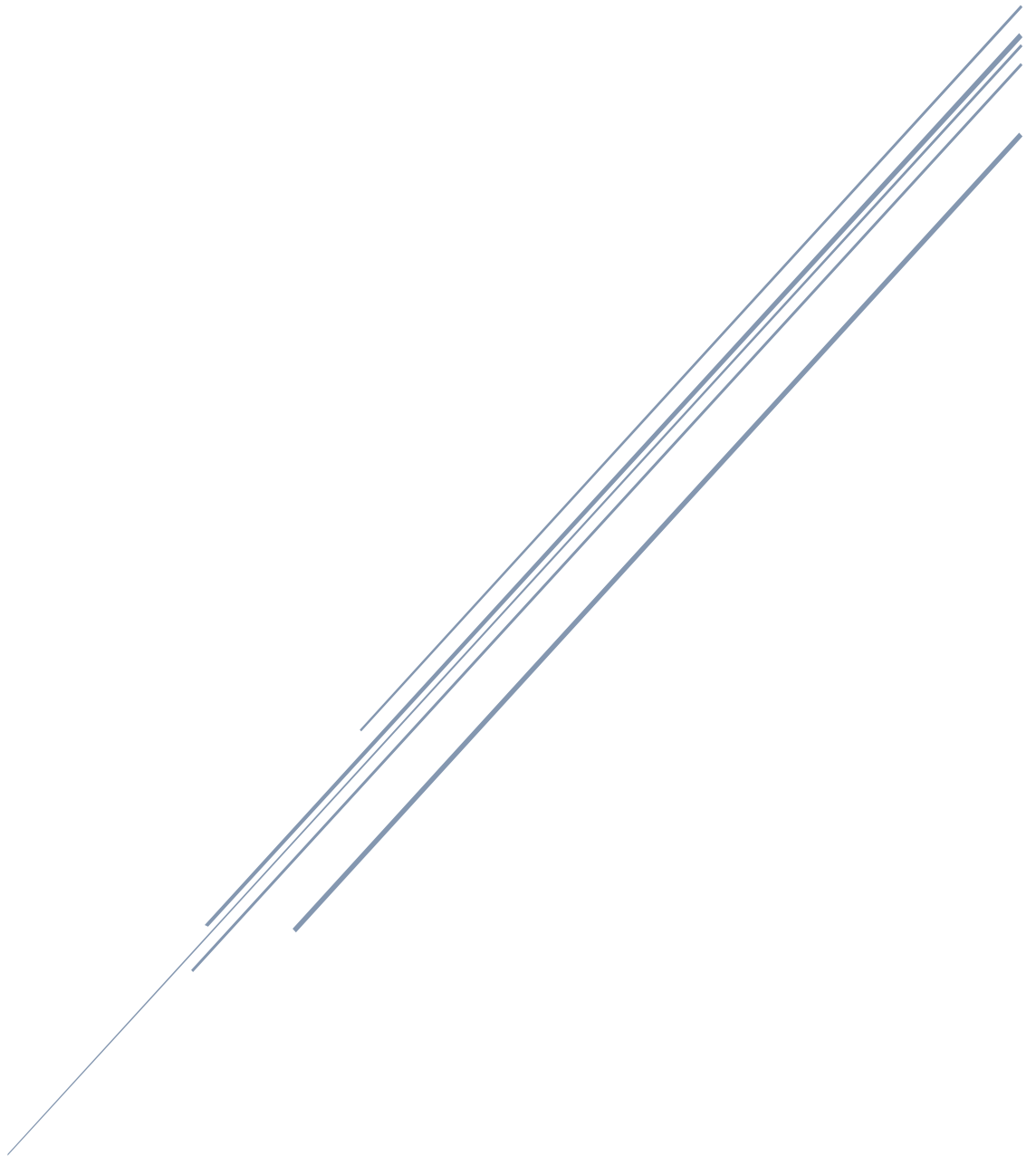


TRABAJO PRÁCTICO FINAL

Alumnos: Iván Castro, Enrique Cerioni



Trabajo Práctico N°2: Resolución de casos de simulación por computadora.

Contenido conceptual:

Etapas de desarrollo a partir de sistemas reales. Modelos lineales y no lineales. Control de aptitud de modelos. Análisis de incertidumbre y sensibilidad.

