# Polikiosco

# Manual técnico

# Índice

## 1. Que del software.

El kiosco interactivo para el Instituto Politécnico Nacional es un proyecto realizado en base a los conocimientos de ingeniería adquiridos durante la carrera, por lo tanto, se ha culminado satisfactoriamente. El kiosco interactivo es un centro de información interactivo que contiene la característica de ser una tecnología de información que a su vez representa la tecnología e interacción más actual que en el mundo globalizado se maneja con un usuario.

Su funcionamiento se basa en la programación de una interfaz interactiva el cual permite que un usuario tenga la posibilidad de interactuar con dicha interfaz de manera física, por medio de una computadora Touch, dicha interfaz tiene como propósito tener la interacción con la comunidad IPN para que se puedan tener información relevante de acuerdo a los eventos de la escuela y tengan la información de manera rápida, concreta y de primera mano.

La información que se mostrará del kiosco interactivo se obtiene por medio de los departamentos encargados de difundir información por medio de flayers, trípticos, hojas pegadas en paredes o vidrios, etc. Dichos departamentos proporcionan información de manera digital, es decir, por medio de imágenes, para así mostrar en el kiosco los eventos y conferencias que en dicha semana se llevaran a cabo.

## 2. Para que del software

Los kioscos interactivos, informáticos o multimedia son una unidad de comunicación informática y transaccional nacen como un medio de acercamiento entre la tecnología, la información, los usuarios y clientes de las organizaciones, adicionalmente responden a procesos de mejoramiento de servicio y automatización de procesos operacionales.

Se propone un kiosco digital interactivo que se planea construir en las instalaciones del Instituto Politécnico Nacional el cual será de gran importancia en el desarrollo tecnológico de esta casa de estudios, es de vital importancia ya que en un mundo globalizado por la tecnología es de mayor aceptación, que los individuos, se sientan más familiarizados con la consulta de información por medios tecnológicos que por los medios convencionales ya conocidos.

## 3. Antecedentes

Las computadoras y las redes están presentes en cada faceta de la vida moderna. Somos altamente dependientes de estas tecnologías para comunicación, transferencia de fondos, administración de utilidades, servicios gubernamentales, acciones militares, y mantenimiento de información confidencial. Utilizamos la tecnología para proporcionar energía, suministro de agua, servicios de emergencia, sistemas de defensa, banca electrónica y servicios de salud pública.

Al mismo tiempo, esta tecnología se está utilizando para llevar a cabo acciones ilegales y maliciosas, tales como el robo de información con fines de lucro, uso fraudulento de sistemas telefónicos, trasmisión ilegal de secretos comerciales y propiedad intelectual, modificación no autorizada de sitios web por razones políticas, interrupción de comunicaciones, revelación de secretos críticos y estrategias nacionales e incluso terrorismo.

De acuerdo con Beatriz Escobedo De La Cruz en su artículo KIOSCOS VIRTUALES EDUCATIVOS: ALTERNATIVA DE INCLUSIÓN DIGITAL/ EDUCATIONAL VIRTUAL KIOSKS: ALTERNATIVE TO DIGITAL INCLUSION de la Revista Iberoamericana de Producción Académica y Gestión Educativa ISSN (2014) menciona que un Kiosco es una terminal de computadora con hardware y software especializado diseñado dentro de una exposición pública que da acceso a la información y aplicaciones para la comunicación, el comercio, el entretenimiento y la educación.

El principal motivo por el cual Beatriz Escobedo realizo esta propuesta es introducir la enseñanza de la ciencia desde las primeras etapas de la educación y facilitar el acceso de la información en las comunidades rurales. Y algunos de los resultados que ha logrado tener este proyecto es que se ha incrementado la inclusión de las comunidades rurales a los avances tecnológicos con el fin de alfabetizar a las comunidades.

La empresa Alveni S.A de C.V desarrolla kioscos digitales enfocados al mercado industrial, diseñando y desarrollando software bilingüe la interfaz gráfica en la plataforma WinPOS Ready 7 enfocado a la petición de sus clientes empresas con la ingeniería, diseño industrial y la administración de sus proyectos.

Como un caso de éxito, Alveni S.A de C.V. realizó una solución de kioscos para la entrega de recibos en instalaciones de procesamiento de metal en la empresa Deacero, una empresa siderúrgica líder con patios de chatarra en México, donde cada camión se pesa al arribo y salida de las instalaciones a fin de determinar la cantidad de metal entregada. Las operadoras de báscula entregan un recibo impreso a cada chofer, que sirve como confirmación de entrega y recepción de venta para el proveedor. Este proceso tarda entre 8 y 10 minutos por camión, provocando largas filas que ocasionan obstrucción en las calles aledañas.

Como solución Alveni creó una solución completa con el kiosco, el cual se comunica directamente con el sistema llamado ERP (entrega, recibo y pago) de Deacero en donde automatiza la entrega de los recibos a cada chofer que descarga metal en sus patios y, otorgando una solución de kiosco hecho a la medida, el proceso tarda sólo 2 a 3 minutos por camión, permitiendo que cada patio de chatarra pueda duplicar su capacidad a más de 250 camiones por día.

También disminuye el tiempo de procesamiento en más de 70%. Ofreciendo resultados satisfactorios en el tiempo de procesamiento para imprimir recibos se redujo más de 70%, de 4-5 minutos a segundos, cada patio aumentó al doble su

capacidad al poder procesar más de 250 camiones por día, el tiempo de espera de cada camión se redujo en más de un 50%.

Como se puede observar el uso de estas plataformas tecnológicas pueden ser de gran ayuda no solo para la difusión de información si no que pueden ser empleadas para el desarrollo de tareas lo cual facilite la realización de actividades para un fin específico, como el caso ya descrito por la empresa Alveni.

En el trabajo Kiosco Informativo Interactivo (KINFO – UTP) realizado por, Ing. Nadia Lee, Ing. Gregorio Santana, Lic. Anthony Martínez, Lic. Norman Rangel de la Universidad Tecnológica de Panamá – CIDITIC que consiste en la instalación de terminales informativas interactivas en cada uno de los edificios del área metropolitana de la Universidad Tecnológica de Panamá y que en su primera etapa tiene como función principal brindar información detallada sobre las instalaciones físicas.

Mencionan que los avances tecnológicos y el interés del público por obtener información inmediata y oportuna han impulsado la necesidad de responder a sus necesidades esenciales al momento de asistir a un lugar, centro o institución a través de mecanismos tecnológicos atractivos, prácticos y por su connotación de actualidad.

Y que como resultados han logrado que su plataforma interactiva sea un gran apoyo o guía informativa, interactiva e innovadora para el Campus Universitario “Dr.

Víctor Levi Sasso” y que, del mismo, se deriven nuevos proyectos de igual o mayor magnitud en cuanto a tecnologías actuales se refiere.

El turismo adoptó originalmente a los sistemas globalizadores (GDS) como las principales herramientas tecnológicas para la promoción de servicios turísticos, los cuales permitían la concentración de la información de múltiples destinos en un sólo sistema centralizado. Una aplicación recurrente de esta tecnología se presenta en una hotelera, ya que forma parte de los servicios que se ofrecen en las habitaciones y le permiten al viajero tener acceso a Internet sin necesidad de una computadora, ni tener que desplazarse al centro de negocios o a algún kiosco.

## 4. Cualidades

Funciona como un medio innovador y de acercamiento entre la tecnología, la información, los usuarios, inclusive para los clientes de las organizaciones, adicionalmente responden a procesos de mejoramiento de servicio y automatización de servicios en específico.

Es capaz de captar la atención de la comunidad IPN mediante su estructura fisica, esto se puede lograr por medio de una implementación en sus

instalaciones, con el objetivo de lograr un incremento en el interés de la comunidad politécnica por la información, eventos organizados, y avisos en general promovidos por el Politécnico.

Representar un beneficio para la misma comunidad, ya que al tener un mejor medio de información, los eventos se pueden clasificar u organizar de acuerdo a las necesidades de los docentes, alumnos y directivos, así como cada uno los departamentos del Politécnico como lo son, difusión cultural, el área deportiva, entre otros más

Las nuevas generaciones se identifican con las nuevas tecnologías, como lo es el kiosco ya que se vive dentro de una época en la cual la revolución tecnología está avanzando a pasos agigantados. En resumen, es Interactiva para los usuarios con una fácil manipulación

# 5. Ventajas

Toda la información se actualizará de manera remota, de tal manera que un administrador pueda actualizar el kiosco y entonces constantemente el kiosco interactivo muestre la información, para que la comunidad politécnica se mantenga informada durante todo su periodo escolar.

Por medio de Ubuntu, el cual es de las distribuciones más estables, eficientes y fáciles de manejar debido a que es un software libre, esto permite poder realizar diversas tareas de manera más fácil. Otra de las ventajas que se encuentra con este sistema operativo es que es bastante seguro ya que es casi nula la existencia de archivos maliciosos como virus lo que, si se encuentra en gran manera en Windows, estas razones y otras hacen que la distribución Ubuntu junto con sus herramientas sean las más ideales para la realización de un trabajo de esta índole.

# 6. Herramientas y programas aplicados

## 6.1 Microsoft Windows

Desde Windows XP, es un sistema operativo multitarea de 32/64 bits para microprocesadores como AMD, Intel y sus diferentes clasificaciones. El sistema operativo Windows cuenta con una gran cantidad de herramientas incluidas de manera implícita que facilitan en gran cantidad las cuestiones de conexión a internet y de visualización, se pueden escoger diferentes lenguajes de programación para hacer aplicaciones con un gran potencial, tanto lógico como visual.

Prácticamente todos los procesadores que soporta este sistema operativo cumplen con los requisitos mínimos para la realización del proyecto, Windows 10 ya tiene todas las características necesarias para interactuar con pantallas táctiles lo cual es una gran ventaja para aprovechar recursos y tiempo en otras áreas del proyecto, por estos motivos hemos elegido el sistema operativo

Windows 10 para la conexión remota para la actualización de la información del kiosco.

De tal forma que Microsoft Windows es el sistema operativo más conveniente para la computadora que se encargara de administrar el kiosco debido a su amplia gama de herramientas para el diseño de interfaces para un administrador que pueda manejar de una forma más sencilla el sistema. Desde siempre Windows ha dado gran énfasis en tener una interfaz agradable e intuitiva.

