

CB100 - ALGORITMOS Y ESTRUCTURAS DE DATOS

Cátedra Ing. Gustavo Schmidt - 2do cuatrimestre 2024

Trabajo Práctico 2: Indexando la ciudad

Objetivo

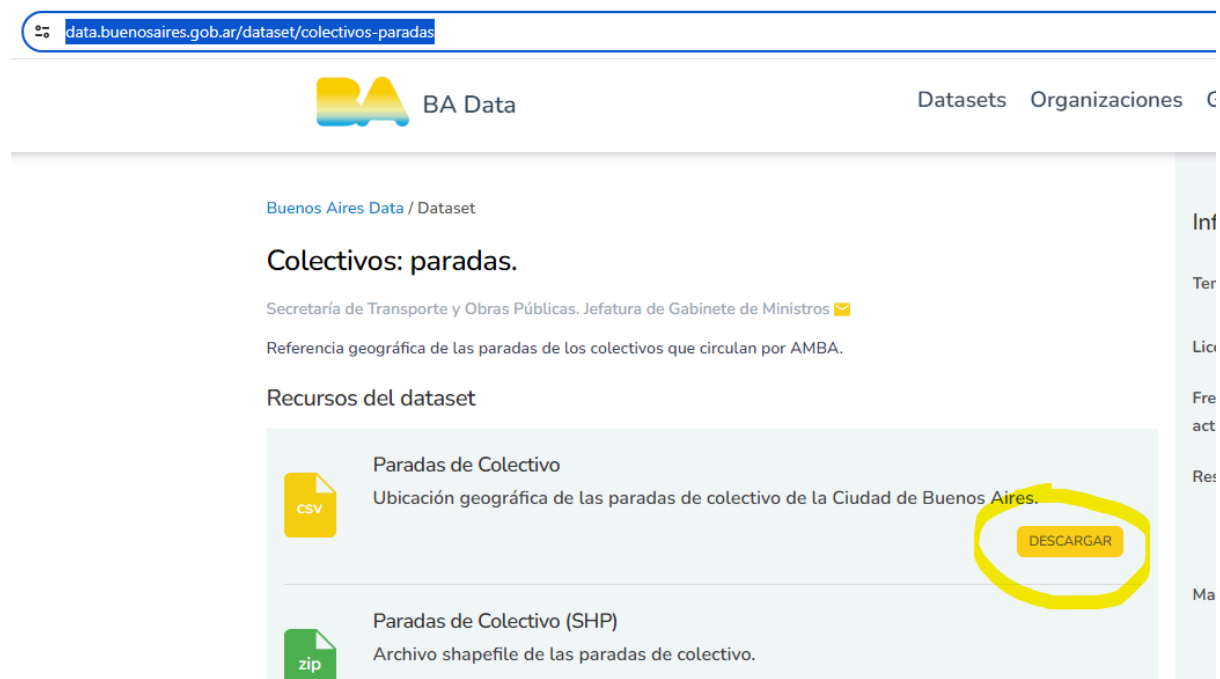
Generar una pieza de software que genere un índice sobre información dada por la ciudad.

Enunciado

Descargar el archivo csv (archivo de texto separando los campos por comas) de la página del gobierno de la ciudad:

Descargar la información de:

<https://data.buenosaires.gob.ar/dataset/colectivos-paradas>



The screenshot shows the 'Buenos Aires Data' portal. The URL bar displays 'data.buenosaires.gob.ar/dataset/colectivos-paradas'. The page header includes the 'BA Data' logo and navigation links for 'Datasets' and 'Organizaciones'. The main content area is titled 'Colectivos: paradas.' and includes the following information:

- Secretaría de Transporte y Obras Públicas, Jefatura de Gabinete de Ministros
- Referencia geográfica de las paradas de los colectivos que circulan por AMBA.
- Recursos del dataset

Under 'Recursos del dataset', there are two entries:

- Paradas de Colectivo** (CSV icon): Ubicación geográfica de las paradas de colectivo de la Ciudad de Buenos Aires. A yellow circle highlights the 'DESCARGAR' button next to this entry.
- Paradas de Colectivo (SHP)** (zip icon): Archivo shapefile de las paradas de colectivo.

