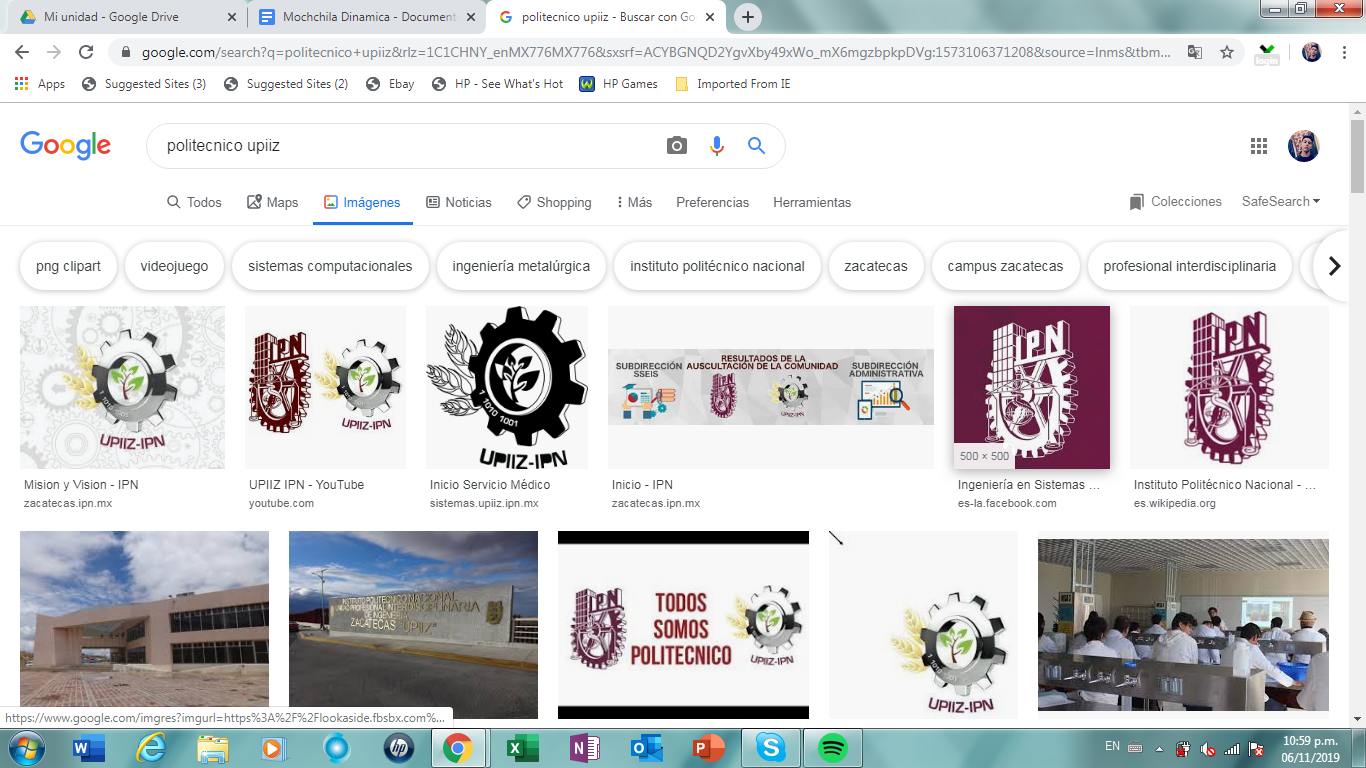
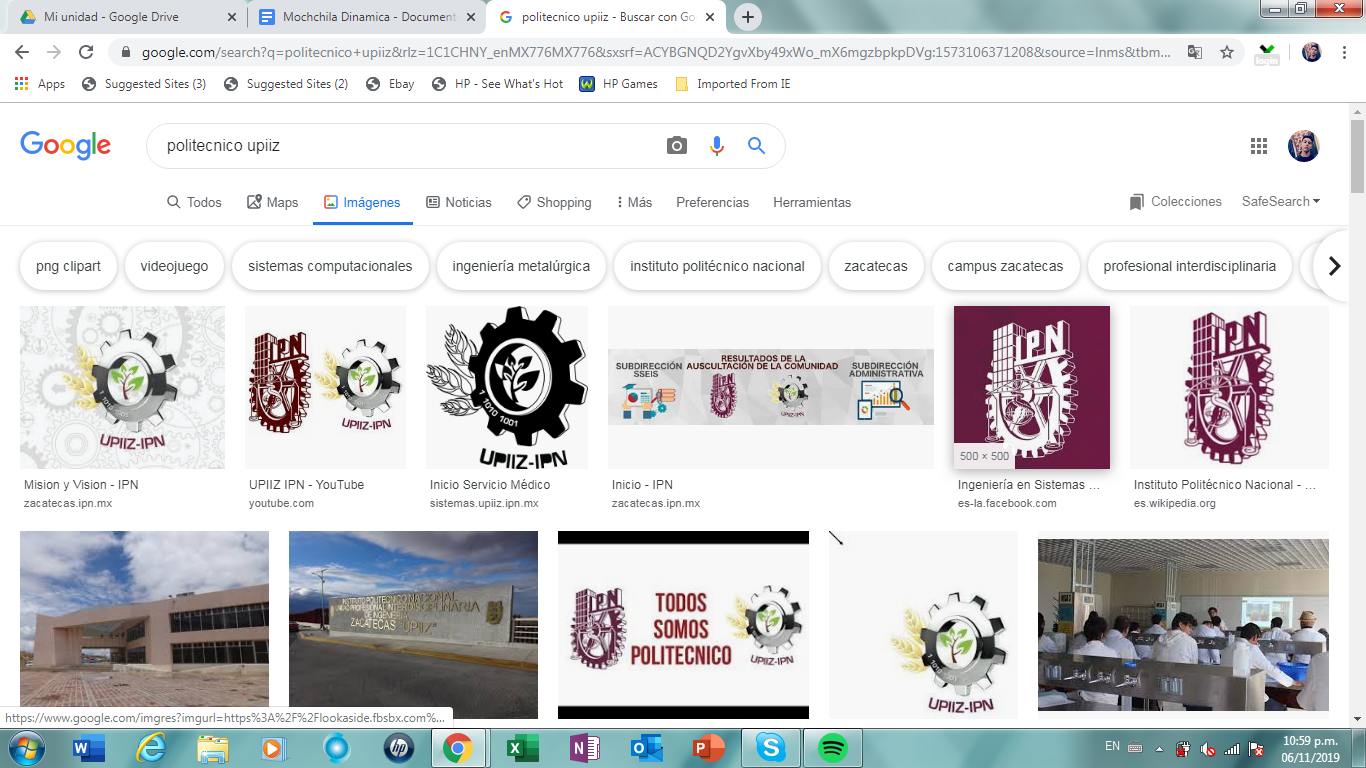
**INSTITUTO POLITÉCNICO NACIONAL.**

