DOCUMENTACIÓN DETALLADA DE CÓDIGO

# MEDISYNC - Ubicaciones y Líneas Específicas

Generado el: 12/08/2025 16:48  
Análisis línea por línea del código fuente

# MEDISYNC.py - Aplicación Principal

Análisis detallado del archivo: MEDISYNC.py

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Función/Clase | Tipo | Líneas | Descripción | Ubicación |
| MedisyncApp: | Class | 49-51 | def \_\_init\_\_(self, root=None): self.db\_manager = DBManager('database/medisync.db') self.current\_use... | MEDISYNC.py:49 |
| \_\_init\_\_ | Function | 52-61 | Sin documentación | MEDISYNC.py:52 |
| setup\_root\_window | Function | 62-74 | if self.root: self.root.title("MEDISYNC - Sistema de Gestión Médica") self.root.geometry("500x400") ... | MEDISYNC.py:62 |
| create\_login\_window | Function | 75-220 | if self.root is None: self.root = tk.Tk() self.root.title("MEDISYNC - Sistema de Gestión Médica") s... | MEDISYNC.py:75 |
| show\_patient\_registration | Function | 221-230 | try: from patient\_registration\_form import create\_patient\_registration\_form create\_patient\_registrat... | MEDISYNC.py:221 |
| center\_window | Function | 231-238 | screen\_width = window.winfo\_screenwidth() screen\_height = window.winfo\_screenheight() x = (screen\_wi... | MEDISYNC.py:231 |
| login | Function | 239-257 | email = self.email\_entry.get().strip() password = self.password\_entry.get().strip() if not email or... | MEDISYNC.py:239 |
| create\_main\_window | Function | 258-274 | self.root = tk.Tk() self.root.title(f"MEDISYNC - {self.current\_user.nombre} {self.current\_user.apell... | MEDISYNC.py:258 |
| toggle\_fullscreen | Function | 275-283 | if self.root.state() == 'zoomed': self.root.state('normal') self.root.geometry("1200x800") self.cent... | MEDISYNC.py:275 |
| create\_menu\_interface | Function | 284-322 | # Frame principal main\_frame = tk.Frame(self.root, bg='#F8FAFC') main\_frame.pack(fill='both', expand... | MEDISYNC.py:284 |
| create\_admin\_menu | Function | 323-396 | # Frame principal para el menú menu\_container = tk.Frame(parent, bg='#F8FAFC') menu\_container.pack(f... | MEDISYNC.py:323 |
| on\_enter | Function | 397-400 | Sin documentación | MEDISYNC.py:397 |
| on\_leave | Function | 401-416 | Sin documentación | MEDISYNC.py:401 |
| switch\_tab | Function | 417-446 | # Actualizar variable de pestaña activa self.active\_tab.set(tab\_name) # Actualizar colores de boton... | MEDISYNC.py:417 |
| create\_dashboard\_tab | Function | 447-550 | # Frame principal con mejor diseño main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill=... | MEDISYNC.py:447 |
| create\_modern\_stats\_card | Function | 551-576 | # Frame de la tarjeta card = tk.Frame(parent, bg='white', relief='solid', bd=1) card.pack(side='left... | MEDISYNC.py:551 |
| create\_stats\_card | Function | 577-589 | card\_frame = tk.Frame(parent, bg=color, width=200, height=100, relief='raised', bd=2) card\_frame.gri... | MEDISYNC.py:577 |
| get\_system\_stats | Function | 590-632 | conn = self.db\_manager.get\_connection() cursor = conn.cursor() try: stats = {} # Total de usuarios... | MEDISYNC.py:590 |
| create\_users\_tab | Function | 633-829 | # Frame principal con mejor diseño main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill=... | MEDISYNC.py:633 |
| load\_users\_data | Function | 830-849 | self.load\_users\_list() # def create\_patients\_tab(self, parent): | MEDISYNC.py:830 |
| create\_appointments\_tab | Function | 850-1078 | # Frame principal main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=Tr... | MEDISYNC.py:850 |
| get\_doctors\_list | Function | 1079-1098 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() | MEDISYNC.py:1079 |
| load\_appointments\_data | Function | 1099-1135 | try: # Limpiar tabla for item in tree.get\_children(): tree.delete(item) # Obtener citas con filtros... | MEDISYNC.py:1099 |
| clear\_appointment\_filters | Function | 1136-1149 | self.search\_var.set("") self.status\_filter.set("Todos") if hasattr(self, 'date\_filter'): self.date\_f... | MEDISYNC.py:1136 |
| get\_filtered\_appointments | Function | 1150-1198 | try: appointments = self.db\_manager.get\_all\_appointments() # Filtrar por texto de búsqueda search\_t... | MEDISYNC.py:1150 |
| filter\_appointments | Function | 1199-1203 | if hasattr(self, 'appointments\_tree'): self.load\_appointments\_data(self.appointments\_tree) def on\_a... | MEDISYNC.py:1199 |
| on\_appointment\_select | Function | 1204-1223 | try: selection = self.appointments\_tree.selection() if not selection: self.clear\_appointment\_details... | MEDISYNC.py:1204 |
| show\_appointment\_details | Function | 1224-1289 | try: # Limpiar detalles anteriores for widget in self.appointment\_details\_frame.winfo\_children(): wi... | MEDISYNC.py:1224 |
| clear\_appointment\_details | Function | 1290-1300 | try: for widget in self.appointment\_details\_frame.winfo\_children(): widget.destroy() tk.Label(self.... | MEDISYNC.py:1290 |
| new\_appointment\_window | Function | 1301-1638 | window = tk.Toplevel(self.root) window.title("🏥 MEDISYNC - Nueva Cita Médica") window.geometry("850x... | MEDISYNC.py:1301 |
| \_on\_mousewheel | Function | 1639-1649 | Sin documentación | MEDISYNC.py:1639 |
| select\_date | Function | 1650-1675 | from datetime import date import calendar date\_window = tk.Toplevel(self.root) date\_window.title("S... | MEDISYNC.py:1650 |
| update\_calendar | Function | 1676-1713 | Sin documentación | MEDISYNC.py:1676 |
| select\_day | Function | 1714-1718 | Sin documentación | MEDISYNC.py:1714 |
| prev\_month | Function | 1719-1726 | Sin documentación | MEDISYNC.py:1719 |
| next\_month | Function | 1727-1750 | Sin documentación | MEDISYNC.py:1727 |
| save\_appointment | Function | 1751-1829 | try: # Validar campos requeridos required\_fields = ['fecha', 'hora', 'paciente\_id', 'doctor\_id', 'mo... | MEDISYNC.py:1751 |
| edit\_appointment | Function | 1830-1851 | try: selection = self.appointments\_tree.selection() if not selection: messagebox.showwarning("Advert... | MEDISYNC.py:1830 |
| edit\_appointment\_window | Function | 1852-2050 | window = tk.Toplevel(self.root) window.title("✏️ Editar Cita Médica") window.geometry("600x700") win... | MEDISYNC.py:1852 |
| update\_appointment | Function | 2051-2130 | try: # Validar campos requeridos required\_fields = ['fecha', 'hora', 'paciente\_id', 'doctor\_id', 'mo... | MEDISYNC.py:2051 |
| view\_appointment\_details | Function | 2131-2392 | try: if appointment\_id is None: selection = self.appointments\_tree.selection() if not selection: mes... | MEDISYNC.py:2131 |
| \_on\_mousewheel | Function | 2393-2400 | try: # Validar transición de estado current\_appointment = self.db\_manager.get\_appointment\_by\_id(appo... | MEDISYNC.py:2393 |
| change\_appointment\_status | Function | 2401-2473 | try: # Validar transición de estado current\_appointment = self.db\_manager.get\_appointment\_by\_id(appo... | MEDISYNC.py:2401 |
| cancel\_appointment\_with\_reason | Function | 2474-2571 | try: # Crear ventana para motivo de cancelación reason\_window = tk.Toplevel(self.root) reason\_window... | MEDISYNC.py:2474 |
| confirm\_cancellation | Function | 2572-2620 | Sin documentación | MEDISYNC.py:2572 |
| edit\_appointment\_from\_details | Function | 2621-2637 | try: appointment = self.db\_manager.get\_appointment\_by\_id(appointment\_id) if not appointment: message... | MEDISYNC.py:2621 |
| refresh\_appointment\_details | Function | 2638-2646 | try: if details\_window and details\_window.winfo\_exists(): details\_window.destroy() self.view\_appoint... | MEDISYNC.py:2638 |
| print\_appointment\_details | Function | 2647-2707 | try: appointment = self.db\_manager.get\_appointment\_by\_id(appointment\_id) if not appointment: message... | MEDISYNC.py:2647 |
| generate\_appointment\_pdf | Function | 2708-2727 | try: appointment = self.db\_manager.get\_appointment\_by\_id(appointment\_id) if not appointment: message... | MEDISYNC.py:2708 |
| copy\_appointment\_to\_clipboard | Function | 2728-2750 | try: appointment = self.db\_manager.get\_appointment\_by\_id(appointment\_id) if not appointment: message... | MEDISYNC.py:2728 |
| email\_appointment\_details | Function | 2785-2790 | messagebox.showinfo("Email", f"Enviando detalles de cita #{appointment\_id} por email...\n" "Funciona... | MEDISYNC.py:2785 |
| log\_appointment\_change | Function | 2791-2804 | try: timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S') user\_name = f"{self.current\_user.nombr... | MEDISYNC.py:2791 |
| confirm\_appointment | Function | 2805-2830 | try: # Verificar si estamos en la vista de doctor if hasattr(self, 'doctor\_appointments\_tree'): sele... | MEDISYNC.py:2805 |
| cancel\_appointment | Function | 2831-2856 | try: # Verificar si estamos en la vista de doctor if hasattr(self, 'doctor\_appointments\_tree'): sele... | MEDISYNC.py:2831 |
| complete\_appointment | Function | 2857-2886 | try: # Verificar si estamos en la vista de doctor if hasattr(self, 'doctor\_appointments\_tree'): sele... | MEDISYNC.py:2857 |
| start\_appointment | Function | 2887-2912 | try: # Verificar si estamos en la vista de doctor if hasattr(self, 'doctor\_appointments\_tree'): sele... | MEDISYNC.py:2887 |
| update\_appointment\_status | Function | 2913-2937 | try: selection = self.appointments\_tree.selection() if not selection: messagebox.showwarning("Advert... | MEDISYNC.py:2913 |
| print\_appointment | Function | 2938-2946 | try: # Aquí puedes implementar la funcionalidad de impresión # Por ahora, mostraremos un mensaje mes... | MEDISYNC.py:2938 |
| create\_medical\_history\_tab | Function | 2947-3129 | # Limpiar contenido previo for widget in parent.winfo\_children(): widget.destroy() # Frame principa... | MEDISYNC.py:2947 |
| ensure\_test\_patients | Function | 3130-3173 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() # Verificar si hay pacientes cu... | MEDISYNC.py:3130 |
| create\_sample\_medical\_records | Function | 3174-3256 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() # Obtener IDs de pacientes y do... | MEDISYNC.py:3174 |
| create\_advanced\_billing\_tab | Function | 3257-3305 | # Inicializar variables de facturación integrada self.init\_integrated\_billing\_system() # Frame prin... | MEDISYNC.py:3257 |
| auto\_launch\_billing\_system | Function | 3306-3310 | # Esta función ya no es necesaria con el sistema integrado pass def manual\_launch\_billing\_system(se... | MEDISYNC.py:3306 |
| manual\_launch\_billing\_system | Function | 3311-3315 | # Esta función ya no es necesaria con el sistema integrado pass def init\_integrated\_billing\_system(... | MEDISYNC.py:3311 |
| init\_integrated\_billing\_system | Function | 3316-3352 | # Variables para el sistema de facturación self.billing\_notebook = None self.appointments\_tree\_billi... | MEDISYNC.py:3316 |
| create\_integrated\_billing\_content | Function | 3353-3546 | # Frame principal con scroll main\_container = tk.Frame(parent, bg='#F8FAFC') main\_container.pack(fil... | MEDISYNC.py:3353 |
| \_on\_mousewheel | Function | 3547-3551 | calc\_card = tk.Frame(parent, bg='white', relief='solid', bd=2) calc\_card.pack(fill='x', pady=(0, 15)... | MEDISYNC.py:3547 |
| create\_modern\_billing\_calculations\_panel | Function | 3552-3678 | calc\_card = tk.Frame(parent, bg='white', relief='solid', bd=2) calc\_card.pack(fill='x', pady=(0, 15)... | MEDISYNC.py:3552 |
| create\_billing\_calculations\_panel | Function | 3679-3752 | calc\_frame = tk.LabelFrame(parent, text="💰 CÁLCULOS", bg='#fff3e0', fg='#e65100', font=('Arial', 10,... | MEDISYNC.py:3679 |
| create\_integrated\_services\_content | Function | 3753-3803 | main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=True, padx=20, pady=... | MEDISYNC.py:3753 |
| create\_integrated\_reports\_content | Function | 3804-3875 | main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=True, padx=20, pady=... | MEDISYNC.py:3804 |
| create\_billing\_status\_bar | Function | 3876-3892 | status\_frame = tk.Frame(parent, bg='#0B5394', height=30) status\_frame.pack(fill='x', side='bottom') ... | MEDISYNC.py:3876 |
| load\_integrated\_billing\_data | Function | 3893-3901 | try: self.load\_appointments\_for\_billing() self.load\_services\_for\_billing() self.update\_status("Datos... | MEDISYNC.py:3893 |
| load\_appointments\_for\_billing | Function | 3902-3993 | print("🔄 DEBUG: Ejecutando load\_appointments\_for\_billing()") if not hasattr(self, 'appointments\_card... | MEDISYNC.py:3902 |
| create\_appointment\_card | Function | 3994-4061 | # Frame de la tarjeta card = tk.Frame(self.appointments\_cards\_frame, bg='white', relief='solid', bd=... | MEDISYNC.py:3994 |
| on\_enter | Function | 4062-4076 | Sin documentación | MEDISYNC.py:4062 |
| on\_leave | Function | 4077-4097 | Sin documentación | MEDISYNC.py:4077 |
| select\_appointment\_card | Function | 4098-4125 | # Desmarcar todas las tarjetas for widget in self.appointments\_cards\_frame.winfo\_children(): if hasa... | MEDISYNC.py:4098 |
| load\_services\_for\_billing | Function | 4126-4170 | if not hasattr(self, 'services\_grid\_frame'): return try: # Limpiar tarjetas anteriores for widget i... | MEDISYNC.py:4126 |
| create\_service\_card | Function | 4171-4227 | # Frame de la tarjeta card = tk.Frame(self.services\_grid\_frame, bg='white', relief='solid', bd=1) ca... | MEDISYNC.py:4171 |
| on\_enter | Function | 4228-4231 | Sin documentación | MEDISYNC.py:4228 |
| on\_leave | Function | 4232-4240 | if not self.current\_appointment\_billing: messagebox.showwarning("Cita", "Por favor seleccione una ci... | MEDISYNC.py:4232 |
| add\_service\_from\_card | Function | 4241-4267 | if not self.current\_appointment\_billing: messagebox.showwarning("Cita", "Por favor seleccione una ci... | MEDISYNC.py:4241 |
| update\_selected\_services\_display | Function | 4268-4302 | if not hasattr(self, 'selected\_services\_frame'): return # Limpiar tarjetas anteriores for widget in... | MEDISYNC.py:4268 |
| create\_selected\_service\_card | Function | 4303-4352 | # Frame de la tarjeta card = tk.Frame(self.selected\_services\_frame, bg='white', relief='solid', bd=1... | MEDISYNC.py:4303 |
| remove\_service\_by\_index | Function | 4353-4362 | try: if 0 <= index < len(self.selected\_services): self.selected\_services.pop(index) self.update\_sele... | MEDISYNC.py:4353 |
| filter\_services | Function | 4363-4408 | search\_term = self.service\_search\_var.get().lower() if not hasattr(self, 'services\_grid\_frame'): re... | MEDISYNC.py:4363 |
| filter\_appointments\_billing | Function | 4409-4412 | self.load\_appointments\_for\_billing() def on\_appointment\_select\_for\_billing(self, event=None): | MEDISYNC.py:4409 |
| on\_appointment\_select\_for\_billing | Function | 4413-4417 | # Esta función ahora es manejada por select\_appointment\_card pass def add\_service\_to\_invoice(self): | MEDISYNC.py:4413 |
| add\_service\_to\_invoice | Function | 4418-4422 | # Esta función ahora es manejada por add\_service\_from\_card messagebox.showinfo("Información", "Use l... | MEDISYNC.py:4418 |
| update\_invoice\_services\_display | Function | 4423-4427 | # Esta función ahora es manejada por update\_selected\_services\_display self.update\_selected\_services\_... | MEDISYNC.py:4423 |
| remove\_service\_from\_invoice | Function | 4428-4485 | if not self.selected\_services: messagebox.showwarning("Sin servicios", "No hay servicios para elimin... | MEDISYNC.py:4428 |
| confirm\_delete | Function | 4486-4498 | Sin documentación | MEDISYNC.py:4486 |
| edit\_service\_in\_invoice | Function | 4499-4522 | if not hasattr(self, 'invoice\_services\_tree') or not self.invoice\_services\_tree: return selection =... | MEDISYNC.py:4499 |
| calculate\_totals | Function | 4523-4542 | subtotal = sum(service['total'] for service in self.selected\_services) try: discount = float(self.d... | MEDISYNC.py:4523 |
