Estructuras de Datos y de la Información/ Data and Information Structures (EDI 17/18)
Proyecto de programación/ Programming Project
Cáceres Opendata

Introducción

Los organismos públicos están haciendo un gran esfuerzo por publicar de forma abierta los datos relacionados con los servicios que proporcionan. En este sentido, el Excmo. Ayuntamiento de Cáceres mantiene un portal de datos en diferentes formatos, públicamente accesibles en la dirección web:

http://opendata.caceres.es

El reto está en el tratamiento eficiente de estos datos para obtener información útil. Con este objetivo, se plantea el proyecto de la asignatura *Estructuras de Datos y de la Información*. Durante el proyecto se realizarán las siguientes tareas generales:

- Obtención y almacenamiento de los datos relevantes para los procesos que se lleven a cabo en el proyecto.
- Diseño de un sistema con las diferentes entidades y estructuras de datos usadas para almacenar los datos. El diseño se realizará siguiendo las metodologías impartidas en clase. En especial, las estructuras de datos utilizadas serán eficientes y adecuadas.
- Implementación de algoritmos para el tratamiento eficiente de los datos con el objetivo de obtener información relevante.
- Evaluación de resultados del diseño y ejecución del sistema.

La calidad del sistema desarrollado se evaluará en base a dos aspectos principales: extensibilidad y flexibilidad de su diseño e implementación, cumplimiento de los principios de orientación a objetos, y eficiencia de los algoritmos implementados.

Como resultado del proyecto, además del código, se debe crear un **informe final**, que incluirá la documentación relativa al análisis, diseño, e implementación del sistema, así como las pruebas y los resultados de la evaluación de la funcionalidad implementada. El formato del informe final se proporcionará por parte del profesorado.

Las tareas del proyecto se realizarán durante el periodo de impartición de la asignatura. Es importante tener en cuenta que las fases enumeradas previamente, lejos de ser secuenciales, siguen un esquema de diseño de sistemas iterativo, en el que se van añadiendo funcionalidades al proyecto de forma incremental.

En este documento se incluye una descripción general de la metodología de trabajo que se seguirá a lo largo del proyecto, junto con las tareas y los aspectos generales.

Metodología de trabajo

Como se ha comentado anteriormente, para realizar este proyecto se seguirá una metodología iterativa e incremental, es decir, el proyecto se deberá implementar a lo largo de la impartición de la asignatura. Este avance será realizado en incrementos en cuanto a la definición de los requisitos del proyecto. Un incremento es abordado en un bloque temporal denominado iteración. Así, la primera iteración abordará un conjunto de requisitos o funcionalidades pequeñas y en cada iteración se irán aumentado el número de funcionalidades a abordar. Por lo tanto, a medida que se termina una iteración se tendrá un producto más completo.

En cada iteración, el conjunto de tareas de desarrollo a realizar está definido, siendo:

1. **Evaluar e importar los datos a tratar**. En esta tarea se deben analizar el conjunto de datos que se van a tratar durante la iteración. Si bien lo datos a utilizar estarán basados en los publicados en la plataforma *Opendata* de Cáceres, se deberán usar los publicados en el Campus Virtual de la asignatura. Posteriormente, se deberán desarrollar todos los algoritmos necesarios, o adaptar los ya existentes, para poder leer dichos datos e importarlos a nuestro proyecto.

2. **Diseño de las entidades y de las estructuras de datos**. En esta actividad se deberá identificar cómo almacenar el conjunto de datos tratados en la primera tarea, y qué estructuras de datos son necesarias para contenerlos. Posteriormente, se realizarán los diagramas UML necesarios con el diseño definido. Finalmente, dicho diseño deberá ser implementado.

3. **Implementación de algoritmos**. Los datos tratados en las tareas anteriores corresponden a una serie de algoritmos y funcionalidades del proyecto que deberán ser implementadas. En esta tarea se deberán implementar dichos algoritmos usando, o modificando si fuese necesario, el diseño del sistema para incluirlos.

4. **Evaluación y prueba del sistema**. En todo momento, durante el desarrollo del proyecto, el alumno debe intentar desarrollar un producto de calidad. En esta tarea se deben realizar todas las actividades necesarias para evaluar la calidad del producto. En concreto, se deberán realizar pruebas unitarias, pruebas de integración y pruebas de rendimiento del sistema.

Como ya se ha indicado anteriormente, todas estas tareas deberán ser abordadas de forma iterativa para que se pueda completar el proyecto con un menor esfuerzo. A continuación, se detalla qué se debe realizar en cada una de las tareas.

