UNIVERSIDAD DISTRITAL

FRANCISCO JOSÉ DE CALDAS

### CIENCIAS DE LA COMPUTACIÓN 1

*PROYECTO FINAL 2019-1*

**Enunciado del problema.**

Una empresa que ofrece el servicio de “paseadores de perros” requiere de un software para recoger y gestionar la información tanto de sus clientes como de sus empleados, con el fin de hallar perfiles que les permitan enfocar el desarrollo de las labores de manera más eficiente. Por lo tanto, ha pensado en usted y su grupo de trabajo para realizar este proyecto.

Para recoger la información, el gerente de la empresa ordena la elaboración de formularios que son entregados en cada una de las sucursales ubicadas a lo largo y ancho de la ciudad (puede haber más de una sucursal en la misma localidad).

Se espera que su desarrollo en general pueda: recibir una solicitud de un cliente ubicado en alguna localidad y asignar un paseador para un perro determinado teniendo en cuenta la localidad en la que dicho perro se encuentra y el tiempo que el cliente ha solicitado.

Debe realizar una simulación del proceso de solicitud y asignación del paseador, de forma tal que la empresa quede convencida de la calidad de su desarrollo y los beneficios que logrará con este software.

.

**ESTRUCTURA DE DATOS**

**En memoria secundaria:**

La información deberá estar almacenada, de manera permanente, en archivos planos los cuales deben ser actualizados al final de la ejecución, dado que, por ejemplo, durante el proceso puede ser excluido o incluido un paseador o un cliente. Debe tenerse la posibilidad de modificar la información ya sea agregando, eliminando o cambiando registros.

Existen por lo menos los archivos planos de sucursales, paseadores y clientes, de la siguiente forma:

**SUCURSALES:**

Nombre de la sucursal, nombre del gerente, localidad a la que pertenece, área que cubre teniendo en cuenta: calle donde inicia, calle donde termina, carrera donde inicia y carrera donde termina. Número de paseadores con los que se cuenta y clientes con los que se cuenta de manera permanente.

**PASEADORES:**

En cada momento de trabajo, un paseador podrá hacerse cargo de máximo dos perros simultáneamente. De cada paseador se desea conocer:

* Sucursal a la que pertenece (es deseable que viva cerca)
* Nombre
* Apellido
* Tipo de identificación: puede ser cédula de ciudadanía o cédula de extranjería
* Número de identificación
* Sexo: F o M
* Teléfono celular
* Teléfono fijo
* E-mail
* Fecha de nacimiento (por razones de seguridad no se contratan menores de edad ni adultos mayores)
* Ciudad de nacimiento
* País de nacimiento
* Dirección
* Barrio
* Horario disponible (cada día puede tener una hora de inicio y finalización diferentes, sin embargo, la franja de trabajo debe ser continua)

**CLIENTES:**

De cada cliente se requiere:

* Nombre
* Apellido
* Número de identificación
* Sexo: F o M
* Localidad de Residencia
* Número de perros que posee.
  + Por cada perro se requiere:
    - Nombre
    - Fecha de nacimiento
    - Raza
    - Tamaño (grande, mediano o pequeño)
    - Tipo de concentrado que consume
    - Resumen de cada paseo:
* Nombre del paseador, fecha, hora y duración del paseo
  + - * Tipo de actividad o entrenamiento realizado
      * Observaciones

**En memoria Principal:**

Con el fin de optimizar las búsquedas deberán utilizarse listas, multilistas, arreglos, pilas, colas o árboles en memoria principal. Para la realización de dichas búsquedas, deberá evitarse la utilización total de la información. Es decir, se espera que apliquen los criterios y conocimientos correspondientes a la eficiencia de los algoritmos. Complejidad en tiempo y memoria.

Por ejemplo, se pueden generar listas tales como:

1. Sucursales pertenecientes a una localidad.
2. Paseadores que trabajan en una sucursal.
3. Paseadores de una localidad.
4. Todos los clientes de una sucursal.
5. Todos los clientes de una localidad.
6. Clientes con más de un perro. (No se puede con árboles)
7. Lista de los perros de una sucursal.
8. Lista de los perros de una localidad.
9. Lista de los perros, con la localidad a la que pertenecen, que cumplen años en un mes determinado. (No se puede con árboles)
10. Usaríamos para las consultas anteriores un árbol de sucursales y otro de localidades.

