**TALLER DE MATEMATICA COMPUTACIONAL**

**TECNICATURA EN DESARROLLADOR DE APLICACIONES INFORMATICAS**

**AÑO 2017**

**FACULTAD DE CIENCIAS EXACTAS**

**UNCPBA**

**NOMBRE: NICOLÁS**

**APELLIDO: BONCORE**

**DNI: 40669266**

**INTRODUCCIÓN:**

El problema planteado nos consigna una nave espacial con un sistema de autentificación, el cual realiza disparos cada vez que introducimos un número de DNI. Mediante la función my\_mex\_service(DNI) vamos a introducir nuestro número de DNI y nos va a devolver un 1 en caso de autorizar el disparo o un 0 en caso contrario. Para realizar este problema debemos implementar una función, la cual, a partir de nuestro DNI y un cierto valor de épsilon, calcule dos fallos consecutivos de realizar el disparo mediante la función my\_mex\_service. Para resolverlo usaremos el método de Montecarlo.

Luego el problema nos dice que cambiemos el valor de épsilon y veamos cual es la probabilidad para esos valores dados. También debemos hacer que imprima por pantalla los valores y las gráficas obtenidas para cada valor de épsilon. También debemos calcular el desvió estándar de las probabilidades parciales en las primeras 20 y las ultimas 20 iteraciones de cada experimento. Por último debemos controlar el tiempo que tarda cada corrida del algoritmo.

**DESARROLLO:**

Para plantear el problema, lo primero que hice fue implementar una función (calcular\_probabilidad\_fallo\_doble) recibiendo mi DNI y un cierto valor de épsilon, para poder hallar la probabilidad estimada de que la función my\_mex\_service tire dos fallos consecutivos.

Cuando la diferencia entre la probabilidad anterior y la probabilidad actual es menor a épsilon, o la función se ejecutó al menos 40 veces, decimos que la función converge, lo cual nos da como resultado las probabilidades\_parciales y un arreglo con todas aquellas probabilidades.

Primero llame a la función “**function** [probabilidades\_parciales, probabilidad\_estimada] = calcular\_probabilidad\_fallo\_doble (épsilon, DNI)” y luego declare las variables necesarias.

Para calcular el tiempo de corrida que tarda cada corrida del algoritmo implemente el “tic” antes del **while**, y el “toc” al final del **while**.

Luego hice que la variable “probabilidad\_actual” sea igual a la variable “probabilidad\_anterior”

Después implemente un **if** para decir que, si la autorizacionDNI era falsa, y la autorizacionDNI2 era falsa, se incrementaría el caso favorable.

Para saber si está bien lo que hicimos, vamos a la ventana de comandos y ponemos “script\_trabajo\_especial” y tirará perfectamente los resultados.

Luego de esto se ejecuta la gráfica para ver los resultados obtenidos.

**RESULTADOS:**

Grafico 1: Probabilidad estimada = 0.780488:

Elapsed time is 0.00350714 seconds.

