



**Tecnológico
de Monterrey**

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Querétaro

TC2008B. Modelación de sistemas multiagentes con gráficas computacionales

Manejo de Tiempo en Unity

Profesores:

Pedro Oscar Pérez Murueta

Denisse Lizbeth Maldonado Flores

Alejandro Fernández Vilchis

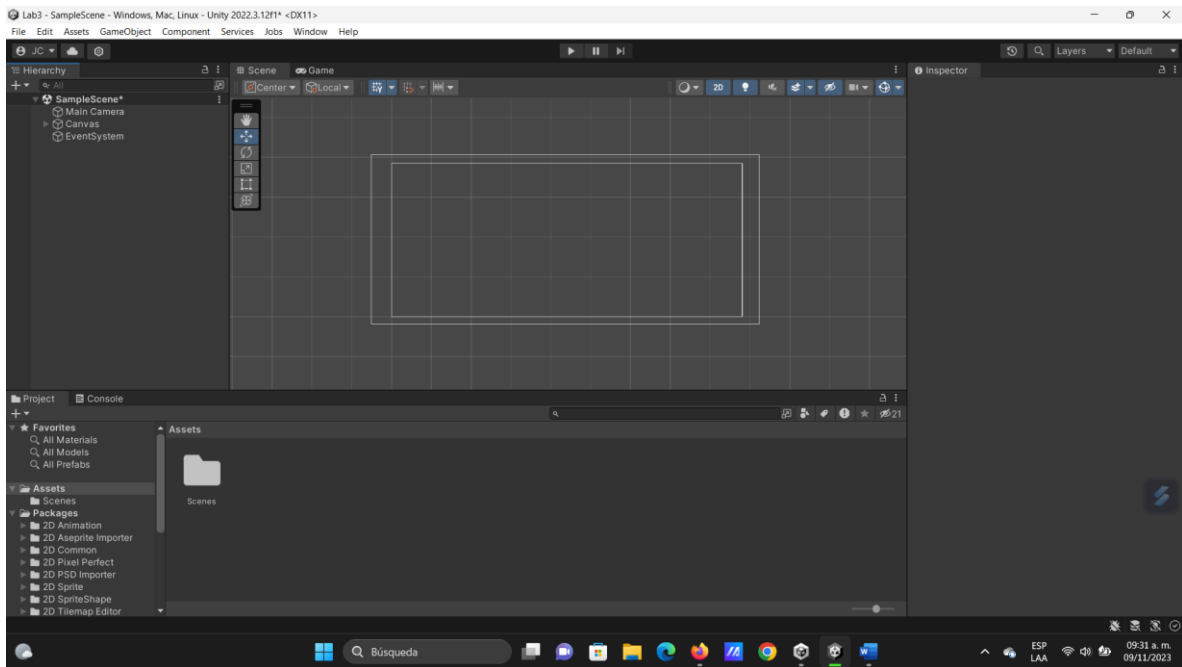
Presenta:

José Emiliano Riosmena Castañón – A01704245

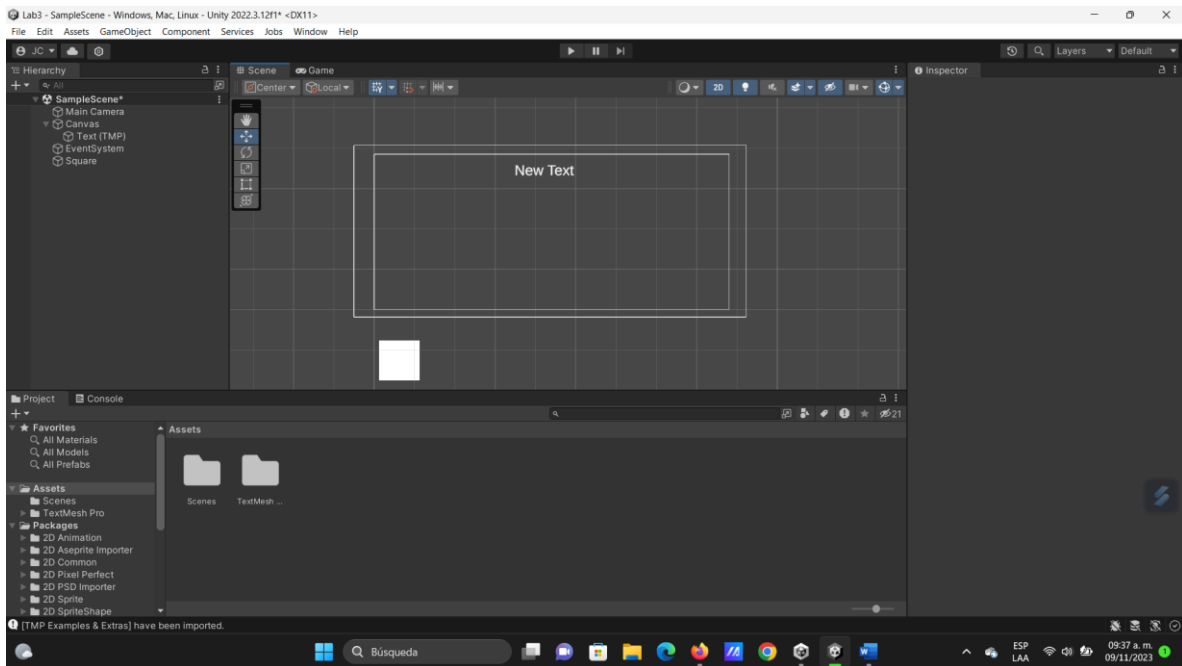
Fecha:

Jueves, 9 de noviembre del 2023

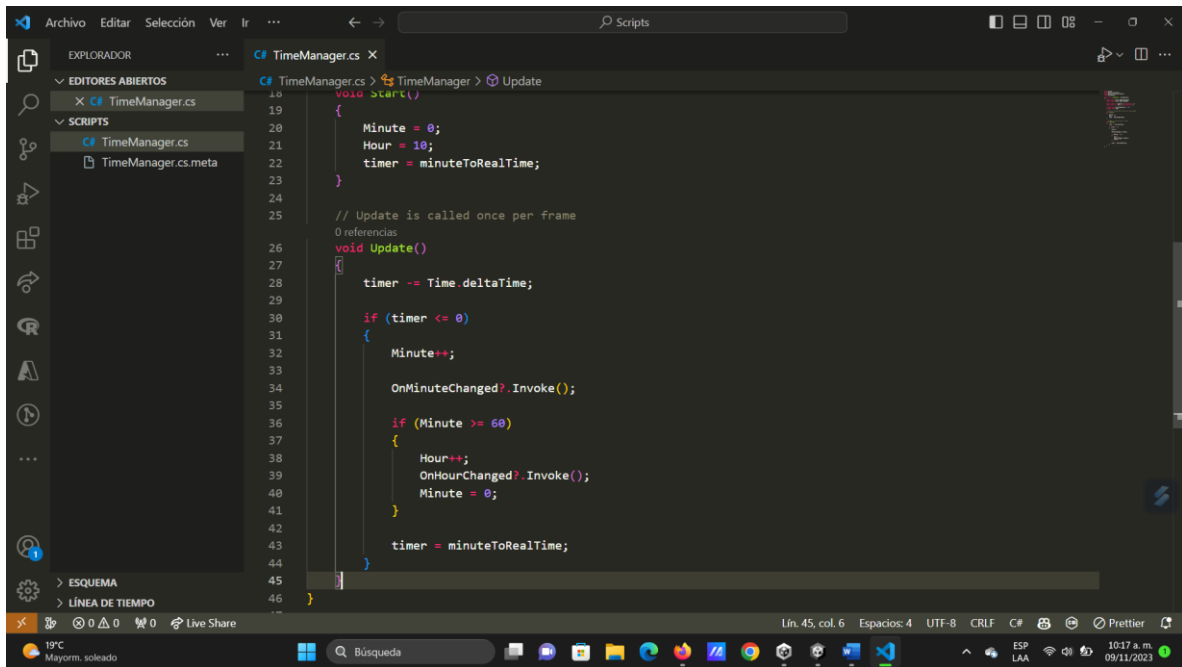
Paso 1



Paso 2



Paso 3

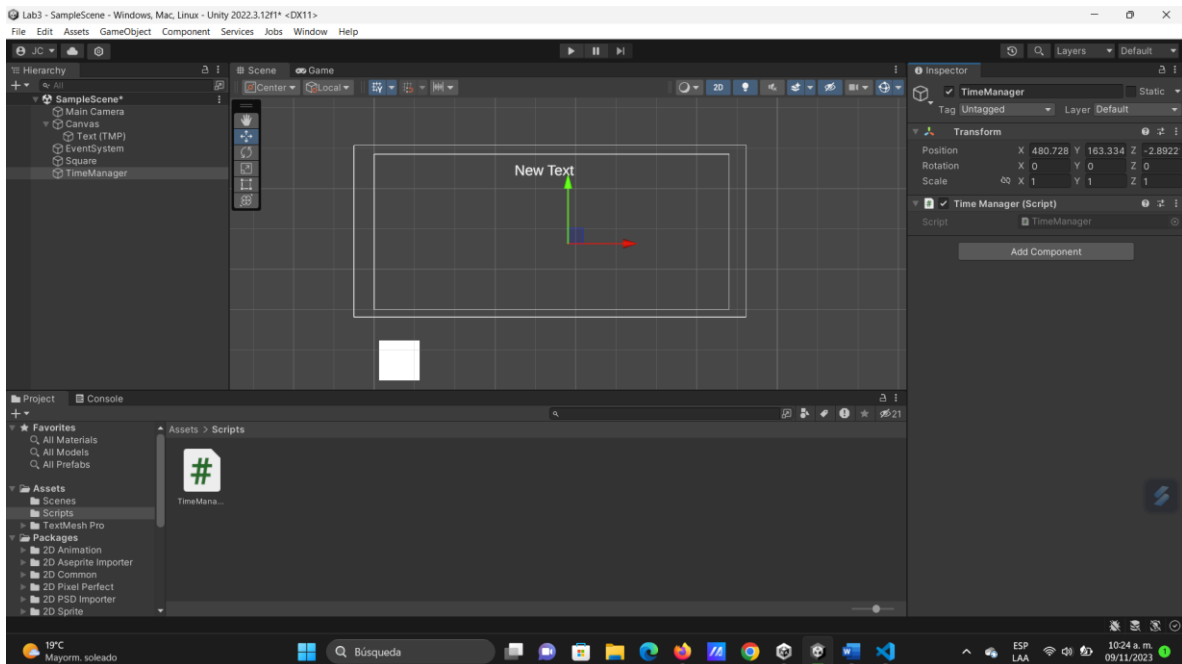


The screenshot shows the Visual Studio Code editor with the `TimeManager.cs` script open. The script is located in the `SCRIPTS` folder. The code is as follows:

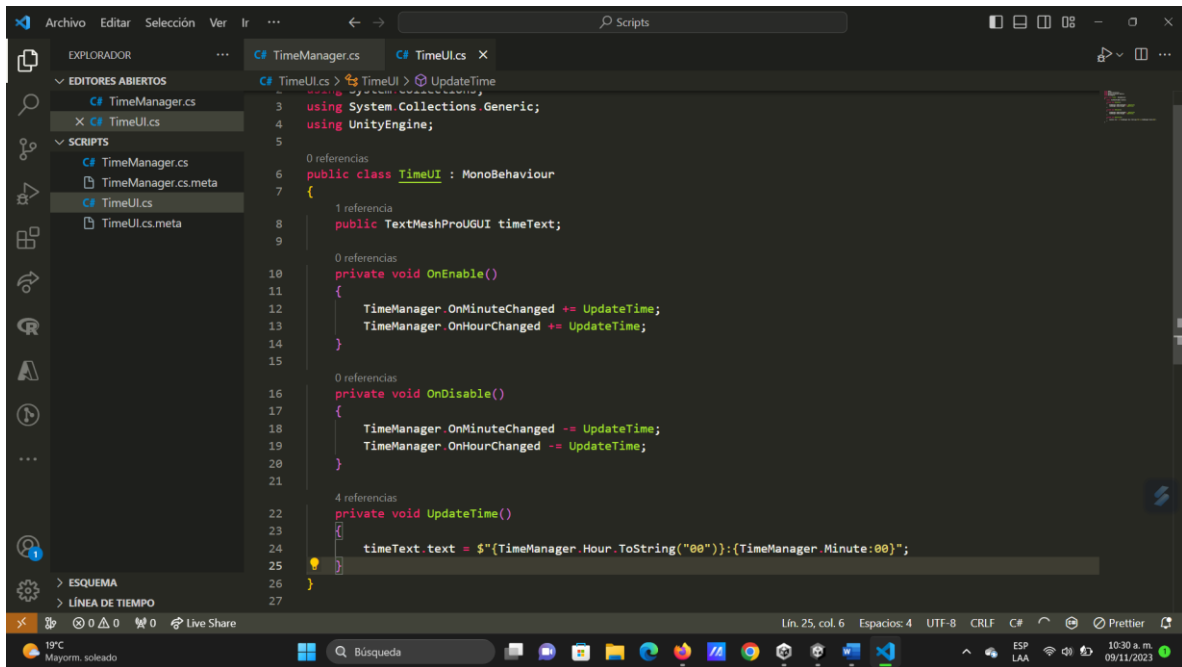
```
18 void Start()
19 {
20     Minute = 0;
21     Hour = 10;
22     timer = minuteToRealTime;
23 }
24
25 // Update is called once per frame
26 void Update()
27 {
28     timer -= Time.deltaTime;
29
30     if (timer <= 0)
31     {
32         Minute++;
33
34         OnMinuteChanged?.Invoke();
35
36         if (Minute >= 60)
37         {
38             Hour++;
39             OnHourChanged?.Invoke();
40             Minute = 0;
41         }
42
43         timer = minuteToRealTime;
44     }
45 }
```

The script is named `TimeManager` and is attached to a `TimeManager` component. The `Start` method initializes `Minute` to 0, `Hour` to 10, and `timer` to `minuteToRealTime`. The `Update` method decreases `timer` by `Time.deltaTime` each frame. If `timer` is less than or equal to 0, it increments `Minute` and calls `OnMinuteChanged`. If `Minute` is greater than or equal to 60, it increments `Hour` and calls `OnHourChanged`, then resets `Minute` to 0. Finally, it resets `timer` to `minuteToRealTime`.

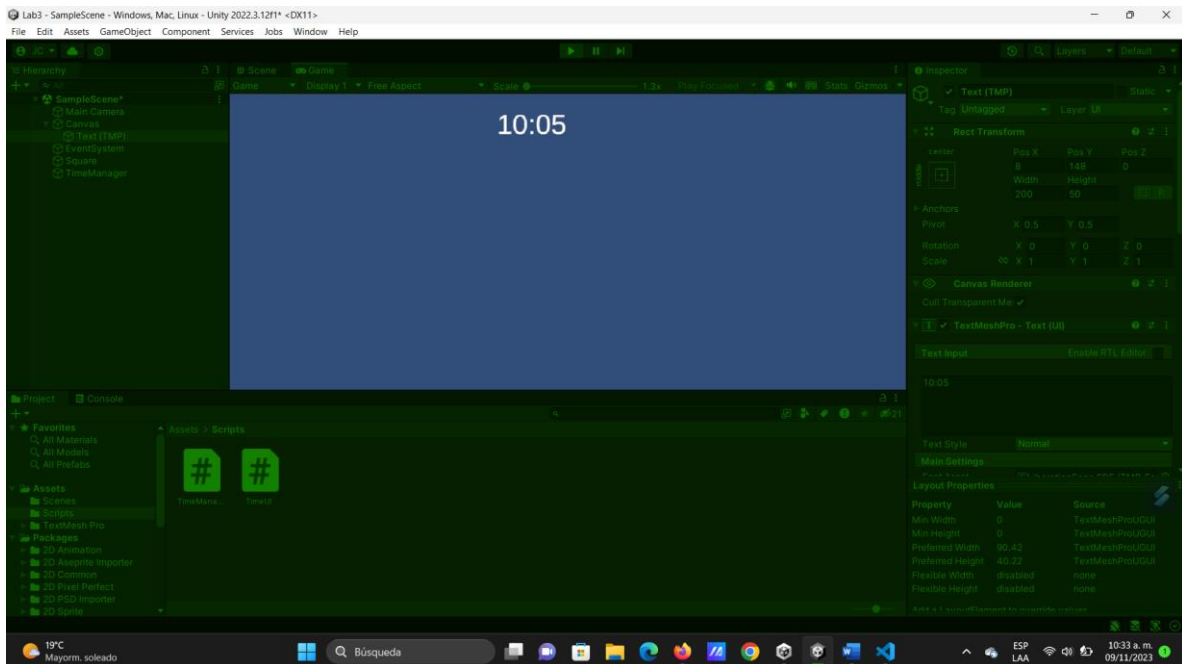
Paso 4



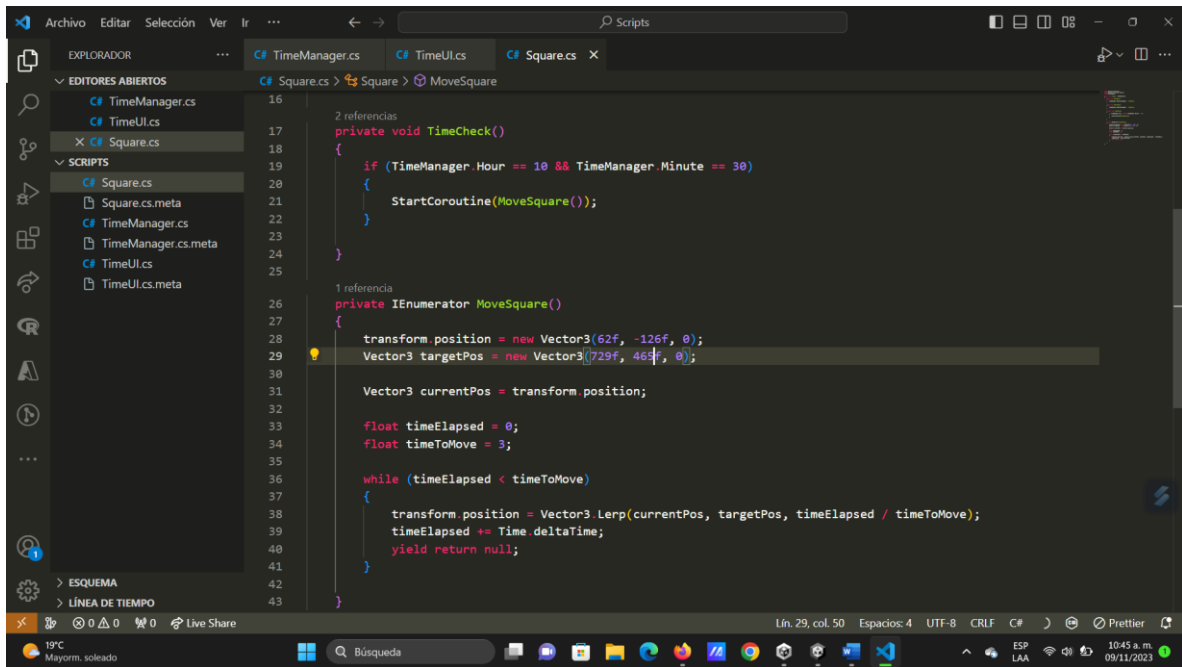
Paso 5



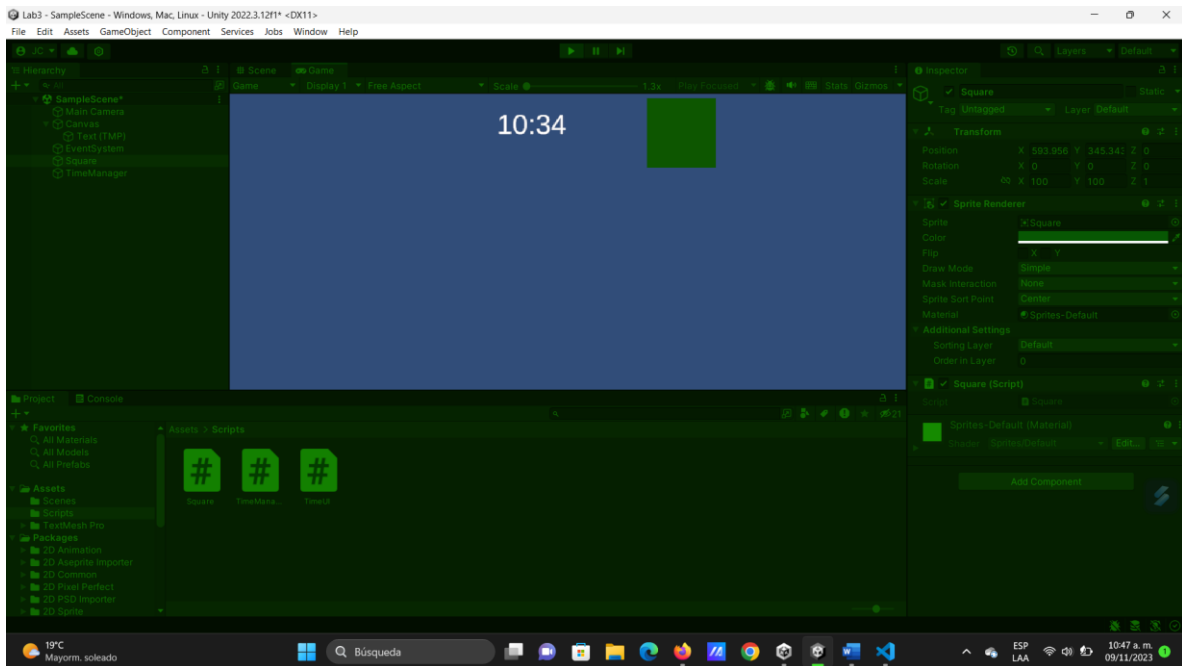
Paso 6



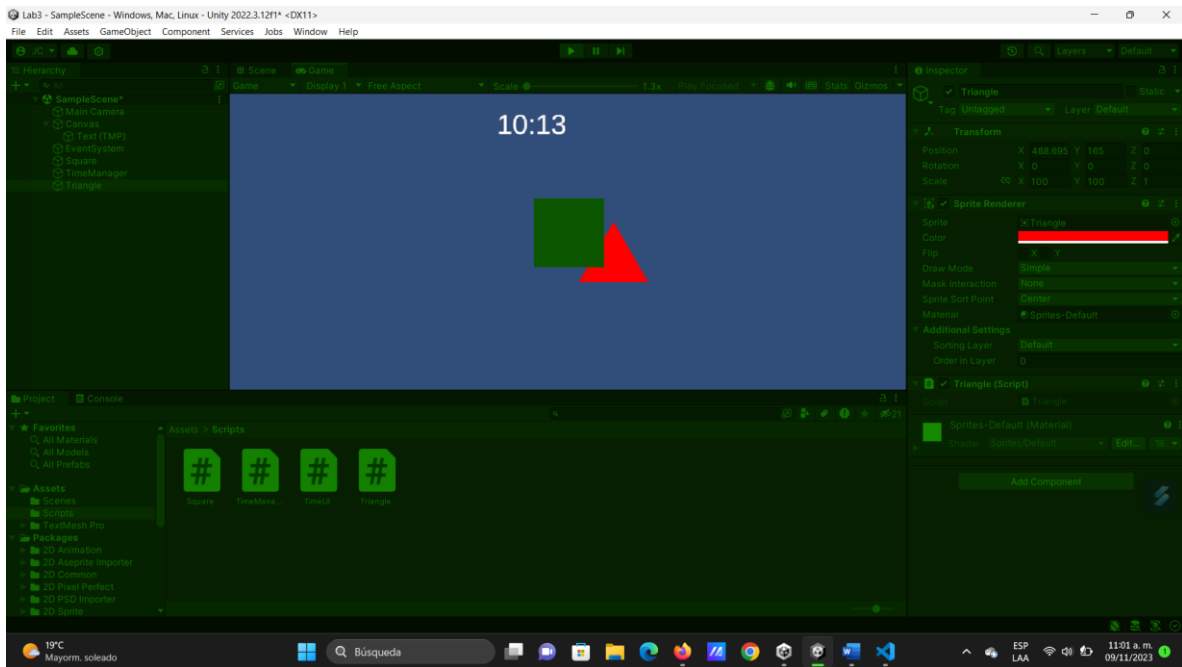
Paso 7



Paso 8



Paso 9



Liga al video

https://youtu.be/iGeiYgOBM_E

Reflexión Lab 3

Este lab ha sido muy interesante pues nos ha permitido trabajar haciendo que los objetos interactuaran en cada tiempo, y esto nos podrá ser muy útil para el laboratorio, pues nos podría permitir construirlo de modo que podamos hacer que la comida aparezca cada cierto tiempo, o que los agentes se muevan cada cierto tiempo, o por ejemplo para permitir que el almacén, se habilite cada cierto tiempo o también permitir que cada cierto tiempo, el almacén se inhabilite. También podemos hacer que la comida este disponible por cierto tiempo y después desaparece.