



10. Diseño del catálogo de competencias. La estructura que se propone que tenga un catálogo de competencias es la siguiente:

- Presentación. En este apartado debe explicarse cuál es el objetivo que se pretende alcanzar a través del catálogo de competencias genéricas y la forma en la que este está estructurado.

- Competencias genéricas. Lo deseable es presentar en una tabla de dos columnas la lista de competencias genéricas requeridas en la empresa y la definición de cada una de dichas competencias.

- Competencias genéricas y niveles asociados. En este apartado se propone diseñar una tabla por cada una de las competencias genéricas que integren el catálogo. La estructura propuesta para dicha tabla es la siguiente: un encabezado con la definición de la competencia genérica, y dos columnas en las que se señalen los distintos niveles en que puede ser requerida la competencia, así como la definición de cada nivel. (Véase tabla 4)

- Competencias específicas comunes a cada puesto-tipo. Lo deseable es presentar tantas tablas como puestos-tipo se hayan identificado. Las mencionadas tablas deberán ser de dos columnas: La lista de competencias específicas comunes requeridas en el puesto-tipo y la definición de cada una de dichas competencias.

- Competencias específicas comunes a cada puesto-tipo y niveles asociados. Al igual que con las competencias genéricas, en este apartado se propone diseñar una tabla por cada una de las competencias específicas comunes que integren el catálogo.

#### 9. CONCLUSIONES

El método que se propone busca la efectividad en la gestión del capital humano. Con respecto al término efectividad, Sergio Hernández (2011, 5) explica que “es la habilidad gerencial de lograr la eficiencia y la eficacia en relación con los recursos y objetivos”.

En efecto, el empleo de catálogos de competencias

permite que en las empresas (así como en cualquier otro tipo de organización) se haga lo que se tiene que hacer en materia de recursos humanos y, además, se realice sacando el mayor provecho posible a los recursos con los que se cuenta.

El catálogo de competencias es, como se dijo inicialmente, una herramienta fundamental para la gestión de recursos humanos con un enfoque por competencias. Sirve de base para la elaboración de otra herramienta denominada perfiles de puesto por competencias, la cual será un referente obligatorio en el desarrollo de procesos de planificación, adquisición, conservación, formación y evaluación de competencias.

#### 10. REFERENCIAS BIBLIOGRÁFICAS

Arias Galicia, Fernando (1980). Administración de recursos humanos. México, Trillas.

Buzón, (2005). La incorporación de plataformas virtuales a la enseñanza: una experiencia de formación on line basada en competencias. Revista latinoamericana de tecnología educativa. Vol. 4, núm. 1.

Certo, Samuel C. (1984). Administración moderna. México, Interamericana.

Chiavenato, Idalberto (2001). Administración de recursos humanos. Colombia, McGraw-Hill.

Fayol, Henri (1961). Administración industrial y general. México, Herrero Hermanos.

González, Rafael (2009). La administración, el medio ambiente y la competitividad. Ciencia Administrativa, número 2. México, Universidad Veracruzana.

Hernández, Sergio y Alejandro Pulido (2011). Fundamentos de gestión empresarial, enfoque basado en competencias. México, McGraw-Hill.

Hernández, Sergio (2011). Introducción a la administración. México, McGraw-Hill.

Koontz et al (2008). Administración, una perspectiva global y empresarial. China, McGraw-Hill.

Mejías, Cristina (2000). Los talentos del siglo XXI. Argentina, Planeta.

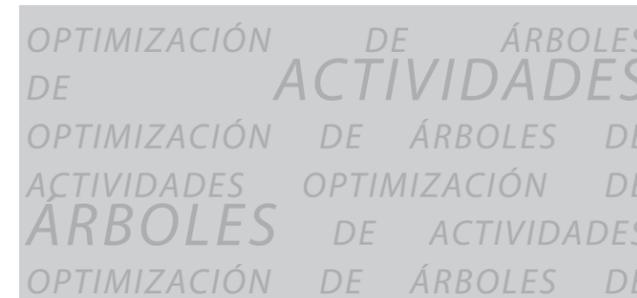
Taylor, Frederick W. (1961). Principios de la administración científica. México, Herrero Hermanos.

Werther, William y Keith Davis (2001). Administración de personal y de recursos humanos. México, McGraw-Hill.

<http://blogs.saschina.org/eugene01pd2018/2010/11/01/culture-iceberg/>

## OPTIMIZACIÓN DE ÁRBOLES DE ACTIVIDADES (WORK BREAKDOWN STRUCTURE), EN LA PLANEACIÓN DE PROYECTOS, UTILIZANDO SIMULACIÓN DE PROCESOS EN UNA MÁQUINA PARALELA.

Dr. Carlos Arturo Vega Lebrún  
Genoveva Rosano Ortega  
Heberto Ferreira Medina.  
José Luis Cendejas Valdez





Sumario: 1. Resumen/Abstract. 2. Introducción. 3. Desarrollo, 4. Datos Obtenidos. 5. Resultados. 6. Discusión y Conclusiones, 7. Referencias bibliográficas

## 1. RESUMEN

La metodología propuesta por el PMI (Project Management Institute, Instituto de Administración de Proyectos, Organismo líder en materia de formación de especialistas en gestión de proyectos), analiza las diferentes etapas en el desarrollo de un proyecto; inicio, planeación, la supervisión, el desarrollo y el cierre. Durante la fase de planeación, uno de los mayores problemas es la asignación de recursos y la programación de tareas. Las metodologías sugieren la construcción de un árbol de actividad conocida como WBS (Work Breakdown Structure: Descomposición de la Estructura de Trabajo), que especifica los procesos y la asignación de recursos. El WBS no proporciona un control de procesos paralelos que se producen habitualmente en cualquier proyecto, las tareas difíciles como la asignación de recursos y la optimización. Esta investigación incluye la optimización basada en la planificación inicial de la WBS y los recursos implicados en un proyecto. En este artículo se describe un algoritmo que se ejecuta muchas tareas en una máquina paralela LAM / MPI (Local Area Multicomputer / Mensaje Passing Interface), para determinar la mejor manera de implementar el proyecto, lo que permite reducir los costos en el alcance del

