

**FORMATO DE GUÍA PARA PROYECTOS INTEGRADORES****CARRERA:** Ingeniería en Ciencias de la Computación**NIVEL:** Segundo**MATERIAS Y DOCENTES INVOLUCRADOS**

Docente	Materia
Walter Gaibor	Estructura de Datos
Rodrigo Tufiño	Programación Orientada a Objetos
Alberto Duchi	Programación Orientada a Objetos

**LÍDER DEL GRUPO:** Alberto Duchi**CORREO INSTITUCIONAL:**  
aduchi@ups.edu.ec**TEMA DEL PROYECTO:** Sistema de votación informático**OBJETIVO DEL PROYECTO:**

- Desarrollar una aplicación de escritorio funcional utilizando el paradigma orientado a objetos y estructura de datos

**DESCRIPCIÓN DEL PROYECTO:**

Se requiere la elaboración de un sistema informático, el cual permita simular el funcionamiento del sistema de votación electrónica (elección en la segunda vuelta). El sistema registra las votaciones de una junta electoral, emitirá el certificado de votación y genera el acta con los resultados. Para implementar este sistema, se debe poner en práctica todos los conocimientos adquiridos en las materias de Programación Orientada a Objetos (POO) y Estructura de Datos.

**Requerimientos funcionales**

Para su normal funcionamiento el aplicativo debe disponer de la siguiente parametrización inicial de funcionamiento:

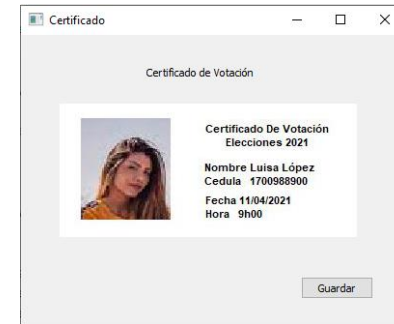
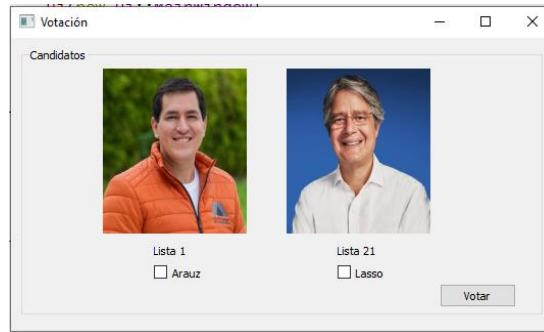
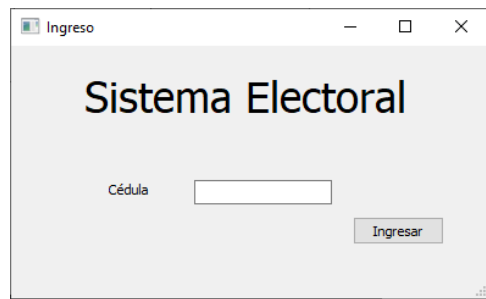
- Listado de personas habilitadas para el sufragio (padrón electoral)
- Fotos de las personas habilitadas para el sufragio (para generar la papeleta de votación).
- La fecha del proceso de votación.
- Hora de inicio y cierre del proceso de electoral.
- Candidatos presidenciales (fotos, lista a la que pertenecen), opciones de voto blanco y voto nulo.

**Módulo de parametrización inicial**

En primer lugar, el usuario administrador ingresa los parámetros necesarios para la puesta en marcha del proceso electoral considerando la lista de habilitadas para el sufragio, los candidatos, opciones de voto nulo y voto blanco, la fecha, las horas de los comicios y encera los contadores. Para la carga de información de los electores, se debe considerar el número de cédula, nombres completos y la fotografía del ciudadano.

