

# Trabajo Práctico - Etapa 1

## *Voto Electrónico*

Martín Hernán Gómez, *Padrón Nro. 85.780*  
martinhgomez@yahoo.com.ar

Ignacio Marambio Catán, *Padrón Nro. 82.694*  
ignacio.marambio@gmail.com

Martín Eduardo Quiroz, *Padrón Nro. 86.012*  
martinedq@yahoo.com.ar

Daniel Shlufman, *Padrón Nro. 88.040*  
incorporado@gmail.com

Lucas Damian Tarcetti, *Padrón Nro. 87.165*  
lucas.tarcetti@yahoo.com.ar

2do. Cuatrimestre de 2011  
75.06 Organización de Datos – Cátedra Lic. Arturo Servetto  
Facultad de Ingeniería, Universidad de Buenos Aires

26/10/11

## Índice

## 1. Introducción

El trabajo consiste crear una aplicación capaz de mantener un sistema de voto electrónico.

### 1.1. Objetivos

El objetivo de este trabajo práctico es que la aplicación sea capaz de mantener información sobre las entidades que componen el sistema de votaciones (votantes, elecciones, candidatos, etc.), y de proveer la posibilidad a un votante de emitir su voto para la correspondiente elección.

## 2. Documentación técnica

### 2.1. Elección del lenguaje de programación

La resolución del trabajo práctico debe ser realizada en plataforma Linux y en lenguaje C++, aprovechando el uso de la programación orientada a objetos.

### 2.2. Entidades

- Distrito ((distrito)i)
- Votante ((DNI)i, NombreyApellido, clave, domicilio, (distrito)ie, ((eleccion)ie)\*)
- Eleccion ((fecha, (cargo)ie)i, ((distrito)ie)+)
- Lista (((eleccion)ie, nombre)i, cantidadVotos)
- Candidato (((lista)ie, (votante)ie, (cargo)ie)i)
- Cargo ((cargo)i)
- Administrador ((usuario)i, clave)

### 2.3. Estructuras de datos

Las estructuras principales elegidas para este trabajo han sido elegidas según la necesidad de cada entidad. Las mismas son:

- Árbol B+
- Dispersión Extensible
- Archivos de Bloques
- Registros Variables

## 2.4. Funcionalidades

En el ámbito del administrador se encuentran las siguientes funcionalidades:

- Mantener Distritos
- Mantener Votantes
- Mantener Elecciones
- Mantener Cargos
- Mantener Listas
- Mantener Candidatos
- Informar Resultados

## 2.5. Descripción de la arquitectura utilizada

Para una mejor observación de los diagramas de clases de la arquitectura utilizada en el programa, las mismas se encuentran documentadas en un archivo externo junto con su descripción, se puede acceder a el a través del archivo '00 Index.html', el mismo se encuentra en la carpeta 'Documentación'. Desde allí se puede navegar a través de todas las clases existentes.

## 2.6. Descripción de cada clase utilizada

A continuación se enumeran las principales clases de la aplicación, no refiriéndonos a sus nombres de archivo sino con su nombre conceptual.

- Clase ArbolBMas: Se utiliza para guardar las listas de votación y para la confección de informes.
- Clase Dispersión: Se utiliza para almacenar todas las demás entidades que no sean las listas de votación.
- Clase Archivo en Bloques: Se usa para darle un sustento en disco a las clases de Árbol B+ y de Dispersión. Ofrece una persistencia en disco en un archivo de bloques.
- Clases de Entidades: Las mismas se utilizan para instanciar cada entidad necesaria al momento de crear una votación. Las mismas incluyen: Distrito, Votante, Eleccion, Lista, Candidato, Cargo, Administrador.

## 2.7. Justificación de uso de cada clase utilizada

El uso del *Árbol B+* se eligió principalmente debido a la opción de poder hacer búsquedas parciales y además a la ventaja de poder acceder secuencialmente a los datos de forma ordenada. El uso del *archivo de dispersión* se eligió debido a que usa la menor cantidad de accesos posibles (solo un acceso) haciendo que se optimice el uso de estos archivos, al ser los mas accedidos en disco. El uso del *archivo de bloques* se eligió por ser el más compatible con respecto a la persistencia del Árbol B+ y del archivo de dispersión, haciendo mas efectivo el uso de los mismos.

## 2.8. Archivos auxiliares

Definiciones lógicas y físicas de todos los archivos que se utilicen (datos maestros, índices, trabajo, control, etc). Descripción de la organización de cada uno de los archivos. Para qué se utiliza. Por qué se utiliza (base teórica).

Dentro de los archivos utilizados para el funcionamiento del programa disponemos de las siguientes estructuras auxiliares:

- Archivo de configuración: El mismo se usa para obtener las rutas de los archivos de dispersión, archivo del árbol, tamaños de cubetas y de nodos.
- Índice: El índice se usa para la obtención de los informes de votaciones por distintos criterios de una misma base de datos.
- Archivo de control: Se utiliza para poder persistir datos referentes a las estructuras de control de las estructuras, como por ejemplo la tabla de dispersión dentro de la estructuras de dispersión.
- Archivo de password: Guarda el *user* y *pass* perteneciente al administrador de votos.

## 2.9. Planificación de tareas

Planificación (identificación de tareas, estimación de duración y asignación)

A grandes rasgos la planificación del proyecto se realizó dentro de las cinco semanas la siguiente manera:

- Semana 1: Planeamiento del problema, elección de las distintas estructuras para las entidades existentes, configuración del sistema operativo, herramientas de compilación, IDE's, etc. Para su correcto funcionamiento en el sistema Linux. Duración aproximada: 1 semana.
- Semana 2: Construcción de las entidades, creación de sus respectivas clases (1 semana). Creación de la clase encargada del manejo en disco, manejador de archivos (1 semana). Comienzo de creación de la clase de Árbol B+ (4 semanas).
- Semana 3: Comienzo de creación de la clase de Dispersión (3 semanas). Creación de la clase encargada del archivo de bloques (2 semanas). Creación de clase bucket y creación de clase nodo.
- Semana 4: Creación de pruebas individuales e integrales (1 semana). Serialización e hidratación de datos (1 semana). Creación del archivo de buckets (1 semana).
- Semana 5: Integración de los distintos módulos (1 semana). Creación de la lógica de votación (1 semana). Implementación de una interfaz para el usuario (1 semana).

## 2.10. Bugs conocidos

A continuación se enumeran ciertas situaciones donde el programa podría presentar dificultades en su proceso:

- La fechas deben tener tener el formato: aaaa/mm/dd para poder procesarse correctamente.
- Si el archivo de configuración no tiene el delimitador ‘//’ dentro del archivo, el mismo no parsea ningún dato, ya que toma al archivo entero como comentario
- El archivo de bloques debido a su estructura interna debe tener un tamaño de bloque que sea múltiplo de 4 bytes, si no es de esta forma podría referenciar erróneamente a un bloque. Esto se solucionó validando el tamaño de bloque al iniciar un archivo.
- En una elección determinanda se podría agregar cualquier distrito sin que se verifique si realmente existe.
- El borrado de algún archivo de control, configuración y/o datos en tiempo de ejecución llevará a una malfunción del programa.

## 2.11. Archivos de Control para las Entidades

### 2.11.1. Archivos de Control

Se encuentran todos dentro de un directorio, especificado a través de un archivo de configuración de la aplicación, y pueden tener jerarquía de subdirectorios interna. Es donde se guarda toda la información necesaria para poder funcionar.

### 2.11.2. Archivos con resultados

Como respuesta a toda interacción, el sistema generará archivos de registro de operaciones (LOGs) en el directorio donde se llame a la aplicación.

## 3. Parte teórica

En la siguiente sección se enumeran distintos conceptos teóricos acerca de las estructuras usadas en la resolución del TP.

### 3.1. Física - Organización

#### 3.1.1. Organización de registros

- La longitud de un registro y/o de un campo variable se delimita con el uso de un indicador de longitud, el cual se antepone a los datos pertenecientes al registro o campo. De esta forma se puede conocer donde termina la sección de datos.
- En todos los casos se guarda como mínimo la información correspondiente a la longitud del segmento de datos pudiendo, en ciertos casos, guardarse mas información, según sea la estructura.

