



Carrera: Técnico Universitario en Programación
Materia: Programación II
Tema: TP Final

Integrantes de la cátedra

Kloster, Daniel – Profesor
Carbonari, Verónica - Jefe de Trabajos Prácticos
Lara Campos, Brian - Jefe de Trabajos Prácticos
Wenner, Maximiliano – Jefe de Trabajos Prácticos

TP Final Programación II. 2025 1C Grupo 6

Integrantes del grupo:

Julieta Rodriguez (Leg:32487)
Gabriel Manattini (Leg: 32406)
Sebastian Ravenna (Leg: 32446)

Tipo de proyecto: Sistema de gestión de archivos

Nombre: Sana Sana Soft

Tema: Sistema de gestión de un consultorio médico.

Descripción: El sistema tendrá por objetivo la administración de los turnos médicos brindados en una clínica. Mediante su uso, el usuario tendrá la posibilidad de dar de alta, consultar o modificar las plantillas de pacientes, médicos y obras sociales. De esta manera, y teniendo en cuenta lo que solicita el paciente, se buscará asignar el turno correspondiente.

Además de la agenda, el sistema permitirá gestionar la facturación de la clínica, llevando no solo el control de las transacciones individuales (ya sean de forma particular o por obra social), sino también el registro global de todas las operaciones las cuales podrán visualizarse mediante reportes.

De esta manera, el sistema brindará de forma rápida y precisa, información útil para la gestión y toma de decisiones empresariales.

Nuestro sistema contará con un menú inicial compuesto por las ocho opciones principales a saber:

MENÚ PRINCIPAL

1. Turno
 2. Paciente
 3. Médico
 4. Obra Social
 5. Facturación
 6. Listados
 7. Reportes
 0. Salir
-

La opción “Turno” será utilizada para gestionar todo aquello relacionado a la administración de turnos:

1. TURNO

1. Crear Turno
 2. Buscar Turno
 - Por DNI del paciente
 3. Modificar Turno (solo por DNI)
 4. Cancelar Turno (solo por DNI)
 0. Volver al Menú Principal
-

La opción “Paciente” servirá para administrar todas aquellas cuestiones que involucren alta, baja o modificación de la plantilla de pacientes. Además, desde aquí se podrá acceder y modificar las historias clínicas:

2. PACIENTE

1. Alta de Paciente
2. Buscar Paciente
 - Por DNI

3. Modificar Paciente (solo por DNI)
4. Baja de Paciente (solo por DNI)
5. Historia Clínica
 - Crear Historia Clínica
 - Buscar Historia Clínica
 - Modificar Historia Clínica
0. Volver al Menú Principal

La opción “Médico” servirá para administrar todas aquellas cuestiones que involucren alta, baja o modificación de la plantilla de médicos y sus respectivas agendas:

3. MÉDICO

1. Alta de Médico
2. Buscar Médico
 - Por DNI
3. Modificar Médico (solo por DNI)
4. Baja de Médico (solo por DNI)
5. Agenda
0. Volver al Menú Principal

La opción “Obra Social” permitirá al usuario acceder y modificar toda aquella información relacionada con las Obras Sociales relacionadas con la clínica:

4. OBRA SOCIAL

1. Alta de Obra Social
2. Buscar Obra Social
 - Por nombre
3. Modificar Obra Social
4. Baja de Obra Social
0. Volver al Menú Principal

La opción “Facturación” servirá para administrar todos los asuntos relacionados con la facturación de los servicios médicos prestados. Además, permitirá al usuario obtener reportes sobre la recaudación según el rango de fecha seleccionado:

5. FACTURACIÓN

1. Crear Factura
2. Buscar Factura
 - Por ID de Factura
3. Modificar Factura (solo modifica el estado)
0. Volver al Menú Principal