| apply\_discount | Function | 4543-4546 | self.calculate\_totals() def calculate\_change(self, event=None): | MEDISYNC.py:4543 |
| calculate\_change | Function | 4547-4573 | try: total\_str = self.total\_var.get().replace('RD$', '').replace(',', '') total = float(total\_str) ... | MEDISYNC.py:4547 |
| generate\_invoice\_pdf\_integrated | Function | 4574-4605 | if not self.current\_appointment\_billing: messagebox.showwarning("Cita", "Por favor seleccione una ci... | MEDISYNC.py:4574 |
| generate\_simple\_pdf | Function | 4606-4615 | try: total\_str = self.total\_var.get().replace('RD$', '').replace(',', '') total = float(total\_str) ... | MEDISYNC.py:4606 |
| save\_invoice\_integrated | Function | 4676-4726 | if not self.current\_appointment\_billing: messagebox.showwarning("Cita", "Por favor seleccione una ci... | MEDISYNC.py:4676 |
| clear\_invoice\_form | Function | 4727-4742 | self.selected\_services = [] self.current\_appointment\_billing = None self.subtotal\_var.set("0.00") s... | MEDISYNC.py:4727 |
| update\_status | Function | 4743-4749 | if hasattr(self, 'status\_label'): self.status\_label.config(text=message) # ==================== FUN... | MEDISYNC.py:4743 |
| load\_services\_management | Function | 4750-4767 | if not hasattr(self, 'services\_management\_tree'): return # Limpiar tabla for item in self.services\_... | MEDISYNC.py:4750 |
| add\_new\_service | Function | 4768-4810 | # Crear ventana de diálogo dialog = tk.Toplevel(self.root) dialog.title("Nuevo Servicio Médico") dia... | MEDISYNC.py:4768 |
| save\_service | Function | 4811-4852 | Sin documentación | MEDISYNC.py:4811 |
| edit\_selected\_service | Function | 4853-4921 | if not hasattr(self, 'services\_management\_tree'): return selection = self.services\_management\_tree.... | MEDISYNC.py:4853 |
| update\_service | Function | 4922-4955 | Sin documentación | MEDISYNC.py:4922 |
| delete\_selected\_service | Function | 4956-4991 | if not hasattr(self, 'services\_management\_tree'): return selection = self.services\_management\_tree.... | MEDISYNC.py:4956 |
| get\_billing\_statistics\_integrated | Function | 4992-5027 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() today = datetime.now().strftime... | MEDISYNC.py:4992 |
| generate\_daily\_report\_integrated | Function | 5028-5052 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() today = datetime.now().strftime... | MEDISYNC.py:5028 |
| generate\_monthly\_report\_integrated | Function | 5098-5101 | messagebox.showinfo("Reporte Mensual", "Función de reporte mensual en desarrollo") def generate\_pen... | MEDISYNC.py:5098 |
| generate\_pending\_report\_integrated | Function | 5102-5105 | messagebox.showinfo("Facturas Pendientes", "Función de reporte de pendientes en desarrollo") def ge... | MEDISYNC.py:5102 |
| generate\_services\_report\_integrated | Function | 5106-5109 | messagebox.showinfo("Ingresos por Servicio", "Función de reporte por servicios en desarrollo") def ... | MEDISYNC.py:5106 |
| create\_billing\_main\_tab | Function | 5110-5113 | pass def create\_billing\_stats\_panel(self, parent): | MEDISYNC.py:5110 |
| create\_billing\_stats\_panel | Function | 5114-5243 | # Panel de estadísticas stats\_frame = tk.LabelFrame( parent, text="📈 ESTADÍSTICAS HOY", font=('Arial... | MEDISYNC.py:5114 |
| create\_complete\_billing\_interface | Function | 5244-5288 | # Header moderno header\_frame = tk.Frame(parent, bg='#1E3A8A', height=80) header\_frame.pack(fill='x'... | MEDISYNC.py:5244 |
| create\_advanced\_billing\_tab | Function | 5289-5341 | # Panel izquierdo - Citas pendientes left\_panel = tk.Frame(parent, bg='white', relief='solid', bd=1)... | MEDISYNC.py:5289 |
| scroll\_appointments | Function | 5342-5346 | Sin documentación | MEDISYNC.py:5342 |
| on\_appointments\_focus\_in | Function | 5347-5401 | Sin documentación | MEDISYNC.py:5347 |
| scroll\_invoices | Function | 5402-5406 | Sin documentación | MEDISYNC.py:5402 |
| on\_invoices\_focus\_in | Function | 5407-5422 | Sin documentación | MEDISYNC.py:5407 |
| debug\_table | Function | 5423-5462 | Sin documentación | MEDISYNC.py:5423 |
| load\_existing\_invoices | Function | 5463-5577 | try: print("🔄 Iniciando carga de facturas existentes...") # Verificar que el widget existe if not h... | MEDISYNC.py:5463 |
| create\_invoice\_form | Function | 5578-5597 | # Crear un canvas con scrollbar para el formulario completo canvas = tk.Canvas(parent, bg='white', h... | MEDISYNC.py:5578 |
| configure\_scroll\_region | Function | 5598-5896 | Sin documentación | MEDISYNC.py:5598 |
| \_on\_mousewheel | Function | 5897-5903 | try: # Limpiar tabla for item in self.billing\_appointments\_tree.get\_children(): self.billing\_appoint... | MEDISYNC.py:5897 |
| load\_pending\_appointments | Function | 5904-5949 | try: # Limpiar tabla for item in self.billing\_appointments\_tree.get\_children(): self.billing\_appoint... | MEDISYNC.py:5904 |
| load\_default\_services | Function | 5950-5988 | # Servicios comunes con precios default\_services = [ ("Consulta General", 2500.00), ("Consulta Espec... | MEDISYNC.py:5950 |
| on\_appointment\_select\_billing | Function | 5989-6051 | try: selection = self.billing\_appointments\_tree.selection() if not selection: print("DEBUG: No hay s... | MEDISYNC.py:5989 |
| get\_doctor\_billing\_info | Function | 6085-6137 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() # Buscar información del doctor | MEDISYNC.py:6085 |
| get\_patient\_insurance\_info | Function | 6138-6187 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() # Buscar información del pacien... | MEDISYNC.py:6138 |
| load\_doctor\_services | Function | 6188-6267 | # Limpiar servicios actuales for item in self.services\_tree.get\_children(): self.services\_tree.delet... | MEDISYNC.py:6188 |
| update\_doctor\_info\_panel | Function | 6268-6270 | Sin documentación | MEDISYNC.py:6268 |
| update\_insurance\_info\_panel | Function | 6278-6282 | if patient\_info['seguro'] and doctor\_acepta\_seguros: # Paciente tiene seguro y doctor acepta | MEDISYNC.py:6278 |
| update\_doctor\_panel\_info | Function | 6313-6368 | if not hasattr(self, 'doctor\_name\_label'): print("DEBUG: No se encontró doctor\_name\_label") return ... | MEDISYNC.py:6313 |
| update\_modern\_insurance\_info | Function | 6369-6485 | if not hasattr(self, 'insurance\_main\_label'): print("DEBUG: No se encontró insurance\_main\_label") re... | MEDISYNC.py:6369 |
| add\_service\_to\_invoice | Function | 6486-6611 | # Verificar si hay una cita seleccionada O una factura seleccionada appointment\_selected = hasattr(... | MEDISYNC.py:6486 |
| update\_services | Function | 6612-6626 | category = category\_var.get() if category in services\_data: services = [f"{name} - RD$ {price:,.2f}"... | MEDISYNC.py:6612 |
| update\_price | Function | 6627-6636 | nonlocal current\_price service\_key = service\_var.get() if service\_key in current\_services: name, pri... | MEDISYNC.py:6627 |
| update\_total | Function | 6637-6656 | quantity = quantity\_var.get() total = current\_price \* quantity total\_label.config(text=f"Total: RD$ ... | MEDISYNC.py:6637 |
| add\_service | Function | 6657-6768 | Sin documentación | MEDISYNC.py:6657 |
| remove\_service | Function | 6769-6791 | Sin documentación | MEDISYNC.py:6769 |
| add\_custom\_service | Function | 6792-6817 | custom\_dialog = tk.Toplevel(dialog) custom\_dialog.title("➕ Servicio Personalizado") custom\_dialog.ge... | MEDISYNC.py:6792 |
| save\_custom | Function | 6818-6866 | Sin documentación | MEDISYNC.py:6818 |
| calculate\_totals | Function | 6867-6925 | subtotal = 0.0 # Sumar todos los servicios for item in self.services\_tree.get\_children(): values = ... | MEDISYNC.py:6867 |
| generate\_unique\_invoice\_number | Function | 6926-6975 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() fecha\_actual = datetime.now() y... | MEDISYNC.py:6926 |
| create\_invoice\_from\_appointment | Function | 6976-7078 | if not hasattr(self, 'selected\_appointment\_id'): messagebox.showwarning("Selección requerida", "Sele... | MEDISYNC.py:6976 |
| clear\_invoice\_form | Function | 7079-7102 | # Limpiar servicios for item in self.services\_tree.get\_children(): self.services\_tree.delete(item) ... | MEDISYNC.py:7079 |
| remove\_service\_from\_invoice | Function | 7103-7111 | selection = self.services\_tree.selection() if selection: self.services\_tree.delete(selection[0]) sel... | MEDISYNC.py:7103 |
| edit\_service\_price | Function | 7112-7131 | selection = self.services\_tree.selection() if not selection: messagebox.showwarning("Selección reque... | MEDISYNC.py:7112 |
| generate\_invoice\_pdf | Function | 7132-7431 | if not hasattr(self, 'selected\_appointment\_id') or not self.selected\_appointment\_id: messagebox.show... | MEDISYNC.py:7132 |
| open\_payment\_window | Function | 7432-7456 | if not hasattr(self, 'selected\_appointment\_id') or not self.selected\_appointment\_id: messagebox.show... | MEDISYNC.py:7432 |
| create\_payment\_window | Function | 7457-7665 | # Ventana principal payment\_window = tk.Toplevel(self.root) payment\_window.title("💳 Procesamiento de... | MEDISYNC.py:7457 |
| calculate\_change | Function | 7666-7697 | Sin documentación | MEDISYNC.py:7666 |
| process\_payment\_and\_generate | Function | 7698-7746 | Sin documentación | MEDISYNC.py:7698 |
| create\_invoice\_in\_database | Function | 7747-7841 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() # Generar número de factura úni... | MEDISYNC.py:7747 |
| generate\_final\_invoice\_pdf | Function | 7842-8081 | try: # Verificar que reportlab esté disponible try: from reportlab.lib.pagesizes import A4 from repo... | MEDISYNC.py:7842 |
| process\_payment\_window | Function | 8082-8147 | # Verificar que la tabla existe if not hasattr(self, 'billing\_invoices\_tree') or not self.billing\_in... | MEDISYNC.py:8082 |
| create\_existing\_invoice\_payment\_window | Function | 8148-8295 | # Ventana principal payment\_window = tk.Toplevel(self.root) payment\_window.title("💳 Procesar Pago - ... | MEDISYNC.py:8148 |
| calculate\_change | Function | 8296-8327 | Sin documentación | MEDISYNC.py:8296 |
| process\_existing\_payment | Function | 8328-8382 | Sin documentación | MEDISYNC.py:8328 |
| view\_invoice\_details\_billing | Function | 8383-8411 | selection = self.billing\_invoices\_tree.selection() if not selection: messagebox.showwarning("Selecci... | MEDISYNC.py:8383 |
| reprint\_invoice\_pdf | Function | 8433-8466 | selection = self.billing\_invoices\_tree.selection() if not selection: messagebox.showwarning("Selecci... | MEDISYNC.py:8433 |
| install\_reportlab | Function | 8467-8482 | try: import subprocess import sys result = messagebox.askquestion("Instalar ReportLab", "¿Desea ins... | MEDISYNC.py:8467 |
| export\_billing\_report\_pdf | Function | 8483-8544 | try: from tkinter import filedialog from reportlab.lib.pagesizes import letter from reportlab.platyp... | MEDISYNC.py:8483 |
| refresh\_billing\_data | Function | 8545-8548 | pass def create\_payments\_tab(self, parent): | MEDISYNC.py:8545 |
| create\_payments\_tab | Function | 8549-8555 | tk.Label(parent, text="💳 Módulo de Pagos", font=('Arial', 16, 'bold'), bg='#F8FAFC').pack(pady=20) t... | MEDISYNC.py:8549 |
| create\_reports\_tab | Function | 8556-8562 | tk.Label(parent, text="📊 Módulo de Reportes", font=('Arial', 16, 'bold'), bg='#F8FAFC').pack(pady=20... | MEDISYNC.py:8556 |
| create\_billing\_config\_tab | Function | 8563-8569 | tk.Label(parent, text="⚙️ Configuración de Facturación", font=('Arial', 16, 'bold'), bg='#F8FAFC').p... | MEDISYNC.py:8563 |
| load\_billing\_data\_integrated | Function | 8570-8626 | try: # Limpiar tabla for item in self.billing\_tree.get\_children(): self.billing\_tree.delete(item) #... | MEDISYNC.py:8570 |
| update\_billing\_statistics | Function | 8627-8663 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() today = datetime.now().strftime... | MEDISYNC.py:8627 |
| update\_pending\_appointments | Function | 8664-8703 | try: self.pending\_appointments\_list.delete(0, tk.END) conn = self.db\_manager.get\_connection() curso... | MEDISYNC.py:8664 |
| quick\_invoice | Function | 8704-8707 | messagebox.showinfo("Factura Rápida", "Use el 'Sistema Completo' para crear facturas con todas las f... | MEDISYNC.py:8704 |
| generate\_billing\_report | Function | 8708-8719 | try: if not PDF\_AVAILABLE: messagebox.showwarning("PDF No Disponible", "La funcionalidad de reportes... | MEDISYNC.py:8708 |
| search\_invoice | Function | 8720-8726 | search\_term = simpledialog.askstring("Buscar Factura", "Ingrese número de factura o nombre del pacie... | MEDISYNC.py:8720 |
| refresh\_billing\_data | Function | 8727-8731 | self.load\_billing\_data\_integrated() messagebox.showinfo("Actualizado", "✅ Datos de facturación actua... | MEDISYNC.py:8727 |
| bill\_selected\_appointment | Function | 8732-8740 | selection = self.pending\_appointments\_list.curselection() if not selection: messagebox.showwarning("... | MEDISYNC.py:8732 |
| show\_billing\_summary | Function | 8741-8764 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() # Obtener datos del mes actual ... | MEDISYNC.py:8741 |
| load\_billing\_data | Function | 8779-8809 | try: # Limpiar tabla for item in tree.get\_children(): tree.delete(item) # Obtener facturas invoices... | MEDISYNC.py:8779 |
| create\_reports\_tab | Function | 8810-8879 | # Frame principal main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=Tr... | MEDISYNC.py:8810 |
| on\_enter | Function | 8880-8881 | Sin documentación | MEDISYNC.py:8880 |
| on\_leave | Function | 8882-8982 | Sin documentación | MEDISYNC.py:8882 |
| create\_stats\_display | Function | 8983-9017 | try: stats = self.get\_system\_stats() # Grid de estadísticas stats\_grid = tk.Frame(parent, bg='#e8f5... | MEDISYNC.py:8983 |
| create\_report\_filters | Function | 9018-9064 | # Filtro de fechas date\_frame = tk.Frame(parent, bg='#fce4ec') date\_frame.pack(fill='x', pady=5) tk... | MEDISYNC.py:9018 |
| darken\_color | Function | 9065-9071 | color = color.lstrip('#') rgb = tuple(int(color[i:i+2], 16) for i in (0, 2, 4)) darker\_rgb = tuple(m... | MEDISYNC.py:9065 |
| generate\_income\_report | Function | 9072-9075 | self.show\_report\_dialog("Reporte de Ingresos", "income") def generate\_pending\_invoices\_report(self)... | MEDISYNC.py:9072 |
| generate\_pending\_invoices\_report | Function | 9076-9079 | self.show\_report\_dialog("Facturas Pendientes", "pending\_invoices") def generate\_users\_report(self): | MEDISYNC.py:9076 |
| generate\_users\_report | Function | 9080-9083 | self.show\_report\_dialog("Estadísticas de Usuarios", "users") def generate\_appointments\_report(self)... | MEDISYNC.py:9080 |
| generate\_appointments\_report | Function | 9084-9087 | self.show\_report\_dialog("Reporte de Citas", "appointments") def generate\_financial\_report(self): | MEDISYNC.py:9084 |
| generate\_financial\_report | Function | 9088-9091 | self.show\_report\_dialog("Estado Financiero", "financial") def generate\_services\_report(self): | MEDISYNC.py:9088 |
| generate\_services\_report | Function | 9092-9095 | self.show\_report\_dialog("Reporte de Servicios", "services") def show\_report\_dialog(self, report\_tit... | MEDISYNC.py:9092 |
| show\_report\_dialog | Function | 9096-9177 | dialog = tk.Toplevel(self.root) dialog.title(f"Configurar {report\_title}") dialog.geometry("500x600"... | MEDISYNC.py:9096 |
| on\_period\_change | Function | 9178-9233 | Sin documentación | MEDISYNC.py:9178 |
| generate\_report | Function | 9234-9259 | # Calcular fechas según el período seleccionado start\_date, end\_date = self.calculate\_date\_range(per... | MEDISYNC.py:9234 |
| preview\_report | Function | 9260-9291 | start\_date, end\_date = self.calculate\_date\_range(period\_var.get(), start\_date\_var.get(), end\_date\_va... | MEDISYNC.py:9260 |
| calculate\_date\_range | Function | 9292-9335 | today = datetime.now() if period == 'Hoy': start\_date = today.replace(hour=0, minute=0, second=0, m... | MEDISYNC.py:9292 |
| select\_date\_calendar | Function | 9336-9354 | if not CALENDAR\_AVAILABLE: return try: from tkinter import simpledialog date\_str = simpledialog.ask... | MEDISYNC.py:9336 |
| show\_report\_preview | Function | 9355-9419 | preview\_window = tk.Toplevel(self.root) preview\_window.title(f"Vista Previa - {config['title']}") pr... | MEDISYNC.py:9355 |
| \_on\_mousewheel | Function | 9420-9423 | try: # Header del reporte self.create\_report\_header(parent, config) # Separador tk.Frame(parent, bg... | MEDISYNC.py:9420 |
| generate\_report\_content | Function | 9424-9457 | try: # Header del reporte self.create\_report\_header(parent, config) # Separador tk.Frame(parent, bg... | MEDISYNC.py:9424 |
| create\_report\_header | Function | 9458-9485 | header\_frame = tk.Frame(parent, bg='white') header\_frame.pack(fill='x', pady=(0, 20)) # Logo y títu... | MEDISYNC.py:9458 |
| create\_report\_summary | Function | 9486-9516 | summary\_frame = tk.LabelFrame(parent, text="📊 Resumen Ejecutivo", font=('Arial', 14, 'bold'), padx=2... | MEDISYNC.py:9486 |
| create\_income\_report\_content | Function | 9517-9600 | content\_frame = tk.LabelFrame(parent, text="💰 Análisis de Ingresos", font=('Arial', 14, 'bold'), pad... | MEDISYNC.py:9517 |
| create\_appointments\_report\_content | Function | 9601-9674 | content\_frame = tk.LabelFrame(parent, text="📅 Análisis de Citas", font=('Arial', 14, 'bold'), padx=2... | MEDISYNC.py:9601 |
| create\_users\_report\_content | Function | 9675-9721 | content\_frame = tk.LabelFrame(parent, text="👥 Análisis de Usuarios", font=('Arial', 14, 'bold'), pad... | MEDISYNC.py:9675 |
| create\_pending\_invoices\_content | Function | 9722-9797 | content\_frame = tk.LabelFrame(parent, text="📋 Facturas Pendientes", font=('Arial', 14, 'bold'), padx... | MEDISYNC.py:9722 |
| create\_financial\_report\_content | Function | 9798-9936 | content\_frame = tk.LabelFrame(parent, text="💰 Estado Financiero", font=('Arial', 14, 'bold'), padx=2... | MEDISYNC.py:9798 |
| create\_services\_report\_content | Function | 9937-10106 | content\_frame = tk.LabelFrame(parent, text="🏥 Análisis de Servicios", font=('Arial', 14, 'bold'), pa... | MEDISYNC.py:9937 |
| create\_report\_footer | Function | 10107-10116 | footer\_frame = tk.Frame(parent, bg='white') footer\_frame.pack(fill='x', pady=(20, 0)) # Separador t... | MEDISYNC.py:10107 |
| get\_report\_summary\_data | Function | 10128-10185 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() summary\_data = [] if config['r... | MEDISYNC.py:10128 |
| generate\_pdf\_from\_preview | Function | 10186-10338 | try: # Verificar si reportlab está disponible try: from reportlab.lib.pagesizes import letter, A4 fr... | MEDISYNC.py:10186 |
| generate\_excel\_from\_preview | Function | 10339-10346 | try: messagebox.showinfo("Excel", "Función de exportación a Excel en desarrollo.\nPor ahora use la f... | MEDISYNC.py:10339 |
| get\_report\_data\_for\_pdf | Function | 10347-10428 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() if config['report\_type'] == 'in... | MEDISYNC.py:10347 |
| install\_reportlab\_for\_reports | Function | 10429-10444 | try: import subprocess import sys result = messagebox.askquestion("Instalar ReportLab", "¿Desea ins... | MEDISYNC.py:10429 |
| generate\_report\_file | Function | 10445-10454 | if config['output\_format'] == 'PDF': self.generate\_pdf\_from\_preview(config) elif config['output\_form... | MEDISYNC.py:10445 |
| apply\_report\_filters | Function | 10455-10461 | period = self.report\_period\_filter.get() report\_type = self.report\_type\_filter.get() messagebox.sho... | MEDISYNC.py:10455 |
| export\_all\_reports\_pdf | Function | 10462-10465 | messagebox.showinfo("Exportar", "Exportando todos los reportes disponibles...") def export\_to\_excel... | MEDISYNC.py:10462 |
| export\_to\_excel | Function | 10466-10469 | messagebox.showinfo("Excel", "Función de exportación a Excel en desarrollo") def email\_reports(self... | MEDISYNC.py:10466 |
| email\_reports | Function | 10470-10473 | messagebox.showinfo("Email", "Función de envío por email en desarrollo") def show\_executive\_dashboa... | MEDISYNC.py:10470 |
| show\_executive\_dashboard | Function | 10474-10477 | messagebox.showinfo("Dashboard", "Dashboard ejecutivo en desarrollo") def refresh\_all\_data(self): | MEDISYNC.py:10474 |
| refresh\_all\_data | Function | 10478-10482 | self.create\_stats\_display(self.stats\_frame if hasattr(self, 'stats\_frame') else None) messagebox.sho... | MEDISYNC.py:10478 |
| configure\_reports | Function | 10483-10486 | messagebox.showinfo("Configuración", "Configuración de reportes en desarrollo") def manage\_report\_t... | MEDISYNC.py:10483 |
| manage\_report\_templates | Function | 10487-10490 | messagebox.showinfo("Plantillas", "Gestión de plantillas en desarrollo") def create\_doctor\_menu(sel... | MEDISYNC.py:10487 |
| create\_doctor\_menu | Function | 10491-10565 | # Frame principal para el menú menu\_container = tk.Frame(parent, bg='#F8FAFC') menu\_container.pack(f... | MEDISYNC.py:10491 |
| switch\_doctor\_tab | Function | 10566-10593 | # Actualizar variable de pestaña activa self.active\_doctor\_tab.set(tab\_name) # Actualizar colores d... | MEDISYNC.py:10566 |
| create\_secretaria\_menu | Function | 10594-10668 | # Frame principal para el menú menu\_container = tk.Frame(parent, bg='#F8FAFC') menu\_container.pack(f... | MEDISYNC.py:10594 |
| switch\_secretaria\_tab | Function | 10669-10696 | # Actualizar variable de pestaña activa self.active\_secretaria\_tab.set(tab\_name) # Actualizar color... | MEDISYNC.py:10669 |
| create\_paciente\_menu | Function | 10697-10771 | # Frame principal para el menú menu\_container = tk.Frame(parent, bg='#F8FAFC') menu\_container.pack(f... | MEDISYNC.py:10697 |
| switch\_patient\_tab | Function | 10772-10799 | # Actualizar variable de pestaña activa self.active\_patient\_tab.set(tab\_name) # Actualizar colores ... | MEDISYNC.py:10772 |
| logout | Function | 10800-10808 | self.current\_user = None self.root.destroy() self.\_\_init\_\_() # ==================== FUNCIONES PARA ... | MEDISYNC.py:10800 |
| create\_doctor\_dashboard | Function | 10809-10998 | main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=True, padx=20, pady=... | MEDISYNC.py:10809 |
| on\_enter | Function | 10999-11000 | Sin documentación | MEDISYNC.py:10999 |
| on\_leave | Function | 11001-11065 | Sin documentación | MEDISYNC.py:11001 |
| create\_doctor\_stats\_card | Function | 11066-11099 | card\_frame = tk.Frame(parent, bg=card\_data['bg\_color'], relief='solid', bd=1) card\_frame.pack(side='... | MEDISYNC.py:11066 |
| create\_appointment\_card | Function | 11100-11142 | card\_frame = tk.Frame(parent, bg='#F9FAFB', relief='solid', bd=1) card\_frame.pack(fill='x', pady=3) ... | MEDISYNC.py:11100 |
| darken\_color | Function | 11143-11154 | # Mapeo simple de colores para efectos hover color\_map = { '#10B981': '#059669', '#3B82F6': '#2563EB... | MEDISYNC.py:11143 |
| create\_doctor\_appointments | Function | 11155-11355 | main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=True, padx=20, pady=... | MEDISYNC.py:11155 |
| on\_appointment\_select | Function | 11356-11366 | try: selection = self.doctor\_appointments\_tree.selection() if selection: # Solo mostrar que hay una ... | MEDISYNC.py:11356 |
| on\_appointment\_double\_click | Function | 11367-11386 | try: selection = self.doctor\_appointments\_tree.selection() if not selection: return item = selectio... | MEDISYNC.py:11367 |
| create\_appointment\_context\_menu | Function | 11387-11421 | self.appointment\_context\_menu = tk.Menu(self.root, tearoff=0) # Agregar opciones al menú con iconos... | MEDISYNC.py:11387 |
| show\_context\_menu | Function | 11422-11440 | try: # Identificar la fila donde se hizo clic derecho item = self.doctor\_appointments\_tree.identify\_... | MEDISYNC.py:11422 |
| update\_context\_menu\_state | Function | 11441-11481 | try: # Índices de las opciones en el menú # 0: Ver Detalles, 1: Separador, 2: Iniciar, 3: Confirmar,... | MEDISYNC.py:11441 |
| context\_view\_details | Function | 11482-11494 | try: selection = self.doctor\_appointments\_tree.selection() if selection: item = selection[0] values ... | MEDISYNC.py:11482 |
| context\_start\_appointment | Function | 11495-11503 | try: selection = self.doctor\_appointments\_tree.selection() if selection: self.start\_appointment() ex... | MEDISYNC.py:11495 |
| context\_confirm\_appointment | Function | 11504-11512 | try: selection = self.doctor\_appointments\_tree.selection() if selection: self.confirm\_appointment() ... | MEDISYNC.py:11504 |
| context\_complete\_appointment | Function | 11513-11541 | try: selection = self.doctor\_appointments\_tree.selection() if not selection: messagebox.showwarning(... | MEDISYNC.py:11513 |
| context\_cancel\_appointment | Function | 11542-11570 | try: selection = self.doctor\_appointments\_tree.selection() if not selection: messagebox.showwarning(... | MEDISYNC.py:11542 |
| context\_confirm\_appointment | Function | 11571-11599 | try: selection = self.doctor\_appointments\_tree.selection() if not selection: messagebox.showwarning(... | MEDISYNC.py:11571 |
| context\_start\_appointment | Function | 11600-11628 | try: selection = self.doctor\_appointments\_tree.selection() if not selection: messagebox.showwarning(... | MEDISYNC.py:11600 |
| show\_appointment\_details\_window | Function | 11629-11795 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() # Obtener datos completos de la... | MEDISYNC.py:11629 |
| print\_appointment\_details | Function | 11796-11799 | messagebox.showinfo("Imprimir", "Función de impresión en desarrollo") def create\_doctor\_patients(se... | MEDISYNC.py:11796 |
| create\_doctor\_patients | Function | 11800-11993 | main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=True, padx=20, pady=... | MEDISYNC.py:11800 |
| on\_enter | Function | 11994-11995 | Sin documentación | MEDISYNC.py:11994 |
| on\_leave | Function | 11996-12010 | Sin documentación | MEDISYNC.py:11996 |
| create\_patient\_context\_menu | Function | 12011-12033 | self.patient\_context\_menu = tk.Menu(self.root, tearoff=0) # Opciones del menú self.patient\_context\_... | MEDISYNC.py:12011 |
| show\_patient\_context\_menu | Function | 12034-12052 | try: # Identificar la fila donde se hizo clic derecho item = self.doctor\_patients\_tree.identify\_row(... | MEDISYNC.py:12034 |
| context\_view\_patient\_profile | Function | 12053-12072 | try: selection = self.doctor\_patients\_tree.selection() if not selection: messagebox.showwarning("Adv... | MEDISYNC.py:12053 |
| context\_view\_medical\_history | Function | 12073-12092 | try: selection = self.doctor\_patients\_tree.selection() if not selection: messagebox.showwarning("Adv... | MEDISYNC.py:12073 |
| context\_schedule\_appointment | Function | 12093-12112 | try: selection = self.doctor\_patients\_tree.selection() if not selection: messagebox.showwarning("Adv... | MEDISYNC.py:12093 |
| filter\_patients | Function | 12113-12119 | try: self.load\_doctor\_patients() except Exception as e: print(f"Error filtrando pacientes: {e}") de... | MEDISYNC.py:12113 |
| refresh\_patients | Function | 12120-12127 | try: self.load\_doctor\_patients() messagebox.showinfo("Actualizado", "Lista de pacientes actualizada"... | MEDISYNC.py:12120 |
| show\_patient\_stats | Function | 12128-12134 | try: messagebox.showinfo("Estadísticas", "Función de estadísticas en desarrollo") except Exception a... | MEDISYNC.py:12128 |
| create\_medical\_records | Function | 12135-12213 | main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=True, padx=20, pady=... | MEDISYNC.py:12135 |
| create\_doctor\_profile | Function | 12214-12279 | main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=True, padx=20, pady=... | MEDISYNC.py:12214 |
| create\_secretaria\_dashboard | Function | 12296-12397 | main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=True, padx=20, pady=... | MEDISYNC.py:12296 |
| create\_secretaria\_appointments | Function | 12398-12492 | main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=True, padx=20, pady=... | MEDISYNC.py:12398 |
| create\_secretaria\_patients | Function | 12493-12581 | main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=True, padx=20, pady=... | MEDISYNC.py:12493 |
| create\_secretaria\_billing | Function | 12582-12814 | main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=True, padx=15, pady=... | MEDISYNC.py:12582 |
| get\_billing\_summary\_secretaria | Function | 12815-12864 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() today = datetime.now().strftime... | MEDISYNC.py:12815 |
| load\_secretaria\_billing\_data | Function | 12865-12936 | try: # Limpiar listas self.secretaria\_pending\_list.delete(0, tk.END) for item in self.secretaria\_inv... | MEDISYNC.py:12865 |
| create\_new\_invoice\_secretaria | Function | 12937-12942 | messagebox.showinfo("Nueva Factura", "Use el 'Sistema Completo' para crear facturas con todas las fu... | MEDISYNC.py:12937 |
| process\_payment\_secretaria | Function | 12943-12948 | messagebox.showinfo("Procesar Pago", "Use el 'Sistema Completo' para procesar pagos con:\n\n" + "• M... | MEDISYNC.py:12943 |
| express\_invoice | Function | 12949-12952 | messagebox.showinfo("Factura Express", "Funcionalidad disponible en Sistema Completo") def search\_p... | MEDISYNC.py:12949 |
| search\_patient\_billing | Function | 12953-12958 | search\_term = simpledialog.askstring("Buscar Paciente", "Ingrese nombre del paciente:") if search\_te... | MEDISYNC.py:12953 |
| generate\_daily\_report | Function | 12959-12962 | messagebox.showinfo("Reporte Diario", "Generación de reportes PDF disponible en Sistema Completo") ... | MEDISYNC.py:12959 |
| refresh\_billing\_secretaria | Function | 12963-12967 | self.load\_secretaria\_billing\_data() messagebox.showinfo("Actualizado", "✅ Datos actualizados correct... | MEDISYNC.py:12963 |
| bill\_appointment\_secretaria | Function | 12968-12978 | selection = self.secretaria\_pending\_list.curselection() if not selection: messagebox.showwarning("Se... | MEDISYNC.py:12968 |
| filter\_invoices\_secretaria | Function | 12979-12987 | # Implementación básica de filtrado filter\_text = self.filter\_var.get().lower() status\_filter = self... | MEDISYNC.py:12979 |
| view\_invoice\_details | Function | 12988-13000 | selection = self.secretaria\_invoices\_tree.selection() if not selection: return item = self.secretar... | MEDISYNC.py:12988 |
| create\_secretaria\_reports | Function | 13001-13064 | main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=True, padx=20, pady=... | MEDISYNC.py:13001 |
| create\_patient\_dashboard | Function | 13104-13306 | # Frame principal con mejor diseño main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill=... | MEDISYNC.py:13104 |
| create\_patient\_appointments | Function | 13307-13379 | # Frame principal con mejor diseño main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill=... | MEDISYNC.py:13307 |
| on\_cal\_select | Function | 13380-13543 | Sin documentación | MEDISYNC.py:13380 |
| on\_enter | Function | 13544-13546 | Sin documentación | MEDISYNC.py:13544 |
| on\_leave | Function | 13547-13560 | Sin documentación | MEDISYNC.py:13547 |
| on\_enter | Function | 13561-13563 | Sin documentación | MEDISYNC.py:13561 |
| on\_leave | Function | 13564-13579 | Sin documentación | MEDISYNC.py:13564 |
| create\_tooltip | Function | 13580-13581 | def show\_tooltip(event): tooltip = tk.Toplevel() tooltip.wm\_overrideredirect(True) tooltip.wm\_geomet... | MEDISYNC.py:13580 |
| show\_tooltip | Function | 13582-13590 | Sin documentación | MEDISYNC.py:13582 |
| hide\_tooltip | Function | 13591-13596 | Sin documentación | MEDISYNC.py:13591 |
| hide\_tooltip | Function | 13597-13603 | try: sel = self.patient\_appointments\_tree.selection() if not sel: self.selected\_appointment\_info.pac... | MEDISYNC.py:13597 |
| on\_patient\_appointment\_select | Function | 13604-13656 | try: sel = self.patient\_appointments\_tree.selection() if not sel: self.selected\_appointment\_info.pac... | MEDISYNC.py:13604 |
| show\_appointment\_context\_menu | Function | 13657-13704 | try: # Seleccionar el item bajo el cursor item = self.patient\_appointments\_tree.identify\_row(event.y... | MEDISYNC.py:13657 |
| create\_patient\_medical\_history | Function | 13705-13816 | # Frame principal main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=Tr... | MEDISYNC.py:13705 |
| load\_patient\_doctors | Function | 13817-13858 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() | MEDISYNC.py:13817 |
| get\_patient\_health\_summary | Function | 13859-13960 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() summary = {} patient\_id = self.... | MEDISYNC.py:13859 |
| filter\_medical\_history | Function | 13961-14066 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() # Query base | MEDISYNC.py:13961 |
| clear\_history\_filters | Function | 14067-14084 | try: self.history\_period\_filter.set('Todos') self.history\_doctor\_filter.set('Todos') if hasattr(sel... | MEDISYNC.py:14067 |
| on\_history\_select | Function | 14085-14095 | try: selection = self.patient\_history\_tree.selection() if selection: # Guardar referencia del regist... | MEDISYNC.py:14085 |
| check\_empty\_history | Function | 14096-14130 | try: if not self.patient\_history\_tree.get\_children(): # Crear frame de mensaje de historial vacío em... | MEDISYNC.py:14096 |
| show\_tooltip | Function | 14131-14152 | try: self.hide\_tooltip() # Ocultar tooltip anterior si existe self.tooltip\_window = tk.Toplevel() ... | MEDISYNC.py:14131 |
| hide\_tooltip | Function | 14153-14161 | try: if hasattr(self, 'tooltip\_window') and self.tooltip\_window: self.tooltip\_window.destroy() self.... | MEDISYNC.py:14153 |
| show\_detailed\_stats | Function | 14162-14192 | try: stats\_window = tk.Toplevel(self.root) stats\_window.title("📊 Estadísticas Detalladas de Salud") ... | MEDISYNC.py:14162 |
| get\_detailed\_health\_stats | Function | 14229-14302 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() stats = {} # Consultas por per... | MEDISYNC.py:14229 |
| analyze\_health\_trends | Function | 14303-14312 | messagebox.showinfo("Análisis de Tendencias", "📈 Función de análisis de tendencias en desarrollo.\n\... | MEDISYNC.py:14303 |
| refresh\_history | Function | 14313-14320 | try: self.load\_patient\_medical\_history() messagebox.showinfo("Actualización", "✅ Historial médico ac... | MEDISYNC.py:14313 |
| generate\_summary\_report | Function | 14321-14330 | messagebox.showinfo("Resumen Ejecutivo", "📄 Función de resumen ejecutivo en desarrollo.\n\n" + "El r... | MEDISYNC.py:14321 |
| show\_health\_charts | Function | 14331-14340 | messagebox.showinfo("Gráficos de Salud", "📈 Función de gráficos interactivos en desarrollo.\n\n" + "... | MEDISYNC.py:14331 |
| configure\_health\_alerts | Function | 14341-14350 | messagebox.showinfo("Alertas Médicas", "⚙️ Función de alertas personalizadas en desarrollo.\n\n" + "... | MEDISYNC.py:14341 |
| export\_history\_pdf | Function | 14351-14373 | try: if not PDF\_AVAILABLE: messagebox.showerror("Error", "ReportLab no está instalado. No se pueden ... | MEDISYNC.py:14351 |
| print\_my\_history | Function | 14374-14383 | messagebox.showinfo("Imprimir", "🖨️ Función de impresión en desarrollo.\n\n" + "Próximamente podrás:... | MEDISYNC.py:14374 |
| email\_my\_history | Function | 14384-14393 | messagebox.showinfo("Envío por Email", "📧 Función de envío por email en desarrollo.\n\n" + "Próximam... | MEDISYNC.py:14384 |
| view\_history\_detail | Function | 14394-14527 | try: # Verificar si hay un registro seleccionado selection = self.patient\_history\_tree.selection() i... | MEDISYNC.py:14394 |
| print\_single\_record | Function | 14528-14533 | messagebox.showinfo("Imprimir Registro", "🖨️ Función de impresión de registro específico en desarrol... | MEDISYNC.py:14528 |
| export\_single\_record\_pdf | Function | 14534-14548 | try: fecha = record\_data.get('fecha', 'N/A') doctor = f"Dr. {record\_data.get('doctor\_nombre', '')} {... | MEDISYNC.py:14534 |
| print\_single\_record | Function | 14549-14563 | try: fecha = record\_data.get('fecha', 'N/A') doctor = f"Dr. {record\_data.get('doctor\_nombre', '')} {... | MEDISYNC.py:14549 |
| email\_single\_record | Function | 14564-14581 | try: fecha = record\_data.get('fecha', 'N/A') doctor = f"Dr. {record\_data.get('doctor\_nombre', '')} {... | MEDISYNC.py:14564 |
| copy\_record\_to\_clipboard | Function | 14582-14592 | try: if not record\_data: messagebox.showwarning("Advertencia", "No hay datos para copiar") return #... | MEDISYNC.py:14582 |
| generate\_medical\_history\_pdf | Function | 14646-14752 | try: if not PDF\_AVAILABLE: raise Exception("ReportLab no está disponible") from reportlab.lib.pages... | MEDISYNC.py:14646 |
| create\_patient\_settings | Function | 14753-14955 | # Frame principal main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=Tr... | MEDISYNC.py:14753 |
| save\_patient\_settings | Function | 14956-14994 | try: if not self.current\_user: messagebox.showerror('Error', 'No hay usuario activo') return # Vali... | MEDISYNC.py:14956 |
| change\_patient\_password | Function | 14995-15055 | print("🔧 DEBUG: Iniciando función change\_patient\_password") dialog = tk.Toplevel(self.root) dialog.... | MEDISYNC.py:14995 |
| create\_password\_field | Function | 15056-15092 | Sin documentación | MEDISYNC.py:15056 |
| save\_password | Function | 15093-15175 | Sin documentación | MEDISYNC.py:15093 |
| open\_patient\_schedule\_dialog | Function | 15176-15295 | schedule\_window = tk.Toplevel(self.root) schedule\_window.title("🗓️ Agendar Nueva Cita Médica") sched... | MEDISYNC.py:15176 |
| get\_selected\_date | Function | 15296-15393 | Sin documentación | MEDISYNC.py:15296 |
| on\_doctor\_change | Function | 15394-15414 | Sin documentación | MEDISYNC.py:15394 |
| save\_appointment | Function | 15415-15493 | Sin documentación | MEDISYNC.py:15415 |
| update\_patient\_appointments\_list | Function | 15494-15506 | try: self.load\_patient\_appointments() # Mostrar mensaje de confirmación total\_citas = len(self.patie... | MEDISYNC.py:15494 |
| refresh\_patient\_appointments | Function | 15507-15513 | try: self.load\_patient\_appointments() except Exception as e: print(f"Error refrescando citas: {e}") ... | MEDISYNC.py:15507 |
| get\_appointment\_status\_info | Function | 15514-15548 | status\_info = { 'pendiente': { 'color': '#F59E0B', 'icon': '⏳', 'description': 'Pendiente de confirm... | MEDISYNC.py:15514 |
| format\_appointment\_time | Function | 15549-15587 | try: if not fecha\_str or not hora\_str: return "Sin programar", "" # Convertir a datetime fecha\_obj ... | MEDISYNC.py:15549 |
| get\_patient\_appointment\_summary | Function | 15588-15644 | try: if not self.current\_user or not hasattr(self.current\_user, 'id'): return None conn = self.db\_m... | MEDISYNC.py:15588 |
| load\_patient\_appointments | Function | 15645-15791 | try: if not hasattr(self, 'patient\_appointments\_tree'): return # Limpiar tabla for i in self.patient... | MEDISYNC.py:15645 |
| get\_appointment\_status\_info | Function | 15792-15807 | status\_map = { 'pendiente': {'icon': '⏳', 'color': '#F39C12', 'description': 'Esperando confirmación... | MEDISYNC.py:15792 |
| format\_appointment\_time | Function | 15808-15855 | try: if not fecha\_str or not hora\_str: return "Sin fecha", "Sin hora", "" # Convertir a datetime dt... | MEDISYNC.py:15808 |
| get\_patient\_appointment\_summary | Function | 15856-15888 | try: if not hasattr(self, 'patient\_appointments\_tree'): return None summary = { 'total': 0, 'pendie... | MEDISYNC.py:15856 |
| view\_my\_appointment\_details | Function | 15889-16050 | try: sel = self.patient\_appointments\_tree.selection() if not sel: messagebox.showwarning("Selección"... | MEDISYNC.py:15889 |
| cancel\_appointment\_from\_details | Function | 16051-16065 | try: if messagebox.askyesno('Confirmar Cancelación', '¿Está seguro que desea cancelar esta cita?\n\n... | MEDISYNC.py:16051 |
| edit\_my\_appointment | Function | 16066-16345 | try: sel = self.patient\_appointments\_tree.selection() if not sel: messagebox.showwarning("Selección"... | MEDISYNC.py:16066 |
| save\_changes | Function | 16346-16475 | Sin documentación | MEDISYNC.py:16346 |
| cancel\_my\_appointment | Function | 16476-16658 | try: sel = self.patient\_appointments\_tree.selection() if not sel: messagebox.showwarning("Selección"... | MEDISYNC.py:16476 |
| confirm\_cancellation | Function | 16659-16740 | Sin documentación | MEDISYNC.py:16659 |
| create\_no\_data\_placeholder | Function | 16741-16772 | try: # Limpiar frame for widget in self.scrollable\_history\_frame.winfo\_children(): widget.destroy() ... | MEDISYNC.py:16741 |
| create\_history\_card | Function | 16773-16953 | try: # Frame principal de la carta con sombra visual card\_frame = tk.Frame(parent, bg='white', relie... | MEDISYNC.py:16773 |
| on\_card\_click | Function | 16954-16967 | Sin documentación | MEDISYNC.py:16954 |
| view\_history\_detail\_card | Function | 16968-17212 | try: # Crear ventana de detalles detail\_window = tk.Toplevel(self.root) detail\_window.title(f"Detall... | MEDISYNC.py:16968 |
| \_on\_mousewheel | Function | 17213-17220 | try: # Limpiar tabla for item in self.medical\_history\_tree.get\_children(): self.medical\_history\_tree... | MEDISYNC.py:17213 |
| load\_patient\_medical\_history | Function | 17221-17310 | try: # Limpiar tabla for item in self.medical\_history\_tree.get\_children(): self.medical\_history\_tree... | MEDISYNC.py:17221 |
| filter\_medical\_history | Function | 17311-17314 | self.load\_patient\_medical\_history() def create\_patient\_billing(self, parent): | MEDISYNC.py:17311 |
| create\_patient\_billing | Function | 17315-17455 | # Frame principal main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=Tr... | MEDISYNC.py:17315 |
| create\_billing\_status\_cards | Function | 17456-17506 | try: # Obtener estadísticas de facturas billing\_stats = self.get\_billing\_statistics() # Configuraci... | MEDISYNC.py:17456 |
| create\_status\_card | Function | 17507-17533 | # Frame de la tarjeta card\_frame = tk.Frame(parent, bg=config['bg\_color'], relief='solid', bd=1, hei... | MEDISYNC.py:17507 |
| get\_billing\_statistics | Function | 17534-17571 | try: if not self.current\_user: return {'pendientes': 0, 'pagadas': 0, 'vencidas': 0, 'total\_monto': ... | MEDISYNC.py:17534 |
| get\_patient\_billing\_summary | Function | 17572-17624 | try: if not self.current\_user: return {'total\_paid': 0, 'total\_pending': 0, 'total\_invoices': 0, 'mo... | MEDISYNC.py:17572 |
| load\_patient\_billing\_data | Function | 17625-17730 | try: if not self.current\_user or not hasattr(self, 'patient\_billing\_tree'): return # Limpiar tabla ... | MEDISYNC.py:17625 |
| refresh\_patient\_billing | Function | 17731-17735 | self.load\_patient\_billing\_data() messagebox.showinfo("Actualizado", "✅ Mis facturas han sido actuali... | MEDISYNC.py:17731 |
| download\_patient\_invoice | Function | 17736-17749 | selection = self.patient\_invoices\_tree.selection() if not selection: messagebox.showwarning("Selecci... | MEDISYNC.py:17736 |
| view\_payment\_history | Function | 17750-17756 | messagebox.showinfo("Historial de Pagos", "Su historial completo de pagos incluye:\n\n" + "• Fechas ... | MEDISYNC.py:17750 |
| request\_account\_statement | Function | 17757-17763 | messagebox.showinfo("Estado de Cuenta", "Su solicitud de estado de cuenta ha sido registrada.\n\n" +... | MEDISYNC.py:17757 |
| billing\_inquiry | Function | 17764-17772 | messagebox.showinfo("Consultas de Facturación", "Para consultas sobre facturación puede:\n\n" + "• C... | MEDISYNC.py:17764 |
| filter\_patient\_billing | Function | 17773-17782 | try: period = self.billing\_period\_filter.get() status = self.billing\_status\_filter.get() self.load\_p... | MEDISYNC.py:17773 |
| clear\_billing\_filters | Function | 17783-17791 | try: self.billing\_period\_filter.set('Todos') self.billing\_status\_filter.set('Todos') self.load\_patie... | MEDISYNC.py:17783 |
| view\_billing\_detail | Function | 17792-17804 | try: selection = self.patient\_billing\_tree.selection() if not selection: messagebox.showwarning("Sel... | MEDISYNC.py:17792 |
| download\_billing\_pdf | Function | 17820-17837 | try: selection = self.patient\_billing\_tree.selection() if not selection: messagebox.showwarning("Sel... | MEDISYNC.py:17820 |
| email\_my\_billing | Function | 17838-17856 | try: selection = self.patient\_billing\_tree.selection() if not selection: messagebox.showwarning("Sel... | MEDISYNC.py:17838 |
| view\_billing\_detail\_simple | Function | 17857-17925 | try: selection = self.patient\_billing\_tree.selection() if not selection: messagebox.showwarning("Adv... | MEDISYNC.py:17857 |
| show\_billing\_summary | Function | 17926-17932 | try: summary = self.get\_patient\_billing\_summary() | MEDISYNC.py:17926 |
| filter\_patient\_invoices | Function | 17945-17949 | # Implementación básica de filtrado para pacientes self.load\_patient\_billing\_data() # Por simplicid... | MEDISYNC.py:17945 |
| view\_patient\_invoice\_details | Function | 17950-17960 | selection = self.patient\_invoices\_tree.selection() if not selection: return item = self.patient\_inv... | MEDISYNC.py:17950 |
| create\_patient\_profile | Function | 17973-18076 | main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=True, padx=20, pady=... | MEDISYNC.py:17973 |
| get\_doctor\_stats | Function | 18094-18157 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() stats = {} doctor\_id = self.cur... | MEDISYNC.py:18094 |
| get\_doctor\_upcoming\_appointments | Function | 18158-18193 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() doctor\_id = self.current\_user.i... | MEDISYNC.py:18158 |
| get\_secretaria\_stats | Function | 18194-18229 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() stats = {} today = datetime.now... | MEDISYNC.py:18194 |
| get\_today\_appointments | Function | 18230-18262 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() today = datetime.now().strftime... | MEDISYNC.py:18230 |
| get\_patient\_info | Function | 18263-18324 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() patient\_id = self.current\_user.... | MEDISYNC.py:18263 |
| get\_patient\_stats | Function | 18325-18373 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() patient\_id = self.current\_user.i... | MEDISYNC.py:18325 |
| get\_patient\_info | Function | 18374-18414 | try: conn = sqlite3.connect('database/medisync.db') conn.row\_factory = sqlite3.Row cursor = conn.cur... | MEDISYNC.py:18374 |
| view\_history\_detail | Function | 18415-18628 | try: if not hasattr(self, 'patient\_history\_tree'): return sel = self.patient\_history\_tree.selection... | MEDISYNC.py:18415 |
| print\_my\_history | Function | 18629-18632 | messagebox.showinfo('Historial', 'Impresión del historial próximamente disponible') def export\_hist... | MEDISYNC.py:18629 |
| export\_history\_pdf | Function | 18633-18636 | messagebox.showinfo('Historial', 'Exportación a PDF próximamente disponible') def email\_my\_history(... | MEDISYNC.py:18633 |
