Ingeniería en Sistemas Computacionales

Fundamentos de Ciencias Computacionales

Primavera 2020

FCC Toolkit

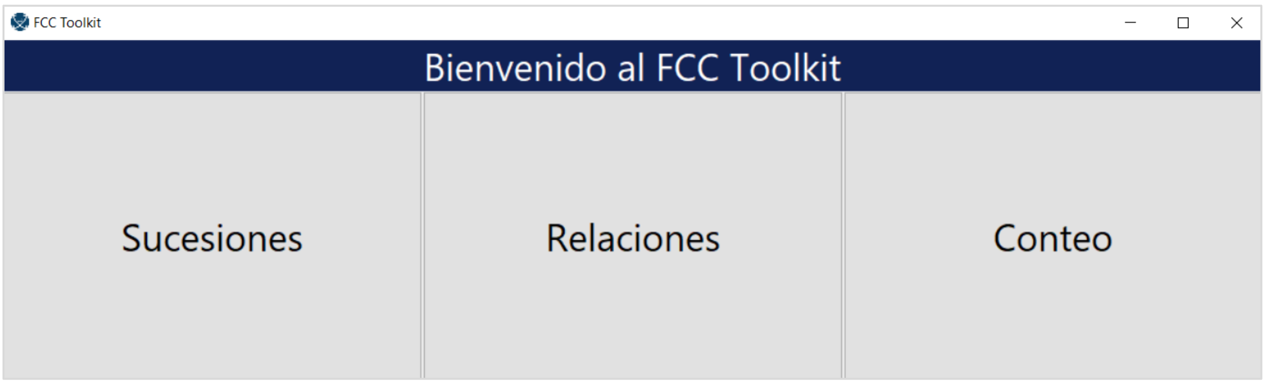
Alumno: Marco Ricardo Cordero Hernández

Profesor: Fernando Velasco Loera

Zapopan, Jal., 18 de May. De 2020

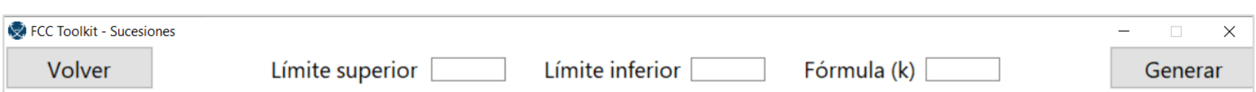
Descripción del programa

El programa ha sido construido de manera en que el usuario pueda interactuar con éste de manera intuitiva y gráfica. El usuario es bienvenido con un menú de opciones en los cuales podrá escoger de entre un grupo de algunos temas vistos a lo largo del curso. La idea del diseño minimalista fue pensada en la importancia de los datos de salida y no en la decoración como tal, sin embargo, esto no significa que no se hayan cuidado los detalles de las interfaces.

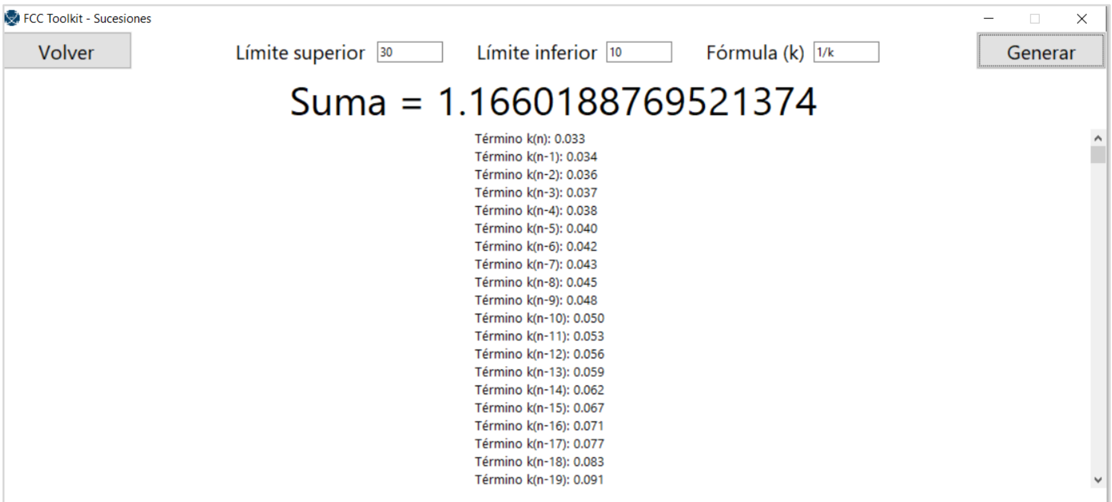


Sucesiones

La primera opción es la de sucesiones. Los campos de entrada están denotados por etiquetas al lado de cada uno.