1. Evaluación e importación de los datos a tratar

Nuestro sistema implementará una serie de procesos sobre los datos previamente obtenidos del portal *Opendata* del Ayuntamiento de Cáceres. En este portal hay múltiples datos de diferentes entidades en forma de **tablas**. Cada tabla contiene uno o más **registros** (filas). Cada fila contiene toda la información de una entidad. Cada fila contiene columnas, que contienen los datos de los distintos capos de dichas entidades. Como ejemplo, tenemos la información de los barrios de Cáceres en la siguiente tabla:

LABEL	Om_perteneceaDistrito
Aguas Vivas	Centro
Charca Musia	Sur
Espíritu Santo	Sur
El Vivero	Oeste
Gredos	Norte
Valdesalor	Pedanías

Los campos de la tabla son **Label (Nombre)**, que se refiere al nombre del barrio, y **Om_perteneceaDistrito,** que identifica el distrito al que pertenece el barrio.

Los datos son proporcionados como archivos de texto en formato **CSV**[1]. Los archivos de texto en este formato contienen los datos en líneas, separando los campos por algún carácter especial, que suele ser coma (,), punto y coma (;), almohadilla (#), etc.

Los datos a obtener serán los correspondientes a las siguientes entidades:

1. Barrios.
2. Vias.
3. Padrón municipal.

La primera tarea del proyecto será comprobar la importación correcta de los datos desde los archivos CSV, para lo cual, se leerán los archivos y se mostrarán por pantalla los campos que se consideren relevantes de cada registro.

2. Diseño del sistema

Nuestro sistema se va a construir siguiendo el paradigma de orientación a objetos. Los datos deben almacenarse en objetos para los cuales hay que diseñar las clases adecuadas.

Algunas de las clases que formarán parte de nuestro sistema serán proporcionadas por el profesorado y servirán de ayuda en la implementación.

Además, se debe identificar qué estructuras de datos son las más adecuadas para almacenar cada información e incluirlas en el diseño del sistema. La implementación básica de algunas de estas estructuras será proporcionada por los profesores de las asignatura.

Importante: Las clases y estructuras de datos utilizadas en el proyecto deben estar justificadas y evaluadas en cuanto a su eficiencia.

En la primera iteración del proyecto, se deben diseñar e implementar las clases y relaciones necesarias para almacenar los *Barrios y Vías* de Cáceres.

En la segunda iteración del proyecto, se deben diseñar e implementar las clases para almacenar el Padrón de Cáceres y las estructuras de datos necesarias para gestionar toda la información del proyecto.

[1] Comma-Sparated Values.

En la <u>tercera iteración</u> del proyecto, se debe evaluar la necesidad de modificar algunas de las estructuras de datos lineales utilizadas para almacenar la información de los *Barrios, Vías y Padrón* con estructuras de datos NO lineales. Cualquier cambio debe estar claramente justificado en el informe final que se debe generar. Además, se deberán implementar los distintos algoritmos del proyecto.

3. Implementación de algoritmos y funcionalidades

Los datos importados y el diseño realizado, debe estar orientado a facilitar el procesamiento de la información. Este procesamiento viene definido por determinadas funcionalidades que deberían ser implementadas porque serían interesantes tanto para el Excmo. Ayuntamiento de Cáceres, como para otras empresas encargadas de realizar análisis estadísticos. A continuación, se presentan las distintas funcionalidades que deben ser implementadas[2].

Los algoritmos a implementar están divididos en dos grupos: funcionalidades no críticas y funcionalidades críticas. Las funcionalidades no críticas deben tener una implementación correcta y eficiente, pero no es necesario que sean ejecutados en el menor tiempo posible. Las funcionalidades críticas son aquellas para los cuales el tiempo de ejecución y la eficiencia es extremadamente importante por lo que es OBLIGATORIO que se compare sus tiempos de ejecución utilizando estructuras de datos lineales y árboles. Dicha comparación deberá quedar claramente expresada en el informe final de resultados indicando qué estructura de datos es la más eficiente, en qué medida y porqué.

Algoritmos:

1. Cargar los datos básicos para que el resto de funcionalidades se puedan desempeñar de forma exitosa. *Nota: cada estudiante debe decidir cómo cargar dichos datos en las estructuras y si están ordenados por algún criterio.*

2. Mostrar todas las vías de un barrio. Esta funcionalidad debe permitir al usuario indicar por consola un barrio y mostrarles por consola todas las vías de dicho barrio ordenadas alfabéticamente por nombre.

3. Mostrar las vías que pertenecen a varios barrios. Se debe mostrar por consola todas vías de Cáceres que están asignadas a varios barrios, detallando también los barrios en los que se encuentran.

[2] **Importante**: los algoritmos deben ser correctos y eficientes en cuanto a tiempo de procesamiento y consumo de memoria. Se debe proporcionar la eficiencia en notación Big-O de todos los algoritmos implementados.

4. Mostrar el número de habitantes de Cáceres agrupados por rango de edad cada 10 años (i.e., de 0 a 9, de 10 a 19, etc.). Opcional: Representar los valores de forma gráfica en la consola.

