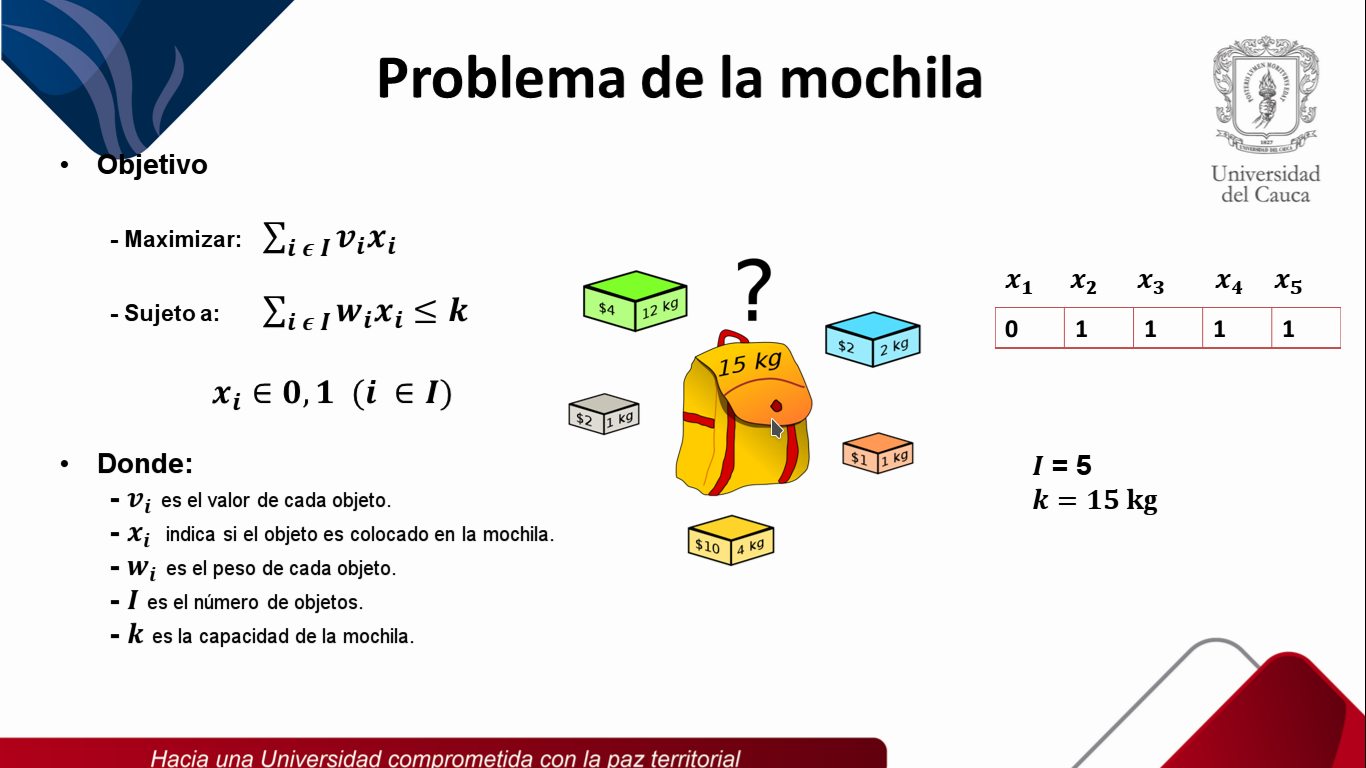
