Informe de proyecto [Elecciones by Service Seller]

Edison Arámbulo[L00404957] and Mateo Beltrán[L00416700]

Universidad de las Fuerzas Armadas
eaarambulo@espe.edu.ec
msbeltran@.espe.edu.ec

Abstract

En el presente informe se da a conocer de manera narrativa la creación de un programa que ayudará a los ciudadanos saber lo más importante de las próximas elecciones en la República del Ecuador. Para obtener lo planteado se utilizó el lenguaje de programación Python mediante una programación orientada a objetos. El programa está constituido por una base de datos que permitirá almacenar y buscar datos de los ciudadanos para las elecciones. Para utilizar de una manera más simple se utilizó una interfaz gráfica creada mediante Tkinter. Se utilizó una clase denominada feriado, para darle ciertas restricciones al ciudadano al momento de las elecciones. Para poder hacer actualizaciones a nuestro programa, se versionó, el respectivo link se encontrará en la sección de resultados.

1 Introducción

Las elecciones generales y locales en la República del Ecuador se celebran cada cuatro años. Ecuador tiene tres apartados los cuales comienza por la elección del Presidente y Vicepresidente. La segunda elección consiste en alcaldes y prefectos de cada provincia. Finalmente, la elección de diputados a nivel provincial y nacional. Las personas que deben participar de manera obligatoria en las elecciones deben tener la cédula vigente, ser mayor a 18 años y menor a 65 años. Para quienes se encuentran en la edad de 16 a 18 años y mayores a 65 años su voto se lo considera facultativo (no obligatorio). El voto es personal y secreto, ninguna entidad política puede obligar a los ciudadanos ecuatorianos a elegirlos como la mejor opción [1]. Cuando un ciudadano se dirige a su lugar de sufragio, va a presenciar una respectiva "mesa", la cual se le denomina "Mesa Electoral". La Mesa Electoral puede estar constituida por estudiantes superiores, trabajadores publicos o privados, los cuales son sorteados por un software específico [2].

La programación orientada a objetos (POO) permite al programador tener un código fuente óptimo. Un programador tiene las herramientas necesarias para dividir el programa completo en secciones, las mismas que están interecomunicadas. Cuando se divide el programa, se obtienen clases que contienen métodos y atributos propios. La clase es el conjunto de varios objetos que presentan características en común (atributos). Cuando se describe el concepto de herencia en POO, corresponde a una dependecia de una súper clase o clase padre. La clase dependiente se la denomina una subclase o clase hija, y está relacionada directamente con la clase padre, debido a que hereda sus atributos y sus métodos. A esa acción de heredar, se la denomina cardinalidad y permite una mejor optimización de código fuente. Los métodos de las respectivas clases, son procedimiento o funciones que ejecutan alguna tarea en específico [3].

El presente informe se da a conocer el proyecto de la unidad dos. Mediante un buen análisis del problema planteado se podrá realizar un método de cuatro pasos para lograr un programa coherente. Esto se logrará con los conceptos estudiados en el transcurso de la unidad. Se hará la buena práctica de docstring y con esto se tendrá un código óptimo y lógico. Además, para futuras actualizaciones se versionó el respectivo programa.

Contexto y motivación Los ecuatorianos en ciertas circunstacias no saben con exactitud si deben estar presentes en la mesa electoral. También existen ciertos aspectos para un voto facultativo, por ende, el aplicativo ayudará a los ciudadanos a identificar su posición en las elecciones. Mediante las respectiva Ley Orgánica Electoral se definieron los respectivos escenarios para los ecuatorianos, si son o no miembros de mesa electoral, si tienen un voto facultativo o la obligación de dirigirse a sufragar.

Problema de estudio Las diferentes restricciones para formar parte de una mesa electoral confunden a los ciudadanos. Pueden formar parte de esta mesa electoral trabajadores públicos o privados y estudiantes de nivel superior, pero se lo hace de una manera aleatoria. El presente programa es importante porque se dará a conocer al ciudadano si está o no apto para ser miembro de la mesa electoral. Mediante esta información el ciudadano puede estar preparado para sufragar y para ser miembro de mesa.

