

**UNIVERSIDAD NACIONAL DE SAN AGUSTÍN
FACULTAD DE PRODUCCIÓN Y SERVICIOS
ESCUELA PROFESIONAL DE CIENCIA DE LA
COMPUTACIÓN**



“DOCUMENTACIÓN DEL APLICATIVO”

ASIGNATURA:
CIENCIA DE LA COMPUTACIÓN II

SEMESTRE:
III

ESTUDIANTES:

Roque Sosa, Owen Haziél
Zhong Callasi, Linghai Joaquin
Ruiz Mamani, Eduardo German
Velásquez Montoya Juan José
Castillo Sancho Sergio Ahmed

Arequipa - Perú

2022

DOCUMENTACIÓN DEL PROYECTO

Título del proyecto	Programa para la agendación de citas de centros médicos
Código del proyecto	Enlace a repositorio en Github:
Tipo de documento adjunto	Informe del Proyecto (PDF)
Organización/Usuario para quien se redacta el proyecto	
Autor (es)	<ul style="list-style-type: none"> • OWEN HAZIEL ROQUE SOSA • Zhong • Eduardo • Velásquez Montoya Juan José • Castillo Sancho Sergio Ahmed
Duración estimada en jornadas:	<p>Fecha de inicio: 1 de Junio de 2022</p> <p>Fecha de Entrega (III Parcial): 19 de agosto de 2022</p>

Se adaptaron las funciones a QT:

Login en QT:

Una función hash criptográfica es un algoritmo que toma una cantidad arbitraria de entrada de datos (una credencial) y produce una salida de tamaño fijo de texto cifrado llamado valor hash, o simplemente "hash". Se incluyó una librería de hashing para encriptar las contraseñas.

Constructor de la clase Login:

Crea el puntero ui.

Destructor de la clase Login:

Elimina el puntero ui.

Función on_btn_login_clicked:

Chequea si el usuario existe en la database, si existe, chequea el rol del usuario y abre la ventana correspondiente.

Archivo citascalendar.cpp:

Constructor citasCalendar:

Permite construir las citas con el id, nombren DNI, telefono, edad, hora, turno, estado de los pacientes con un médico (con código y fecha que coincidan).

Destructor citasCalendar.

Función updateDataCalendar:

Actualiza las citas.

Función on_pushButton_clicked:

Función on_tableView_clicked:

Genera la tabla con la información de la cita: nombre, dni, telefono, edad, hora.

Archivo dialog_agregar_citas.cpp:

Constructor Dialog_agregar_citas

Destructor Dialog_agregar_citas

Función on_agregar_clicked:

Toma el input del usuario, chequea si es válido. Si lo es, lo agrega a la base de datos.

Función on_volver_clicked:

Cuando el usuario hace click en el botón VOLVER, el programa crea un nuevo objeto-vistadm y lo muestra, luego cierra la ventana.

Archivo newmed.cpp

Constructor newmed:

Permite crear la ventana mediante la función setupUi(this)

Esta función es generada por QT Designer.

Destructor newmed.

Función on_button_ingresar_clicked:

Inserta una nueva fila en la tabla medico y luego obtiene la clave primaria de esa fila y la inserta en la tabla usuarios con el usuario y la clave.

La consulta retorna el número de filas afectadas por la consulta(query).

Archivo newpac.cpp (Nuevo paciente):

Constructor newpac:

Permite crear la ventana mediante la función setupUi(this)

Esta función es generada por QT Designer.

Destructor newpac.

Función on_button_ingresar_clicked:

Inserta una nueva fila en la tabla paciente y luego obtiene la clave primaria de esa fila (id).

Archivo vistaadm.cpp:

Constructor y destructor para generar la ventana.

Función on_btn_agregarpac_clicked:

Crea un objeto para el nuevo paciente.

Abre una nueva ventana y cierra la actual.

Función on:btn:agregarmed:clicked:

Crea un objeto para el nuevo médico.

Abre una nueva ventana y cierra la actual.

Función on:btn:agregarcita:clicked:

Crea un objeto para la nueva cita.

Abre una nueva ventana y cierra la actual.

Archivo widget.cpp:

Arma la base de datos.

Chequea si está abierta. Si no está abierta, retorna error.

Si está abierta, procede a crear una nueva instancia de la clase QSqlTableModel, seleccionar la data necesaria de la database y prepararla para la vista de la tabla (tableview).

Anexos: Funciones originales elaboradas en C++ que sirvieron como base para la documentación:

Main:

Clase one:

Es una clase con las funciones virtuales puras get() y mostrar()

Clase infoDeCita:

Es una clase pública hija de la clase one.

Función get:

Es usada para tomar la información del paciente.

Función mostrar:

Muestra la información del paciente.

Clase citaMedico:

Hereda de la clase InfoDeCita.

Función get:

Escribe la data del objeto a1 hacia el archivo medico1.txt:

a1 -> medico1.txt

Función mostrar:

Si el documento está vacío, muestra un mensaje y regresa al menú.

De otro modo, lee el documento y muestra el contenido.

Clase medico:

Hereda de la clase pública one.

Se usa para guardar información sobre un médico.

Función get:

Guarda información del doctor en un archivo medico.txt

Función mostrar:

Muestra la información.

Clase paciente:

Hereda de la clase one.

Guarda la información del paciente.

Función get:

Guarda información del paciente en un archivo paciente.txt

Función mostrar:

Muestra la información.

Clase administrador:

Hereda de la clase one.

Guarda la información del paciente.

Función get:

Guarda información del administrador en un archivo administrador.txt

Función mostrar:

Muestra la información.

Función reservarCita:

Compara el input del usuario con los código de la clase medico (medico class). Si coinciden, se llama a la función get de la clase medico.

Función mostrarCita:

Compara el input del usuario con los código de la clase medico (medico class). Si coinciden, se llama a la función mostrar

Función contrasena: //

Lee la contraseña de un archivo password .txt y la compara con el input. Si coinciden, se dirige al menú. Si no, la función se llama a sí misma.

Función cambiarContrasena: // esto menos

Abre el archivo password.txt, consulta el ingreso de una nueva contraseña y escribe la nueva contraseña en el archivo.

Archivo menudoctor.cpp:

Es importante aclarar que al incluir `<QShared Pointer>`, se incluye un puntero inteligente que elimina el objeto cuando está fuera de rango.

Constructor menuDoctor:

Contiene una función que se llama para preparar la interfaz del usuario (ui).

Establece el texto de la etiqueta `txt_bienvenida` con el contenido "Bienvenido".

Asigna el valor del parámetro `idreferencia` a la variable `idReferencia`.

Finalmente, conecta la señal `clicked()` del botón `btn_citasCalendar` a `btn_citasCalendar_clicked()` de la clase `menuDoctor`.

Destructor menuDoctor.

Función `on_btn_citasCalendar_clicked`:

Si no se ha creado el calendario de citas (es decir, `wdg_citas==nullptr`), se crea el objeto `citasCalendar(idReferencia)` y se muestra.

Si se ha creado, solo se muestra.

Función `on_btn_logout_clicked`:

Si el usuario hace clic en el botón de cierre de sesión, aparecerá un cuadro de mensaje que le preguntará si está seguro de que desea para cerrar sesión. Si el usuario hace clic en 'Aceptar', el programa se cerrará. Si el usuario hace clic en 'Cancelar', el programa seguirá funcionando.