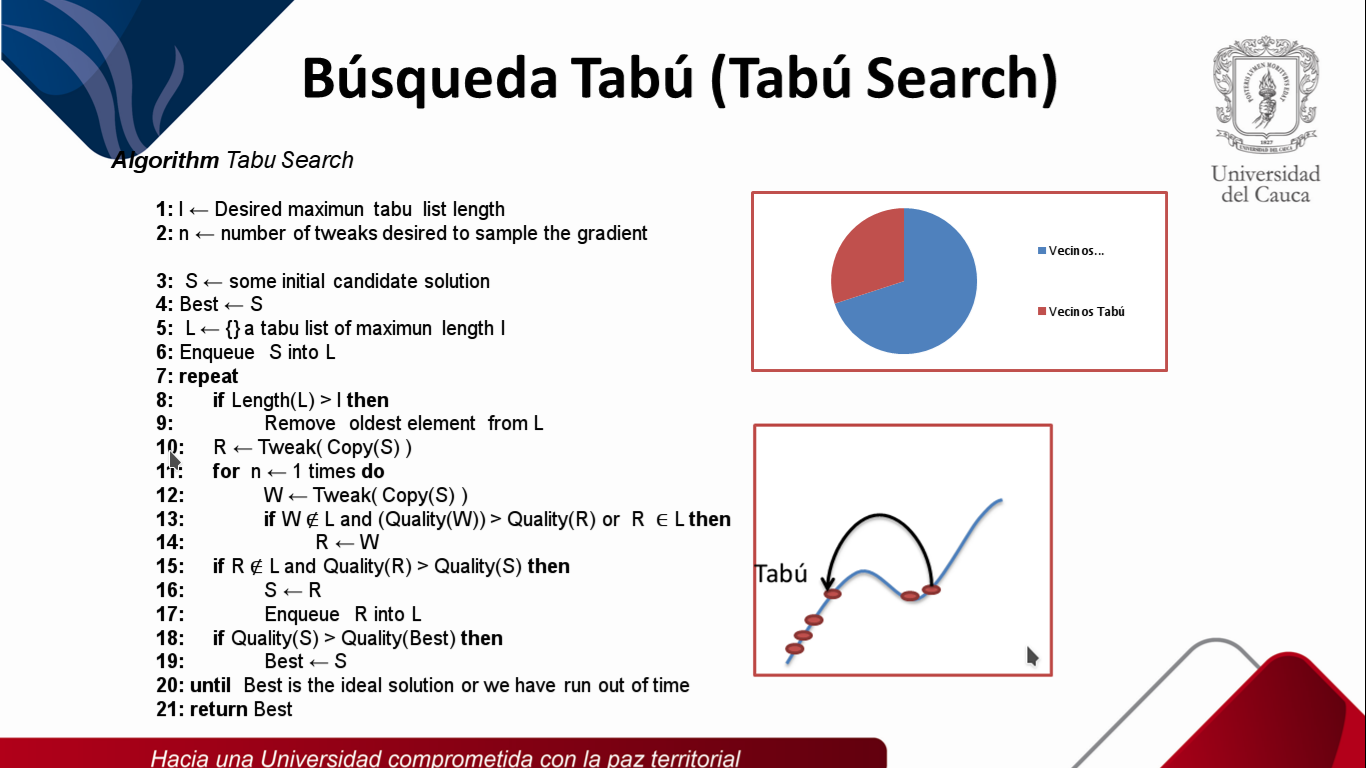
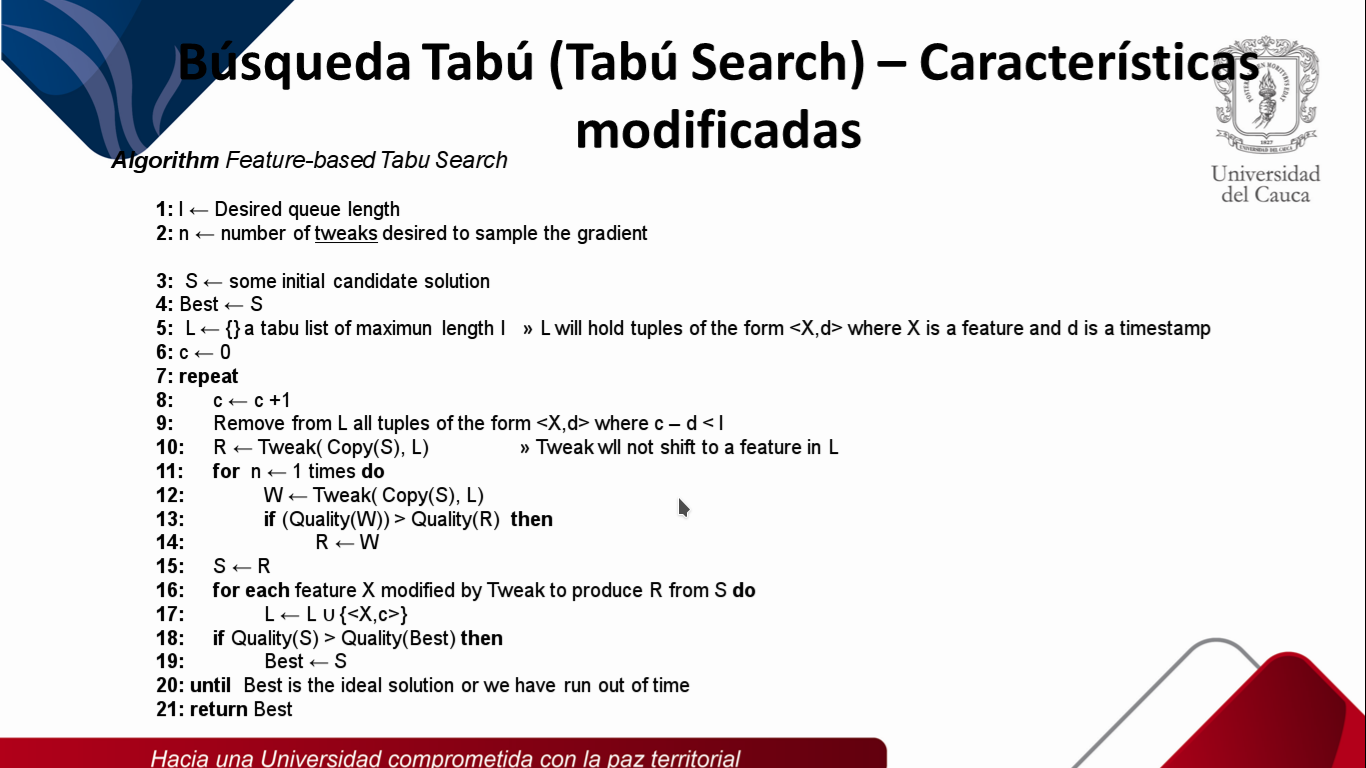
**Problema de la Mochila – Búsqueda Tabú**