**3.1.2. Consideraciones acerca del Archivo de Bloques**

1. Los bloques tienen tamaño fijo, determinado por primera y única vez al crearse el archivo.
2. Existen 4 tipos de bloques: Data, Metadata, Removed y Head
3. Head: Existe un solo bloque de este tipo por archivo y siempre se ubica al principio del mismo (en la posición 0).  
El bloque Head tiene en su estructura:  
|currmetadata(int)|maxblocknum(int)|blocksize(int)| Espacio sin uso (int)|...  
...|Espacio sin uso (int)|
4. Data: Este bloque se usa íntegramente para guardar datos del usuario  
El bloque Data tiene en su estructura:  
|datos(int)|...|datos(int)|...|datos(int)| (Todo el espacio reservado para datos, sin metadata).
5. Metadata: Bloque que se encarga del control de los bloques de datos borrados (Removed). Aquí se guardan las referencias a bloques removed.  
El bloque de Metadata tiene en su estructura:  
| Metadata Anterior(int)| currPos(int)| ID bloqLibre (int)| ...| ID bloqLibre (int)|
6. Removed: Son bloques de datos que han sido borrados por el usuario, no se utiliza ningún atributo para identificarlos, los mismos están referenciados en el bloque de metadata como 'libre'.
7. Internamente todos los atributos de metadata son manejados como int. En el caso donde desde afuera se piden los bloques de datos el bloque se obtiene como un char\*.
8. El currpos empieza desde el primer byte del bloque, incluyendo los bytes cabeceras. O sea que el 1er dato de metadata está en el byte 8 (2\*size-of(int))
9. Cada vez que cambia el 'currmetadata' o 'maxblocknum' se escribe en disco con serializehead (que escribe 2 veces en disco por cada vez que se lo llama)
10. Cada vez que cambia el 'currpos' se escribe en disco (se escribe en el bloque de metadata). Esto sucede al pedir un bloque nuevo o al borrar un bloque de datos
11. Está contemplado el caso donde el metadata actual no tiene bloques libres, y entonces este mismo pasa a ser un bloque disponible (Caso límite).
12. Está contemplado el caso donde borro un data, pero el metadata actual está totalmente lleno, pasando el bloque 'D' a ser un bloque 'M' (Caso límite).
13. Si pido un metadata nuevo tengo que ver si es el 1ro, si esto es así, su valor ".anterior"(posición [1] dentro del bloque de int's) tiene que ser = Cero

14. No hace falta un getblock de 'R' porque esos bloques siempre van a ser accedidos a través de newblock (con parámetro 'D' o 'M')
15. No hace falta un newblock de 'R' porque esos bloques siempre van a ser creados a través de delblock
16. En una primera instancia, se propuso etiquetar cada bloque para su identificación. Pero luego, para simplificar el funcionamiento y para no invadir espacio en el bloque de Datos, se optó por no usar etiquetas en los bloques.

## 3.2. Índices - Búsqueda

### 3.2.1. Hashing

Función de hashing utilizada. Criterio de elección.

Para una rápida recuperación de la información se decide organizar dicho archivo con el método de dispersión extensible. Se utiliza un hashing extensible de valores sufijos. La función de hashing utilizada es  $f(x) = (x) \bmod (\text{tamaño de tabla})$

### 3.2.2. Árbol B+

Nuestro árbol B+ es relativamente genérico, a excepción de la clave de ordenamiento que es invariablemente un string, los datos son una entidad genérica y en general no me interesa que se guarda ahí.

Está implementado de tal manera que también está separado casi completamente de la parte que lidia con el almacenamiento teniendo 2 precondiciones ineludibles. el lugar donde se encuentra alojada la raíz debe ser invariante y las estructuras que representan a los punteros a los bloques deben ser representadas como enteros.

La separación se logró mediante la clase ffile y tuvo tanto éxito que con mínimas modificaciones fue posible pasar de un árbol cuyo almacenamiento era la memoria a otro cuyo almacenamiento era un archivo de bloques.

El árbol está implementado en 4 clases diferentes, la ya mencionada ffile, bplustree, inner\_node y leaf\_node estando estas últimas intimamente relacionadas de manera tal que una está declarada como friend de la otra. Esta decisión se tomó para no exponer cosas críticas de las clases que necesariamente tenían que usarse entre inner\_node y leaf\_node.

En memoria, estas clases utilizan un vector de pares de enteros y cadenas de caracteres para los nodos internos (la clase inner\_node) y pares de cadenas y vectores en las hojas (la clase leaf\_node). Estas 2 clases, al serializarse se transforman en una cadena larga de caracteres (que se guarda en un vector) representada por pares longitud, datos. Además a esto se agrega un entero más que representa la cantidad de pares, longitud, datos que existen en la cadena en adición a un encabezado que es una I o una L según sea el nodo un nodo interno o una hoja y un par de enteros más que representan el nodo que contiene los datos cuya clave es mayor al nodo actual o el nodo que tiene datos menor al primer par del nodo.

En este momento, el árbol se comporta como un árbol B+ clásico para los agregados pero no hace rebalanceos en caso de los borrados.



Mas alla de los enteros que tienen tamaño fijo, el resto de las estructuras son dinamicas. El objetivo de esta implementacion fue utilizar la mayor cantidad de las funciones de la STL que fueran posible y, como desafortunadamente, la STL, si bien tiene una funcion de busqueda que funciona en contenedores como los vectores utilizados para almacenar los datos en la memoria de este árbol, ésta solo devuelve si un dato se encuentra o no en el contenedor pero no su posición o sus datos por lo que la búsqueda dentro de los nodos es lineal.

Esto también fue una pre condicion para poder implementar la función que luego de una búsqueda devuelva el dato inmediatamente mayor al ultimo devuelto.

Para ordenar los datos en memoria, sin embargo si se utiliza la función `sort()` del grupo de funciones incluidas en `<algorithms>`

## 4. Documentación de usuario

### 4.1. Instalación del sistema

La instalación usará un archivo `makefile`, el cual realizará automaticamente toda la operación de compilación de todos los archivos de código fuente, recorriendo todas las subcarpetas necesarias para una operación exitosa.

Para la instalación del sistema debemos descomprimir el archivo conteniendo el código fuente en la carpeta donde se desea realizar la misma. Una vez realizada la descompresión debemos ejecutar por consola, situados en la ruta elegida, el comando `make` el cual compilará todos los archivos fuente y como consecuencia obtendremos el archivo ejecutable listo para usar.

### 4.2. Ejecución del sistema

Una vez creado el archivo ejecutable debemos iniciar el programa<sup>1</sup> pasando por parámetro la ruta del archivo de configuración, esto es un requisito obligatorio para comenzar con el programa, mediante los argumentos `-c <rutaArchivoConfiguración>`. De esta forma, desde el directorio donde se creó el ejecutable, se inicia el sistema:

```
./voto -c ../ArchivosAuxiliares/config.txt
```

**Nota:** `-c` es el flag para ubicar el archivo de configuración. Para ver las demás opciones podemos ingreasar al programa con `-h`

A su vez el archivo de configuración proveerá todos los requisitos necesarios para ubicar los demás archivos relativos al programa, ya sean archivos de datos, control, configuración o el archivo de password.

#### 4.2.1. Interfaz de administrador

Al iniciar el programa el administrador de la votación deberá autenticarse para poder entrar al sistema, una vez ingresado tendrá un menú con opciones donde podrá:

---

<sup>1</sup>Para iniciar el programa, al administrador se debe autenticar. User: undomiel, Pass: aragorn

- Mantener Distritos.
- Mantener Votantes.
- Mantener Elecciones.
- Mantener Cargos.
- Mantener Listas.
- Mantener Candidatos.
- Informar Resultados.

#### 4.2.2. Interfaz de usuario

Una vez iniciada una votación, se dispondrá de una interfaz para el usuario 'votante'. En esta instancia el votante podrá:

- Autenticarse
- Emitir voto
- Corregir voto
- Informar voto

## 5. Segunda etapa

Método de Vigènere

Se empleará el método para cifrar los reportes generados: Listas, Elección, Distrito.

Para la resolución se necesitará generar una clave  $k$  de dimensión  $n$ . Cada elemento de la c

Una opción resulta en considerar al alfabeto como todo el conjunto de los elementos repres

Objeto: Vigènere

Desde el punto de vista de la programación se crearía la clase Vigènere que sería la encar

El criptosistema es simétrico por lo que al instanciar la clase se le asignaría la clave q

```
/*  
  
 * Instancia un objeto cifrador, el tamaño del alfabeto está implícito  
  
 * y se asume resulta ser de 256.  
  
*/  
  
Vigenere(string clave);    // constructor
```

La clase tendría como miembro dicha clave.

También existe un constructor alternativo para poder emplear alfabetos de menor tamaño.

```
/*  
    * Intancia un objeto cifrador, el tamaño del alfabeto se define mediante  
    * cardinalAlfabeto.  
*/  
  
Vigenere(string clave,int cardinalAlfabeto);
```

La clave se pasa por valor de forma tal que el objeto instanciado tenga un copia interna d

La encriptación manejaría strings, recibiría en mensaje y lo cifraría devolviendo el corre

```
/*  
    * Cifra un mensaje obteniendo y devuelve un criptograma.  
    *  
    * Si no se está haciendo un cifrado en bloques entonces hay que invocar  
    * al método reiniciarCifrador previamente, excepto en el primer uso.  
    *  
    * El método reserva un string en memoria por lo que debe ser liberado luego  
    * de ser empleado.  
*/  
  
string* cifrar(string* mensaje);
```

De esta forma haríamos que se desentienda del origen del objeto que está manejando, que po

Si necesitáramos descifrar el mensaje entonces se usaría el siguiente método, que resulta

```
/*  
    * Descifra un criptograma, devolviendo el mensaje original.  
    *  
    * Si no se está haciendo un descifrado en bloques entonces hay que invocar
```

```
* al método reiniciarDescifrador previamente, excepto en el primer uso.
```

```
*
```

```
* El método reserva un string en memoria por lo que debe ser liberado luego
```

```
* de ser empleado.
```

```
*/
```

```
string* descifrar(string* criptograma);
```

El descifrador recibe el criptograma, lo descifra con la clave y devuelve el mensaje como

Hay dos situaciones diferentes en las cuales se debe usar Vigenere. Una es para cifrar un

En el caso que se quiere cifrar un texto plano solamente con la misma clave y la misma ins

En la otra situación debería invocarse previamente a cada invocación a los métodos de cifr

```
/*
```

```
* Para el manejo en bloques el método tiene un contador interno de la
```

```
* última posición leída, si se quiere encriptar un nuevo mensaje debe
```

```
* reiniciarse dicho contador.
```

```
*/
```

```
void reiniciarCifrador();
```

```
/*
```

```
* Para el manejo en bloques el método tiene un contador interno de la
```

```
* última posición leída, si se quiere descryptar un nuevo mensaje debe
```

```
* reiniciarse dicho contador.
```

```
*/
```

```
void reiniciarDescifrador();
```

Por lo tanto el objeto final estaría constituido por los siguientes miembros privados:

```
// Almacena internamente la clave empleada para la encriptación y descryptación.
```

```
string      clave;
```

```
// Contiene el tamaño del alfabeto empleado.  
  
int      cardinalAlfabeto;  
  
// Mantiene la última posición empleada de la clave por el cifrador.  
  
int      posicionCifrador;  
  
// Mantiene la última posición traducida de la clave por el descifrador.  
  
int      posicionDescifrador;
```

### Criptanálisis

Para realizar el criptoanálisis una opción es resolverlo empleando el método de Kasiski.

#### 1. Desarrollo del método:

El método consiste en realizar una búsqueda de fragmentos de bits repetidos para luego cal

Una vez obtenida se procede a realizar un análisis de frecuencia de caracteres en una form

Luego conociendo previamente el idioma en que se encuentra el criptograma, se procede a bu

Como tenemos 256 posibilidades, hacemos un prueba sobre un archivo para obtener un histogr

En base a pruebas realizadas por el grupo se obtuvo el siguiente gráfico:

El gráfico que se observa muestra las frecuencias obtenidas en una prueba para los 255 car

A continuación se agrega el mismo gráfico pero acotado a los valores imprimibles.

Como se puede observar el carácter con mayor frecuencia es el 32, que resulta ser el espaci

En el texto de prueba se obtuvieron en porcentaje:

espacio: 16%

a : 9%

e : 10%

o : 6,7%

Es decir que como es esperable las minúsculas son más probables que las mayúsculas, y el c

Es destacable que con sólo los caracteres espacio, a, e y o se emplean en el 36% de un tex

Finalmente se descifra el mensaje.

Objeto: Kasiski

Es objeto contendrá un índice con las posiciones de las cadenas encontradas. Esto se resol

También sería necesario resolver el cálculo del máximo común divisor. Sólo consiste en apl

Finalmente armar un vector de analizadores de frecuencias (un objeto) que contendrá las fr

Finalmente como resultado final el método devuelve la clave.

```
private:
```

```
string      claveEncontrada;
```

```
int         mcd;
```

```
list<Patron> listaPatrones;
```

```
int gcd(int a, int b);
```

```
bool noExiste(Patron patron);
```

```
void agregarPatron(Patron patron, int posicion);
```

```
public:
```

```
void generarPatrones(string);
```

```
void listarPatrones();
```

```
int MCDDistancias();
```

```
void analisisDeFrecuencias(string);
```

```
string romper();
```

Objeto: Patrón

Contiene un conjunto de caracteres que se repite a lo largo del texto a descrifrar. Tambié

```
    // almacena el patrón hallado.
```

```
string palabra;
```

```
// lista de distancias entre patrones recurrentes.
```

```
list<int> listaDistancias;

También contiene un conjunto de métodos:

// constructor del objeto, recibe el patrón en un string.
Patron(string);

// agrega una distancia a la lista de distancias.
void agregarDistancia(int distancia);

// devuelve el string correspondiente al patrón que caracteriza el objeto.
string getPalabra();

// devuelve el iterador al primer elemento de la lista de distancias.
list<int>::iterator devolverDistancias();

// devuelve true mientras el iterador pasado por parámetro no apunte al último elemento
// de la lista.
bool noEsUltimo(list<int>::iterator iterador);
```

Objeto: Analizador de frecuencias

Contiene un conjunto de frecuencias de todos los elementos a analizar. Así como también lo

private:

```
vector<int> listaDeFrecuencias;
```

```
bool supremo;
```

public:

```
AnalizadorDeFrecuencias(int cantidadElementos);
```

```
bool incrementarFrecuencia(int posicion);
```

```
bool existeSupremo();
```

```
int posicionMaximaFrecuencia();
```

```
void mostrarListaDeFrecuencias();  
  
virtual ~AnalizadorDeFrecuencias();
```

Sobre RSA:

Se empleará el método para cifrar los datos de los votantes.

El sistema empieza generando por primera y única vez el par de claves privada y pública que

Utilización de campos finitos en el algoritmo de RSA:

Se utilizará el algoritmo de euclides extendido para generar el inverso multiplicativo o  $s$

También se usará exponenciación por cuadrados para la operación  $a^b \bmod q$ .

Por esta razón se debe tener especial cuidado en la cantidad de bits que se utilizará para

Limitaremos el tamaño de los números primos a entre 3 y 97. Siendo los demás números resul

Objeto RSA

Desde el punto de vista de la programación se crearía la clase RSA que será la encargada d

Encriptación:

El votante instancia un objeto RSA con sus atributos cargados apartir de un archivo.

La encriptación se realiza sobre el string serializado del votante.

Desde serializar se llama al método de encriptación de RSA pasándole el string serializado

Se toma cada carácter del string y se lo convierte en número entero.

Se realizan las operaciones de campos finitos y se devuelve un numero grande.

Se almacena el número grande en un buffer auxiliar.

Luego de almacenar todos los caracteres transformados en numeros en el buffer auxiliar, se

Todos los votantes se cifran con la misma clave, ésta se genera una única vez y se almacen

Desencriptación:

El deserializar de votante recibe el string serializado y encriptado.

El deserializar de votante instancia un objeto RSA.



Se inicializan los atributos del RSA apartir de la clave privada y pública guardadas en un

Se descripta el string que recibió el deserializar de votante a partir del método desenc

Desencriptar recibe el string serializado y encriptado y procede a desencriptar cada carác

Se realizan las operaciones de campos finitos y se devuelve un numero entero.

Cada número entero (carácter desencriptado) se carga en un string. Al final se devuelve es

Criptoolisis:

Se genera una lista de números primos entre 3 y 97 dado que hemos decidido acotar los núme

Se realizan multiplicaciones entre todos los numeros hasta lograr encontrar el par que de

El ataque por fuerza bruta es el más sencillo y eficaz dado el pequeño rango de números po

## 6. Corridas de prueba

A continuación se detallan las pruebas realizadas sobre el funcionamiento del programa. Se emplearon las pruebas detalladas en el enunciado del informe, las cuales pasaron con éxito.

Corridas de prueba:

Ingreso:

INGRESO APROBADO

Bienvenido al sistema de gestión de elecciones

¿ Desea eliminar la base de datos y comenzar de 0? S/N

S

Menú principal:

Opciones:

- 1) Mantener Distritos
  - 2) Mantener Votantes
  - 3) Mantener Elecciones
  - 4) Mantener Cargos
  - 5) Mantener Listas
  - 6) Mantener Candidatos
  - 7) Informar Resultados
  - 8) Habilitar Elecciones
  - 9) Habilitar Votantes para elección
  - 10) salir
- Opcion: 1

Menú de Distrito:

Opciones:

- 1) Alta Distrito
  - 2) Baja Distrito
  - 3) Modificar Distrito
  - 4) Volver atrás
  - 5) Ver Distritos
- Opcion: 1

Ingrese el nombre del distrito:

Misiones

Operacion OK

Menú de Votante:

Opciones:

- 1) Alta Votante
- 2) Baja Votante
- 3) Modificar Votante

- 4) Volver atrás
- 5) Alta Automática
- 6) Ver votantes

Opcion: 1

Ingrese el DNI del votante: 14254983

Ingrese nombre: Daniel

Ingrese apellido: Martinez

Ingrese la clave: 8754

Clave: 8754

Ingrese el domicilio: San Luis 2728

dom: San Luis 2728

Ingrese el nombre del distrito:

Misiones

Distrito: Misiones

Operacion OK

Opciones:

- 1) Alta Votante
- 2) Baja Votante
- 3) Modificar Votante
- 4) Volver atrás
- 5) Alta Automática
- 6) Ver votantes

Opcion: 5

Ingrese la cantidad de votantes a ingresar: 3

- Nombre: Mariel Iacub
- DNI: 1
- Password: 8335
- Domicilio: Haiti 2793
- Distrito: Jujuy

Elecciones Anteriores:

-----

El votante no participo de ninguna eleccion a la fecha

- Nombre: Ivan Lopez
- DNI: 2
- Password: 6498
- Domicilio: Montiel 3061
- Distrito: Santa Fe

Elecciones Anteriores:

-----

El votante no participo de ninguna eleccion a la fecha

- Nombre: Fernanda Rodriguez
- DNI: 3
- Password: 4940
- Domicilio: Udaondo 1425
- Distrito: Ciudad Autonoma de Buenos Aires

Elecciones Anteriores:

-----

El votante no participo de ninguna eleccion a la fecha

Menu de Cargo:

Opciones:

- 1) Alta Cargo
- 2) Baja Cargo
- 3) Modificar Cargo
- 4) Volver atrás
- 5) Ver cargos

Opcion: 1

Ingrese el nombre del cargo principal:

Presidente

Desea agregar subcargos? (S/N)S

Ingrese el nombre del subcargo:

Vice Presidente

Desea agregar más subcargos? (ingrese 'S' para seguir)N

Operacion OK

Menú de Elección:

Opciones:

- 1) Alta Eleccion
- 2) Baja Eleccion
- 3) Modificar Eleccion
- 4) Volver atrás
- 5) Ver elecciones

Opcion: 1

Ingrese la fecha de la elección: 19991010

Ingrese el cargo: Panadero

No existe cargo/cargo no valido

¿ Desea repetir? (S/N): S

Ingrese el cargo: Presidente

Ingrese el nombre del distrito:

Panaderia

El distrito no existe

Desea agregar más distritos? (ingrese 'S' para seguir)S

Ingrese el nombre del distrito:

Misiones

Distrito agregado

Desea agregar más distritos? (ingrese 'S' para seguir)N

Operacion OK

Opciones:

- 1) Alta Eleccion
- 2) Baja Eleccion
- 3) Modificar Eleccion
- 4) Volver atrás
- 5) Ver elecciones

Opcion: 5

tamaño de la tabla de dispersion: 1

2048 B0 :(free=958) : (cant: 1):

Fecha: 19991010

Cargo Principal: Presidente

Distrito: Misiones

Tabla de hash (size: 1): 0

Tabla de dispersion (size: 1): 1

Menú de Lista:

Opciones:

- 1) Alta Lista
- 2) Baja Lista
- 3) Modificar Lista
- 4) Volver atrás
- 5) Ver Listas

Opcion: 1

Ingrese la fecha de la elección: 19991110

Ingrese el cargo: Zapatero

No existe cargo/cargo no valido

¿ Desea repetir? (S/N): S

Ingrese el cargo: Presidente

Ingrese la lista: Datos

Operación exitosa

Opciones:

1) Alta Lista

2) Baja Lista

3) Modificar Lista

4) Volver atrás

5) Ver Listas

Opcion: 5

19991010    Presidente blanco

19991110    Presidente Datos

Menú Candidato:

Opciones:

1) Alta Candidato

2) Baja Candidato

3) Modificar Candidato

4) Volver atrás

5) Mostrar Candidatos

Opcion: 1

Ingrese el numero de DNI: 1

Ingrese la fecha de la elección: 19991010

Ingrese el cargo: Gobernador

Ingrese el nombre de la lista: FIUBA

Operacion OK

Menú Habilitar Elecciones

Opciones:

1) Habilitar Eleccion

2) Ver elecciones habilitadas

3) Salir

Opcion: 1

Ingrese la fecha de la elección: 19991010

Ingrese el cargo: Presidente

Elección habilitada

Opciones:

1) Habilitar Eleccion  
2) Ver elecciones habilitadas  
3) Salir  
Opcion: 2

ELECCIONES ACTIVAS

-----  
Eleccion 1  
Fecha: 19991010  
Cargo Principal: Presidente  
-----

Menú habilitar votante:  
Caso Automático:  
Ingrese la cantidad de votos a realizar: 4800

Ingrese modo de votación: Automático (a) o Manual (m)a.  
Bienvenido Jessica Michel  
Ingrese su Password  
INGRESO AUTORIZADO

Las elecciones activas en las que usted emitir su voto son las siguientes  
Eleccion 1:  
Fecha: 19991010  
Cargo Principal: Presidente  
-----

Indique el numero de eleccion en la cual desea sufragar  
Usted eligio la eleccion 1  
Si es correcto presione s sino n

Estas son sus boletas a elegir

Lista 1  
Nombre: ARI  
Lista 2  
Nombre: Izquierda  
Lista 3  
Nombre: PJ  
Lista 4  
Nombre: PRO  
Lista 5  
Nombre: Socialista  
Lista 6  
Nombre: UCR  
Lista 7  
Nombre: blanco

Elija su boleta en base al numero de opcion indicado

La opcion elegida es: 1

LISTA: ARI

Si esta seguro presione s si desea corregir su voto presione n

Caso Manual:

Ingrese la cantidad de votos a realizar: 1

Ingrese modo de votación: Automático (a) o Manual (m)m

Bienvenido al sistema de voto electronico de los Gutierrez

Ingrese su DNI:

5002

Su dni es: 5002

Presione S para confirmar, N para cancelar

s

Bienvenido Daniel Martinez

Ingrese su Password

5002

INGRESO AUTORIZADO

Las elecciones activas en las que usted emitir su voto son las siguientes

Eleccion 1:

Fecha: 19991010

Cargo Principal: Presidente

-----

Indique el numero de eleccion en la cual desea sufragar

1

Usted eligio la eleccion 1

Si es correcto presione s sino n

s

Estas son sus boletas a elegir

Lista 1

Nombre: Datos

Lista 2

Nombre: blanco

Elija su boleta en base al numero de opcion indicado

1

La opcion elegida es: 1

LISTA: Datos

Si esta seguro presione s si desea corregir su voto presione n

s

Menú de informes:

Opciones:

1) Informe por elección



2) Informe por lista  
 3) Informe por distrito  
 4) Volver atrás  
 5) Mostrar archivo de conteo  
 6) Mostrar archivo de conteo ordenado por distrito  
 Opcion: 1

Ingrese la fecha de la elección: 19991010

Ingrese el cargo: Presidente

\*\*\*\*\* GENERO EL INFORME POR ELECCION \*\*\*\*\*

Fecha	Cargo	Lista	Cantidad d
19991010	Presidente	ARI	2533
19991010	Presidente	Izquierda	1283
19991010	Presidente	PJ	1267
19991010	Presidente	PRO	1332
19991010	Presidente	Socialista	1305
19991010	Presidente	UCR	1280
19991010	Presidente	blanco	0

Ingrese una tecla para continuar

Opciones:

1) Informe por elección  
 2) Informe por lista  
 3) Informe por distrito  
 4) Volver atrás  
 5) Mostrar archivo de conteo  
 6) Mostrar archivo de conteo ordenado por distrito  
 Opcion: 2

Ingrese la fecha de la elección: 19991010

Ingrese el cargo: Presidente

Ingrese la lista: Socialista

\*\*\*\*\* GENERO EL INFORME POR LISTA \*\*\*\*\*

Lista a informar: Socialista

Fecha	Nombre de lista	Cantidad de votos
19991010	Socialista	1305
	Cargo principal Presidente	
	Subcargo 1 Vicepresidente	

Ingrese una tecla para continuar

Opciones:

- 1) Informe por elección
  - 2) Informe por lista
  - 3) Informe por distrito
  - 4) Volver atrás
  - 5) Mostrar archivo de conteo
  - 6) Mostrar archivo de conteo ordenado por distrito
- Opcion: 3

Ingrese el nombre del distrito: Misiones

\*\*\*\*\* GENERO EL INFORME POR DISTRITO \*\*\*\*\*

Distrito a informar: Misiones

Fecha	Cargo	Lista ganadora	Cantidad d
19991010	Presidente	ARI	108

Ingrese una tecla para continuar

Opciones:

- 1) Informe por elección
  - 2) Informe por lista
  - 3) Informe por distrito
  - 4) Volver atrás
  - 5) Mostrar archivo de conteo
  - 6) Mostrar archivo de conteo ordenado por distrito
- Opcion: 5

19991010	Presidente	ARI	Buenos Aires
19991010	Presidente	ARI	Catamarca
19991010	Presidente	ARI	Chaco
19991010	Presidente	ARI	Chubut
19991010	Presidente	ARI	Ciudad Autonoma de Buenos Aires
19991010	Presidente	ARI	Cordoba
19991010	Presidente	ARI	Corrientes
19991010	Presidente	ARI	Entre Rios
19991010	Presidente	ARI	Formosa
19991010	Presidente	ARI	Jujuy
19991010	Presidente	ARI	La Pampa
19991010	Presidente	ARI	La Rioja
19991010	Presidente	ARI	Mendoza
19991010	Presidente	ARI	Misiones
19991010	Presidente	ARI	Neuquen
19991010	Presidente	ARI	Rio Negro
19991010	Presidente	ARI	Salta
19991010	Presidente	ARI	San Juan
19991010	Presidente	ARI	San Luis
19991010	Presidente	ARI	Santa Cruz
19991010	Presidente	ARI	Santa Fe
19991010	Presidente	ARI	Santiago del Estero

19991010	Presidente	ARI	Tierra del Fuego
19991010	Presidente	ARI	Tucuman
19991010	Presidente	Izquierda	Buenos Aires
19991010	Presidente	Izquierda	Catamarca
19991010	Presidente	Izquierda	Chaco
19991010	Presidente	Izquierda	Chubut
19991010	Presidente	Izquierda	Ciudad Autonoma de Buenos Aires
19991010	Presidente	Izquierda	Cordoba
19991010	Presidente	Izquierda	Corrientes
19991010	Presidente	Izquierda	Entre Rios
19991010	Presidente	Izquierda	Formosa
19991010	Presidente	Izquierda	Jujuy
19991010	Presidente	Izquierda	La Pampa
19991010	Presidente	Izquierda	La Rioja
19991010	Presidente	Izquierda	Mendoza
19991010	Presidente	Izquierda	Misiones
19991010	Presidente	Izquierda	Neuquen
19991010	Presidente	Izquierda	Rio Negro
19991010	Presidente	Izquierda	Salta
19991010	Presidente	Izquierda	San Juan
19991010	Presidente	Izquierda	San Luis
19991010	Presidente	Izquierda	Santa Cruz
19991010	Presidente	Izquierda	Santa Fe
19991010	Presidente	Izquierda	Santiago del Estero
19991010	Presidente	Izquierda	Tierra del Fuego
19991010	Presidente	Izquierda	Tucuman
19991010	Presidente	PJ	Buenos Aires
19991010	Presidente	PJ	Catamarca
19991010	Presidente	PJ	Chaco
19991010	Presidente	PJ	Chubut
19991010	Presidente	PJ	Ciudad Autonoma de Buenos Aires
19991010	Presidente	PJ	Cordoba
19991010	Presidente	PJ	Corrientes
19991010	Presidente	PJ	Entre Rios
19991010	Presidente	PJ	Formosa
19991010	Presidente	PJ	Jujuy
19991010	Presidente	PJ	La Pampa
19991010	Presidente	PJ	La Rioja
19991010	Presidente	PJ	Mendoza
19991010	Presidente	PJ	Misiones
19991010	Presidente	PJ	Neuquen
19991010	Presidente	PJ	Rio Negro
19991010	Presidente	PJ	Salta
19991010	Presidente	PJ	San Juan
19991010	Presidente	PJ	San Luis
19991010	Presidente	PJ	Santa Cruz
19991010	Presidente	PJ	Santa Fe
19991010	Presidente	PJ	Santiago del Estero
19991010	Presidente	PJ	Tierra del Fuego
19991010	Presidente	PJ	Tucuman

19991010	Presidente	PRO	Buenos Aires
19991010	Presidente	PRO	Catamarca
19991010	Presidente	PRO	Chaco
19991010	Presidente	PRO	Chubut
19991010	Presidente	PRO	Ciudad Autonoma de Buenos Aires
19991010	Presidente	PRO	Cordoba
19991010	Presidente	PRO	Corrientes
19991010	Presidente	PRO	Entre Rios
19991010	Presidente	PRO	Formosa
19991010	Presidente	PRO	Jujuy
19991010	Presidente	PRO	La Pampa
19991010	Presidente	PRO	La Rioja
19991010	Presidente	PRO	Mendoza
19991010	Presidente	PRO	Misiones
19991010	Presidente	PRO	Neuquen
19991010	Presidente	PRO	Rio Negro
19991010	Presidente	PRO	Salta
19991010	Presidente	PRO	San Juan
19991010	Presidente	PRO	San Luis
19991010	Presidente	PRO	Santa Cruz
19991010	Presidente	PRO	Santa Fe
19991010	Presidente	PRO	Santiago del Estero
19991010	Presidente	PRO	Tierra del Fuego
19991010	Presidente	PRO	Tucuman
19991010	Presidente	Socialista	Buenos Aires
19991010	Presidente	Socialista	Catamarca
19991010	Presidente	Socialista	Chaco
19991010	Presidente	Socialista	Chubut
19991010	Presidente	Socialista	Ciudad Autonoma de Buenos Aires
19991010	Presidente	Socialista	Cordoba
19991010	Presidente	Socialista	Corrientes
19991010	Presidente	Socialista	Entre Rios
19991010	Presidente	Socialista	Formosa
19991010	Presidente	Socialista	Jujuy
19991010	Presidente	Socialista	La Pampa
19991010	Presidente	Socialista	La Rioja
19991010	Presidente	Socialista	Mendoza
19991010	Presidente	Socialista	Misiones
19991010	Presidente	Socialista	Neuquen
19991010	Presidente	Socialista	Rio Negro
19991010	Presidente	Socialista	Salta
19991010	Presidente	Socialista	San Juan
19991010	Presidente	Socialista	San Luis
19991010	Presidente	Socialista	Santa Cruz
19991010	Presidente	Socialista	Santa Fe
19991010	Presidente	Socialista	Santiago del Estero
19991010	Presidente	Socialista	Tierra del Fuego
19991010	Presidente	Socialista	Tucuman
19991010	Presidente	UCR	Buenos Aires
19991010	Presidente	UCR	Catamarca

## 6 CORRIDAS DE PRUEBA

19991010	Presidente	UCR	Chaco
19991010	Presidente	UCR	Chubut
19991010	Presidente	UCR	Ciudad Autonoma de Buenos Aires
19991010	Presidente	UCR	Cordoba
19991010	Presidente	UCR	Corrientes
19991010	Presidente	UCR	Entre Rios
19991010	Presidente	UCR	Formosa
19991010	Presidente	UCR	Jujuy
19991010	Presidente	UCR	La Pampa
19991010	Presidente	UCR	La Rioja
19991010	Presidente	UCR	Mendoza
19991010	Presidente	UCR	Misiones
19991010	Presidente	UCR	Neuquen
19991010	Presidente	UCR	Rio Negro
19991010	Presidente	UCR	Salta
19991010	Presidente	UCR	San Juan
19991010	Presidente	UCR	San Luis
19991010	Presidente	UCR	Santa Cruz
19991010	Presidente	UCR	Santa Fe
19991010	Presidente	UCR	Santiago del Estero
19991010	Presidente	UCR	Tierra del Fuego
19991010	Presidente	UCR	Tucuman
19991010	Presidente	blanco	Buenos Aires
19991010	Presidente	blanco	Catamarca
19991010	Presidente	blanco	Chaco
19991010	Presidente	blanco	Chubut
19991010	Presidente	blanco	Ciudad Autonoma de Buenos Aires
19991010	Presidente	blanco	Cordoba
19991010	Presidente	blanco	Corrientes
19991010	Presidente	blanco	Entre Rios
19991010	Presidente	blanco	Formosa
19991010	Presidente	blanco	Jujuy
19991010	Presidente	blanco	La Pampa
19991010	Presidente	blanco	La Rioja
19991010	Presidente	blanco	Mendoza
19991010	Presidente	blanco	Misiones
19991010	Presidente	blanco	Neuquen
19991010	Presidente	blanco	Rio Negro
19991010	Presidente	blanco	Salta
19991010	Presidente	blanco	San Juan
19991010	Presidente	blanco	San Luis
19991010	Presidente	blanco	Santa Cruz
19991010	Presidente	blanco	Santa Fe
19991010	Presidente	blanco	Santiago del Estero
19991010	Presidente	blanco	Tierra del Fuego
19991010	Presidente	blanco	Tucuman

Cantidad de votos en total: 9000

Ingrese una tecla para continuar

Opciones:

- 1) Informe por elección
- 2) Informe por lista
- 3) Informe por distrito
- 4) Volver atrás
- 5) Mostrar archivo de conteo
- 6) Mostrar archivo de conteo ordenado por distrito

Opcion: 6

Buenos Aires	19991010	Presidente	ARI
Buenos Aires	19991010	Presidente	Izquierda
Buenos Aires	19991010	Presidente	PJ
Buenos Aires	19991010	Presidente	PRO
Buenos Aires	19991010	Presidente	Socialista
Buenos Aires	19991010	Presidente	UCR
Buenos Aires	19991010	Presidente	blanco
Catamarca	19991010	Presidente	ARI
Catamarca	19991010	Presidente	Izquierda
Catamarca	19991010	Presidente	PJ
Catamarca	19991010	Presidente	PRO
Catamarca	19991010	Presidente	Socialista
Catamarca	19991010	Presidente	UCR
Catamarca	19991010	Presidente	blanco
Chaco	19991010	Presidente	ARI
Chaco	19991010	Presidente	Izquierda
Chaco	19991010	Presidente	PJ
Chaco	19991010	Presidente	PRO
Chaco	19991010	Presidente	Socialista
Chaco	19991010	Presidente	UCR
Chaco	19991010	Presidente	blanco
Chubut	19991010	Presidente	ARI
Chubut	19991010	Presidente	Izquierda
Chubut	19991010	Presidente	PJ
Chubut	19991010	Presidente	PRO
Chubut	19991010	Presidente	Socialista
Chubut	19991010	Presidente	UCR
Chubut	19991010	Presidente	blanco
Ciudad Autonoma de Buenos Aires	19991010	Presidente	ARI
Ciudad Autonoma de Buenos Aires	19991010	Presidente	Izquierda
Ciudad Autonoma de Buenos Aires	19991010	Presidente	PJ
Ciudad Autonoma de Buenos Aires	19991010	Presidente	PRO
Ciudad Autonoma de Buenos Aires	19991010	Presidente	Socialista
Ciudad Autonoma de Buenos Aires	19991010	Presidente	UCR
Ciudad Autonoma de Buenos Aires	19991010	Presidente	blanco
Cordoba	19991010	Presidente	ARI
Cordoba	19991010	Presidente	Izquierda
Cordoba	19991010	Presidente	PJ
Cordoba	19991010	Presidente	PRO
Cordoba	19991010	Presidente	Socialista

## 6 CORRIDAS DE PRUEBA

Cordoba	19991010	Presidente	UCR
Cordoba	19991010	Presidente	blanco
Corrientes	19991010	Presidente	ARI
Corrientes	19991010	Presidente	Izquierda
Corrientes	19991010	Presidente	PJ
Corrientes	19991010	Presidente	PRO
Corrientes	19991010	Presidente	Socialista
Corrientes	19991010	Presidente	UCR
Corrientes	19991010	Presidente	blanco
Entre Rios	19991010	Presidente	ARI
Entre Rios	19991010	Presidente	Izquierda
Entre Rios	19991010	Presidente	PJ
Entre Rios	19991010	Presidente	PRO
Entre Rios	19991010	Presidente	Socialista
Entre Rios	19991010	Presidente	UCR
Entre Rios	19991010	Presidente	blanco
Formosa	19991010	Presidente	ARI
Formosa	19991010	Presidente	Izquierda
Formosa	19991010	Presidente	PJ
Formosa	19991010	Presidente	PRO
Formosa	19991010	Presidente	Socialista
Formosa	19991010	Presidente	UCR
Formosa	19991010	Presidente	blanco
Jujuy	19991010	Presidente	ARI
Jujuy	19991010	Presidente	Izquierda
Jujuy	19991010	Presidente	PJ
Jujuy	19991010	Presidente	PRO
Jujuy	19991010	Presidente	Socialista
Jujuy	19991010	Presidente	UCR
Jujuy	19991010	Presidente	blanco
La Pampa	19991010	Presidente	ARI
La Pampa	19991010	Presidente	Izquierda
La Pampa	19991010	Presidente	PJ
La Pampa	19991010	Presidente	PRO
La Pampa	19991010	Presidente	Socialista
La Pampa	19991010	Presidente	UCR
La Pampa	19991010	Presidente	blanco
La Rioja	19991010	Presidente	ARI
La Rioja	19991010	Presidente	Izquierda
La Rioja	19991010	Presidente	PJ
La Rioja	19991010	Presidente	PRO
La Rioja	19991010	Presidente	Socialista
La Rioja	19991010	Presidente	UCR
La Rioja	19991010	Presidente	blanco
Mendoza	19991010	Presidente	ARI
Mendoza	19991010	Presidente	Izquierda
Mendoza	19991010	Presidente	PJ
Mendoza	19991010	Presidente	PRO
Mendoza	19991010	Presidente	Socialista
Mendoza	19991010	Presidente	UCR

Mendoza	19991010	Presidente	blanco
Misiones	19991010	Presidente	ARI
Misiones	19991010	Presidente	Izquierda
Misiones	19991010	Presidente	PJ
Misiones	19991010	Presidente	PRO
Misiones	19991010	Presidente	Socialista
Misiones	19991010	Presidente	UCR
Misiones	19991010	Presidente	blanco
Neuquen	19991010	Presidente	ARI
Neuquen	19991010	Presidente	Izquierda
Neuquen	19991010	Presidente	PJ
Neuquen	19991010	Presidente	PRO
Neuquen	19991010	Presidente	Socialista
Neuquen	19991010	Presidente	UCR
Neuquen	19991010	Presidente	blanco
Rio Negro	19991010	Presidente	ARI
Rio Negro	19991010	Presidente	Izquierda
Rio Negro	19991010	Presidente	PJ
Rio Negro	19991010	Presidente	PRO
Rio Negro	19991010	Presidente	Socialista
Rio Negro	19991010	Presidente	UCR
Rio Negro	19991010	Presidente	blanco
Salta	19991010	Presidente	ARI
Salta	19991010	Presidente	Izquierda
Salta	19991010	Presidente	PJ
Salta	19991010	Presidente	PRO
Salta	19991010	Presidente	Socialista
Salta	19991010	Presidente	UCR
Salta	19991010	Presidente	blanco
San Juan	19991010	Presidente	ARI
San Juan	19991010	Presidente	Izquierda
San Juan	19991010	Presidente	PJ
San Juan	19991010	Presidente	PRO
San Juan	19991010	Presidente	Socialista
San Juan	19991010	Presidente	UCR
San Juan	19991010	Presidente	blanco
San Luis	19991010	Presidente	ARI
San Luis	19991010	Presidente	Izquierda
San Luis	19991010	Presidente	PJ
San Luis	19991010	Presidente	PRO
San Luis	19991010	Presidente	Socialista
San Luis	19991010	Presidente	UCR
San Luis	19991010	Presidente	blanco
Santa Cruz	19991010	Presidente	ARI
Santa Cruz	19991010	Presidente	Izquierda
Santa Cruz	19991010	Presidente	PJ
Santa Cruz	19991010	Presidente	PRO
Santa Cruz	19991010	Presidente	Socialista
Santa Cruz	19991010	Presidente	UCR
Santa Cruz	19991010	Presidente	blanco



## 6 CORRIDAS DE PRUEBA

---

Santa Fe	19991010	Presidente	ARI
Santa Fe	19991010	Presidente	Izquierda
Santa Fe	19991010	Presidente	PJ
Santa Fe	19991010	Presidente	PRO
Santa Fe	19991010	Presidente	Socialista
Santa Fe	19991010	Presidente	UCR
Santa Fe	19991010	Presidente	blanco
Santiago del Estero	19991010	Presidente	ARI
Santiago del Estero	19991010	Presidente	Izquierda
Santiago del Estero	19991010	Presidente	PJ
Santiago del Estero	19991010	Presidente	PRO
Santiago del Estero	19991010	Presidente	Socialista
Santiago del Estero	19991010	Presidente	UCR
Santiago del Estero	19991010	Presidente	blanco
Tierra del Fuego	19991010	Presidente	ARI
Tierra del Fuego	19991010	Presidente	Izquierda
Tierra del Fuego	19991010	Presidente	PJ
Tierra del Fuego	19991010	Presidente	PRO
Tierra del Fuego	19991010	Presidente	Socialista
Tierra del Fuego	19991010	Presidente	UCR
Tierra del Fuego	19991010	Presidente	blanco
Tucuman	19991010	Presidente	ARI
Tucuman	19991010	Presidente	Izquierda
Tucuman	19991010	Presidente	PJ
Tucuman	19991010	Presidente	PRO
Tucuman	19991010	Presidente	Socialista
Tucuman	19991010	Presidente	UCR
Tucuman	19991010	Presidente	blanco

Cantidad de votos en total: 9000

## Referencias

- [1] Folk, Michael J. Zoellick, Bill. *File Structures*.
- [2] Smith, Peter. Barnes, G. *Files and Databases: An Introduction*. Addison-Wesley.
- [3] Documentación: Texmaker 3.0.2 (para la redacción de este informe)  
  
<http://www.xm1math.net/texmaker/>

## **7. Apendice - Enunciado TP Etapa 1**

A continuación se agrega el enunciado de la Etapa 1, correspondiente a este Trabajo Práctico.



UNIVERSIDAD DE BUENOS AIRES

FACULTAD DE INGENIERIA

DEPARTAMENTO DE COMPUTACIÓN

75.06 – ORGANIZACIÓN DE DATOS

**TRABAJO PRÁCTICO**

***VOTO ELECTRÓNICO***

***ETAPA 1***

---

SEPTIEMBRE 2011

# Índice General

<b>1</b>	<b>Introducción.....</b>	<b>3</b>
<b>2</b>	<b>Enunciado.....</b>	<b>4</b>
2.1	Entidades .....	6
2.2	Estructuras de Datos .....	6
2.3	Accesorios .....	7
<b>3</b>	<b>Criterio de aprobación .....</b>	<b>8</b>
3.1	Entrega .....	8
3.2	Documentación.....	8
<b>4</b>	<b>Referencias .....</b>	<b>10</b>

# 1 Introducción

Este documento consiste en el enunciado del trabajo práctico de la asignatura. En el mismo se especifican los requerimientos de cada etapa de entrega, dejando de lado el cronograma de entregas que se encuentran en la página o grupo de correo de comunicación de la cátedra respectivamente.

Toda aclaración, indicación o respuesta a consultas (ofrecidas en clase o mediante el grupo yahoo) serán tomadas como extensión y parte explícita de este enunciado.

La forma de trabajo con los grupos es descripta en el Reglamento de Trabajos Prácticos de la Cátedra ([http://materias.fi.uba.ar/7506C/blog/?page\\_id=9](http://materias.fi.uba.ar/7506C/blog/?page_id=9))

El trabajo consistirá en implementar una aplicación capaz de resolver la problemática del voto electrónico.

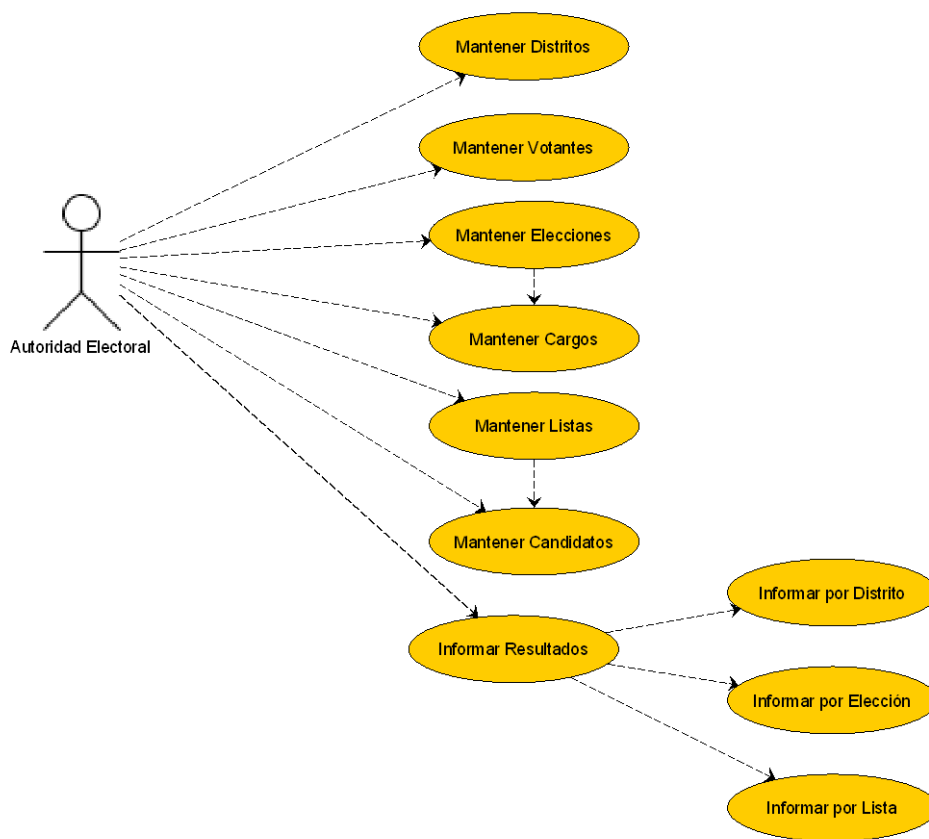
## 2 Enunciado

La aplicación deberá ser capaz de mantener información sobre las entidades que componen el sistema de votaciones (votantes, elecciones, candidatos, etc.), y de proveer la posibilidad a un votante de emitir su voto para la correspondiente elección.

La Primer Entrega se avocará a la aplicación de los conceptos de Organización de Archivos.

### Funcionalidades mínimas

Tipo: URNA ELECTRÓNICA DE REGISTRO DIRECTO



**Mantener Distritos:** Es el Alta, Baja y Modificación de los Distritos que delimitan las elecciones.

**Mantener Votantes:** Es el Alta, Baja y Modificación del padrón de votantes en la elección.

**Mantener Elecciones:** Es el Alta, Baja y Modificación del evento electoral.

**Mantener Cargos:** Es el Alta, Baja y Modificación de los cargos a ser electos en la Elección.

**Mantener Listas:** Es el Alta, Baja y Modificación de las listas de candidatos a los cargos a elegirse en un distrito.

**Mantener Candidatos:** Es el Alta, Baja y Modificación de los Candidatos, que deben ser votantes, a los cargos puestos en juego en la elección distrital.

