }

        public bool existeMinuto(int minuto)
        {
            for (int i = 0; i < minutos.Length; ++i)
            {
                if (minuto == minutos[i])
                {
                    return true;
                }
            }
            return false;
        }

        public void eraseBracket(int first, int last)
        {
            for (int i = 0; i < minutos.Length; ++i)
            {
                if (minutos[i] >= first && minutos[i] <= last)
                {
                    minutos[i] = -1;
                }
            }
        }

        public string getIdentificador()
        {
            return identificador;
        }

    }
}
```

Recurso Camión

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Planificador
{
   class RecursoCamion
   {
      string tipo;
      string identificador;
      int[] minutos;

      public RecursoCamion(string identificador,string tipo)
      {
         this.tipo = tipo;
         this.identificador = identificador;
         int horaInicio = 0;
         int horaFin = 2400;
         int contador = 0;
         for (int i = horaInicio; i < horaFin; ++i)
         {
            ++contador;
            if (((i / 10) % 10) == 6)
            {
               i = i + 40;
            }
         }
         minutos = new int[contador - 1];
         contador = horaInicio;
         for (int i = 0; i < minutos.Length; ++i)
         {
            minutos[i] = contador;
            ++contador;
```

```
        if (((contador / 10) % 10) == 6)
        {
            contador = contador + 40;
        }
    }
}

public bool existeMinuto(int minuto)
{
    for (int i = 0; i < minutos.Length; ++i) {
        if (minuto == minutos[i]) {
            return true;
        }
    }
    return false;
}

public void eraseBracket(int first,int last)
{
    for (int i = 0; i < minutos.Length; ++i)
    {
        if (minutos[i]>=first&& minutos[i] <= last)
        {
            minutos[i]=-1;
        }
    }
}

public int[] getMinutos()
{
    return minutos;
}

public string getIdentificador()
{
    return identificador;
}
```

```
        public string getTipo()
        {
            return tipo;
        }

        public int getMinuto(int i)
        {
            return minutos[i];
        }

        public void setMinuto(int index,int value)
        {
            minutos[index]=value;
        }
    }
}
```

Recurso Chofer

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Planificador

/// esta es la clase del chofer. Cada if llena la hora fin y la hora inicio, dependiendo del turno del chofer
// Los turnos que se tomaran en cuenta fueron los del mapeo.
{
    class RecursoChofer
    {
        string turno;
        string identificador;
        int[] minutos;
                // este es el constructor de la clase.
```

```
public RecursoChofer(string identificador, string turno)
{
   int horaInicio=0;
   int horaFin=0;
   this.identificador = identificador;
   this.turno = turno;

                    // tomando en cuenta el turno de cada chofer, se asigna la hora inicio y hora fin.
   if (turno.Equals("C") || turno.Equals("FC")){
      horaInicio = 300;
      horaFin = 1130;
   }
   else if (turno.Equals("D") || turno.Equals("FD")){
      horaInicio = 400;
      horaFin = 1300;
   }
 else if (turno.Equals("F"))
   {
      horaInicio = 0;
      horaFin = 2400;
   }
else if (turno.Equals("P")|| turno.Equals("FP"))
   {
      horaInicio = 1500;
      horaFin = 2330;
}
   else if (turno.Equals("O") || turno.Equals("FO")){
      horaInicio = 1400;
      horaFin = 2230;
   }
   else if (turno.Equals("Q") || turno.Equals("FQ")){
      horaInicio = 1600;
      horaFin = 2400;
   }
   else if (turno.Equals("W") || turno.Equals("FW")){
      horaInicio = 2100;
      horaFin = 2400;
   }
```

```
    else if (turno.Equals("W1") || turno.Equals("FW1"))
    {
        horaInicio = 0;
        horaFin = 530;
    }
    int contador = 0;
    for (int i = horaInicio; i < horaFin; ++i)
    {
        ++contador;
        if (((i / 10) % 10) == 6)
        {
            i = i + 40;
        }
    }
    minutos = new int[contador-1];
    contador = horaInicio;
    for (int i = 0; i < minutos.Length; ++i){
        minutos[i] = contador;
        ++contador;
        if (((contador / 10) % 10) == 6){
            contador = contador + 40;
        }
    }
}

public bool existeMinuto(int minuto)
{
    for (int i = 0; i < minutos.Length; ++i)
    {
        if (minuto == minutos[i])
        {
            return true;
        }
    }
    return false;
}

public void eraseBracket(int first, int last)
```

```
        {
            for (int i = 0; i < minutos.Length; ++i)
            {
                if (minutos[i] >= first && minutos[i] <= last)
                {
                    minutos[i] = -1;
                }
            }
        }
             // devuelve el indentificador
        public string getIdentificador()
        {
            return identificador;
        }

    }
}
```

Vuelo

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Planificador

/// Este es la clase vuelo. Tiene como atributos un identificador, un camion/camioneta, un chofer y
un asistente.
{
    class Vuelo
    {

        string identificador;
        private int[] minutos;
        private string camion=null;
```

```
        private string chofer=null;
        private string asistente=null;

            // Este es el constructor de la clase vuelo.
        public Vuelo(string identificador,int horaInicio,int horaFin,int offset) {
          this.identificador = identificador;
                    // esto es para convertir las horas para facilitar el algoritmo
          int aux = offset / 60;
          offset = offset % 60;
          horaInicio = horaInicio - (100*aux)-offset;
          if (((horaInicio / 10) % 10) >= 6)
          {
            horaInicio = horaInicio - 40;
          }
          horaFin = horaFin + (100 * aux) + offset;
          if (((horaFin / 10) % 10) >= 6)
          {
            horaFin = horaFin + 40;
          }
          int contador = 0;
                    // Es ciclo comienza con la hora inicio y va iterando hasta llegar a la hora fin
          for (int i = horaInicio; i <horaFin; ++i)
          {
            ++contador;
            if (((i / 10) % 10) == 6)
            {
              i = i + 40;
            }
          }
                    // la variable contador nos da los minutos. Se realiza contador -1 para facilidad
del arreglo.
          minutos = new int[contador-1];
          contador = horaInicio;
          for (int i= 0; i<minutos.Length; ++i){
            minutos[i] = contador;
            ++contador;
            if (((contador/10)%10)==6) {
              contador = contador + 40;
```

```
        }
    }
}

        /// esta funcion recibe un camion y lo asigna
public void setCamion(string camion)
{
    this.camion=camion;
}
        /// retotnar un camion/camioneta
public string getCamion()
{
    return camion;
}
        /// obtener el identicador
public string getIdentificador()
{
    return identificador;
}

        /// asignar un chofer
public void setChofer(string chofer)
{
    this.chofer = chofer;
}

        /// retornar un chofer
public string getChofer()
{
    return chofer;
}

        /// colocar un asistente
public void setAsistente(string asistente)
{
    this.asistente = asistente;
}
```

```
            /// retornar un asistente
        public string getAsistente()
        {
           return asistente;
        }

            /// retorna un array de minutos
        public int[] getMinutos(){
           return minutos;
        }

            /// retorna un minuto en especifico
        public int getMinuto(int i)
        {
           return minutos[i];
        }

            /// retorna la hora inicio
        public int getHoraInicio()
        {
           return minutos[0];
        }

            /// retorna la hora fin
        public int getHoraFin()
        {
           return minutos[minutos.Length-1];
        }

    }
}
```

BIBLIOGRAFÍA

[1] Glover. F.; Heuristics for integer programming using surrogate constraints. Decis. Sci.,8:156–166 (1977)

[2] Glover. F.; A Template For Scatter Search And Path Relinking; School of Business, University of Colorado, 1998

[3] Escobar A.; "Algoritmo de Búsqueda Tabú"; Optimización matemática Algoritmo de Búsqueda.; Universidad Tecnológica de Pereira - Colombia 2014

[4] Nemhauser G.; Wolsey L. A.; Integer and Combinatorial Optimization; Jhon Willey &Sons; 1998 y 1999.

[5] Martí R., Laguna M.; Scatter Search: Diseño Básico y Estrategias Avanzadas

[6] Campos, V., F. Glover, M. Laguna and R. Martí; An Experimental Evaluation of a Scatter Search for the Linear Ordering Problem, Journal of Global Optimization, 21, 397-414, 2001.

[7] CHARNES, A., COOPER W. W.; Deterministic Equivalents for Optimizing and Satisficing under Chance Constraints; Operations Research, Vol 11, Nº 1, Jan- Feb, 1963, pág. 18-39.

[8] Méndez M.; "Algoritmos Evolutivos y Preferencias del Decisor Aplicados a Problemas de Optimización Multiobjetivo Discretos"; Universidad de Las Palmas de Gran Canaria; Septiembre 2008.

[9] E. Zitzler and L. Thiele; Multiobjective evolutionary algorithms: A compar- ative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation; 3(4):257–271, 1999.

[10] http://servicio.bc.uc.edu.ve/ingenieria/revista/Inge-Industrial/voliv12/art08.pdf Journal of Heuritics; Web site: http://mauricio.resende.info/doc/guidelines.pdf

[11] Fred Glover*, Manuel Laguna*t and Rafael Marti**, Fundamentals of scatter search and path relinking, Control and Cybernetics val. 29 (2000) No. 3

[12] Martí, R., M. Laguna and F. Glover; "Principles of Scatter Search," European Journal of Operational Research; vol. 169, no. 2, pp. 359-372; (2006)

[13] Rafael Martí, Manuel Laguna, Vicente Campos Scatter Search vs. Genetic Algorithms, An Experimental Evaluation with Permutation Problems.

[14] M.g.c. resende, c.c. ribeiro, f. Glover, and r. Martí; Handbook of Metaheuristics.

[15] Aarts y Lenstra, 2003; Blum y Roli, 2003; Glover y Kochenberger, 2003; Martí, 2003; Melián, Moreno y Vega, 2003; Osman y Kelly, 1996, y Sait y Youssef, 1999.

[16] Procedimientos Metaheurísticos en Optimización Combinatoria, Rafael Martí

[17] M.G.C. Resende, C.C.Ribeiro, F. Glover and R. Martí.; Scatter Search and Path-Relinking: Fundamentals, Advances and Applications.

[18] M.G.C. Resende, R. Martí, M. Gallego, and A. Duarte.; GRASP and path relinking for the max-min diversity problem Computers and Operations Re-search; 2008.

[19] H. Pinol a, J.E. Beasley, Scatter Search and Bionomic Algorithms for the aircraft landing problem; European Journal of Operational Research 171 (2006) 439–462

[20] V. Ciesielski, P. Scerri; An anytime algorithm for scheduling of aircraft landing times using genetic algorithms; Australian Journal of Intelligent Information Processing;Systems 4 (1997) 206–213.

[21] Glover , F.; Genetic algorithms and scatter search: Unsuspected potentials, Statistics and Computing 4 (1994) 131– 140.

[22] Laguna M. and Armentano V.; Lessons from Applying and Experimenting with Scatter Search, Technical Report; University of Colorado, 2001.

[23] Laguna M. and Martí R.; The OptQuest Callable Library, Optimization Software Class Libraries, Voss and Woodruff (Eds.), Kluwer Academic Publishers, Boston; 193-215, 2000.