| email\_my\_history | Function | 18637-18640 | messagebox.showinfo('Historial', 'Envío por email próximamente disponible') def get\_patient\_upcomin... | MEDISYNC.py:18637 |
| get\_patient\_upcoming\_appointments | Function | 18641-18677 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() patient\_id = self.current\_user.... | MEDISYNC.py:18641 |
| get\_patient\_reminders | Function | 18678-18708 | # Esta función puede expandirse para incluir recordatorios específicos # Por ahora devolvemos record... | MEDISYNC.py:18678 |
| new\_appointment\_quick | Function | 18709-18712 | self.new\_appointment\_window() def new\_patient\_quick(self): messagebox.showinfo("Acción", "Funcional... | MEDISYNC.py:18709 |
| new\_patient\_quick | Function | 18713-18715 | self.switch\_tab("Facturación Avanzada") messagebox.showinfo("Sistema de Facturación", "🚀 Accediendo ... | MEDISYNC.py:18713 |
| process\_payment\_quick | Function | 18716-18725 | self.switch\_tab("Facturación Avanzada") messagebox.showinfo("Sistema de Facturación", "🚀 Accediendo ... | MEDISYNC.py:18716 |
| generate\_invoice\_quick | Function | 18726-18736 | self.switch\_tab("Facturación Avanzada") messagebox.showinfo("Sistema de Facturación", "🚀 Accediendo ... | MEDISYNC.py:18726 |
| manage\_waiting\_list | Function | 18737-18739 | try: # Limpiar tabla for item in self.doctor\_appointments\_tree.get\_children(): self.doctor\_appointme... | MEDISYNC.py:18737 |
| daily\_report | Function | 18740-18742 | try: # Limpiar tabla for item in self.doctor\_appointments\_tree.get\_children(): self.doctor\_appointme... | MEDISYNC.py:18740 |
| load\_doctor\_appointments | Function | 18743-18808 | try: # Limpiar tabla for item in self.doctor\_appointments\_tree.get\_children(): self.doctor\_appointme... | MEDISYNC.py:18743 |
| filter\_doctor\_appointments | Function | 18809-18812 | self.load\_filtered\_appointments() def apply\_appointment\_filters(self): | MEDISYNC.py:18809 |
| apply\_appointment\_filters | Function | 18813-18871 | try: # Obtener valores de filtros period\_filter = self.appointment\_filter.get() status\_filter = self... | MEDISYNC.py:18813 |
| search\_appointments\_by\_patient | Function | 18872-18875 | self.apply\_appointment\_filters() def on\_appointment\_select(self, event): | MEDISYNC.py:18872 |
| on\_appointment\_select | Function | 18876-18885 | try: selection = self.doctor\_appointments\_tree.selection() if selection: # Panel de detalles elimina... | MEDISYNC.py:18876 |
| show\_appointment\_details | Function | 18886-18997 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() # Obtener datos completos de la... | MEDISYNC.py:18886 |
| refresh\_appointments | Function | 18998-19002 | self.load\_doctor\_appointments() messagebox.showinfo("Actualizado", "Lista de citas actualizada corre... | MEDISYNC.py:18998 |
| schedule\_new\_appointment | Function | 19003-19006 | messagebox.showinfo("Función", "Función de nueva cita - implementar según necesidades") def show\_ap... | MEDISYNC.py:19003 |
| show\_appointment\_stats | Function | 19007-19065 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() # Estadísticas básicas stats\_qu... | MEDISYNC.py:19007 |
| edit\_appointment | Function | 19066-19074 | selection = self.doctor\_appointments\_tree.selection() if not selection: messagebox.showwarning("Sele... | MEDISYNC.py:19066 |
| print\_appointment | Function | 19075-19083 | selection = self.doctor\_appointments\_tree.selection() if not selection: messagebox.showwarning("Sele... | MEDISYNC.py:19075 |
| load\_doctor\_patients | Function | 19084-19127 | try: # Limpiar tabla for item in self.doctor\_patients\_tree.get\_children(): self.doctor\_patients\_tree... | MEDISYNC.py:19084 |
| load\_medical\_patients | Function | 19128-19156 | try: # Limpiar listbox self.medical\_patients\_listbox.delete(0, tk.END) doctor\_id = self.current\_use... | MEDISYNC.py:19128 |
| view\_appointment\_details | Function | 19157-19236 | try: selection = self.doctor\_appointments\_tree.selection() if not selection: messagebox.showwarning(... | MEDISYNC.py:19157 |
| complete\_appointment | Function | 19237-19269 | try: appointment\_id = self.get\_appointment\_id\_from\_selection() if not appointment\_id: messagebox.sho... | MEDISYNC.py:19237 |
| cancel\_appointment | Function | 19270-19325 | try: appointment\_id = self.get\_appointment\_id\_from\_selection() if not appointment\_id: messagebox.sho... | MEDISYNC.py:19270 |
| confirm\_cancel | Function | 19326-19349 | Sin documentación | MEDISYNC.py:19326 |
| add\_appointment\_notes | Function | 19350-19399 | try: appointment\_id = self.get\_appointment\_id\_from\_selection() if not appointment\_id: messagebox.sho... | MEDISYNC.py:19350 |
| save\_notes | Function | 19400-19423 | Sin documentación | MEDISYNC.py:19400 |
| search\_patients | Function | 19424-19474 | try: search\_term = self.patient\_search\_entry.get().lower() # Limpiar tabla for item in self.doctor\_... | MEDISYNC.py:19424 |
| view\_patient\_profile | Function | 19475-19616 | try: selection = self.doctor\_patients\_tree.selection() if not selection: messagebox.showwarning("Adv... | MEDISYNC.py:19475 |
| view\_medical\_history | Function | 19617-19736 | try: selection = self.doctor\_patients\_tree.selection() if not selection: messagebox.showwarning("Adv... | MEDISYNC.py:19617 |
| schedule\_appointment | Function | 19737-19771 | try: selection = self.doctor\_patients\_tree.selection() if not selection: messagebox.showwarning("Adv... | MEDISYNC.py:19737 |
| load\_patient\_medical\_records | Function | 19772-19823 | try: selection = self.medical\_patients\_listbox.curselection() if not selection: return # Obtener ID... | MEDISYNC.py:19772 |
| create\_medical\_record | Function | 19824-19915 | try: selection = self.medical\_patients\_listbox.curselection() if not selection: messagebox.showwarni... | MEDISYNC.py:19824 |
| save\_record | Function | 19916-19962 | Sin documentación | MEDISYNC.py:19916 |
| view\_medical\_record\_detail | Function | 19963-20062 | try: selection = self.medical\_records\_tree.selection() if not selection: messagebox.showwarning("Adv... | MEDISYNC.py:19963 |
| edit\_medical\_record | Function | 20063-20075 | try: selection = self.medical\_records\_tree.selection() if not selection: messagebox.showwarning("Adv... | MEDISYNC.py:20063 |
| print\_medical\_record | Function | 20076-20088 | try: selection = self.medical\_records\_tree.selection() if not selection: messagebox.showwarning("Adv... | MEDISYNC.py:20076 |
| edit\_doctor\_profile | Function | 20089-20145 | try: # Crear ventana de edición de perfil profile\_window = tk.Toplevel(self.root) profile\_window.tit... | MEDISYNC.py:20089 |
| save\_profile | Function | 20146-20210 | Sin documentación | MEDISYNC.py:20146 |
| change\_password | Function | 20211-20262 | try: # Crear ventana de cambio de contraseña password\_window = tk.Toplevel(self.root) password\_windo... | MEDISYNC.py:20211 |
| change\_password\_action | Function | 20263-20307 | Sin documentación | MEDISYNC.py:20263 |
| configure\_schedule | Function | 20308-20384 | try: # Crear ventana de configuración de horarios schedule\_window = tk.Toplevel(self.root) schedule\_... | MEDISYNC.py:20308 |
| save\_schedule | Function | 20385-20428 | Sin documentación | MEDISYNC.py:20385 |
| complete\_appointment\_from\_details | Function | 20429-20444 | try: appointment\_id = self.get\_appointment\_id\_from\_selection() if appointment\_id: response = message... | MEDISYNC.py:20429 |
| cancel\_appointment\_from\_details | Function | 20445-20461 | try: appointment\_id = self.get\_appointment\_id\_from\_selection() if appointment\_id: response = message... | MEDISYNC.py:20445 |
| add\_appointment\_notes\_from\_details | Function | 20462-20469 | try: self.add\_appointment\_notes() except Exception as e: messagebox.showerror("Error", f"Error agreg... | MEDISYNC.py:20462 |
| schedule\_appointment\_for\_patient | Function | 20470-20536 | try: # Crear ventana de programación de cita appointment\_window = tk.Toplevel(self.root) appointment... | MEDISYNC.py:20470 |
| save\_appointment | Function | 20537-20595 | Sin documentación | MEDISYNC.py:20537 |
| view\_patient\_medical\_history | Function | 20596-20704 | try: # Obtener información del paciente patient\_info = self.get\_patient\_full\_info(patient\_id) if not... | MEDISYNC.py:20596 |
| create\_medical\_record\_for\_patient | Function | 20705-20786 | try: patient\_name = f"{nombre} {apellido}" # Crear ventana de nuevo historial record\_window = tk.To... | MEDISYNC.py:20705 |
| save\_record | Function | 20787-20825 | Sin documentación | MEDISYNC.py:20787 |
| edit\_medical\_record\_detail | Function | 20826-20833 | try: messagebox.showinfo("Editar", "Función de edición desde detalles - En desarrollo\n\n" "Esta fun... | MEDISYNC.py:20826 |
| print\_medical\_record\_detail | Function | 20834-20843 | try: messagebox.showinfo("Imprimir", "Función de impresión desde detalles - En desarrollo\n\n" "Esta... | MEDISYNC.py:20834 |
| load\_patients\_for\_medical\_history | Function | 20844-20897 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() # Obtener filtro de búsqueda se... | MEDISYNC.py:20844 |
| search\_patients\_medical | Function | 20898-20901 | self.load\_patients\_for\_medical\_history() def on\_patient\_select\_medical(self, event): | MEDISYNC.py:20898 |
| on\_patient\_select\_medical | Function | 20902-20926 | selection = self.patients\_medical\_tree.selection() if selection: item = self.patients\_medical\_tree.i... | MEDISYNC.py:20902 |
| show\_patient\_info\_medical | Function | 20927-20963 | # Limpiar frame for widget in self.patient\_info\_frame.winfo\_children(): widget.destroy() # Crear ta... | MEDISYNC.py:20927 |
| show\_default\_medical\_info | Function | 20964-20983 | # Limpiar frames for widget in self.patient\_info\_frame.winfo\_children(): widget.destroy() # Limpiar... | MEDISYNC.py:20964 |
| load\_medical\_records\_for\_patient | Function | 20984-21022 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() | MEDISYNC.py:20984 |
| on\_medical\_record\_select | Function | 21023-21031 | selection = self.medical\_records\_tree.selection() if selection: item = self.medical\_records\_tree.ite... | MEDISYNC.py:21023 |
| add\_medical\_record | Function | 21032-21039 | if not self.selected\_patient\_id: messagebox.showwarning("Advertencia", "Por favor seleccione un paci... | MEDISYNC.py:21032 |
| open\_medical\_record\_form | Function | 21040-21271 | form\_window = tk.Toplevel() title = "Editar Consulta" if edit\_mode else "Nueva Consulta Médica" form... | MEDISYNC.py:21040 |
| on\_cancel\_enter | Function | 21272-21273 | Sin documentación | MEDISYNC.py:21272 |
| on\_cancel\_leave | Function | 21274-21290 | Sin documentación | MEDISYNC.py:21274 |
| on\_save\_enter | Function | 21291-21292 | Sin documentación | MEDISYNC.py:21291 |
| on\_save\_leave | Function | 21293-21303 | Sin documentación | MEDISYNC.py:21293 |
| \_on\_mousewheel | Function | 21304-21384 | try: # Validar campos obligatorios if not form\_vars['fecha\_consulta'].get(): messagebox.showerror("E... | MEDISYNC.py:21304 |
| create\_medical\_record | Function | 21385-21391 | if hasattr(self, 'selected\_patient\_id') and self.selected\_patient\_id: self.add\_medical\_record() else... | MEDISYNC.py:21385 |
| view\_medical\_record\_detail | Function | 21392-21425 | if not self.selected\_medical\_record\_id: messagebox.showwarning("Advertencia", "Por favor seleccione ... | MEDISYNC.py:21392 |
| show\_medical\_record\_details | Function | 21426-21527 | details\_window = tk.Toplevel() details\_window.title("Detalle de Consulta Médica - MEDISYNC") details... | MEDISYNC.py:21426 |
| edit\_medical\_record | Function | 21528-21535 | if not self.selected\_medical\_record\_id: messagebox.showwarning("Advertencia", "Por favor seleccione ... | MEDISYNC.py:21528 |
| print\_medical\_record | Function | 21536-21570 | # Obtener el registro seleccionado directamente del TreeView selection = self.medical\_records\_tree.s... | MEDISYNC.py:21536 |
| get\_patient\_from\_medical\_record | Function | 21571-21604 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() | MEDISYNC.py:21571 |
| open\_medical\_history\_print\_window | Function | 21605-21745 | window = tk.Toplevel(self.root) window.title("🖨️ Imprimir Historial Médico") window.geometry("900x70... | MEDISYNC.py:21605 |
| load\_patient\_records\_for\_print | Function | 21746-21834 | # Limpiar registros anteriores for widget in self.records\_scrollable\_frame.winfo\_children(): widget.... | MEDISYNC.py:21746 |
| toggle\_all\_records | Function | 21835-21841 | if hasattr(self, 'selected\_records') and self.selected\_records: for var in self.selected\_records.val... | MEDISYNC.py:21835 |
| update\_medical\_history\_preview | Function | 21842-21877 | if patient\_info is None: # Obtener info del paciente desde el registro seleccionado patient\_info = s... | MEDISYNC.py:21842 |
| generate\_medical\_history\_content | Function | 21878-21962 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() # Obtener registros seleccionad... | MEDISYNC.py:21878 |
| generate\_medical\_history\_pdf | Function | 21963-22183 | try: # Verificar que selected\_records exista if not hasattr(self, 'selected\_records') or not self.se... | MEDISYNC.py:21963 |
| create\_medical\_note\_from\_appointment | Function | 22184-22235 | try: # Obtener datos de la cita conn = self.db\_manager.get\_connection() cursor = conn.cursor() | MEDISYNC.py:22184 |
| edit\_doctor\_profile | Function | 22236-22237 | Sin documentación | MEDISYNC.py:22236 |
| change\_password | Function | 22238-22239 | Sin documentación | MEDISYNC.py:22238 |
| configure\_schedule | Function | 22240-22243 | if hasattr(self, 'doctor\_appointments\_tree'): # Solo recargar las citas con filtros, sin recursión s... | MEDISYNC.py:22240 |
| create\_new\_appointment | Function | 22244-22245 | if hasattr(self, 'doctor\_appointments\_tree'): # Solo recargar las citas con filtros, sin recursión s... | MEDISYNC.py:22244 |
| show\_appointment\_calendar | Function | 22246-22247 | if hasattr(self, 'doctor\_appointments\_tree'): # Solo recargar las citas con filtros, sin recursión s... | MEDISYNC.py:22246 |
| apply\_appointment\_filters | Function | 22248-22253 | if hasattr(self, 'doctor\_appointments\_tree'): # Solo recargar las citas con filtros, sin recursión s... | MEDISYNC.py:22248 |
| load\_filtered\_appointments | Function | 22254-22335 | try: # Obtener valores de filtros period\_filter = getattr(self, 'appointment\_filter', None) status\_f... | MEDISYNC.py:22254 |
| load\_appointment\_doctors | Function | 22336-22345 | try: doctors = self.db\_manager.get\_all\_doctors() doctor\_names = ['Todos'] + [f"Dr. {doc['nombre']} {... | MEDISYNC.py:22336 |
| load\_secretaria\_appointments | Function | 22346-22376 | try: # Placeholder - implementar carga real appointments = self.db\_manager.get\_all\_appointments() #... | MEDISYNC.py:22346 |
| edit\_appointment | Function | 22377-22398 | try: selection = self.appointments\_tree.selection() if not selection: messagebox.showwarning("Advert... | MEDISYNC.py:22377 |
| confirm\_appointment | Function | 22399-22424 | try: # Verificar si estamos en la vista de doctor if hasattr(self, 'doctor\_appointments\_tree'): sele... | MEDISYNC.py:22399 |
| cancel\_appointment | Function | 22425-22450 | try: # Verificar si estamos en la vista de doctor if hasattr(self, 'doctor\_appointments\_tree'): sele... | MEDISYNC.py:22425 |
| call\_patient | Function | 22451-22451 | Sin documentación | MEDISYNC.py:22451 |
| generate\_appointment\_invoice | Function | 22452-22452 | Sin documentación | MEDISYNC.py:22452 |
| register\_new\_patient | Function | 22453-22453 | user\_window = tk.Toplevel(self.root) user\_window.title("Gestión de Usuarios - MEDISYNC") user\_window... | MEDISYNC.py:22453 |