**UPIIZ.**

**MATERIA: Análisis De Algoritmos**

**Empate de cadenas**

**DOCENTE: Roberto Oswaldo Cruz**

**Leija**

**ALUMNO: César Delgado Padilla**

**GRUPO: 3CM1.**

**FECHA: 14/11/2019**

**Marco Teórico:**

### **Algoritmos de búsqueda simple de subcadenas**

También llamados por su denominación en inglés **Single string Matching**. En este tipo de algoritmos sólo se busca una subcadena a la que llamamos patrón, es decir el objetivo es encontrar todas las ocurrencias del patrón p dentro del texto. Este tipo de algoritmos se suelen agrupar en alguno de los siguientes tipos

1. Fuerza bruta. La idea es ir deslizando el patrón sobre el texto de izquierda a derecha, comparándolo con las subcadenas del mismo tamaño que empiezan en cada carácter del texto.
2. Leer todos los caracteres del texto uno a uno modificando en cada paso algunas variables que permitan identificar posibles ocurrencias. A este tipo pertenecen los algoritmos de [Knuth-Morris-Pratt](https://es.wikipedia.org/wiki/Algoritmo_Knuth-Morris-Pratt), [Shift-Or](https://es.wikipedia.org/w/index.php?title=Algoritmo_Shift-Or&action=edit&redlink=1) o búsqueda simple con autómata determinista.
3. Buscar el patrón en una ventana que se desliza a lo largo del texto. Para cada posición de esta ventana buscamos de derecha a izquierda un sufijo de la ventana que corresponda a un sufijo del patrón. A este tipo pertenecen los algoritmos de [Boyer-Moore](https://es.wikipedia.org/wiki/Algoritmo_de_b%C3%BAsqueda_de_cadenas_Boyer-Moore), [Boyer-Moore-Horspool](https://es.wikipedia.org/w/index.php?title=Algoritmo_Boyer-Moore-Horspool&action=edit&redlink=1)​ y [Sunday Quick Search](https://es.wikipedia.org/w/index.php?title=Algoritmo_Sunday_Quick_Search&action=edit&redlink=1).​ Este tipo de algoritmos no suelen funcionar bien cuando el tamaño del patrón es pequeño y hay una probabilidad alta de encontrarlo en el texto.
4. La búsqueda se realiza de derecha a izquierda dentro de una ventana, pero en este esquema se busca el sufijo más largo en la ventana que es subcadena del patrón. Ejemplos de este tipo de algoritmos son [BDM](https://es.wikipedia.org/w/index.php?title=Algoritmo_BDM&action=edit&redlink=1), [BNDM](https://es.wikipedia.org/w/index.php?title=Algoritmo_BNDM&action=edit&redlink=1) y [BOM](https://es.wikipedia.org/w/index.php?title=Algoritmo_BOM&action=edit&redlink=1).​ Este tipo de algoritmos para patrones pequeños no suelen funcionar bien.
5. Esquemas basados en funciones hash. Ejemplo de este tipo de algoritmos es el de [Karp-Rabin](https://es.wikipedia.org/wiki/Algoritmo_Karp-Rabin)

[Karp-Rabin](https://es.wikipedia.org/wiki/Algoritmo_Karp-Rabin)

• Calcula un valor hash para el patrón, y para cada subsecuencia de M-caracteres de texto.

• Si los valores hash son diferentes, se calcula una valor para la siguiente secuencia.

• Si los valores hash son iguales se usa una comparación de Fuerza Bruta.