Trabajo relacionado En el año 2014 en la provincia de Santo Domingo de los Tsáchilas se utilizó un software que le permite al ciudadano votar de manera rápida y ágil. Se utilizó un software denominado **Smartmatic Auditable Election Systems** (SAES) que proviene de la empresa Smartmatic. Para la mejor utilización de este sistema, se hicieron respectivas capacitaciones, para el voto facultativo, se visitaron diferentes colegios. Para el voto obligatorio se hicieron campañas en los diferentes barrios y sectores de la provincia Tsáchila. Mediante este software, la provincia logró ser la primera en dar los respectivos resultados confiables y legítimos del día de las elecciones[4].

En el año 2016, específicamente en España, se implantó un bosquejo de un sistema electoral que llevará elecciones rápidas y transparentes. El Comité de Ministros del Consejo de Europa recomendó al gobierno español que los ciudadanos tenga una previa capacitación para que las elecciones sean ágiles. El software que se promocionó contiene diferente información de los candidatos y sus propuestas, los ciudadanos tendrán la capacidad de decir por cual partido elegir. El aplicativo propuesto se simplificará el tiempo de conteo de votos y por ende, se tendrán resultados más rápidos[5].

Resumen de contribución Service Seller proporcionará a los ecuatorianos información valiosa para poder dirigirse a sufragar. Los ciudadanos sabrán con exactitud si rol en las próximas elecciones, lo más importante es saber si son miembros de mesa electoral. Además, se les impartirá ciertas directrices para los días previos de las elecciones, con esto estarán bien informados. Todo lo que se plantea en el programa se rige a la Ley Orgánica Electoral.

2 Antecedentes

Alrededor del año 1990 se inician muchos sucesos para el desarrollo del software libre, Guido van Rossum es uno de los promotores a este avance tecnológico, que en la actualidad se evidencia a gran escala.Guido van Rossum es el creador del lenguaje de programación Python, que es el que abre las puertas al desarrollo de tecnologías libres que permite a los usuarios analizar, utilizar,

mejorar un software en específico. Python es un lenguaje de programación muy avanzado y ágil, al momento de codificar es muy sencillo a comparación de otros lenguajes como por ejemplo el lenguajes de programación C o C++, con esto se da un mejor manejo para utilizar en programación orientada a objetos, programación que será utilizada en este proyecto [6].

En programación orientada a objetos surgen conceptos como la clase, objeto y método que corresponde a la misma clase. Una clase se define como una agrupación de elementos que tienen características en común. A estos elementos se los denominan objetos y cada uno de estos objetos contiene atributos específicos, cada objeto tiene su trabajo en la clase. Un método es una acción que el programador describe o formula para darle solución al problema planteado, una clase pueden contener varios objetos y varios métodos. Cuando se menciona el conjunto de métodos y los respectivos datos dentro de una misma clase se define como el encapsulamiento, lo cual permite la interacción de datos en esa misma clase, no se puede interactuar datos de una clase con otra clase independiente.

3 Método

Ahora viene la parte "medular" del reporte, donde explica qué realizo para desarrollar su producto software. Nuevamente, organícelo en párrafos con títulos. Como en cada sección comienza con una descripción muy breve de la sección.

3.1 Diseño

Presente su diseño de manera concisa. Concéntrese en aspectos novedosos, pero evite los detalles de implementación. Utilice pseudocódigo (código fuente) y figuras para explicar mejor su diseño, pero también preséntelos y explíquelos en el texto.

3.2 Análisis

Argumente sobre la corrección de su diseño y por qué espera que funcione mejor que trabajos ya existentes. Si corresponde, mencione cómo se relaciona su diseño con los límites teóricos.

3.3 Optimización

Explique cómo optimizó su diseño y lo ajustó a situaciones específicas. En general, tan importante como los resultados finales es demostrar que adoptó un enfoque estructurado y organizado para la optimización y que explica por qué hizo lo que hizo. Tenga cuidado de argumentar por qué su optimización no rompe la corrección de su diseño, establecido anteriormente. A menudo es una buena estrategia explicar un diseño o protocolo en refinamientos escalonados, para convencer más fácilmente al lector de que es correcto.

