

TP1: Introducción a Diseño de una Solución y Manejo de archivos

El trabajo práctico número 1 tiene fecha de entrega para el día **26/09**. El objetivo de este trabajo práctico es el desarrollo que implique la utilización de los TDAs antes implementados, el diseño de nuevos TDAs necesarios para este desarrollo, y el uso de archivos y flujo de entrada, salida y error.

Introducción

Se acercan las elecciones y no hay tiempo que perder. A pesar de las denuncias en su contra, el gobernador Pólez se postulará para su re-re-re-reelección. Como el actual sistema de votación está siendo cuestionado por denuncias de fraude, el gobernador Pólez ha coordinado la implementación del sistema de boleta única electrónica para darle credibilidad a los comicios venideros. También ha dispuesto otros cambios: en primer lugar, el transporte de las urnas se hará por *Empresa de correos Pólez e hijos*, la impresión de las nuevas boletas únicas la hará *Pólez calcográfica* y el sistema informático para el escrutinio lo desarrollará *Pólez Consulting*.

En este contexto, nos han encargado la implantación de un componente del nuevo sistema de boleta única electrónica: la máquina de impresión de la boleta. Lamentablemente el gobernador Pólez se nos adelantó y ha encargado la fabricación de las máquinas (por licitación directa) al fabricante *Pólez hardware y asociados*, dejándonos el software que se ejecutará sobre la plataforma. Por esta razón, el programa respetará un protocolo de órdenes por entrada estándar (`stdin`).

Consigna

Implementar en Go el sistema de gestión de votos que debe cumplir el siguiente protocolo.

Protocolo

Este programa formará parte de un sistema automatizado. En lugar de presentar un menú con opciones en la pantalla, se esperará que el programa respete un protocolo de comunicación, en el cual se reciben comandos por la entrada estándar (`stdin`). Cada comando recibido debe ser procesado y su resultado debe ser escrito en la salida estándar (`stdout`). **Esto incluye mensajes de error**. Todo debe escribirse por la salida estándar.

Cada línea de la entrada corresponde a un comando, y todos los comandos respetan un determinado formato. El formato estará compuesto del nombre del comando y sus parámetros, separados por espacios, en caso de requerirlos.

Ejecución del programa

El programa debe ejecutarse de la siguiente forma:

```
$ ./rerepolez ARCHIVO_LISTA ARCHIVO_PADRON
```

Donde:

- ARCHIVO_LISTA**: Un archivo .csv donde están las respectivas boletas a votar. Cada línea representa un partido político, que tiene nombre de dicho partido, nombre de el/la candidata a presidente, nombre de el/la candidata a gobernador, y luego el del candidato a intendente.
- ARCHIVO_PADRON**: Un archivo .txt con los documentos habilitados a votar en la mesa. Este archivo tiene un documento por línea. Cada documento escrito en formato numérico, sin puntos.

Al empezar a ejecutar, el programa debe empezar a levantar los archivos y prepara la mesa electoral. Al terminar de preparar todo, debe imprimir por salida estándar (`stdout`) lo que corresponda a la salida según el estado.

Más allá de los ejemplos particulares que mostramos en el enunciado, aseguramos que, tanto en un escenario real como en las pruebas, los números de DNI (válidos) serán en general similares a los reales, y puede usarse esa información a conveniencia.

Salida

- ERROR: Faltan parámetros**: si faltó alguno de los parámetros
- ERROR: Lectura de archivos**: si hubo un error en la lectura de los archivos (o archivos inexistentes, o sin permisos)

Ejemplo:

```
./rerepolez lista_candidatos.csv padron-que-no-existe.csv
ERROR: Lectura de archivos
./rerepolez lista_candidatos.csv -->le falta el otro archivo
ERROR: Faltan parámetros
./rerepolez lista_candidatos.csv padron.csv
<Se queda esperando por comandos>
```

En caso de errores, el programa debe terminar su ejecución.

Comando: **ingresar**

Formato

ingresar <NumeroDNI>

Descripción:

El votante ingresa este comando al momento de entrar en la fila de personas esperando a usar la máquina.

Parámetros

- NumeroDNI: Un número entero positivo mayor a 0 que representa el número de documento del votante.