| import\_patients | Function | 22454-22454 | user\_window = tk.Toplevel(self.root) user\_window.title("Gestión de Usuarios - MEDISYNC") user\_window... | MEDISYNC.py:22454 |
| export\_patients | Function | 22455-22455 | user\_window = tk.Toplevel(self.root) user\_window.title("Gestión de Usuarios - MEDISYNC") user\_window... | MEDISYNC.py:22455 |
| search\_patients\_secretaria | Function | 22456-22456 | user\_window = tk.Toplevel(self.root) user\_window.title("Gestión de Usuarios - MEDISYNC") user\_window... | MEDISYNC.py:22456 |
| clear\_patient\_search | Function | 22457-22457 | user\_window = tk.Toplevel(self.root) user\_window.title("Gestión de Usuarios - MEDISYNC") user\_window... | MEDISYNC.py:22457 |
| load\_secretaria\_patients | Function | 22458-22461 | user\_window = tk.Toplevel(self.root) user\_window.title("Gestión de Usuarios - MEDISYNC") user\_window... | MEDISYNC.py:22458 |
| open\_user\_management | Function | 22462-22601 | user\_window = tk.Toplevel(self.root) user\_window.title("Gestión de Usuarios - MEDISYNC") user\_window... | MEDISYNC.py:22462 |
| configure\_scroll\_region | Function | 22602-22606 | Sin documentación | MEDISYNC.py:22602 |
| configure\_canvas\_width | Function | 22607-22620 | Sin documentación | MEDISYNC.py:22607 |
| \_on\_mousewheel | Function | 22621-22653 | Sin documentación | MEDISYNC.py:22621 |
| force\_scroll\_update | Function | 22654-22680 | Sin documentación | MEDISYNC.py:22654 |
| load\_users\_list | Function | 22681-22750 | try: # Limpiar tabla for item in self.users\_tree.get\_children(): self.users\_tree.delete(item) conn ... | MEDISYNC.py:22681 |
| load\_user\_stats | Function | 22751-22774 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() # Total de usuarios cursor.exec... | MEDISYNC.py:22751 |
| on\_user\_select | Function | 22797-22815 | selection = self.users\_tree.selection() if selection: item = self.users\_tree.item(selection[0]) valu... | MEDISYNC.py:22797 |
| update\_actions\_scroll | Function | 22816-22823 | if hasattr(self, 'actions\_canvas') and hasattr(self, 'scrollable\_actions\_frame'): self.scrollable\_ac... | MEDISYNC.py:22816 |
| enable\_action\_buttons | Function | 22824-22828 | for btn in self.action\_buttons: btn.config(state='normal') def disable\_action\_buttons(self): | MEDISYNC.py:22824 |
| disable\_action\_buttons | Function | 22829-22833 | for btn in self.action\_buttons: btn.config(state='disabled') def show\_default\_info(self): | MEDISYNC.py:22829 |
| show\_default\_info | Function | 22834-22844 | for widget in self.user\_info\_frame.winfo\_children(): widget.destroy() self.user\_info\_label = tk.Lab... | MEDISYNC.py:22834 |
| load\_selected\_user\_info | Function | 22845-22883 | if not self.selected\_user\_id: return try: conn = self.db\_manager.get\_connection() cursor = conn.cur... | MEDISYNC.py:22845 |
| update\_user\_info\_display | Function | 22884-22925 | if not self.selected\_user\_data: return # Limpiar frame for widget in self.user\_info\_frame.winfo\_chi... | MEDISYNC.py:22884 |
| search\_users | Function | 22926-22929 | self.load\_users\_list() def filter\_users(self, event=None): | MEDISYNC.py:22926 |
| filter\_users | Function | 22930-22933 | self.load\_users\_list() def clear\_user\_search(self): | MEDISYNC.py:22930 |
| clear\_user\_search | Function | 22934-22943 | if hasattr(self, 'user\_search\_entry'): self.user\_search\_entry.delete(0, tk.END) if hasattr(self, 'us... | MEDISYNC.py:22934 |
| refresh\_user\_list | Function | 22944-22948 | self.load\_users\_list() messagebox.showinfo("Actualizado", "Lista de usuarios actualizada correctamen... | MEDISYNC.py:22944 |
| add\_new\_user | Function | 22949-22952 | self.open\_user\_form() def edit\_selected\_user(self): | MEDISYNC.py:22949 |
| edit\_selected\_user | Function | 22953-22959 | if not self.selected\_user\_id: messagebox.showwarning("Selección requerida", "Por favor seleccione un... | MEDISYNC.py:22953 |
| open\_user\_form | Function | 22960-23142 | form\_window = tk.Toplevel() title = "Editar Usuario" if edit\_mode else "Nuevo Usuario" form\_window.t... | MEDISYNC.py:22960 |
| open\_calendar | Function | 23143-23169 | try: from tkcalendar import DateEntry calendar\_window = tk.Toplevel() calendar\_window.title("Selecc... | MEDISYNC.py:23143 |
| select\_date | Function | 23170-23192 | Sin documentación | MEDISYNC.py:23170 |
| save\_manual\_date | Function | 23193-23204 | Sin documentación | MEDISYNC.py:23193 |
| on\_user\_type\_change | Function | 23205-23218 | # Limpiar frame específico for widget in specific\_frame.winfo\_children(): widget.destroy() user\_typ... | MEDISYNC.py:23205 |
| create\_doctor\_specific\_fields | Function | 23219-23230 | grid = tk.Frame(parent) grid.pack(fill='x') tk.Label(grid, text="Especialidad \*:", font=('Arial', 1... | MEDISYNC.py:23219 |
| create\_patient\_specific\_fields | Function | 23231-23272 | grid = tk.Frame(parent) grid.pack(fill='x') # Seguro médico tk.Label(grid, text="Seguro Médico:", f... | MEDISYNC.py:23231 |
| save\_user | Function | 23273-23428 | try: # Validar campos obligatorios if not form\_vars['nombre'].get().strip(): messagebox.showerror("E... | MEDISYNC.py:23273 |
| validate\_unique\_email | Function | 23429-23451 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() if edit\_mode and hasattr(self, ... | MEDISYNC.py:23429 |
| generate\_expediente\_number | Function | 23452-23479 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() # Obtener el último número de e... | MEDISYNC.py:23452 |
| validate\_unique\_email | Function | 23480-23501 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() if edit\_mode: cursor.execute("S... | MEDISYNC.py:23480 |
| generate\_expediente\_number | Function | 23502-23525 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() # Obtener el último número de e... | MEDISYNC.py:23502 |
| hash\_password | Function | 23526-23530 | import hashlib return hashlib.sha256(password.encode()).hexdigest() def change\_user\_password(self): | MEDISYNC.py:23526 |
| change\_user\_password | Function | 23531-23585 | if not self.selected\_user\_id: messagebox.showwarning("Selección requerida", "Por favor seleccione un... | MEDISYNC.py:23531 |
| save\_password | Function | 23586-23626 | Sin documentación | MEDISYNC.py:23586 |
| activate\_user | Function | 23627-23657 | if not self.selected\_user\_id: messagebox.showwarning("Selección requerida", "Por favor seleccione un... | MEDISYNC.py:23627 |
| deactivate\_user | Function | 23658-23688 | if not self.selected\_user\_id: messagebox.showwarning("Selección requerida", "Por favor seleccione un... | MEDISYNC.py:23658 |
| view\_user\_details | Function | 23689-23788 | if not self.selected\_user\_id: messagebox.showwarning("Selección requerida", "Por favor seleccione un... | MEDISYNC.py:23689 |
| send\_user\_email | Function | 23789-23840 | if not self.selected\_user\_id: messagebox.showwarning("Selección requerida", "Por favor seleccione un... | MEDISYNC.py:23789 |
| send\_email | Function | 23841-23861 | Sin documentación | MEDISYNC.py:23841 |
| view\_access\_history | Function | 23862-23933 | if not self.selected\_user\_id: messagebox.showwarning("Selección requerida", "Por favor seleccione un... | MEDISYNC.py:23862 |
| delete\_user | Function | 23934-23990 | if not self.selected\_user\_id: messagebox.showwarning("Selección requerida", "Por favor seleccione un... | MEDISYNC.py:23934 |
| request\_appointment | Function | 23991-23994 | self.switch\_patient\_tab("appointments") def view\_my\_history(self): | MEDISYNC.py:23991 |
| view\_my\_history | Function | 23995-23998 | self.switch\_patient\_tab("history") def view\_my\_bills(self): | MEDISYNC.py:23995 |
| view\_my\_bills | Function | 23999-24002 | self.switch\_patient\_tab("billing") def update\_my\_profile(self): | MEDISYNC.py:23999 |
| update\_my\_profile | Function | 24003-24005 | self.switch\_patient\_tab("settings") def request\_new\_appointment(self): messagebox.showinfo("Acción",... | MEDISYNC.py:24003 |
| request\_new\_appointment | Function | 24006-24008 | try: import calendar from datetime import datetime, timedelta # Crear ventana de calendario cal\_win... | MEDISYNC.py:24006 |
| select\_date\_enhanced | Function | 24009-24076 | try: import calendar from datetime import datetime, timedelta # Crear ventana de calendario cal\_win... | MEDISYNC.py:24009 |
| navigate\_month | Function | 24077-24086 | Sin documentación | MEDISYNC.py:24077 |
| update\_calendar | Function | 24087-24128 | Sin documentación | MEDISYNC.py:24087 |
| select\_day | Function | 24129-24150 | Sin documentación | MEDISYNC.py:24129 |
| select\_quick\_date | Function | 24151-24166 | Sin documentación | MEDISYNC.py:24151 |
| clear\_appointment\_form | Function | 24167-24207 | try: # Limpiar variables StringVar for var\_name, var in appointment\_vars.items(): if var\_name == 'es... | MEDISYNC.py:24167 |
| setup\_form\_validation | Function | 24208-24209 | def validate\_date(\*args): date\_str = appointment\_vars['fecha'].get() if date\_str: try: # Validar for... | MEDISYNC.py:24208 |
| validate\_date | Function | 24210-24221 | Sin documentación | MEDISYNC.py:24210 |
| validate\_time | Function | 24222-24237 | Sin documentación | MEDISYNC.py:24222 |
| save\_appointment\_enhanced | Function | 24238-24334 | try: # Obtener valores del formulario fecha = appointment\_vars['fecha'].get().strip() hora = appoint... | MEDISYNC.py:24238 |
| check\_appointment\_conflict | Function | 24358-24388 | try: # Obtener todas las citas del doctor para la fecha appointments = self.db\_manager.get\_all\_appoi... | MEDISYNC.py:24358 |
| update\_available\_hours | Function | 24389-24426 | try: doctor\_text = doctor\_var.get() if not doctor\_text or "Seleccione" in doctor\_text: return # Ext... | MEDISYNC.py:24389 |
| get\_doctor\_schedule | Function | 24427-24445 | try: # Aquí puedes implementar la lógica para obtener horarios específicos del doctor # Por ahora re... | MEDISYNC.py:24427 |
| show\_more\_hours | Function | 24446-24480 | try: hours\_window = tk.Toplevel(self.root) hours\_window.title("⏰ Horarios Disponibles") hours\_window... | MEDISYNC.py:24446 |
| select\_hour | Function | 24481-24498 | Sin documentación | MEDISYNC.py:24481 |
| set\_appointment\_hour | Function | 24499-24516 | try: # Aquí podrías agregar verificación de disponibilidad # Por ahora simplemente establecemos la h... | MEDISYNC.py:24499 |
| find\_and\_set\_hour\_entry | Function | 24517-24540 | try: if hasattr(widget, 'winfo\_children'): for child in widget.winfo\_children(): if hasattr(child, '... | MEDISYNC.py:24517 |
| get\_billing\_summary | Function | 24541-24541 | try: conn = self.db\_manager.get\_connection() cur = conn.cursor() stats = {} # Citas de hoy today =... | MEDISYNC.py:24541 |
| get\_secretaria\_quick\_stats | Function | 24542-24627 | try: conn = self.db\_manager.get\_connection() cur = conn.cursor() stats = {} # Citas de hoy today =... | MEDISYNC.py:24542 |
| get\_patient\_health\_summary | Function | 24628-24628 | try: period = self.billing\_period\_filter.get() status = self.billing\_status\_filter.get() self.load\_p... | MEDISYNC.py:24628 |
| get\_patient\_billing\_summary | Function | 24629-24629 | try: period = self.billing\_period\_filter.get() status = self.billing\_status\_filter.get() self.load\_p... | MEDISYNC.py:24629 |
| get\_patient\_profile\_data | Function | 24630-24630 | try: period = self.billing\_period\_filter.get() status = self.billing\_status\_filter.get() self.load\_p... | MEDISYNC.py:24630 |
| get\_patient\_personal\_stats | Function | 24631-24634 | try: period = self.billing\_period\_filter.get() status = self.billing\_status\_filter.get() self.load\_p... | MEDISYNC.py:24631 |
| filter\_patient\_billing | Function | 24635-24644 | try: period = self.billing\_period\_filter.get() status = self.billing\_status\_filter.get() self.load\_p... | MEDISYNC.py:24635 |
| clear\_billing\_filters | Function | 24645-24653 | try: self.billing\_period\_filter.set('Todos') self.billing\_status\_filter.set('Todos') self.load\_patie... | MEDISYNC.py:24645 |
| view\_billing\_detail | Function | 24654-24666 | try: selection = self.patient\_billing\_tree.selection() if not selection: messagebox.showwarning("Sel... | MEDISYNC.py:24654 |
| download\_billing\_pdf | Function | 24682-24699 | try: selection = self.patient\_billing\_tree.selection() if not selection: messagebox.showwarning("Sel... | MEDISYNC.py:24682 |
| email\_my\_billing | Function | 24700-24717 | try: selection = self.patient\_billing\_tree.selection() if not selection: messagebox.showwarning("Sel... | MEDISYNC.py:24700 |
| show\_billing\_summary | Function | 24718-24724 | try: summary = self.get\_patient\_billing\_summary() | MEDISYNC.py:24718 |
| view\_history\_detail\_simple | Function | 24738-24926 | try: selection = self.medical\_history\_tree.selection() if not selection: messagebox.showwarning("Adv... | MEDISYNC.py:24738 |
| \_on\_mousewheel | Function | 24927-24936 | Sin documentación | MEDISYNC.py:24927 |
| update\_appointment\_status\_db | Function | 24937-24960 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() if notes: | MEDISYNC.py:24937 |
| get\_appointment\_id\_from\_selection | Function | 24961-24993 | try: selection = self.doctor\_appointments\_tree.selection() if not selection: return None item = sel... | MEDISYNC.py:24961 |
| save\_appointment\_notes\_db | Function | 24994-25012 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() | MEDISYNC.py:24994 |
| save\_medical\_record\_db | Function | 25013-25033 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() | MEDISYNC.py:25013 |
| update\_doctor\_profile\_db | Function | 25034-25061 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() # Actualizar tabla usuarios | MEDISYNC.py:25034 |
| update\_doctor\_password\_db | Function | 25062-25080 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() | MEDISYNC.py:25062 |
| verify\_doctor\_password | Function | 25081-25105 | try: import hashlib conn = self.db\_manager.get\_connection() cursor = conn.cursor() cursor.execute(... | MEDISYNC.py:25081 |
| save\_doctor\_schedule\_db | Function | 25106-25144 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() # Crear tabla de horarios si no... | MEDISYNC.py:25106 |
| get\_doctor\_schedule\_db | Function | 25145-25171 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() | MEDISYNC.py:25145 |
| create\_appointment\_db | Function | 25172-25199 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() | MEDISYNC.py:25172 |
| get\_patient\_full\_info | Function | 25200-25233 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() | MEDISYNC.py:25200 |
| search\_patients\_for\_doctor | Function | 25234-25261 | try: conn = self.db\_manager.get\_connection() cursor = conn.cursor() | MEDISYNC.py:25234 |
| format\_datetime\_for\_display | Function | 25262-25272 | try: if not datetime\_str: return 'N/A' dt = datetime.fromisoformat(datetime\_str) return dt.strftime... | MEDISYNC.py:25262 |
| format\_date\_for\_display | Function | 25273-25288 | try: if not date\_str: return 'N/A' if 'T' in date\_str: dt = datetime.fromisoformat(date\_str) return... | MEDISYNC.py:25273 |
| calculate\_age | Function | 25289-25309 | try: if not birth\_date: return 0 if isinstance(birth\_date, str): birth\_dt = datetime.strptime(birth... | MEDISYNC.py:25289 |
| validate\_email | Function | 25310-25315 | import re pattern = r'^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$' return re.match(pattern, ema... | MEDISYNC.py:25310 |
| validate\_phone | Function | 25316-25323 | # Eliminar espacios y guiones phone\_clean = phone.replace(' ', '').replace('-', '').replace('(', '')... | MEDISYNC.py:25316 |