El link del archivo csv es:

<https://cdn.buenosaires.gob.ar/datosabiertos/datasets/transporte-y-obras-publicas/colectivos-paradas/paradas-de-colectivo.csv>

El formato en columnas es:

	A	B	C	D	E	F	G	H	I	J	K	L	M	
1	CALLE	ALT PLANO	DIRECCION	coord_X	coord_Y	COMUNA	BARRIO	L1	l1_sen	L2	l2_sen	L3	l3_sen	L4
2	DEFENSA	1524	1524 DEFENSA	-58.370994	-34.625658	"1"	SAN TELMO	"22"	V	"53"	I			
3	DEFENSA	1528	1528 DEFENSA	-58.370999	-34.62571	"1"	SAN TELMO	"29"	I					
4	BARTOLOME MITRE	722	MITRE, BARTOLOME 722	-58.376843	-34.60701	"1"	SAN NICOLAS	"105"	V					
5	REGIMIENTO DE PATRICIOS AV.	51	51 REGIMIENTO DE PATRICIOS AV.	-58.370663	-34.630225	"4"	BARRACAS	"93"	I	"70"	V	"74"	I	
6	REGIMIENTO DE PATRICIOS AV.	389	389 REGIMIENTO DE PATRICIOS AV.	-58.37036	-34.633409	"4"	BARRACAS	"10"	I	"22"	I			
7	REGIMIENTO DE PATRICIOS AV.	435	435 REGIMIENTO DE PATRICIOS AV.	-58.370277	-34.634011	"4"	BARRACAS	"24"	I	"46"	V			
8	REGIMIENTO DE PATRICIOS AV.	471	471 REGIMIENTO DE PATRICIOS AV.	-58.370232	-34.634299	"4"	BARRACAS	"39"	V	"93"	I			
9	BARTOLOME MITRE	906	906 MITRE, BARTOLOME	-58.379658	-34.607215	"1"	SAN NICOLAS	"105"	V					
10	REGIMIENTO DE PATRICIOS AV.	673	673 REGIMIENTO DE PATRICIOS AV.	-58.369933	-34.636199	"4"	BARRACAS	"70"	V	"74"	I			

