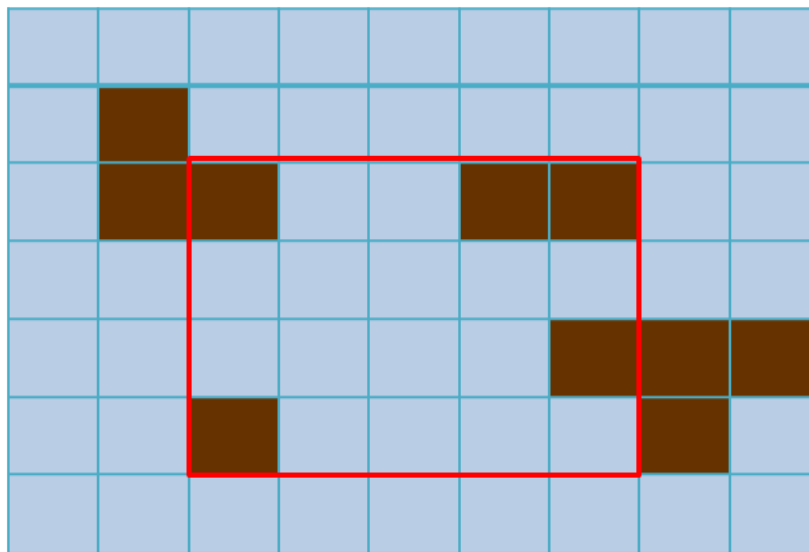


Cercando Islas

En las Islas Galápagos se encuentra uno de los ecosistemas más rico del planeta Tierra. Quizás la especie más destacada aquí es la tortuga Galápagos. Esta especie está en peligro de extinción y con el objetivo de protegerla se ha decidido construir una “cerca” marítima rectangular alrededor de estas islas. Para ello se han tomado imágenes satelitales con información de donde se encuentran dichas islas. Se desea también ahorrar cercado, lo que equivale a encontrar el rectángulo de menor área, de forma tal que todas las islas tengan al menos un pedazo de área dentro del rectángulo.

Figura 1. Mapa de las islas con el cercado mínimo.



Note que es suficiente que esté solo un “pedazo” de cada isla. Para tratar el problema computacionalmente se representa la imagen satelital como una *array* bidimensional de `bool`, donde `true` significa tierra y `false` agua.

Usted debe implementar una biblioteca de clases `CercandoIslas` (proyecto de tipo *Class Library*) con un método dentro de la clase `Cartografia` de nombre `Cercar` y cuya signatura sea la siguiente:

```
namespace CercandoIslas
{
    public class Cartografia
    {
        public static Rectangulo Cercar (bool[,] mapa)
        {
            // TODO: Implemente aquí su ejercicio
        }
    }
}
```

```
}  
}
```

La clase **Rectangulo** se encuentra en el espacio de nombres **Weboo.Utiles** en la biblioteca **Utiles.dll**.

```
public class Rectangulo  
{  
    //Representa la esquina superior izquierda del rectangulo  
    public Punto Localizacion{get;set;}  
    //Devuelve o modifica el ancho del rectangulo  
    public int Ancho {get;set;}  
    //Devuelve o modifica el alto del rectangulo  
    public int Alto {get;set;}  
}  
public class Punto  
{  
    public int X{get;set;}  
    public int Y{get;set;}  
}
```

En el caso de que no exista rectángulo, su método de debe devolver **null**.

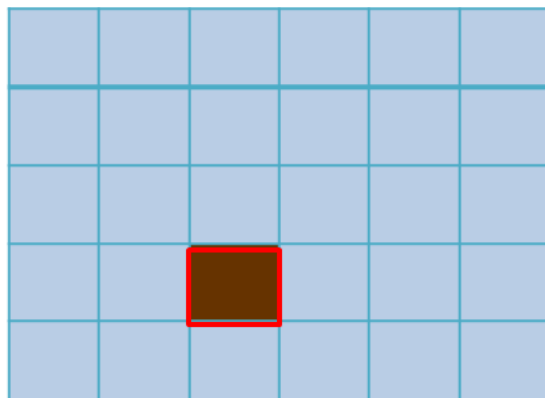
Ejemplo 1.

Si su método es llamado con el mapa de la figura 1, este deberá devolver un rectángulo cuya **Localizacion** es (2,2), su **Alto** es 4 y su **Ancho** 5

Ejemplo 2.

Si su método es llamado con el mapa de la figura 2, deberá devolver un rectángulo cuya **Localizacion** es (3,2), su **Alto** es 1 y su **Ancho** 1

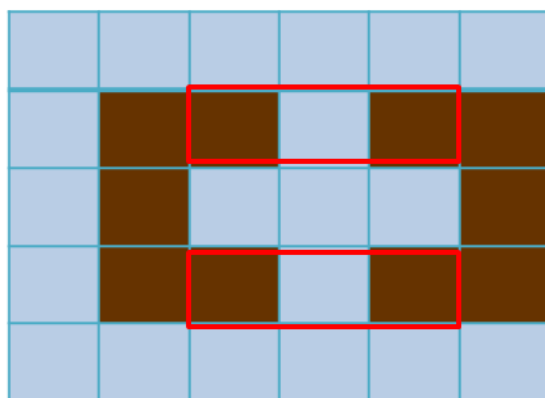
Figura 2. Mapa con una sola isla.



Ejemplo 3.

En el caso de que tenga más de un rectángulo de área mínima, puede retornar cualquier opción (Figura 3)

Figura 3 Un mapa con varios rectángulos de área mínima.



Nota: Usted debe utilizar las clases que se le brindan en `Utiles.dll` ([Punto](#) y [Rectangulo](#)), en vez de hacer su propia implementación