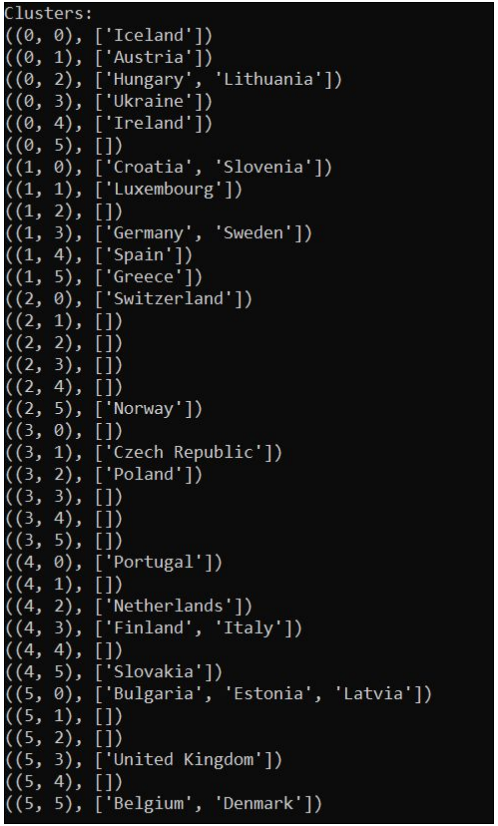
Redes Neuronales: Aprendizaje *no* Supervisado

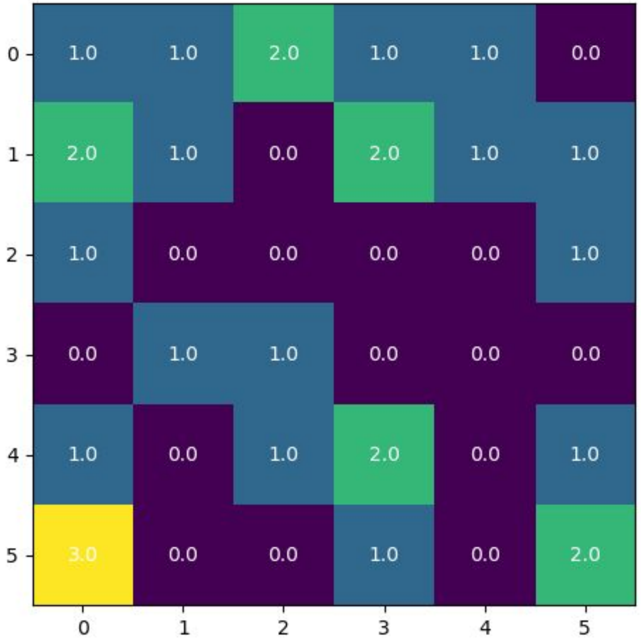
Grupo 6: Katan, Paganini

Ejercicio 1.a
Kohonen

Agrupamientos generados



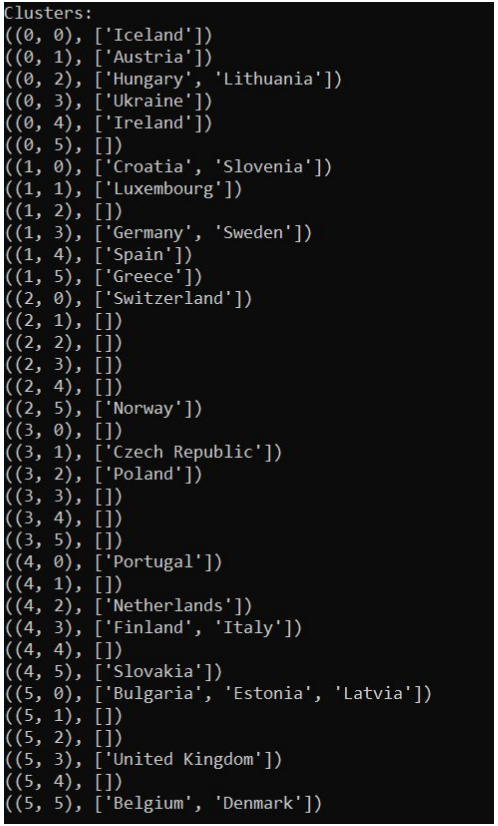
```
Clusters:
((0, 0), ['Iceland'])
((0, 1), ['Austria'])
((0, 2), ['Hungary', 'Lithuania'])
((0, 3), ['Ukraine'])
((0, 4), ['Ireland'])
((0, 5), [])
((1, 0), ['Croatia', 'Slovenia'])
((1, 1), ['Luxembourg'])
((1, 2), [])
((1, 3), ['Germany', 'Sweden'])
((1, 4), ['Spain'])
((1, 5), ['Greece'])
((2, 0), ['Switzerland'])
((2, 1), [])
((2, 2), [])
((2, 3), [])
((2, 4), [])
((2, 5), ['Norway'])
((3, 0), [])
((3, 1), ['Czech Republic'])
((3, 2), ['Poland'])
((3, 3), [])
((3, 4), [])
((3, 5), [])
((4, 0), ['Portugal'])
((4, 1), [])
((4, 2), ['Netherlands'])
((4, 3), ['Finland', 'Italy'])
((4, 4), [])
((4, 5), ['Slovakia'])
((5, 0), ['Bulgaria', 'Estonia', 'Latvia'])
((5, 1), [])
((5, 2), [])
((5, 3), ['United Kingdom'])
((5, 4), [])
((5, 5), ['Belgium', 'Denmark'])
```



