# **INFORME DE PROYECTO**

# Sistema de Gestión de Salud y Medicamentos



## Integrantes:

Pablo, Rodríguez Olsavsky Ricardo, Adducci Recofsky

## INTRODUCCIÓN

El sistema a desarrollar permitirá gestionar los planes farmacológicos que utilizan los usuarios de la aplicación. Para esto contará con un menú principal y una serie de submenú que serán la interfaz gráfica para la gestión de los medicamentos.

## DESCRIPCIÓN DETALLADA DEL SISTEMA

El objetivo principal del sistema es ayudar a los usuarios (pacientes), a poder administrar los planes farmacológicos, los cuales serán recetados por Profesional.

El sistema será utilizado por dos tipos de usuarios en una primera versión:

- 1) Profesional: Quienes pueden emitir planes farmacológicos
- 2) Pacientes: Quienes llevan a cabo el plan farmacológico

Por lo que se deduce, que los accesos a los menúes variarán en base al tipo de usuario logueado.

Un Determinado paciente realizará una consulta profesional a un Profesional, el mismo le generará en el sistema un plan farmacológico y lo exportará como un archivo que le otorgará al paciente a través de un medio de almacenamiento como por ejemplo un pendrive o por correo electrónico. El paciente comprará los medicamentos recetados para llevar adelante el plan farmacológico (también puede comprar medicamentos de venta libre) y cargará la información detallada de los medicamentos comprados en el sistema.

El Profesional puede modificar el plan farmacológico o indicar algún refuerzo necesario.

Cada vez que el paciente ingrese al sistema, el mismo leerá la fecha local y en base al plan farmacológico en curso, realizará diversas comprobaciones para ayudar al paciente a llevar a cabo su plan de forma correcta.

El sistema verificará y notificará:

- 1) Para cuantos dias le queda medicación
- 2) El stock de medicamentos disponibles

El sistema emitirá alertas de tipo:

- 1) Informativas: cuando quede poco de algún medicamento..
- 2) De advertencia:
  - a. si con el stock que posee no llegara a completar el plan en curso.
  - b. Si se actualizó el plan y el stock no se corresponde con el nuevo plan.

Además de las alertas, el sistema permitirá realizar o exportar una serie de reportes en los cuales podrá consultar:

- 1) El avance del plan farmacológico.
- 2) La cantidad (stock) de medicamentos adquirido.

Para la versión 2.0 del sistema, se desarrollará una funcionalidad que permitirá consultar una web para extraer información de la misma, en la cual se expondrá información relacionada a los descuentos sobre los medicamentos de venta libre otorgados por diversas obras sociales y prepagas

## **SALIDAS DEL SISTEMA**

La aplicación elaborará tres tipos de salidas. Las mismas se categorizan en Listados.

## **INFORMES**

A desarrollar

## **CONFIGURACIONES**

A desarrollar

## **LISTADOS**

Logueado Como Profesional:

- Listado de Planes farmacológicos realizados
  - o Por fecha
  - o Por apellido de paciente
- Listado de pacientes
  - Por Apellido
  - o Por DNI
  - Por obra social
  - o Por número de historia clínica
- Listado de estudios realizados
  - Por historia clínica

## Como paciente

- Listado de Medicación actual
  - Ordenados por stock
  - Ordenados por fármaco
  - Ordenados por nombre
- Listado de Profesional
  - Por especialidad
  - Por apellido
  - o Por matrícula
- Listado de estudios realizados

## **CONSULTAS**

## Logueado Como Profesional:

- Consultas de pacientes por
  - o DNI
  - o Rango de edad
  - Apellido
  - Obra social/Prepaga
  - Historia Clínica
  - o Por fecha
- Consulta de Planes farmacológicos realizados
  - o Por Apellido del paciente
  - o Por DNI del paciente
  - Por obra social
  - o Por historia clínica
  - Por fecha
- Historia Clínica
  - Por Apellido del paciente
  - o Por fecha
  - o Por DNI
  - o Por número de historia clínica
- Consulta de Estudios realizados
  - o Por apellido del paciente
  - o Por fecha
  - o Por DNI
  - o Por número de historia clínica

## Logueado como paciente:

- Consulta de medicación:
  - Por fármaco
  - o Por marca
- Consulta de Estudios
- Consulta de planes farmacológicos
- Profesional
  - o Por matrícula
  - o Por Apellido
  - Por especialidad

## **INFORMES**

A continuación se detallan los informes que el sistema generará

- Medicaciones consumidas
- Consultas realizadas
- Planes farmacológicos realizados

- Estudios Realizados
- Pacientes atendidos
- Profesionales consultados

## ANEXO I: DIAGRAMA DE CLASES

#### Fecha

- dia: int
- mes: int anio: int
- + getDia(): int

- + getMes(): int + getAnio(): int + setDia(valor : int): void
- + setMes(valor : int): void + setAnio(valor : int): void