**REQUERIMIENTOS FUNCIONALES.**

**Consultas:**

* Dada un área (definida por calles y carreras) determinar qué sucursales podrían hacerse cargo de perros en esa área. (NO TIENE SENTIDO HACERLO CON ÁRBOLES).
* Dado un horario y una localidad, mostrar la lista de los paseadores que podrían estar disponibles. (Para la búsqueda de la localidad se puede con árboles, de resto se trabajaría con listas).
* Dado un cliente y un horario, lista de los paseadores que podrían hacerse cargo de sus perros (teniendo en cuenta las limitaciones de número de perro por paseador). (SE PUEDE CREAR UN ARBOL DE CLIENTES PARA LA BUSQUEDA Y DE RESTO TRABAJARÍAMOS CON LISTAS).
* Dada una localidad, mostrar todos los clientes que poseen perros de una raza dada. (PARA LA BUSQUEDA DE LA LOCALIDAD SE TRABAJA CON ÁRBOLES Y DE RESTO CON LISTAS)(SE PODRÍA HACER CON MULTILISTA ESTÁTICA).
* Dado un paseador mostrar la lista de perros con fecha, hora y duración de cada paseo realizado en un rango de tiempo dado .(SE CREARÍA UN ÁRBOL “PASEADOR” Y DE RESTO CON LISTAS).
* Dado un perro, mostrar los paseadores con fecha, hora, duración y tipo de actividad realizada en cada paseo dentro de un rango dado. (SE CREA UN ÁRBOL DE “PERROS” Y PARA EL RESTO DE LA BUSQUEDA CON LISTAS).
* Dada una localidad, mostrar la lista de los perros de un tamaño específico adscritos a dicha localidad. (SE UTILIZA EL ÁRBOL DE LOCALIDAD ).
* Dado un rango de edad, listar todos los perros que están en dicho rango haciendo explicita la edad, el dueño y la localidad a la que pertenece. Cabe aclarar que la precisión de la edad estará dada solamente en términos de años y meses. (LISTAS)

**Simulación de Solicitud de paseador.**

* Creará las estructuras en memoria principal a partir de los archivos planos.
* Permitirá modificar los datos en memoria principal. Retirar un registro, agregar uno nuevo, modificar uno existente. Es importante anotar que sólo se debe tener un registro por cada cliente, para ello, puede asignársele un identificador único que puede corresponder por ejemplo a su cédula.
* Simulará el proceso de solicitud de paseador, recibiendo la localidad en la que se encuentra el (los) perro(s), y el tiempo solicitado, asignando uno o más paseadores según sea el caso. Teniendo en cuenta el horario de trabajo del paseador y el número máximo de perros que puede cubrir simultáneamente. Para facilitar esta simulación, se limitará el proceso de la siguiente manera:
  + Los paseos duran mínimo 1 hora
  + Los paseos solo se podrán solicitar en horas exactas ejemplo (7:00-8:00) y cada uno debe durar un rango de horas completas (1 hora, 2 horas, etc, no se tiene en cuenta minutos ni segundos)
* Simulará la finalización del día donde para cada localidad mostrará los resultados:

Por tamaño: número de perros atendidos, porcentaje de perros pertenecientes a esa localidad. Número de perros externos atendidos, con la sucursal a la que pertenecen (este dato es proporcionado por la localidad que atendió el perro, no aquella en la que está inscrito).

Por horario: localidad a la que pertenece, Nombre de cada perro, edad, tamaño, actividad desarrollada y localidad en la que se encuentra inscrito.

* Estadísticas:
  + - Reporte Diario total: Total perros atendidos, total perros que no se pudieron cubrir, total servicios en la localidad a la que pertenecen, total servicios en una localidad diferente a la inscrita. En cantidad y porcentaje.
    - Reporte por localidad: Total perros atendidos, total perros que no se pudieron cubrir, total servicios en la localidad a la que pertenecen, total servicios en una localidad diferente a la inscrita. En cantidad y porcentaje.
    - Reporte por paseador: Total perros atendidos, total perros que no se pudieron cubrir, total servicios en la localidad a la que pertenecen, total servicios en una localidad diferente a la inscrita. En cantidad y porcentaje.
* Finalizada la simulación actualizará los archivos planos con los cambios realizados en memoria principal. Lo cual permitirá nuevos procesos de simulación. Tenga en cuenta que una vez iniciada la simulación, no podrá eliminarse ni agregarse ningún registro.

Contará con los menús y submenús que faciliten la realización de cada una de las consultas y la simulación.

Debe aplicar programación orientada a objetos y el principio de alta cohesión y bajo acoplamiento.

Debe aplicar los conceptos de optimización estudiados en el curso. No limitarse a garantizar el funcionamiento del programa, sin embargo, se podrá usar información redundante (repetida) sólo en los casos que sea estrictamente necesario y que tengan soporte argumentativo que demuestre una mejora sustancial, en términos de comportamiento de los algoritmos.

**Condiciones de entrega**

1. Se puede elaborar en grupos de por lo menos dos y máximo tres personas (Es un trabajo en grupo). Se sustentará individualmente.
2. Se entrega a través de aula virtual una carpeta comprimida con: código fuente, ejecutable, archivos planos y documento soporte. Sencillo manual para ejecución (si se requiere).
3. El programa debe correr en DevC++,sin requerir ningún tipo de ajuste.
4. Documento soporte incluye: Diseño de la solución, Contenido y estructura de archivos planos, Estructuras de datos en memoria principal definidas, incluyendo diagrama.
5. Archivo fuente debidamente autodocumentado.
6. Los archivos planos contendrán por lo menos: 10 sucursales, 10 paseadores y 10 clientes por sucursal.

**Docente**: ***Deicy Alvarado.***