

Trabajo Práctico Único

Objetivos:

- Integrar los contenidos de implementación de estructuras de datos.
- Reforzar la utilización del lenguaje de programación java para la implementación de aplicaciones de escritorio.
- Desarrollar una aplicación que reúna, proceso de volumen considerable de datos, respaldo en estructuras de datos propias y presentación de dichos datos en una aplicación de escritorio.

Los temas que se evalúan en este trabajo práctico son los que se detallan a continuación: Programación Orientada a Objetos en Java, Estructuras de datos en Java, procesamiento de archivos de texto, e Interfaces visuales de usuario.

Caso de Estudio:

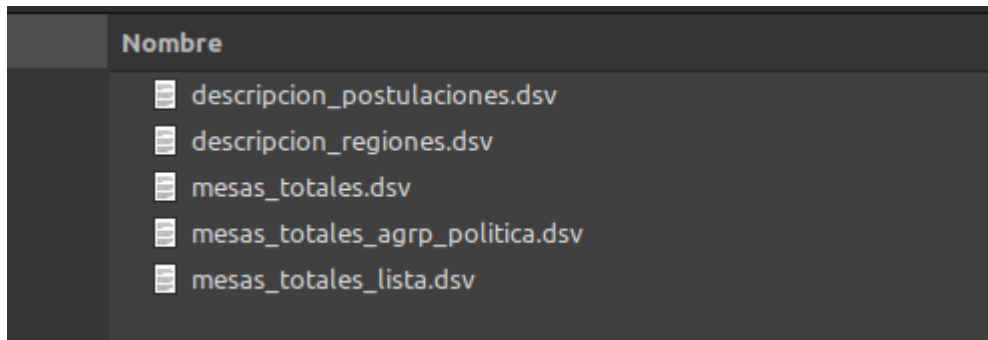
Durante el año 2019 se llevaron a cabo elecciones en Argentina en las que se eligen distintos cargos de representantes a lo largo y a lo ancho de todo el país. Durante el año se eligieron diversos cargos en los distintos distritos y de acuerdo a la autonomía propia de cada uno de ellos, sin embargo hay una elección en la que todos deben concordar que es la elección de presidente de la nación, dicha elección tiene por ley distintas instancias como son las PASO, la elección GENERAL y el Ballotage en caso de ser necesario, pero no vamos a ocuparnos de dichas instancias en el presente trabajo para nosotros van a ser indistintas.

Para cada instancia de elección se realiza un escrutinio en el que se validan las urnas y las actas de conteo de votos para todo el país y al finalizar el mismo los resultados son publicados de forma que queden disponibles para cualquier ciudadano Argentino que desee revisarlos o procesarlos para su análisis y justamente este análisis es el que proponemos para el trabajo práctico.



Del siguiente link: <https://datos.gob.ar/dataset/otros-resultados-provisorios-elecciones-primarias-abiertas-simultaneas-obligatorias-paso-2019> se pueden descargar los mencionados resultados para las elecciones PASO próximas pasadas y los archivos que de allí se descargan van a ser nuestro objeto de trabajo.

Al descargar y descomprimir el archivo resultados-provisorios-paso-2019.zip que al que lleva el link, encontraremos los siguientes archivos:



cada uno de ellos contiene información del escrutinio, aunque nosotros solo vamos a utilizar 3:

- descripcion_postulaciones.dsv
- descripcion_regiones.dsv
- mesas_totales_agrp_politica.dsv.

En el caso del primero descripcion_postulaciones.dsv es el archivo que contiene todos los cargos que se eligieron simultáneamente en la elección y de todos esos cargos solo nos interesará 1 que es el cargo de “Presidente y Vicepresidente de la República”.

El archivo es un archivo de texto separado por pipes ‘|’ y cuya primera línea tiene información del significado de cada segmento de línea de las líneas de datos:

```
CODIGO_CATEGORIA|NOMBRE_CATEGORIA|CODIGO_AGRUPACION|NOMBRE_AGRUPACION|CODIGO_LISTA|NOMBRE_LISTA
0008010000000000|Diputados Ciudad Autónoma de Buenos Aires|01-504|CONSENSO
```

