

Sistema de Monitoreo y control de pacientes en una Unidad de cuidados intensivos.

Daniel Leonardo González Moreno
Bogotá, Fundación Universitaria del Área Andina

ABSTRACT

En la actualidad y con la situación presente, resultado de la pandemia del covid-19, se ha requerido de software que le permita a las unidades médicas tener acciones inmediatas y eficaces para poner freno al avance de la enfermedad en pacientes críticos que llegan a estar en condiciones de salud complicadas. Aun cuando existen sistemas eficaces que logran tener un funcionamiento óptimo, es importante saber sobre su funcionamiento e innovar en las distintas características que se pueden implementar adicionalmente en pro de mejorar.

Índice de Términos

- **Python:** Es un lenguaje de programación.
- **Xampp:** Aplicación que brinda características para el montaje de bases de datos y herramientas de servidor web.
- **Consultas a base de datos:** Transacciones que se pueden llevar a cabo en una Base de datos.
- **Saturación de oxígeno:** Es la medida de la cantidad de oxígeno disponible en la sangre.

I. INTRODUCCIÓN

Actualmente existen muchos sistemas de información que permiten tener un control transaccional y eficaz para el registro de pacientes y su historia clínica, la intención del presente proyecto es determinar este funcionamiento y adicionar funciones que mejoren y amplíen la gama de opciones en cuanto a monitoreo y toma de medidas en caso de que el paciente entre en un estado crítico. Para lo anterior se usaron herramientas de desarrollo y gestión de bases de datos como lo fueron Python y SQL.

Siendo en el proyecto un factor determinante (variable) la saturación de oxígeno en el cuerpo humano, fue necesario revisar los estándares generales, recomendados y críticos de este componente, ya que uno de los síntomas más frecuentes e importantes de un paciente en condición de hospitalización por covid-19 es la dificultad para respirar y síntomas de ahogamiento.

Es por esto que el dato variable e importante del monitoreo del paciente en el presente proyecto esta predisposto por los niveles de oxígeno para que de esta forma el software se encargue de tomar decisiones que permitan actuar de forma rápida y evitar sucesos críticos irreparables o incluso puedan desencadenar la muerte.

Saturación de Oxígeno	
% de Saturación	Concepto
Mayor a 90 %	Valores Óptimos(Paciente sin problemas respiratorios)
Entre 85-90 %	Tratamiento inmediato e Hipoxemia moderada.
Menor a 85%	Enfermo grave, Hipoxemia severa.
Menor a 80%	Enfermo crítico: Posible entubación y ventilación mecánica.

Fig1: Niveles de saturación de Oxígeno- Elaboración propia

II. FUNCIONALIDAD

El presente proyecto presenta una funcionalidad que comprende desde el registro de pacientes que ingresen a una unidad médica hasta el formulario de inicio de sesión del médico de cabecera, para que realice evaluaciones periódicas de los pacientes que han ingresado al sistema. El dato clave en este caso es la saturación del oxígeno, el cual actualmente no se encuentra programado ya que su funcionamiento se basa en la recepción del dato de un oxímetro que va a estar conectado todo el tiempo al paciente, de esta forma se activará una alarma si los niveles de oxígenos no son los óptimos y se enviará un correo electrónico o mensaje de texto a la unidad médica para su atención inmediata.

III. PROYECCION

El proyecto tiene actualmente un registro y monitoreo de pacientes, cabe destacar que tiene bastantes oportunidades de mejora para garantizar un mejor y más completo funcionamiento en el marco de las urgencias médicas. Actualmente la importancia de la automatización y la inteligencia artificial es un ítem trascendental en la ejecución de tareas de la sociedad moderna por lo cual fije mi atención en los siguientes ítems que por cuestiones de tiempo no fue posible implementar inmediatamente.

A. Oxímetro basado en pulsos.

Definitivamente el proyecto requiere de esto para que una de sus tareas fundamentales se lleve a cabo, con la inclusión de

este dispositivo se puede garantizar la supervisión automatizada en tiempo real del estado del paciente, dato que se irá almacenando en el formulario de monitoreo e irá cambiando a medida que el oxígeno tome diversos valores. Lo anterior se podrá realizar con la herramienta de desarrollo “Python”, la librería *machine* nos permitirá tener las características que necesitamos para llevar a cabo la parte lógica.

```
from machine import*
from machine import Pin, PWM
```

Fig. 2. Importación de librerías en Python

Por otra parte a nivel de insumos físicos necesitaremos para llevar a feliz término el laboratorio de un arduino, protoboard, resistencias, pantalla oled y pulsadores.

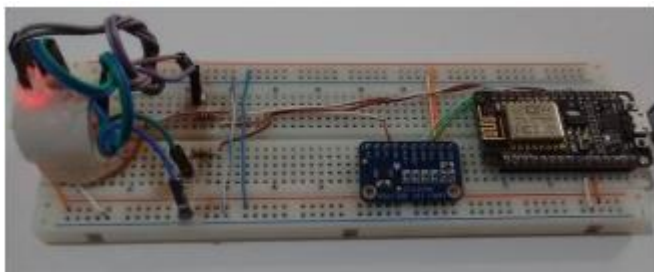


Fig3: Montaje de Oxímetro en protoboard: Fuente externa: Prototype of a Pulse Oximeter, Facultad de Ciencias de la Computación, Puebla, México

B. Envío de correo cuando los valores se encuentren fuera de los parámetros óptimos.

Las notificaciones cuando la variable de oxigenación en la sangre cambie se realizarán a través de un condicional el cual tendrá la estructura: Si el valor de la oxigenación es menor a los estándares óptimos debe proceder a enviarse un mensaje al cuerpo médico para que atienda de inmediato al paciente.

```
# import necessary packages
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
import smtplib

# create message object instance
msg = MIMEMultipart()

message = "Thank you"

# setup the parameters of the message
password = "your_password"
msg['From'] = "your_address"
msg['To'] = "to_address"
msg['Subject'] = "Subscription"

# add in the message body
msg.attach(MIMEText(message, 'plain'))

# create server
server = smtplib.SMTP('smtp.gmail.com: 587')

server.starttls()

# Login Credentials for sending the mail
server.login(msg['From'], password)

# send the message via the server.
server.sendmail(msg['From'], msg['To'], msg.as_string())

server.quit()

print "successfully sent email to %s:" % (msg['To'])
```

Fig4: Método y librerías necesarias para el procedimiento. Fuente: code.tutsplus

IV. MANUAL DE USUARIO

A. Pantalla Inicial

La pantalla inicial de la aplicación se abrirá a partir del archivo *main.py*, el cual consta del nombre de la unidad médica y tres botones: *Registrar Paciente*, *Ingresa* y *Salir* cada uno de ellos programado para llevar a cabo su función principal.

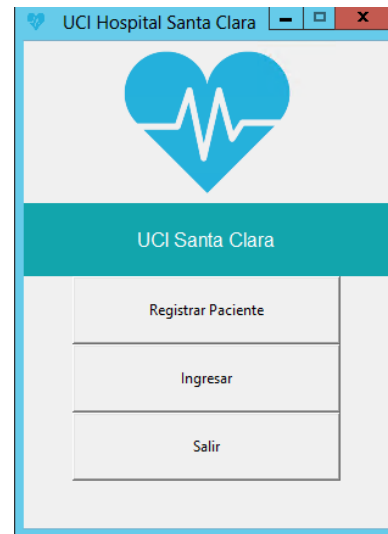


Fig. 5. Form Pantalla Inicial.

B. Form: Registro de paciente:

Es la pantalla en la cual se van a registrar los datos del paciente que ingresa a la unidad médica, consta de cajas de texto para insertar los datos y dos botones, uno para guardar la información y el otro para salir de la pantalla de registro.

Fig. 6. Form Pantalla Inicial

C. Form: Registro de paciente (Botón guardar):

El botón permitirá guardar en la base de datos los datos insertados.

D. Form: Ingreso:

Es el menú de login de la aplicación en el cual el personal médico debe autenticarse para ingresar y validar la información de los pacientes previamente ingresados

Autenticación de Usuarios

Ingrese su nombre de usuario y contraseña

Nombre de Usuario:

Contraseña:

Ingresar

Fig. 7. Form login

E. Form: Monitoreo

Este formulario es en el que se podrá buscar la información a partir del número de documento del paciente, solo se podrá acceder a él cuándo el login sea exitoso, dispone de dos botones uno para realizar la consulta contra la base de datos y el otro para salir del formulario de consulta.

Autenticación de Usuarios

Ingrese su nombre de usuario y contraseña

Nombre de Usuario: dgonzalez

Contraseña: ****

Ingresar

Bienvenido

Usuario y clave correcta

Aceptar

Fig. 8. Login Exitoso

Monitoreo Pacientes UCI

Número de identificación

Fecha de ingreso

Nombres:

Apellidos:

Sexo:

Edad:

RH:

Saturación de Oxígeno:

Consultar

Salir

Fig. 9. Pantalla inicial de monitoreo de pacientes

F. Botón Consulta, Form monitoreo

Es el encargado de accionar la consulta contra la base de datos, es necesario haber diligenciado el número de documento.

Monitoreo Pacientes UCI

Número de identificación

1013648881

Fecha de ingreso

25/07/2021

Nombres:

Daniel

Apellidos:

Gonzalez

Sexo:

M

Edad:

27

RH:

O+

Saturación de Oxígeno:

Consultar

Salir

Fig. 10. Form con información traída a partir del número de documento y el botón "consultar"

Como se puede apreciar en el campo de saturación de oxígeno no trae ninguna información ya que el desarrollo del dispositivo que debe tomar este dato no se realizó. Con la implementación del oxímetro este valor variaría en tiempo real y cada que se requiera consultar el comportamiento en pantalla.

V. MANUAL DEL DESARROLLADOR

A. Resumen

El presente documento describe las diferentes formas y procedimientos técnicos e informáticos del sistema de información "Sistema de Monitoreo y control de pacientes en una Unidad de cuidados intensivos". La intención del actual texto es familiarizar al personal capacitado para realizar modificaciones, revisiones y mantenimiento a dicho sistema.

A. Objetivos

- Relacionar el manual de usuario anteriormente especificado para garantizar el acceso y operatividad óptimos requeridos para su implementación y uso.
- Relacionar detalles técnicos y creativos que se enuncian en el presente documento para de esta forma garantizar la continuidad de la operación y la mejora continua del sistema de información.
- Especificar la descripción de los archivos relevantes del sistema los cuales serán pilar de la configuración y soporte lógico del mismo.

B. Requerimientos

El software necesitó los siguientes componentes para su desarrollo y funcionamiento óptimos según las pruebas realizadas:

- Python 3.9.6
- Xampp: Servidor MySQL
- Visual Studio Code
- Tipo de servidor: MariaDB
- Versión: 10.4.20

Librerías:

- Tkinter
- Pymysql (Requiere instalación)
- Datetime

C. Requerimientos Hardware

El sistema fue desarrollado sobre la siguiente infraestructura física:

- Servidor físico Proliant ML110 GEN 7
- RAM: 6 GB
- Disco: 512 GB
- Windows Server 2012

D. Configuración y puesta en marcha

Para la estructuración del sistema se requirió de disponer de diversas herramientas a nivel lógico y de infraestructura, los cuales están relacionados en los párrafos anteriores. Para obtener el producto final se realizaron los siguientes pasos.

- Destinación de infraestructura.
- Instalación de software que permitiera interacción entre el desarrollador y la máquina.
- Análisis del planteamiento
- Diseño de Solución.
- Diseño Bases de Datos
- Desarrollo/ Programación
- Pruebas
- Implementación.

E. Modelo Entidad relación:

Para la estructuración de las bases de datos no se usaron relaciones entre tablas. El proyecto dispone de dos tablas una de login, otra de monitoreo y registro de pacientes:

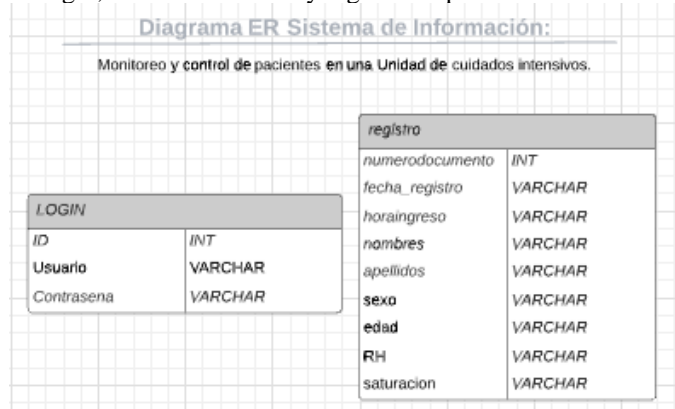


Fig. 11 Modelo ER, diseñado con lucidchart

A. Configuración del Sistema

- Estructura Base de datos: La base de datos tiene una estructura realizada a partir de SQL, para la cual se

implementó XAMPP. Actualmente dispone de dos tablas, una de login y otra de registro, ésta última será en la cual repose toda la información de los pacientes que se inserten en el sistema de información.



Fig. 12. Estructura Base de datos en XAMPP



Fig. 13. Tabla de login en XAMPP

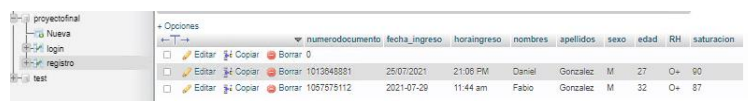


Fig.14. Tabla de login en XAMPP

- Estructura Raiz

El actual proyecto dispone de la siguiente estructura de directorios:

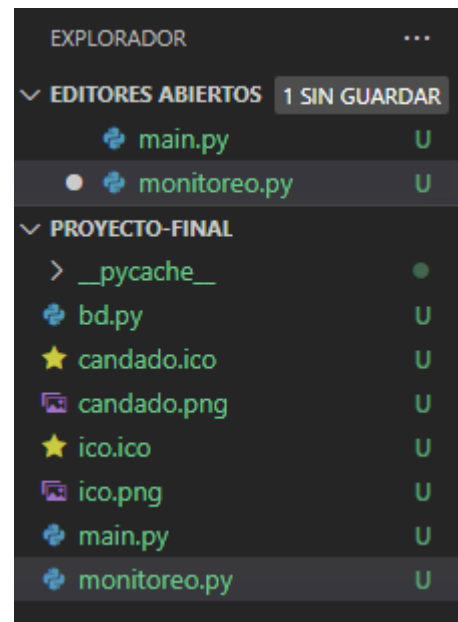


Fig. 15. Esquema tomado de visual studio code

- Estructura Aplicación

El actual proyecto contiene el siguiente esquema que se realizó para completar el producto final.

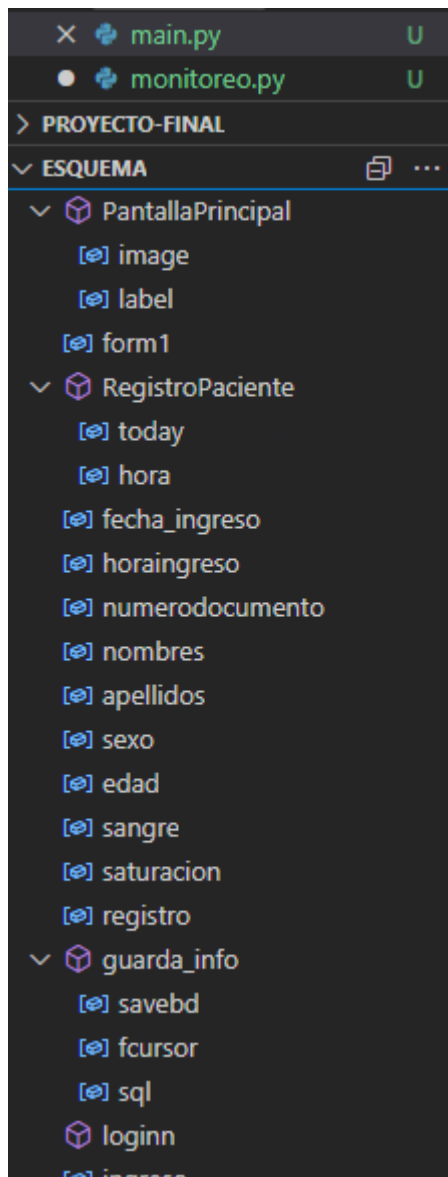


Fig. 16. Esquema main.py, visual studio code

El proyecto consta de una estructura simple en la cual interactúan dos archivos relacionados entre sí, cada uno con diversas funciones que garantizan el funcionamiento de la aplicación. La pantalla de inicio le ofrece al usuario la posibilidad de elegir diversas opciones según el requerimiento que se tenga, por una parte para registrar un paciente que ingresa a la unidad médica, este paso no requiere autenticación. Por otra parte tenemos la opción de ingresar a revisar el estado de los pacientes, en esta opción si se debe autenticar el usuario, de esta forma accederá a un formulario muy similar al de registro y tendrá la posibilidad de hacer la consulta contra la base de datos del paciente que desee a partir del número de identificación.