Los objetivos de diseño que cumple Microsoft Windows son la seguridad, compatibilidad con aplicaciones de Windows, ampliabilidad, es decir, su capacidad del sistema para mantenerse actualizado conforme a los avances tecnológicos informáticos, el soporte internacional, ya que proporciona soporte para diferentes configuraciones nacionales mediante la API de soporte de idioma nacional, etc.

## 6.2 GNU/Linux

Es un sistema operativo que se ha vuelto uno de los más importantes en uso hoy en día, porque– trae a la PC todo el poder y la flexibilidad de las estaciones de trabajo Unix, además de un conjunto completo de aplicaciones de Internet y una interfaz de escritorio totalmente funcional. Las distribuciones de Linux incluyen características que se han vuelto un estándar, como los escritorios; la compatibilidad de Unix; los servidores de red; y varias aplicaciones de software como aplicaciones de oficina, multimedia e Internet.

Linux también es un sistema operativo Unix totalmente funcional. Tiene todas las características estándar de un sistema Unix poderoso, incluido un conjunto completo de shell de Unix como BASH, TCSH y Z. Es un sistema operativo rápido y estable de fuente abierta para computadoras personales (PC) y estaciones de trabajo; ofrece servicios de Internet a nivel profesional, herramientas de desarrollo extensas, interfaces gráficas de usuario (GUIs) completamente funcionales y gran cantidad de aplicaciones que van desde suites para oficina, hasta aplicaciones multimedia.

Linux fue desarrollado a principios de la década de 1990 por Linus Torvalds, junto con programadores de todo el mundo. Como sistema operativo, Linux realiza muchas funciones de Unix, Macintosh, Windows y Windows NT. Sin embargo, se distingue por su poder y flexibilidad, además de su disponibilidad gratuita. La mayor parte de los sistemas operativos de PC, como Windows, empezaron su desarrollo en los confines de PCs pequeñas y restringidas, que sólo recientemente se han vuelto máquinas más versátiles.

## 6.3 Ubuntu

Ubuntu es una distribución de GNU/Linux y que se distribuye como software libre, el cual incluye su propio entorno de escritorio denominado Unity. Su nombre

proviene de la ética homónima, en la que se habla de la existencia de uno mismo como cooperación de los demás.

Está orientado al usuario promedio, con un fuerte enfoque en la facilidad de uso y en mejorar la experiencia del usuario. Está compuesto de múltiple software normalmente distribuido bajo una licencia libre o de código abierto. Estadísticas web sugieren que la cuota de mercado de Ubuntu dentro de las distribuciones Linux es, aproximadamente, del 49 %, y con una tendencia a aumentar como servidor web.

## 6.4 Lenguajes de programación

Dentro de la experiencia en la programación, consiste en una serie de instrucciones ordenada, es decir, es una programación estructurada la cual lleva un orden en bloques estructurados (líneas de código estructurado). Aunque el término de bloques estructurados no se aplica estrictamente a C o C++ (por ejemplo), se denomina comúnmente simplemente como un lenguaje estructurado. En realidad, dentro de cada uno de los lenguajes de programación como, C, C++, Java, Android, tienen muchas similitudes unos con otros lenguajes estructurados.

Para la programación, el procesador de la computadora tiene un papel importante, pues el procesador debe ser capaz de interpretar el algoritmo empleado por medio de las instrucciones de la programación estructurada, lo que significa comprender las instrucciones de cada paso y realizar las operaciones correspondientes. Los lenguajes utilizados para escribir programas de computadoras son los lenguajes de programación y programadores son los escritores y diseñadores de programas. El proceso de traducir un algoritmo en pseudocódigo a un lenguaje de programación se denomina codificación, y el algoritmo escrito en un lenguaje de programación se denomina código fuente.

En la realidad la computadora no entiende directamente los lenguajes de programación, sino que se requiere un programa que traduzca el código fuente a otro lenguaje que sí entiende la máquina directamente, pero muy complejo para las personas; este lenguaje se conoce como lenguaje máquina y el código correspondiente código máquina. Los programas que traducen el código fuente escrito en un lenguaje de programación, tal como C++, a código máquina, el cual se denominan como traductor.

### 6.4.1 C++

Le lenguaje C++ se remonta desde el lenguaje de programación C. Después de contemplar algunos lenguajes de programación para el diseño del proyecto, consideramos principalmente los lenguajes C#, Java, C, C++.

El lenguaje C++ representa el resultado de los esfuerzos realizados para proporcionar las ventajas de la programación Orientada a Objetos a un lenguaje clásico, muy extendido en la programación de sistemas. Se trata de una

extensión del lenguaje C que presenta claras influencias del lenguaje Simula, que a su vez puede considerarse como el precursor de los lenguajes POO.

El lenguaje C++ entiende por clase un tipo de dato definido por el programador que contiene toda la información necesaria para construir un objeto de dicho tipo y el conjunto de operaciones que permiten manejarlo (métodos).Un objetivo que oferta C++ es ayudar a escribir programas POO tan pronto y tan fácil como sea posible, por un lado, muchas características de C++ se han heredado de C, de tal forma que, aunque la estructura global de un programa pueda ser POO, son necesarios los conocimientos profundos de la estructura que utiliza C. Prácticamente C++ corre en todos los sistemas operativos, desde Windows (principalmente), Unix, Linux, Mac.

### 6.4.2 SDL2

Las funciones de SDL permiten desde simplemente pintar un pixel hasta renderizar imágenes, gestión de efectos de sonido y música. Una de sus grandes virtudes es el tratarse de una biblioteca multiplataforma, siendo compatible oficialmente con los sistemas Microsoft Windows, GNU/Linux, Mac OS y QNX, además de otras arquitecturas y sistemas como Sega Dreamcast, GP32, GP2X, etc.

Por estas razones hemos escogido las librerías SDL y algunos de sus complementos como SDL_Image o SDL_sound que proporciona herramientas para programar aplicaciones multimedia.

## 6.5 FTP (PROTOCOLO DE TRANSFERENCIA DE ARCHIVOS)

Es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmisión Control Protocolo), se basa en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar o enviar archivos, independientemente del sistema operativo utilizado en cada equipo.

El Servicio FTP es ofrecido por la capa de Aplicación del modelo de capas de red TCP/IP al usuario, utilizando el puerto lógico de red 20 para transferencia de datos y el puerto lógico 21 para el control de datos. Un problema básico de FTP es que está pensado para ofrecer la máxima velocidad en la conexión, pero no la máxima seguridad, ya que todo el intercambio de información, desde el login y password del usuario en el servidor hasta la transferencia de cualquier archivo, se realiza en texto claro sin ningún tipo de cifrado, con lo que un posible atacante puede capturar este tráfico, acceder al servidor, o apropiarse de los archivos transferidos. Una ventaja que nos ofrece ftp es la conexión de forma remota, en cualquier parte del mundo se pueda accesar al servidor.

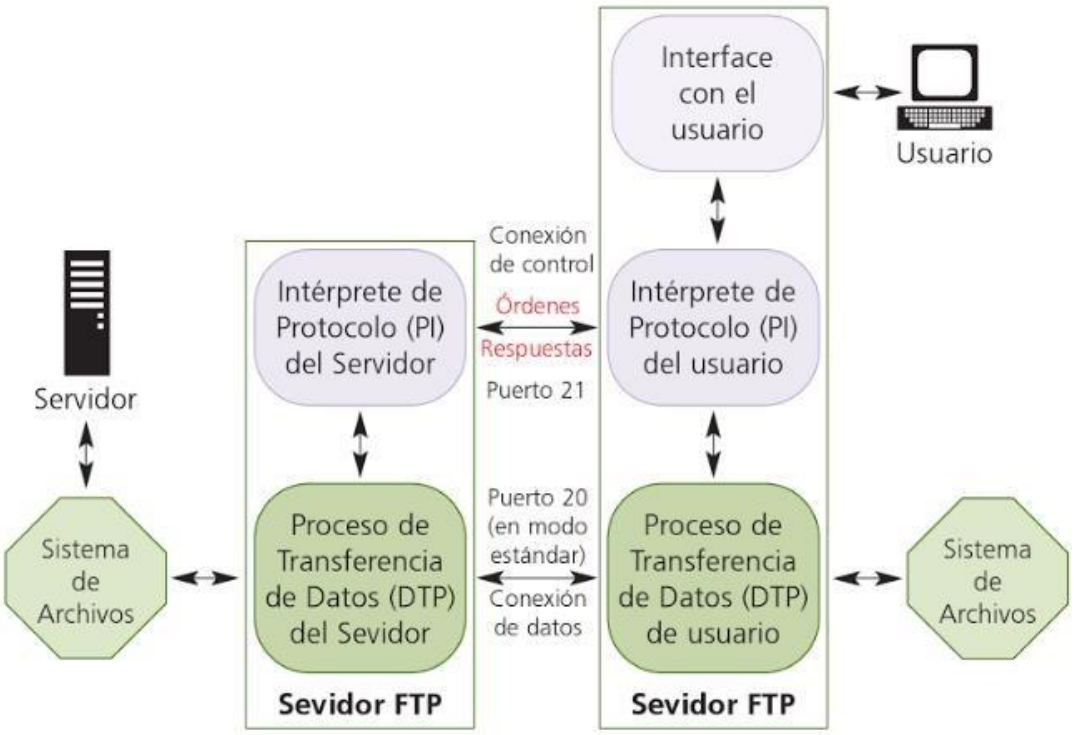



Gráfico 1. Esquema del servidor FTP

FTP es un servicio que permite transferir ficheros entre cliente-servidor de manera bidireccional, permitiendo una serie de acciones para la administración de los ficheros como lo es subir, bajar, borrar, renombrar, mover, crear carpetas, borrar carpetas.

En el modelo, el intérprete de protocolo (PI) de usuario, inicia la conexión de control en el puerto 21. Las órdenes FTP estándar las genera el PI de usuario y se transmiten al proceso servidor a través de la conexión de control. Las respuestas estándar se envían desde el PI del servidor al PI de usuario por la conexión de control como respuesta a las órdenes.

## 6.6 Técnicas de compilación

### 6.6.1 Compiladores

El proceso de traducción de un programa fuente escrito en un lenguaje de alto nivel a un lenguaje máquina comprensible por la computadora, se realiza mediante programas llamados “traductores”. Los traductores de lenguaje son programas que traducen a su vez los programas fuente escritos en lenguajes de alto nivel a código máquina. Los traductores se dividen en compiladores e intérpretes.

### 6.6.2 Intérpretes

Un intérprete es un traductor el cual toma un programa fuente, lo traduce y, posteriormente lo ejecuta. Los programas intérpretes clásicos como Visual Basic, se utilizan para circunstancias especiales, como el diseño de una aplicación con base de datos, aplicaciones con ayuda de ventanas.

