



TECNOLOGICO NACIONAL DE MÉXICO



**TECNOLOGICO
NACIONAL DE MÉXICO**

INSTITUTO TECNOLÓGICO DE CIUDAD

MADERO



"POR MI PATRIA Y POR MI BIEN"

Proyecto de Fin de Curso:

Parte #1:

Problema de Selección de Trabajos

Alumno: Anastasio Salas Juan Carlos

Num. de Control:

21070315

Materia:

Análisis y Diseño de Algoritmos

Horario:

12:00 – 13:00 hrs.

Análisis y Diseño de Algoritmos

Proyecto de Fin de Curso

Parte #1:

Instrucciones:

En una fábrica se producen diversos productos cada uno con una utilidad o ganancia específica. La producción deseada de cada día se ingresa en una base de datos en una tupla: (id de tarea, duración en minutos, utilidad, programada). El problema consiste en seleccionar la combinación de tareas que en una jornada de 8 horas puedan llevarse a cabo obteniendo el mayor beneficio. Las tareas a tomar en cuenta son todas las que su variable programada tenga un valor falso. La salida es la lista de tareas que se programarán, el tiempo total de ejecución en horas, la ganancia esperada y como efecto colateral el cambio de estatus en la entrada correspondiente de la base de datos de no programada a programada.

Información a Considerar

Base de Datos

El proyecto requiere el uso mandatorio de un gestor y servicio de base de datos. Para los propósitos del mismo se optó por utilizar MySQL como servicio y MySQL Workbench como Gestor de Base de Datos.

Tablas

A continuación se muestra la estructura de la tabla que se utilizará para este proyecto:

```
• drop schema if exists jobs;
• create schema jobs;
• use jobs;

• create table tareas(
    id_tarea int auto_increment primary key,
    duracion int not null,
    utilidad double not null,
    programado bit default 0 not null
);
```

Además para facilitar la ejecución de Query's se optó por la creación y uso posterior de procedimientos almacenados (PA) como se muestra a continuación:



```
25 • DROP PROCEDURE IF EXISTS newTask;
26 delimiter //
27 • create procedure newTask(duracion1 int, utilidad1 int)
28 begin
29
30 declare exit handler for sqlexception
31 begin
32 show errors limit 1;
33 rollback;
34 end;
35
36 declare exit handler for sqlwarning
37 begin
38 show warnings limit 1;
39 rollback;
40 end;
41
42 start transaction;
43 insert into tareas(duracion,utilidad,programado) value (duracion1,utilidad1,0);
44 commit;
45 end //
46 delimiter ;
```

```
delimiter //
create procedure finishTask(id_task int)
begin
declare exit handler for sqlexception
begin
show errors limit 1;
rollback;
end;
declare exit handler for sqlwarning
begin
show warnings limit 1;
rollback;
end;
start transaction;
update tareas set programado = 1 where id_tarea = id_task;
commit;
end //
delimiter ;

DROP PROCEDURE IF EXISTS getPendientes;
delimiter //
```

Conexión.

Para poder trabajar con la base de datos MySQL el proyecto requiere de la implementación de la librería de oracle jdbc:connector. Este se agrega y se implementa a las dependencias del build.gradle:

```
dependencies { this: DependencyHandlerScope
    // Note, if you develop a library, you should use compose.desktop.common.
    // compose.desktop.currentOs should be used in launcher-sourceSet
    // (in a separate module for demo project and in testMain).
    // With compose.desktop.common you will also lose @Preview functionality
    implementation(compose.desktop.currentOs)
    implementation("org.jetbrains.androidx.lifecycle:lifecycle-viewmodel-compose:2.8.0")
    implementation("mysql:mysql-connector-java:8.0.30")
}
```

ADVERTENCIA: Deberá de cambiar las credenciales de acceso a la base de datos por unas propias para ejecutar el programa correctamente.

```
class DBConnection {
    private val URL="jdbc:mysql://localhost:3306/jobs"
    private val USER="root"
    private val PASSWORD = "2015anastasio"
```

Validaciones

- A continuación se presentan algunos puntos a tomar en cuenta con respecto a las limitaciones del programa:
- El programa no ejecutará ninguna operación si no hay datos de entrada.
- El programa en su módulo de registro de tareas está validado contra campos vacíos o con formatos inadecuados.
- **ADVERTENCIA:** El programa no implementa contramedidas para las inyecciones de SQL.
- La interfaz no es confiable para tareas en altas cantidades, si bien se pudo implementar un ScrollState, debido a limitaciones del resto del diseño que se tuvo que dejar fijo la altura del contenedor. Un número grande de tareas corre el riesgo de ocultar el resto de la interfaz.
- La duración de las tareas dentro del programa se miden en MINUTOS ENTEROS, es decir no se aceptan duraciones con parte fraccionaria.
- La ganancia o utilidad SI acepta valores con unidades fraccionarias

Confirmación de Valores

Con el propósito de comprobar la eficacia y rectitud del algoritmo implementado se optó por reducir el tiempo disponible a 5 min. Por tanto, antes de usar el programa verificar que el tiempo límite administrado por la variable "limit" esté correctamente declarado como 480:

```
var totalUtility = mutableStateOf( value: "")  
var notificaciones = mutableStateOf( value: "")  
val limit = 5
```