#### Informes

- · ID\_Informe: int · historia: \*char
- numero: int
- fecha: Fecha
- ID\_Profesional: int
- ID Paciente: int
- + getID(): int + getHistoria(): char\*
- + getNumero(): int + getFecha(): Fecha
- + getProfesional(): Profesional
- + getPaciente(): Paciente + setID(valor : int): void
- + setHistoria(valor : char): void + setNumero(valor : int): void

**TipoDeProfesional** 

- + setFecha(valor: Fecha): void + setProfesional(valor: Profesional): void
- + setPaciente(valor : Paciente): void

## Profesional

+ setID(valor: int) : void + setIdEspecialidad(valor: Especialidad) : void

+ aetMatricula(): int getID(): int

+ getIdEspecialidad() : Especialidad + setMatricula(valor : int): void

- ID\_TipoDeProfesional: int matricula: int nombre: char[50] ID\_TipoDePorfesional : int ID\_Especialidad : int
- + aetID0: int
- + getNombre(): criai + setID(valor: int): void
- + setNombre(valor : char\*): void

### ObraSocial

- ID ObraSocial: int
- nombre: char[50]
- + getID(): int
- + getNombre(): char\* + setID(valor : int): voi
- setID(valor:int):void
- setNombre(valor : char\*): void

### Medicacion

- ID\_Medicacion: int
- farmaco: char[50]
- marca: [50]
- dosisMg : float horario\_medicacion: int
- ID\_Presentacion: int
- estado: bool
- vencimiento : Fecha
- + getID(); int + getFarmaco(); char\*
- + getMarca(): char\* + getDosis(): float
- + getHorariosMedicacion() : int + getPresentacion() : int
- + getEstado() : bool + getVencimiento() : Fecha

- + getvencrmento() : Fecha + setfD(valor : int): void + setFarmaco(valor : char\*): void + setMarca(valor : char\*): void + setDosis(valor : float) : void + setHorariosMedicacion(valor : int) : void
- + setPresentacion(valor : Presentacion) : voi
- + setEstado(valor : bool) : void + setVencimiento (valor : Fecha) : void

### Persona

- # ID: int
- # nombres: char[50]
- # apellidos: char[50
- # fecha\_nacimiento: Fecha]
- #ID\_Genero: int # edad: int
- # estado: bool
- # DNI: int
- + getID(): int
- + getNombres(): char\* + getApellidos(): char\*
- + getFechaNacimeinto(): Fecha
- + aetGenero(): int
- + getEdad(): int
- + getDNI(): int
- + getEstado(): bool

- + getEstatoo(, bool + setID(not)r: int); void + setNombres(valor: char\*); void + setApellidos(valor: char\*); void + setFechaNacimeinto(valor: Fecha); void
- + setGenero(valor : Genero):void
- + setEdad(valor : int): void + setDNI(valor : int): void
  - D Extends Extends

## Genero

- ID\_Genero: int nombre : char[50]
- + getID0: int
- + getNombre(): char\*
- + setID(valor: int): void + setNombre(valor: char\*): void

#### Especialidad

- ID\_Especialidad: int nombre: char[50]
- + aetID(): int
- + getNombre(): char\*
- + setID(valor: int): void + setNombre(valor: char\*): void

## Paciente

- nro\_afiliado: int
- ID ObraSocial: int plan\_farmacologico: PlanFarmacologico
- historia\_clinica : char 1 nro historia: int
- + getNroAfiliado(): int
- + getNoAlinado(): Int + getObraSocial(): ObraSocial + getPlanFarmacologico(): PlanFarmacologico
- + getHistoriaClinica(): char\* + getNroHistoria(): int
- setNroAfiliado(valor : int): void + setObraSocial(valor : ObraSocial): void
- + setPlanFarmacologico(valor: PlanFarmacologico): void
- + setHistoriaClinica(valor: char\*): void + setNroHistoria(valor: int): void

## PlanFarmacologico

- ID\_PlanFarmacologico: int
- emision: Fecha
- actualizacion: Fecha ID Paciente: int
- ID\_Profesional: int
- ID\_TipoMedicacion: int
- estado: hool
- + getID() : int
- + getEmision() : Fecha + getActualizacion() : Fe getActualizacion() : Fecha
- + getPaciente() : Paciente + getProfesional() : Profesional
- + getTipoMedicacion(): TipoMedicacion
- + getNotas() : char\* + getEstado() : bool
- + getEstado() : bool + setID(valor: int) : void
- + setEmision(valor: Fecha): void + setActualizacion(valor: Fecha): void + setPaciente(valor: Paciente): void
- + setProfesional(valor : Profesional) : void + setTipoMedicacion(valor: TipoMedicacion) : void
- setNotas(valor: char\*): void setEstado(valor: char\*): void

- **TipoMedicacion**
- ID TipoMedicacion: int nombre: char[50]
- + getID(): int
- + getNombre(): char\* + setID(valor : int): void
- + setNombre(valor : char\*): void

## Presentacion

- ID\_Presentacion: int
- nombre: char[50]
- + getID(): int + getNombre(): char\*
- + setID(valor: int): void + setNombre(valor: char\*): void