## ABSTRACT

The methodology proposed by the Project Management Institute (PMI), analyzes different stages in the development of a project; beginning, planning, monitoring, implementing and finishing. During the planning phase, one of the major problems is the resources allocation and tasks scheduling. The methodologies suggest the construction of an activity tree known as WBS (Work Breakdown Structure), which specifies processes and resource allocation. The WBS does not provide control of parallel processes that generally occur in any project, difficult tasks like resource allocation and optimization. This project

includes optimization based on the initial planning of the WBS and the resources involved in a project. In this article is described an algorithm that runs many tasks in a parallel machine LAM/MPI (Local Area Multicomputer / Message Passing Interface), determining the best way to implement the project, allowing the costs reducing in the scope.

Palabras clave: Instituto de Administración de Proyectos, Planeación de Proyectos, Descomposición de la Estructura de Trabajo, Multi cómputo de Área Local, Simulación de Procesos

Keywords: Project Management Institute, Project Planning, Work Breakdown Structure, Local Area Multicomputer, Process Simulation.

## 2. INTRODUCCIÓN

El PMI propone una serie de buenas prácticas para el desarrollo de proyectos de diferente índole, en el PMBOK (Libro Editado por el PMI) se definen varias fases de desarrollo de un proyecto (Institute, 2009); Inicio, planeación, control, ejecución y cierre, ver figura 1. En la fase de planeación la metodología del PMI propone la construcción de un árbol de actividades WBS (Institute, 2009), (Kanik, 2005) donde a grandes rasgos se especifican las actividades, la asignación de recursos y los entregables de un proyecto.

enfoque ascendente y generalmente se basa en la construcción de un listado de tareas o actividades que se van relacionando de abajo hacia arriba hasta lograr un entregable.

3. Otros principios generales. La construcción del WBS debe apegarse a varios principios que permiten construir un estructura jerárquica de todas las tareas que el proyecto tendrá, estos principios pueden revisarse en el PMBOK del PMI (Institute, 2009).

Esta metodología propone una serie de recomendaciones para llevar a cabo el proyecto y no tiene un control preciso sobre la ejecución de procesos en paralelo o la simultaneidad de procesos (conurrencia), complicando así la asignación de los recursos y su optimización. El esquema de planeación propuesto por el PMI está basado en las mejores prácticas y deja a la experiencia de los administradores la optimización y control de las tareas, lo que conlleva a costos extraordinarios si no se tienen herramientas de apoyo.

En esta investigación se propone la optimización basándose en una búsqueda de la mejor forma de ejecutar las tareas definidas en la planeación inicial del WBS; en la figura 2 se observa un ejemplo; se muestra un árbol con 23 actividades y 5 entregables, para un proyecto de telecomunicaciones (Espejel & Ferreira, 2005).

Figura 1. Etapas en el desarrollo de un proyecto donde existe el paralelismo de actividades llamado "Fast Tracking".

El WBS provee un desglose jerárquico de tareas, es un modelo que describe cada una de las tareas a realizar en un proyecto. Esta descripción se realiza en varios niveles progresivos de detalle sobre cada una de las tareas (con identificadores). Es el instrumento de planificación que posibilita un mayor detalle en la fase de planeación. Una de las formas más comunes de construir un WBS es utilizando tres principios:

1. La regla del 100%. Cuando se efectúa el desglose de las actividades o tareas en sucesivos niveles (subordinados a los anteriores), el próximo nivel en la descomposición de un elemento el (nivel-hijo) deberá contener y representar el 100% del trabajo aplicable al nivel inmediato superior (elemento padre).