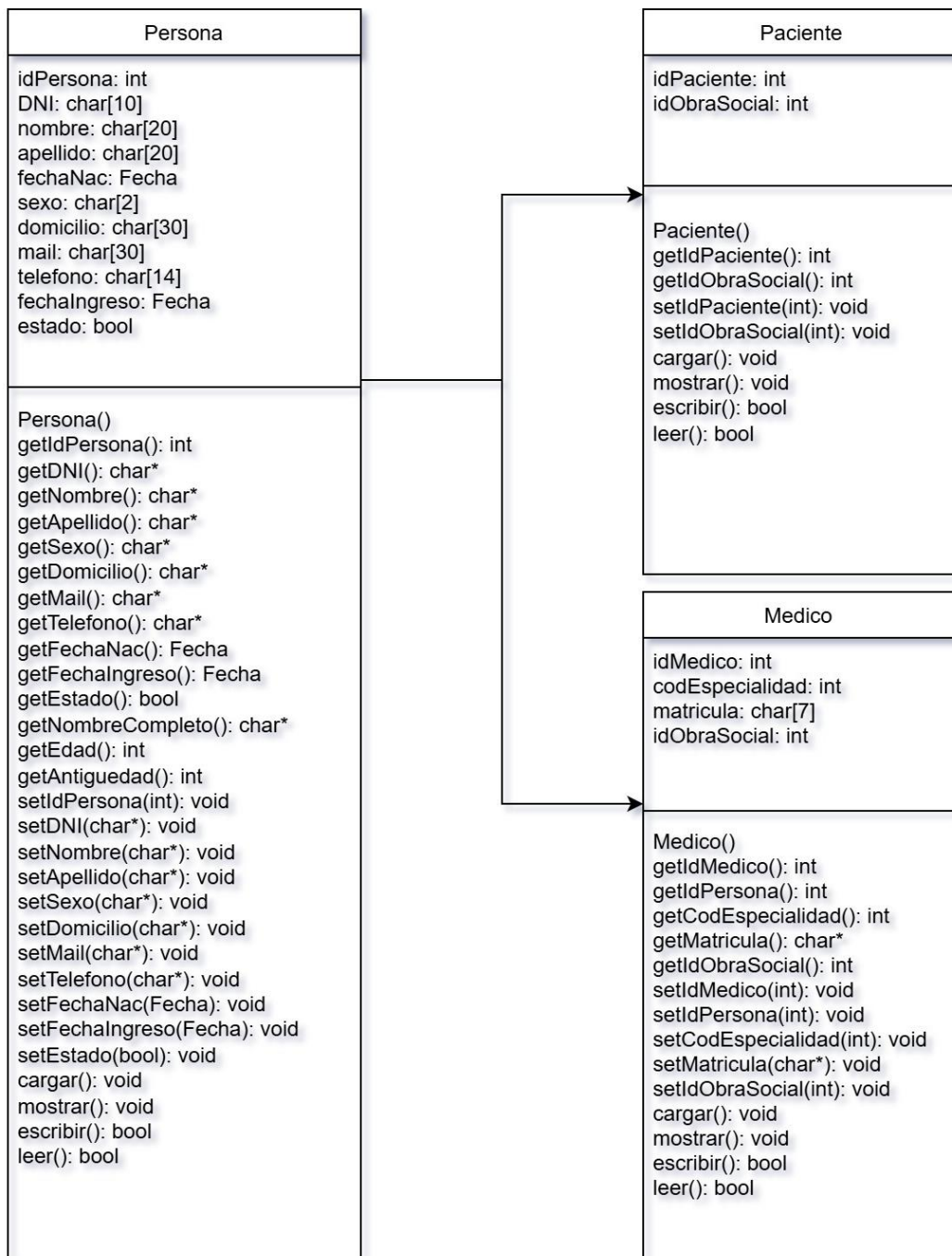
6. LISTADOS

1. Listar Turnos
 - Turnos Libres por Médico (según rango de fechas)
 - Turnos Libres por Especialidad (según rango de fechas)
 - Turnos Ocupados por Médico (según rango de fechas)
 - Turnos Ocupados por Especialidad (según rango de fechas)
 - Todos los Turnos (según rango de fechas)
2. Listar Pacientes
 - Listar Todos (por apellido)
 - Listar Activos (por apellido)
 - Listar Inactivos (por apellido)
3. Listar Historias Clínicas
 - Listar todas las historias clínicas (ordenadas por fecha)
 - Listar todas las historias clínicas (ordenadas por DNI)
4. Listar Médicos
 - Listar Todos (Activos/Inactivos, por apellido)
 - Listar por Especialidad (solo Activos, por apellido)
5. Listar Obras Sociales
 - Listar Todas (por orden alfabético)
 - Listar Activas (por orden alfabético)
 - Listar Inactivas (por orden alfabético)
6. Listar Facturas
 - Listar todas las facturas (ordenadas por fecha)
0. Volver al Menú Principal

