

# Algoritmos y Programación II (75.41/95.15)

## Trabajo práctico n.º 1

Segundo cuatrimestre 2015

### 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.

## Consigna

Implementar en C 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).

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.

## Comando: abrir

### Formato

```
abrir ARCHIVO_LISTA ARCHIVO_PADRON
```

### Descripción

Representa la apertura de la mesa electoral.

### Parámetros

- ARCHIVO\_LISTA: Un archivo .csv donde están las respectivas boletas a votar.
- ARCHIVO\_PADRON: Un archivo .csv con los documentos habilitados a votar en la mesa.

## Salida

- OK: si no se produjeron errores.
- ERROR1: si hubo un error en la lectura de los archivos (o archivos inexistentes).
- ERROR2: si la mesa ya estaba previamente abierta.

## Ejemplo:

```
abrir lista_candidatos.csv padron.csv
OK
abrir lista_candidatos.csv -->le falta el otro archivo
ERROR1
abrir lista_candidatos.csv padron.csv
ERROR2
```

## Comando: ingresar

### Formato

ingresar DNI número

### Descripción:

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

### Parámetros

- DNI: El tipo de documento del votante. Puede ser
  - DNI (Documento Nacional de Identidad – Tapa Verde – Original),
  - DNID (Documento Nacional de Identidad – Tapa Verde – Duplicado),

- DNIT (Documento Nacional de Identidad – Tapa Verde - Triplicado),
  - DNIC (Documento Nacional de Identidad – Tapa Verde – Cuatriplicado),
  - DNI-EA (Documento Nacional de Identidad – Tapa Celeste – Original),
  - DNI-EB (Documento Nacional de Identidad – Tapa Celeste – Duplicado),
  - LE (Libreta de Enrolamiento)
  - LC (Libreta Cívica).
- número: Un número entero positivo mayor a 0 que representa el número de documento del votante.

## Salida

- OK: si no se produjeron errores.
- ERROR3: si la mesa no está abierta.
- ERROR4: si el número de DNI es menor o igual a 0.

## Ejemplo:

```

ingresar DNI 36109000
OK
ingresar DNID 40100900
OK
ingresar DNI-EA -12345678
ERROR4

```

## Comando: votar

### Formato

```
votar ["inicio"] | <IDPartido> | ["deshacer"] | "fin"
```

## Descripción

Este comando se usa para abrir un proceso de votación, elegir un candidato para un cargo o deshacer una elección realizada previamente.

## Parámetros

Puede llevar 4 tipos de parámetros:

### “inicio” (palabra clave)

Es el primer comando y el que inicia un ciclo de cargos para votar.

**Salida** Primero se validará el DNI del votante contra el padrón. Si el votante ya hubiese votado antes o no se encontrase en el padrón, se devolverá un código de error.

Si el votante esta habilitado, se debe mostrar un listado de candidatos, junto a un ID que el votante usará para decidir, y el cargo de presidente (es el primer cargo que se elige).

```
>>> votar inicio
Cargo: Presidente
1: Frente para la Derrota: Alan Información
2: Pre: Jesus de Nazaret
3: ++A (Mas-Mas-sa): Ignacio Gundersen
```

En caso de error devolverá alguno de los siguientes códigos:

- ERROR3: si la mesa no está abierta.
- ERROR5: si la persona ya votó antes.
- ERROR6: si el número y tipo de documento no está en el archivo `padrón.csv`.
- ERROR7: en caso de que no existan votantes en espera.
- ERROR10: en cualquier otro caso no contemplado.

### ***IDPartido***

En respuesta al comando anterior, se elige un ID de partido como voto a un cargo.

**Salida** Debe mostrar un mensaje “OK”, y un nuevo listado de candidatos, junto a un ID que el votante usará para decidir, y el siguiente cargo en la decisión, siendo el orden Presidente -> Gobernador -> Intendente

```
>>> votar 1
OK
Cargo: Gobernador
1: Frente para la Derrota: Ignacio Lineas
2: Pre: Javier Colision
3: ++A (Mas-Mas-sa): Emisor Bello
```

### **“deshacer” (palabra clave)**

En cualquier paso de la votación, se puede deshacer lo elegido en el paso inmediatamente anterior

**Salida** Debe mostrar un mensaje “OK”, y el listado de candidatos, junto a un ID que el votante usará para volver a decidir.

```
>>> votar deshacer
OK
Cargo: Presidente
1: Frente para la Derrota: Alan Información
2: Pre: Jesus de Nazaret
3: ++A (Mas-Mas-sa): Ignacio Gundersen
```

En caso de error devolvera alguno de los siguientes códigos:

- ERROR3: si la mesa no está abierta o no hay un voto en proceso.
- ERROR8: en caso de que no existan operaciones para deshacer.
- ERROR10: en caso de comando inválido.

**“fin” (palabra clave)**

Cierra el proceso de votación, grabando el voto.