De este archivo nos va a interesar extraer todas las agrupaciones políticas que se presentan para el cargo de “Presidente y Vicepresidente de la República”

```
897 0001000000000000|Intendente, concejales, consejeros escolares Buenos Aires - Partido Azul|137|CONSENSO FEDERAL|0001000000000000.137.A.0001000000000000
898 0001000000000000|Presidente y Vicepresidente de la República|135|JUNTOS POR EL CAMBIO|0001000000000000.135.A.0001000000000000
899 0001000000000000|Presidente y Vicepresidente de la República|131|FRENTE NOS|0001000000000000.131.A.0001000000000000
900 0001000000000000|Presidente y Vicepresidente de la República|13|MOVIMIENTO AL SOCIALISMO|0001000000000000.13.A.0001000000000000
901 0001000000000000|Presidente y Vicepresidente de la República|136|FRENTE DE TODOS|0001000000000000.136.A.0001000000000000
902 0001000000000000|Presidente y Vicepresidente de la República|36|PARTIDO AUTONOMISTA|0001000000000000.36.A.0001000000000000
903 0001000000000000|Presidente y Vicepresidente de la República|132|FRENTE PATRIOTA|0001000000000000.132.A.0001000000000000
904 0001000000000000|Presidente y Vicepresidente de la República|133|FRENTE DE IZQUIERDA Y DE TRABAJADORES - UNIDAD|0001000000000000.133.A.0001000000000000
905 0001000000000000|Presidente y Vicepresidente de la República|137|CONSENSO FEDERAL|0001000000000000.137.A.0001000000000000
906 0001000000000000|Presidente y Vicepresidente de la República|87|UNITE POR LA LIBERTAD Y LA DIGNIDAD|0001000000000000.87.A.0001000000000000
907 0001000000000000|Presidente y Vicepresidente de la República|57|MOVIMIENTO DE ACCION VECINAL|0001000000000000.57.A.0001000000000000
908 0006020000100000|Diputados Provinciales Buenos Aires - Sección 1|135|JUNTOS POR EL CAMBIO|0006020000100000.135.A.0006020000100000
```

Será necesario recuperar las filas mencionadas guardando específicamente código de categoría (String), el código de agrupación (int) y el nombre de la agrupación (String). Y almacenar dichos datos en alguna estructura de

datos en memoria para su futura utilización. Notar que el código de categoría se repite para todas las filas de este cargo y es único respecto de los demás cargos.

El siguiente archivo para procesar será el archivo `descripción_regiones.dsv`, que también es un archivo de texto separado por pipes '|' y cuya estructura de campos es la siguiente:

```
1 CODIGO_REGION|NOMBRE_REGION|
2 06021112 DE OCTUBRE
```

el código de región es una cadena de caracteres que para comprender su codificación hace falta algo más de información, a los efectos de la geografía electoral el país está dividido en 24 distritos, las 23 provincias y la Ciudad Autónoma de Buenos Aires, cada distrito está dividido en secciones electorales y cada sección en circuitos electorales por esto el código de región puede corresponder a cualquiera de las 3 secciones antes mencionadas y se interpreta como sigue:

- Si el código de región es de 2 dígitos, corresponde a un distrito y el nombre será el nombre del distrito
- Si el código de región es de 5 dígitos corresponde a una sección electoral y el nombre es el nombre asignado a la misma, notar que dentro de los 5 dígitos de la sección los 2 primeros son el distrito al que pertenece y los 3 últimos son el identificador de la sección.
- Finalmente si el código de región es de 11 dígitos corresponde a un circuito electoral y el nombre será entonces el nombre asignado al circuito, nuevamente notar que en los 11 dígitos los 5 primeros corresponden al código de región de la sección electoral a la que pertenece el circuito y los restantes que en este caso serán alfanuméricos serán del circuito electoral.

```
5856 16007000031|CIRCUITO ÑORQUINCO
5857 01|CIUDAD AUTÓNOMA DE BUENOS AIRES
5858 10012|COCHINOCA
```

por ejemplo, en las filas de la imagen se puede ver en la primera fila el circuito Ñorquinco que pertenece a la sección electoral 007 del distrito 16 – Río Negro, en la segunda fila el distrito 01 Ciudad Autónoma de Buenos Aires y en la 3era fila la sección electoral 012 CochinoCA correspondiente al distrito 10 – Jujuy.

Nuevamente será necesario cargar los datos de cada una de las entidades mencionadas y almacenarlas de forma que luego puedan ser accedidas a medida que vayan siendo requeridas al procesar el conteo.