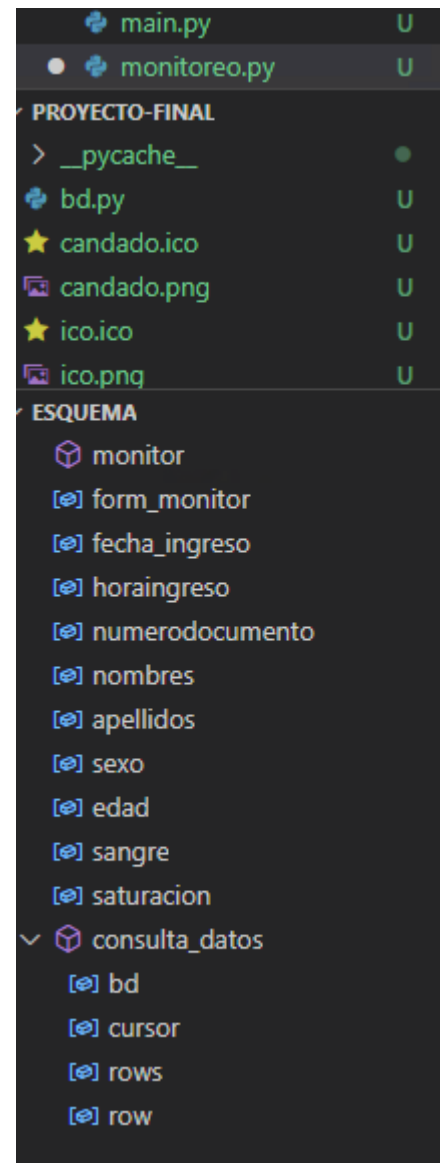


Fig. 17. Esquema monitoreo.py, visual studio code

- Estructura Librerías:

```
main.py > ...
1  """Importación Librerías"""
2  import tkinter
3  from tkinter import *
4  from tkinter import messagebox
5  import pymysql
6  from pymysql import *
7  from monitoreo import monito
8  from datetime import *
9  import time
10
```

Fig. 17. Vista previa de librerías usadas en main.py, visual studio code.

Tkinter: Se usó esta librería para la creación de los formularios y botones requeridos para la aplicación.

```

def PantallaPrincipal():
    global form1
    form1=Tk()
    form1.geometry("300x400")
    form1.title("UCI Hospital Santa Clara")
    form1.iconbitmap("ico.ico")
    form1.resizable(False,False)

    image=PhotoImage(file="ico.png")
    image=image.subsample(2,2)
    label=Label(image=image)
    label.pack()

    Label (text="UCI Santa Clara", bg="#12A5A5", fg="white").pack()

    Button(text="Registrar Paciente", height="3", width="15", bg="#12A5A5", fg="white").pack()

    Button(text="Ingresar", height="3", width="15", bg="#12A5A5", fg="white").pack()

```

Fig. 18. Diseño y Creación de formulario a partir de Tkinter, Visual Studio Code.

Pymysql: Se usó para realizar la conexión y consultas con la base de datos.

```

def guarda_info():
    savebd=pymysql.connect(
        host="localhost",
        user="root",
        passwd="Pa55w.rd",
        db="proyectofinal"
    )
    fcursor=savebd.cursor()

    sql= "INSERT INTO registro (numerodocumento, fecha_ingreso, horaingreso) VALUES (%s,%s,%s)"
    datos=(numerodocumento, fecha_ingreso, horaingreso)
    fcursor.execute(sql, datos)
    savebd.commit()
    savebd.close()

```

Fig. 19. Creación de conexión a base de datos y elaboración de consulta, Visual Studio Code

Datetime: Se usa la librería para tener la hora y la fecha instantánea en el formulario del registro del paciente.

```

today = date.today()
hora = time.strftime('%H:%M:%S', time.localtime())
horaingreso= Label (registro, text= today, bg="#12A5A5", fg="white").pack(anchor= E)
fecha_ingreso= Label (registro, text= hora, bg="#12A5A5", fg="white").pack(anchor= E)

```

Fig. 19. Definición de fecha y hora, Visual Studio Code

Monitoreo: Se importa para tener acceso al formulario de registro.

VI. CONCLUSIONES

- La atención hospitalaria requiere de una mayor automatización, la incursión de nuevas tecnologías que incluyan soluciones basadas en la inteligencia artificial puede ahorrar costos y ayudar a prestar una mejor y más precisa atención a pacientes críticos.
- A nivel de desarrollo se evidencia que herramientas como Python ofrecen mayores facilidades en la implementación de nuevas herramientas, como también su facilidad en el aprendizaje que le garantiza al desarrollador un manejo rápido en su estructura y familiarización.

- La interacción entre software y hardware definitivamente a tenido un salto gigante en el camino desde hace diez años, cuando para tan siquiera encender un led había que efectuar un proceso largo y complejo de programación y montaje.
- El curso de programación en Python es una excelente opción para dar los primeros pasos en este ámbito de la ingeniería, por su facilidad en la interpretación del lenguaje es una puerta a grandes resultados en el desarrollo de software.

VII. REFERENCIAS

- [1] Cisco Networking Academy, Diplomado de Programación en Python.
- [2] github.com.
- [3] Research in Computing Science(Mexico).
- [4] Introducción a las Bases de Datos(platzi.com)
- [5] Ejemplos de la GUI de Python, likegeeks.com
- [6] Login Page in Python with MySQL Database with Source Code, itsourcecode.com

Autor:

González Moreno Daniel Leonardo
Ingeniería de Sistemas (X Semestre)
Fundación Universitaria del Área Andina.