

SVIM

Módulos de desafío

Resumen

Este documento presenta la especificación de requerimientos de los módulos de desafíos del sistema SVIM. Para cada desafío se presentarán: los objetivos, los requerimientos funcionales, las especificaciones técnicas y los métodos de prueba en los casos que sea posible.

Objetivos generales

- Aplicar los conocimientos adquiridos durante el cursado a la ampliación del sistema desarrollado.
- Apreciar las ventajas de un desarrollo modular a la hora de agregar ampliaciones al sistema.
- Desarrollar la capacidad de investigar en forma independiente la existencia de bibliotecas o herramientas del lenguaje para la resolución de problemas.

Descripción de desafíos

1 Interfaz de usuario

1.1 *Objetivos específicos*

- Ampliar el conocimiento de la creación de interfaces con la librería Qt.

1.2 *Especificaciones funcionales*

El sistema deberá reemplazar la interfaz de consola por una interfaz de ventana, interactiva con el usuario. La interfaz deberá contener tres botones, la pantalla de graficación de la imagen y otros componentes visuales. Los botones deberán permitir: abrir una imagen, aplicar el filtro de color y guardar la imagen. La imagen deberá graficarse inmediatamente luego de que se carga, del mismo modo que luego que se aplica el filtro. Los atributos de la imagen deberán mostrarse en los otros componentes visuales.

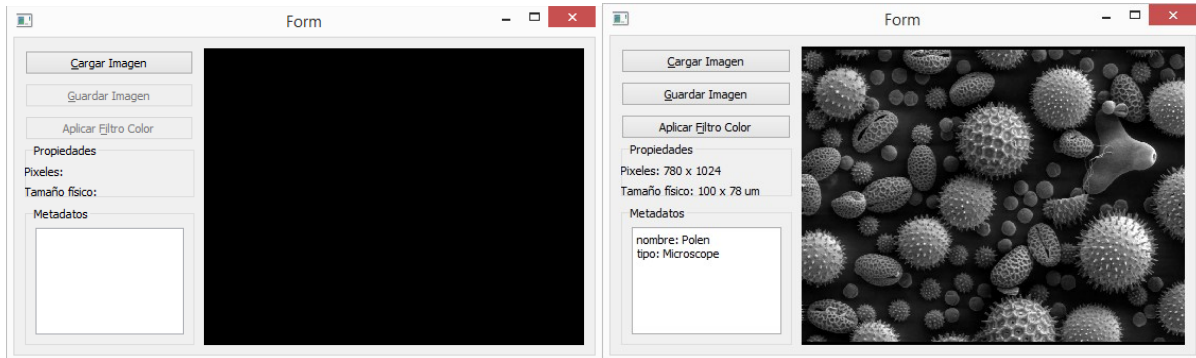
1.3 *Detalles técnicos*

Deberán tenerse en cuenta los casos excepcionales, como por ejemplo, cuando se trata de aplicar el filtro sin que esté la imagen cargada en memoria. En esos casos, esas funcionalidades deberán estar deshabilitadas. Para la carga y guardado de archivos se deberán utilizar cuadros de diálogos que permitan seleccionar un archivo navegando en el directorio. Se podrán agregar opcionalmente otras funcionalidades en la interfaz. En la página <http://doc.qt.io/qt-4.8/opensourceedition.html> se puede

iniciar el estudio para este desafío.

1.4 Método de prueba

Se podrá utilizar cualquiera de las imágenes disponibles para este desafío. En las siguientes imágenes se muestra un ejemplo de interfaz.



2 Propiedades de color de la imagen

2.1 Objetivos específicos

- Proveer información adicional de los colores de la imagen.

2.2 Especificaciones funcionales

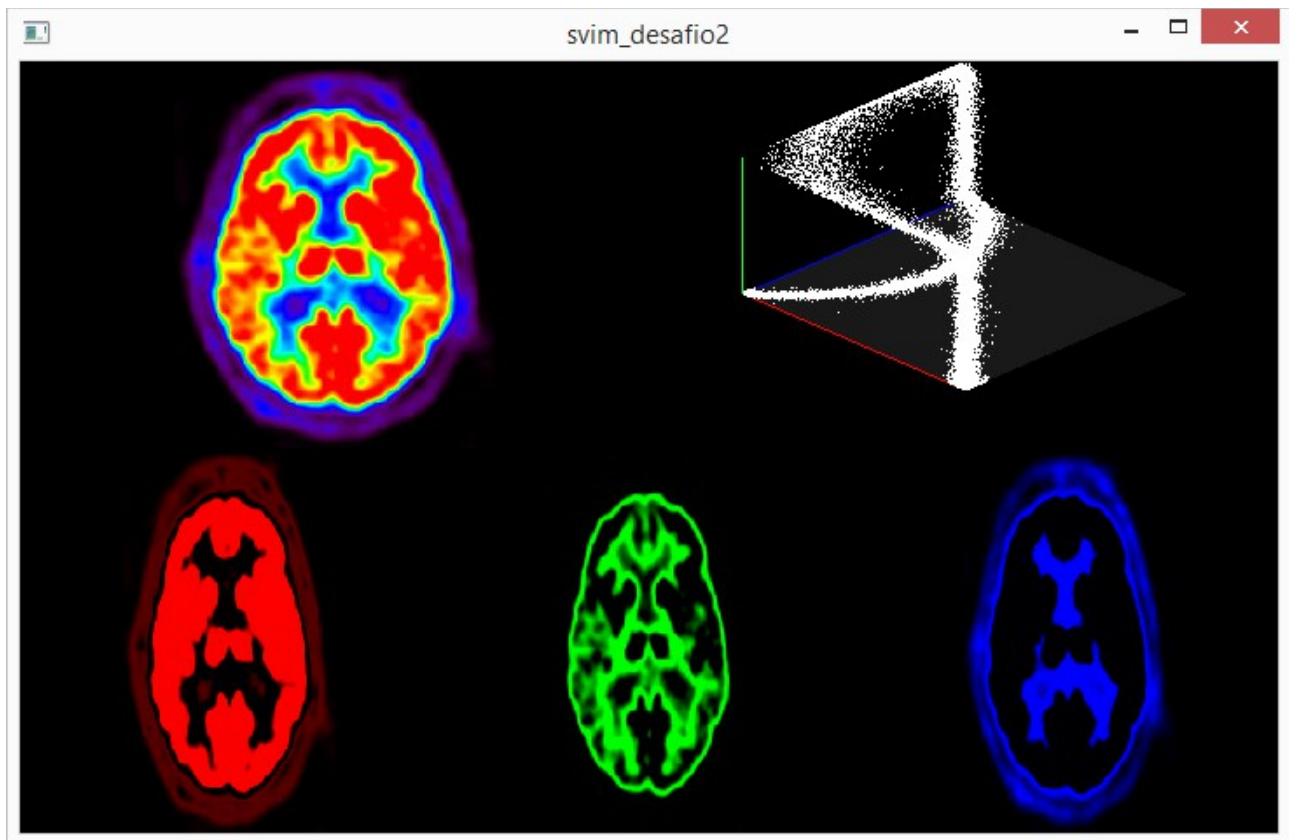
En simultáneo con el gráfico de la imagen, se deberán realizar un gráfico en 3D de la densidad de los colores de la imagen y 3 gráficos de los componentes de color de la imagen.

2.3 Detalles técnicos

Para el gráfico de la densidad de color, se utilizará un conjunto de puntos, cuyas coordenadas estarán dadas por la intensidad de las componentes de color. De esta manera, un pixel de color (1.0, 0.5, 0.2) tiene las coordenadas $x = 1.0$, $y = 0.5$, $z = 0.2$. Para graficar una imagen de una componente de color se deberán anular las otras 2 componentes. Es decir, para graficar la imagen de canal rojo, los colores de los pixeles sólo se formarán usando la componente roja.

2.4 Método de prueba

Se podrá utilizar cualquier imagen de las provistas. En la siguiente imagen se muestra una posible implementación con la imagen *pet01.mif*.



3 Múltiples imágenes por código

3.1 Objetivos específicos

- Dibujar varias imágenes que tengan un mismo metadato.

3.2 Especificaciones funcionales

El sistema deberá proveer la funcionalidad de graficar un conjunto de imágenes que cumplan con un criterio definido por el usuario. Por ejemplo, dibujar todas las imágenes que tengan el metadato con código "tipo" y valor "MRI".

3.3 Detalles técnicos

No se deberán dibujar todas las imágenes en una misma ventana, sino que se deberá generar una secuencia en la cual la imagen graficada va cambiando a un ritmo constante (puede ser cada 3 segundos), pasando por todas las imágenes del código elegido por el usuario. Para esta secuenciación se podrá utilizar un temporizador, como, por ejemplo, el provisto por la librería <QTimer>.

3.4 Método de prueba

Se podrá utilizar cualquier imagen de las provistas. Si se selecciona el tipo MRI se deberá ver 3 imágenes; hay 2 del tipo Ecografía; 4 tipo X-Ray.