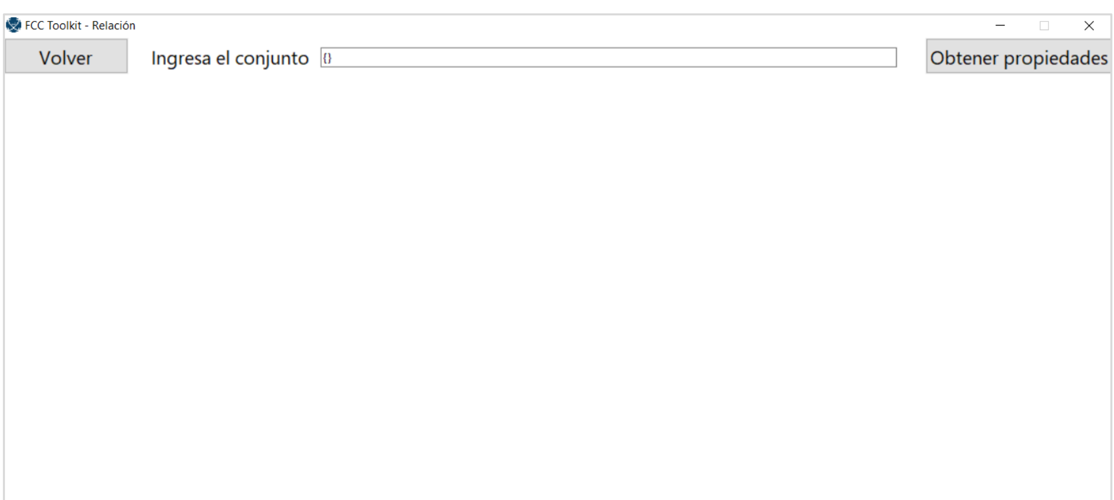


El botón **Volver** termina la parte actual de la aplicación y retorna el menú inicial. Los campos de entrada de límites deben ser números enteros, de lo contrario, una alerta se mostrará. El campo para la fórmula especifica como se debe ingresar una expresión en términos de k, es decir, la función dependerá de esta variable. Por ejemplo: 2^k; 1/k; 30k + 7; etc. El ingreso de esta fórmula está pensado en la regla básica de computación que dice “las computadoras hacen lo que el usuario les indica”, esto se refiere a que si una fórmula es ingresada sin formatear, los resultados serán diferentes de lo deseado (2^k+1 ≠ 2^[k+1]). Para finalizar las operaciones del usuario y ya con los datos ingresados, se deberá presionar el botón **Generar** para obtener la suma total y los términos calculados.

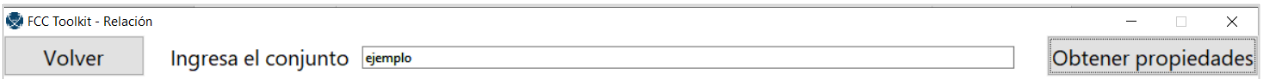


La imagen anterior muestra el cálculo con el caso $\sum_{10}^{30}\frac{1}{k}$, caso que arroja 21 términos. Del lado derecho se puede apreciar una barra de desplazamiento que controla el despliegue y exhibición de cada uno de los términos individuales calculados. Se recomienda no hacer ninguna operación que de antemano se conozca que generará resultados muy grandes (∞ + ∞ = ∞).

