Práctica 6. Mallas regulares

Sesiones de prácticas: 2

Objetivos

Utilizar mallas regulares para minimizar el tiempo de búsqueda de las farmacias más cercanas.

Descripción de la EEDD

La aplicación desarrollada hasta ahora no ha considerado a los usuarios o clientes de las farmacias, ni tampoco su ubicación real. En esta práctica vamos a utilizar una malla regular para saber dónde están las farmacias más cercanas de cara a los usuarios. Se puede seguir la implementación de la Lección 17 añadiendo la nueva funcionalidad que se muestra a continuación:

```
template <Typename T>
class MallaRegular {
            …
public:
      MallaRegular(float aXMin, float aYMin, float aXMax, float aYMax, int
nDiv);
….
      vector<T> buscarCercana(float xcentro, float ycentro, int n=1);
      T buscar (float x, float y);
      unsigned maxElementosPorCelda();
      float promedioElementosPorCelda();
}
```

Esta Malla Regular o Grid almacenará las farmacias de acuerdo a sus posiciones (latitud, longitud) y el objetivo será conocer qué farmacia está más cerca de los usuarios del sistema (a los que también se les dará coordenadas). Las funciones *maxElementosPorCelda()* y *promedioElementosPorCelda()* sirven de métrica para conocer la carga y distribución de los puntos en todas las casillas de la estructura y decidir así el tamaño de ésta. La función *buscarFarmacias()* devuelve las *n* farmacias más cercanas a una coordenada *(x,y)* dada, que corresponderá con las posiciones de potenciales usuarios del sistema. Podemos asumir que todo tipo *T* con el que se instancie la clase tiene los métodos getX() y getY() implementados, que en nuestro caso son respectivamente la longitud y la latitud en coordenadas UTM.

Descripción de la práctica:

El diseño UML (última página del documento) ahora añade la clase Usuario, que serán los compradores de medicamentos en las farmacias.

Usuario:
El sistema MediExpress les indica a los usuarios cuál es la farmacia o farmacias más cercanas a su posición, y de este modo las localiza más fácilmente usando la función *Usuario::getFarmaciaCercana()*, que le devuelve una, dos o tres más cercanas (según el valor de *n*). Una vez elegida la farmacia (se supone que se desplaza hacia ella) el cliente pregunta por un medicamento dando el nombre de éste, que bien puede ser su nombre completo o partes de ese nombre, tal y como se hizo en la práctica anterior (por ejemplo "MAGNESIO"), mediante la función *Usuario::quieroMedicam()*. Entonces la farmacia busca el/los medicamento/s que correspondan a ese nombre devolviendo la lista/vector de objetos del tipo *PaMedicamento*. Si no hay, la lista se devuelve vacía. Si hay existencias de alguno de esos medicamentos, el usuario ya puede decidir comprar uno concreto con *Usuario::comprarMedicam()* indicando el número de unidades que quiere. Si no hay suficientes, al menos se llevará las que queden en stock indicando el número que finalmente se lleva a casa.

MediExpress:
Esta clase tiene la misma funcionalidad que la Práctica 5, pero añadiendo los usuarios que usan la aplicación para localizar las farmacias más cercanas a su posición mediante la función *MediExpress::buscarFarmacias()*, que localiza hasta 3 farmacias más cercanas al usuario.

El resumen de contenedores a utilizar son:

- *Farmacia::order* → **std::map<int, Stock>** relación de composición que mantiene el stock y, por tanto, los medicamentos ordenados por ID.
- *MediExpress::pharmacy* → **std::multimap<string, Farmacia>** con clave la provincia.
- *MediExpress::idMedication* →**ThashMedicam** con clave el ID del PaMedicamento
- *MediExpress::nombMedication* → **std::multimap<string, PaMedicamento*>** relación de asociación usando cada palabra del nombre del medicamento como clave.
- *MediExpress::labs* → **std::list<Laboratorio>** para mantener los laboratorios sin orden específico.
- *MediExpress::users* → **std::map<int, Usuario>** para almacenar los usuarios ordenados por ID.

- *MediExpress::grid* → **MallaRegular<Farmacia *>** para gestionar las farmacias por su ubicación.

Programa de prueba:

A continuación se describen los pasos que debe seguir el constructor de la clase MediExpress, y que tomará ahora los nombres de los **4** ficheros como parámetros:

1. Leer los ficheros de las prácticas anteriores y cargar la información en los contenedores de la STL y la tabla hash como en la Práctica 5 teniendo en cuenta que **se usará un nuevo fichero de farmacias: farmacias-coord.csv**.
2. Para establecer la nueva relación *MediExpress::users*, leer el nuevo fichero de usuarios que se llama **usuarios.csv**.
3. Para la nueva relación *MediExpress::grid* se usará la maya regular descrita anteriormente. Para ello, recorrer secuencialmente todas las farmacias, tomar su dirección de memoria e insertarlo adecuadamente en la malla regular utilizando sus coordenadas UTM. La malla se crea a partir de las coordenadas antes citadas. Se consideran *(aXmin, aYmin)* las coordenadas X e Y más pequeñas, localizadas al leer el fichero de farmacias, y *(aXmax, aYmax)* las coordenadas máximas en X e Y respectivamente, también obtenidas de ese fichero. El número de casillas en la malla vendrá determinado por el **máximo de farmacias por casilla** una vez insertadas las farmacias en la malla, debiendo estar entre 10-15. Por lo tanto, se debe probar con diferentes números de casillas de la malla hasta obtener ese promedio. Además, se debe calcular cuántas farmacias tiene la casilla más poblada usando las funciones de métrica descritas anteriormente.
4. Al igual que en prácticas anteriores, enlazar de forma automatizada los medicamentos (un total de 3310) con los laboratorios (un total de 1579), incluyendo a los 152 ubicados en Madrid. Para ello de nuevo usar el vector *vMedi*, buscar las claves, obtener los PaMedicamento* y hacer el enlace. Enlazar también las farmacias con los medicamentos de forma consecutiva cada 100 medicamentos, como se hizo anteriormente, haciendo que haya inicialmente en stock 10 medicamentos de cada tipo.

Una vez que la estructura está inicializada realizar las siguientes acciones:

5. Cada uno de los 34 usuarios de la provincia de Jaén (*Usuario::provincia*) van a ir a la farmacia de Úbeda para buscar los siguientes compuestos (nombre completo o parcial) de forma secuencial, es decir, el primero va por el medicamento (a), el segundo por el (b), el tercero por el (c), el cuarto por el (a) y así sucesivamente. Si no existen dichos medicamentos se piden 10 unidades como hasta ahora. Mostrar en pantalla la información de los usuarios, medicamentos que compran, stock, y toda la

información relevante que sea necesaria para comprobar el proceso. Comprobar también si para todos ellos la farmacia de Úbeda es la más cercana, mostrando también dicha información. Los compuestos son:
   a. MAGNESIO CLORURO HEXAHIDRATO
   b. LIDOCAINA HIDROCLORURO
   c. MENTA PIPERITA
6. De nuevo, los usuarios ubicados en Sevilla van a comprar cualquier tipo de "MAGNESIO" y cada cual lo hará en la farmacia más cercana a su posición actual. De nuevo, se llevará a casa cualquier tipo de magnesio que esté en stock en esa farmacia más cercana y si no hay, entonces se hace la petición de "MAGNESIO OXIDO". Listar convenientemente a cada usuario sevillano, la farmacia a la que va y cómo está el stock de todos los magnesios y si le hace falta hacer o no el pedido a dicha farmacia.
7. Las autoridades sanitarias madrileñas han prohibido que se dispense cualquier tipo de BISMUTO. Buscar todas las farmacias de Madrid que tienen algún tipo de BISMUTO en stock. Comprobar cuáles de los usuarios de esta comunidad tienen a una de esas farmacias como una de las 3 más cercanas. Que esos usuarios compren una unidad de BISMUTO antes de que entre la restricción. Listar todo este proceso en pantalla de forma conveniente.
8. Una vez que los usuarios han hecho esa compra, eliminar el BISMUTO de todas esas farmacias y comprobar que ya no están disponibles.

Para los que trabajan por parejas:

La asociación de farmacias de la provincia de Jaén ha decidido incluir en MediExpress una farmacia en Jaén capital. Incluye en el multimapa esta nueva farmacia:

12345678A JAEN JAEN FARMACIA NUEVA PASEO DE ESPAÑA 35 23009

sus coordenadas son (lat, lon), (37.78710, -3.79104)

Asignar todos los medicamentos que contengan la palabra "MAGNESIO". Recuerda incluir 10 unidades de cada medicamento para el stock. Localiza el usuario más cercano a esa farmacia y mostrarlo por pantalla. Dicho usuario va a esa farmacia para comprar 3 unidades de "MAGNESIO OXIDO". Muestra el stock de ese medicamento antes y después de la compra.

VOLUNTARIO: Visualización 2D con la clase Img

Para poder visualizar el resultado de forma gráfica, se adjunta a esta práctica la clase **Img**. Esta clase es en realidad una matriz de píxeles, creada tomando como parámetro el tamaño de un recuadro en número de píxeles en (tamaFilas, tamaColumnas). Esta clase es capaz de dibujarse como imagen de modo que cada consulta generará un fichero imagen resultado con *Img::guardar()*. Ejecutar la función *main()* que aparece junto al código para comprobar el funcionamiento y visualizar la imagen resultante. En la imagen adjunta a la práctica se puede ver la disposición espacial de todas las farmacias.

Para que esta clase sea útil en nuestro caso, hacemos coincidir las esquinas de la imagen con las del cuadro de trabajo donde se incluyen todas las coordenadas de las farmacias. Consideramos, pues, que la esquina inferior izquierda de nuestros datos es la latitud y longitud mínima de las posiciones de las farmacias y la esquina superior derecha es la latitud y longitud máxima de las farmacias. En el ejemplo de prueba se pinta un recuadro y se pinta también un pixel azul.

Utilizar esta clase para dibujar con distinto color cada uno de los usuarios. De igual forma, dibujar todas las farmacias en otro color. Todo esto debe mostrarse en una imagen de 600*600.

Universidad de Jaén

Departamento de Informática

Prácticas de Estructuras de Datos

Grado en Ingeniería en Informática

Curso 2025/2026