5. Realizar un algoritmo que muestre un listado con todas las nacionalidades de los habitantes de cáceres (junto con el número de ciudadanos de cada nacionalidad) ordenados de forma descendente por número de habitantes.

6. Mostrar por consola el número de habitantes por provincia de nacimiento, ordenados alfabéticamente por provincia, excluyendo habitantes no nacidos en España.

7. Mostrar por consola el barrio con mayor número de habitantes para un rango de edad indicado por el usuario por consola.

8. Generar un fichero con el nivel de estudios de todos los ciudadanos de un barrio y el número de habitantes para cada nivel, ordenado de mayor a menor por el número de habitantes.

9. Mostrar por consola el barrio con mayor porcentaje de hombres y el barrio con mayor porcentaje de mujeres.

10. Generar un fichero que contenga un listado con el nombre y número de habitantes que proceden de los pueblos de una provincia, detallada por el usuario por consola.

11. Generar un fichero con el listado de los lugares de nacimiento de los vecinos de una calle, introducida por el usuario por consola (Algoritmo Crítico).

12. Escribir un algoritmo que muestre el número de personas nacidas en una provincia dada para todas las vías que comiencen por una subcadena determinada (Algoritmo Crítico).

Finalmente, el sistema desarrollado debe presentar una interfaz que proporcione al usuario (personal del Ayuntamiento o de las empresas externas) la posibilidad de ejecutar tantas veces como lo desee las funcionalidades anteriores.

Condiciones que deben cumplir la implementación de los algoritmos.

- Puedes usar las estructuras auxiliares que consideres oportuno. No se permiten duplicados de los datos en diferentes estructuras. Por tanto, solo existirá una copia de cada objeto en un momento dado, esto es, todas las E. de Datos deben almacenar punteros a los mismos, lo que garantizará la no duplicidad.
- La gestión de la memoria dinámica debe ser correcta.

- Para todos los algoritmos implementados debes justificar el uso de las estructuras de datos. Debes proporcionar la eficiencia del algoritmo en notación Big-O, incluyendo caso más favorable si lo consideras oportuno.
- En el caso de los algoritmos marcados como críticos, deberás realizar varias implementaciones utilizando estructuras de datos lineales y árboles, y comparar su rendimiento mediante la clase ***Time***[3], comprobando la diferencia que existe en cuanto a tiempos de ejecución. El estudio y la implementación de cada uno de los algoritmos debe ser completo y detallado, y la documentación debe ser incluida en el informe final.

4. Evaluación y prueba del sistema

Es necesario asegurar que el código entregado funciona correctamente, para ello es necesario crear las **pruebas unitarias** y de **integración** necesarias que lo garanticen.
No es necesario realizar pruebas sobre las estructuras de datos entregadas por los profesores.

Como actividad principal de esta fase se deberá calcular la **complejidad de cada algoritmo** y realizar **pruebas de rendimiento.** Estas pruebas deben medir el tiempo que tarda en realizarse cada una de las funcionalidades y algoritmos del sistema. Como se ha indicado anteriormente, para poder realizar estas pruebas, los profesores proporcionarán un conjunto de clases que podrán ser utilizadas para medir el tiempo de ejecución.

Aspectos generales

- Este proyecto se realizará en grupos de dos.

- Se debe entregar la implementación (en código C++) completa. El proyecto debe incluir toda la funcionalidad indicada en el enunciado del proyecto.

- El proyecto entregado debe funcionar correctamente para las herramientas de desarrollo utilizadas en los laboratorios (S.O. Debian 8, IDE Eclipse, GCC 4.6). Nota: Si se utiliza alguna extensión al lenguaje c++, dichas extensiones deben quedar perfectamente detalladas en el informe final.

- Cada estudiante es responsable de probar antes de la entrega el funcionamiento correcto de su proyecto en dichos laboratorios.

- En la ***documentación interna*** del proyecto se incluirá:
  - Para cada clase:

[3] Esta clase, para medir el tiempo de ejecución de un algoritmo, será proporcionada por los profesores de la asignatura.

    - Nombre del autor
    - Descripción de la clase
    - Descripción de los atributos
- Para cada método:
    - Descripción del método
    - Precondiciones y postcondiciones
    - Parámetros
    - Complejidad

PRUEBA DE EVALUACIÓN

El día del examen final de la asignatura se realizarán las correspondientes pruebas de evaluación del proyecto, en el horario que se publicará en la convocatoria oficial. Esta prueba solo hay que hacerla si se ha entregado el proyecto en el plazo indicado.

La prueba se realizará de manera individual.

CALIFICACIÓN DEL PROYECTO

Como requisito previo a la evaluación del proyecto está la superación de la prueba de evaluación. Si no se supera, la nota será 2. Si es copia de otro proyecto, la nota será 0 en el bloque y en toda la asignatura.

La evaluación se hará en función del desarrollo de la aplicación, de la documentación y de la prueba de evaluación del proyecto.