2. Proceder de abajo-arriba (Bottom-up WBS Development). También es conocido como



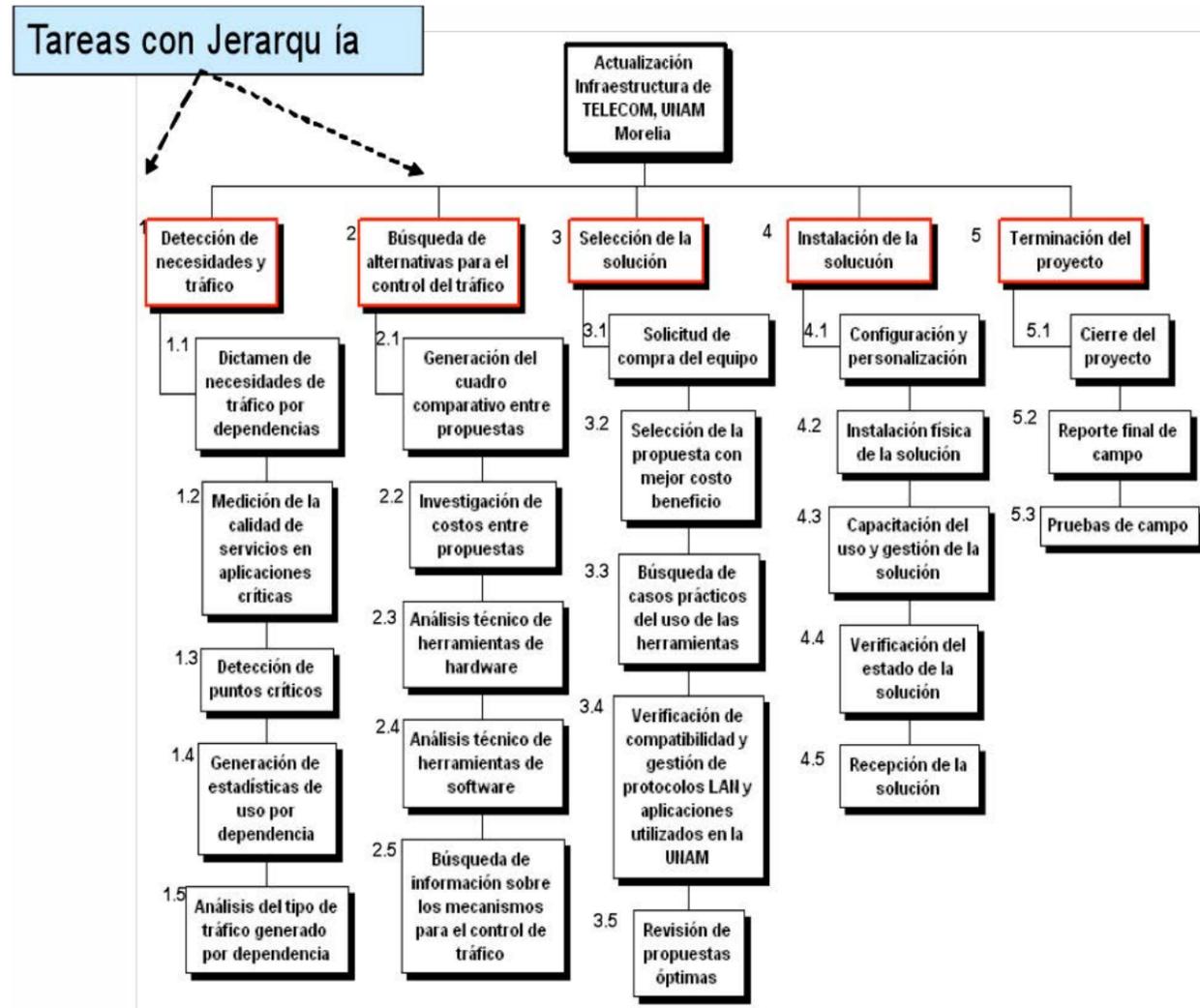


Figura 2. WBS para un proyecto de telecomunicaciones en la UNAM Campus Morelia.

Una asignación de recursos mal planeada puede arrojar mayores costos que al final modifican en gran medida el plan inicial (Kanik, 2005). Se propone la optimización de la planeación de tareas, utilizando un método de búsqueda exhaustivo para encontrar la mejor forma de llevar a cabo el plan. Este método permitirá simular procesos en paralelo y su impacto en la asignación y liberación de los recursos así como los riesgos que pueden ocurrir para cada una de las actividades del proyecto.

El proceso de planeación típico puede observarse en el esquema de la Figura 3 (Srivastava, 2000), en la cual se observa que en la administración de proyectos no se contemplan los procesos en paralelo y su impacto en la asignación de recursos.

### 3. DESARROLLO

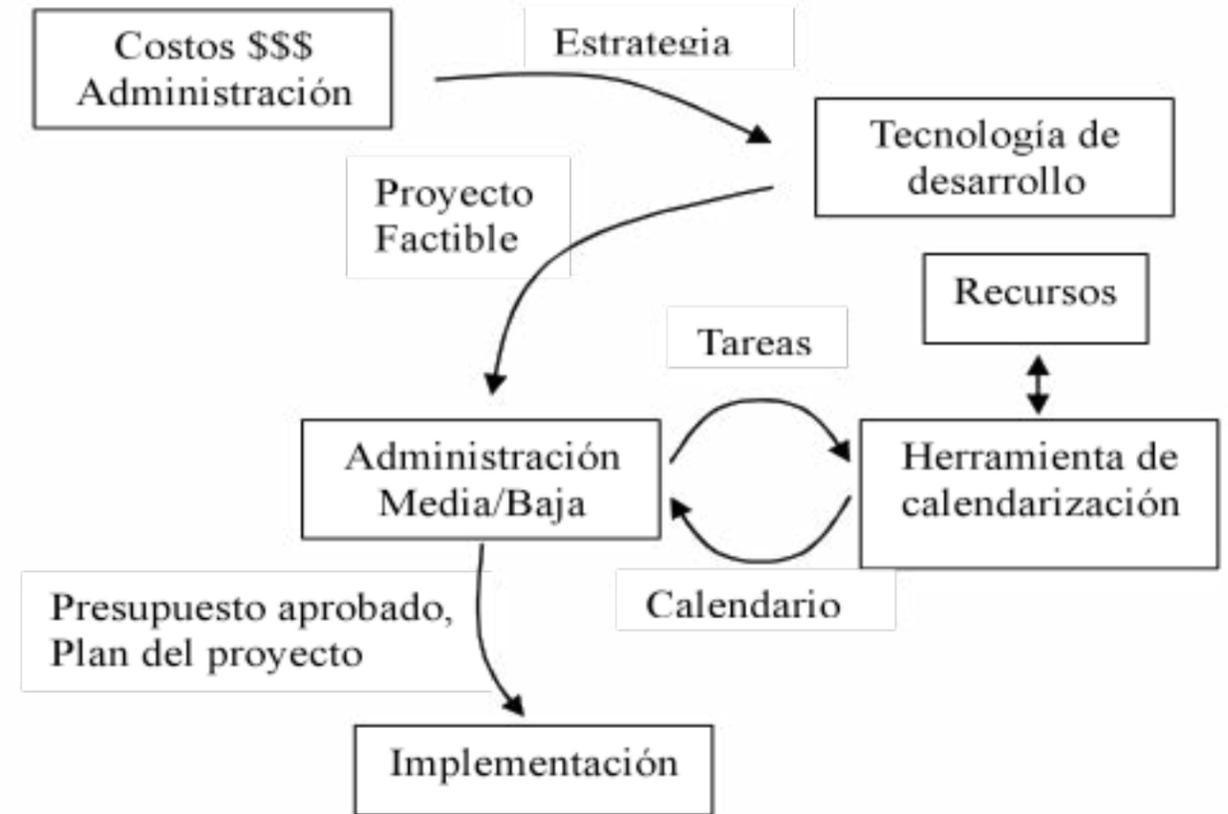


Fig. 3. Proceso de planeación clásico que realizan los administradores.