****

**Algoritmos base**

****

****

**Análisis**

Se implementó el ***Algorithm Tabu Search***y una variación de este como lo es el ***Algorithm Feature-based Tabu Search.***

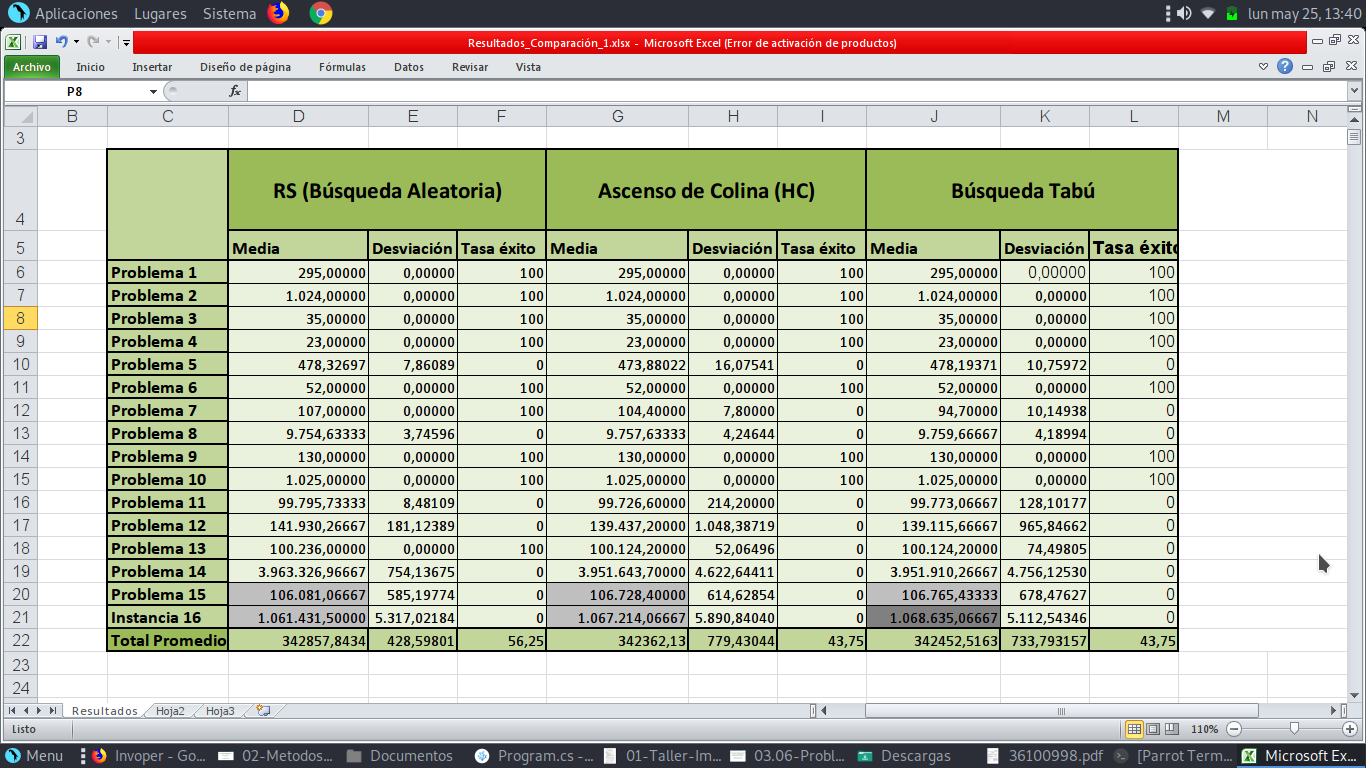
**Parámetros:** Los parámetros de los algoritmos se escogen después de hacer varias pruebas manuales. No se automatizó por el tiempo que están demorando los algoritmos con los últimos problemas, que tienen demasiados datos.

Para la longitud de la lista tabú se decide escoger el número de items o objetos que están disponibles para empezar a resolver el problema. En el ***Algorithm Feature-based Tabu Search*** la lista tabú siempre tendrá este tamaño como máximo, ya que trabaja con las características del problema que en este caso son las representaciones de los objetos. Para hacer una comparación equitativa de los algoritmos se decide escoger el mismo tamaño de la lista tabú.

En cuanto al número de Tweaks n se le asigna 2, ya que al probar con un número mayor, los algoritmos se alejaban del óptimo conocido. Lo anterior se puede dar por que solo se está utilizando un número de evaluaciones de 1000.

**Comparaciones:** Además de los dos algoritmos mencionados anteriormente, también se resuelve el problema con el algoritmo de **Búsqueda Aleatoria** y el algoritmo de **Ascenso a la colina (HC)**, que presentaron un mejor rendimiento en la mayoría de los resultados obtenidos.

