

## PROYECTO LABORATORIO PROGRAMACIÓN II ESPECIFICACIONES GENERALES V-1.0

El día 31 de mayo debemos presentar una Propuesta de Valor y hacer una exposición del mismo en la IV Feria de ExpoRAIS. Debo conversar personalmente con el Profesor Gerard Páez Monzón en lo referente a nuestra intervención en la expoRAIS. Tenemos 17 días (tiempo más que suficiente, puesto que somos muchos) para tener un producto a presentar.

Lo que quiero presentar para ese día es lo siguiente:

**Producto:** PR2 Taxes Control System V-1.0

Un sistema de recaudación de impuestos programado en C++, haciendo uso del entorno de desarrollo gráfico QtCreator para escritorio. Nuestra primera propuesta de valor debe mostrar las potencialidades del sistema y mostrar el nivel de programación obtenido durante el curso.

### Filosofía del producto:

Se quiere crear un producto que permita facilidad de recaudación de impuestos para un grupo de usuarios (contribuyentes), y permita a otros (administradores) ver lo recaudado hasta el momento.

### Esquema de funcionalidad:

La primera ventana debe mostrar un login. Si el login es incorrecto, debe mostrar un diálogo donde se indique el error. De acuerdo a si el usuario es administrador o contribuyente, se deben mostrar distintas cosas.

Si el usuario es contribuyente, se debe mostrar la siguiente secuencia:

Se debe mostrar un menú donde se muestren varias opciones:

\* **Agregar cuenta bancaria.** Los datos a leer son los mismos que los de una transferencia de cualquier banco (Titular, cédula, tipo de cuenta, cédula, correo, etc). Si el número de Cuenta ya está registrado, entonces mostrar un diálogo donde indique que hay un error.

\* **Declarar inmueble.** El inmueble debe contener un atributo especial, que es el precio de compra. A partir de ese precio de compra, se deducirá un impuesto del 10%. Si un usuario tiene más de dos inmuebles, por cada inmueble nuevo, se debe agregar un impuesto especial del 25% de su precio declarado.

\* **Declarar automóvil.** El automóvil debe contener un atributo especial, que es el precio de compra. A partir de ese precio de compra, se deducirá un impuesto del 12%. Si un usuario tiene más de dos automóviles, por cada automóvil nuevo, se debe agregar un impuesto especial del 25% de su precio declarado.

Si el usuario es un administrador, se debe mostrar una ventana donde se muestre una lista de todos los usuarios contribuyentes registrados, lo declarado por cada uno de ellos, y el total recaudado de todos los usuarios de la plataforma.

Para cualquier tipo de usuario, debe haber una opción de salir del sistema, y en todo momento debe mostrarse el usuario actual y si es contribuyente o administrador (en la esquina inferior derecha).

### **Notas de ingeniería:**

Se usará el patrón de diseño MVC (Model-View-Controller) para construir la aplicación. Cada nueva interfaz representa las vistas, cada clase creada para manejar la información representa los modelos, y finalmente, las funciones definidas a partir de cada Widget agregado y de su acción representa las funciones del controlador.

A excepción de los diálogos de error, el programa debe correr en una única ventana. Para ello lo recomendado es usar QStackedWidget.

Se debe mostrar un diseño intuitivo para un usuario final. Se recomienda estudiar los distintos estilos de Qt y ver algunos ejemplos que se encuentran en la web.

Usen StackOverflow para buscar la solución a muchas de sus dudas. Usen git para repartirse el trabajo.