

Registrar archivos txt

```
public boolean registrarPartidoTxt(String archivo) {  
    try {  
        lo sobrescribe  
        PrintWriter escritor = new PrintWriter(new FileWriter(archivo,true)); // true incluye, si no esta  
        String registro=this.sigla+";"+this.nombre+";"+this.habilitado;  
        escritor.println(registro);  
        escritor.close();  
    } catch (IOException e) {  
        // TODO Auto-generated catch block  
        return false;  
    }  
    return true;  
}
```

Se debe crear una interfaz para generar los archivos

```
package y;  
  
public interface Archivos {  
    static final String archivoPartidos="partidos.txt";  
    static final String archivoRecintos= "recintos.txt" ;  
    static final String archivoVotante="votantes.txt";  
}
```

Leer archivos txt

```
static ArrayList<PartidoPolitico> leerPartidosTxt(String archivo){
    ArrayList<PartidoPolitico> partidos = new ArrayList<PartidoPolitico>();

    try {
        BufferedReader lector = new BufferedReader(new FileReader(archivo));
        String linea;
        while((linea = lector.readLine()) != null) {
            String [] datos = linea.split(";");
            if (datos.length < 3) {
                System.out.println("Línea inválida: " + linea);
                continue; // ignora la línea malformada
            }
            String sigla = datos[0];
            String nombre = datos[1];
            boolean hab = Boolean.parseBoolean(datos[2]);
            partidos.add(new PartidoPolitico(sigla, nombre, hab));
        }
        lector.close();
    } catch (FileNotFoundException e) {
        System.out.println("Ha ocurrido un error al encontrar el archivo");
    } catch (IOException e) {
        System.out.println("Ha ocurrido un error al recibir los datos");
    }

    return partidos;
}
```

Reescribir

```
static boolean reescribirPartido(ArrayList<PartidoPolitico> partidos, String archivo) {  
    try {  
        PrintWriter escritor = new PrintWriter(new FileWriter(archivo,true));  
        for (PartidoPolitico pp : partidos) {  
            String registro=pp.getSigla()+";"+pp.getNombre()+";"+pp.isHabilitado();  
            escritor.println(registro);  
        }  
        escritor.close();  
    } catch (IOException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
    return true;  
}
```

Usa esto en los constructores

SetResizable(false);

setBounds(100, 100, 656, 399); tamaños

Coloca un boton y luego limpia

```
txtNombre.setText("");  
txtSigla.setText("");
```

Registrar con botón

```
JButton btnregistrar = new JButton("Resgistrar");
panBotones.add(btnregistrar);
btnregistrar.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        String sigla= txtSigla.getText() ;
        String nombre= txtNombre.getText() ;
        PartidoPolitico pp = new PartidoPolitico(sigla,nombre);
        if(pp.registrarPartidoTxt(Archivos.archivoPartidos)) {
            JOptionPane.showMessageDialog(btnregistrar, "Se registro correctamente");
            txtNombre.setText("");
            txtSigla.setText("");
        }
        else {
            JOptionPane.showMessageDialog(btnregistrar, "Ocurrio un error (intentaste registrar a
evo.. )");
        }
    }
});
```

Leer

```
// Leer votantes desde archivo
ArrayList<Votante> votantes = Votante.LeerVotanteTxt(Archivos.archivoVotante);

// Crear arreglo de Strings para el JList
String[] datosVotantes = new String[votantes.size()];
for (int i = 0; i < votantes.size(); i++) {
    datosVotantes[i] = votantes.get(i).toString();
}
```

```
String[] datosVotantes1 = new String[votantes.size()];
for (int i = 0; i < votantes.size(); i++) {
    datosVotantes1[i] = votantes.get(i).toString(); // Usar toString() de Votante
}
JList<String> listVotantes = new JList<>(datosVotantes);
listVotantes.setBackground(new Color(192, 192, 192));
JScrollPane scrollVotantes = new JScrollPane(listVotantes); //importante si usas scrool
panCentro.add(scrollVotantes, BorderLayout.CENTER);
```