Para lograr mejorar la planeación se desarrolló un algoritmo que ejecuta procesos en paralelo en una máquina LAM/MPI (LAM, 2006), determinando así la mejor forma de llevar a cabo un proyecto, reduciendo costos y riesgos.

Se propone la optimización basada en un método que consiste en simular procesos en un planificador de tareas (scheduler) en donde se analizan la ejecución de las tareas definidas del proyecto y su impacto en la asignación y liberación de los recursos requeridos para el proyecto. Es importante mencionar que los riesgos que pueden ocurrir al ejecutar las tareas es un factor de impacto en la simulación, ya que puede aumentar o reducir los costos totales del proyecto. El algoritmo busca diferentes escenarios de ejecución del plan basado en la información

inicial del WBS; contienen las tareas a realizar, su interdependencia, los recursos a utilizar y el costo de cada una de ellas. Ver Figura 2.

En su tesis Yusuf Kanik (Kanik, 2005) describe que para la planeación y la programación de actividades en proyectos se han realizado varios estudios desde los años 50. El problema de la programación de un proyecto envuelve básicamente actividades, recursos, relaciones de precedencia, restricciones y objetivos. Con estos elementos es posible controlar la programación de actividades en un proyecto, donde el problema más complejo es organizarlas de tal forma que las restricciones, los recursos, las relaciones y los objetivos se cumplan. A los problemas de asignación de recursos para actividades con restricciones son llamados Resource Constrained Project Scheduling Problem

(RCPSP). En la Figura 4 se muestra la estrategia que se utilizó para construir la herramienta de simulación del WBS y el método de ejecución de procesos.

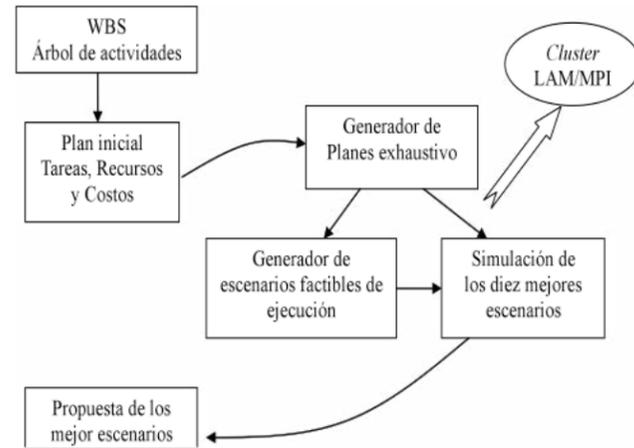


Fig. 4. Propuesta del planificador de tareas y la generación de escenarios para la ejecución en una LAM.

Para construir el generador de escenarios y probar sus propiedades se realizó una especificación en una Red de Petri (PN) (Srivastava, 2000), donde se describe y modela el algoritmo en paralelo propuesto. Esta red fue desarrollada con la herramienta de CPNTools (Colored Petri Net, CPN) (Murata, 1989). En la Figura 5 se muestra el modelo basado en la CPN que permitió analizar propiedades de concurrencia, asignación, ejecución y liberación de recursos de las diferentes tareas definidas en el WBS.

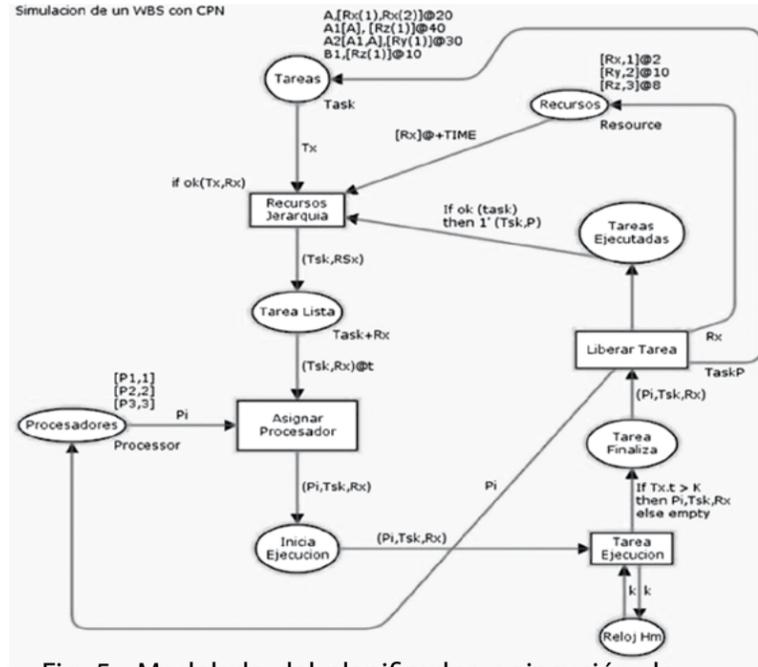


Fig. 5. Modelado del planificador y ejecución de tareas en concurrencia utilizando CPNTools.

Del modelado gráfico anterior de la Figura 4, se obtuvieron los elementos necesarios para la programación del algoritmo en un lenguaje con Programación Orientada a Objetos (POO), en donde las diferentes formas de generar un escenario son analizadas mediante la simulación de procesos, la concurrencia y el paralelismo. En la Figura 6 se muestra el modelo Orientado a Objetos (OO) desarrollado en BlueJ (Barnes & Kölling, 2008), se muestran las clases y los objetos necesarios para el planificador de tareas (scheduler).