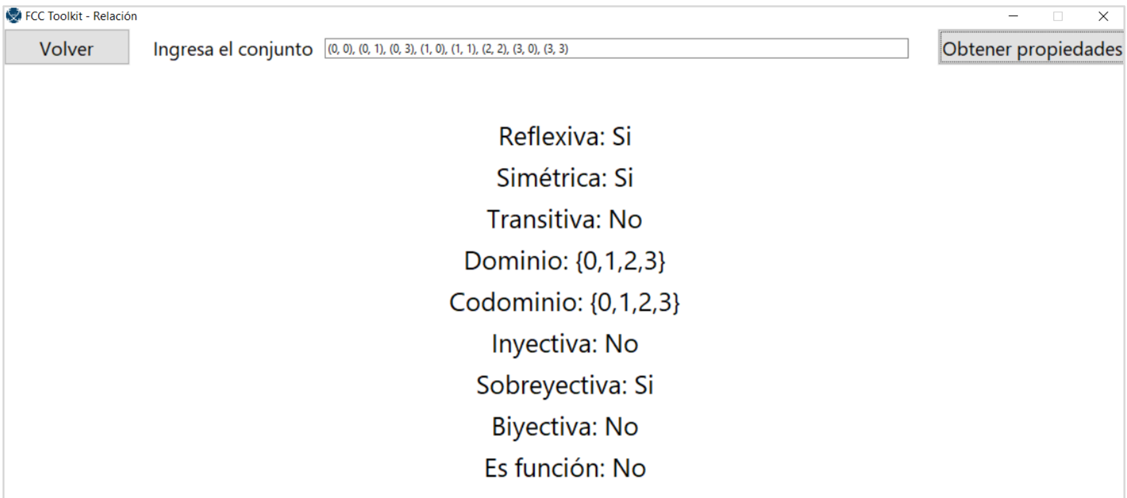
Relaciones



La segunda opción es la de relaciones. La forma de ingresar la relación puede ser de dos maneras: mediante llaves {(a,b),(c,d),…}; y de forma directa (1,2),(3,),…,(x,y). El ingreso de los datos no está limitado a números, esto significa que también se pueden ingresar caracteres alfanuméricos. Es muy importante recalcar que las entradas no deben contener espacios entre sus elementos, de ser el caso, el programa no detectará como válida la entrada.

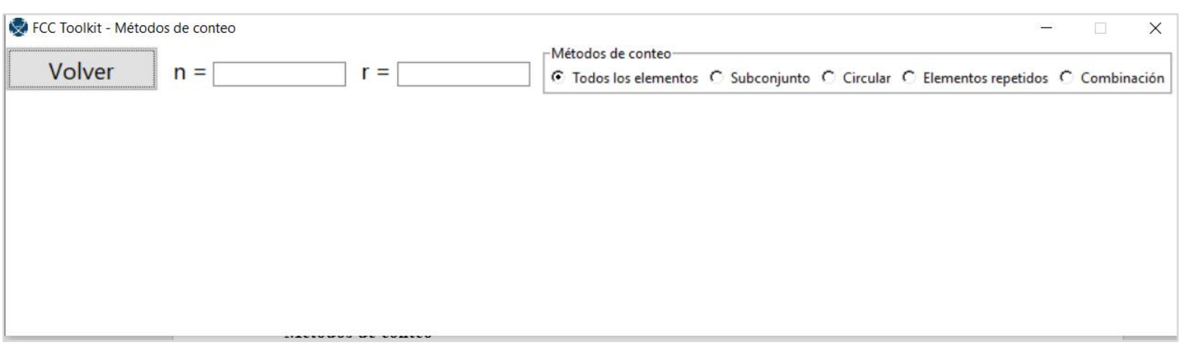


Si se desea obtener una demostración del funcionamiento se puede ingresar la palabra clave **ejemplo** el cual retornará un conjunto previamente seleccionado.



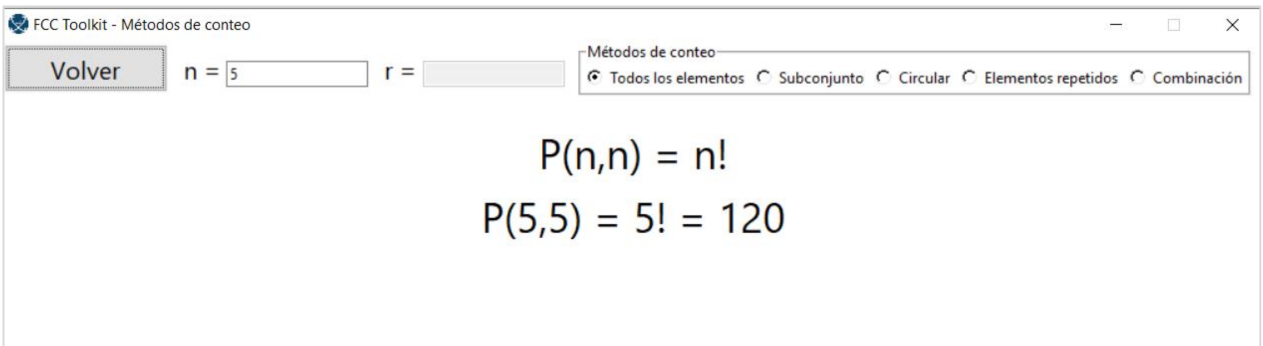
Se pueden observar tres características primordiales: reflexividad, simetría y transitividad. El dominio y codominio (rango) son mostrados a manera de conjunto ascendente. Existen cuatro propiedades adicionales que de igual manera son determinadas a partir de las operaciones realizadas. Si se desea hacer una nueva operación basta con borrar el contenido de ingreso y poner uno nuevo.