- Iteraciones: 20000
- Datos estandarizados
- 6x6 neuronas
- eta(t) = 1/t

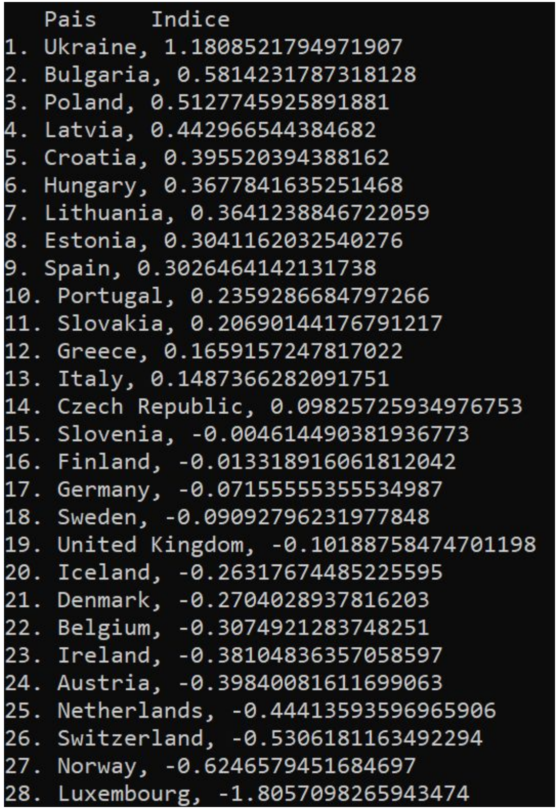
Analizando qué países fueron agrupados

Agrupamientos generados usando Kohonen



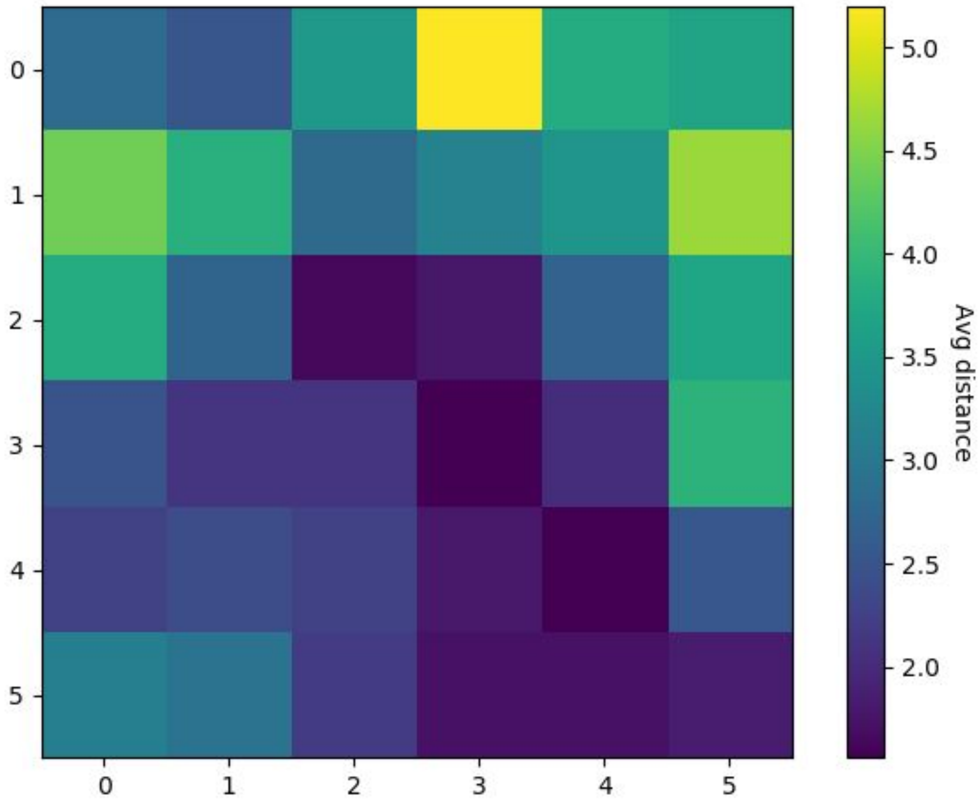
```
Clusters:
((0, 0), ['Iceland'])
((0, 1), ['Austria'])
((0, 2), ['Hungary', 'Lithuania'])
((0, 3), ['Ukraine'])
((0, 4), ['Ireland'])
((0, 5), [])
((1, 0), ['Croatia', 'Slovenia'])
((1, 1), ['Luxembourg'])
((1, 2), [])
((1, 3), ['Germany', 'Sweden'])
((1, 4), ['Spain'])
((1, 5), ['Greece'])
((2, 0), ['Switzerland'])
((2, 1), [])
((2, 2), [])
((2, 3), [])
((2, 4), [])
((2, 5), ['Norway'])
((3, 0), [])
((3, 1), ['Czech Republic'])
((3, 2), ['Poland'])
((3, 3), [])
((3, 4), [])
((3, 5), [])
((4, 0), ['Portugal'])
((4, 1), [])
((4, 2), ['Netherlands'])
((4, 3), ['Finland', 'Italy'])
((4, 4), [])
((4, 5), ['Slovakia'])
((5, 0), ['Bulgaria', 'Estonia', 'Latvia'])
((5, 1), [])
((5, 2), [])
((5, 3), ['United Kingdom'])
((5, 4), [])
((5, 5), ['Belgium', 'Denmark'])
```

Tabla generada usando PCA



```
   Pais    Indice
1. Ukraine, 1.1808521794971907
2. Bulgaria, 0.5814231787318128
3. Poland, 0.5127745925891881
4. Latvia, 0.442966544384682
5. Croatia, 0.395520394388162
6. Hungary, 0.3677841635251468
7. Lithuania, 0.3641238846722059
8. Estonia, 0.3041162032540276
9. Spain, 0.3026464142131738
10. Portugal, 0.2359286684797266
11. Slovakia, 0.20690144176791217
12. Greece, 0.1659157247817022
13. Italy, 0.1487366282091751
14. Czech Republic, 0.09825725934976753
15. Slovenia, -0.004614490381936773
16. Finland, -0.013318916061812042
17. Germany, -0.0715555355534987
18. Sweden, -0.09092796231977848
19. United Kingdom, -0.10188758474701198
20. Iceland, -0.2631767448522595
21. Denmark, -0.2704028937816203
22. Belgium, -0.3074921283748251
23. Ireland, -0.38104836357058597
24. Austria, -0.39840081611699063
25. Netherlands, -0.44413593596965906
26. Switzerland, -0.5306181163492294
27. Norway, -0.6246579451684697
28. Luxembourg, -1.8057098265943474
```

Distancias promedio entre neuronas vecinas

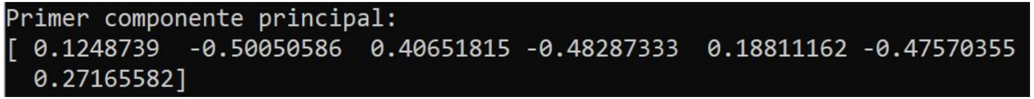


- Para cada neurona, se calculó la distancia promedio de sus pesos con los de sus 4 neuronas vecinas

Ejercicio 1.b
Oja

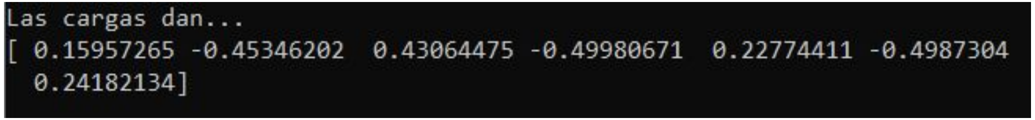
Resultados obtenidos para la primer componente

Usando PCA

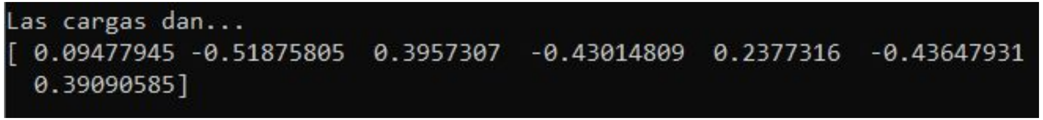


```
Primer componente principal:
[ 0.1248739  -0.50050586  0.40651815 -0.48287333  0.18811162 -0.47570355
  0.27165582]
```

Usando un perceptrón lineal, con la regla de Oja, y 7 pesos (múltiples intentos)



```
Las cargas dan...
[ 0.15957265 -0.45346202  0.43064475 -0.49980671  0.22774411 -0.4987304
  0.24182134]
```

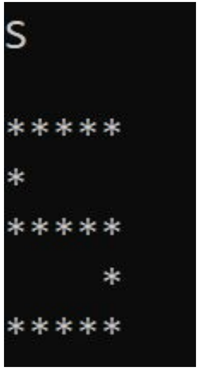


```
Las cargas dan...
[ 0.09477945 -0.51875805  0.3957307  -0.43014809  0.2377316  -0.43647931
  0.39090585]
```

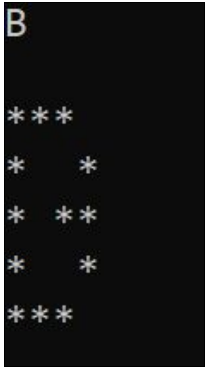
- Épocas: 500
- Datos estandarizados
- eta(t+1) = eta(t) / (t+1)
- eta(0) = 0.13

Ejercicio 2
Hopfield

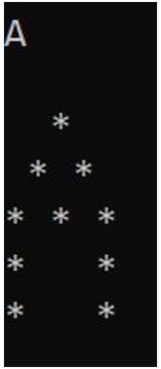
Patrones de letras almacenados



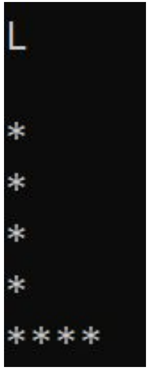
```
S

*****
*
*****
    *
*****
```



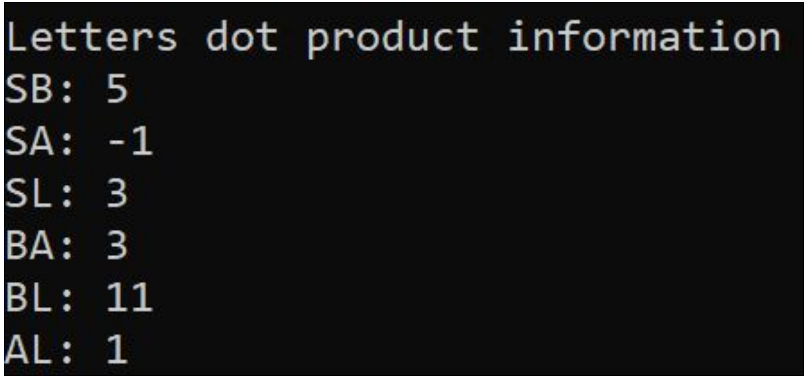
```
B

***
*  *
* **
*  *
***
```



```
A

  *
 * *
* * *
*   *
*   *
```



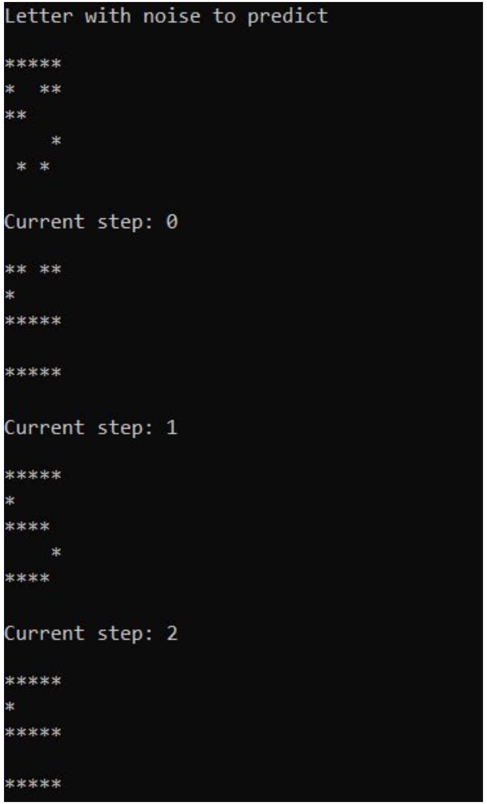
```
L

*
*
*
*
****
```



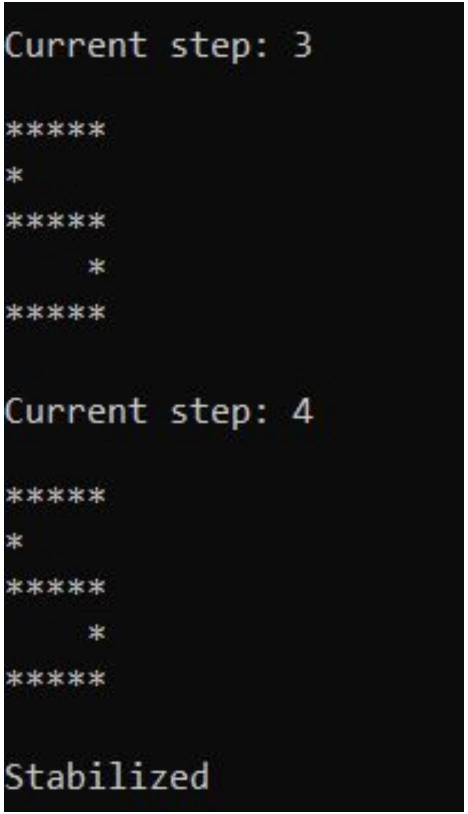
```
Letters dot product information
SB: 5
SA: -1
SL: 3
BA: 3
BL: 11
AL: 1
```

- Para generar un patrón de consulta a partir de una de las letras, se alteraron 10 coordenadas aleatorias del vector que representa la letra original, invirtiendo el valor en dichas coordenadas

Predicciones de patrones con ruido

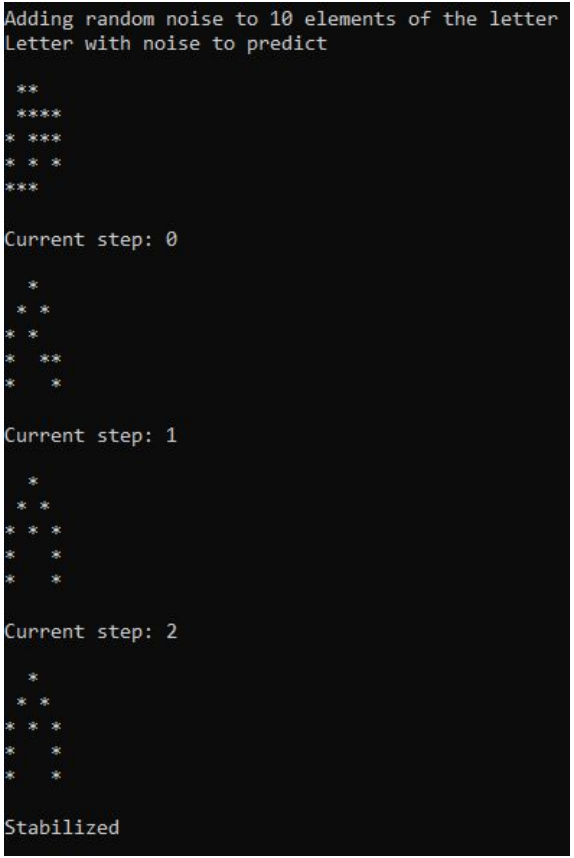


```
Letter with noise to predict

*****
*  **
**
    *
 * *

Current step: 0

** **
*
*****

*****

Current step: 1

*****
*
****
    *
****

Current step: 2

*****
*
*****

*****
```

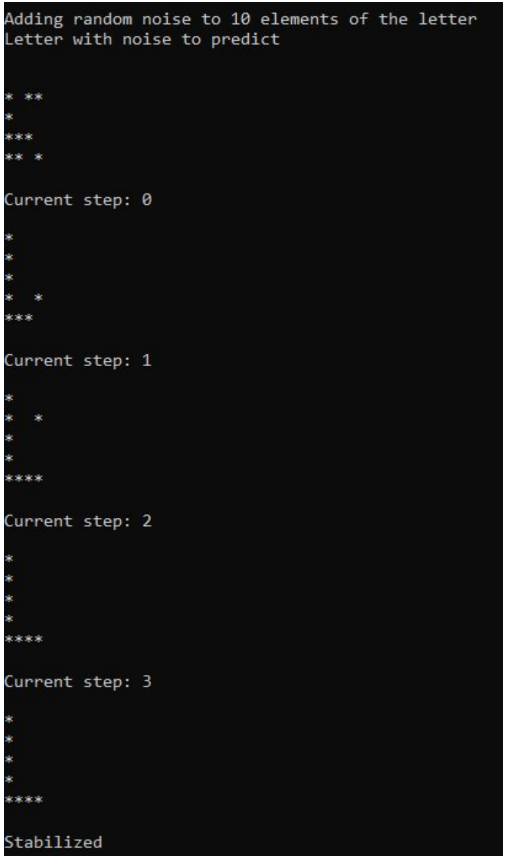


```
Current step: 3

*****
*
*****
    *
*****

Current step: 4

*****
*
*****
    *
*****

Stabilized
```

Predicciones de patrones con ruido

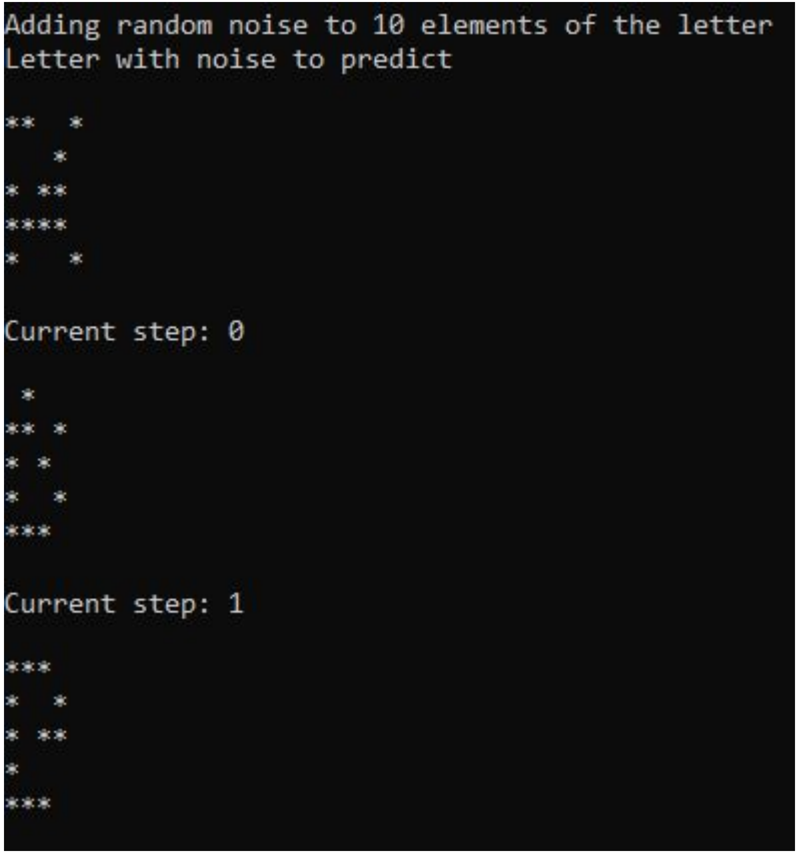


```
Adding random noise to 10 elements of the letter
Letter with noise to predict

 **
 ****
* ***
* * *
***

Current step: 0

  *
 * *
* *
*  **
*   *

Current step: 1

  *
 * *
* * *
*   *
*   *

Current step: 2

  *
 * *
* * *
*   *
*   *

Stabilized
```

