ANEXO TP FUNCIONAL 2DA ENTREGA:

3.6) Punto 6:

- Al querer *cargar* y *ejecutar* el programa que suma 10 y 22 en el procesador con memoria infinita lo que visualizamos en la consola es una secuencia de números que no termina nunca, ya que *ejecutar* devuelve un microprocesador y lo estamos visualizando en la consola junto con su memoria de datos, que es infinita.

- Al tratar de ver si la memoria de ese microprocesador está ordenada, la función no nos va a devolver nada, porque se queda iterando y analizando todos los elementos de la memoria, que es infinita.

- Lo que podemos hacer con esa lista infinita es ejecutar funciones como *take* o *drop.* No podemos aplicarle funciones que tengan que analizar la lista completa, como *length.*

4) CASOS DE PRUEBA:

4.2) Prueba de los programas:

ITEM 1: > *(ejecutar.cargarPrograma suma10y22) xt8088*

Microprocesador {acumuladorA = 32, acumuladorB = 0, programCounter = 4, memoria = [0,0,0,0,.....,0], mensajeError = "", programa = [<function>,<function>,<function>,<function>]}

ITEM 2: > *(ejecutar.cargarPrograma division2por0) xt8088*

Microprocesador {acumuladorA = 2, acumuladorB = 0, programCounter = 6, memoria = [2,0,...,0], mensajeError = "DIVISION BY ZERO", programa = [<function>,<function>,<function>,<function>,<function>,<function>]}

ITEM 3: Se puede ver en el 3.3) Punto 3

4.3) Pruebas sobre IFNZ:

ITEM 1: > *ifNZ [lodv 3, swap] fp20*

Microprocesador {acumuladorA = 24, acumuladorB = 3, programCounter = 2, memoria = [], mensajeError = "", programa = []}

ITEM 2: > *ifNZ [lodv 3, swap] xt8088*

Microprocesador {acumuladorA = 0, acumuladorB = 0, programCounter = 0, memoria = [0,...,0], mensajeError = "", programa = []}

4.4) Depuración de un programa:

> *((length . depurar xt8088) [swap, nop, lodv 133, lodv 0, str 1 3, str 2 0])*
2

4.5) "Orden" de la memoria:

ITEM 1: > *memoriaOrdenada at8086*
True

ITEM 2: > *memoriaOrdenada microDesorden*
False