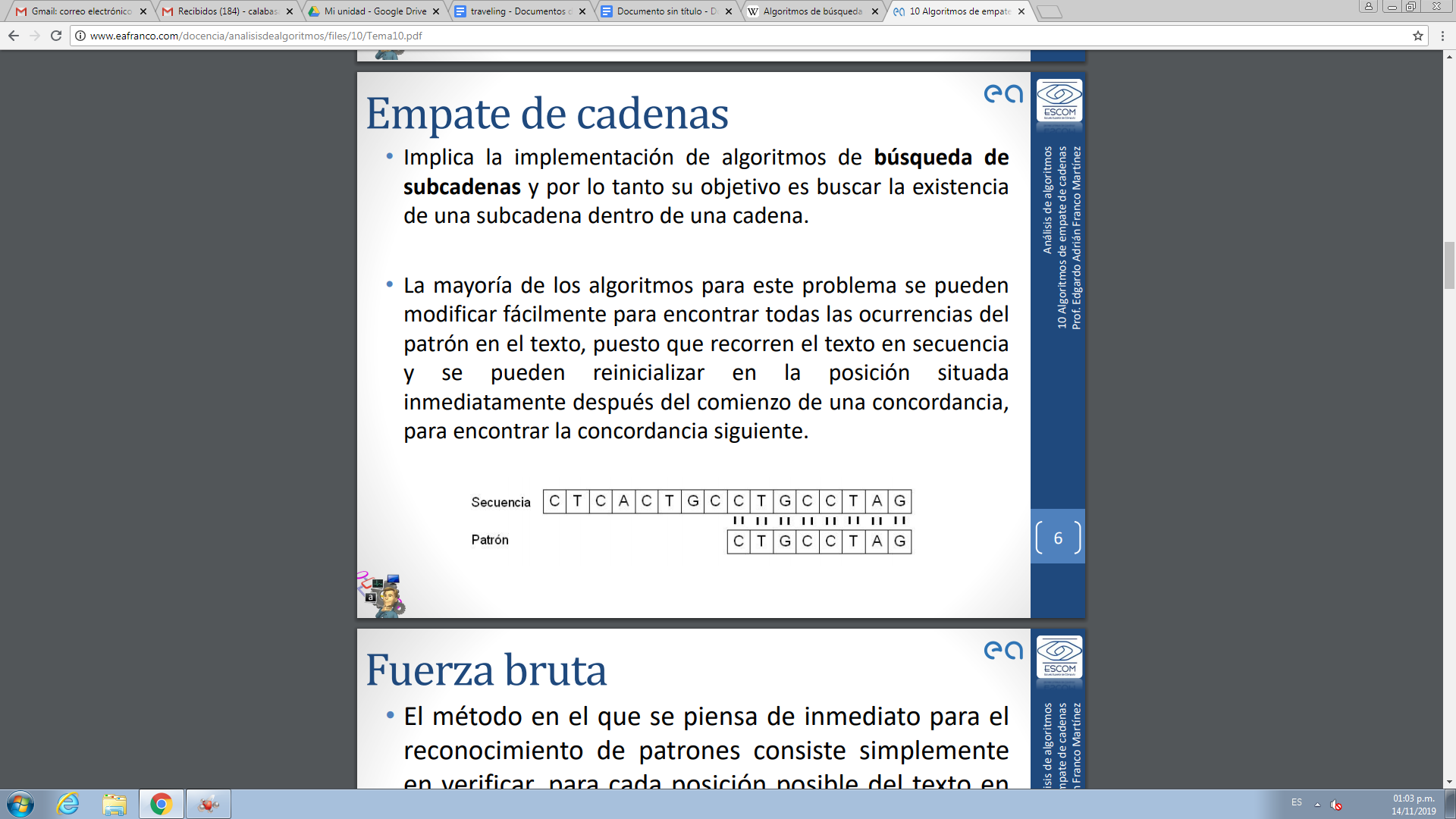
COMPLEJIDAD

Dado un patrón de M caracteres de longitud, y un texto de N caracteres de longitud.