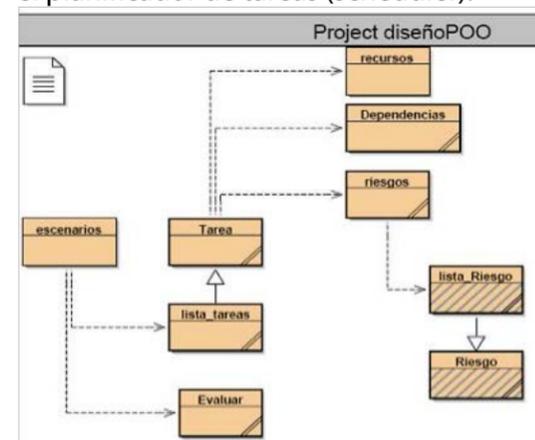


Fig. 6. Modelado de los objetos utilizando POO y la herramienta BlueJ.

El Backus-Naur Form (BNF) que se utilizó para definir los distintos elementos del Modelo se muestra a continuación en la Figura 7.

**Especificación BNF:**

**WBS ::=** <Entregables>

**Entregable x ::=** <Entregables> <Entregable> | <Entregable>

**Entregable ::=** <Tareas> <Dependencias> | <Tareas>

**Tareas ::=** <Tarea s> <Tarea> | <Tarea>

**Tarea ::=** <Identificador> <Tiempo planeado de ejecución> <Costo planeado> <Riesgos> <Recursos necesarios> <Tiempo de Inicio> <Tiempo de Fin>

**Dependencias ::=** <Tareas> <Tipo\_Dependencia> | <Tarea>

**Tipo Dependencia ::=** <Inicio-Inicio> | <Inicio-Fin> | <Fin-Inicio> | <Fin-Fin>

**Riesgo ::=** <Identificador> <Probabilidad Ocurriencia> <Intervalo de Tiempo> <Intervalo de costos>

**Recursos necesarios ::=** <Identificador> <Cantidad>

**Recursos disponibles ::=** <Identificador> <Cantidad>

Fig. 7. Definición de los componentes de la herramienta utilizando una gramática de BNF. Los principales métodos (Kanik, 2005) que se utilizaron para analizar y generar los escenarios mediante el modelo anterior, son:

- Métodos de búsqueda "tabú" (ruta crítica)
- Métodos heurísticos (rutinas seriales y paralelas)

#### 4. DATOS OBTENIDOS

Se utilizó el diseño OO para desarrollar el planificador; se programaron las clases y objetos necesarios, se hicieron pruebas para diferentes proyectos en donde se comprobó el funcionamiento del algoritmo para la generación y evaluación de escenarios; se desarrolló en el lenguaje GNU C++ en una plataforma Linux con la distribución Fedora Core 7. Se desarrolló la aplicación con el lenguaje LAM/MPI, en la Figura 8 se muestra diagrama de ejecución en paralelo.

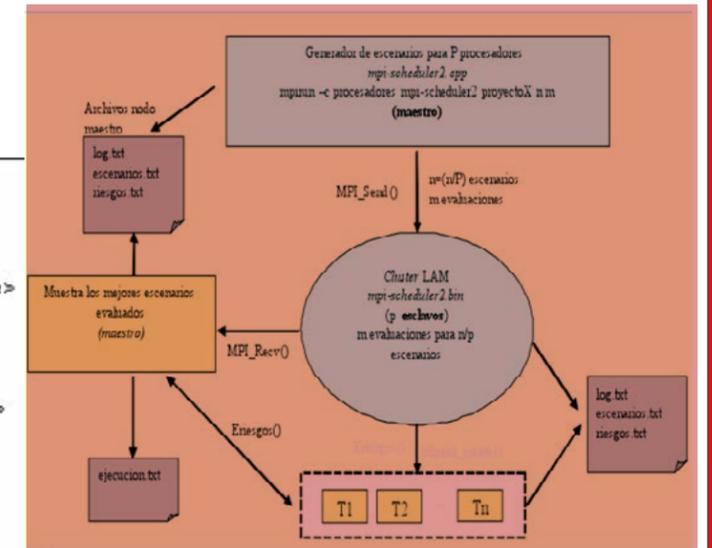


Fig. 8. Diagrama de ejecución de tareas en una máquina LAM/MPI.

El algoritmo propuesto para generar los escenarios se describe a continuación:

- Inicio. Mientras existan tareas sin ejecutar, realizar:
  1. Identificar todas las actividades cuyas restricciones de precedencia permitan su ejecución inmediata y agregarlas a una cola de ejecución.
  2. Si la cola de ejecución no se encuentra vacía, entonces realizar:
    - 2.1. Aplicar una permutación aleatoria a la cola de ejecución
    - 2.2. Obtener una tarea j de la cola de ejecución
      - 2.2.1 Si existen recursos suficientes para ejecutar j, hacerlo
      - 2.2.2 Si no repetir paso 2.2 hasta finalizar la cola
    - 2.3 Repetir desde el paso 1 para un nuevo conjunto de tareas
  3. Si no existen tareas para ejecutar se genera un error "No se puede ejecutar ninguna Tarea".
  4. Generación de resultados

Se construyó el planificador de tareas utilizando hilos (pthreads) y las funciones de ejecución de lenguaje MPI-2 (LAM, 2006). Se obtuvieron las estadísticas de la evaluación de cada uno de los escenarios; se elige el óptimo, el moderado y el más pesimista de cada de una de las simulaciones, se clasifican mediante la fórmula que se muestra a continuación:





$$i=1m (TEjTP + CEjCP)2$$

Clasificación de un escenario en el mejor, moderado y pesimista, para n escenario evaluados m veces.