4 Extensión de funcionalidades

4.1 Objetivos específicos

- Ampliar las capacidades de procesamiento y almacenamiento de las imágenes.

4.2 Especificaciones funcionales

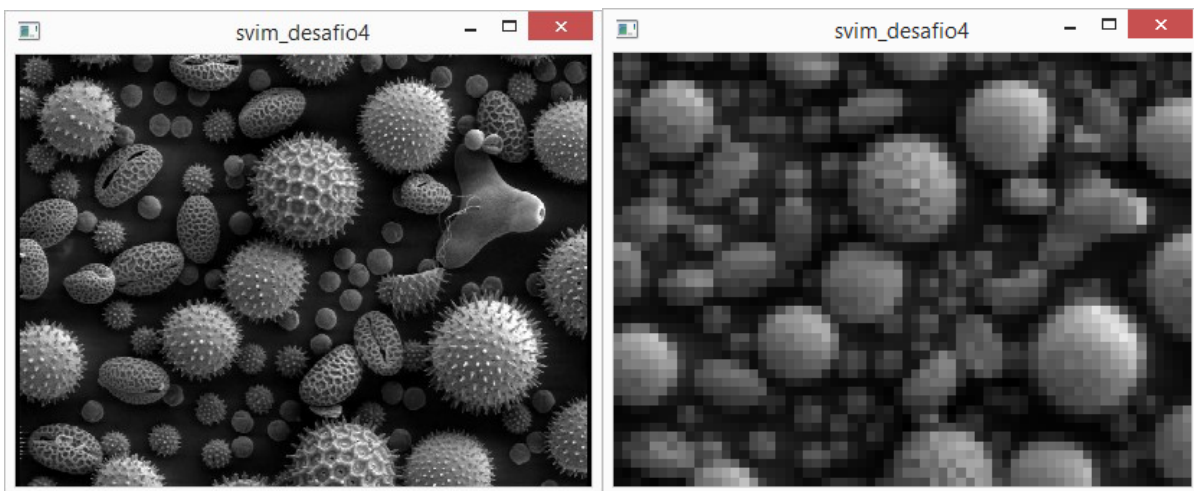
El sistema se deberá extender proveyendo las siguientes funcionalidades: un nuevo procesamiento: Pixelado; un nuevo formato de archivo: PPM; el usuario deberá poder revertir los procesamientos aplicados a una imagen (se deberán conservar en memoria las imágenes previas al procesamiento).

4.3 Detalles técnicos

El pixelado genera una imagen de la mitad de resolución (mitad de filas y mitad de columnas), de esta manera cada pixel de la imagen procesada ocupará físicamente el lugar de 4 pixeles de la imagen original. El formato de archivo PPM está descrito en detalle en la siguiente página: <http://netpbm.sourceforge.net/doc/#formats>. El programa tendrá que detectar la extensión del archivo para saber si trabajará con archivos MIF o PPM.

4.4 Método de prueba

Se podrá utilizar cualquier imagen de las provistas. En las siguientes imágenes se muestra el efecto del pixelado aplicado 4 veces sobre la misma imagen.



5 Autenticación de usuario

5.1 Objetivos específicos

- Agregar gestión de usuarios al sistema.

5.2 Especificaciones funcionales

El sistema deberá contar con un módulo de gestión de usuarios, mediante el cual el usuario de la aplicación deberá iniciar sesión con una contraseña antes de poder

utilizar la aplicación.

Deberá existir un usuario “admin” el cual será el encargado de gestionar los usuarios. Cuando se inicie sesión con este usuario la interfaz sólo mostrará las funcionalidades para agregar, eliminar, modificar o listar los usuarios.

5.3 Detalles técnicos

Se deberá tener en cuenta que las contraseñas deberán contar con cierta complejidad: deberán tener al menos 8 caracteres, al menos una minúscula, al menos una mayúscula y al menos un número.

Las contraseñas no deberán verse en la interfaz mientras son tipeadas, por lo cual se deberá implementar algún mecanismo que oculte esta información.

Se deberá contemplar que el almacenamiento de estas claves deberá estar encriptado de alguna forma para que nadie pueda accederlas.

Opcionalmente, se podrá utilizar una interfaz de ventanas para el inicio de sesión. Esta ventana deberá contener las casillas para ingresar el usuario y la clave, y botones para iniciar sesión o salir del programa. Se podrá iniciar el estudio de las interfaces en la página <http://doc.qt.io/qt-4.8/opensourceedition.html>.

5.4 Método de prueba

Se deberá probar el inicio de sesión con el usuario admin y todas las funcionalidades de gestión de usuarios. Luego se deberá probar el inicio de sesión de un usuario diferente.