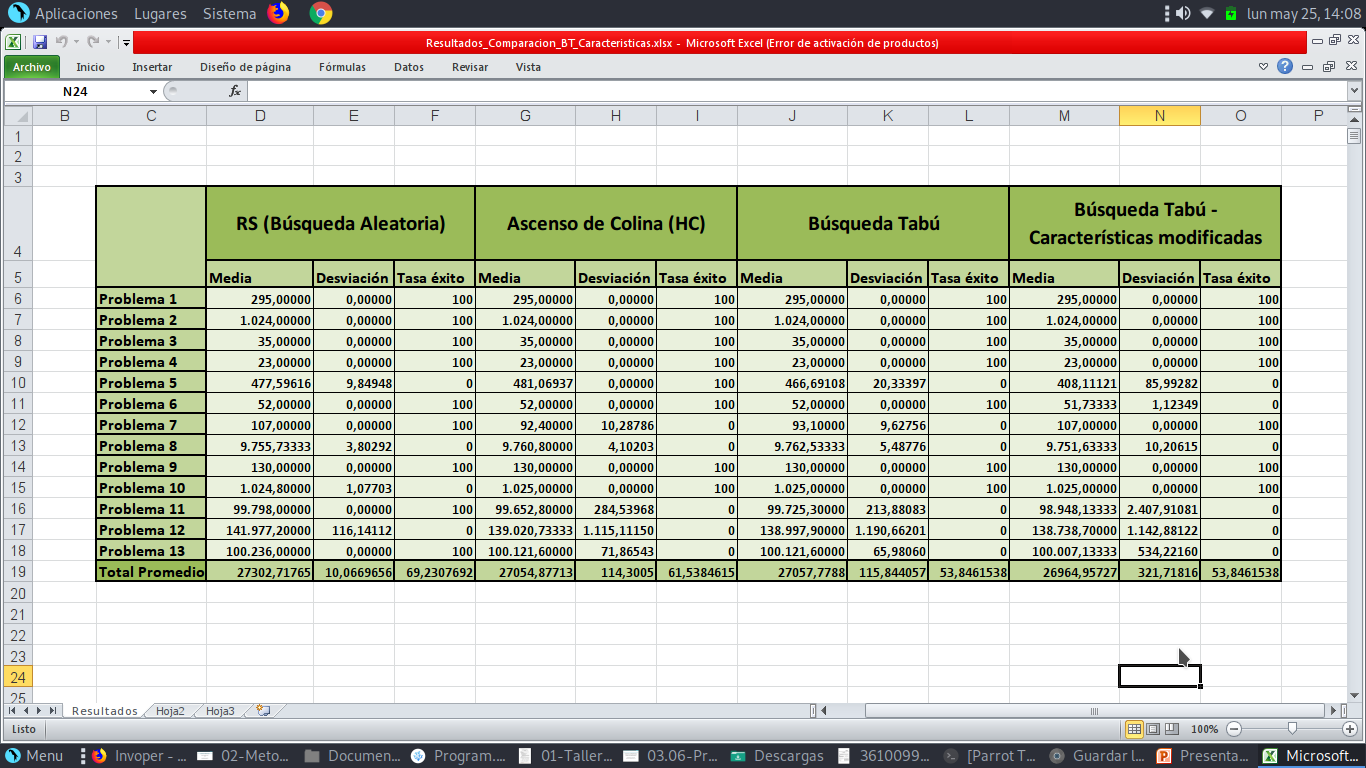
1. Al solucionar los 16 problemas de la mochila con los algoritmos: **Búsqueda aleatoria, HC y Búsqueda Tabú** se obtuvo que:

**Tabla 1 – Búsqueda Tabú**

En la **Tabla 1** se observa que la **RS** tiene los mejores datos, con una **Media** **promedio** y una **Tasa de éxito** más alta que los demás y con una **Desviación estándar** menor.

La **Búsqueda Tabú**  supera a HC por muy poco, presentando la misma **Tasa de éxito.**

Al mirar los resultados, observamos que en los últimos 2 problemas la **RS** pierde con los otros dos algoritmos, y en el último problema; la **Búsqueda Tabú**  es la de mejor resultado. En estos últimos dos resultados por el tamaño de los problemas se empieza a notar que la **RS** pierde rendimiento a medida que aumenta la complejidad del problema, Y que la **Búsqueda Tabú**  empieza a ser mejor en problemas más complejos.

1. **Búsqueda Tabú con características:**

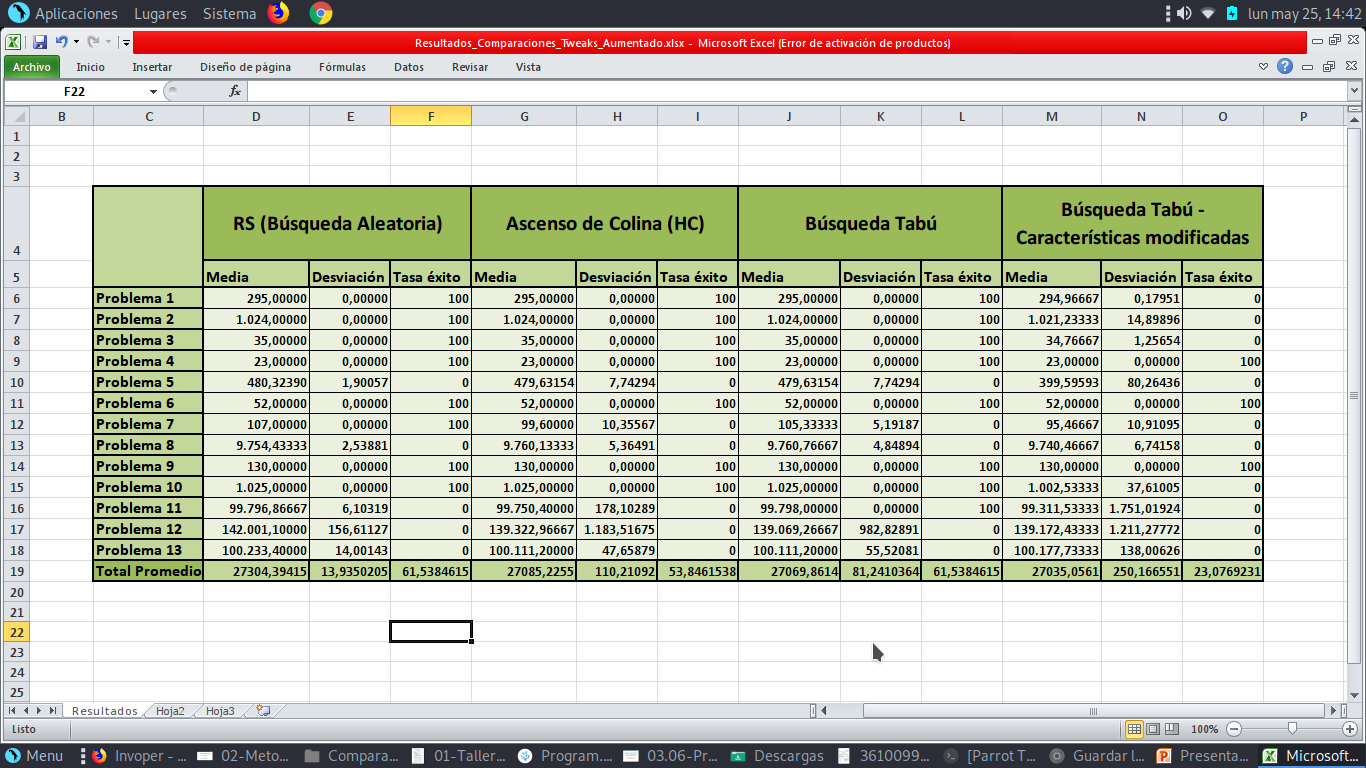
**Tabla 2: Búsqueda Tabú con características**

La **Búsqueda Tabú con características**  en teoría es el mejor de estos 4 algoritmos para solucionar el problema de la mochila, por la forma en que se trata la lista tabú.

Al resolver 13 problemas con los 4 algoritmos, nos llevamos varias sorpresas. **Búsqueda Tabú con características** fué la de menor rendimiento, con una Tasa de éxito igual a la **Búsqueda Tabú.**

Un posible factor de los anterior pudo ser la implementación del **Tweak** que se debió cambiar para este algoritmo por su naturaleza diferente (Manejo de la lista tabú con características), aunque se modificó el **Tweak** de la solución se intentó hacerse en lo más mínimo para no tener consecuencias.

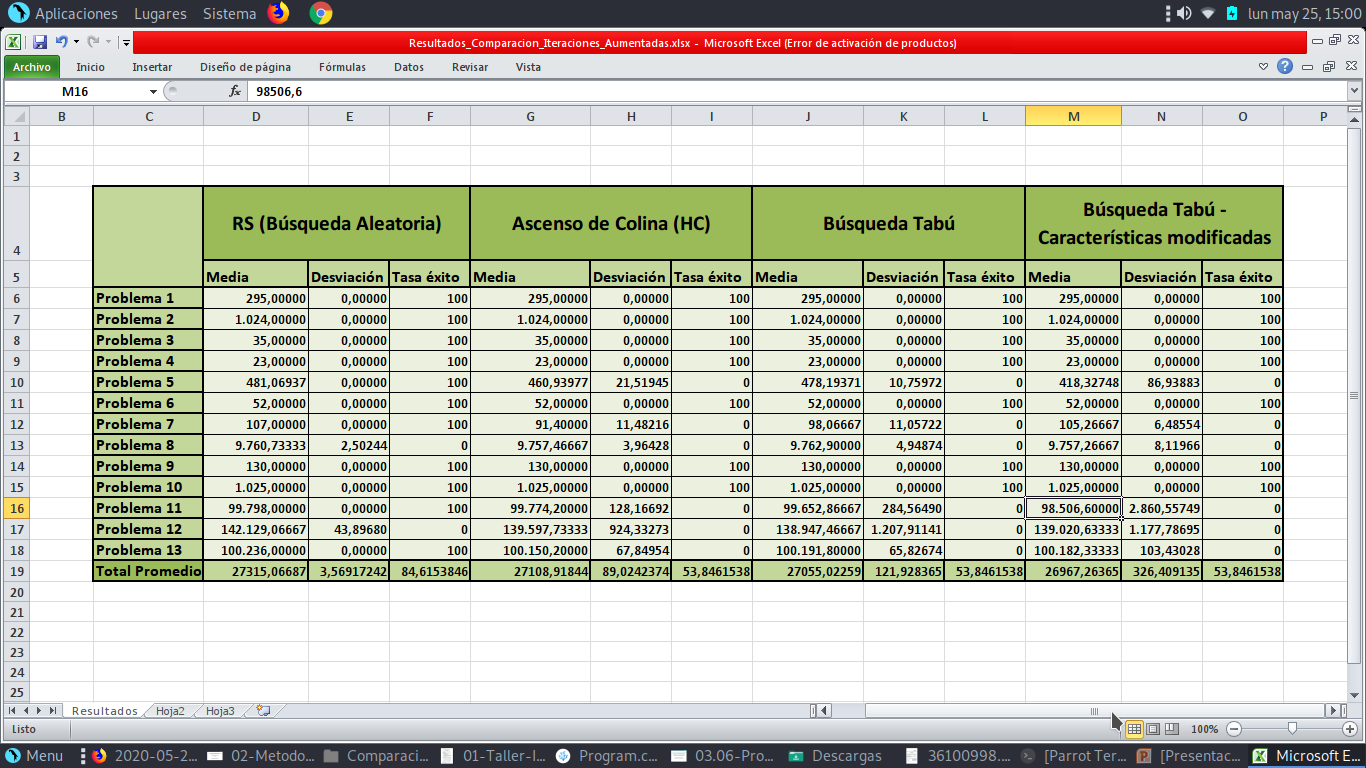
1. Prueba de **parámetros** en las **Búsquedas Tabú** para mejorar su rendimiento:

Para mejorar el rendimiento de las **Búsquedas Tabú,**  se intenta aumentando el número **n** correspondiente al número de **Tweaks.**