Métodos de conteo



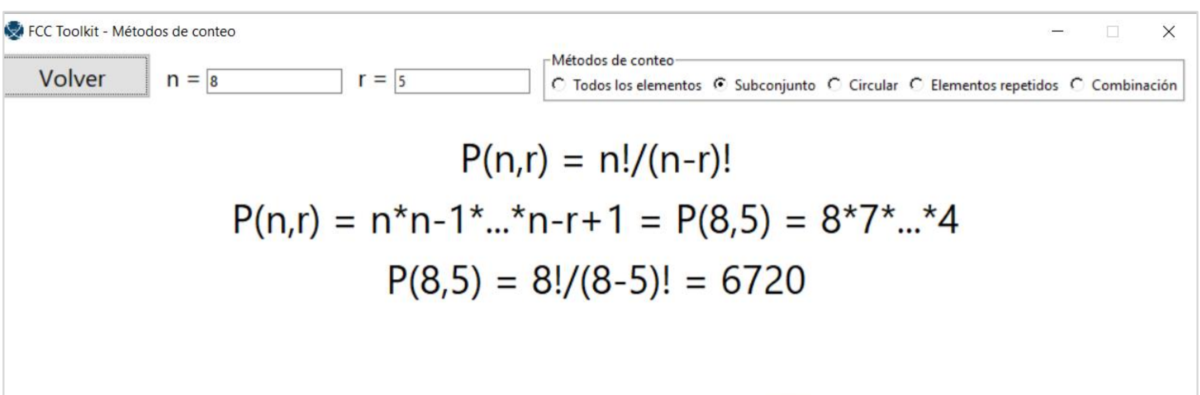
La última de las opciones recae en los métodos de conteo. Las opciones se han agrupado a la derecha de los campos de entrada de manera organizada, siendo los primeros 4 elementos de permutaciones y el último de combinación. Los campos de entrada cambian dinámicamente dependiendo la opción que se ingrese, por ejemplo, si se selecciona la primera opción sólo se requerirá del valor de entrada de **n** y el valor de **r** será desechado. Si se ingresan valores alfanuméricos para opciones diferentes a **Elementos repetidos**, un error será arrojado; si se ingresa un valor **r** mayor a **n** en las opciones en donde se requieran ambos datos, un error será mostrado indicando esto. Para todas las opciones se mostrarán las fórmulas que se utilizarían para calcular las operaciones deseadas, tanto como la versión sin sustitución de valores como la manipulada. Finalmente, el resultado concreto de la operación será desplegado.

A continuación se muestran algunos ejemplos.



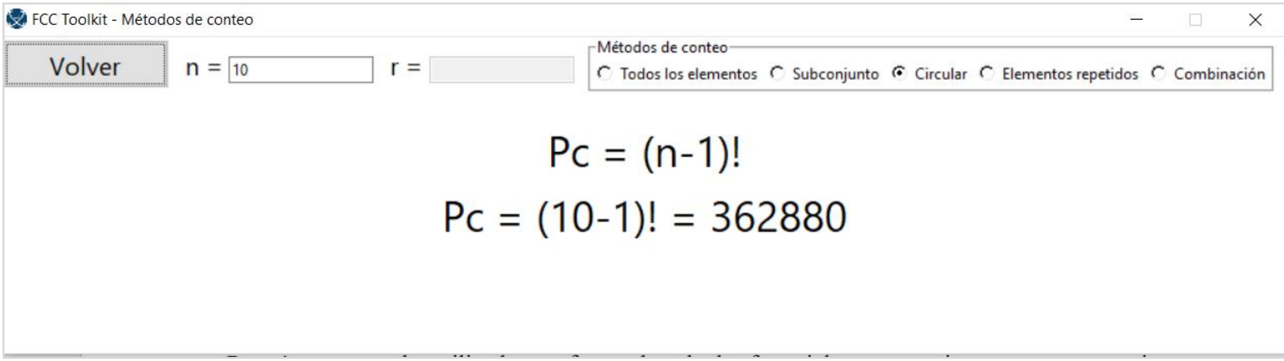
La opción de todos los elementos (nPn) es una simple operación de factorial. Para el ejemplo, **n** ha sido **5** que se calcula de manera en que los recursos del equipo del usuario no sean agotados prematuramente (así como en todas las operaciones presentes).

La caja de entrada para **r** ha sido desactivada, indicando que para el caso no es útil obtener algún valor para la variable. Para reactivarla basta con hacer **click** sobre ella.

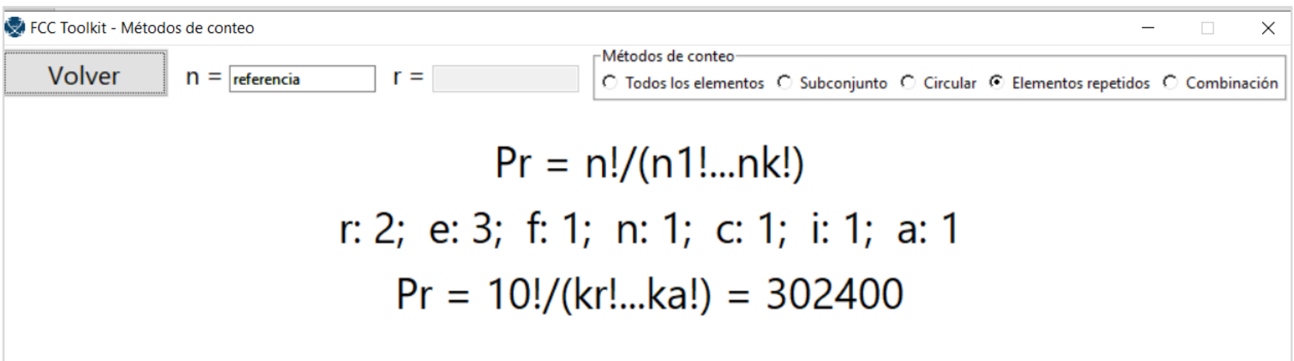


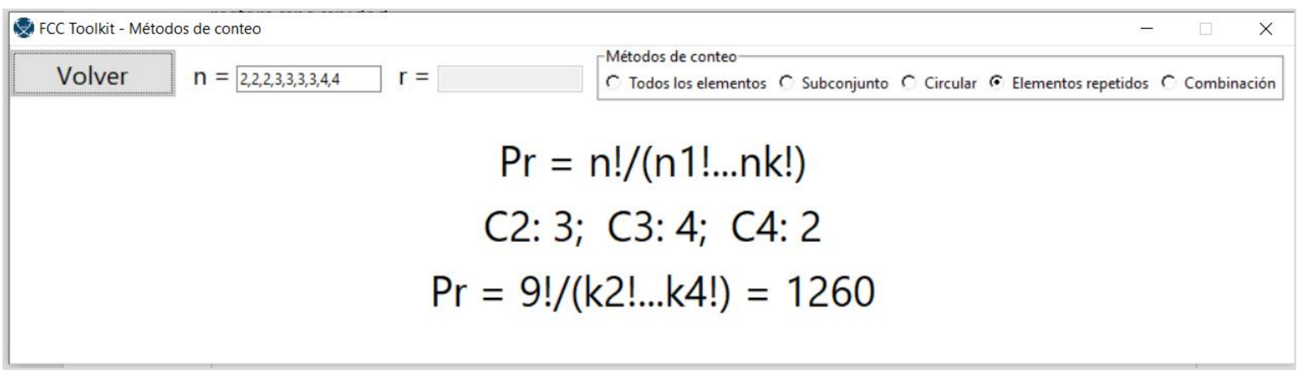
Para un subconjunto (nPr, n tomados r a la vez) es necesario el ingreso de ambos valores **n** y **r**.

Para éste caso se ha utilizado una forma de calcular factoriales que requieren menos memoria y significan tiempos de ejecución más eficaces.



Para una permutación circular (P_c) es por igual un factorial simple, sólo que al argumento se le restará una unidad.