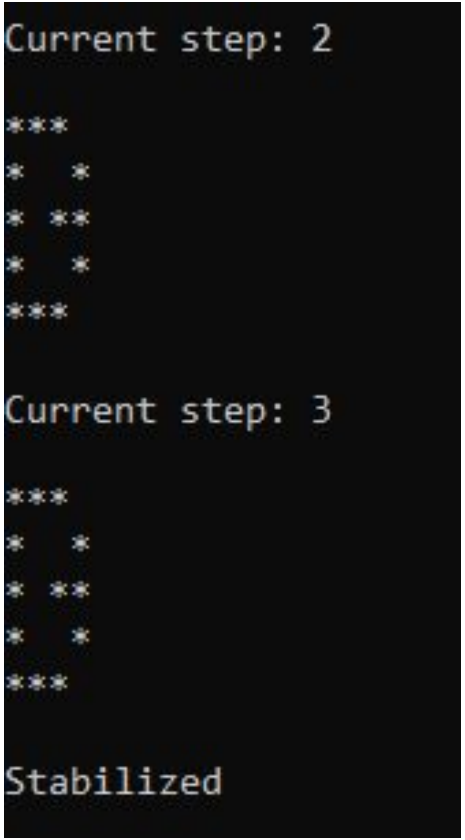


```
Adding random noise to 10 elements of the letter
Letter with noise to predict


* **
*
***
** *

Current step: 0

*
*
*
*  *
***

Current step: 1

*
*  *
*
*
****

Current step: 2

*
*
*
*
****

Current step: 3

*
*
*
*
****

Stabilized
```

Predicciones de patrones con ruido

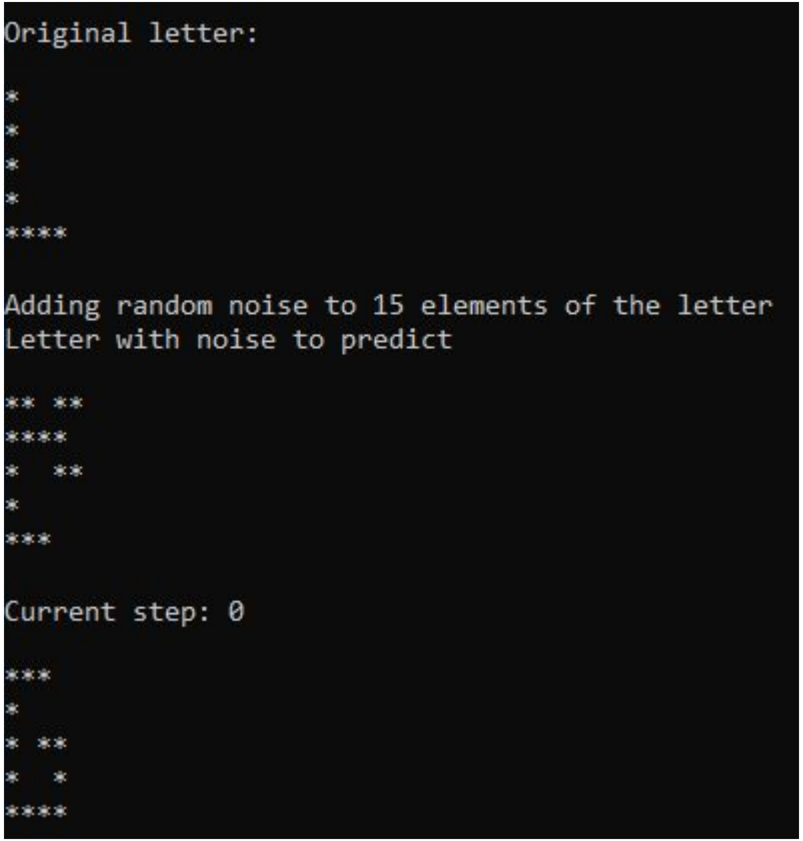


```
Adding random noise to 10 elements of the letter
Letter with noise to predict

**  *
   *
* **
****
*   *

Current step: 0

 *
** *
* *
*  *
***

Current step: 1

***
*  *
* **
*
***
```