**Informar Resultados:** Es el Alta de informes a mostrar por pantalla bajo tres posibles criterios, un Distrito determinado, una Elección determinada o una Lista determinada.

*Almacenamiento y Archivos de Control para las Entidades* = Se encuentran todos dentro de un directorio, especificado a través de un archivo de configuración de la aplicación, y pueden tener jerarquía de subdirectorios interna. Es donde se guarda toda la información necesaria para poder funcionar.

*Comandos por consola* = Toda la interacción del usuario con el sistema se realiza a través de comandos por consola.

*Archivos con resultados* = Como respuesta a toda interacción, el sistema generará archivos de registro de operaciones (LOGs) en el directorio donde se llame a la aplicación.

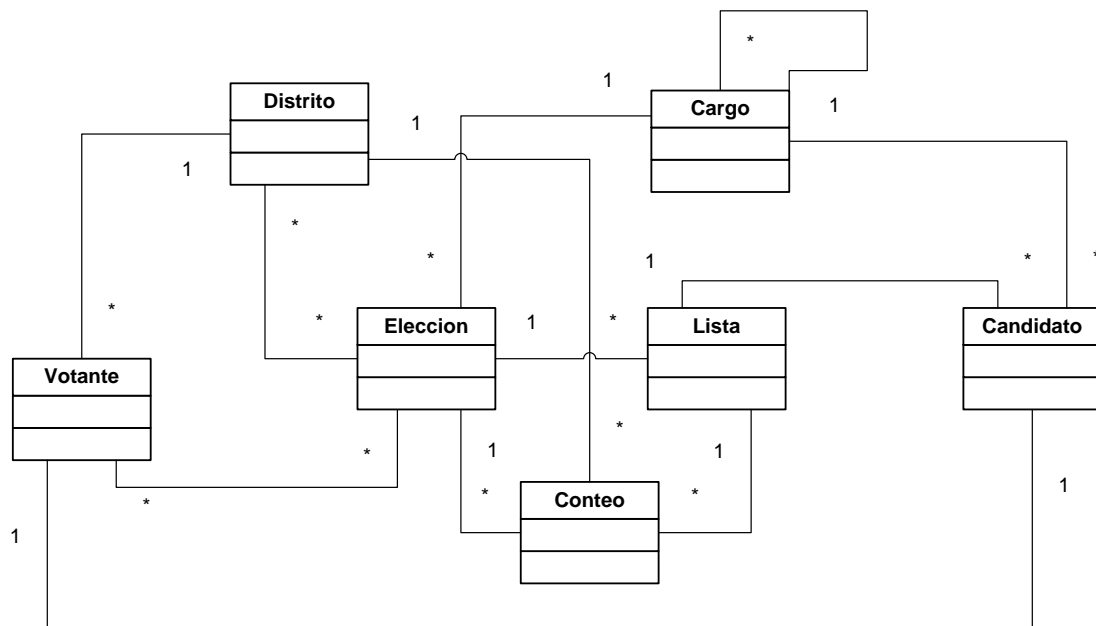
La resolución del trabajo práctico debe ser realizada en plataforma Linux y lenguajes C o C++ (preferentemente respetando el estándar ANSI), y la entrega debe constar de un Makefile para su compilación. Además debe funcionar en calidad de Usuario (user) del sistema operativo.



## 2.1 Entidades

Respecto a las entidades que conformarán la información necesaria para resolver esta problemática, a continuación presentamos un diagrama básico de entidades y sus relaciones.

El objetivo de esta etapa es resolver la aplicación, utilizando las herramientas brindadas por el módulo Organización de Archivos. Por ello se plantea la siguiente relación entre esos conceptos.



**Distrito** ((distrito)i)

**Votante** ((DNI)i, NombreApellido, clave, domicilio, (distrito)ie, ((eleccion)ie)\*)

**Eleccion** ((fecha, (carga)ie)i, ((distrito)ie)+)

**Lista** (((eleccion)ie, nombre)i)

**Conteo** (((lista)ie, (distrito)ie, (eleccion)ie)i, cantidad)

**Candidato** (((lista)ie, (votante)ie, (carga)ie)i)

**Carga** ((carga)i, (carga)\*)

**Administrador** ((usuario)i, clave)

## 2.2 Estructuras de Datos

Para la implementación, deberán elegirse, con criterios bien definidos y explicados por el grupo, las estructuras de datos en disco que se utilizarán para cada uno de los archivos previamente definidos.

Es importante presentar al tutor o docente a cargo del grupo la estrategia a utilizar para resolver el TP para validar que sea correcta y cumpla los requerimientos mínimos de complejidad que requiere el mismo.

Las estructuras a elegir son las siguientes:

- Árbol B
- Árbol B+
- Árbol B#
- Dispersión Extensible
- Dispersión Fija con Zona de Desborde.
- Archivos de Registros Variables
- Archivos de Registros Fijos
- Archivos de Bloques

La resolución del trabajo práctico debe ser realizada en plataforma Linux y lenguajes C o C++ (preferentemente respetando el estándar ANSI).

## **2.3      Accesorios**

Deberán, además, crearse votantes aleatorios automatizados para resolver el ingreso de votos de varias elecciones, que muestren la correcta funcionalidad del sistema.

Además, será necesario registrar en un archivo de LOG todas las operaciones del votante: Acceso, Acceso Fallido, voto, voto ingresado, cambio de voto.

## 3 Criterio de aprobación

Como se especifica en el Reglamento de la materia existe un criterio mínimo para poder acceder a una re-entrega en cada etapa. A continuación se menciona una lista de requerimientos que forman parte de dicho criterio. No cumplir con alguna de ellas implica no cumplir el mínimo requerido. Pero no vale la inversa, es decir, cumplir con ellas no implica cumplir con el criterio mínimo.

### 3.1 *Entrega*

La entrega debe constar de un Makefile para su compilación, y el sistema debe funcionar en calidad de Usuario (user) del sistema operativo.

Además, deberá entregarse la documentación que se describe en la página de la materia

([http://materias.fi.uba.ar/7506C/docs/wiki/doku.php?id=tp:requisitos\\_de\\_documentacion](http://materias.fi.uba.ar/7506C/docs/wiki/doku.php?id=tp:requisitos_de_documentacion))

Los grupos a formar serán de 5 integrantes.

El periodo de resolución de esta etapa del trabajo práctico es de 5 semanas de la fecha de presentación del mismo (Sábado 22/10 00hs).

### 3.2 *Documentación*

#### ▪ *General*

- Diagrama de clases o módulos (según corresponda)
- Especificación de cada clase o módulo (según corresponda)
- Diagramas de secuencia o intercambio de mensajes entre capas. Mostrar escenarios.
- Planificación (identificación de tareas, estimación de duración y asignación)
- Bugs conocidos
- Manual de usuario. Indicaciones generales del trabajo práctico, modo de instalación y ejemplos de uso.

#### ▪ *Física – Organización*

##### *Organización de registros*

- ¿Cómo delimitan la longitud de un registro y de un campo variable?. Mostrar los campos que posee y cuanto espacio ocupa cada uno.
- Indicar que información administrativa se utiliza.

#### ▪ *Índices – Búsqueda*

##### *Hashing*

- Función de hashing utilizada. Criterio de elección.

- Tamaño de Buckets.
- ¿Cuál es el factor de empaquetamiento que utilizan?

### ***Árbol B+***

- *Sequence set*: ¿Cómo delimitan la longitud de un registro y de un campo variable?. Mostrar los campos que posee y cuanto espacio ocupa cada uno. Política utilizada para split / concatenación de bloques. ¿Hacen algún tipo de redistribución?
- *Index set*: Mostrar los campos que posee y cuanto espacio ocupa cada uno. ¿Cómo se generan los separadores? ¿Cómo se eliminan separadores? Indicar que condiciones deben surgir en cada caso. ¿Existe algún tipo de concatenación de separadores dentro de la página? ¿Qué método se utiliza para la búsqueda de un separador dentro de la página? (binaria, secuencial, otro).

## 4 Referencias

Folk, Michael. Zoellick, Hill. *File Structures*. 724 páginas.

Smith, Peter. Barnes, G. *Files and Databases: An Introduction*. Addison-Wesley.