Herramienta de Revenue Assurance

Proguide SRL - Tecpetrol SA

Tabla de contenidos

[Resumen 3](#_Toc477182883)

[Lectura de archivos 4](#_Toc477182884)

[Planificación de servicios 4](#_Toc477182885)

[Objetivo 4](#_Toc477182886)

[Generación de stack de tareas por servicio. 4](#_Toc477182887)

[Generación del registro de cuadrillas. 7](#_Toc477182888)

[Asignación de cuadrillas a las tareas. 7](#_Toc477182889)

Resumen

El siguiente documento detalla las funcionalidades de la herramienta desarrollada en R para la planificación estratégica de la operación del yacimiento de “el tordillo”.

# Lectura de archivos

Se escribe una función “read\_task\_file” que toma como input el path del directorio y lee el archivo “2.1 Matriz de tareas por servicio v05.xlsx” y devuelve una lista llamada “task\_list” con nombres “Tareas RSM”, “Tareas L”, “Backup Logistica” correspondientes a los 3 dataset del archivo con los mismos nombres de campos.

Corregir datos faltantes para “Tareas RSM”: Tipo = “otro”, “PrioridadDesc” = “Baja”, columnas K-X =0.

Corregir datos faltantes para “Tareas L”: TBD

# Planificación de servicios

## Objetivo

Se necesita estimar el dimensionamiento de la cantidad de cuadrillas de cada servicio mediante la generación de tareas previstas y la simulación de asignación de recursos a dichas tareas y la medición de la calidad de servicio en términos de cumplimiento de dichas tareas. Se considera un año de simulación.

## **Generación de stack de tareas por servicio.**

Se escribe una función “generate\_task\_stacks” que toma como input:

* “task\_list\_RSM” = task\_list[[‘Tareas RSM’]].
* “act\_level” = lista con nombres de acuerdo al siguiente detalle, tomar valores de prueba de acuerdo a la indicación:
  + ‘PozosBM’ = 361
  + ‘PozosBM\_A’=113
  + ‘PozosBM\_B’=91
  + ‘PozosBM\_CDE’=194
  + ‘PozosBM\_X’=8
  + ‘PozosBES’ = 82
  + PozosBES\_A=14
  + PozosBES\_B=10
  + PozosBES\_CDE=55
  + PozosBES\_X=3
  + PozoIny=106
  + Bateria=37
  + Satelite=28
  + PlantaPetroleo=2
  + PlantaGas=1
  + PlantaAgua=3
  + Subestacion=37
  + Gasoducto=37
  + Motogen=4
  + Oleoducto=1
  + Colector=42
  + Oficina=1

El output de la función es una lista de datasets para cada servicio con el total de las tareas generadas de acuerdo al nivel de actividad. El output se llama stack\_list y tiene como nombres los servicios: “Recorredores”, “Supervisores”, etc. tomados como los valores (únicos) del campo Servicio del dataset “task\_list\_RSM”. Cada tipo de tarea del “task\_list\_RSM” con el campo Activa = “Si” se agrega en el stack (el output) *n* veces con

*n*= FrecuenciaAnualxInstalacion \* cantidadInstalaciones \* tasa de cada instalación.

Por ejemplo, se generan un total de 5054 de la tarea tipo “Acondiciona boca de pozo” ya que 5054 = 14\*361\*1.

El output es una lista con datasets generados por la función (un dataset para cada servicio) con los siguientes campos con su correspondiente valor o detalle de cálculo.

* task\_id. Identificador univoco entero.
* task. Copiado del “task\_list\_RSM”.
* task\_time. Copiado del TiempoEstimadoxTarea(hs) de “task\_list\_RSM”.
* priority. Copiado del “task\_list\_RSM”.
* waste. Copiado del MermaProduccion de “task\_list\_RSM”.
* location = NA. TBD
* transport\_time = 0. TBD
* total\_time = tiempo\_tarea + tiempo\_transporte
* day = para cada tarea se genera un número random entero entre 1 y 365.
* completed = FALSE
* start\_day = NA. TBD
* cuad = NA
* end\_day = NA. TBD
* reprog = NA. TBD

Nota: los campos Tipo, PrioridadDesc, Lugar del objeto “task\_list\_RSM” no son tenidos en cuenta.

## Generación del registro de cuadrillas.

Se escribe una función generate\_cuad\_record que toma como input el dia (número entero 1:365), y num\_cuad: la cantidad de cuadrillas. Devuelve como output un dataset llamado cuad\_record con los siguientes campos.

* cuad\_id: identificador entero de 1 a num\_cuad (total de filas del dataset = num\_cuad)
* day\_time2free= day + 8/24
* lunch=FALSE
* end\_day=FALSE
* ot=FALSE

## Asignación de cuadrillas a las tareas.

Objetivo

Se necesita simular la asignación de *n* cuadrillas al stack de tareas a realizar de cada servicio. Para tal fin se utilizan una función de asignación principal y varias funciones intermedias.

Se escribe una función que toma como input:

* el stack de tareas de un servicio: Por ejemplo, task\_stack = stack\_list[[‘Recorredores’]].
* num\_cuad: la cantidad de cuadrillas a asignar.
* turnos= (1, 2 ,3 ,4) con el siguiente detalle:

1. 8 horas días corridos.
2. 8 horas días hábiles
3. 12 horas días corridos
4. 12 horas días hábiles

* over\_time = FALSE (si se permite que una cuadrilla haga horas extras)
* save\_output = TRUE (se graba el output en un archivo)

y devuelve un registro de tareas asignadas a las cuadrillas (assignment\_record) con los siguientes campos:

* cuad\_id: el número de cuadrilla que se asignó a la tarea.
* task\_id: id de la tarea asignada.
* start\_day: tiempo de inicio de la realización de la tarea
* end\_day: tiempo de finalización de la tarea asignada
* ot: cantidad de horas extras trabajadas.

Para la asignación es necesario incorporar la siguiente lógica.

De acuerdo a si los turnos son días hábiles o días corridos, se comienza en el dia 1 al 365 o siguiendo la siguiente serie: 1, 2, 3, 4, 5, 8, 9, 10, 11, 12, …, 358, 359, 360, 361, 362, 365, para los días corridos.

assignment\_record = empty\_dataset

day=1

while day<366 do

cuad\_record = generate\_cuad\_record(num\_cuad, day)

daily\_assigment(cuad\_record, task\_stack, turnos)

day = day +1

end while

La función daily\_assigmnet toma como input: cuad\_record, task\_stack, turnos y como output:

* escribe una fila en el dataset assignment\_record
* modifica registros correspondientes a task\_stack
* modifica registros de cuad\_record

De acuerdo al siguiente diagrama: