

"UNIVERSIDAD DE INGENIERIA & TECNOLOGIA"

NOMBRE: Lucero Anggela

APELLIDO: Yauri Gonzales

PROFESOR: Luis Peña

CURSO: Investigación de Operaciones I

TEMA: Gestión de Proyectos

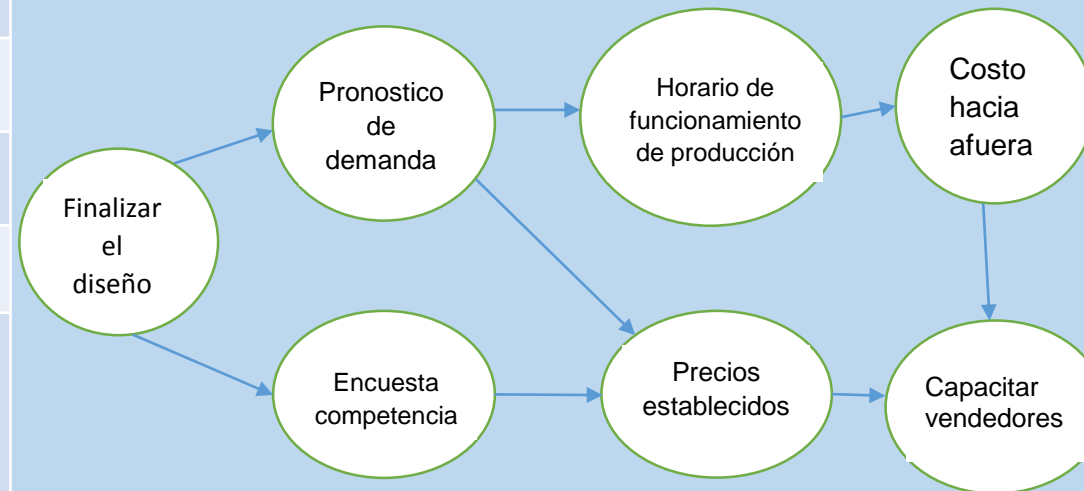
2015-Perú



PERT (Program Evaluation and Review technique)

Widgets de inalámbricos está a punto de lanzar un nuevo producto: el Widget de Solar. Para garantizar que el lanzamiento se producirá en el tiempo, WW quiere realizar un análisis PERT las tareas previas a la puesta en marcha. Hacerlo les permitirá identificar la ruta crítica de las tareas que deben completarse en tiempo para garantizar la introducción oportuna de Widget Solar. Las tareas que deben realizarse antes de la introducción y su esperado tiempo de terminación se enumeran en la siguiente tabla:

Tarea	Semanas
Finalizar el diseño	10
Previsión de demanda	14
Encuesta competencia	3
Precios establecidos	3
Horario funcionamiento de producción	7
Costo hacia fuera	4
Capacitar vendedores	10



Relaciones de precedencia de lanzamiento de producto

***GESTION
DE
PROYECTO***

❑ Código en el Software Lingo

MODEL:

TASKS / DESIGN, FORECAST, SURVEY, PRICE, SCHEDULE, COSTOUT, TRAIN/: TIME, ES, LS, SLACK;

PRED(TASKS, TASKS) /

DESIGN,FORECAST,

DESIGN,SURVEY,

FORECAST,PRICE,

FORECAST,SCHEDULE,

SURVEY,PRICE,

SCHEDULE,COSTOUT,

PRICE,TRAIN,

COSTOUT,TRAIN /;

ENDSETS

DATA:

TIME = 10, 14, 3, 3, 7, 4, 10;

ENDDATA

@FOR(TASKS(J)| J #GT# 1:

ES(J) = @MAX(PRED(I, J): ES(I) + TIME(I)));

@FOR(TASKS(I)| I #LT# LTASK:

LS(I) = @MIN(PRED(I, J): ES(J) - TIME(I));

@FOR(TASKS(I): SLACK(I) = LS(I) - ES(I));

ES(1) = 0;

LTASK = @SIZE(TASKS);

LS(LTASK) = ES(LTASK);

DATA:

@TEXT() = @TABLE(PRED);

End data

END

} Restricciones

***GESTION
DE
PROYECTO***

❑ Solución y formulación del código:



```
MODEL:
SETS: |
    TASKS / DESIGN, FORECAST, SURVEY, PRICE, SCHEDULE, COSTOUT,
    TRAIN/: TIME, ES, LS, SLACK;
```

- **TASK:** Son las tareas que deben realizarse antes de ejecutar el proyecto.
- **TRAIN:**
- **TIME.-** Tiempo para completar la tarea
- **ES.-** Hora de inicio mas temprana posible para la tarea
- **LS.-** Hora de inicio mas tarde posible para la tarea
- **SLACK.-** Diferencia entre ES y LS para la tarea

```
DATA:
    TIME = 10, 14, 3, 3, 7, 4, 10;
ENDDATA
```

- **TIME:** Es el determinado tiempo de terminación del proyecto.

```
PRED( TASKS, TASKS) /
```

- **PRED:** Precedencia, son tareas que deben de realizarse antes que otras

***GESTION
DE
PROYECTO***

❑ RESTRICCIONES:

```
!RESTRICCIONES:  
1. Restricción de tiempo max. de tareas, ;  
@FOR( TASKS( J) | J #GT# 1:  
ES( J) = @MAX( PRED( I, J): ES( I) + TIME( I)));
```

- Para cada tarea (J) mayor que 1, el tiempo de inicio más temprano de la tarea (J) será igual al máximo tiempo en el par de precedencia de (I, J) al tiempo de inicio más temprano de (I) + tiempo de duración de la tarea (I).

```
!2. Restricción de tiempo min. de tareas;  
@FOR( TASKS( I) | I #LT# LTASK:  
LS( I) = @MIN( PRED( I, J): ES( J) - TIME( I) );  
);
```

- Para cada tarea (I), que sea menor que la última tarea, el tiempo de inicio más tarde de la tarea (I) es igual al mínimo tiempo en el par de la precedencia (I, J) al tiempo de inicio más temprano de (J) – tiempo de duración de la tarea (I)

```
!3. Restricción de holgura;  
@FOR( TASKS( I): SLACK( I) = LS( I) - ES( I));  
ES( 1) = 0;  
LTASK = @SIZE( TASKS);  
LS( LTASK) = ES( LTASK);
```

- Para cada tarea (I), la holgura de dicha tarea será igual al tiempo de inicio mas tarde de la tarea (I) – el tiempo de inicio mas temprano de la tarea (I).
- El tiempo de inicio mas temprano de la tarea 1 será igual a 0.