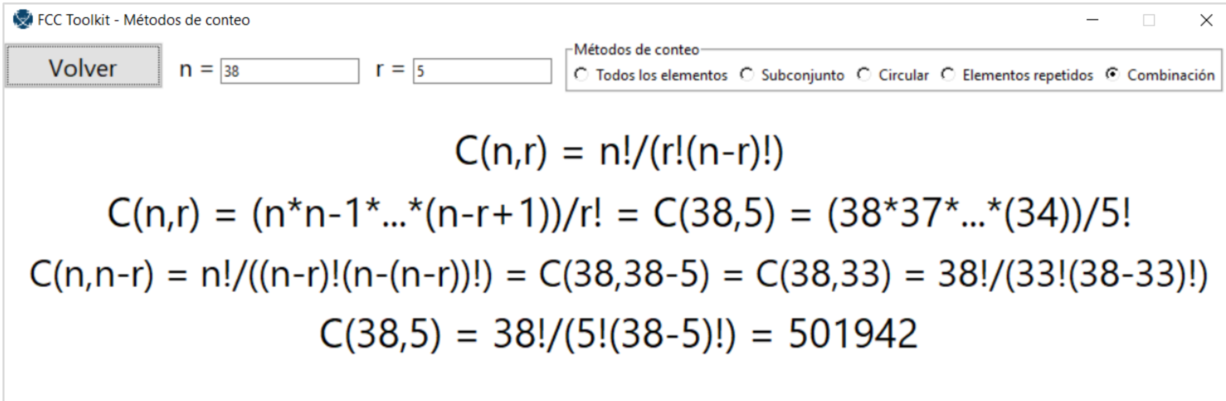




La opción de elementos repetidos ($k_1,k_2,\ldots,k_nPn$) es quizás la más interesante de todas las operaciones. La entrada **n** acepta dos modos de datos que son indicados cada que la opción se selecciona: una palabra o una lista de números enteros separada por comas. En cualquier caso, el valor mínimo de elementos requerido para operar es de 2.

Si la opción por la que se ha optado ha sido la de una palabra, el programa dividirá la palabra en cada uno de sus elementos, contará las ocurrencias de cada uno y luego desplegará estos elementos. Finalmente, se mostrará el resultado de la permutación con repeticiones.

Si la opción seleccionada es la de una lista de enteros, se indicará en cada elemento **C**onteo del elemento y luego se desplegará el resultado.



Para finalizar, la opción de combinación (nCr) también requiere de ambos valores de entrada.

Importancia en desarrollo profesional

El conjunto de herramientas desarrolladas tiene un gran peso e importancia en el desarrollo del resto de la carrera que he seleccionado. Las oportunidades de emplear los conocimientos adquiridos pueden aparecer de la nada y que mejor que aplicarlos mediante un proceso automatizado. Uno como ingeniero debe ser capaz de retener mucha información y aplicaciones en la vida real, porque siempre será diferente 1 que 0 y será mejor contar con n conocimientos a n-1.

El tema de las sucesiones es una cosa sumamente amplia que muchos de nosotros no percibimos pero que, sin embargo, siempre está de alguno u otra manera en el entorno en el que nos encontramos. Hablando de un enfoque más personal, las materias como cálculo integral, vectorial, multivariable y demás utilizan en algún punto series de funciones, es decir, sucesiones de funciones con variables que requieren de sumas que tienden incluso al infinito. Esto puede parecer un proceso muy largo e incluso imposible, sin embargo, si no se cuenta con los conocimientos suficientes para resolver una problemática donde esté involucrado lo anterior, se pueden aproximar algunos resultados mediante sumas cada vez más grandes.

El trabajo y manipulación de funciones está ligado fuertemente con las matemáticas. Por ejemplo, la propiedad biyectiva de una función puede ahorrar trabajo en una integral definida, ya que si una función es par se puede reducir el trabajo en el cálculo de los valores usando el teorema de funciones pares. También se pueden determinar los dominios y rangos extendidos de ciertas funciones al analizar la manera en que se comporta un conjunto de datos relacionados a las funciones. No se puede dejar de lado la aplicación en la reducción de procesos dentro de teoría de computación, siendo esto la piedra angular de muchas invenciones actuales-.

Más adelante en la carrera, existe un curso de probabilidad y estadística, en donde la aparición de los métodos de conteo está más que asegurada y con el adelanto de la próxima teoría será en cierto modo más ligero el trabajo racional que se empleará. Dichos conocimientos son tan útiles y sencillos de implementar en situaciones cotidianas como lo podría ser la formación de un grupo de alumnos.

El mundo actual no está como para despreciar cualquier tipo de conocimiento y este proyecto me ha hecho reforzar esta creencia.

Bibliografía

Python 3.8.1. (2019). Python. Recuperado de www.python.org

effbot (2009). Fredrik Lundh. Recuperado de www.effbot.org