El sistema de traducción tiene como consistencia, traducir la primera sentencia del programa a lenguaje máquina, se detiene la traducción, se ejecuta la sentencia; a continuación, se traduce la siguiente sentencia, se detiene la traducción, se ejecuta la sentencia y así sucesivamente hasta terminar el programa. Algunos de los compiladores más usados son C, C++, C#, Java, Android, entre otros.

## 6.7 Herramientas usadas para hacer el software.

Un compilador es un programa que traduce los programas fuente escritos en lenguaje de alto nivel a lenguaje máquina. La traducción del programa completo se realiza en una sola operación denominada compilación del programa; es decir, se traducen todas las instrucciones del programa en un solo bloque. El programa compilado y depurado (eliminados los errores del código fuente).

### 6.7.1 Compilador GCC

De acuerdo al artículo de Víctor Gonzales(vaginbar, 2016), GCC es un compilador integrado del proyecto GNU para C, C++, Objective C y Fortran; es capaz de recibir un programa fuente en cualquiera de estos lenguajes y generar un programa ejecutable binario en el lenguaje de la máquina donde ha de correr. La sigla GCC significa "GNU Compiler Collection". Originalmente significaba "GNU C Compiler"; todavía se usa GCC para designar una compilación en C. G++ refiere a una compilación en C++.

Sintaxis.

gcc [ opción | archivo ] ...

g++[ opción | archivo ] ...

Las opciones van precedidas de un guion, como es habitual en UNIX, pero las opciones en sí pueden tener varias letras; no pueden agruparse varias opciones tras un mismo guion. Algunas opciones requieren después un nombre de archivo o directorio, otras no. Finalmente, pueden darse varios nombres de archivo a incluir en el proceso de compilación.

### 6.7.2 Sublime Text 3

Sublime Text 3 para Ubuntu es un gran editor de código ( aun que no la única opción válida ) y texto con un largo tiempo de vida. Según dice la Wikipedia:

“**Está escrito en C++ y sus plugins se desarrollan en Python**. Inicialmente fue desarrollado como una extensión de Vim”.

Su primer lanzamiento fue en 2008 y desde entonces no ha parado de mejorar, en parte gracias a sus plugins. Con ellos conseguirás convertir un gran editor en una herramienta magnífica.

- El editor **Sublime Text permite un resaltado de sintaxis muy rico** proporcionando así un mejor rendimiento a la hora de producir nuestros códigos.
- Package control: con la función control de paquetes (Ctrl + Mayús + P), **se pueden instalar funciones para nuestro editor** como: como ordenar, cambiar la sintaxis y cambiar la configuración de la sangría de nuestro código. Además de multitud de plugins para cada una de las necesidades que puedan surgir durante su uso.
- Adaptable: el texto generado con **Sublime Text 3 es altamente adaptable**. Nos proporciona bindings, menús, snippets, macros, etc. Hay que destacar que **casi todo en Sublime Text 3 es personalizable mediante archivos JSON**.
- Multiplataforma: Sublime Text está disponible para OS X, Windows y Linux. Una licencia es todo lo que se necesita para usar Sublime Text en cada equipo que poseas, sin importar el sistema operativo sobre el que se utilice. En caso de no tener la licencia solo se mostrará un mensaje de cuando en cuando para recordarte que tu editor está funcionando sin ella.
- Soporte nativo para multitud de idiomas: **Este editor soporta de forma nativa 43 lenguajes de programación** y texto plano. Por si fuese poco, puedes añadir más mediante sus plugins.
- Kit de herramientas de interfaz de usuario personalizable: Sublime Text utiliza un conjunto de herramientas de interfaz de usuario altamente personalizables. **Este interface está optimizado para ser veloz**. Aprovecha la funcionalidad nativa de cada plataforma.
- Modo “libre de distracciones”: Se puede ejecutar el “Modo libre de distracciones” usando el menú “View / **Distraction free mode**” cuando necesites estar concentrado exclusivamente en el código que estas generando.

- Han añadido nuevas definiciones de sintaxis de C ++, JavaScript y Rust con mayor precisión y rendimiento. Además de muchas otras mejoras de resaltado de sintaxis.
- Se ha mejorado el rendimiento de renderizado en el OSX. Especialmente en pantallas de alta resolución.
- Ha mejorado del comportamiento de la corrección ortográfica.
- Se ha mejorado el comportamiento durante la indexación de archivos con varias ventanas abiertas.
- Se agregó un motor de regex personalizado que coincide con varias expresiones regulares en paralelo. Con esto han conseguido que la carga e indexación de archivos sea más rápida.
- Mejora del soporte de Unicode.

- Incorpora muchas mejoras proporcionadas por la comunidad a los paquetes anteriores, con mejoras significativas en HTML, CSS, JavaScript, Go, D y SQL.
- Añadido Panel Switcher a la barra de estado.
- Mejor manejo de archivos problemáticos durante la indexación.
- Detección mejorada del cambio de archivos.
- Corregido el uso de CPU alto causado por un índice dañado. Esto le ocurría a algunos usuarios actualizando desde 3065.
- Añadidos iconos de la barra lateral.
- Se han añadido indicadores de carga en la barra lateral.
- La barra lateral recuerda qué carpetas que el usuario expande.
- La sincronización automática de citas también ha sido mejorada.

Fuente: https://ubunlog.com/instalar-sublime-text-3-ubuntu/

# 7. Módulos y algoritmos desarrollados

## 7.1 Herencia y Jerarquía de clases

La herencia se manifiesta con la creación de un tipo definido por el usuario (clase), que puede heredar las características de otra clase ya existente o derivar las suyas a otra nueva clase. Cuando se hereda, las clases derivadas reciben las características (estructuras de datos y funciones) de la clase original, a las que se pueden añadir nuevas características o modificar las características heredadas.

La derivación de las clases consigue la reutilización efectiva del código de la clase base para sus necesidades. C++ utiliza un sistema de herencia jerárquica. Es decir, se hereda una clase de otra, creando nuevas clases a partir de clases ya existentes.

Solo se pueden heredar clases, no funciones ordinarias ni variables. Una clase utilizada para derivar nuevas clases se denomina clase base (padre, superclase o ascendiente). Una clase creada de otra clase se denomina clase derivada o subclase. La terminología supone una clase base o clase padre, y una clase derivada o clase hija. Esta relación supone un orden de jerarquía simple. A su vez, una clase derivada puede ser utilizada como una clase base para derivar más clases. Por consiguiente, se puede construir jerarquías de clases, en las que cada clase sirve como padre o raíz de una nueva clase.

## 7.2 Diagrama de flujo del software

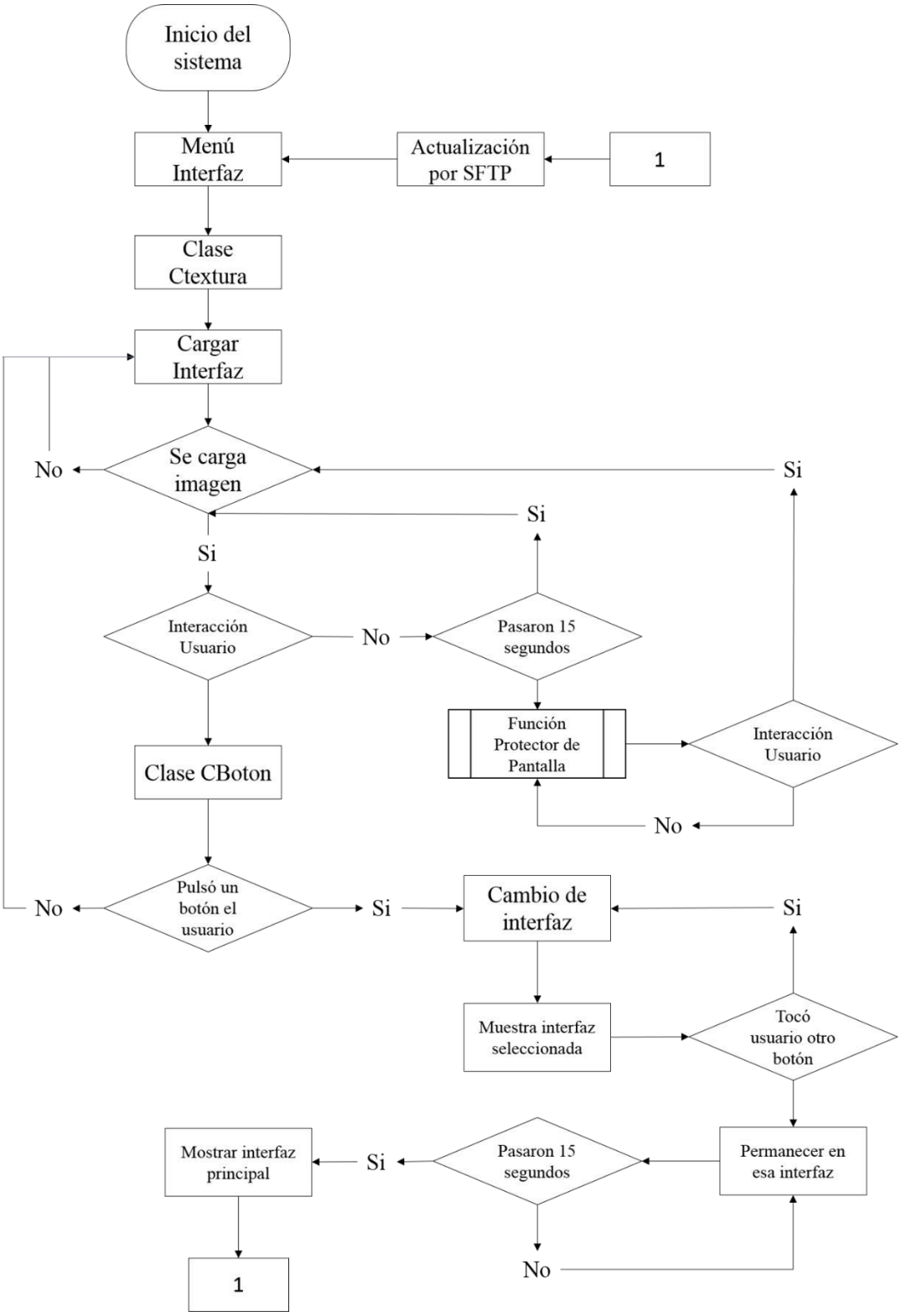
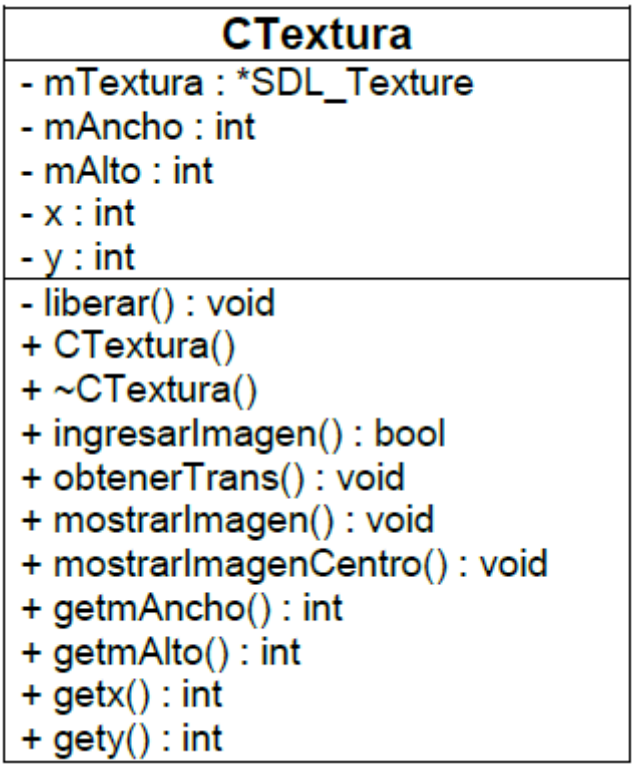



Diagrama de flujo 1 Funcionamiento del

## 7.3 Clase CTextura




Esta clase como su nombre lo indica muestra en pantalla alguna imagen que tengamos guardada en el disco duro de nuestra computadora únicamente usando la ruta e indicando el tamaño y la posición de la imagen. A continuación, se muestra el código fuente del archivo CTextura.h, el cual solo contiene las declaraciones de atributos y funciones miembro de la clase.

Se incluyen los archivos SDL2/SDL.h y SDL2/SDL_image.h, el primer archivo contiene las declaraciones de las clases de SDL y el segundo archivo contiene las declaraciones de algunas clases necesarias para trabajar con imágenes de cualquier formato, también se incluye el archivo string que contiene la declaración de la clase string el cual solo se usa para usar en el prototipo de la función ingresarImagen() ya que este necesita conocer la ruta de la imagen en forma de string.

En la sección privada de la clase se encuentra la declaración del apuntador de un objeto tipo SDL_Texture que es la que hace todo el proceso que requiere una textura, pero con esta clase se pretende facilitar su uso y agregar medidas para que el programe no colapse, por ejemplo, si la ruta de la imagen no es la correcta, usando solo la clase SDL_Texture, el programa su cuelga y en la mayoría de los

casos se tiene que reiniciar la computadora, con la clase CTexture cuando ocurre un error similar, envía un mensaje a la terminal de Linux indicando que la ruta es incorrecta o no existe y en caso de que sea necesario finaliza el programa de manera segura.

También hay dos datos de tipo entero que son para guardar el ancho y alto de la textura en pixeles, la función privada liberar() la usa principalmente el destructor de la clase que como su nombre lo indica, es para liberar la memoria ram de las imágenes que se han cargado, también pone en 0 todos los atributos de la clase. En la sección publica de la clase está el destructor y constructor, el destructor llama a la función privada liberar() y el constructor inicializa con 0 todos los atributos de la clase y en caso del apuntador de tipo SDL_Texture lo inicializa con NULL.

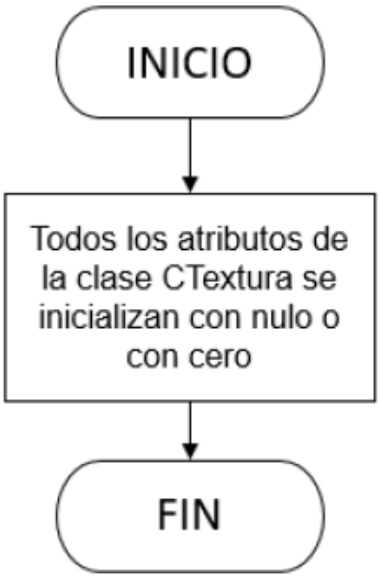



*Diagrama de flujo 1 Constructor de CTextura*

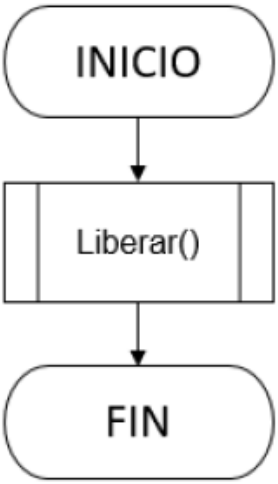



*Diagrama de flujo 2 Destructor CTextura.*

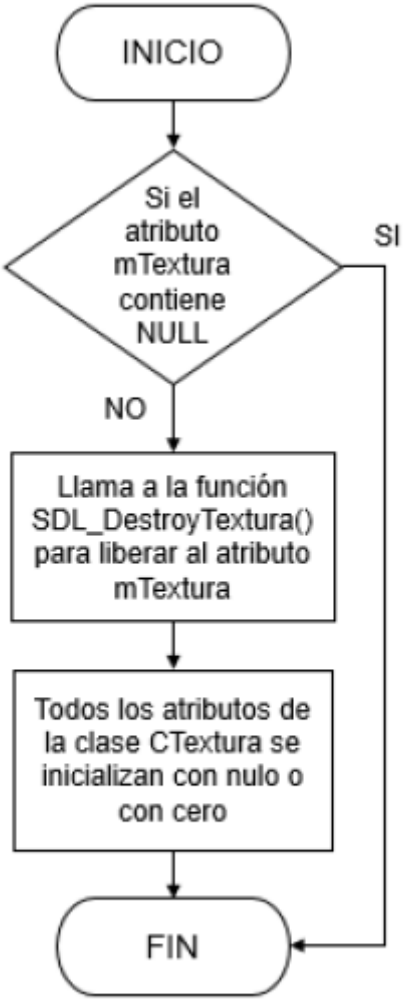
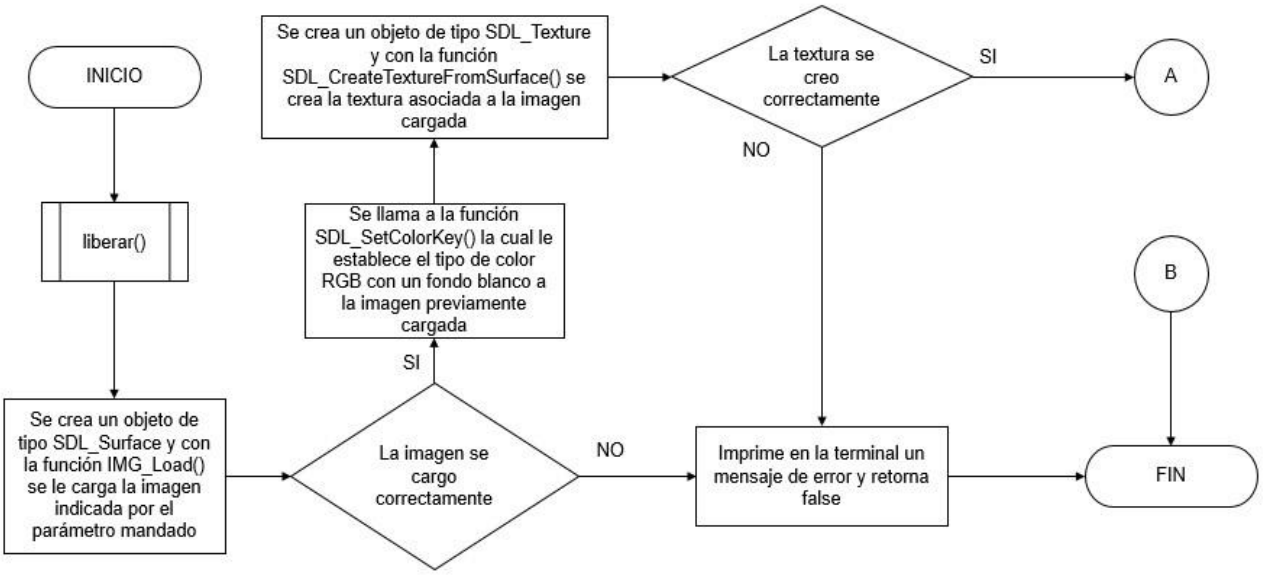



*Diagrama de flujo 3 CTextura::liberar()*

### 7.3.1 Definición de la función ingresarImagen()



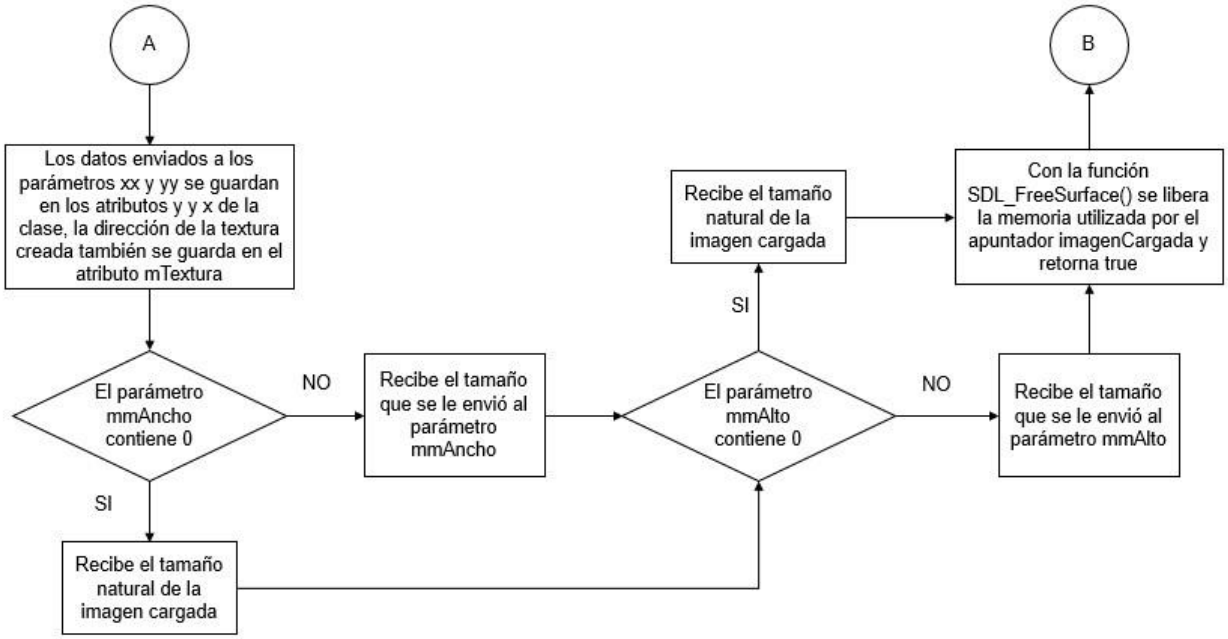




Diagrama de flujo 5

La función miembro ingresarImagen() sirve para cargar alguna imagen que esta guardada en el disco duro de la computadora, como se ve tiene dos parámetros, el primero es de tipo string el cual es para pasarle a la función la ruta de la imagen que se requiere cargar

(ej. “/home/admi/Documentos/Imagenes/imagen.png”) y el segundo parámetro es un apuntador de tipo SDL_Render, el cual es para obtener la dirección del objeto que se encargara de renderizar la imagen. Tiene como retorno un valor de tipo booleano, siendo False cuando hay un error y True cuando la imagen se cargó con éxito.

En la definición de la función como primer paso es liberar al objeto de tipo CTextura de cualquier otra imagen que pueda estar cargada llamando a la función privada liberar(), a continuación se crea e inicializa un apuntador de tipo SDL_Surface el cual es un objeto que representa a un área de memoria que se puede dibujar, como se puede apreciar para cargar la imagen al apuntador imagenCargada, se llama a la funcion IMG_Load() el cual recibe como argumento una cadena literaria que dice la ubicación de la imagen a cargar.

Muchas de las clases de SDL tienen un valor de retorno entero el cual si retorna False indica que hubo algún error al cargar la imagen, en la linea 31 hay una condicional if que en caso de que haya un error la función termina retornando False, si no hay ningún error el flujo de la función continua.

La función SLD_SetColorKey() establece la clave de color a un tipo de dato SDL_Surface, teniendo esto creamos un apuntador de tipo de dato SDL_Texture que servirá como variable auxiliar, le cargamos la superficie que hemos creado (imagen cargada) y la pasamos por una condicional que compruebe que se ha cargado correctamente.

Una vez realizado esto, pasamos todos los datos a los atributos del objeto mTextura recibe el contenido del apuntador auxiliar mTextura y mAncho y mAlto reciben el tamaño en pixeles de la imagen previamente cargada. Una vez que todo este hecho se libera el apuntador imagenCargada para evitar que la memoria ram se sature y finalmente la función miembro retorna el valor True indicando que la imagen se cargó correctamente al objeto CTextura.

### 7.3.2 Definición de la función obtenerTrans()

La mayoría de las veces es necesario que no se imprima el tamaño real de una imagen, por lo tanto es imprescindible tener una función con la que se pueda modificar el tamaño para adecuarlo al monitor o según las necesidades del proyecto.

```
48 void CTextura::obtenerTrans(int x, int y)
49 {
50     mAncho=x; mAlto=y;
51 }
```

### 7.3.3 Definición de la función mostrarImagen()

```
53 void CTextura::mostrarImagen(int x, int y, SDL_Renderer *Rendere, int mmAncho, int mmAlto)
54 {
55     if(mmAncho!=0 && mmAlto!=0)
56     {
57         SDL_Rect renderQuad = { x, y, mmAncho, mmAlto };
58         SDL_RenderCopy(Rendere, mTextura, NULL, &renderQuad);
59     }
60     else
61     {
62         SDL_Rect renderQuad = { x, y, mAncho, mAlto };
63         SDL_RenderCopy(Rendere, mTextura, NULL, &renderQuad);
64     }
```

La función pública mostrarImagen() permite imprimir una textura previamente cargada con las características más básicas posibles, el origen que representa la ubicación en donde estará la parte superior izquierda de la imagen está dado por x y y, el parámetro SDL_Renderer es necesario porque necesita un objeto con el cual la imagen será renderizada, en el caso de este proyecto solo hay un apuntador de tipo SDL_Renderer que servirá para todas las imágenes.

Si se desea, se puede modificar el tamaño de impresión de la imagen usando los parámetros mmAncho y mmAlto, como se puede apreciar al momento de modificar sus parámetros, logrando así permitir la personalización de sus ajustes, estos tienen un valor por default de 0.

Si la función es llamada solo con los primeros 3 parámetros, se imprime el tamaño real de la imagen (el tamaño real se guarda desde un principio con la función ingresarImagen() ) de lo contrario si mmAlto y mmAncho contienen alguna medida específica, se crea un cuadro de ese tamaño al cual se le copia la textura.

### 7.3.4 Definición de la función mostrarImagenCentro()

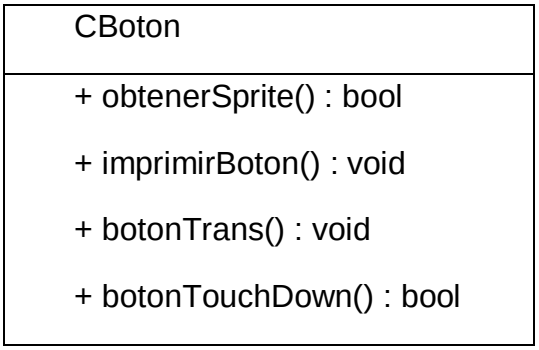
Es muy común que mostremos imágenes que estén en el centro del monitor, pero para ello se considera tanto el tamaño del monitor como el de la imagen, para esto hicimos esta función que realiza todos los cálculos necesarios para que la imagen este justo en el centro.

```
67 void CTextura::mostrarImagenCentro(SDL_Renderer *Rendere)
68 {
69     int x = 1366 / 2 - mAncho / 2;
70     int y = 768 / 2 - mAlto / 2;
71
72     SDL_Rect renderQuad = { x, y, mAncho, mAlto };
73     SDL_RenderCopy(Rendere, mTextura, NULL, &renderQuad);
74 }
```

Básicamente calcula el lugar en el que tendría que estar la esquina superior izquierda de la imagen para que este en el centro del monitor. Las funciones getmAncho() y getmAlto() retornan los valores de los atributos mAlto y mAncho.

## 7.4 Clase CBoton

Un botón en la interfaz gráfica es la textura de un botón al que se puede tocar y este a su vez envía un mensaje al sistema para que se ejecute algún proceso, la clase CTextura no puede realizar este proceso por si solo, por esto es que se define la clase CBoton. CTextura hereda todos sus atributos y funciones a CBoton, de esta manera se le asigna a CBoton la característica y función de interactuar con el usuario.




La herencia es de tipo “protected”, de esta forma la clase derivada podrá hacer uso de las funciones públicas de la clase base CTextura, sin embargo, esas funciones no podrán ser utilizadas fuera de la clase derivada. CBoton tiene como atributos privados dos variables de tipo entero, estas variables sirven para conocer la ubicación| del botón y de esta manera la función botonTouchDown() podrá saber si el usuario toco el botón o si el usuario tocó algún lugar fuera del botón dentro de la pantalla.

### 7.4.1 Definición de la función obtenerSprite()

```
10 bool CBoton::obtenerSprite(std::string imagen, SDL_Renderer *Rendere)
11 {
12   return ingresarImagen(imagen, Rendere);
13 }
```

La función obtenerSprite() llama a la función ingresarImagen() del objeto CTextura, cuya función ya se definió anteriormente en CTextura.cpp. La función obtenerSprite() sólo utiliza la función ingresarImagen() para redefinir el nombre de la función ya definida, debido a que se está tratando de otro objeto, ya que

nos permite una mejor practicidad para identificar los objetos, con el cual se está trabajando.

### 7.4.2 Definición de la función imprimirBoton()

```
15 void CBoton::imprimirBoton(int x, int y, SDL_Renderer *Rendere, int xx, int yy)
16 {
17   mostrarImagen(x, y, Rendere, xx, yy);
18   posicion_x = x;
19   posicion_y = y;
20 }
```

Dicha función tiene como prototipo cinco parámetros int x e int y, indican la posición en la cual el programador va posicionar el botón, SDL_Renderer es un tipo de dato el cual se utiliza para renderizar la imagen que se usará como botón, int xx e int yy ajustan el tamaño del botón de acuerdo a las necesidades del diseño, en caso de que no se mande ningún argumento, por default tomará el tamaño real de la imagen.

Su labor principal de ésta función es imprimir en la pantalla un botón, y posteriormente guarda la posición donde se ubica en el monitor dicho botón, dentro de la función imprimirBoton() se asignan los valores a las variables posición_x y posición_y las cuales se utilizan para guardar el conocer en todo momento la posición del botón.

### 7.4.3 Definición de la función botonTrans()

```
22 void CBoton::botonTrans(int x, int y, int xx, int yy)
23 {
24   posicion_x = x;
25   posicion_y = y;
26   obtenerTrans(xx,yy);
27
28 }
```

Esta función permite crear un botón transparente, es decir, sin imagen dentro de cualquier lugar de la pantalla. De manera similar que la función imprimirBoton(), contiene 4 parámetros los cuales ajustan el tamaño de la imagen transparente y asignan la posición de la imagen.

Aquí es obligatorio asignar un tamaño al botón transparente, dentro del llamado de la función, ya que en realidad no se toma alguna imagen de alguna carpeta o dirección.

### 7.4.4 Definición de la función botonTouchDown()

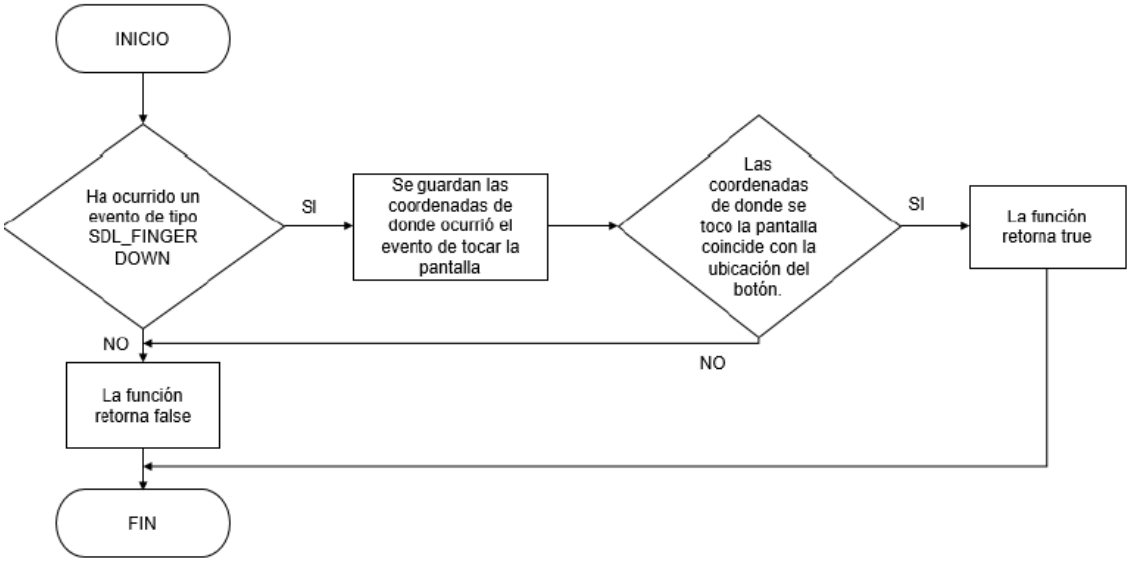



*Diagrama de flujo 5 CBoton::botonTouchDown()*

La función miembro ingresarImagen() sirve para cargar alguna imagen que esta guardada en el disco duro de la computadora, como se ve tiene dos parámetros, el primero es de tipo string el cual es para pasarle a la función la ruta de la imagen que se requiere cargar

(ej. “/home/admi/Documentos/Imagenes/imagen.png”) y el segundo parámetro es un apuntador de tipo SDL_Render, el cual es para obtener la dirección del objeto que se encargara de renderizar la imagen. Tiene como retorno un valor de tipo booleano, siendo False cuando hay un error y True cuando la imagen se cargó con éxito.

En la definición de la función como primer paso es liberar al objeto de tipo CTextura de cualquier otra imagen que pueda estar cargada llamando a la función privada liberar(), a continuación se crea e inicializa un apuntador de tipo SDL_Surface el cual es un objeto que representa a un área de memoria que se puede dibujar, como se puede apreciar para cargar la imagen al apuntador imagenCargada, se llama a la funcion IMG_Load() el cual recibe como argumento una cadena literaria que dice la ubicación de la imagen a cargar.

Muchas de las clases de SDL tienen un valor de retorno entero el cual si retorna False indica que hubo algún error al cargar la imagen, en la linea 31 hay una condicional if que en caso de que haya un error la función termina retornando False, si no hay ningún error el flujo de la función continua.

La función SLD_SetColorKey() establece la clave de color a un tipo de dato SDL_Surface, teniendo esto creamos un apuntador de tipo de dato SDL_Texture que servirá como variable auxiliar, le cargamos la superficie que hemos creado

(imagen cargada) y la pasamos por una condicional que compruebe que se ha cargado correctamente.

Una vez realizado esto, pasamos todos los datos a los atributos del objeto mTextura recibe el contenido del apuntador auxiliar mTextura y mAncho y mAlto reciben el tamaño en pixeles de la imagen previamente cargada. Una vez que todo este hecho se libera el apuntador imagenCargada para evitar que la memoria ram se sature y finalmente la función miembro retorna el valor True indicando que la imagen se cargó correctamente al objeto CTextura.

### 7.4.5 Definición de la función obtenerTrans()

La mayoría de las veces es necesario que no se imprima el tamaño real de una imagen, por lo tanto, es imprescindible tener una función con la que se pueda modificar el tamaño para adecuarlo al monitor o según las necesidades del proyecto.

```
48 void CTextura::obtenerTrans(int x, int y)
49 {
50     mAncho=x; mAlto=y;
51 }
```

### 7.4.6 Definición de la función mostrarImagen()

```
53 void CTextura::mostrarImagen(int x, int y, SDL_Renderer *Rendere, int mmAncho, int mmAlto)
54 {
55     if(mmAncho!=0 && mmAlto!=0)
56     {
57         SDL_Rect renderQuad = { x, y, mmAncho, mmAlto };
58         SDL_RenderCopy(Rendere, mTextura, NULL, &renderQuad);
59     }
60     else
61     {
62         SDL_Rect renderQuad = { x, y, mAncho, mAlto };
63         SDL_RenderCopy(Rendere, mTextura, NULL, &renderQuad);
64     }
```

La función pública mostrarImagen() permite imprimir una textura previamente cargada con las características más básicas posibles, el origen que representa la ubicación en donde estará la parte superior izquierda de la imagen está dado por x y y, el parámetro SDL_Renderer es necesario porque necesita un objeto con el cual la imagen será renderizada, en el caso de este proyecto solo hay un apuntador de tipo SDL_Renderer que servirá para todas las imágenes.

Si se desea, se puede modificar el tamaño de impresión de la imagen usando los parámetros mmAncho y mmAlto, como se puede apreciar al momento de modificar sus parámetros, logrando así permitir la personalización de sus ajustes, estos tienen un valor por default de 0.

Si la función es llamada solo con los primeros 3 parámetros, se imprime el tamaño real de la imagen (el tamaño real se guarda desde un principio con la función ingresarImagen() ) de lo contrario si mmAlto y mmAncho contienen alguna medida específica, se crea un cuadro de ese tamaño al cual se le copia la textura.

### 7.4.7 Definición de la función mostrarImagenCentro()

Es muy común que mostremos imágenes que estén en el centro del monitor, pero para ello se considera tanto el tamaño del monitor como el de la imagen, para esto hicimos esta función que realiza todos los cálculos necesarios para que la imagen este justo en el centro.

```
67 void CTextura::mostrarImagenCentro(SDL_Renderer *Rendere)
68 {
69     int x = 1366 / 2 - mAncho / 2;
70     int y = 768 / 2 - mAlto / 2;
71
72     SDL_Rect renderQuad = { x, y, mAncho, mAlto };
73     SDL_RenderCopy(Rendere, mTextura, NULL, &renderQuad);
74 }
```

Básicamente calcula el lugar en el que tendría que estar la esquina superior izquierda de la imagen para que este en el centro del monitor. Las funciones getmAncho() y getmAlto() retornan los valores de los atributos mAlto y mAncho.

## 7.5 Clase CInterfaz

La clase CInterfaz utiliza a las clases CBoton y CTextura, esta clase se creó con el propósito de hacer al archivo principal (Main.cpp) en donde se encuentra la función main() menos extensa ya que sin utilizar esta clase tendremos que hacer múltiples declaraciones de las clases CTextura y CBoton, sin esta clase prácticamente seria declarar un obtejo CTextura y un objeto CBoton por cada uno de estos que contenga la interfaz, y lo mismo sucede si queremos que se muestren en pantalla, tendríamos que llamar a la función encargada de mostrar en pantalla la textura de cada objeto. La única desventaja es que al declarar un Objeto de CInterfaz, automáticamente se crean 50 objetos CBoton y 50 CTextura.

| CInterfaz |
| --- |
| - botones[50] : *CBoton<br>- texturas[50] : *CTextura<br>- iBotones : int<br>- Texturas : int<br>- render : *SDL_Renderer |
| + CInterfaz()<br>+ ~CInterfaz()<br>+ cargarTextura() : bool<br>+ cargarBoton() : bool<br>+ asociarVentana() : bool<br>+ mostrar() : int<br>+ touchBoton() : int |

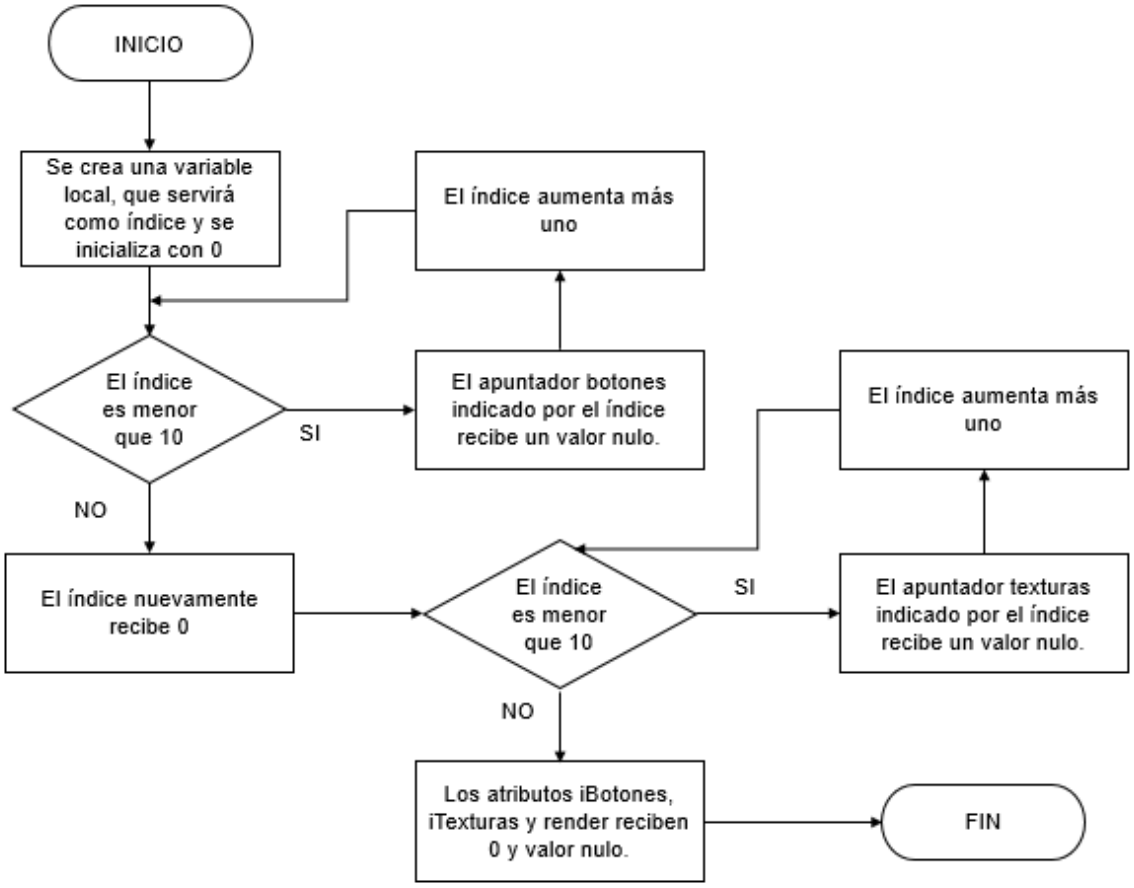



*Diagrama de flujo 6. Constructor de la clase CInterfaz*

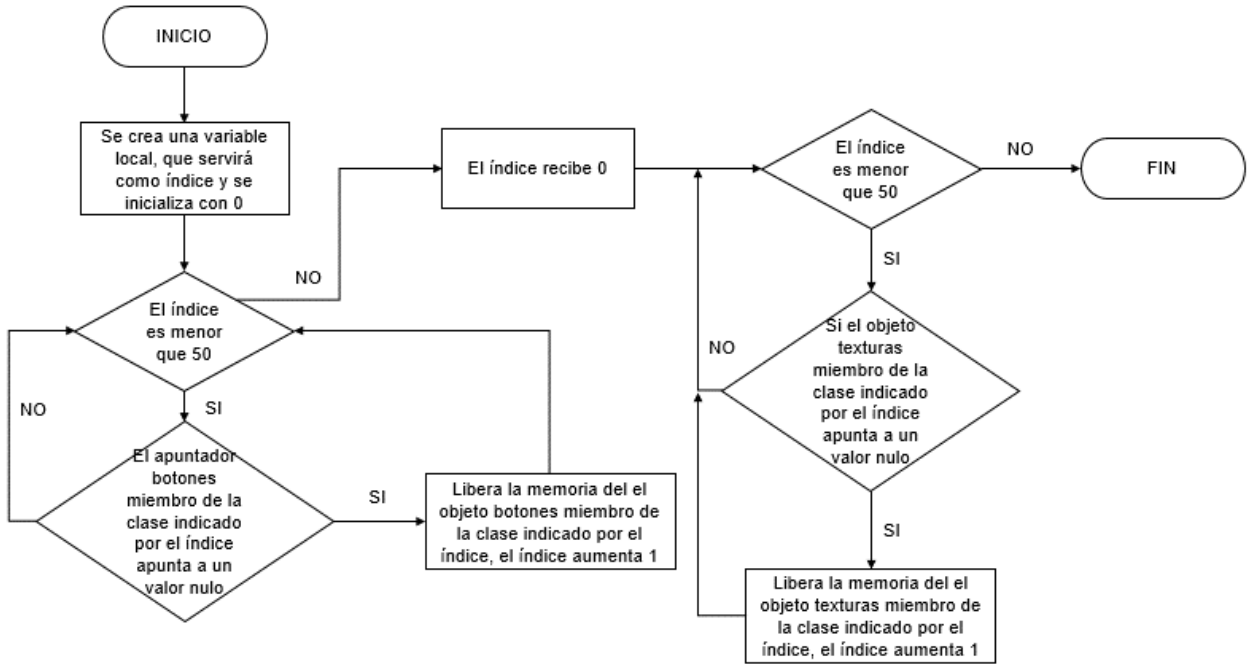



*Diagrama de flujo 7 del destructor de la clase CInterfaz*

### 7.5.1 Definición de la función cargarTextura()

Esta función se utiliza para agregar imágenes a una interfaz, esta función miembro facilita las cosas para un programador porque la función únicamente requiere la posición, el tamaño y la ruta en donde se encuentra almacenada la imagen, el hecho de que se cargue una textura a la interfaz no significa que esta se mostrara en la pantalla.

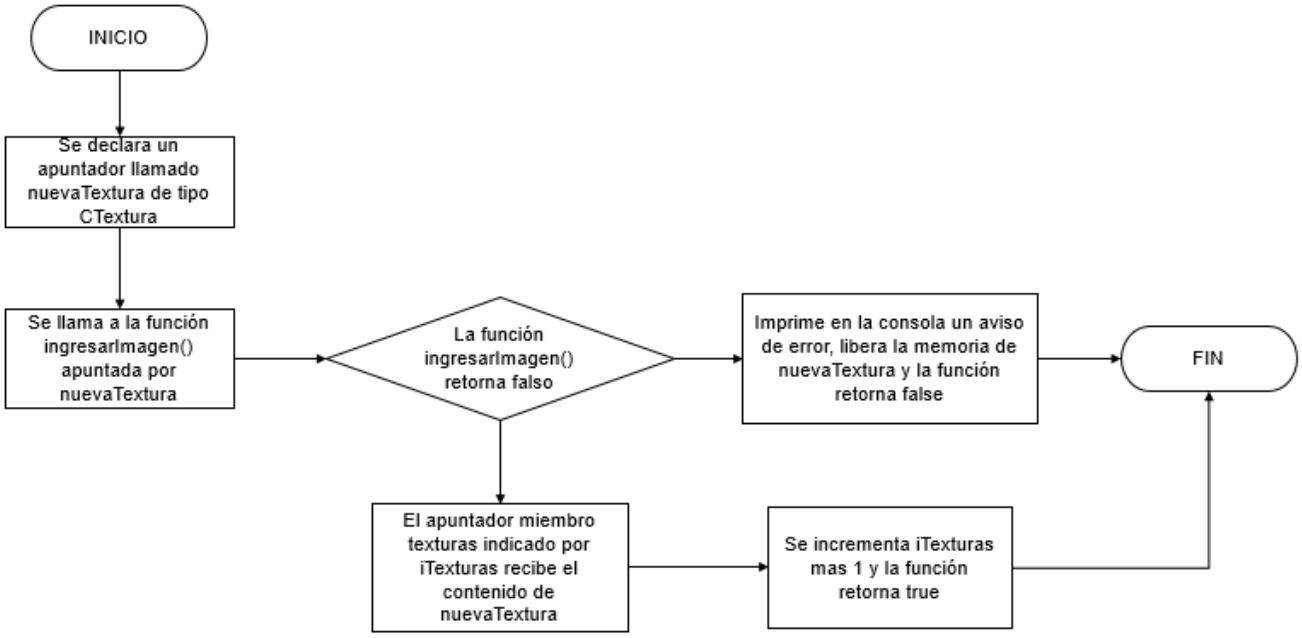



*Diagrama de flujo 8 Cinterfaz::cargarTextura()*

### 7.5.2 Definición de la función cargarBoton()

Esta función prácticamente es idéntica a la de cargarTextura(), con la única diferencia de que la imagen es cargada con todas las características de un botón.

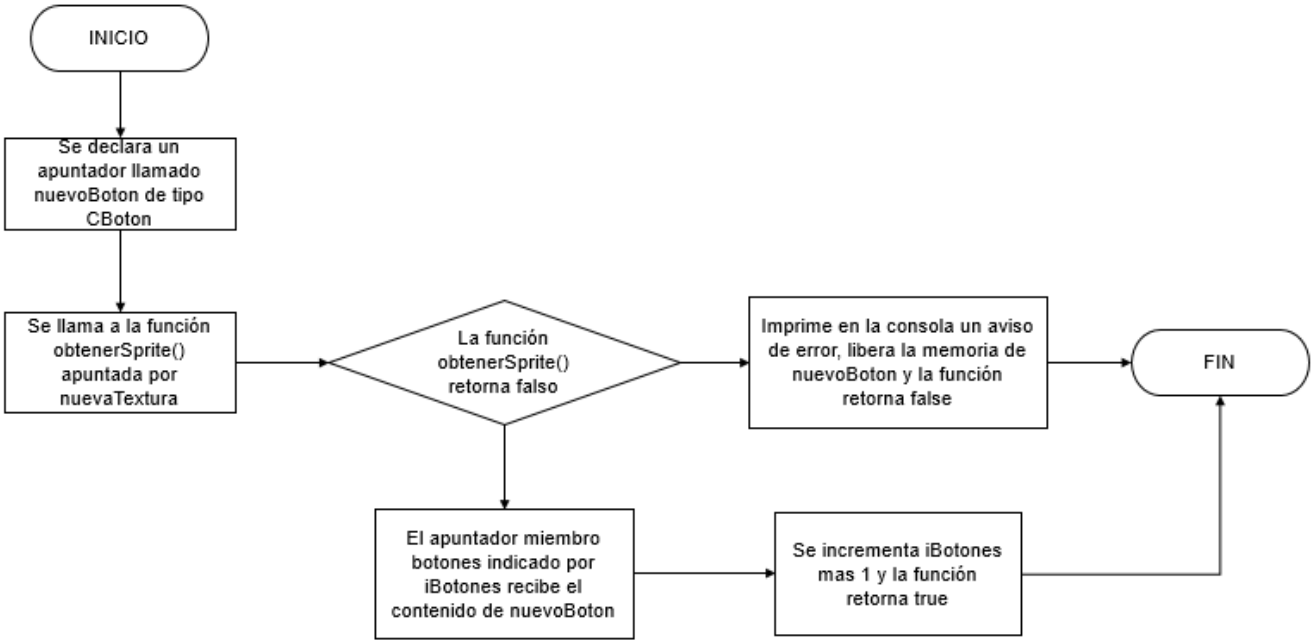



*Diagrama de flujo 9 Cinterfaz::cargarBoton()*

### 7.5.3 Definición de la función asociarVentana()

Esta función es necesaria ya que el programa necesita saber con qué objeto de tipo render se renderizaran las imágenes de la interfaz, las librerías SDL tiene esta característica, por lo tanto, no podemos evitarla.

### 7.5.4 Definición de la función mostrar()

Como anteriormente mencionamos que el hecho de cargar texturas y botones a una interfaz no significa que éstas se muestren en la pantalla por lo tanto, la función mostrar() como su nombre lo indica, tiene la tarea de mostrar en la pantalla la interfaz con las texturas y botones que se le cargaron previamente, esta función tiene dos modalidades, la primera y la más básica, es mostrar todos los elementos cargados a la interfaz, y la segunda es mostrar únicamente un elemento que elijamos.