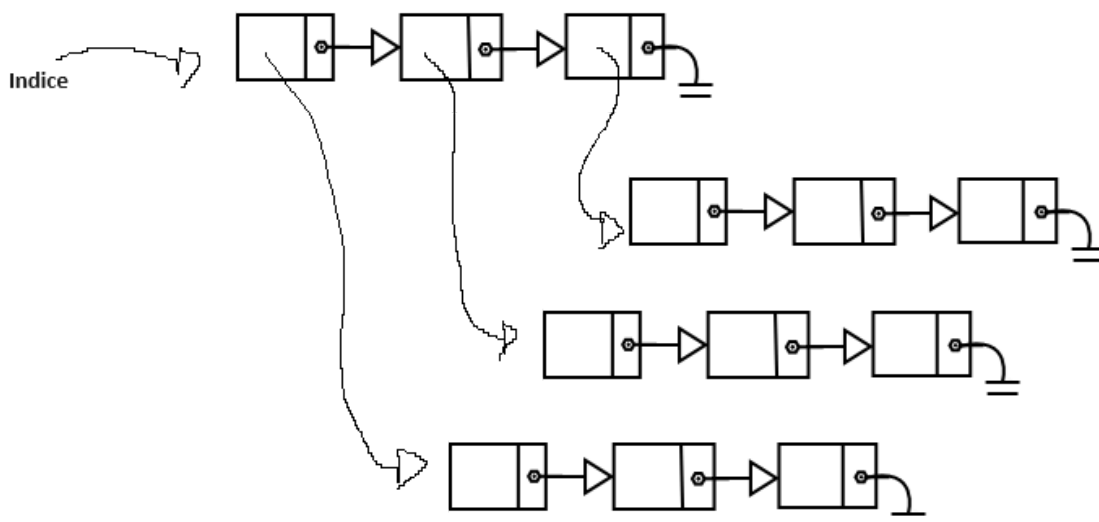
y al verlo como archivo de texto es:

```

CALLE,ALT PLANO,DIRECCION,coord_X,coord_Y,COMUNA,BARRIO,L1,l1_sen,L2,l2_sen,L3,l3_sen,L4,l4_sen,L5,l5_sen,L6,l6_sen
DEFENSA,"1524",1524 DEFENSA,-58.370994,-34.625658,"1",SAN TELMO,"22",V,"53",I,,,,,,,,,
DEFENSA,"1528",1528 DEFENSA,-58.370999,-34.62571,"1",SAN TELMO,"29",I,,,,,,,,,
BARTOLOME MITRE,"722","MITRE, BARTOLOME 722",-58.376843,-34.60701,"1",SAN NICOLAS,"105",V,,,,,,,,,
REGIMIENTO DE PATRICIOS AV.,"51",51 REGIMIENTO DE PATRICIOS AV.,-58.370663,-34.630225,"4",BARRACAS,"93",I,"70",V,"74",I,,,,,
REGIMIENTO DE PATRICIOS AV.,"389",389 REGIMIENTO DE PATRICIOS AV.,-58.37036,-34.633409,"4",BARRACAS,"10",I,"22",I,,,,,,,,,
REGIMIENTO DE PATRICIOS AV.,"435",435 REGIMIENTO DE PATRICIOS AV.,-58.370277,-34.634011,"4",BARRACAS,"24",I,"46",V,,,,,,,,,
REGIMIENTO DE PATRICIOS AV.,"471",471 REGIMIENTO DE PATRICIOS AV.,-58.370232,-34.634299,"4",BARRACAS,"39",V,"93",I,,,,,,,,,
BARTOLOME MITRE,"906","906 MITRE, BARTOLOME",-58.379658,-34.607215,"1",SAN NICOLAS,"105",V,,,,,,,,,
REGIMIENTO DE PATRICIOS AV.,"673",673 REGIMIENTO DE PATRICIOS AV.,-58.369933,-34.636199,"4",BARRACAS,"70",V,"74",I,,,,,
BARTOLOME MITRE,"1208","MITRE, BARTOLOME 1208",-58.383645,-34.60748,"1",SAN NICOLAS,"105",V,,,,,,,,,
REGIMIENTO DE PATRICIOS AV.,"717",717 REGIMIENTO DE PATRICIOS AV.,-58.369854,-34.636722,"4",BARRACAS,"39",V,"93",I,,,,,,,,,
REGIMIENTO DE PATRICIOS AV.,"761",761 REGIMIENTO DE PATRICIOS AV.,-58.369813,-34.636987,"4",BARRACAS,"10",I,"24",I,,,,,,,,,
REGIMIENTO DE PATRICIOS AV.,"787",787 REGIMIENTO DE PATRICIOS AV.,-58.369785,-34.637171,"4",BARRACAS,"22",I,"46",V,,,,,,,,,
BARTOLOME MITRE,"1444","1444 MITRE, BARTOLOME",-58.386906,-34.607741,"1",SAN NICOLAS,"7",V,"8",V,,,,,,,,,
BARTOLOME MITRE,"1464","1464 MITRE, BARTOLOME",-58.387152,-34.607756,"1",SAN NICOLAS,"105",V,"98",I,,,,,,,,,

```

El grupo tiene que formar una estructura de 2 niveles de listas de listas, donde el primer nivel es el Barrio y el segundo nivel la dirección de la parada, copiando la estructura siguiente:



Por teclado deberán hacer un menu para responder las siguientes preguntas:

- 1) Listado de cantidad de paradas por barrio
- 2) Parada más cercana a una coordenada ingresada por teclado
- 3) Listado de paradas de una línea de colectivo
- 4) Listado de cantidad de paradas por linea de colectivo
- 5) Dado un barrio y una línea de colectivo, imprimir las paradas ordenadas por distancia.