• Complejidad Pre-procesamiento: O(M)

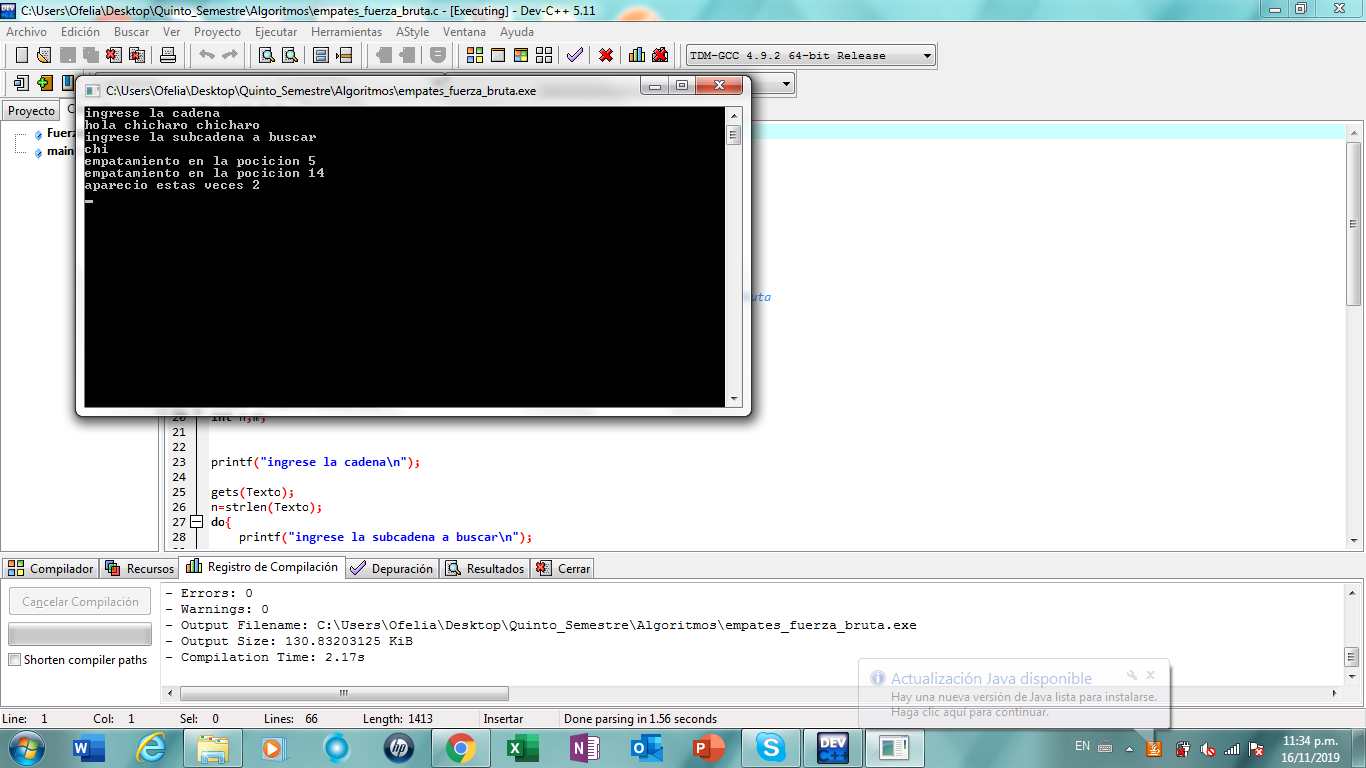
• Complejidad Búsqueda: O(MN)