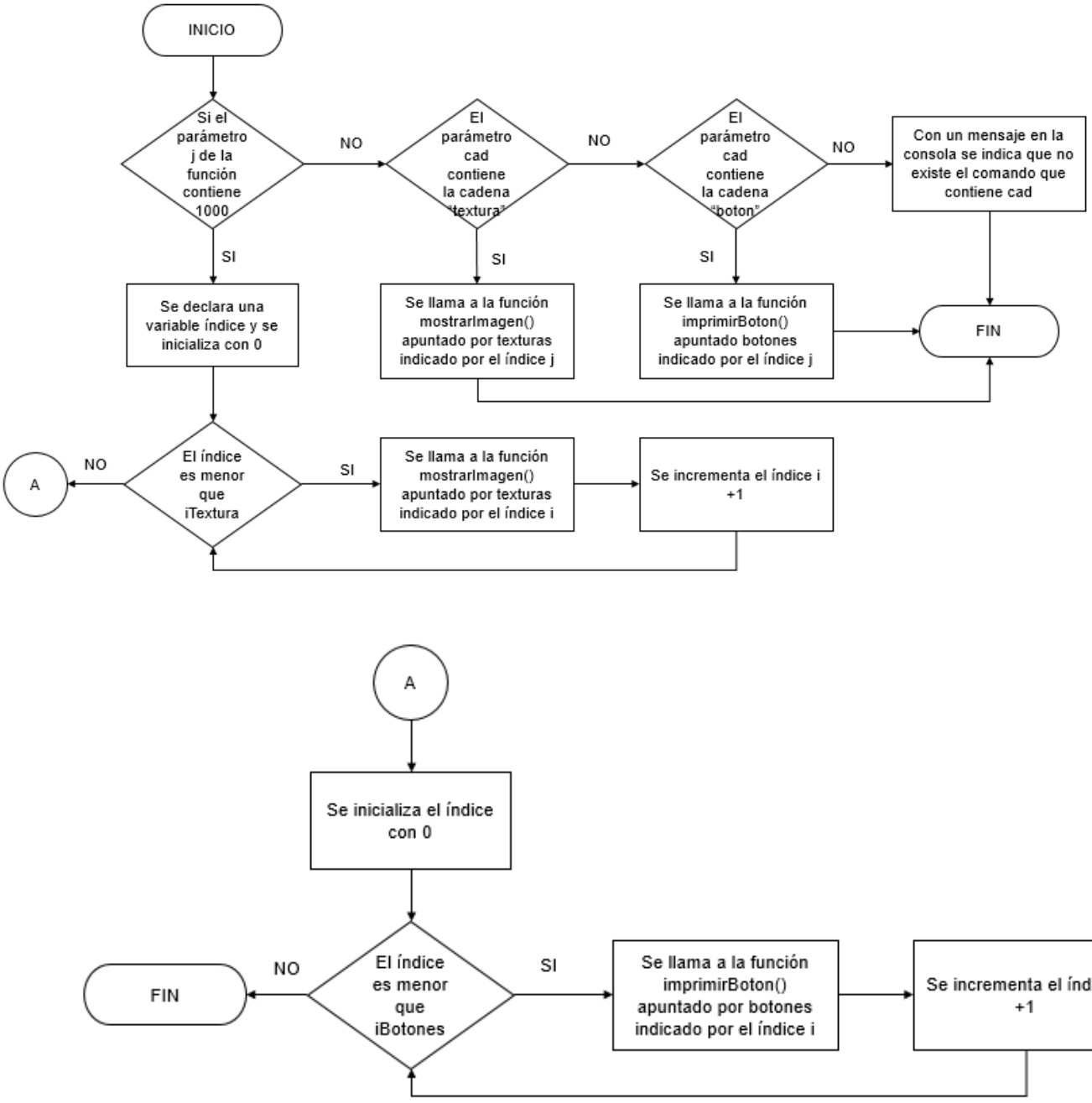



*Diagrama de flujo 10 Diagrama de la función CInterfaz::mostrar()*

### 7.5.5 Definición de la función touchBoton()

Esta función pone a la escucha a todos los botones de la interfaz, si alguno de los botones fue tocado por el usuario, la función touchBoton retorna el ID del botón.

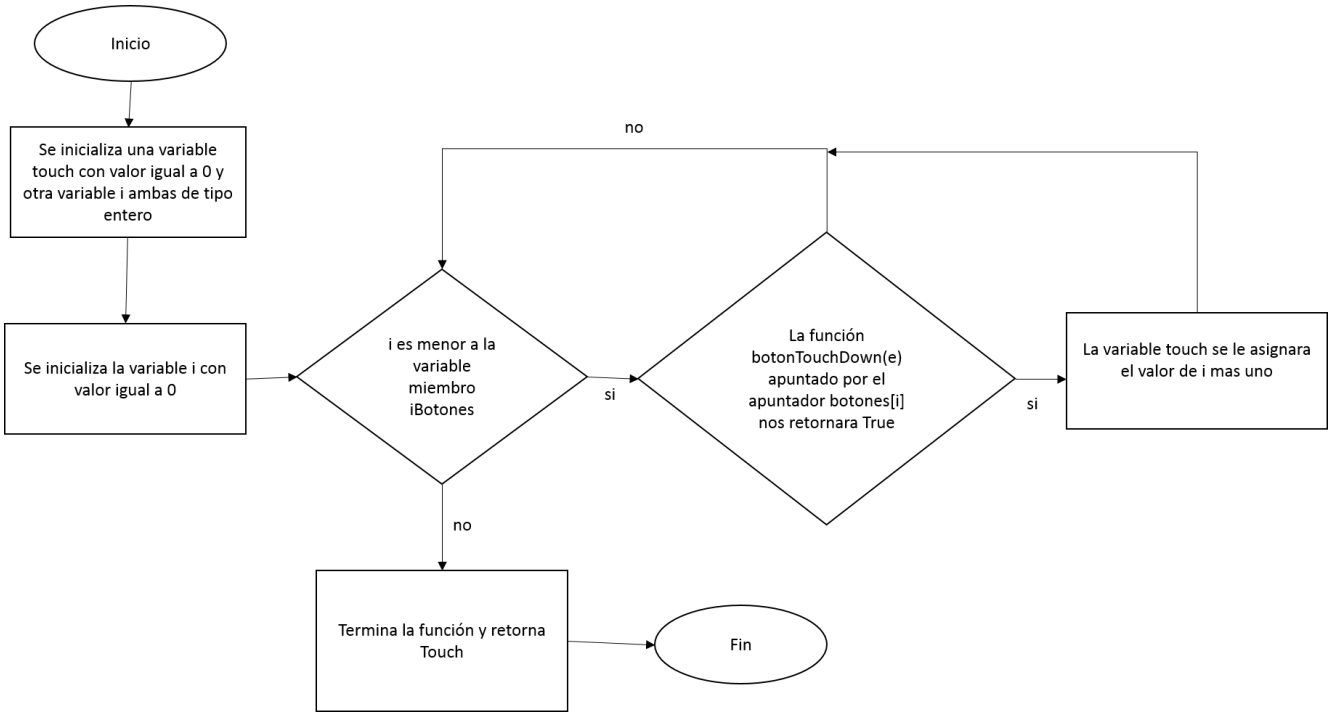



*Diagrama de flujo 12 Función CInterfaz::touchBoton()*

## 7.6 Estructura kiosco.

Para la realización de la estructura se planteó en primer lugar la realización de un diseño asistido por computadora esto para poder visualizar de manera gráfica el diseño propuesto. Para esto se tomaron en cuenta diversos modelos de los cuales se tomó el mejor y el más practico que es el que se muestra en la siguiente imagen.

Figura 3.1 Diseño vitual de la estructura del kiosco, vista lateral.

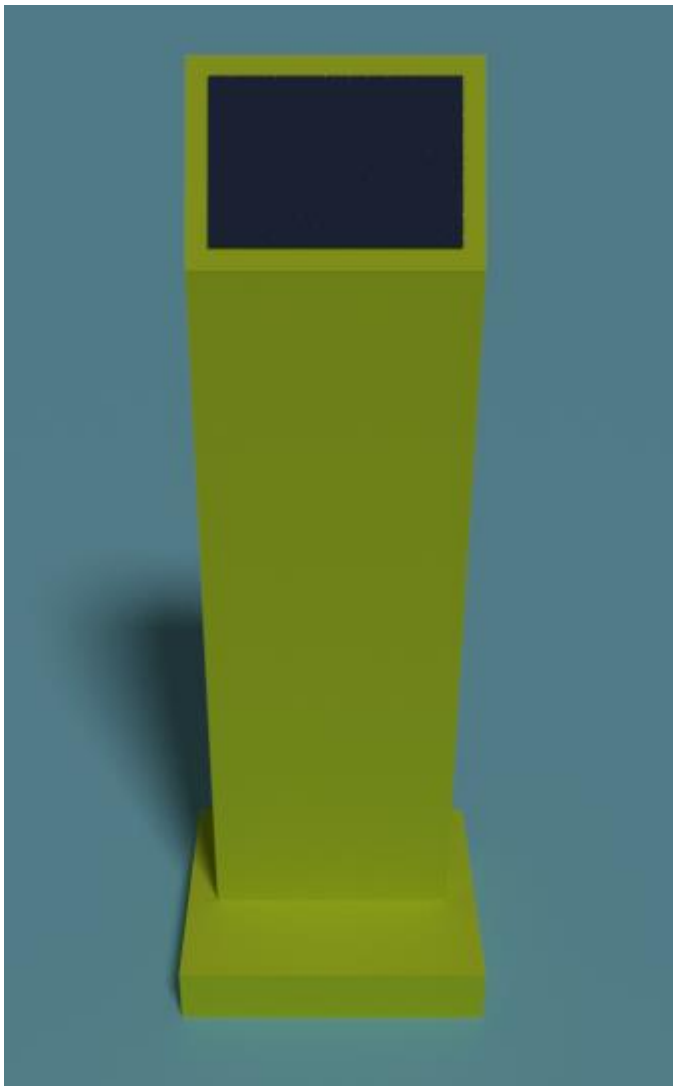


Figura 3.1 Diseño vitual de la estructura del kiosco, vista frontal.

Como se puede visualizar en la imagen del diseño realizado con la herramienta Blender el cual nos permite visualizar de manera real el tamaño que tendrá y como se puede llegar a visualizar una vez concluido, con esto se tiene una predicción y una perspectiva mejor de las dimensiones que se van a emplear en la construcción de la estructura ya que esta herramienta nos permite hacer el diseño con las medidas propuestas.

También se propone realizar una maqueta a escala de la estructura esto para tener un modelo de manera física de la estructura y con esto poder observar más a detalle las consideraciones que se deben tomar en cuenta para su construcción tales como, inconvenientes en el diseño, visualización de manera real de la estructura esto de manera a escala para hacer las correcciones pertinentes. Como conclusión de lo anterior se confirmó que el modelo utilizado es el más ideal para el modelo final de la estructura del kiosco.

Las pruebas anteriores además de ayudarnos con lo mencionado en los párrafos anteriores nos evitaran hacer gastos innecesarios y malgastar recursos y hacer que el proyecto sea más sustentable.