Una vez que toda la información anterior esté cargada en memoria en estructuras de datos que permitan su acceso de forma rápida de acuerdo al identificador que corresponda podremos comenzar a procesar el archivo final que es el que tiene la información de los conteos de votos por agrupación política, el archivo `mesas_totales_agrp_politicas.dsv`, dicho archivo como los anteriores también es un archivo de texto separado por pipes '|' y su estructura contiene los siguientes datos:

```
1 CODIGO_DISTrito|CODIGO_SECCION|CODIGO_CIRCUITO|CODIGO_MESA|CODIGO_CATEGORIA|CODIGO_AGRUPACION|VOTOS_AGRUPACION
2 0110100110100100000110100100001X100010000000000011311
```

**Trabajo Único: Tablas de Dispersión
(implementación por direccionamiento abierto).**

Ciclo 2020

los 3 primeros campos en la estructura son distrito, sección y circuito de acuerdo a lo revisado en el archivo de regiones, el cuarto campo corresponde al número de mesa que incluye en su estructura el distrito y la sección y luego es un número correlativo por distrito, el quinto campo corresponde al código de categoría o código de cargo de acuerdo a lo revisado en el archivo de descripción de postulaciones, el sexto campo es el número de agrupación política de los recuperados también en el archivo de postulaciones y finalmente el último campo corresponde al conteo de votos en esa mesa, para esa agrupación política para esa categoría de cargo.

A continuación se muestra un esquema de referencias entre los archivos previamente analizados y el archivo de conteos de mesa:

```
5856 16007000031 CIRCUITO NORQUINCO
5857 011 CIUDAD AUTÓNOMA DE BUENOS AIRES
5858 10012 COCHINOCA
```

```
897 0001000000000000 Intendente, Concejales, Consejeros Escolares Buenos Aires - Partido Azul
898 0001000000000000 Presidente y Vicepresidente de la República 135 JUNTOS POR EL CAMBIO 00010
899 0001000000000000 Presidente y Vicepresidente de la República 131 FRENTE NOS 000100000000000
900 0001000000000000 Presidente y Vicepresidente de la República 13 MOVIMIENTO AL SOCIALISMO 000
901 0001000000000000 Presidente y Vicepresidente de la República 156 FRENTE DE TODOS 00010000000
902 0001000000000000 Presidente y Vicepresidente de la República 36 PARTIDO AUTONOMISTA 0001000
903 0001000000000000 Presidente y Vicepresidente de la República 132 FRENTE PATRIOTA 00010000000
904 0001000000000000 Presidente y Vicepresidente de la República 133 FRENTE DE IZQUIERDA Y DE T
905 0001000000000000 Presidente y Vicepresidente de la República 137 CONSENSO FEDERAL 000100000
906 0001000000000000 Presidente y Vicepresidente de la República 87 UNITE POR LA LIBERTAD Y LA
907 0001000000000000 Presidente y Vicepresidente de la República 57 MOVIMIENTO DE ACCION VECINA
908 0006020000100000 Diputados Provinciales Buenos Aires - Sección 1 1135 JUNTOS POR EL CAMBIO 0
```

```
29583 01 01001 01001000019 0100100517X 000901001000000 01-504 13
29584 01 01001 01001000019 0100100517X 000901001000000 13 3
29585 01 01001 01001000019 0100100517X 000901001000000 512
29586 01 01002 01002000020 0100200518X 000100000000000 13 0
29587 01 01002 01002000020 0100200518X 000100000000000 131 1
29588 01 01002 01002000020 0100200518X 000100000000000 132 0
29589 01 01002 01002000020 0100200518X 000100000000000 133 6
29590 01 01002 01002000020 0100200518X 000100000000000 135 124
```

Algunos elementos a notar:

- En el archivo de mesas hay una fila por cargo, y agrupación dentro de esa mesa con lo cual una mesa va a tener n filas en el archivo de conteos tantas como cargos y agrupaciones fueran posibles ser elegidos en esa mesa.
- Notar también que de entre todos los conteos solo nos interesan los correspondientes al cargo “Presidente y Vicepresidente de la República” con lo cual de los algo más de 3 millones de filas en el archivo vamos a filtrar bastante volumen por este punto.
- Por otro lado, también tener en cuenta que al hacer referencia a los códigos de distrito, sección y circuito el archivo de totales de mesa toma los códigos completos con la inclusión anterior, revisar esto de acuerdo a como ud. haya implementado los valores de códigos para luego ser identificados.
- Finalmente notar que el proceso debe realizar el conteo de acuerdo con el código de agrupación guardando la información de la mesa (el número de mesa puntualmente para luego poder buscarla).