La selección de los escenarios se hace mediante la simulación de n escenarios (j=1 a n), evaluados m veces (i=1 a m), el criterio se basa en la sumatoria de los tiempos de ejecución de un escenario evaluado (TEj) entre el tiempo del escenario secuencial (TP) y el costo del escenario evaluado (CEj) entre el costo del escenario secuencial (CP). De estas evaluaciones se arroja el diagrama de la Figura 10 siguiente, que muestra la generación de 30 escenarios y 100 evaluaciones del proyecto de telecomunicaciones anteriormente descrito en la Figura 2 anterior.

<p>Actualización de la infraestructura de telecomunicaciones UNAM, Campus Morelia Proyecto2.txt, 23 tareas y 5 entregables, Tiempo secuencial: 87 d, Costo: \$21,700.00</p>
<p><b>Resultado de la ejecución para 2 nodos:</b> <b>Procesador 0:</b> Inicia P0 P0 ejecución de un proyecto: Wed Jul 1 19:59:49 2009 P0 Tareas cargadas: 23 P0 Inicia ejecución aleatoria de Tareas: Escenarios del proyecto: 30 Se envían 15 escenarios al P1:</p>
<p>P0 recibe de P1-&gt; 15 Escenarios evaluados <b>P0 Número de tareas/hilos ejecutados: 35190</b> Escenarios (30) evaluados (100) y clasificados por promedio Moderado P0 Fin programa <b>Procesador 1:</b> Inicia P1 P1 parámetros Riesgos-&gt;n:15 m:100 rank:1 proyecto: proyecto2 P1 Ejecución proyecto: Tue Dec 16 00:16:33 2008 Tareas cargadas: 23 P1 Lectura de valores P1 <b>Ejecución de riesgos</b> para los escenarios de proyecto: proyecto2 Se ejecuta 100 veces cada escenario 15 P1 Escenarios: <b>P1 Tiempo Ejecución: 274 seg, 4 min</b> <b>Termina ejecución (hilos:34500)</b></p>
<p><b>Escenarios Generados (30):</b> <b>Escenario 0:</b> S 75, C 21700 -&gt;T5 I:1 F:2-&gt;T10 I:3 F:4-&gt;T15 I:5 F:5-&gt;T14 I:6 F:7-&gt;T9 I:5 F:14-&gt;T20 I:15 F:19-&gt;T19 I:20 F:21-&gt;T13 I:20 F:22-&gt;T12 I:23 F:25-&gt;T11 I:26 F:27-&gt;T4 I:26 F:28-&gt;T3 I:29 F:30-&gt;T8 I:29 F:33-&gt;T7 I:34 F:38-&gt;T18 I:39 F:44-&gt;T6 I:45 F:49-&gt;T2 I:50 F:54-&gt;T1 I:55 F:59-&gt;T17 I:60 F:62-&gt;T16 I:63 F:67-&gt;T23 I:64 F:68-&gt;T22 I:69 F:71-&gt;T21 I:72 F:74 : <b>Escenario 11:</b> S 77, C 21700 -&gt;T15 I:1 F:1-&gt;T14 I:2 F:3-&gt;T20 I:1 F:5-&gt;T5 I:6 F:7-&gt;T19 I:8 F:9-&gt;T13 I:8 F:10-&gt;T4 I:11 F:13-&gt;T3 I:14 F:15-&gt;T12 I:14 F:16-&gt;T2 I:17 F:21-&gt;T11 I:22 F:23-&gt;T1 I:24 F:28-&gt;T18 I:29 F:34-&gt;T17 I:35 F:37-&gt;T16 I:38 F:42-&gt;T10 I:43 F:44-&gt;T9 I:45 F:54-&gt;T8 I:55 F:59-&gt;T7 I:60 F:64-&gt;T6 I:65 F:69-&gt;T23 I:69 F:71-&gt;T22 I:71 F:73-&gt;T21 I:74 F:76-&gt; : <b>Escenario 29:</b> S 81, C 21700 -&gt;T15 I:1 F:1-&gt;T14 I:2 F:3-&gt;T20 I:1 F:5-&gt;T10 I:6 F:7-&gt;T19 I:8 F:9-&gt;T9 I:8 F:17-&gt;T13 I:18 F:20-&gt;T18 I:21 F:26-&gt;T12 I:27 F:29-&gt;T11 I:30 F:31-&gt;T8 I:30 F:34-&gt;T7 I:35 F:39-&gt;T6 I:40 F:44-&gt;T17 I:45 F:47-&gt;T16 I:48 F:52-&gt;T5 I:53 F:54-&gt;T4 I:55 F:57-&gt;T3 I:58 F:59-&gt;T2 I:60 F:64-&gt;T1 I:65 F:69-&gt;T23 I:70 F:74-&gt;T22 I:75 F:77-&gt;T21 I:78 F:80-&gt;</p>
<p><b>No. de evaluaciones (100), Mejores 10 escenarios:</b> <b>Escenario 11-&gt; Opt:0.798282, Mod:1.0155, Pes:1.292, Var:0.0250559, DesvEst: 0.15829</b> Escenario 1-&gt; <b>Opt:0.697738</b>, Mod:1.01613, Pes:1.39233, Var:0.039462, DesvEst: 0.19865 Escenario 6-&gt; Opt:0.82279, Mod:1.02807, Pes:1.32007, Var:0.0276725, DesvEst: <b>0.16635</b> Escenario 13-&gt; Opt:0.737903, Mod:1.03559, Pes:1.3862, Var:0.0364444, DesvEst: 0.19090 Escenario 7-&gt; Opt:0.809192, Mod:1.04278, Pes:1.43075, Var:0.0320508, DesvEst: 0.17902 Escenario 14-&gt; Opt:0.735933, Mod:1.04299, Pes:1.40262, Var:0.0327625, DesvEst:0.18100 Escenario 4-&gt; Opt:0.812526, Mod:1.04964, Pes:1.42396, Var:0.0379817, DesvEst: 0.19488 Escenario 8-&gt; Opt:0.700781, Mod:1.05133, Pes:1.40489, Var:0.0428308, DesvEst: 0.20695 Escenario 3-&gt; Opt:0.776498, Mod:1.05253, Pes:1.45161, Var:0.038708, DesvEst: 0.196743 Escenario 5-&gt; Opt:0.784402, Mod:1.05507, Pes:1.30272, Var:0.0292746, DesvEst: 0.17109</p>

Figura 10. Generación de 30 escenarios y 100 evaluaciones del proyecto.