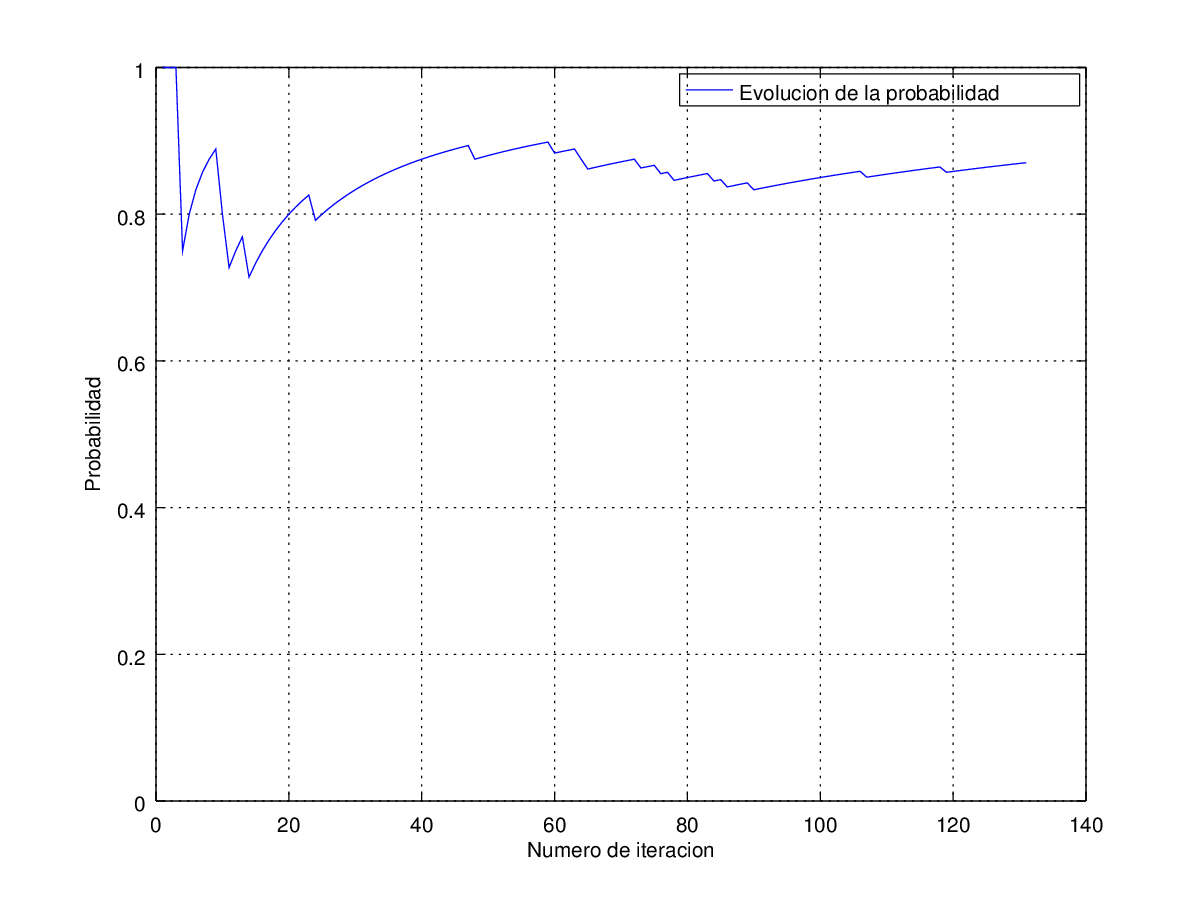


Grafico 2: Probabilidad estimada = 0.878049:

Elapsed time is 0.00250387 seconds.

