



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

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

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)

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

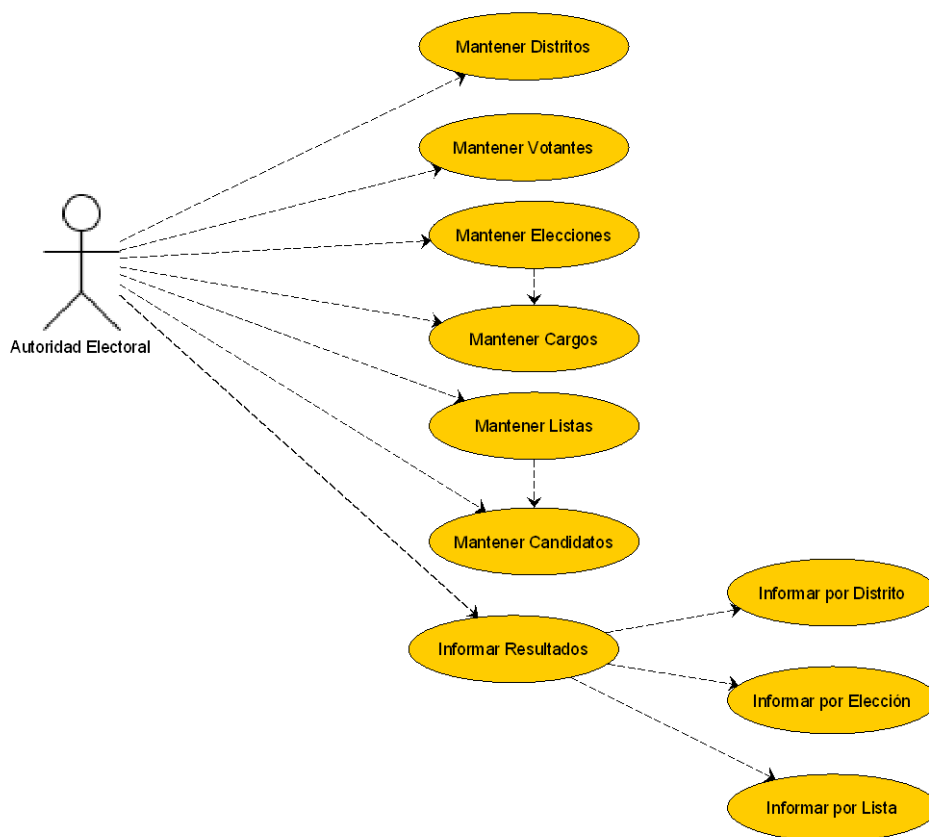
2 Enunciado

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.

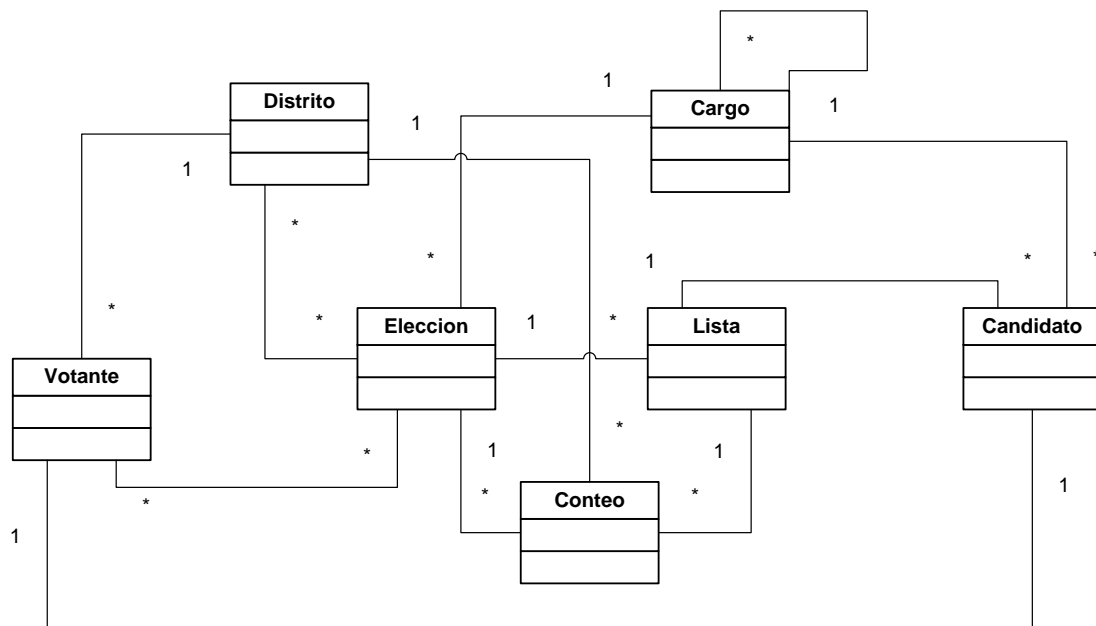
Archivos con resultados = Como respuesta a toda interacción, el sistema generará archivos de registro de operaciones (LOGs) en el directorio donde se llame a la aplicación.

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)

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.