Como se observa en la figura anterior el mejor escenario evaluado (moderado) fue el número 11 arrojando un tiempo de 77 días de ejecución y en promedio un costo de \$21,700.00, es importante señalar que esta evaluación se realizó considerando los riesgos que pueden ocurrir en ciertas tareas (que incrementan o reducen el tiempo y el costo de ejecución), la desviación estándar del escenario con respecto al mejor escenario secuencial fue de 0.15829. El diagrama de Gantt del escenario 11 puede observarse en la Figura 11 siguiente.

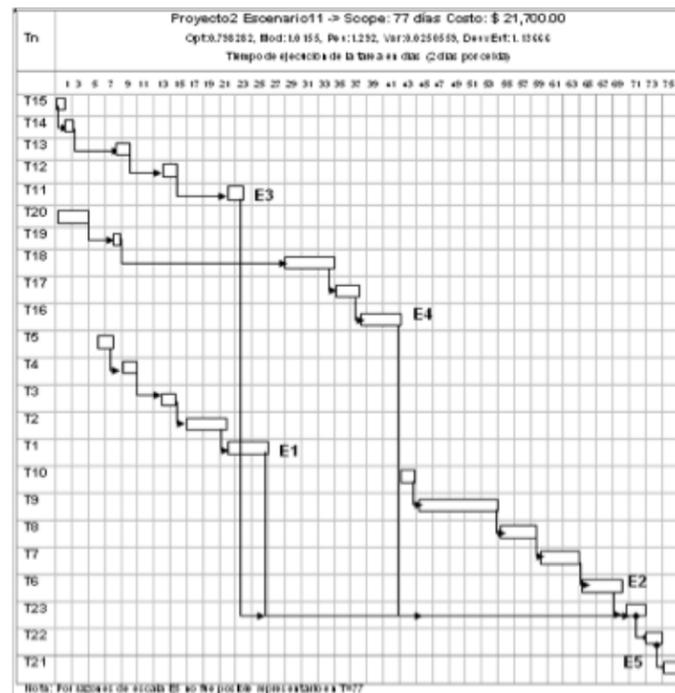


Fig. 11. Mejor escenario para un proyecto con 23 tareas y n=30 escenarios m=100 evaluaciones, ejecutado en 2 procesadores.

## 5. RESULTADOS

Al evaluar el proyecto se observó que el tiempo de ejecución es más eficiente en P procesadores (donde  $P \geq 1$ ), esta tendencia puede observarse en la siguiente Figura 12.

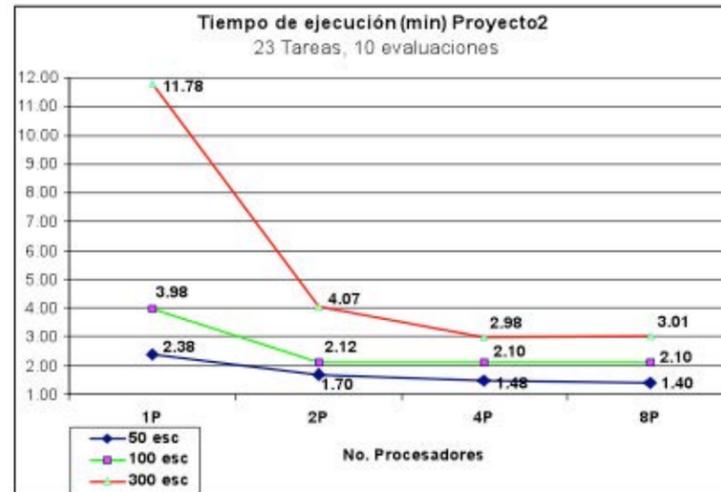
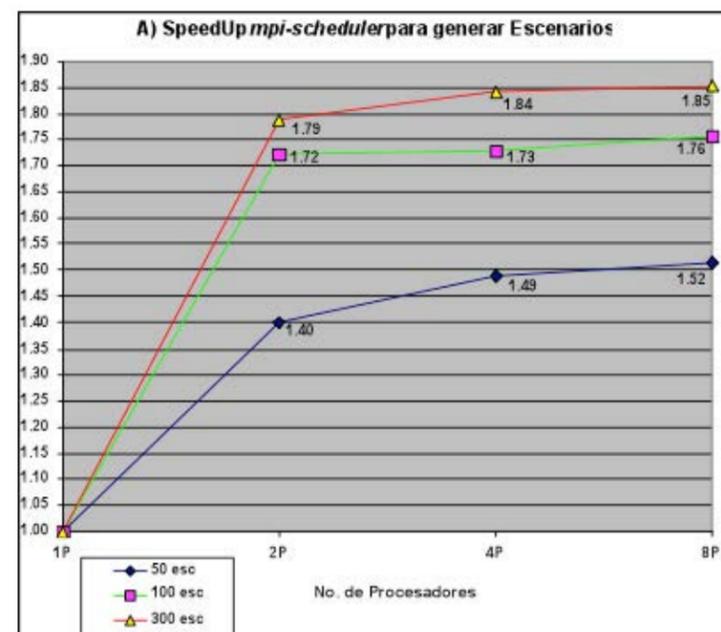


Fig. 12. Tiempo de ejecución para P procesadores en la máquina LAM/MPI.