**EJEMPLO:**

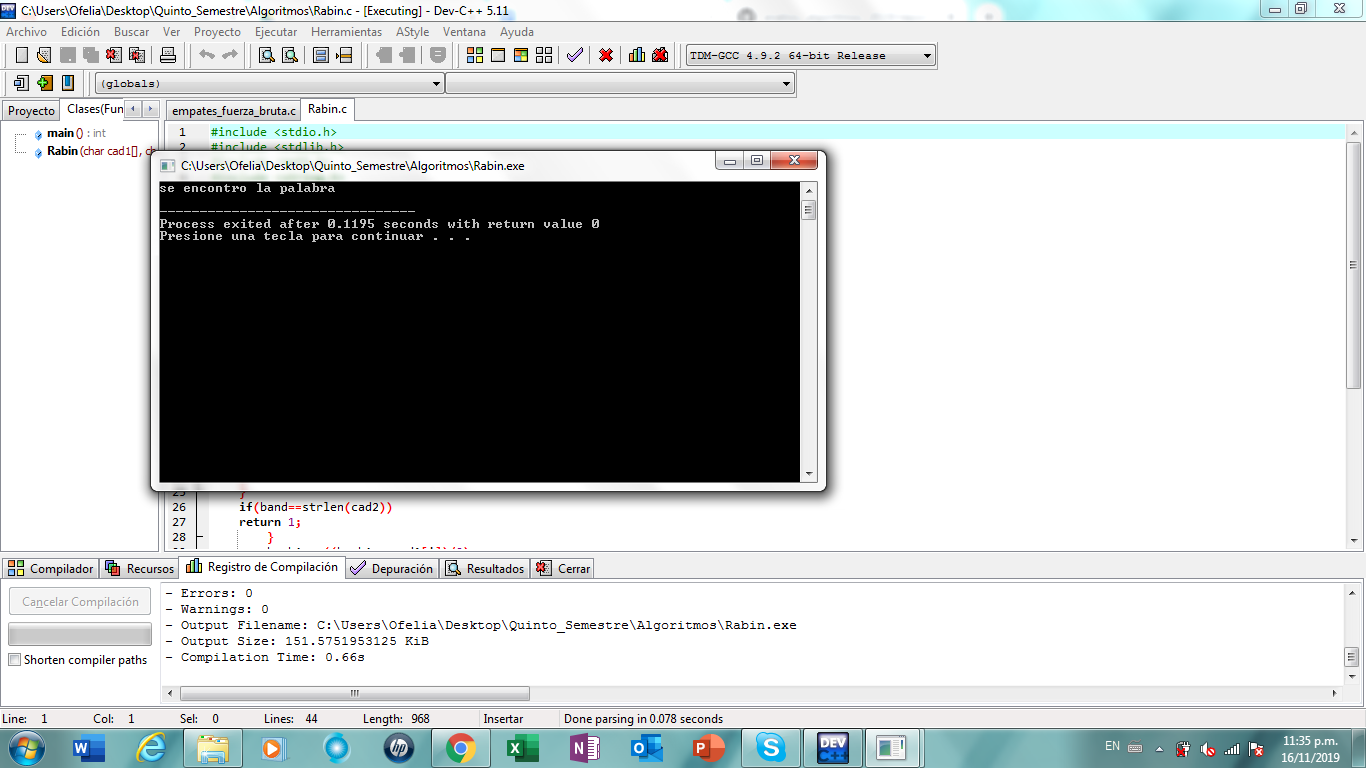
****

**esto solo dice si lo encontró o no y donde fue**

**Ejemplo con el código por fuerza bruta:**

****

**ejemplo con el código de Rabin**

****

**Conclusiones:**

**el empate de cadenas es un algoritmo un poco sencillo que solo compara dos cadenas, las cuales, la cadena debió ser más grande que la subcadena y regresa que se encontró, si así fue o no se encontró si no fue así,**