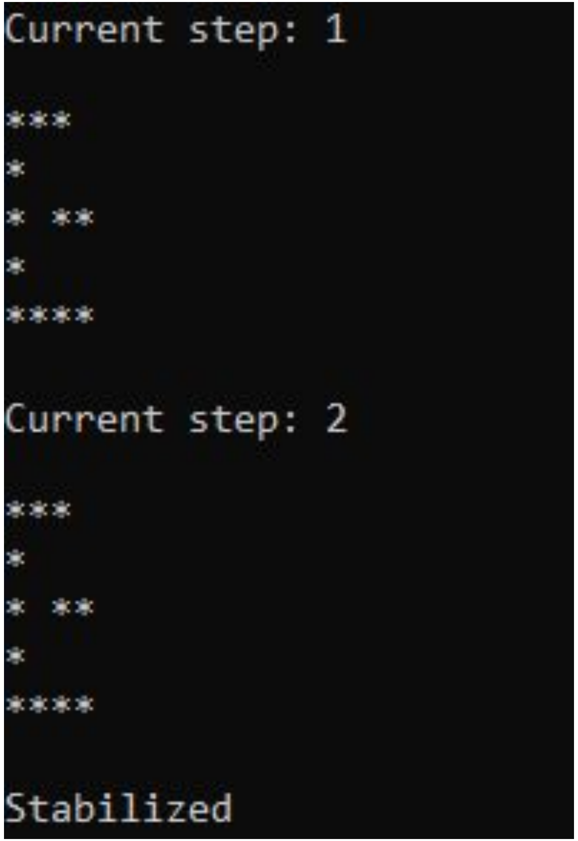


```
Current step: 2

***
*  *
* **
*  *
***

Current step: 3

***
*  *
* **
*  *
***

Stabilized
```

Identificando estados espúreos

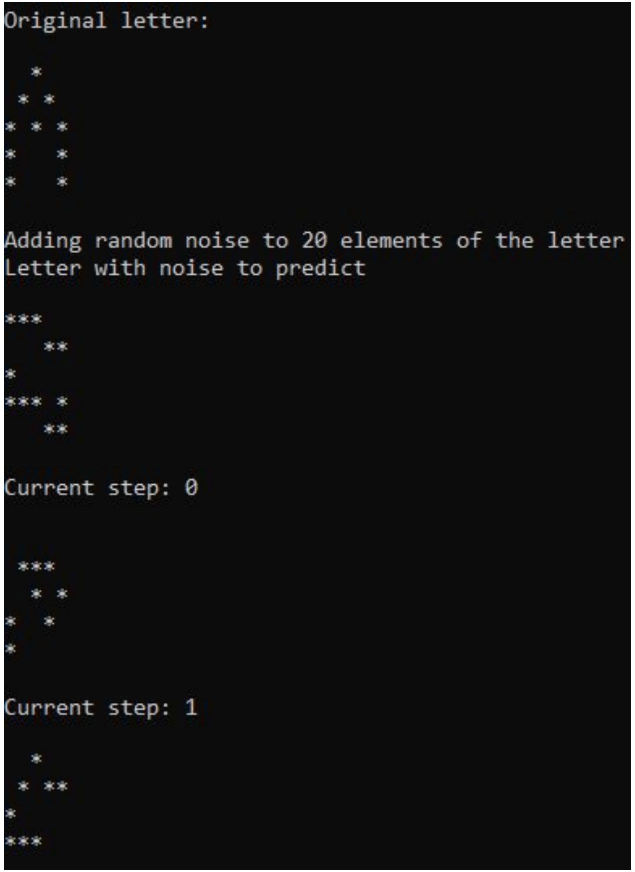


```
Original letter:

*
*
*
*
****

Adding random noise to 15 elements of the letter
Letter with noise to predict

** **
****
*  **
*
***

Current step: 0

***
*
* **
*  *
****
```

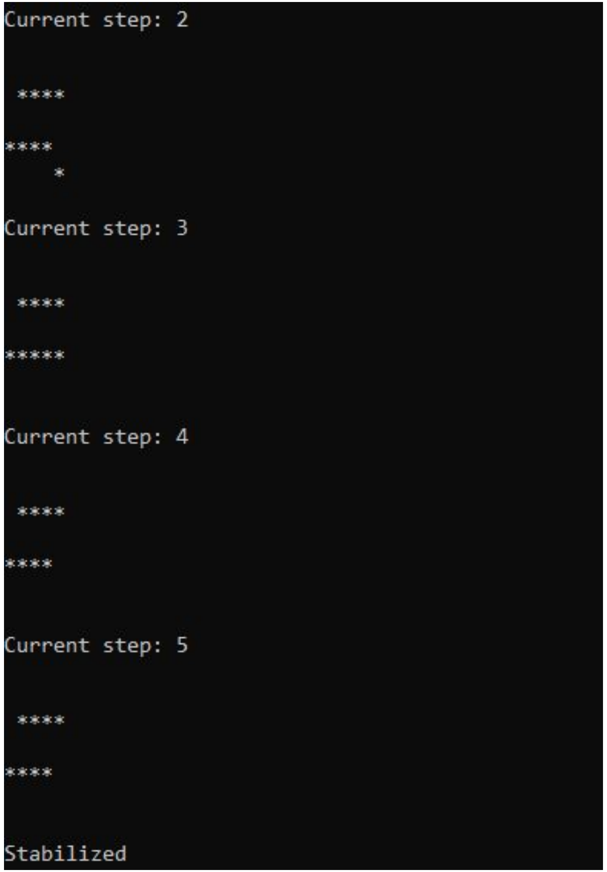


```
Current step: 1

***
*
* **
*
****


Current step: 2

***
*
* **
*
****


Stabilized
```