En la Figura 13 se muestra la eficiencia del algoritmo basado en el speedup (aceleración) del conjunto de computadoras (Clúster) de acuerdo a la fórmula siguiente y Figura 13:

$$SpeedUp = \frac{TiempoSecuencial}{TiempoParalelo}$$



## 6. DISCUSIÓN Y CONCLUSIONES

El algoritmo programado y desarrollado son las herramientas prog-scheduler2 y mpi-scheduler2 que se basa en la generación de escenarios (factibles de ejecución) utilizando un método aleatorio; que consiste en identificar aquellas tareas que pueden llevarse a cabo (dependencias y recursos) y ejecutarlas (simulación), obteniendo así un escenario factible. Para darle confianza a los escenarios generados el algoritmo utiliza los riesgos para simular su ocurrencia y así analizar los diferentes escenarios posibles de un proyecto; el optimista (pocos riesgos), el moderado (con riesgos moderados en tiempo y costo) y el pesimista (donde ocurren mayores riesgos). Se obtiene la varianza y la desviación estándar como estadísticos de esta simulación.

El enfoque de la simulación permitió validar los escenarios utilizando la ejecución exhaustiva hasta estabilizar el modelo y poder así ofrecer un escenario confiable. Se elaboró un modelo basado en un método formal conocido como CPN, que permitió diseñar en forma gráfica las diferentes partes en las que consiste el algoritmo y prever con esto las posibles fallas en el modelo programado, ayudó también a entender el problema y visualizar como podría desarrollarse en un lenguaje con POO. Es importante señalar que este método de diseño gráfico es una alternativa y existen varios más. La herramienta programada en un lenguaje con POO facilitó el desarrollo, puesto que los objetos son programados con encapsulación y polimorfismo, esto permitió concentrarse en resolver el problema de la generación de escenarios y la evaluación de riesgos. El funcionamiento fue probado con varios proyectos y en efecto se resuelve el problema de planeación utilizando la simulación de tareas, sus estados, los recursos y los riesgos que implican dichos proyectos. El objeto tiempo diseñado como un objeto persistente permitió ejecutar el proyecto con un tiempo ficticio apegado al reloj del sistema. Para el programa mpi-scheduler2.cpp el reloj maestro lo controla el procesador central o nodo maestro, las tareas ejecutadas en los nodos esclavos solo simulan los tic (avance

del reloj) que requieren para terminar la tarea. El programa se enfoca en la utilización de recursos renovables y no considera el uso de recursos no-renovables que bien podría ser una extensión o mejora en el futuro. El desarrollo en LAM/MPI permitió observar la eficiencia de la herramienta para grandes proyectos, donde las miles de tareas podrían llegar a tardar varios minutos e incluso horas para generar y evaluar los escenarios. El uso del lenguaje MPI y la máquina LAM no afectaron el desempeño de la herramienta, el número de mensajes intercambiado entre el nodo maestro y esclavo es en realidad muy pequeño, para un proyecto con n escenarios al ejecutarse en un Clúster el número de mensajes puede calcularse por la fórmula siguiente (Cálculo del número de mensajes intercambiados entre nodos):

$$\left(\frac{n}{P} * NT + 4\right) * 2 * (P - 1)$$

En donde NT se define como el número de Tareas del proyecto y P el número de procesadores utilizados en el Cluster, donde n/P representa el bloque de información o los escenarios que le tocan a cada uno de los procesadores. La herramienta desarrollada ayuda en gran medida a los responsables del proyecto a tomar decisiones que toman en cuenta los riesgos. Es posible construir otra simulación de planes utilizando métodos alternativos como:

- Algoritmos genéticos
- Redes neuronales
- Programación dinámica
- Métodos de simulación alternos



## 7. REFERENCIAS

- [1] Barnes D. & Kölling M. (2008). Objects First with Java, A Practical Introduction Using BlueJ. Fourth Edition.
- [2] Espejel M. & Ferreira H. (2005). Proyecto de actualización de la infraestructura de telecomunicaciones en la UNAM campus Morelia, UNAM. México.
- [3] Holzmann G. (1991). "Design and validation of computer protocols", Bell Laboratories, Murray Hill New Jersey. PRENTICE-HALL.
- [4] Holzmann G (1997). The Model Checker SPIN, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 23, NO.
- [5] Institute, P. (2009). A Guide to the Project Management Body of Knowledge (PMBOK® Guide). Fourth Edition, Official Spanish Translation, Paperback. Project Management Institute.
- [6] Kanik Y. (2005). "Software project scheduling, software project performance measurement and control". Thesis, School of Informatics of the Middle East Technical University.
- [7] LAM, M. (2006). LAM/MPI User's Guide. Version 7.1.2. The LAM/MPI Team, Open Systems Lab. Disponible en: <http://www.lam-mpi.org/>.
- [8] Murata T. (1989). "Petri Nets: Properties, Analysis and Applications", Article invited, PROCEEDINGS OF THE IEEE, VOL. 77, NO. 4.
- [9] Srivastava B. (2000). "Efficient planning by effective resource reasoning". Arizona State University.

## PROTOTIPO DE ASPA PARA UN AEROGENERADOR TIPO VERTICAL.

Ing. María Izbeth Lara Lavallo  
Ing. Manuel Bello Hernández

PROTOTIPO DE ASPA  
PROTOTIPO DE ASPA PROTOTIPO DE  
ASPA PROTOTIPO DE ASPA PROTOTIPO  
DE ASPA PROTOTIPO DE  
ASPA PROTOTIPO DE ASPA ASPA  
PROTOTIPO DE ASPA PROTOTIPO DE ASPA