3.4 Implementation

No es necesario explicar su código. Sin embargo, en algunos casos puede ser relevante destacar las contribuciones adicionales dadas por su implementación.

Ejemplos de tales contribuciones son:

- *Abstracciones y módulos*: si su implementación está bien separada en módulos interactivos con responsabilidades separadas, podría explicar esta estructura y por qué es buena/mejor que alguna otra estructura alternativa.

- *Optimización*: si dedica mucho tiempo a optimizar su código, por ejemplo, utilizando herramientas de creación de perfiles, el resultado de dicha optimización puede presentarse como una contribución. En este caso, razón por la que el código optimizado funciona mejor.
- *Marco de evaluación*: si implementó un marco o una aplicación para evaluar su implementación contra el trabajo existente, o en un escenario específico, este marco puede presentarse como una contribución.

4 Evaluación Experimental

Aquí evalúas tu trabajo mediante experimentos. Empiece de nuevo con un resumen muy breve de la sección. Sigue la estructura típica.

4.1 Configuración experimental

Especifique el contexto y la configuración de sus experimentos. Esto incluye, por ejemplo, qué hardware (VM) está ejecutando, qué sistema operativo están ejecutando estas máquinas, cómo están conectadas, ...

También explique cómo genera carga para su sistema y qué parámetros utilizó aquí. La idea general es incluir suficiente información para que otros puedan reproducir sus experimentos. Con ese fin, debe proporcionar un conjunto detallado de instrucciones para repetir sus experimentos. Estas instrucciones no deben incluirse en el informe, pero deben proporcionarse como parte del repositorio de código fuente en GitHub, generalmente como el archivo texttt README.md, o como scripts de Shell o scripts de Ansible.

Si sus experimentos dan resultados extraños o inesperados, analice, perfile y depure su código. **No repita simplemente los experimentos hasta que den los resultados esperados.**

Por último, ejecutar experimentos consume mucho tiempo y es posible que deba pasar por varias rondas de experimentos, depuración y optimización. **No demore la ejecución de experimentos hasta el final del período de su proyecto.**

4.2 Resultados

Los resultados de sus experimentos. Compare diferentes variantes de su diseño (por ejemplo, con y sin optimizaciones) o compare el rendimiento con otros diseños o sistemas. Los gráficos deben mostrar el promedio de varias ejecuciones (al menos 10 como regla general), incluidas las barras de error, los percentiles o los valores mínimo/máximo. Realice experimentos para evaluar su sistema en condiciones normales de funcionamiento, cuando experimente fallas o ataques, o con diferentes cargas de trabajo.

5 Conclusión

Aquí debe resumir lo que hizo y por qué esto es importante. No tome el resumen y lo ponga en tiempo pasado. Recuerde, ahora el lector ha leído (con suerte) el artículo, por lo que es una situación muy diferente a la del resumen. Trate de resaltar los resultados importantes y diga las cosas que realmente desea transmitir, como declaraciones de alto nivel (por ejemplo, creemos que ... es el enfoque correcto para ... LAN, la técnica .... debería ser aplicable ....). Ser breve.

References

[1] S. Pachano, "El proceso electoral de ecuador," *Quito, Ecuador: FLACSO*, 2004.

[2] J. ELECTORAL, "Elecciones 2020," *Quito (Ecuador)*, 2000.

[3] J. Ada, "Conceptos de poo,"

[4] N. V. Carreño, "Voto electrónico en ecuador: Un reto cumplido y la puerta a la automatización electoral," *Revista Elecciones*, vol. 13, no. 14, pp. 75–82, 2014.

[5] M. Presno, "Premisas para la introducción del voto electrónico en la legislación electoral española (premises regarding the introduction of electronic voting (e-voting) into the spanish electoral law)," *Revista de Estudios Polıticos*, vol. 173, pp. 277–304, 2016.

[6] I. Challenger-Pérez, Y. Dıaz-Ricardo, and R. A. Becerra-Garcıa, "El lenguaje de programación python," *Ciencias Holguın*, vol. 20, no. 2, pp. 1–13, 2014.