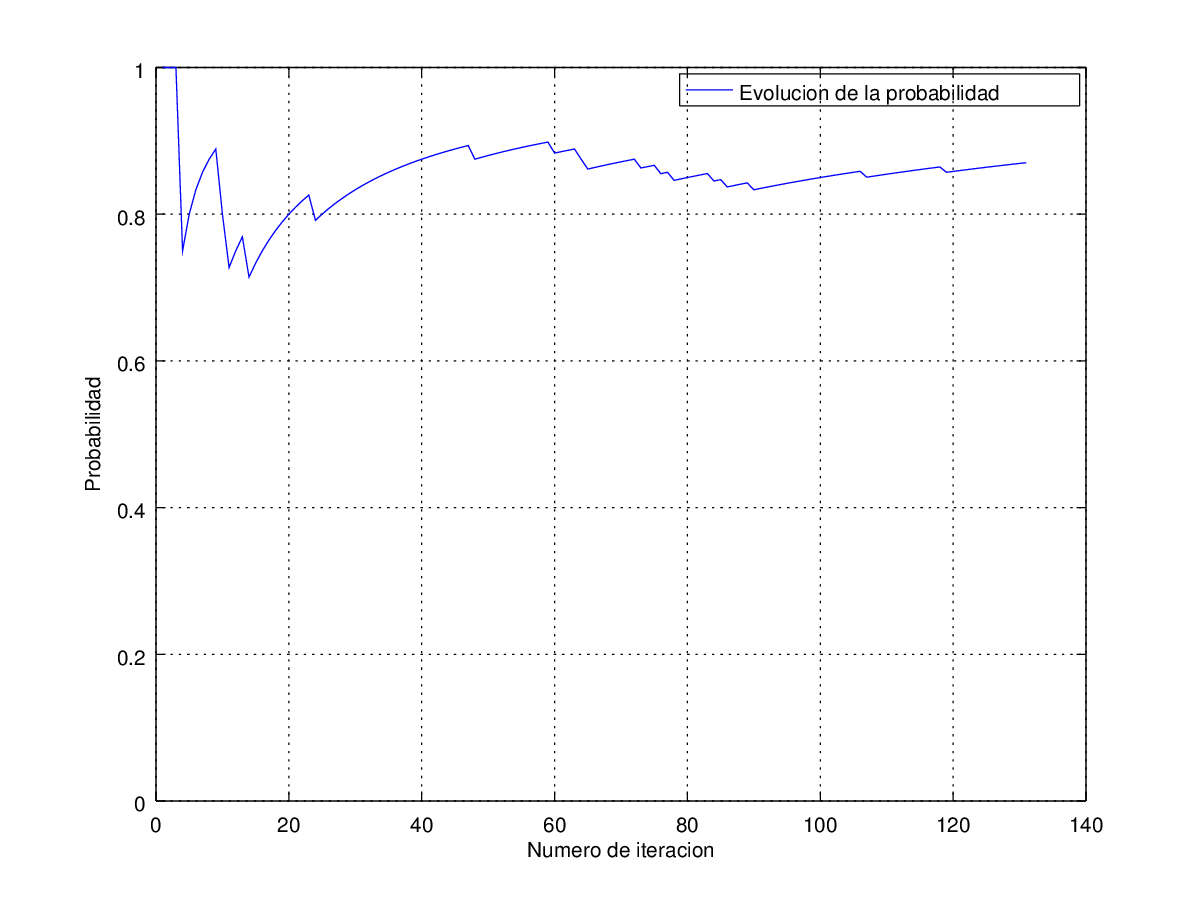
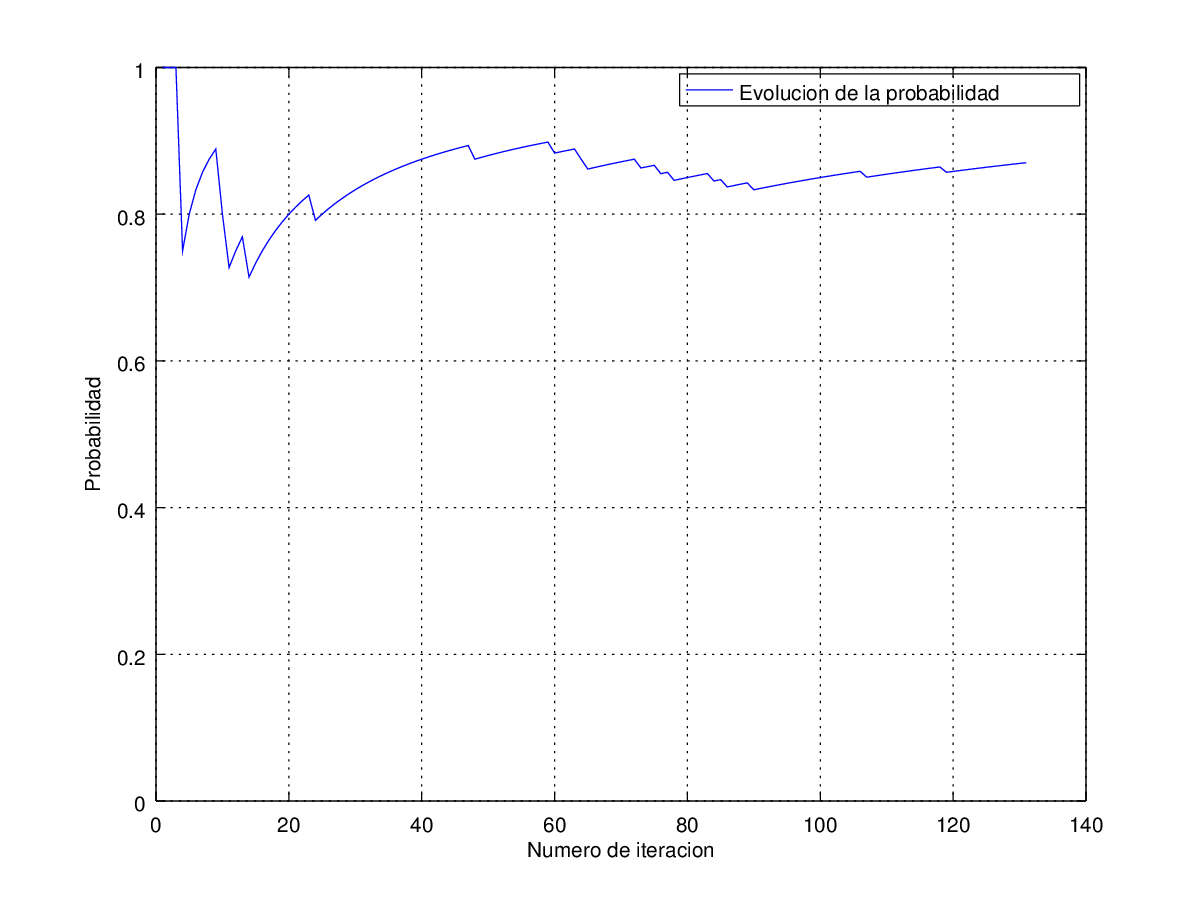


Grafico 3: Probabilidad estimada = 0.870229:

Elapsed time is 0.00852299 seconds.



**CONCLUSIÓN:**

Por los resultados vistos en las gráficas podemos decir que, cuanto más pequeño sea el valor de épsilon, más exacto será el valor de la probabilidad.

Analizando las gráficas podemos decir que cuanto más grande sea el número de iteraciones, también más exacto será el valor de la probabilidad.

Podemos observar que cuanto más chico es el épsilon, las primeras 20 iteraciones tienden a ser un desvío más grande y las ultimas 20 un desvío mas pequeño.