**Salida** Debe mostrar un mensaje “OK” o alguno de los siguientes códigos de error:

- ERROR3: si la mesa no está abierta.
- ERROR9: en caso de que no se hayan elegido todas las categorías.
- ERROR10: en cualquier otro caso no contemplado.

```
>>> votar fin
OK
```

### Ejemplo integrador:

```
>>> votar inicio
Cargo: Presidente
1: Frente para la Derrota: Alan Información
2: Pre: Jesus de Nazaret
3: ++A (Mas-Mas-sa): Ignacio Gundersen
```

```
>>> votar 1
OK
Cargo: Gobernador
1: Frente para la Derrota: Ignacio Lineas
2: Pre: Javier Colision
3: ++A (Mas-Mas-sa): Emisor Bello
```

```
>>> votar 2
OK
Cargo: Intendente
1: Frente para la derrota: Federico Besante
2: Pre: Yenifer Woolite
3: ++A (Mas-Mas-sa): Daniela Peligro
>>> votar 2
```

```
OK
>>> votar deshacer
OK
Cargo: Intendente
1: Frente para la derrota: Federico Besante
2: Pre: Yenifer Woolite
3: ++A (Mas-Mas-sa): Daniela Peligro

>>> votar 1
OK
>>> votar fin
OK
```

En este ejemplo, el votante eligió el partido 1 para Presidente e Intendente, y el partido 2 para Gobernador.

Si alguno de los pasos se ejecuta en el orden incorrecto (por ejemplo, **votar fin** antes de un **votar inicio**) se devolverá **ERROR10** y se continuará en la espera de un comando válido. No se descarta lo votado hasta el momento. Por simplicidad, no se permite votar en blanco.

## Comando: cerrar

### Formato

`cerrar`

### Descripción

Cierra la máquina de votación, impidiendo que se emitan nuevos votos.

### Parámetros

No lleva ninguno.



## Salida

Debe mostrar un resumen de los votos a favor de cada partido político y por cada cargo, en el siguiente formato:

```
<nombre del partido 1>:  
Presidente: X votos  
Gobernador: Y votos  
Intendente: Z votos
```

```
<nombre del partido 2>:  
Presidente: X votos  
Gobernador: Y votos  
Intendente: Z votos
```

En caso de error devolverá alguno de los siguientes códigos:

- ERROR11: En caso de que aún queden votantes ingresados sin emitir su voto

## Ejemplo:

```
>>> cerrar  
Frente para la derrota:  
Presidente: 1 votos  
Gobernador: 1 votos  
Intendente: 0 votos
```

```
Pre:  
Presidente: 0 votos  
Gobernador: 0 votos  
Intendente: 0 votos
```

```
++A (Mas-Mas-sa):  
Presidente: 0 votos  
Gobernador: 0 votos  
Intendente: 1 votos
```

## Pruebas

Junto con la especificación se provee de pruebas automáticas para el programa completo. Estas pruebas serán de utilidad para revisar que el programa cumpla con la especificación del protocolo de entrada/salida. Para la aprobación del Trabajo Práctico es requisito que el programa implementado pase todas las pruebas.

## Criterios de aprobación

A continuación describimos criterios y lineamientos que deben respetarse en el desarrollo del trabajo.

## Utilización de estructuras de datos

Para la realización de este trabajo es necesario utilizar las estructuras de datos vistas en clase (vector dinámico, pila, cola, lista), además de crear las estructuras adicionales que se consideren necesarias. Todas las estructuras deben estar implementadas de la forma más genérica posible y correctamente documentadas.

## Programa

El programa debe cumplir los siguientes requerimientos:

- Debe estar adecuadamente estructurado y modularizado, utilizando funciones definidas de la forma más genérica posible, sin caer en lo trivial.
- El código debe ser claro y legible.
- El código debe estar comentado y las funciones definidas, adecuadamente documentadas.
- El programa debe compilar sin advertencias ni mensajes de error, debe correr sin pérdidas de memoria, uso de valores sin inicializar, o errores en general. Es decir que, el programa debe correr en valgrind sin errores.

- Además, claro, debe satisfacer la especificación de la consigna y pasar todas las pruebas automáticas.

## Informe

El informe deberá consistir de las siguientes partes:

- Carátula con la información del alumno/a y el ayudante asignado (en caso de saberlo de antemano).
- Análisis y diseño: Describir la solución elegida para resolver cada problema propuesto, y cómo se lleva a cabo. En particular, mencionar cómo es el flujo del programa, qué algoritmos y estructuras de datos se utilizan y por qué, y cuál es el orden de ejecución en tiempo y espacio de cada operación.
- Implementación: Incluir aquí todo el código fuente utilizado (en formato monoespaciado, para facilitar su lectura). También opcionalmente, toda explicación adicional que consideren necesaria, referencias utilizadas, dificultades encontradas, cambios o mejoras que se podrían hacer a futuro y conclusiones. El informe debe estar lo más completo posible, con presentación y formato adecuados. Por ejemplo, este enunciado cumple con los requerimientos de un informe bien presentado.

## Entrega

El trabajo consiste en:

- El informe impreso.
- El informe digital, en formato `.pdf`. Para el código fuente utilizar una tipografía `monoespacio`.
- Una versión digital de todos los archivos de código fuente, separados del informe, en un archivo comprimido (`.zip` o `.tar.gz`). Los dos últimos deben enviarse a la dirección `tps.7541rw@gmail.com`, colocando como asunto:

TP1 - Padrón 1 - Apellido 1 - Padrón 2 - Apellido 2

Se aclara que por código fuente se entiende todos los archivos .h y .c, el archivo Makefile para poder compilar, y todos los archivos adicionales que sean necesarios para ejecutar el programa. No deben entregarse nunca archivos .o u otros archivos compilados.

Este trabajo práctico se desarrolla en forma **grupal**. El informe impreso debe entregarse en clase. El plazo de entrega vence el Viernes 16 de Octubre de 2015.