



# ALGORITMOS GENÉTICOS

Son algoritmos de optimización estocásticos basados en población, inspirados en la evolución de Darwin (la supervivencia del mejor adaptado).

Bien...

Y esto, con qué se come?



Cuando queremos optimizar algo, por ejemplo, el recorrido que debe hacer el motoquero que entrega paquetes, cómo hacemos esa búsqueda?

- Podemos clasificar por CÓMO se realiza:
  - \* Blind Search: búsqueda exhaustiva de todas las opciones
  - \* Guided Search: utilizar los resultados obtenidos en pasos anteriores para guiar la búsqueda del óptimo:
    - @ Single State Search: usa valores parecidos a la primer solución (si es poco compleja, la búsqueda converge)
    - @ Population State Search; usa un población de soluciones para ir mejorando (tarda más en converger)
- ... o si es ESTOCÁSTICA (orden aleatorio)  
o DETERMINISTA (haciendo siempre el mismo orden)

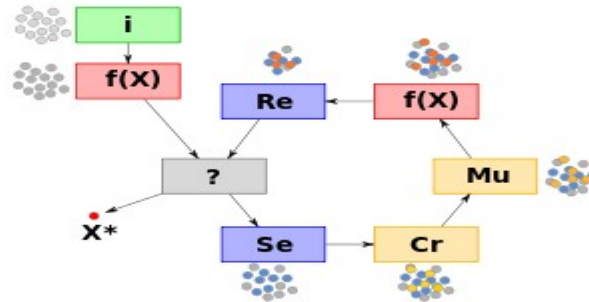


# Funcionamiento

- Cada candidata a solución se denomina INDIVIDUO
- Cada INDIVIDUO tiene GENOTIPO, GENOMA ó CROMOSOMA (es la estructura de datos del individuo)
- Un GEN es una posición dentro de la estructura de datos, el ALELO es el dato en sí
- Partimos de un conjunto de Individuos cada uno con Alelos distintos
- Para encontrar a los individuos óptimos, debemos evaluarlos con una función llamada FENOTIPO (fitness)
- Hacemos una selección: nos quedamos con los individuos con mejor fitness



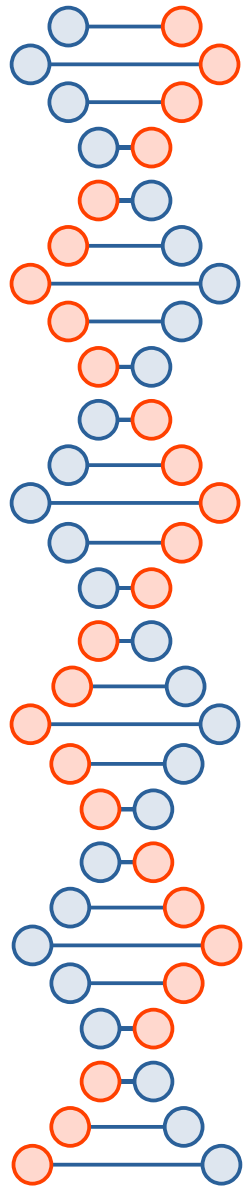
- Los individuos seleccionados se juntarán para tener descendencia (proceso de CRIANZA o breeding)
- Para criar, hay dos operadores genéticos:
  - \* CROSSOVER: combinación de 2 individuos
  - \* MUTACIÓN: cambiar un individuo actual
- Luego de la generación de Descendencia, se vuelve a iterar calculando el fitness de los nuevos individuos y sucesivos pasos



**Algoritmo genético** i: inicialización, f(X): evaluación, ?: condición de término, Se: selección, Cr: cruzamiento, Mu: mutación, Re: reemplazo, X\*: mejor solución.

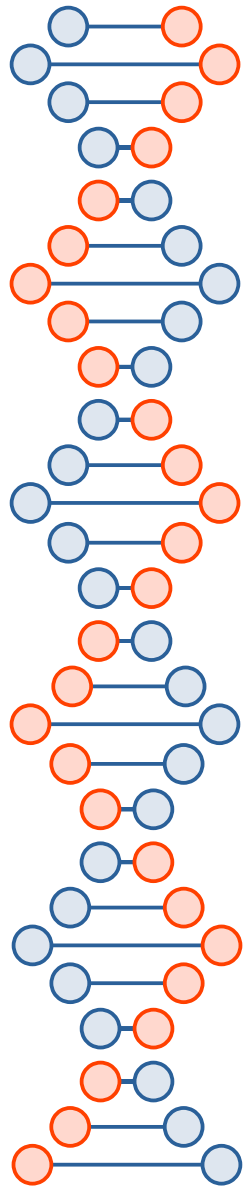
# Cómo seleccionar individuos

- Método de la ruleta: la probabilidad de Breeding es proporcional al fitness de individuo ( $\text{prob} = \text{fitness} / \text{suma\_tot}$ )
- Ranking: si hay individuos con parecido fitness, la Ruleta no sabrá elegir al mejor -> ordenar N individuos con mejor fitness
- Selección de estado estacionario: eliminar individuos con mal fitness, se los reemplaza por otros criados a partir de los de mejor fitness
- Selección de Torneo: se crean parejas aleatoriamente, de cada pareja se elige al que tiene mejor fitness
- Elitismo: se hacen reproducciones parciales, los que tienen mejor fitness se 'clonan'



# Cómo obtener nuevos individuos

- Por CROSSOVER:
  - \* Cruce de un punto: se dividen los alelos de los padres en 2, el hijo se lleva una parte de cada progenitor
  - \* Cruce de varios puntos: idem anterior, pero la división de alelos será en 3 ó más grupos
  - \* Cruce uniforme: los alelos del hijo se eligen de manera aleatoria a partir de los de los padres
  - \* Listas ordenadas: si todos los alelos no son elegibles, se pone una serie de condiciones para evitarlos
- Por MUTACIÓN:
  - \* Mutación aleatoria: se muta aleatoriamente un alelo
  - \* Shrink: se suma un valor aleatorio de una distribución normal ( u otro tipo) a uno de los valores del individuo



- Hay varias maneras de implementar en Python
    - \* librerías como PyGAD
    - \* crear nuestro propio algoritmo genético: como acá
- Optimización con algoritmo genético Python by Joaquín Amat Rodrigo, available under a Attribution 4.0 International (CC BY 4.0) at [https://www.cienciadedatos.net/documentos/py01\\_optimizacion\\_ga](https://www.cienciadedatos.net/documentos/py01_optimizacion_ga)
- Ejemplo sencillo con PyGAD