

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERIA EN CIENCIAS Y SISTEMAS

MANUAL TECNICO

Proyecto #1

Edgar Rolando Alvarez Rodriguez 202001144

Fecha: 03-Mayo-2021

INDICE

INTRODUCCION	3
Objetivo del Manual:	3
REQUERIMIENTOS.....	4
Requisitos de Hardware	4
Requisitos de software	4
EJECUCION DEL PROGRAMA.....	5
CLASES UTILIZADAS	5
Procedimientos	6
Funciones.....	7
MAIN.PY	7
GESTOR.PY	7
FLUJO DEL PROGRAMA	8

INTRODUCCION

El presente manual técnico tiene como finalidad describir el diseño del programa llamado "Proyecto 2" que proporciona una interfaz gráfica para la administración de un hospital por medio de una página web.

Objetivo del Manual:

Instruir el uso adecuado del Sistema de Información, para el acceso oportuno y adecuado en la instalación de este, mostrando los pasos a seguir en el proceso de instalación, así como la descripción de los archivos relevantes del sistema los cuales nos orienten en la configuración y soporte de este.

REQUERIMIENTOS

Requisitos de Hardware

Una computadora con:

- o Procesador
- o Memoria RAM
- o Tarjeta gráfica
- o Disco duro
- o Teclado completo
- o Mouse o Batería

Requisitos de software

- o Sistema operativo: Windows, Mac o Linux.
- o Un navegador Instalado

EJECUCION DEL PROGRAMA

1. Se debe abrir el navegador
2. Se debe acceder a este sitio web:

[UHospital \(storage.googleapis.com\)](http://UHospital(storage.googleapis.com))

<http://storage.googleapis.com/ipc1front202001144/FrontEnd/Medilab/index.html#footer>

CLASES UTILIZADAS

Se utilizaron 6 objetos para las distintas funcionalidades del sistema

Citas: esta clase se encarga de instanciar el objeto "Cita" que será utilizado posteriormente en distintos ámbitos

Enfermeros: esta clase se encarga de instanciar el objeto "Enfermero" que será utilizado posteriormente en distintos ámbitos

Medicamentos: esta clase se encarga de instanciar el objeto "Medicamento" que será utilizado posteriormente en distintos ámbitos

Médicos: esta clase se encarga de instanciar el objeto "Medico" que será utilizado posteriormente en distintos ámbitos

Pacientes: esta clase se encarga de instanciar el objeto "Paciente" que será utilizado posteriormente en distintos ámbitos

Solicitudes: esta clase se encarga de instanciar el objeto "Solicitud" que será utilizado posteriormente en distintos ámbitos

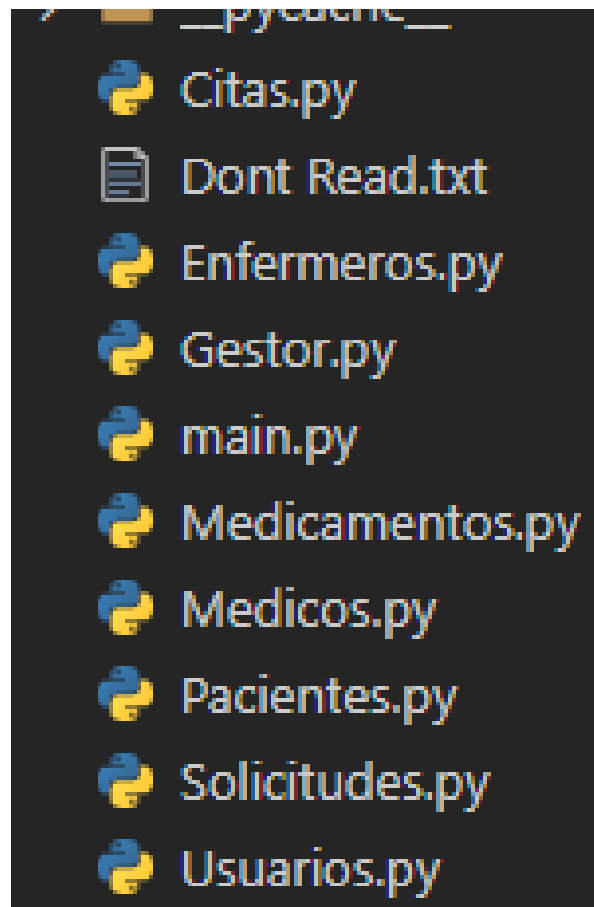
Usuarios: esta clase se encarga de instanciar el objeto "Usuario" que será utilizado posteriormente en distintos ámbitos

Procedimientos

Main.py:

Esta clase se encarga de conectar el BackEnd con el FrontEnd y redireccionarlos al “Gestor.py” que es la clase que se encarga de todas las funcionalidades del sistema

Gestor.py: esta clase se encarga de realizar todos los procedimientos que el FronEnd necesite para su buen funcionamiento



Funciones

Es el intermediario entre la Vista y el Modelo, se encarga de controlar las interacciones del usuario en la vista, contienen el código necesario para responder a las acciones que se solicitan en la aplicación.

MAIN.PY

```
BackEnd > main.py > ...
158 #CITAS
159 @app.route('/obtenercitas')
160 def obtener_citas():
161     return gestor.obtener_citas()
162 #
163
164 #MEDICAMENTOS
165 @app.route('/medicamentos',methods=['POST'])
166 def crearMedicamento():
167     datom = request.json
168     gestor.crearMedicamento(datom['nombremed'],datom['descripcionmed'],datom['preciomed'],datom['cantidadmed'])
169     return '{"Estado":"Medicamento Creado"}'
170
171 @app.route('/obtenermedicamentos')
172 def obtener_medicamentos():
173     return gestor.obtener_medicamentos()
174
175 @app.route('/medicamentos/<nombremed>',methods=['PUT'])
176 def actualizarmedicamento(nombremed):
177     datom = request.json
178     if gestor.actualizar_medicamento(nombremed,datom['descripcionmed'],datom['preciomed'],datom['cantidadmed']):
179         return '{"data":"Actualizado"}'
180     return '{"data":"Error"}'
181
182 @app.route('/cargamedicamentos',methods=['POST'])
183 def cargamed():
184     datoe = request.json
185     gestor.cargamasivamed(datoe['data'])
186     return '{"data":"Cargados"}'
187
```

GESTOR.PY

```
BackEnd > Gestor.py > Gestor > registrar_medimento
123     self.medicos.remove(x)
124     return True
125     return False
126
127 def eliminar_enfermero(self,nombre,autor):
128     for x in self.enfermeros:
129         if x.nombree==nombre and x.usere == autor:
130             self.enfermeros.remove(x)
131             return True
132     return False
133
134 def eliminar_solicitud(self,nombre,autor):
135     for x in self.solicitudes:
136         if x.fechac==nombre and x.horac==autor:
137             self.solicitudes.remove(x)
138             return True
139     return False
140
141
142 def eliminar_medicamentos(self,nombre,autor):
143     for x in self.medicamentos:
144         if x.nombremed==nombre and x.preciomed==autor:
145             self.medicamentos.remove(x)
146             return True
147     return False
148
149 #Iniciar Sesion
150 def iniciar_sesion(self,user,password):
151     for x in self.usuarios:
152         if (x.password==password and x.user==user.lower()):
153             return json.dumps(x.__dict__)
154     else:
```

FLUJO DEL PROGRAMA

Se ingresa al sistema, el programa espera a recibir los datos correspondientes a usuario y contraseña, luego de ello ejecuta la opción según los datos ingresados, verifica el rol que ingresa al sistema y muestra su contenido correspondiente