## Funciones Principales de MEDISYNC.py

### 🔧 \_\_init\_\_

|  |  |
| --- | --- |
| Ubicación | MEDISYNC.py, líneas 52-61 |
| Tipo | Function |
| Signatura | def \_\_init\_\_(self, root=None): |
| Descripción | Sin documentación específica |

##### Primeras líneas del código:

def \_\_init\_\_(self, root=None):  
 self.db\_manager = DBManager('database/medisync.db')  
 self.current\_user = None  
 self.root = root  
 self.users\_tree = None  
 if root is None:  
 self.create\_login\_window()  
 else:  
 self.setup\_root\_window()

### 🔧 create\_login\_window

|  |  |
| --- | --- |
| Ubicación | MEDISYNC.py, líneas 75-220 |
| Tipo | Function |
| Signatura | def create\_login\_window(self): |
| Descripción | if self.root is None: self.root = tk.Tk() self.root.title("MEDISYNC - Sistema de Gestión Médica") self.root.geometry("750x850") self.root.resizable(False, False) # Centrar ventana self.center\_window(self.root, 750, 850) # Aplicar tema ttk try: from ui\_theme import apply\_theme apply\_theme(self.root) except Exception: pass # Configurar fondo self.root.configure(bg='#F8FAFC') |

##### Primeras líneas del código:

def create\_login\_window(self):  
 """Crear ventana de login moderna y estética"""  
 if self.root is None:  
 self.root = tk.Tk()  
   
 self.root.title("MEDISYNC - Sistema de Gestión Médica")  
 self.root.geometry("750x850")  
 self.root.resizable(False, False)  
   
 # Centrar ventana

### 🔧 login

|  |  |
| --- | --- |
| Ubicación | MEDISYNC.py, líneas 239-257 |
| Tipo | Function |
| Signatura | def login(self): |
| Descripción | email = self.email\_entry.get().strip() password = self.password\_entry.get().strip() if not email or not password: messagebox.showerror("Error", "Por favor ingrese email y contraseña") return user = self.db\_manager.authenticate\_user(email, password) if user: self.current\_user = user self.root.destroy() self.create\_main\_window() else: messagebox.showerror("Error", "Credenciales inválidas") self.password\_entry.delete(0, tk.END) def create\_main\_window(self): |

##### Primeras líneas del código:

def login(self):  
 """Procesar login"""  
 email = self.email\_entry.get().strip()  
 password = self.password\_entry.get().strip()  
   
 if not email or not password:  
 messagebox.showerror("Error", "Por favor ingrese email y contraseña")  
 return  
   
 user = self.db\_manager.authenticate\_user(email, password)

### 🔧 create\_main\_window

|  |  |
| --- | --- |
| Ubicación | MEDISYNC.py, líneas 258-274 |
| Tipo | Function |
| Signatura | def create\_main\_window(self): |
| Descripción | self.root = tk.Tk() self.root.title(f"MEDISYNC - {self.current\_user.nombre} {self.current\_user.apellido} ({self.current\_user.tipo\_usuario.title()})") # Configurar pantalla completa self.root.state('zoomed') # Para Windows - pantalla completa self.root.configure(bg='#F8FAFC') # Opcional: permitir salir de pantalla completa con Escape self.root.bind('<Escape>', lambda e: self.root.state('normal')) self.root.bind('<F11>', lambda e: self.toggle\_fullscreen()) self.create\_menu\_interface() self.root.mainloop() def toggle\_fullscreen(self): |

##### Primeras líneas del código:

def create\_main\_window(self):  
 """Crear ventana principal según el rol del usuario"""  
 self.root = tk.Tk()  
 self.root.title(f"MEDISYNC - {self.current\_user.nombre} {self.current\_user.apellido} ({self.current\_user.tipo\_usuario.title()})")  
   
 # Configurar pantalla completa  
 self.root.state('zoomed') # Para Windows - pantalla completa  
 self.root.configure(bg='#F8FAFC')  
   
 # Opcional: permitir salir de pantalla completa con Escape

### 🔧 create\_admin\_menu

|  |  |
| --- | --- |
| Ubicación | MEDISYNC.py, líneas 323-396 |
| Tipo | Function |
| Signatura | def create\_admin\_menu(self, parent): |
| Descripción | # Frame principal para el menú menu\_container = tk.Frame(parent, bg='#F8FAFC') menu\_container.pack(fill='both', expand=True) # Header del menú con estilo moderno header\_frame = tk.Frame(menu\_container, bg='#1E3A8A', height=70) header\_frame.pack(fill='x') header\_frame.pack\_propagate(False) # Contenido del header header\_content = tk.Frame(header\_frame, bg='#1E3A8A') header\_content.pack(expand=True, fill='both', padx=30, pady=10) # Logo y título title\_frame = tk.Frame(header\_content, bg='#1E3A8A') title\_frame.pack(side='left', expand=True, fill='both') tk.Label(title\_frame, text="🏥", font=('Arial', 20), bg='#1E3A8A', fg='white').pack(side='left', pady=5) tk.Label(title\_frame, text="MEDISYNC", font=('Arial', 18, 'bold'), bg='#1E3A8A', fg='white').pack(side='left', padx=(10, 0), pady=5) |

##### Primeras líneas del código:

def create\_admin\_menu(self, parent):  
 """Crear menú de administrador con diseño moderno"""  
 # Frame principal para el menú  
 menu\_container = tk.Frame(parent, bg='#F8FAFC')  
 menu\_container.pack(fill='both', expand=True)  
   
 # Header del menú con estilo moderno  
 header\_frame = tk.Frame(menu\_container, bg='#1E3A8A', height=70)  
 header\_frame.pack(fill='x')  
 header\_frame.pack\_propagate(False)

### 🔧 create\_dashboard\_tab

|  |  |
| --- | --- |
| Ubicación | MEDISYNC.py, líneas 447-550 |
| Tipo | Function |
| Signatura | def create\_dashboard\_tab(self, parent): |
| Descripción | # Frame principal con mejor diseño main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=True) # Header principal con gradiente visual header\_frame = tk.Frame(main\_frame, bg='#1E3A8A', height=80) header\_frame.pack(fill='x') header\_frame.pack\_propagate(False) # Contenido del header header\_content = tk.Frame(header\_frame, bg='#1E3A8A') header\_content.pack(expand=True, fill='both', padx=30, pady=15) # Título principal title\_frame = tk.Frame(header\_content, bg='#1E3A8A') title\_frame.pack(side='left', fill='y') tk.Label(title\_frame, text="📊 Dashboard del Sistema", font=('Arial', 20, 'bold'), bg='#1E3A8A', fg='white').pack(anchor='w') |

##### Primeras líneas del código:

def create\_dashboard\_tab(self, parent):  
 """Crear pestaña de dashboard con diseño moderno"""  
 # Frame principal con mejor diseño  
 main\_frame = tk.Frame(parent, bg='#F8FAFC')  
 main\_frame.pack(fill='both', expand=True)  
   
 # Header principal con gradiente visual  
 header\_frame = tk.Frame(main\_frame, bg='#1E3A8A', height=80)  
 header\_frame.pack(fill='x')  
 header\_frame.pack\_propagate(False)

### 🔧 create\_users\_tab

|  |  |
| --- | --- |
| Ubicación | MEDISYNC.py, líneas 633-829 |
| Tipo | Function |
| Signatura | def create\_users\_tab(self, parent): |
| Descripción | # Frame principal con mejor diseño main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=True) # Header con gradiente visual header\_frame = tk.Frame(main\_frame, bg='#1E3A8A', height=70) header\_frame.pack(fill='x') header\_frame.pack\_propagate(False) # Contenido del header header\_content = tk.Frame(header\_frame, bg='#1E3A8A') header\_content.pack(expand=True, fill='both', padx=20, pady=10) # Título principal title\_frame = tk.Frame(header\_content, bg='#1E3A8A') title\_frame.pack(side='left', fill='y') tk.Label(title\_frame, text="👥 Gestión de Usuarios", font=('Arial', 18, 'bold'), bg='#1E3A8A', fg='white').pack(anchor='w') |

##### Primeras líneas del código:

def create\_users\_tab(self, parent):  
 """Crear pestaña de gestión completa de usuarios integrada"""  
 # Frame principal con mejor diseño  
 main\_frame = tk.Frame(parent, bg='#F8FAFC')  
 main\_frame.pack(fill='both', expand=True)  
   
 # Header con gradiente visual  
 header\_frame = tk.Frame(main\_frame, bg='#1E3A8A', height=70)  
 header\_frame.pack(fill='x')  
 header\_frame.pack\_propagate(False)

### 🔧 create\_appointments\_tab

|  |  |
| --- | --- |
| Ubicación | MEDISYNC.py, líneas 850-1078 |
| Tipo | Function |
| Signatura | def create\_appointments\_tab(self, parent): |
| Descripción | # Frame principal main\_frame = tk.Frame(parent, bg='#F8FAFC') main\_frame.pack(fill='both', expand=True, padx=20, pady=20) # Header con diseño moderno header\_frame = tk.Frame(main\_frame, bg='#1E3A8A', height=80, relief='raised', bd=2) header\_frame.pack(fill='x', pady=(0, 20)) header\_frame.pack\_propagate(False) # Contenido del header header\_content = tk.Frame(header\_frame, bg='#1E3A8A') header\_content.pack(fill='both', expand=True, padx=20, pady=15) # Título del módulo title\_frame = tk.Frame(header\_content, bg='#1E3A8A') title\_frame.pack(side='left', fill='y') tk.Label(title\_frame, text="📅", font=('Arial', 24), bg='#1E3A8A', fg='white').pack(side='left') |

##### Primeras líneas del código:

def create\_appointments\_tab(self, parent):  
 """Crear pestaña de citas con diseño moderno similar a usuarios"""  
 # Frame principal  
 main\_frame = tk.Frame(parent, bg='#F8FAFC')  
 main\_frame.pack(fill='both', expand=True, padx=20, pady=20)  
   
 # Header con diseño moderno  
 header\_frame = tk.Frame(main\_frame, bg='#1E3A8A', height=80, relief='raised', bd=2)  
 header\_frame.pack(fill='x', pady=(0, 20))  
 header\_frame.pack\_propagate(False)

### 🔧 new\_appointment\_window

|  |  |
| --- | --- |
| Ubicación | MEDISYNC.py, líneas 1301-1638 |
| Tipo | Function |
| Signatura | def new\_appointment\_window(self): |
| Descripción | window = tk.Toplevel(self.root) window.title("🏥 MEDISYNC - Nueva Cita Médica") window.geometry("850x750") window.configure(bg='#F8FAFC') window.transient(self.root) window.grab\_set() window.resizable(False, False) # Centrar ventana window.update\_idletasks() x = (window.winfo\_screenwidth() // 2) - (850 // 2) y = (window.winfo\_screenheight() // 2) - (750 // 2) window.geometry(f"850x750+{x}+{y}") # Header mejorado con gradiente visual header\_frame = tk.Frame(window, bg='#1E3A8A', height=100) header\_frame.pack(fill='x') header\_frame.pack\_propagate(False) |

##### Primeras líneas del código:

def new\_appointment\_window(self):  
 """Ventana para crear nueva cita con diseño mejorado"""  
 window = tk.Toplevel(self.root)  
 window.title("🏥 MEDISYNC - Nueva Cita Médica")  
 window.geometry("850x750")  
 window.configure(bg='#F8FAFC')  
 window.transient(self.root)  
 window.grab\_set()  
 window.resizable(False, False)