Objetivos:

- Desarrollo de software de computadora que simule sistemas reales.
- Análisis de los datos generados por software de computadora.
- Análisis de incertidumbre y sensibilidad de las variables de un modelo simulado.

Ejercicios:

Ejercicio 1: Sistema no lineal simple: el caso presa-predador.

Realizar un programa interactivo del caso presa-predador visto en prácticas anteriores, que permita estudiar las salidas del programa cuando se cambien las condiciones de contornos y parámetros de entrada. Graficar en pantalla los diagramas poblacionales y el diagrama de fase.

Deberá presentar un informe que contenga:

Una explicación de los algoritmos principales utilizados en el programa.

Pruebas de simulación:

- Una simulación con los valores por defecto: 500 liebres y 10 zorros.
- Dos simulaciones distintas cambiando las condiciones iniciales.

Para cada caso realizar una interpretación de los gráficos de comportamiento de las poblaciones, incluyendo capturas de los gráficos poblacionales y de diagrama de fase.

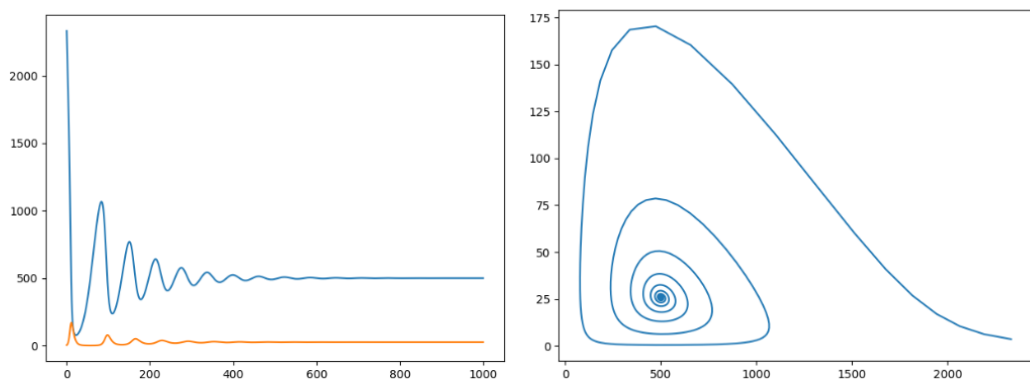
Resolución:

Tras la resolución del caso, que se realizó a través del lenguaje de programación "Python" en el cual obtuvimos los resultados de cómo varían los gráficos en el diagrama de elongación y diagrama de fase. Estos resultados se ven reflejados tras hacer una modificación en las condiciones iniciales.

En primer lugar representamos los gráficos correspondientes a los valores indicados por Pulliafito.

Condiciones iniciales:

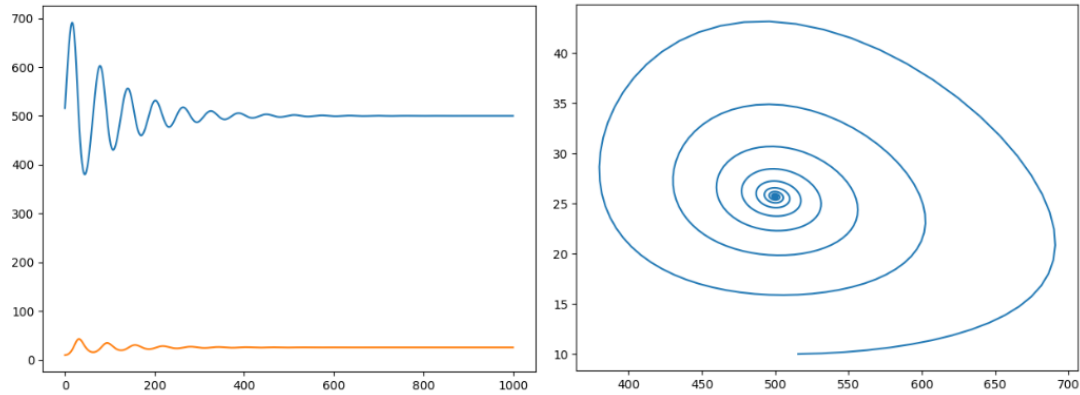
- Liebres: 2500
- Zorros: 2



En segundo lugar representamos los graficos intercambiando dos veces los valores de las condiciones iniciales como se ve muestra a continuación:

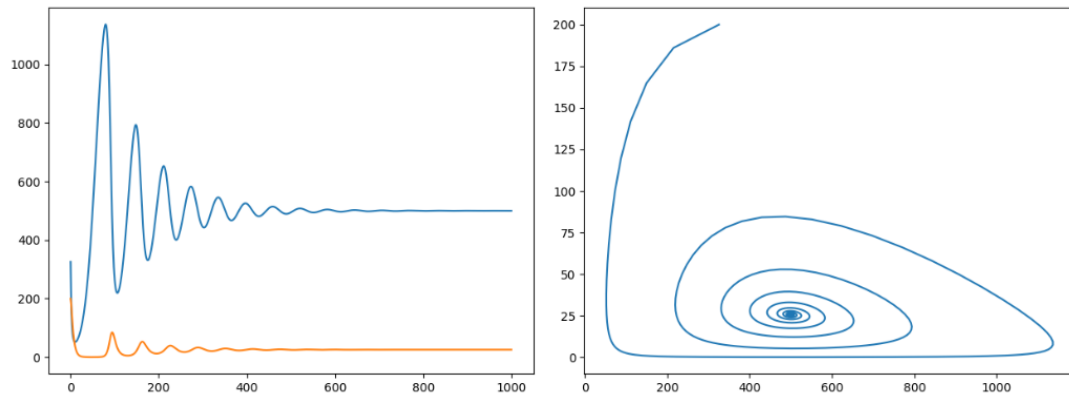
Condiciones iniciales:

- Liebres: 500
- Zorros: 10



Condiciones iniciales:

- Liebres: 500
- Zorros: 200



Ejercicio 2: El juego de la vida

El “juego” consiste en desarrollar un patrón de figuras que evolucionen de acuerdo a reglas predeterminadas, a partir de una configuración inicial y un conjunto de reglas. La competencia consiste en descubrir nuevas formas originales y calcular cuántas generaciones evoluciona el sistema antes de que se repitan o desaparezcan. Esto se realiza en una grilla de celdas que se estira al infinito en todas las direcciones. Este efecto lo lograremos considerando que la última columna de la derecha tenga por vecina a la primera columna de la izquierda, y lo mismo con la fila de abajo y de arriba. Una celda viva se marcará con un 1 o se pintará de un color gris oscuro, mientras que una celda muerta se marcará con un 0 o se pintará con un color blanco.

Las reglas son:

- “Una celda viviente sobrevive únicamente si tiene 2 o 3 celdas vecinas vivas”
- “El nacimiento de una nueva celda se da si esta tiene exactamente 3 celdas vivas vecinas”

Estas simples reglas tienen un potencial asombroso de generar patrones complejos, dependiendo del patrón inicial. En todo momento el programa debe mostrar la evolución de la grilla e ir mostrando el estado del sistema, es decir, la cantidad de celdas vivas y muertas, además del número de generaciones que han pasado hasta el momento. Se debe permitir el ingreso de celdas vivas en cualquier posición de la grilla antes de comenzar el juego. La grilla por efecto deberá tener 10 x 10. El programa se debe detener en caso de que se hayan muerto todas las celdas.

Resolución:

En este ejercicio, realizamos el juego de la vida en el lenguaje de programación C, con sus respectivas reglas, en el cual al iniciar ingresamos el número de filas y columnas que deseamos elegir para imprimir los patrones cargados. Una vez indicadas las filas y columnas procedemos a seleccionar un patrón a través de un Switch.

```

<-- JUEGO DE LA VIDA -->

Ingrese un numero de filas mayor o igual a 20:
20

Ingrese un numero de columnas mayor o igual a 20:
20

Selecciona el estado inicial:
1)figura
2)parpadeador
3)sapo
4)panal
5)nave ligera

```

Como ejemplo representamos el patrón del “sapo” a continuación:



Ejercicio 3: El método de Montecarlo

Una compañía de productos alimenticios necesita tomar decisiones estratégicas de ventas futuras, y uno de los datos que requiere es la estimación de visitas a su sitio web para los próximos 100 días.