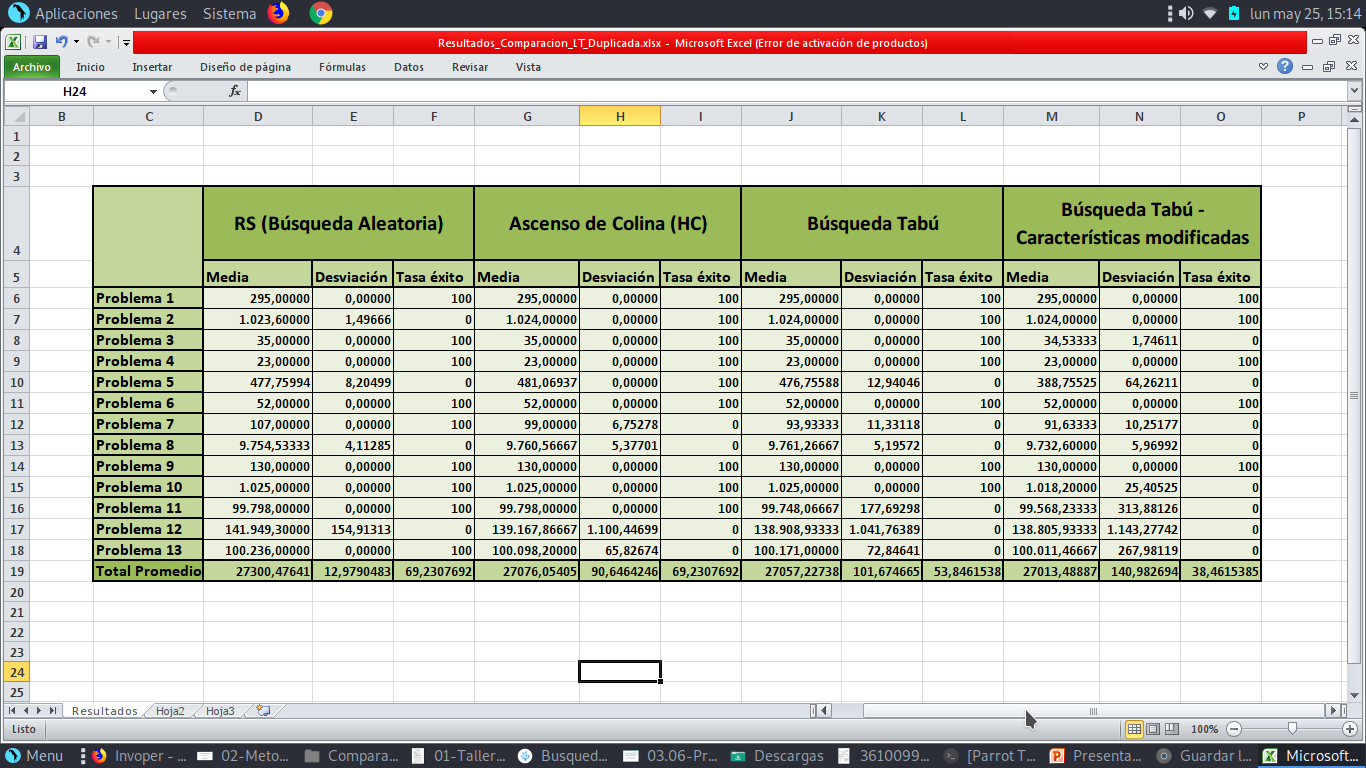
**Tabla 3: Prueba de parámetros – Número de Tweaks**

Al aumentar el número de **Tweaks – n,** la **Búsqueda Tabú con características** bajo de rendimiento notablemente y llegando a una **Tasa de éxito** del 23.1% que es muy baja. La **Búsqueda Tabú** aumentó su **Tasa de éxito** llegando al 61.53%.

También se aumentaron el número de evaluaciones a un número de 10.000 lo que regresó la Tasa de éxito de la **Búsqueda Tabú con características** a casi 53.9%, igual que la **Búsqueda Tabú** y sin sobrepasar a los algoritmos restantes. Cabe resaltar que la **Búsqueda aleatoria** obtuvo una **Tasa de éxito**  del 84%.

**Tabla 4: Prueba de parámetros – 10.000 evaluaciones**

Por último se intentó aumentando el tamaño máximo de la lista tabú **l** de la **Búsqueda Tabú** al doble.

**Tabla 5: Prueba de parámetros – Aumentando la capcidad de la lista tabú**

Pero no se obtuvieron mejoras.

La convergencia de las **Búsquedas Tabú** depende de una buena solución inicial, este puede ser un factor que nos explica el bajo rendimiento de los algoritmos.

**Conclusiones:**

* El algoritmo de **Búsqueda Aleatoria** tiene un alto rendimiento en problemas pequeños y no en problemas complejos por su alto grado de aleatoriedad.
* El número de evaluaciones en un algoritmo no nos define soluciones más óptimas.
* Los algoritmos de estado simple son eficientes, pero para su buen aprovechamiento se debe tener un buen conocimiento del problema que se quiere resolver.
* Un algoritmo por se más complejo no arroja mejores soluciones.