# database\_manager.py - Gestor de Base de Datos

Análisis detallado del archivo: database\_manager.py

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Función/Clase | Tipo | Líneas | Descripción | Ubicación |
| User: | Class | 15-27 | Sin documentación | database\_manager.py:15 |
| Cita: | Class | 29-38 | Sin documentación | database\_manager.py:29 |
| HistorialMedico: | Class | 40-50 | Sin documentación | database\_manager.py:40 |
| Factura: | Class | 52-63 | Sin documentación | database\_manager.py:52 |
| DatabaseManager: | Class | 64-66 | def \_\_init\_\_(self, db\_path='database/medisync.db'): self.db\_path = db\_path self.lock = threading.Lo... | database\_manager.py:64 |
| \_\_init\_\_ | Function | 67-73 | os.makedirs(os.path.dirname(self.db\_path), exist\_ok=True) def get\_connection(self): | database\_manager.py:67 |
| ensure\_database\_exists | Function | 74-77 | os.makedirs(os.path.dirname(self.db\_path), exist\_ok=True) def get\_connection(self): | database\_manager.py:74 |
| get\_connection | Function | 78-83 | conn = sqlite3.connect(self.db\_path, check\_same\_thread=False) conn.row\_factory = sqlite3.Row return ... | database\_manager.py:78 |
| get\_simple\_connection | Function | 84-87 | return sqlite3.connect(self.db\_path, check\_same\_thread=False) def create\_tables(self): | database\_manager.py:84 |
| create\_tables | Function | 88-252 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: # Tabla usuarios | database\_manager.py:88 |
| hash\_password | Function | 253-256 | return hashlib.sha256(password.encode()).hexdigest() def create\_default\_users(self): | database\_manager.py:253 |
| create\_default\_users | Function | 257-334 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: default\_users = [ { 'nombr... | database\_manager.py:257 |
| authenticate\_user | Function | 335-368 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: password\_hash = self.hash\_... | database\_manager.py:335 |
| get\_all\_users | Function | 369-410 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: if tipo\_usuario: | database\_manager.py:369 |
| get\_all\_patients | Function | 411-438 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: | database\_manager.py:411 |
| get\_all\_doctors | Function | 439-465 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: | database\_manager.py:439 |
| get\_all\_appointments | Function | 466-493 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: | database\_manager.py:466 |
| get\_all\_invoices | Function | 494-519 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: | database\_manager.py:494 |
| get\_pending\_invoices | Function | 520-546 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: | database\_manager.py:520 |
| create\_appointment | Function | 547-580 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: # Crear fecha\_hora si vien... | database\_manager.py:547 |
| get\_appointment\_by\_id | Function | 581-620 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: | database\_manager.py:581 |
| update\_appointment | Function | 621-653 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: | database\_manager.py:621 |
| update\_appointment\_status | Function | 654-675 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: | database\_manager.py:654 |
| cancel\_appointment\_with\_reason | Function | 676-705 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: # Actualizar estado y agre... | database\_manager.py:676 |
| delete\_appointment | Function | 706-724 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: cursor.execute('DELETE FRO... | database\_manager.py:706 |
| get\_all\_patients | Function | 725-755 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: | database\_manager.py:725 |
| get\_all\_doctors | Function | 756-786 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: | database\_manager.py:756 |
| create\_invoice | Function | 787-815 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: | database\_manager.py:787 |
| pay\_invoice | Function | 816-838 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: | database\_manager.py:816 |
| get\_user\_by\_id | Function | 839-871 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: | database\_manager.py:839 |
| get\_medical\_insurances | Function | 872-892 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: | database\_manager.py:872 |
| get\_monthly\_income | Function | 893-920 | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: | database\_manager.py:893 |
| get\_available\_doctors\_global | Function | 922-936 | try: doctors = db\_manager.get\_all\_doctors() return [ { 'id': doc['id'], 'nombre\_completo': f"{doc['n... | database\_manager.py:922 |

## Funciones Principales de database\_manager.py

### 🔧 \_\_init\_\_

|  |  |
| --- | --- |
| Ubicación | database\_manager.py, líneas 67-73 |
| Tipo | Function |
| Signatura | def \_\_init\_\_(self, db\_path='database/medisync.db'): |
| Descripción | os.makedirs(os.path.dirname(self.db\_path), exist\_ok=True) def get\_connection(self): |

##### Primeras líneas del código:

def \_\_init\_\_(self, db\_path='database/medisync.db'):  
 self.db\_path = db\_path  
 self.lock = threading.Lock()  
 self.ensure\_database\_exists()  
 self.create\_tables()  
 self.create\_default\_users()

### 🔧 create\_tables

|  |  |
| --- | --- |
| Ubicación | database\_manager.py, líneas 88-252 |
| Tipo | Function |
| Signatura | def create\_tables(self): |
| Descripción | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: # Tabla usuarios |

##### Primeras líneas del código:

def create\_tables(self):  
 """Crear todas las tablas necesarias"""  
 with self.lock:  
 conn = self.get\_connection()  
 cursor = conn.cursor()  
   
 try:  
 # Tabla usuarios  
 cursor.execute('''  
 CREATE TABLE IF NOT EXISTS usuarios (

### 🔧 authenticate\_user

|  |  |
| --- | --- |
| Ubicación | database\_manager.py, líneas 335-368 |
| Tipo | Function |
| Signatura | def authenticate\_user(self, email, password): |
| Descripción | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: password\_hash = self.hash\_password(password) |

##### Primeras líneas del código:

def authenticate\_user(self, email, password):  
 """Autenticar usuario"""  
 with self.lock:  
 conn = self.get\_connection()  
 cursor = conn.cursor()  
   
 try:  
 password\_hash = self.hash\_password(password)  
 cursor.execute('''  
 SELECT id, nombre, apellido, email, telefono, tipo\_usuario, activo

### 🔧 get\_all\_users

|  |  |
| --- | --- |
| Ubicación | database\_manager.py, líneas 369-410 |
| Tipo | Function |
| Signatura | def get\_all\_users(self, tipo\_usuario=None): |
| Descripción | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: if tipo\_usuario: |

##### Primeras líneas del código:

def get\_all\_users(self, tipo\_usuario=None):  
 """Obtener todos los usuarios"""  
 with self.lock:  
 conn = self.get\_connection()  
 cursor = conn.cursor()  
   
 try:  
 if tipo\_usuario:  
 cursor.execute('''  
 SELECT id, nombre, apellido, email, telefono, tipo\_usuario, activo

### 🔧 get\_all\_patients

|  |  |
| --- | --- |
| Ubicación | database\_manager.py, líneas 411-438 |
| Tipo | Function |
| Signatura | def get\_all\_patients(self): |
| Descripción | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: |

##### Primeras líneas del código:

def get\_all\_patients(self):  
 """Obtener todos los pacientes"""  
 with self.lock:  
 conn = self.get\_connection()  
 cursor = conn.cursor()  
   
 try:  
 cursor.execute('''  
 SELECT u.id, u.nombre, u.apellido, u.email, u.telefono, u.tipo\_usuario, u.activo,  
 p.numero\_expediente, p.tipo\_sangre, p.alergias, p.contacto\_emergencia,

### 🔧 get\_all\_appointments

|  |  |
| --- | --- |
| Ubicación | database\_manager.py, líneas 466-493 |
| Tipo | Function |
| Signatura | def get\_all\_appointments(self): |
| Descripción | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: |

##### Primeras líneas del código:

def get\_all\_appointments(self):  
 """Obtener todas las citas"""  
 with self.lock:  
 conn = self.get\_connection()  
 cursor = conn.cursor()  
   
 try:  
 cursor.execute('''  
 SELECT c.\*,   
 up.nombre || ' ' || up.apellido as paciente\_nombre,

### 🔧 create\_appointment

|  |  |
| --- | --- |
| Ubicación | database\_manager.py, líneas 547-580 |
| Tipo | Function |
| Signatura | def create\_appointment(self, appointment\_data): |
| Descripción | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: # Crear fecha\_hora si viene por separado if 'fecha\_cita' in appointment\_data and 'hora\_cita' in appointment\_data: fecha\_hora = f"{appointment\_data['fecha\_cita']} {appointment\_data['hora\_cita']}" else: fecha\_hora = appointment\_data.get('fecha\_hora', '') |

##### Primeras líneas del código:

def create\_appointment(self, appointment\_data):  
 """Crear nueva cita"""  
 with self.lock:  
 conn = self.get\_connection()  
 cursor = conn.cursor()  
   
 try:  
 # Crear fecha\_hora si viene por separado  
 if 'fecha\_cita' in appointment\_data and 'hora\_cita' in appointment\_data:  
 fecha\_hora = f"{appointment\_data['fecha\_cita']} {appointment\_data['hora\_cita']}"

### 🔧 get\_all\_patients

|  |  |
| --- | --- |
| Ubicación | database\_manager.py, líneas 725-755 |
| Tipo | Function |
| Signatura | def get\_all\_patients(self): |
| Descripción | with self.lock: conn = self.get\_connection() cursor = conn.cursor() try: |

##### Primeras líneas del código:

def get\_all\_patients(self):  
 """Obtener todos los pacientes para combobox"""  
 with self.lock:  
 conn = self.get\_connection()  
 cursor = conn.cursor()  
   
 try:  
 cursor.execute('''  
 SELECT u.id, u.nombre, u.apellido   
 FROM usuarios u

# patient\_registration\_form.py - Formulario de Registro

Análisis detallado del archivo: patient\_registration\_form.py

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Función/Clase | Tipo | Líneas | Descripción | Ubicación |
| create\_patient\_registration\_form | Function | 14-112 | # Crear ventana de registro register\_window = tk.Toplevel(parent) register\_window.title("🩺 MEDISYNC... | patient\_registration\_form.py:14 |
| validate\_email | Function | 113-194 | Sin documentación | patient\_registration\_form.py:113 |
| toggle\_password | Function | 195-210 | Sin documentación | patient\_registration\_form.py:195 |
| check\_password\_strength | Function | 211-299 | Sin documentación | patient\_registration\_form.py:211 |
| cancel\_registration | Function | 328-331 | try: # Validar términos if not fields['acepta\_terminos'].get(): messagebox.showerror("Error", "Debe ... | patient\_registration\_form.py:328 |
| validate\_and\_register | Function | 332-407 | try: # Validar términos if not fields['acepta\_terminos'].get(): messagebox.showerror("Error", "Debe ... | patient\_registration\_form.py:332 |
| on\_mousewheel | Function | 450-455 | Sin documentación | patient\_registration\_form.py:450 |
| focus\_next\_widget | Function | 456-462 | section\_frame = tk.Frame(parent, bg='#f8f9fa') section\_frame.pack(fill='x', pady=(0, 20)) # Header ... | patient\_registration\_form.py:456 |
| create\_section | Function | 463-484 | section\_frame = tk.Frame(parent, bg='#f8f9fa') section\_frame.pack(fill='x', pady=(0, 20)) # Header ... | patient\_registration\_form.py:463 |
| create\_patient\_user | Function | 485-547 | try: conn = db\_manager.get\_simple\_connection() cursor = conn.cursor() # Verificar si el email ya ex... | patient\_registration\_form.py:485 |
| get\_seguro\_id\_by\_name | Function | 548-565 | if not seguro\_name or seguro\_name == 'Sin Seguro': seguro\_name = 'Sin Seguro' conn = db\_manager.get... | patient\_registration\_form.py:548 |

## Funciones Principales de patient\_registration\_form.py

### 🔧 create\_patient\_registration\_form

|  |  |
| --- | --- |
| Ubicación | patient\_registration\_form.py, líneas 14-112 |
| Tipo | Function |
| Signatura | def create\_patient\_registration\_form(parent, db\_manager): |
| Descripción | # Crear ventana de registro register\_window = tk.Toplevel(parent) register\_window.title("🩺 MEDISYNC - Registro de Nuevo Paciente") register\_window.geometry("900x800") register\_window.resizable(False, False) register\_window.grab\_set() register\_window.configure(bg='#f8f9fa') # Centrar ventana register\_window.transient(parent) register\_window.update\_idletasks() x = (register\_window.winfo\_screenwidth() // 2) - (900 // 2) y = (register\_window.winfo\_screenheight() // 2) - (800 // 2) register\_window.geometry(f"700x800+{x}+{y}") # ==================== HEADER MODERNO ==================== header\_frame = tk.Frame(register\_window, bg='#2c3e50', height=80) header\_frame.pack(fill='x') |

##### Primeras líneas del código:

def create\_patient\_registration\_form(parent, db\_manager):  
 """Crear formulario de registro de pacientes con diseño moderno y amigable"""  
   
 # Crear ventana de registro  
 register\_window = tk.Toplevel(parent)  
 register\_window.title("🩺 MEDISYNC - Registro de Nuevo Paciente")  
 register\_window.geometry("900x800")  
 register\_window.resizable(False, False)  
 register\_window.grab\_set()  
 register\_window.configure(bg='#f8f9fa')

### 🔧 validate\_and\_register

|  |  |
| --- | --- |
| Ubicación | patient\_registration\_form.py, líneas 332-407 |
| Tipo | Function |
| Signatura | def validate\_and\_register(): |
| Descripción | try: # Validar términos if not fields['acepta\_terminos'].get(): messagebox.showerror("Error", "Debe aceptar los términos y condiciones para continuar") return # Validar campos requeridos required\_fields = { 'nombre': 'Nombre', 'apellido': 'Apellido', 'email': 'Correo electrónico', 'telefono': 'Teléfono', 'password': 'Contraseña' } for field, label in required\_fields.items(): value = fields[field].get().strip() if not value: messagebox.showerror("Error", f"El campo '{label}' es obligatorio") |

##### Primeras líneas del código:

def validate\_and\_register():  
 """Validar y registrar paciente con validaciones avanzadas"""  
 try:  
 # Validar términos  
 if not fields['acepta\_terminos'].get():  
 messagebox.showerror("Error", "Debe aceptar los términos y condiciones para continuar")  
 return  
   
 # Validar campos requeridos  
 required\_fields = {

### 🔧 create\_patient\_user

|  |  |
| --- | --- |
| Ubicación | patient\_registration\_form.py, líneas 485-547 |
| Tipo | Function |
| Signatura | def create\_patient\_user(db\_manager, user\_data, patient\_data): |
| Descripción | try: conn = db\_manager.get\_simple\_connection() cursor = conn.cursor() # Verificar si el email ya existe cursor.execute("SELECT COUNT(\*) FROM usuarios WHERE email = ?", (user\_data['email'],)) if cursor.fetchone()[0] > 0: return False, "El email ya está registrado en el sistema" # Hash de la contraseña password\_hash = hashlib.sha256(user\_data['password'].encode()).hexdigest() # Insertar usuario |

##### Primeras líneas del código:

def create\_patient\_user(db\_manager, user\_data, patient\_data):  
 """Crear nuevo usuario paciente en la base de datos con manejo de errores mejorado"""  
 try:  
 conn = db\_manager.get\_simple\_connection()  
 cursor = conn.cursor()  
   
 # Verificar si el email ya existe  
 cursor.execute("SELECT COUNT(\*) FROM usuarios WHERE email = ?", (user\_data['email'],))  
 if cursor.fetchone()[0] > 0:  
 return False, "El email ya está registrado en el sistema"

# RESUMEN DE UBICACIONES IMPORTANTES

## 📍 MEDISYNC.py

|  |  |
| --- | --- |
| Líneas | Contenido |
| Líneas 1-30 | Importaciones y configuración inicial |
| Líneas 50-65 | Constructor de la clase MedisyncApp |
| Líneas 75-200 | Ventana de login y autenticación |
| Líneas 240-260 | Creación de ventana principal |
| Líneas 270-350 | Menú de administrador |
| Líneas 400-500 | Dashboard y estadísticas |
| Líneas 600-800 | Gestión de usuarios |
| Líneas 900-1200 | Gestión de citas médicas |
| Líneas 1300-1600 | Formulario de nueva cita |

## 📍 database\_manager.py

|  |  |
| --- | --- |
| Líneas | Contenido |
| Líneas 1-25 | Importaciones y modelos de datos |
| Líneas 50-60 | Constructor DatabaseManager |
| Líneas 70-200 | Creación de tablas |
| Líneas 250-280 | Autenticación de usuarios |
| Líneas 300-380 | Consultas de usuarios y pacientes |
| Líneas 400-430 | Consultas de doctores |
| Líneas 500-580 | Gestión de citas |
| Líneas 650-680 | Sistema de facturación |

## 📍 patient\_registration\_form.py

|  |  |
| --- | --- |
| Líneas | Contenido |
| Líneas 15-50 | Función principal de registro |
| Líneas 60-150 | Sección de información personal |
| Líneas 160-220 | Sección de información médica |
| Líneas 230-280 | Contacto de emergencia |
| Líneas 300-350 | Validación y procesamiento |
| Líneas 450-500 | Creación en base de datos |