En base a lo descripto anteriormente usted debe desarrollar las clases necesarias respetando las reglas del paradigma orientado a objetos en lenguaje Java para soportar los siguientes requerimientos de acuerdo a las etapas propuestas de implementación:

1. **Etapla 1:** Estructura de Datos.

1.1. Siguiendo el modelo de implementación del concepto de Tabla Hash con Listas de Desborde, se solicita ahora implementar una clase TSB_OAHashtable (por TSB Open Addressing Hash Table), que represente una Tabla Hash pero con estrategia de Direccionamiento Abierto para la resolución de colisiones.

La clase debe ser implementada en forma rigurosa, siguiendo el modelo ya presentado para la clase TSBHashtable. Esto implica:

- 1.1.1. Implementar la interface Map<K, V> y desde ella, los mismos métodos que se implementaron para la clase TSBHashtable.
- 1.1.2. Definir dentro de la clase TSB_OAHashtable una clase interna Entry que implemente la interface Map.Entry<K, V> para representar a cada par que se almacene en la tabla.
- 1.1.3. Definir dentro de la clase TSB_OAHashtable las tres clases internas para gestionar las vistas stateless de claves, de valores y de pares de la tabla, incluyendo a su vez en ellas las clases internas para representar a los iteradores asociados a cada vista.
- 1.1.4. Redefinir en la clase TSB_OAHashtable los métodos equals(), hashCode(), clone() y toString() que se heredan desde Object.
- 1.1.5. Definir en la clase TSB_OAHashtable los métodos rehash() y contains(value) que no vienen especificados por Map, pero son especialmente propios de la clase (emulando a java.util.Hashtable).

1.2. La idea es tomar como modelo al que se presentó para la clase TSBHashtable en la Ficha 10, e implementar los mismos métodos publicos y protegidos de esa clase, más las clases internas citadas. Obviamente, con relación a los atributos y métodos privados, cada grupo de trabajo hará su propia propuesta y diseño.

2. **Etapla 2:** Procesar Archivos

2.1. Implementar el proceso que cargue en memoria las agrupaciones políticas del archivo descripcion_postulaciones.dsv, teniendo en cuenta el filtro para cargar solo una vez cada

agrupación cuando corresponda el código de cargo al de "Presidente y Vicepresidente de la República"

2.2. Implementar un proceso que cargue en memoria las regiones y sus descripciones de acuerdo con la codificación expresada en la documentación del caso.

2.3. Implementar un proceso que realice el conteo en cascada (conteo total, conteo por distrito, conteo por sección, conteo por circuito y finalmente los resultados de la mesa).

La idea de este algoritmo es que para cada línea del archivo `mesas_totales_agrp_politicas.dsv`

1. cargue la línea en memoria
2. realice la acumulación en el contador total de acuerdo con el código de agrupación.
3. busque el distrito por código de distrito y realice el conteo en el distrito de acuerdo con el código de agrupación, si el distrito no existe aún debe agregarlo.
4. busque la sección dentro del distrito por código de sección y realice el conteo en la sección de acuerdo con el código de agrupación, si la sección no existiera aún se debe agregar.
5. busque el circuito dentro de la sección por código de circuito y realice el conteo en el circuito de acuerdo con el código de agrupación, si no existiera el circuito debe agregarse.
6. finalmente busque la mesa por número de mesa dentro del circuito y realice el asiento de la cantidad en la mesa de acuerdo con el código de agrupación, si la mesa no se creó aún se debe agregar.

3. **Etapas 3:** Interfaz de usuario.

3.1. Implementar la interfaz que de respuesta al modelo descripto.

3.2. Prototipo y definiciones

4. **Etapas 4 Bonus:** [Opcional] almacenamiento en base de datos

4.1. Almacenar los datos luego de procesado en una base de datos.

4.2. Cargar los datos de una base de datos al iniciar la aplicación.

Nota: sería ideal que para las etapas 1 y 2 construya tests unitarios que permitan validar el funcionamiento del código desarrollado.