Salida

- OK**: si no se produjeron errores.
- ERROR: DNI incorrecto**: si el número de DNI no es correcto (es menor o igual a 0, o no es un número).
- ERROR: DNI fuera del padrón**: si el DNI de la persona votando no se encuentra en el padrón de la mesa actual.

En caso de error, simplemente esperar por un nuevo ingresa de un comando.

Ejemplo:

```
ingresar 36109000
OK
ingresar 40100900
OK
ingresar -12345678
ERROR: DNI incorrecto
ingresar 123
ERROR: DNI fuera del padrón
```

Comando: **votar**

Formato

votar <TIPO-VOTO> <NumeroLista>

Descripción

Este comando se utiliza para votar por alguna alternativa. Por simplicidad y evitar que alguien escriba mal el nombre de un partido político, se decidió que los votantes ingresen su número de lista. El número de lista es directamente el número de línea dentro del archivo de partidos que ocupa cada partido. El votante que votará será el que se encuentre primero en la fila (quien haya llegado antes).

Parámetros

Puede llevar 2 tipos de parámetros:

- El **TIPO-VOTO**, que puede ser **Presidente**, **Gobernador** o **Intendente**.
- El número de lista, explicado anteriormente.

Una persona puede votar por diferentes partidos políticos para Gobernador y Presidente, por ejemplo. Al mismo tiempo, si alguien decidiera votar primero para Presidente por un partido, y luego decide cambiar de opinión y votar nuevo para Presidente para otro partido, se sobrescribirá el voto por la nueva alternativa. A cambio, considerar el siguiente comando (**deshacer**).

El número de lista 0 está reservado para impugnar el voto (equivalente a dejar una feta de salame en el sobre). En caso de impugnarse el voto, se impugna completo, no para un partido en particular. El votante puede seguir efectuando votos (sería meter boletas en el mismo sobre), pero ninguno de estos contará. Para des-impugnar el voto, se debe haber deshecho esa operación (nuevamente, ver comando **deshacer**, más adelante).

Salida

- **OK**: si no se produjeron errores.
- **ERROR: Fila vacía**: si no hay nadie esperando para votar.
- **ERROR: Tipo de voto inválido**: si el tipo de voto no es uno de los enunciados anteriormente.
- **ERROR: Alternativa inválida**: si el número de lista es incorrecto (no es una opción válida).
- **ERROR: Votante FRAUDULENTO: <NRO DNI>**: si el DNI de la persona votando **ya emitió y finalizó su voto** (ver **fin-votar**). Se debe indicar también el número de DNI de la persona que está incurriendo en este delito contra la democracia.

En caso de producirse el Error de votante fraudulento, se descarta el padrón de la fila. En caso de los demás errores, simplemente se espera por otro ingreso que sea correcto. En caso que se produzca más de un error a la vez, el error que se debe priorizar al que esté antes en este listado. Es decir, se debe validar en tal orden.

Ejemplo:

```
ingresar 30000000
OK
ingresar 16000000
OK
ingresar 123
OK
votar Presidente 1
OK
votar Intendente 5
OK
votar Gobernador 8
ERROR: Alternativa inválida
votar Presi 5
ERROR: Tipo de voto inválido
votar Gobernador 4
OK
votar Presidente 2
OK
votar DT_de_la_selection -10
ERROR: Tipo de voto inválido
```

El estado del voto del votante con DNI 30.000.000 hasta aquí sería:

```
Presidente: Lista 2
Gobernador: Lista 4
Intentente: Lista 5
```

Comando: deshacer

Formato

deshacer

Descripción

Deshace la última operación de voto del votante actual. Es decir, ningún voto ya finalizado (ver siguiente comando) puede verse alterado por la aplicación de este comando. El voto completo del votante actual queda en el estado exactamente anterior a la última utilización del comando **votar**.

Parámetros

Ninguno.

Salida

- **OK**: si no se produjo ningún error.
- **ERROR: Fila vacía**: si no hay nadie esperando votar
- **ERROR: Votante FRAUDULENTO: <NRO DNI>**: si el DNI de la persona votando **ya emitió y finalizó su voto** (ver **fin-votar**). Nuevamente, se indica el número de DNI de quien incurre en el delito.
- **ERROR: Sin voto a deshacer** si no hay nada que deshacer (el votante actual aún no realizó votaciones, o todas sus votaciones fueron ya deshechas).