7. REPORTES

1. Facturación
 - Recaudación Anual
 - Recaudación por Médico
 - Recaudación por Obra Social
 - Recaudación por Consulta Particular
 - Recaudación por Especialidad
2. Pacientes
 - Pacientes atendidos por Especialidad
 - Atención Anual de Pacientes
3. Médicos
 - Consulta Anual por Médico

Clases a desarrollar:



Especialidad
codEspecialidad: int descripcion: char[30] valorConsulta: float estado: bool
Especialidad() getCodEspecialidad(): int getDescripcion(): char* getValorConsulta(): float getEstado(): bool setCodEspecialidad(int): void setDescripcion(char*): void setValorConsulta(float): void setEstado(bool): void cargar(): void mostrar(): void escribir(): bool leer(): bool

ObraSocial
idObraSocial: int nombreObraSocial: char[30] estado: bool
ObraSocial() getIdObraSocial(): int getNombreObraSocial(): char* getEstado(): bool setIdObraSocial(int): void setNombreObraSocial(char*): void setEstado(bool): void cargar(): void mostrar(): void escribir(): bool leer(): bool

Turno
idTurno: int idAgenda: int idPaciente: int estado: bool
Turno() getIdTurno(): int getAgenda: int getIdPaciente(): int getEstado(): bool setIdTurno(int): void setAgenda: int setIdPaciente(int): void setEstado(bool): void cargar(): void mostrar(): void escribir(): bool leer(): bool

Agenda
idAgenda: int idMedico: int codEspecialidad: int fechaTurno: Fecha horaTurno: Hora asignado: bool estado: bool
Agenda() getIdAgenda(): int getIdMedico(): int getCodEspecialidad(): int getFechaTurno(): Fecha getHoraTurno(): char* getAsignado(): bool getEstado(): bool setIdTurno(int): void setIdMedico(int): void setCodEspecialidad(int): void setFechaTurno(Fecha): void setHoraTurno(char*): void setAsignado(): bool setEstado(bool): void cargar(): void mostrar(): void escribir(): bool leer(): bool

Fecha
dia: int mes: int anio: int
Fecha() Fecha(int d, int m, int a) getDia(): int getMes(): int getAnio(): int setDia(int): void setMes(int): void setAnio(int): void toString(): char* esValida(): bool cargar(): void mostrar(): void escribir(): bool leer(): bool

Hora
hora: int minutos: int
Hora() Hora(int h, int m) getHora(): int getMinutos(): int setHora(int): void setMinutos(int): void toString(): char* esValida(): bool cargar(): void mostrar(): void escribir(): bool leer(): bool

Factura
idFactura: int fechaFacturacion: Fecha dniPaciente: char[10] idMedico: int codEspecialidad: int importe: float tipoFactura: char[2] estado: bool
Facturacion() getIdFactura(): int getFechaFacturacion(): Fecha getDniPaciente(): char* getIdMedico(): int getCodEspecialidad(): int getImporte(): float getTipoFactura(): char* getEstado(): char* setIdFactura(int): void setFechaFacturacion(Fecha): void setDniPaciente(char*): void setIdMedico(int): void setCodEspecialidad(int): void setImporte(float): void setTipoFactura(char*): void setEstado(char*): void cargar(): void mostrar(): void escribir(): bool leer(): bool

HistoriaClinica
idHistoria: int idPaciente: int idMedico: int fechaConsulta: Fecha descripcion: char[300] estado: bool
HistoriaClinica() getIdHistoria(): int getIdPaciente(): int getIdMedico(): int getFechaConsulta(): Fecha getDescripcion(): char* getEstado(): bool setIdHistoria(int): void setIdPaciente(int): void setIdMedico(int): void setFechaConsulta(Fecha): void setDescripcion(char*): void setEstado(bool): void cargar(): void mostrar(): void escribir(): bool leer(): bool