

Universidad del Valle de Guatemala
Programación Orientada a Objetos
Ingeniería en Ciencias de la Computación y TI
Sección 10 – Ciclo 2021

Segunda Fase de Proyecto

Integrantes:

Abner Iván García Alegría - 21285
Sebastián José Solorzano Pérez - 21826
Javier Aejandro Prado Ramírez - 21486
Angel Gabriel Perez Figueroa – 21298

Identificación de requisitos Funcionales

Funcionalidad del Programa:

1. Brindar a los usuarios una herramienta que les facilite el registro de datos de sus pacientes
2. Calendarizar las veces de medicación de cada paciente según sea su necesidad
3. Registrar la información necesaria de cada paciente, obteniendo cuales son sus problemas, síntomas, medicamentos en necesidad, ayudas, etc.

Lista de opciones:

1. Registro de Persona: Se le solicita al usuario; nombre, DPI, edad, sangre, # de sala, medico y sexo, estos datos son la información fundamental para el registro de todas las personas.
2. Buscar Persona: Se le buscará a determinado paciente, ya siendo por nombre completo o por número de DPI
3. asignación de Medicamento: Se le solicita al médico, la información del nombre del medicamento, tiempo de medicamento en el paciente, veces al día del medicamento y las horas recetadas.
4. Mostrar Resultados: Se le muestra al usuario la información requerida de manera organizada y almacenada en un archivo para mantener un registro de toda la información requerida por el paciente.
5. Salida: Fin del programa con toda la información que se ha sido ingresada en el programa.

Priorización de los requisitos funcionales encontrados:

Para contextualizar el desarrollo del programa:

Prioridad Fundamental 1: que los datos que han sido ingresados en el programa sean guardados con éxito en un archivo, para facilitar el conocimiento de la información almacenada. En caso de haber cambios de turnos por parte de los médicos y/o enfermeros(as), puedan conocer los datos y no perder el control que ha sido guardado con anterioridad.

Asimismo, que los datos de los pacientes sean ingresados correctamente, para no generar confusiones en los demás doctores que tomarán en consecuencia la información de los pacientes.

Identificación y Descripción de clases

Clase Display (Funcionalidad: A través de esta clase se llevará a cabo la impresión de todas las respuestas en que el usuario pueda interactuar, se implementará un menú que servirá como referencia para conocer la información necesaria de cada paciente)

Propiedades	Visibilidad	Métodos	Visibilidad
Scanner	Privada	Directory: String	Publica
Arraylist	Privada	Menu: int	Publica
ClaseFile		Mismatch(): void	Publica
		Espacio: String[]	Publica
		PacientData: String[]	Publica
		Show: void(String)	Publica

Clase Main (Funcionalidad: Recogerá los valores introducidos en la ejecución del programa. Es el que servirá como punto de partida para el funcionamiento del programa. En otras palabras, controlará el programa direccionando las llamadas a las otras clases (Manipulador del archivo)).

Imports	Métodos
IOException	Main(String[] args)
FileWriter	
BufferedReader	
ClaseFile	

Composiciones: InputMismatchException

Clase Medicación (Funcionalidad: Tiene como objetivo principal el control y análisis de los medicamentos que el/la paciente debe ingerir, según sea su caso. La única responsabilidad de esta clase es asignar un control médico al paciente en cuestión)

Propiedades	Visibilidad	Métodos	Visibilidad
Name: String	Privada	setDiagnostico	Publica
Time: float	Privada	setMedicación	Publica
Ingerir: boolean	Privada	getTime: String	Publica
Medicación: String	Privada	setTime: float	Publica
ClaseFile			

Agregaciones: Arraylist

Clase Paciente (Funcionalidad: Información completa del paciente, es decir, todos los datos que se anotarán en el registro proporcionado con anticipación, los datos que se tomarán tendrán cohesión con la clase medicación, ya que a través de esta información se podrá realizar el proceso de medicamento a cada paciente.)

Propiedades	Visibilidad	Métodos	Visibilidad
Nombre: String	Privada	Paciente (String[] Data)	Publica
DPI: int	Privada	getData: String	Publica
Sangre: String	Privada	getDiagnostico; String	Publica
Edad: int	Privada	getEdad: int	Publica
NumSala: int	Privada	getSangre: String	Publica
Sexo: String	Privada		
Diagnostico: String	Privada		
Clase File			

Errores (Exception) esperados:

InputMismatch

cuando el usuario ingrese un tipo de dato erróneo, es decir que se le esté pidiendo un número e ingrese una letra. Si eso sucede, por medio de un catch, se evita que el programa colapse, indicándole al usuario que su entrada de datos no es válida y que intente de nuevo. Un claro ejemplo puede ser en la implementación del menú de opciones y cuando se crean instancias con los datos dados por el médico, enfermero(a).

IOException

Cuando se interrumpe de manera forzosa el registro de datos. Para esto se le puede avisar al usuario que el archivo que ha sido generado puede contener problemas de ejecución, para evitar esto se le informará al usuario que el archivo está mal, y se le pedirá que sea eliminado del directorio donde se tiene guardado.

FileNotFoundException

Al usar el programa por primera vez no existirá un archivo con toda la información, por ende, en caso de no se haya creado, el programa autogenerará uno para su implementación. De esta manera al hacer uso del programa, ya se tendrá un archivo con la información requerida.