### Módulo de votaciones

En segundo lugar, los usuarios (votantes) acceden al sistema con su número de cédula para realizar su votación de forma electrónica. El sistema debe validar que el ciudadano conste en el padrón electoral, registrado en el proceso anterior. Además, se debe validar que se encuentre dentro del horario de votación permitido. Una vez que el usuario realice el ingreso, deberá seleccionar el candidato de su preferencia, u opción de voto blanco o voto nulo. Internamente el sistema deberá almacenar el criterio de votación: fecha, hora de la votación. Cabe mencionar que estos datos deben ser almacenada en un dispositivo de almacenamiento secundario (disco) para evitar pérdidas de información. Luego de realizar la votación, el aplicativo debe generar el certificado de votación correspondiente permitiendo al elector la descarga de este documento en formato de imagen.



### Módulo de resultados

Por último, al momento de finalizar las votaciones el usuario delegado del CNE podrá ejecutar la pantalla de resultados donde se visualizará el número de votos que obtuvo cada candidato, los votos nulos y blancos. Esta información se podrá descargar en un archivo CSV (Excel) y una imagen de los resultados. Adicional, se debe generar un histograma por horas con el número de votantes que acudieron a la junta electoral.



**Requerimientos no funcionales:**

El diseño y desarrollo de la aplicación debe realizarse considerando los siguientes requisitos:

**Programación Orientada a Objetos**

- Modelar el sistema utilizando diagramas de clases y bocetos para interfaces.
- Programar aplicando los conceptos de POO y utilizando el patrón MVC.
- Aplicar conceptos de diseño UX para las interfaces de usuario.
- Almacenar correctamente la información.
- Funcionar correctamente sin errores.

**Estructura de Datos**

- Utilizar al menos dos estructuras de datos diferentes para manejar la información de la aplicación (Colas, Pilas, Listas, árboles, etc.)

**Puntos extra**

Se consideran puntos extra para las siguientes actividades:

- Utilización de GitHub para la gestión del código fuente entre los miembros del grupo.
- Crear un instalador de la aplicación para Microsoft Windows.
- Consideración de las siguientes Idiomas: español, inglés, quichua

**RESULTADOS DE APRENDIZAJE QUE SE EVALUARÁN EN EL PROYECTO INTEGRADOR**

MATERIA	RESULTADOS DE APRENDIZAJE
1. Estructura de datos	Identifica las características y generalidades de las técnicas para realizar programación dinámica.
2. Programación Orientada a Objetos	Utiliza el patrón MVC para el desarrollo de aplicaciones de software. Crea interfaces gráficas de usuario empleando técnicas de modelado de objetos con estándares de internacionalización.

**HERRAMIENTAS SUGERIDAS PARA EL PROYECTO INTEGRADOR**

Herramientas	MATERIAS	
	Estructura de Datos	Programación Orientada a Objetos
	C++	C++ / Framework Qt
		Qt Creator
		GitHub
		Diagramador

**PLANIFICACIÓN DE AVANCES Y NOTAS**

SEGUNDO PARCIAL				
Estructura de Datos	ACTIVIDAD O ENTREGABLE	FECHA DE ENTREGA	DESCRIPCIÓN	NOTA PARCIAL
			Análisis de estructuras de datos a implementar	2
			Revisión de la estructura de datos árbol a desarrollar	2
	NOTA TOTAL DE AVANCES			4
	NOTA DE DEFENSA DEL PROYECTO INTEGRADOR			6
Programación Orientada a Objetos	ACTIVIDAD O ENTREGABLE	FECHA DE ENTREGA	DESCRIPCIÓN	NOTA PARCIAL
	Diagramas		Diseño de interfaces gráficas y diagrama de clases.	4
	Prototipo funcional		Implementación de clases, módulo de administración	6
	NOTA TOTAL DE AVANCES			10
	NOTA DE DEFENSA DEL PROYECTO INTEGRADOR			10

### RUBRICA PARA DEFENSA DEL PROYECTO INTEGRADOR

<b>TEMA</b>	: Sistema de votación informático
<b>NIVEL</b>	Segundo
<b>CARRERA</b>	Ingeniería en Ciencias de la Computación
<b>MATERIAS</b>	Estructura de Datos Programación Orientada a Objetos
<b>DOCENTES</b>	Walter Gaibor Rodrigo Tufiño Alberto Duchi
<b>PERIODO</b>	58 (marzo 2021 – agosto 2021)
<b>FECHA DE ELABORACIÓN</b>	23/04/2021
<b>PUNTAJE TOTAL DE LA DEFENSA</b>	10