En caso de producirse el Error de votante fraudulento, se descarta el padrón de la fila. En caso de los demás errores, simplemente se espera por otro ingreso que sea correcto. En caso que se produzca más de un error a la vez, el error que se debe priorizar al que esté antes en este listado. Es decir, se debe validar en tal orden.

Ejemplo:

Siguiendo con el ejemplo anterior: deshacer

El estado actual del voto del votante con DNI 30.000.000 será:

Presidente:	Lista 1
Gobernador:	Lista 4
Intentente:	Lista 5

La operación que se deshizo fue la de haber votado para presidente a lista 2.

Comando: fin-votar

Formato

fin-votar

Descripción

Termina el proceso de votación para el votante actual en la fila. Se emite su voto completo, considerando su alternativa elegida para Presidente, Gobernador e Intendente. Si no eligió alguna alternativa para uno o más tipos, se contarán como votos en blanco.

Al finalizar el voto, se añadirán al conteo de las alternativas electas para Presidente, Gobernador e Intendente correspondientes, incluyendo el voto en blanco. En caso de haberse impugnado el voto, no se contará para ninguna, y simplemente se irá a un conteo de votos impugnados. En cualquier caso, el votante dejará de estar en la fila, para que pueda votar el siguiente de la misma.

Parámetros

Ninguno.

Salida

- OK: si no se produjo ningún error.
- ERROR: Fila vacía: si no hay nadie esperando para votar.
- ERROR: Votante FRAUDULENTO: <NRO DNI>: si el DNI de la persona votando ya emitió y finalizó su voto (ver fin-votar).
Mismo a lo mencionado anteriormente respecto al número de DNI.

En caso de producirse el Error de votante fraudulento, se descarta el padrón de la fila. En caso de los demás errores, simplemente se espera por otro ingreso que sea correcto.

Ejemplo:

Continuando el ejemplo anterior:

deshacer
fin-votar

El voto finalmente emitido será:

Presidente:	Lista 1
Gobernador:	Voto en Blanco
Intentente:	Lista 5

Fin de ejecución del programa

Una vez que se hubiera terminado el ingreso por entrada estándar (es decir, stdin finalizó), se debe imprimir el resultado de los comicios para esta mesa con el formato mostrado a continuación. En caso de haber alguien “a medio votar”, así como otros votantes en la fila esperando, se deberá imprimir ERROR: Ciudadanos sin terminar de votar y se descartarán sus votos, considerando que no cumplieron con su deber civil yendo a votar los últimos 15 minutos. De ser posible, se les cobrará la correspondiente multa allí mismo, pero esto queda fuera del alcance del trabajo práctico. Esto será llevado a cabo por Pólez Cobraciones SRL.

Salida

Presidente:
Votos en Blanco: Z votos
<nombre del partido 1> - Postulante: X votos
<nombre del partido 2> - Postulante: Y votos
...
Gobernador:
Votos en Blanco: Z2 votos
<nombre del partido 1> - Postulante: X2 votos
<nombre del partido 2> - Postulante: Y2 votos
...
Intendente:
Votos en Blanco: Z3 votos
<nombre del partido 1> - Postulante: X3 votos
<nombre del partido 2> - Postulante: Y3 votos
...
Votos impugnados: W votos

Ejemplo integrador:

Para este ejemplo integrador, que el archivo de boletas es:

Frente para la Derrota,Alan Información,Ignacio Líneas,Esteban Federico Pre,Jesús Sintasi,Javier Colisión,Yennifer Woolite ++A,Ignacio Gundersen,Emisor Bello,Daniela Peligro

Y el Padrón electoral es el siguiente:

12345678
98765432
10000000
99999999

El ejemplo:

```
ingresar 12345678
OK
ingresar 123
ERROR: DNI fuera del padrón
ingresar 123a
ERROR: DNI incorrecto
ingresar 12345678
OK
ingresar 98765432
OK
ingresar 100000000
OK
votar Presidente 1
OK
votar Presidente 4
ERROR: Alternativa inválida
votar Gobernador 3
OK
votar Intendente 1
OK
votar Intendente 3
OK
deshacer
OK
votar RectorDeLaUBA 1
ERROR: Tipo de voto inválido
fin-votar
OK
votar Presidente 1
ERROR: Votante FRAUDULENTO: 12345678
deshacer
ERROR: Sin voto a deshacer
votar Presidente 0
OK
deshacer
OK
votar Presidente 3
OK
votar Gobernador 3
OK
votar Intendente 3
OK
fin-votar
OK
fin-votar
OK
votar Presidente 1
ERROR: Fila vacía
fin-votar
ERROR: Fila vacía
```

Al terminar el programa, la salida final sería la siguiente:

Presidente:
Votos en Blanco: 1 voto
Frente para la Derrota - Alan Informaión: 1 voto
Pre - Jesús Sintasi: 0 votos
++A - Ignacio Gundersen: 1 voto
Gobernador:
Votos en Blanco: 1 voto
Frente para la Derrota - Ignacio Líneas: 0 votos
Pre - Javier Colsión: 0 votos
++A - Emisor Bello: 2 votos
Intendente:
Votos en Blanco: 1 voto
Frente para la Derrota - Esteban Federico: 1 voto
Pre - Yennifer Woolite: 0 votos
++A - Daniela Peligro: 1 voto
Votos Impugnados: 0 votos

Notar que en todos los casos, si la cantidad de votos es 1, se debe emitir “1 voto”, en singular.

Archivos provistos por el curso

Como se menciona anteriormente, este trabajo práctico tiene como objetivo tanto la utilización de TDAs previamente implementados, así como diseñar nuevos. En este caso, como primera incursión, les proveemos nosotros de un diseño sugerido para implementar, con un TDA **Votante**, un **struct Voto** (que no es un TDA), y un TDA **Partido**. Proveemos también la firma de las primitivas/métodos, y la estructura de sus respectivas implementaciones, pero sin la implementación de las primitivas. Esta deberían completarla ustedes. El objetivo de esto es guiarlos a implementar un buen diseño. Pueden modificar el diseño propuesto como a ustedes mejor les parezca. En futuros TP's no se les dará diseño alguno, y esperamos que diseñen correctamente, implementando los TDAs que les parezca correctos según la necesidad. El adjunto en [el sitio de descargas](#) contiene las carpetas **diseno_alumnos.zip** y **pruebas.zip**.

También se les otorgará un archivo con la definición de los errores antes mencionados (como errores de Go). En caso de haber alguna discrepancia entre el enunciado y dicho archivo (respecto a alguna mayúscula o lo que fuere), será válido lo indicado en el archivo, salvo que se indique lo contrario.

Aparte de esto, les brindaremos unos archivos que funcionarán de prueba. Estos **no** son como las pruebas anteriores que vinieron ejecutando, sino que serán pruebas de Input/Output (IO, o entrada/salida). Para ejecutar dichas pruebas, deben descomprimir el archivo de pruebas y, previa compilación del código del TP mediante, ejecutar (estando parados en el directorio donde se encuentren las pruebas):

```
$ ./pruebas.sh <RUTA_AL_EJECUTABLE_TP>
```

Donde **RUTA_AL_EJECUTABLE_TP** es la ruta del ejecutable que hayan conseguido tras la compilación de su código. La ruta puede ser relativa al directorio actual, o absoluta.

Criterios de aprobación

El código entregado debe ser claro y legible y ajustarse a las especificaciones de la consigna. Debe compilar sin advertencias y correr sin errores.

La entrega incluye, obligatoriamente, los siguientes archivos de código:

- El código implementado específicamente para la implementación de este TP.
- Cualquier otro código que fuera necesario. Aclaración: en caso de utilizar código de TDAs implementados, y evitar tener que copiar y pegar dicho código dentro de la carpeta para que ejecute correctamente, recomendamos tener el módulo de TDAs y el del TP1 tal ([como se explica en el video de cómo armar módulos para las diferentes entregas](#)).
- un archivo *go.mod* que permita compilar y ejecutar el módulo en cuestión. El nombre del módulo debería ser *rerepolez*.

La entrega se realiza únicamente en forma digital a través del [sistema de entregas](#), con todos los archivos mencionados en un único archivo ZIP.