Cuestionario

Responder el siguiente Cuestionario:

- 1) ¿Qué es un svn?
- 2) ¿Qué es git?
- 3) ¿Qué es Github?
- 4) ¿Qué es un valgrind?

Normas de entrega

Trabajo práctico grupal: 5 personas. Se deberá elegir un nombre de Grupo.

Reglas generales: respetar el Apéndice A.

El índice se debe implementar utilizando la clase Lista. Todos los objetos deben estar en memoria dinámica.

Se deberá subir un único archivo comprimido al campus, en un link que se habilitará para esta entrega. Este archivo deberá tener un nombre formado de la siguiente manera:

Nombre de Grupo-TP2.zip

Deberá contener los archivos fuentes (no los binarios), el informe del trabajo realizado, las respuestas al cuestionario, el manual del usuario y el manual del programador (Todo en el mismo PDF).

La fecha de entrega vence el día viernes 22/06/24 a las 23.59hs.

Se evaluará: funcionalidad, eficiencia, algoritmos utilizados, buenas prácticas de programación, modularización, documentación, gestión de memoria y estructuras de datos.

Apéndice A

- 1) Usar las siguientes convenciones para nombrar identificadores.
 - a) Clases y structs: Los nombres de clases y structs siempre deben comenzar con la primera letra en mayúscula en cada palabra, deben ser simples y descriptivos. Se concatenan todas las palabras. Ejemplo: Coche, Vehiculo, CentralTelefonica.
 - b) Métodos y funciones: Deben comenzar con letra minúscula, y si está compuesta por 2 o más palabras, la primera letra de la segunda palabra debe comenzar con mayúscula. De preferencia que sean verbos. Ejemplo: arrancarCoche(), sumar().
 - c) Variables y objetos: las variables siguen la misma convención que los métodos. Por Ejemplo: alumno, padronElectoral.
 - d) Constantes: Las variables constantes o finales, las cuales no cambian su valor durante todo el programa se deben escribir en mayúsculas, concatenadas por "_". Ejemplo: ANCHO, VACIO, COLOR_BASE.
- 2) El lenguaje utilizado es C++, esto quiere decir que se debe utilizar siempre C++ y no C, por lo tanto una forma de darse cuenta de esto es no incluir nada que tenga .h, por ejemplo `#include <iostream>` .
- 3) No usar sentencias 'using namespace' en los .h, solo en los .cpp. Por ejemplo, para referenciar el tipo string en el .h se pone `std::string`.
- 4) No usar 'and' y 'or', utilizar los operadores '&&' y '||' respectivamente.
- 5) Compilar en forma ANSI. Debe estar desarrollado en linux con eclipse y g++. Utilizamos el estándar C++98.
- 6) Chequear memoria antes de entregar. No tener accesos fuera de rango ni memoria colgada.
- 7) Si el trabajo práctico requiere archivos para procesar, entregar los archivos de prueba en la entrega del TP. Utilizar siempre rutas relativas y no absolutas.
- 8) Entregar el informe explicando el TP realizado, manual de usuario y manual del programador.
- 9) Comentar el código. Todos los tipos, métodos y funciones deberían tener sus comentarios en el .h que los declara.
- 10) Modularizar el código. No entregar 1 o 2 archivos, separar cada clase o struct con sus funcionalidades en un .h y .cpp
- 11) No inicializar valores dentro del struct o .h.
- 12) El tablero en el TP 2 tiene que ser listas de listas.

13) Si cualquier estructura de control tiene 1 línea, utilizar {} siempre, por ejemplo:

```
for(int i = 0; i < 10; i++) {  
    std::cout << i;  
}
```