MATERIA	DIMENSIÓN	RESULTADO DE APRENDIZAJE	EXCELENTE (2 PUNTOS)	BUENO (1.5 PUNTOS)	MALO (0.9 PUNTOS)	DEFICIENTE (0.2 PUNTOS)	NOTA DE DEFENSA	NOTA DE AVANCES
Estructura de Datos	Estructuras de Datos	Identifica las características y generalidades de las técnicas para realizar programación dinámica.	Las estructuras de datos que se utilizan en el proyecto son 100% dinámicas.	Las estructuras de datos que se utilizan en el proyecto son 75% dinámicas.	Las estructuras de datos que se utilizan en el proyecto son 50% dinámicas.	No utiliza estructuras dinámicas.		
	Árbol binario	Identifica las características y generalidades de las técnicas para realizar programación dinámica.	El árbol binario despliega la información del nombre y la nota de cada estudiante.	El árbol binario despliega la información del nombre, pero no la nota de cada estudiante.	El árbol binario despliega la nota sin saber de que estudiante es la nota.	El árbol no despliega información.		
Materia	DIMENSIÓN	RESULTADO DE APRENDIZAJE	EXCELENTE (2.5 PUNTOS)	BUENO (1.5 PUNTOS)	MALO (1 PUNTO)	DEFICIENTE (0.2 PUNTOS)	NOTA DE DEFENSA	NOTA DE AVANCES

	Patrón MVC	Construye el sistema aplicando correctamente el paradigma de programación orientada a objetos (POO) y utilizando el patrón de diseño MVC.	Aplica todos los conceptos de POO estudiados en las diferentes partes del sistema y utiliza el patrón MVC durante la programación.	Aplica parcialmente los conceptos de POO estudiados en las diferentes partes del sistema y/o no es clara la utilización del patrón MVC durante la programación.	Utiliza pocas clases para resolver el problema planteado. Mucho código se encuentra disperso y/o desordenado, y/o no utiliza correctamente el patrón MVC.	Toda la aplicación se encuentra programada dentro de las interfaces gráficas y/o no existen clases y/o no utiliza MVC para la programación	10	10
	Interfaces gráficas.	Crea interfaces gráficas de usuario empleando técnicas de modelado de objetos con estándares de internacionalización.	La interfaz gráfica es clara, tiene estructura y se adapta a los requerimientos solicitados.	La interfaz gráfica es clara pero poca relación con el contenido y con el requerimiento solicitado	La interfaz gráfica es clara y tiene una escasa relación el contenido y con lo solicitado en el planteamiento del problema.	La interfaz gráfica es confusa no tiene un orden establecido		
	Almacenamiento	Maneja errores y guarda información en dispositivos de almacenamiento secundarios.	Genera, lee y muestra los archivos solicitados de manera correcta. El aplicativo se ejecuta y produce los resultados correctos y cumple con todas las especificaciones solicitadas.	Genera, lee y muestra los archivos solicitados de manera parcial. El aplicativo se ejecuta y produce los resultados correctos y cumple con la mayoría de las especificaciones.	Tiene dificultades al genera, lee y mostrar los archivos solicitados. El aplicativo produce los resultados esperados, pero no los muestra correctamente.	No genera, lee y mostrar los archivos solicitados. El aplicativo muestra datos incorrectos.		

	Funcionamiento	Crea aplicaciones informáticas de escritorio funcionales.	El programa funciona y cumple con todos los requerimientos	El programa funciona parcialmente. Algunos requerimientos no han sido cumplidos	El programa tiene errores y no funciona bien. Existen requerimientos que no han sido programados	El programa no funciona.		
NOTA TOTAL DEL PROYECTO INTEGRADOR							20	

FIRMA DE LOS DOCENTES	<p><b>Estructura de Datos:</b>  <b>Docente:</b> Walter Gaibor</p> <p><b>Programación Orientada a Objetos:</b>  <b>Docente:</b> Rodrigo Tufiño  <b>Docente:</b> Alberto Duchi</p>
-----------------------	--