Identificando estados espúreos

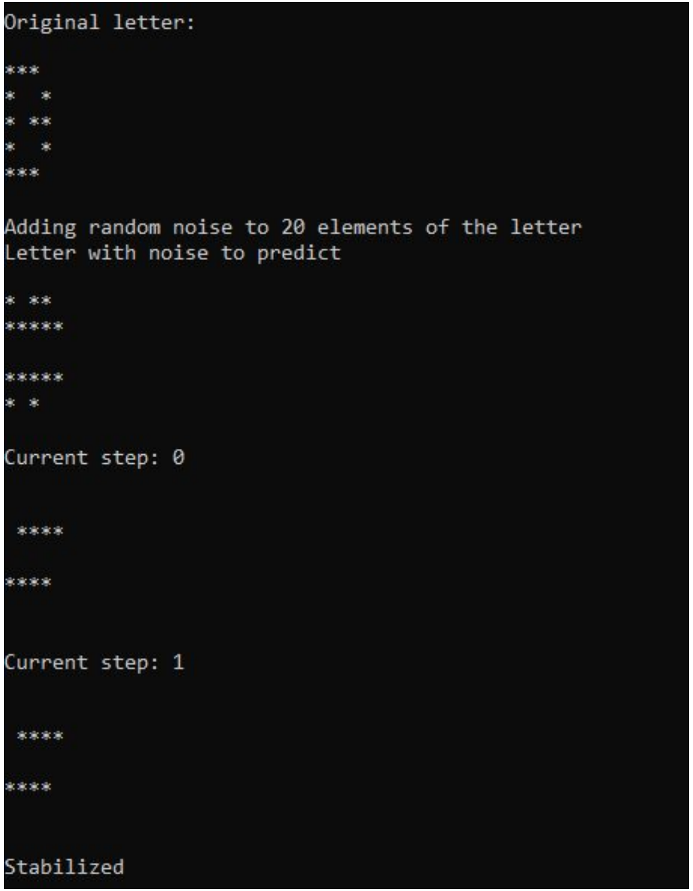


```
Original letter:

  *
 * *
* * *
*   *
*   *

Adding random noise to 20 elements of the letter
Letter with noise to predict

***
   **
*
*** *
   **

Current step: 0


 ***
  * *
*  *
*

Current step: 1

  *
 * **
*
***
```



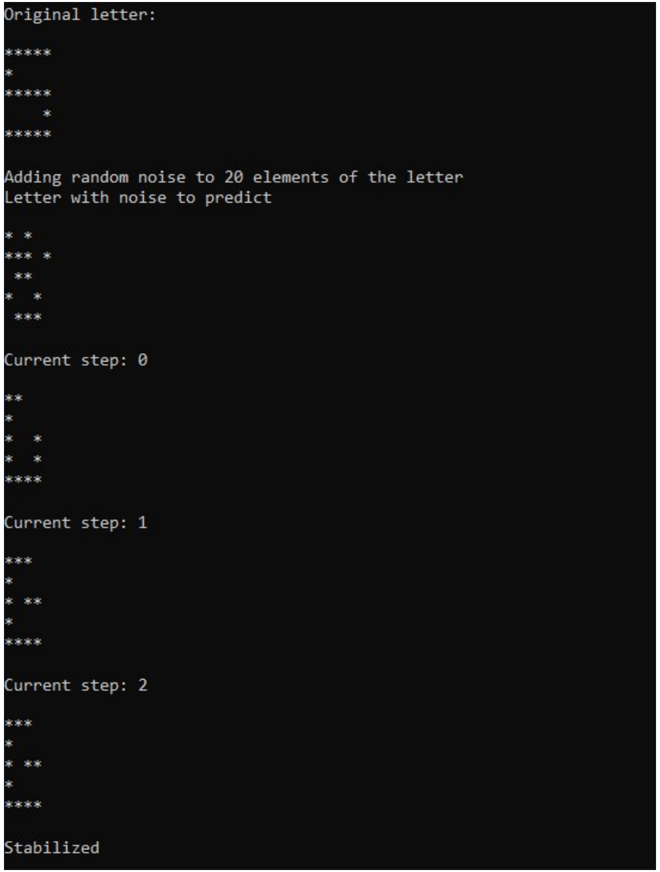
```
Current step: 2


 ****

****
    *

Current step: 3


 ****

*****


Current step: 4


 ****

****


Current step: 5


 ****

****


Stabilized
```

Identificando estados espúreos



```
Original letter:

***
*  *
* **
*  *
***

Adding random noise to 20 elements of the letter
Letter with noise to predict

* **
*****

*****
* *

Current step: 0


 ****

****


Current step: 1


 ****

****


Stabilized
```

Identificando estados espúreos



```
Original letter:

*****
*
*****
    *
*****

Adding random noise to 20 elements of the letter
Letter with noise to predict

* *
*** *
 **
*  *
 ***

Current step: 0

**
*
*  *
*  *
****

Current step: 1

***
*
* **
*
****

Current step: 2

***
*
* **
*
****

Stabilized
```