



# Introducción

Estructura de Datos

Dr. Víctor de la Cueva

[vcueva@itesm.mx](mailto:vcueva@itesm.mx)

## Sumando números

- Reglas del juego:
  - Se tienen dos jugadores.
  - Cada uno selecciona un número del 1 al 9 en forma alternada.
  - No se pueden repetir los números.
  - El primero que logre tener 3 números que sumen 15 gana.

```

Function A* sta
  allocate:
  openat :=
  come from:
  g_score: sta
  h_score: sta
  while open:
    current:
    if cur:
      ret:
      remove:
      new cur:
      for sta:
        if:
          ter:
          if:
      return fail
Function recon:
  if current:
    p := re:
    return:
  else:
    return:

```

## Estructura de datos

- En ciencias de la computación, una estructura de datos es una **forma particular de organizar** datos en una computadora para que puedan ser **utilizados de manera eficiente** [1].
- Todos los datos se guardan en la RAM de la computadora, la cual tiene un ordenamiento secuencial.
- La idea principal es guardar cierta **información adicional** para que sea más fácil el acceso a dichos datos, con el objetivo de almacenarlos y usarlos de una forma eficiente (poco espacio y rápido acceso).

```

Function A* sta
  allocate:
  openat :=
  come from:
  g_score: sta
  h_score: sta
  while open:
    current:
    if cur:
      ret:
      remove:
      new cur:
      for sta:
        if:
          ter:
          if:
      return fail
Function recon:
  if current:
    p := re:
    return:
  else:
    return:

```

## La receta mágica en las ciencias computacionales

- Las estructuras de datos están **íntimamente relacionadas con los algoritmos**.
- Muchos algoritmos **se vuelven más eficientes** si se utiliza la estructura de datos adecuada.
- Los algoritmos y las estructuras de datos están **inherentemente relacionadas** para crear programas:

**Algorithms + Data Structures = Programs**

- Libro de Niklaus Wirth, escrito en 1976.



```
Function A* at:  
  calculate:  
    operat :=  
    come_from:  
  
  g_score[st:  
    // get node  
  h_score[st:  
  
  while oper:  
    current:  
    if cur:  
      ret:  
  
    remove:  
    new cur:  
    stop sta:  
    if:  
  
    tim:  
  
    if:  
  
  
  return fail  
  
Function recon:  
  if current:  
    p := re:  
    return:  
  else:  
    return:
```

## Referencias

- [1] Wikipedia ([https://es.wikipedia.org/wiki/Estructura\\_de\\_datos](https://es.wikipedia.org/wiki/Estructura_de_datos)).
- [2] Elliot B. Koffman. Objects, abstraction, data structures, and design: using C++. John Wiley & Sons Inc. (2005).