El sysadmin de la empresa nos informa que el servidor web que aloja el sitio es propio, y que además cuenta con una consola de monitoreo, desde hace 250 días, que puede brindarnos información histórica de las visitas al sitio.

Se le solicita que desarrolle un software que, tomando los 250 datos históricos de visitas diarias al sitio web, y sabiendo que los mismos se distribuyen normalmente, permita estimar los datos futuros para los próximos 100 días.

Deberá presentar:

- El código fuente del software desarrollado en algún lenguaje de programación de su elección.
- La explicación del algoritmo principal.
- Los cálculos de las medidas estadísticas (media y varianza), tanto de los datos históricos proporcionados por la consola de monitoreo, como de los datos generados por su software.

La siguiente tabla representa las visitas al sitio web durante los últimos 250 días.

Día	Visitas	Día	Visitas	Día	Visitas	Día	Visitas	Día	Visitas	Día	Visitas	Día	Visitas	Día	Visitas
1	33	32	22	63	24	95	24	126	23	157	21	188	27	219	34
2	18	33	37	64	18	96	35	127	40	158	16	189	10	220	32
3	27	34	15	65	19	97	12	128	25	159	10	190	32	221	24
4	36	35	29	66	43	98	19	129	18	160	23	191	13	222	30
5	25	36	11	67	9	99	17	130	16	161	26	192	29	223	26
6	24	37	36	68	37	100	21	131	20	162	27	193	27	224	28
7	30	38	24	69	32	101	28	132	23	163	29	194	22	225	29
8	23	39	18	70	26	102	14	133	20	164	21	195	17	226	19
9	29	40	19	71	25	103	22	134	31	165	13	196	25	227	21
10	38	41	23	72	18	104	39	135	23	166	32	197	33	228	21
11	23	42	12	73	16	105	27	136	20	167	22	198	46	229	30
12	22	43	21	74	14	106	18	137	38	168	27	199	17	230	26
13	19	44	7	75	23	107	33	138	30	169	28	200	35	231	25
14	21	45	30	76	37	108	39	139	19	170	35	201	17	232	18
15	18	46	6	77	31	109	30	140	15	171	24	202	20	233	22
16	28	47	26	78	34	110	32	141	6	172	16	203	22	234	21
17	12	48	31	79	21	111	26	142	23	173	14	204	30	235	37
18	20	49	24	80	22	112	32	143	23	174	25	205	31	236	24
19	22	50	31	81	26	113	19	144	21	175	32	206	27	237	25
20	22	51	25	82	16	114	35	145	17	176	22	207	21	238	35
21	22	52	21	83	24	115	26	146	27	177	32	208	32	239	18
22	34	53	25	85	25	116	20	147	19	178	24	209	18	240	16
23	32	54	21	86	32	117	23	148	23	179	33	210	19	241	29
24	15	55	24	87	11	118	23	149	44	180	32	211	25	242	25
25	14	56	30	88	30	119	23	150	19	181	32	212	26	243	20
26	16	57	29	89	30	120	23	151	23	182	25	213	20	244	26
27	32	58	18	90	17	121	11	152	20	183	24	214	33	245	28
28	17	59	19	91	18	122	20	153	24	184	28	215	19	246	32
29	22	60	16	92	36	123	15	154	22	185	23	216	30	247	16
30	37	61	17	93	14	124	19	155	18	186	16	217	11	248	15
31	11	62	20	94	31	125	33	156	23	187	27	218	34	249	42
														250	29

Resolución:

Para la resolución de este ejercicio, tuvimos que optar en realizar una planilla de Excel para poder entender mejor el problema. Luego resolvimos el problema en el lenguaje de programación “Python”. El código consiste en lo siguiente:

Creamos una función en la cual generamos dos variables (media y varianza) para los valores que se van a generar, y utilizamos la función **“random.uniform(0 , 1)”** para generar valores entre 0 y 1.

Luego dentro del **“while”** creamos un **“for”** que lee toda la lista, donde encuentra a que intervalo pertenece a la frecuencia acumulada conseguida. Se actualiza la media y se agrega el valor a una lista **“aux”**.

Por último, para la representación gráfica del problema utilizamos una función llamada **“hist”** para representar los valores en un histograma:

