



Universidad Andrés Bello

Proyecto “PET Manager” Seminario de Título 1

Sebastián Inostroza Hurtado

INGENIERIA EN COMPUTACION E INFORMATICA

PROFESOR GUIA: PATRICIO CASTILLO

VIÑA DEL MAR - CHILE

11 ENERO 2019

FORMAR

TRANSFORMAR

Contenidos

1.	1. Introducción	6
2.	Fundamentación del Problema	8
2.1.	Análisis de la situación actual.	8
2.2.	Análisis de la problemática.	8
2.2.1.	Técnica de los cinco por qué.	9
2.2.2.	Técnica de Ishikawa.	10
2.3.	Objetivos.	11
2.3.1.	Objetivo general.	11
2.3.2.	Objetivos específicos.	11
2.4.	Diagrama de alto nivel.	12
2.5.	Arquitectura de la solución.	13
2.6.	Solución ideal.	14
2.6.1.	Alcances y Limitaciones.	14
2.6.2.	Restricciones.	14
2.7.	Alternativas de solución.	15
2.7.1.	Procedimientos manuales.	15
2.7.2.	Cambios en procedimientos actuales.	15
2.7.3.	Alternativas disponibles en el mercado.	15
2.8.	Solución propuesta.	16
2.8.1.	Factibilidades.	16
2.8.2.	Diseño de alto nivel.	17
2.8.3.	Requerimientos de alto nivel.	20
3.	Planificación del proyecto.	22
3.1.	Metodología de desarrollo.	22
3.2.	Planificación del proyecto.	22
3.2.1	Backlog Proyecto PET Manager.	25
3.3.	Gestión de la configuración.	28
3.4.	Gestión de cambios.	29
3.5.	Gestión de riesgos.	31
3.6.	Entorno de desarrollo.	35
4.	Resultados .	37

4.1. Análisis de resultados.	37
4.2. Diseño detallado.	58
4.3. Evidencia de control de cambios. .	65
4.4. Situación futura.	67
4.4.1. Resultados esperados.	68
5. Conclusiones.	70
5.1. Conclusión.	70
5.2. Problemas abiertos.	71
5.3. PostMortem.	71
5.4. Trabajo futuro.	73

Resumen

El año pasado se inauguró el primer centro especializado para la realización de los exámenes PET de la quinta región ubicado en Clínica Bupa Reñaca, ellos solicitan el radiofármaco al ciclotrón más cercano el cual que se encuentra ubicado en Santiago. El tiempo de despacho del radiofármaco es aproximadamente de 2 horas y varía mucho dependiendo del día y del tráfico, pues el tránsito es vía terrestre con límites de velocidad establecidas por ley, por lo que no hay una hora segura de llegada.

La Empresa del ciclotrón se compromete a entregar a las 10:30 am un mínimo de 110 mci (mili-curies) valor el cual puede fluctuar para más o menos su valor, pues no siempre el radiofármaco llega a tal hora. Además puede existir la posibilidad de solicitar mayor cantidad de dosis para poder realizar mayor número de pacientes.

Es imprescindible tener un conteo específico del radiofármaco para poder organizar la agenda de pacientes y así poder garantizar que los pacientes citados se podrán realizar.

Actualmente se trabaja con un activímetro que mide la radiación que llega, y que se dosifica para ser inyectada a cada paciente. Esta manipulación de la dosis es realizada por los Tecnólogos Médicos, que deben ir calculando la dosis que se les inyecta a cada paciente e ir dosificando de mejor manera el Radiofármaco, con el fin de que cada uno de éstos pueda recibir la dosis adecuada. En una Pizarra se anota la Hora de Inyección del fármaco al paciente, la hora en la cual el paciente debe ingresar al examen , estos tiempos de deben adecuar a tiempos entre pacientes y del uso de la máquina PET.

El problema actual radica en que para tener un conteo preciso de radiofármaco, se deben calcular muchas variables que dificultan la fluidez del trabajo, además para tener un conteo preciso se debe medir el frasco de origen cada vez para ir chequeando que los valores de radiación serán adecuados para los pacientes, siendo al final del día un poco incierto la capacidad de hacer o no un paciente extra o incluso un paciente programado.

Ante ésto, la necesidad de crear una aplicación que permita gestionar los tiempos de trabajo y entregue un valor de dosis sin necesidad de exponer al personal más de lo adecuado, es algo imperante en este servicio.

Capítulo 1: Introducción

1. 1. Introducción

El PET (Tomografía por Emisión de Positrones) es una técnica de diagnóstico no invasiva que permite realizar imágenes que muestran el metabolismo y el funcionamiento de tejidos y órganos, basándose en el consumo de glucosa.

Todos los tejidos en el cuerpo humano consumen glucosa, pero aquellos en los que se está produciendo una proliferación maligna consumen mucho más. Inyectando vía endovenosa al paciente una solución de glucosa con un marcador radioactivo (pero inocuo) y pasándolo por el PET, éste recoge la radiación que en ese momento emite.

Esa radiación permite al sistema operativo del PET dibujar un mapa del cuerpo en el que se recogen, si las hubiera, las mayores concentraciones de glucosa, de modo que el médico puede determinar la malignidad de un bulto ya conocido o incluso detectar metástasis en sus grados más incipientes.

El PET se utiliza fundamentalmente para comprobar la malignidad de los tumores. Sucede que las células tumorales, por su metabolismo oxidativo, necesita más glucosa. Para detectar dónde están o cómo actúan, se introducen radiofármacos como la fluordesoxiglucosa, o f18-fdg, que permite detectar casi todos los tumores y cuya radiación decae a la mitad en un tiempo de 109,77 minutos aprox.

Las sustancias radiactivas se desintegran con mayor o menor rapidez, según de cual se trate. Se llama vida media al tiempo requerido para que la mitad de una sustancia desaparezca. Por ejemplo, la vida media del Radio es de 1620 años. De esta forma, si tenemos 500gr. de Radio, al cabo de 1620 años habrá sólo 250gr. 1620 años después, es decir dentro de 3240 años, habrá 125gr. Y así sucesivamente. A lo largo de cada semiperiodo, la radiactividad desciende primero a la mitad, luego a una cuarta parte, etc.

La mayoría de los radionucleidos emisores de positrones presentan vidas medias extremadamente cortas, lo cual necesita su producción cerca del sitio de utilización. Todos son producidos por ciclotrón en vez de reactor. Los más usados son el Flúor-18, el Carbono-11, el Oxígeno-15 y el Nitrógeno-13, todos los cuales son capaces de marcar moléculas orgánicas fisiológicamente utilizadas por el metabolismo celular. La única excepción para la necesidad de producción cerca del sitio de aplicación es el Flúor-18, el que por su mayor vida media puede ser usado en lugares distantes 1 – 2 horas del ciclotrón.

Para PET es la necesidad de una química automatizada de alta velocidad para asegurar que el marcado se pueda lograr con mínima exposición radiactiva de los operadores y una pérdida aceptable de actividad por decaimiento antes de ser administrada al paciente.

Capítulo 2: Fundamentación del problema

2. Fundamentación del Problema

2.1. Análisis de la situación actual.

Actualmente, en clínica Bupa Reñaca, se realizan 5 pacientes confirmados diariamente, teniendo una lista de espera de 2 semanas para pacientes ambulatorios. Los pacientes urgentes u hospitalizados son incorporados dentro de estos 5 pacientes, pero bajo condiciones excepcionales se acepta como 6to paciente teniendo que tener extremado cuidado con la dosis a repartir para los pacientes, ocupando el mínimo para “asegurarnos” de que alcanzara la dosis al final de la jornada. Esta mínima cantidad influye con la calidad del examen, debiendo realizar un examen de mayor duración para compensar esta falta de dosis y al hacer el examen de mayor duración va repercutiendo en los pacientes futuros, pues la dosis para éstos va disminuyendo, creando una reacción en cadena, el cual concluye en que al final de la jornada puede sobrar o faltar radiofármaco y por ende no realizar el 6to paciente. La forma de asegurarnos de que podemos realizar cierta cantidad de pacientes es midiendo entre cada paciente la actividad (dosis) que va quedando en el contenedor principal y poder realizar un fraccionamiento correcto de las dosis entre los pacientes.

2.2. Análisis de la problemática.

Para identificar la problemática utilizamos la técnica de los 5 ¿Por qué? de forma extendida, buscando las posibles causas y el diagrama de Ishikawa que denotara de mejor forma los actores principales.

“El número de pacientes diarios a examinar es impreciso ocupando las técnicas de trabajo actual y medidas de radioprotección”.

2.2.1. Técnica de los cinco por qué.

Problema	Porque? 1	Porque? 2	Porque? 3	Porque? 4	Porque? 5	Solución
Imprecisión en numero de pacientes diarios	1- No hay certeza de la actividad del radiofármaco	1.1- No se realizan cálculos.	1.1.1- Cálculos Engorrosos	1.1.1.1- Muchas variables		Herramienta generadora de cálculos
			1.1.2- Toman mucho tiempo	1.1.2.1- Repetitivas.		Herramienta generadora de cálculos
		1.2- Medir la actividad del frasco contenedor	1.2.1- El fármaco emite radiación constantemente	1.2.1.1- Para Irradiar lo menos posible al operador	1.2.1.1.1- Leyes de radio protección. 1.2.1.1.2- Es perjudicial	Evitar Exposición
	2-No hay certeza de cuanta dosis llegara	2.1-El radiofármaco viene de Santiago	2.1.1-Producción solo en Santiago	2.1.1.1- Alto costo y mant.		X
			2.1.2- Transporte vía terrestre	2.1.2.1-Ley de Transporte		
	3-Pacientes pueden ser complejos o con mórbidos importantes	3.1- Mal acceso Venoso	3.1.1-Condición del Paciente			X
		3.2- Glicemia Elevada	3.2.1-Condición del Paciente			X
		3.3-Tener Mórbidos	3.3.1-Condición del Paciente			X
	4-Exámenes pueden tener mayor duración	4.1- Patologías específicas	4.1.1- Para estudio correcto			X
		4.2- Menos dosis al paciente	4.2.1-No se dosifico la dosis	4.2.1.1- No se realizaron cálculos.		Herramienta info. en tiempo real
		4.3- complicaciones durante el examen	4.3.1- Condición del Paciente			X

Figura 1- Cuadro 5 Porques.

2.2.2. Técnica de Ishikawa.

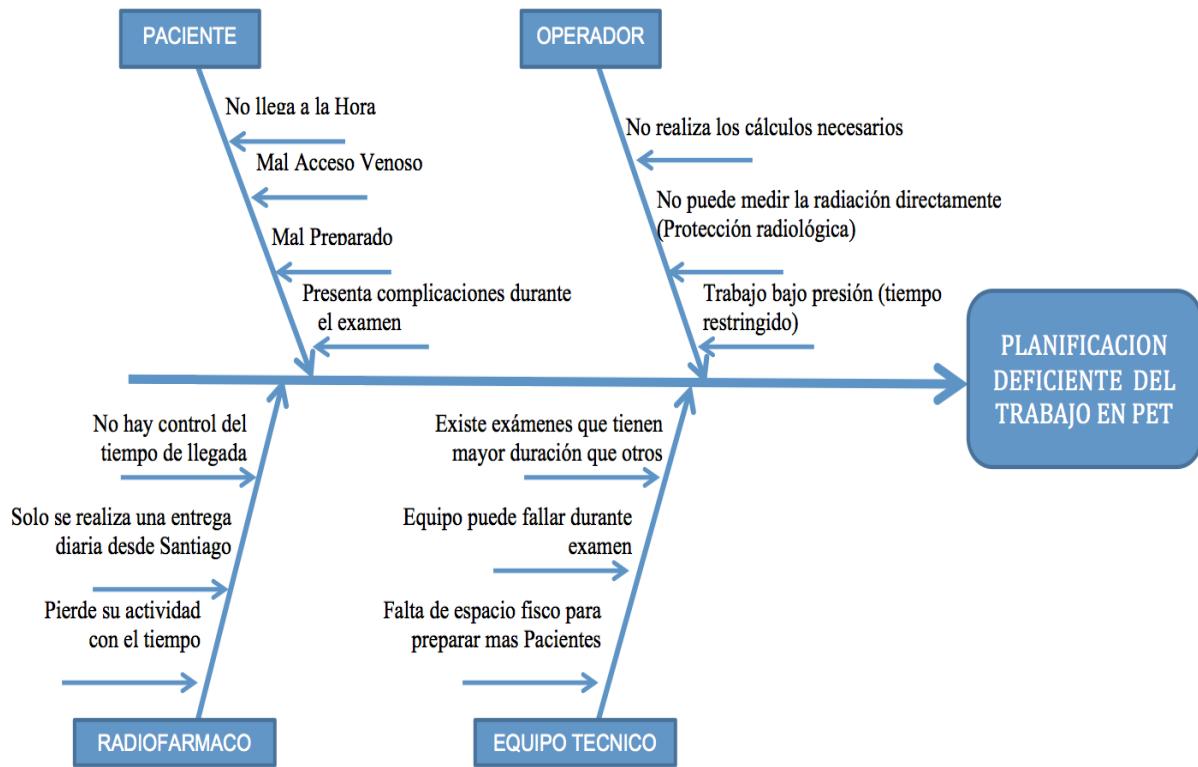


Figura 2- Diagrama Ishikawa

2.3. Objetivos.

2.3.1. Objetivo general.

- “Tener de manera rápida y sencilla los valores de Actividad radiactiva del radiofármaco en cualquier momento de la jornada, sin necesidad de irradiar de manera adicional al personal Operador (Tecnólogos Médicos)”.

2.3.2. Objetivos específicos.

- 1. Lograr que el valor calculado de dosis (mCi) tenga un margen de error de no más de 5% con el Radiofármaco.
- 2. Disminuir los valores de Dosimetría de los operadores (Dosis recibida por los operadores) en un 5% a los 3 meses de su implementación.
- 3. Lograr que la cantidad de mL del Radiofármaco restante al finalizar la jornada tenga un margen de error de no más de 10% con el valor calculado por el programa.
- 4. Generar una visualización de la actividad radiactiva instantánea con una actualización mínima cada 1 minuto y una resolución en minutos para los cálculos.
- 5. Generar una visualización de la actividad radiactiva proyectada en un instante seleccionable de la jornada, tardando en desplegar los datos menos de 1 segundo.
- 6. Lograr un cálculo de cantidad de ml con un margen de error de no más de 20% para cierta radiactividad requerida, Ej. se requiere una dosis de 8 mCi, según el cálculo se debe sacar del contenedor 1,5 mL para ese momento específico de la jornada.

Objetivo Esp.	Metrica	Unidad	Criterio de Exito	Metodo
1	Margen de error	%mCi	<5	Comparar con activímetro
2	Disminucion Dosimetria	%mSv	<5	Revision Informe dosimetricos CCHEN
3	Margen de error	%mL	<10	Comparar con jeringas medidoras
4	Comprobación valores	Segundo	<1	Comprobador de valores de tiempo.
5	Tiempo Procesamiento	Segundo	<1	Medidor de tiempo de Procesamiento.
6	Margen de error	%mCi	<20	Comparar actividad dosificada con activímetro.

Tabla 1 – Objetivos Específicos y Metodos de medición

2.4. Diagrama de alto nivel.

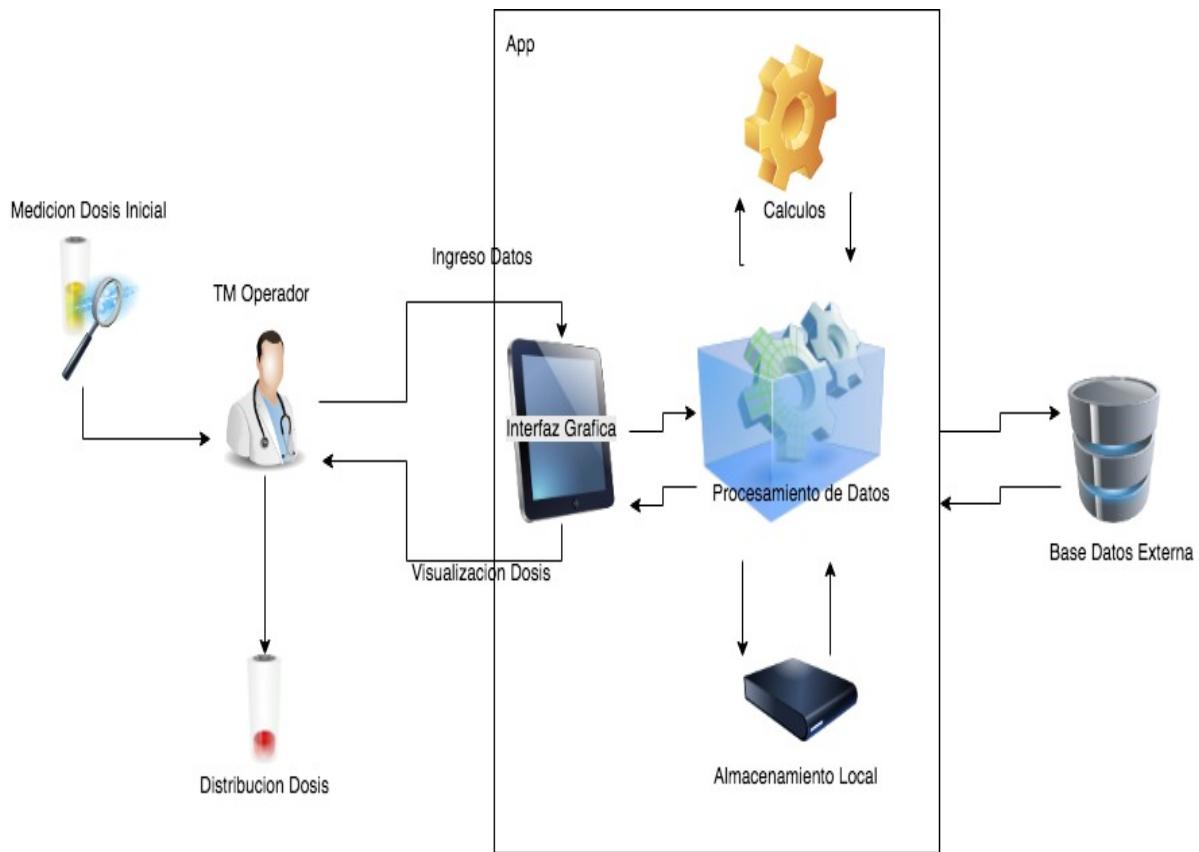


Figura 3 - Diagrama Alto nivel

2.5. Arquitectura de la solución.

Inicial:

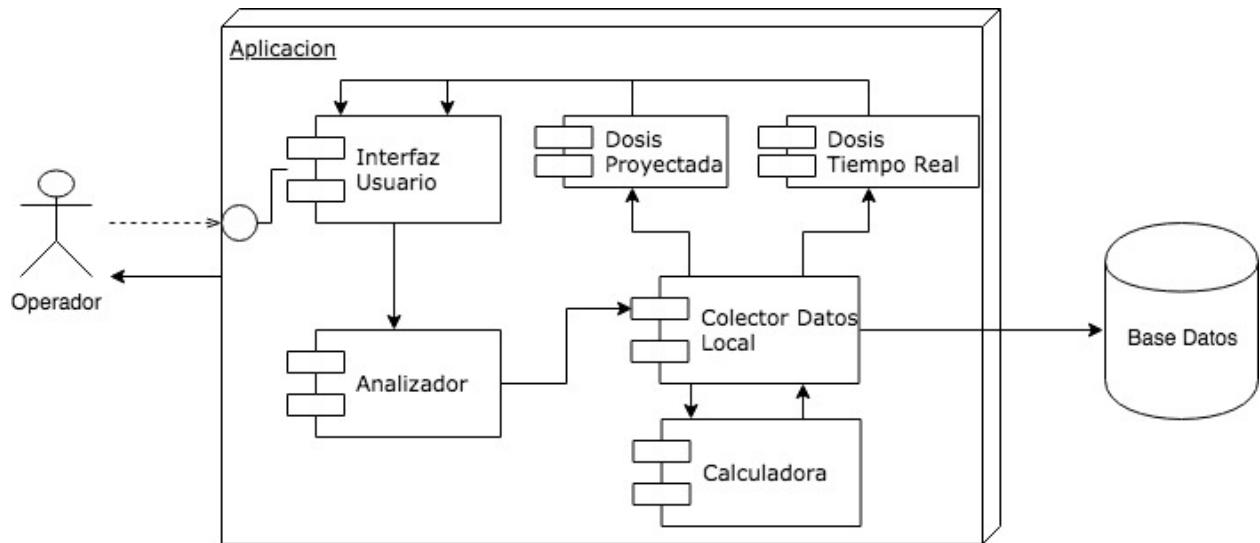


Figura 4 - Diagrama arquitectura alto nivel- Inception.

Version 0.3:

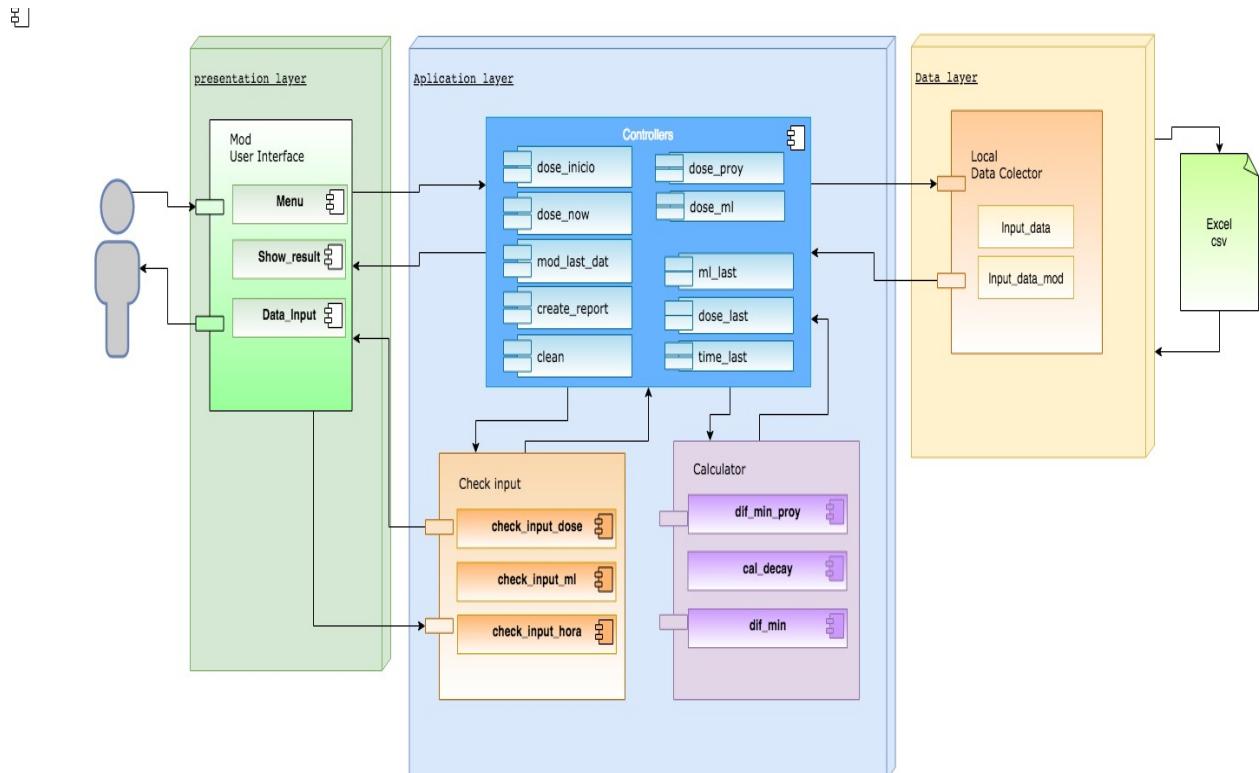


Figura 5 - Diagrama de arquitectura de alto nivel V0.3

2.6. Solución ideal.

- Contar con un sistema robotizado de gestionamiento de dosis, que permita manipular y dosificar la dosis de manera inteligente y permita optimizar los recursos y el tiempo para cada examen.

2.6.1. Alcances y Limitaciones.

- La Aplicación no será capaz de tomar los datos de manera automática.
- El ingreso de datos para la aplicación es manual y depende del operador.
- La Aplicación sólo podrá ser manipulada por los Tecnólogos Médicos capacitados.
- Para este proyecto, la Aplicación será desarrollada sólo para un tipo de dispositivo y sistema operativo.
- Para este proyecto, los reportes se enviarán como archivos a una ubicación predeterminada, dado que todavía no se cuenta con una base de datos.
- Para este proyecto, no se entregará información de Gestiónamiento de pacientes, sólo datos calculados.

2.6.2. Restricciones.

- No existe actualmente un plan de financiamiento para desarrollo de aplicaciones médica en Clínica Bupa Reñaca fuera del departamento informático.
- Se deberán respetar las leyes internas de manejo de información médica dentro del recinto.
- El Grupo de Trabajo es mínimo, por lo que los tiempos de ejecución para el desarrollo de la aplicación serán elevados.
- Para la utilización la aplicación deberá pasar por el comité de Ética de Clínica Bupa Reñaca.
- El riesgo de no cumplir con los objetivos propuestos en los tiempos estipulados generará desmotivación con el cliente y stakeholders, además de hacer menos plausible la asignación de recursos.

2.7. Alternativas de solución.

A la fecha actual, no se han agregado nuevas alternativas de soluciones, por lo que se mantiene la comparativa de las soluciones propuesta:

2.7.1. Procedimientos manuales.

- 1- Los cálculos pueden realizarse de manera manual, o estimar valores para gestionar la dosis en los pacientes.
- 2- Disminuir la calidad del examen ocupando el mínimo de radiofármaco por cada paciente.

2.7.2. Cambios en procedimientos actuales.

- 3- Aumentar el número de operadores y dejar destinado a uno que sólo se preocupe de la distribución y cálculos de dosis.
- 4- Pedir mayor cantidad de radiofármaco.
- 5- Instalar un Ciclotrón en la quinta región que realice despachos locales de radiofármaco.

2.7.3. Alternativas disponibles en el mercado.

- 6- Software y App Calculadoras de dosis disponibles que sólo realizan un cálculo básico de radiactividad inicial y final.
- 7- Activímetro Posee software que realiza un cálculo de decaimiento que no es posible aplicar a una agenda estructurada, al igual que las calculadoras solo se pueden realizar cálculos iniciales y finales.

Soluciones	Criterios						Valoración de la Solución
	Costos Implementación	Tiempos de Implementación	Impacto Negativo Productividad	Problemas Factibilidad técnica	Impacto Negativo Calidad actual	Dificultad cobertura Problema	
Nº1	3	1	1	2	1	1	9
Nº2	3	3	2	3	1	1	13
Nº3	1	2	3	1	3	3	13
Nº4	1	3	3	3	2	2	14
Nº5	1	1	3	1	2	2	10
Nº6	1	2	2	2	2	1	10
Nº7	2	1	2	2	2	1	10

Alta = 1

Medio = 2

Baja = 3

Figura 6 - Cuadro comparativo alternativas de solución

2.8. Solución propuesta.

- Creación de sistema capaz de realizar cálculos de decaimiento radiactivo para Flúor F-18.
- La aplicación será efectiva y eficiente, en términos de conducir a pocos errores y requerir poco tiempo de uso por parte del usuario.
- Herramienta será utilizada por Tecnólogos Médicos encargados de PET.
- Tendrá una Entrada de Datos (Actividad, Hora, mL).
- Generará Salida de Datos (Actividad) en tiempo real y proyectados a cualquier momento de la jornada de trabajo.
- Generará reportes de Distribución de dosis de la jornada para el día de trabajo.

2.8.1. Factibilidades.

Podemos clasificarlas en desde tres perspectivas:

Económica

- No se cuenta con recursos asignados, pero si los resultados son positivos podría abrirse financiamiento, por lo tanto el proyecto sería económicamente factible. Falta evaluación detallada.

Técnica

- No se cuenta con un equipo dedicado para la aplicación, pero se considera suficiente utilizar uno de los equipos de ofimática disponibles, resguardando el uso de recursos.

Operativa

- La Aplicación sólo podrá ser manipulada por los Tecnólogos Médicos capacitados.

2.8.2. Diseño de alto nivel.

□

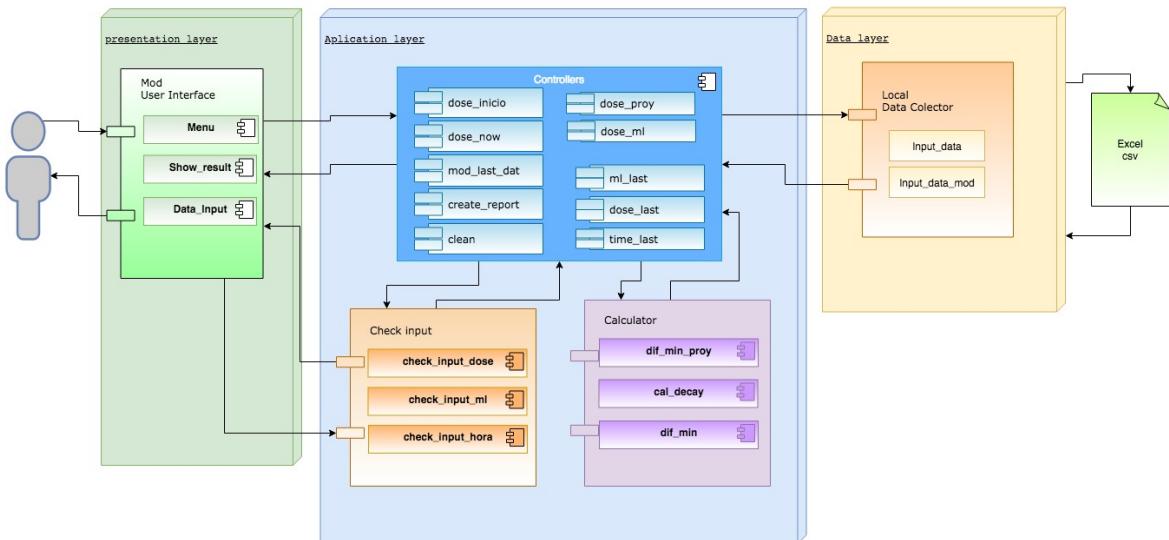


Figura 7 - Diagrama de alto nivel de la aplicación.

Algunos de los módulos principales a implementar son:

- **Modulo Cal_decay:**

- Implementacion: Python 3.7.1
- Librería gráfica: por definir
- Base de datos: local
- Objetivo: Realizar el cálculo del decaimiento radiactivo de un radiofármaco.
- Datos de Entrada:
 - Requiere el valor numérico de Dosis inicial medida en milicuries (mCi) con dos decimales Ej 180.39
 - Requiere el valor entero de tiempo en Minutos (min) que transcurre desde la actividad inicial a la hora de medida requerida Ej 55.
- Datos de Salida:
 - Valor de Decaimiento radiactivo de flúor en milicuries (mCi) de FDG en el periodo de tiempo establecido.

El Módulo cal_decay está relacionado con cualquier módulo que solicite un cálculo de decaimiento radiactivo para el flour.

- Modulo **Check_input**:

- Implementacion: Python 3.7.1
- Librería gráfica: por definir
- Base de datos: local
- Objetivo: Realizar filtrado de datos erróneos ingresados por el usuario.
- Datos de Entrada:
 - def check_input_dose / Ingreso de valor tipo float.
 - def check_input_ml / Ingreso de valor tipo float.
 - def check_input_time / Ingreso de valor tipo int para ingresar cantidad de minutos.
 - def check_input_hora / Ingreso de valor tipo str que transforma a estructura de tiempo, para luego filtrar en formato HH:MM .
- Datos de Salida:
 - def check_input_dose / retorna valor tipo float limitado a 2 decimales.
 - def check_input_ml / retorna valor tipo float.
 - def check_input_time / retorna valor entero.
 - def check_input_hora / retorna valor formato HH:MM .

El módulo Check_input está relacionado con cualquier módulo que implique interacción del usuario para el ingreso de datos.

- Modulo **dif_min**:

- Implementacion: Python 3.7.1
- Librería gráfica: por definir
- Base de datos: local
- Objetivo: Calcular los minutos que han transcurrido desde una hora determinada a la hora actual.
- Datos de Entrada:
 - Hora determinada en formato HH:MM
- Datos de Salida:
 - Minutos en formato Int.

Este Módulo se provee los minutos para distintos módulos de cálculos, se relaciona directamente con el módulo de dose_now, módulo Local_data_colector, dose_last y time_last.

- Modulo **dif_min_proy**:

- Implementacion: Python 3.7.1
- Librería gráfica: por definir
- Base de datos: local
- Objetivo: Calcular los minutos que existen entre dos horas determinadas.
- Datos de Entrada:
 - Hora determinada en formato HH:MM
- Datos de Salida:
 - Minutos en formato Int.

Este Módulo se provee los minutos para distintos módulos de cálculos, se relaciona directamente con el modulo de dose_proy, modulos check_input.

2.8.3. Requerimientos de alto nivel.

Requerimientos Funcionales:

1. Realizar cálculos de decaimiento radiactivo para el Radiofármaco Flúor 18 FDG.
2. Permitir ingresar datos de Dosis inicial en mCi, mL y hora.
3. Mostrar el estado de decaimiento radiactivo actual.
4. Permitir ingresar datos de Dosis suministradas en mL y hora.
5. Mostrar cálculos de Dosis en mCi, Tiempo en formato Hora:minutos y volumen en mL.
6. Generar reporte diario con las Dosis utilizadas para cada examen realizado.
7. Presentar menú de selección de alternativas para las distintas tareas.

Requerimientos No Funcionales:

- La aplicación debe ser intuitiva y fácil de utilizar.
 - Se evaluará que el tiempo de aprendizaje por un nuevo usuario sea menor a 3 horas.
- Se debe resguardar las identidades de los pacientes.
 - A través de la aplicación y sus reportes no se puede acceder a información de pacientes.
- La aplicación debe ser liviana para un computador de ofimática de bajo costo.
 - Funcionará en Windows 7 32 bits, con menos de 500 MB de memoria y 2 GB de disco.
- Mostrar dosimetría actual de forma fácil de visualizar y entender.
 - Un usuario reconocerá la información ubicándose al menos a 3 metros de la pantalla.
- Permitir enviar reportes por correo de forma sencilla.
 - Se evaluará enviando un reporte sin requerir software especializado adicional.
- Mostrar mensajes de error informativos y orientados al usuario final.
 - Un usuario debe ser capaz de interpretar al menos 3 tipos de mensajes posibles.

Capítulo 3: Planificación del proyecto

3. Planificación del proyecto.

3.1. Metodología de desarrollo.

Metodología Ágil : Incremental con base en Kanban.

Roles

En Kanban no se definen Roles, pero sí podemos mencionar para el orden del proyecto las siguientes figuras:

Cliente: Felipe Vera.
Product owner: Patricio Castillo.
Flow Master: Sebastián Inostroza.

3.2. Planificación del proyecto.

El proyecto está limitado por fechas establecida en los Hitos evaluables. Debido a ésto se ha decidido tomar las fechas de evaluación de los hitos como fechas límite de entregables o liberaciones de producto al cliente.

Se define una carta Gantt con las fechas establecidas.

Como objetivo se pretende entregar al finalizar el Trimestre un Mockup funcional, que sea capaz de generar las funcionalidades propuestas a esa fecha por el cliente y los stakeholders.

Nombre	Inicio	Vencimiento	Duraci...
PET Manager	6 Nov 2018	18 Mar 2019	132d
1 Incepcion	6 Nov 2018	20 Nov 2018	15d
5 Iteración 1	26 Nov 2018	11 Dic 2018	16d
12 Iteración 2	11 Dic 2018	26 Dic 2018	16d
25 Iteración 3	25 Dic 2018	7 Ene 2019	14d
39 Iteración 4	4 Ene 2019	21 Ene 2019	18d

Figura 8 - Cuadro planificación proyecto.

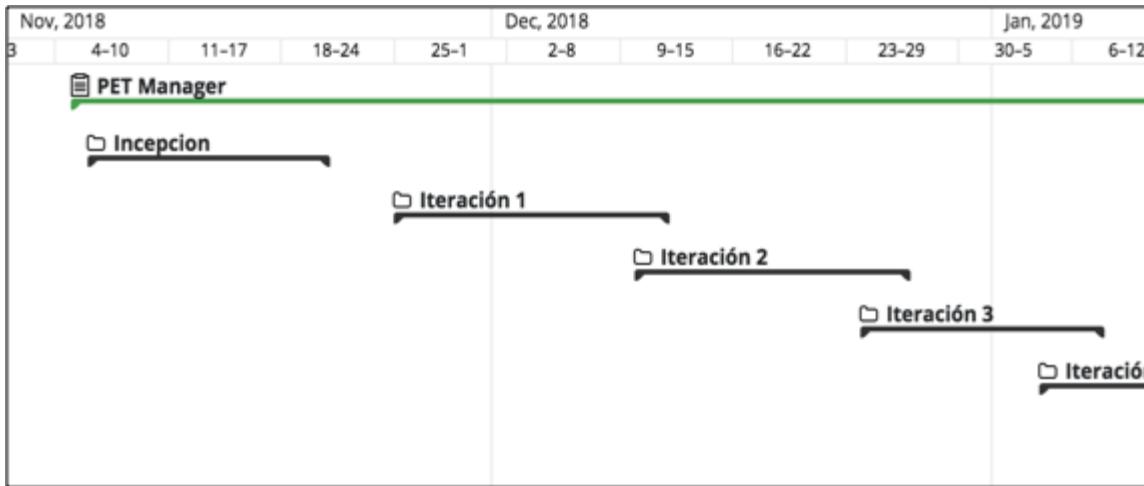


Figura 9 - Cuadro planificación proyecto PET Manager desplegado.

Para la Metodología de Kanban se deben definir:

Políticas para Procesos:

- 1- Encargado de Mover las Tareas : Flow Master
- 2- Encargado de decidir las Prioridades : Flow Master
- 3- ¿Cuando se traslada una Tarea a Hecho?:
 - Cuando se realizan las pruebas de funcionalidad y son aceptadas.
 - Cuando se finaliza la tarea.
- 4- ¿Cuando se traslada una Tarea a la lista de Version?:
 - Cuando se logra la entrega final de la versión trabajada.

Política del Sistema:

- 1 - Para cada entrega se realizará reunión para obtener Feedback de Cliente y Stakeholders (Service delivery meeting).
- 2 - Se realizarán Reuniones con el cliente cada 2 semanas solo si son requeridas por el grupo o el cliente.
- 3 - Los tiempos de entrega se adaptaran a los tiempos de evaluación de los Hitos durante el semestre.Estimación: Hito 3 - Entrega - Mockup Básico Funcional.

Establecer WIP:

- 1 - Los Límites de Tareas son los siguientes:

Lista :

En Proceso (3)

En Pruebas (3)

Clases de servicio

Las clases de servicio son una clasificación de los posibles ítems de trabajo que pueden surgir. Para nuestro Proyecto Tenemos una clasificación de 4 tipos:

- Urgente:** Tareas que necesita de resolución en el menor tiempo posible. Debe ser la primera en resolverse.
- Importante:** Tareas que tiene cierta prioridad de realización . Deben ser las segundas en resolverse.
- Menos Importante:** Tareas que poseen prioridad mediana. Deben ser las terceras en resolverse.
- Puede esperar:** Tareas de baja prioridad. Deben ser las últimas en resolverse.

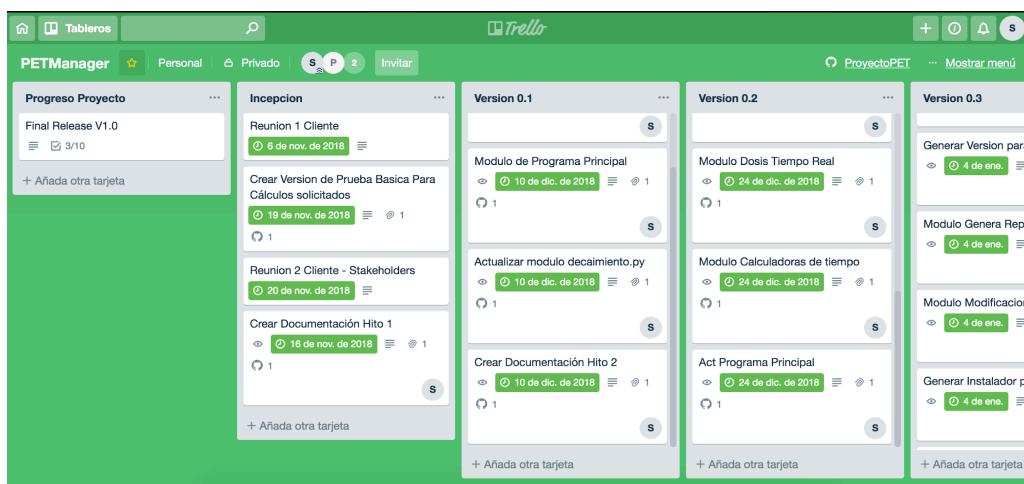


Figura 10 – Software de apoyo a Metodología Kanban (Trello)

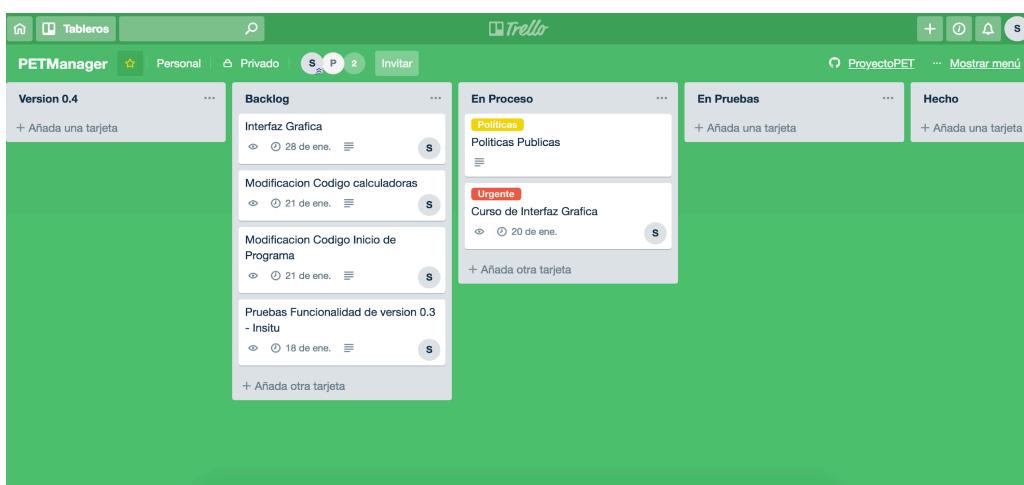


Figura 11 – Vista de Iteración actual Sw Trello.

3.2.1 Backlog Proyecto PET Manager.

Objetivos específicos(OE)

OE1 -Lograr que el valor calculado de dosis (mCi) tenga un margen de error de no más de 5% con el Radiofármaco.

OE2 -Disminuir los valores de Dosimetría de los operadores (Dosis recibida por los operadores) en un 5% a los 3 meses de su implementación

OE3 -Lograr que la cantidad de mL del Radiofármaco restante al finalizar la jornada tenga un margen de error de no más de 10% con el valor calculado por el programa.

OE4 -Generar una visualización de la actividad radiactiva instantánea con una actualización mínima cada 1 minuto y una resolución de minuto para los cálculos.

OE5 -Generar una visualización de la actividad radiactiva proyectada en un instante seleccionable de la jornada, tardando en desplegar los datos menos de 1 segundo.

OE6 -Lograr un cálculo de cantidad de ml con un margen de error de no más de 20% para cierta radiactividad requerida, Ej. se requiere una dosis de 8 mCi, según el cálculo se debe sacar del contenedor 1,5 mL para ese momento específico de la jornada.

Requerimientos Funcionales(RF)

RF1 -Realizar cálculos de decaimiento radiactivo para el Radiofármaco Flúor 18 FDG.

RF2 -Permitir ingresar datos de Dosis inicial en mCi, mL y hora.

RF3 -Mostrar el estado de decaimiento radiactivo actual.

RF4 -Permitir ingresar datos suministrados de Dosis, mL y hora para cada paciente.

RF5 -Mostrar cálculos de Dosis en mCi, Tiempo en formato Hora:minutos y volumen en mL.

RF6 -Generar reporte diario con las Dosis utilizadas para cada examen realizado.

RF7 -Presentar menú de selección de alternativas para las distintas tareas.

Requerimientos No Funcionales(RNF)

RNF1 -La aplicación debe ser intuitiva y fácil de utilizar.

RNF2 -Se debe resguardar las identidades de los pacientes.

RNF3 -La aplicación debe ser liviana para un computador de ofimática de bajo costo.

RNF4 -Mostrar dosimetría actual de forma fácil de visualizar y entender.

RNF5 -Permitir enviar reportes por correo de forma sencilla.

RNF6 -Mostrar mensajes de error informativos y orientados al usuario final.

Backlog PET Manager a la versión 0.3

BACKLOG	V0.1	V0.2	V0.3	Estado
RF1	100%	100%	100%	Logrado
RF2	20%	100%	100%	Logrado
RF3	0%	50%	100%	Logrado
RF4	10%	100%	100%	Logrado
RF5	50%	100%	100%	Logrado
OE1	0%	30%	80%	No logrado
OE2	0%	0%	0%	No logrado
OE3	0%	20%	50%	No logrado
OE4	30%	100%	100%	Logrado
OE5	0%	50%	100%	Logrado
OE6	20%	80%	100%	Logrado
RNF1	0%	0%	0%	No logrado
RNF2	0%	100%	100%	Logrado
RF6	0%	20%	100%	Logrado
RF7	20%	50%	100%	Logrado
RNF3	0%	0%	100%	Logrado
RNF4	20%	50%	80%	No logrado
RNF5	0%	0%	0%	No logrado
RNF6	0%	30%	80%	No logrado

Tabla 2 – Backlog de Objetivos y Requisitos alto nivel – PET Manager

Requerimientos Propuestos vs Versiones Realizadas - Mitigación de riesgos

Para demostrar la viabilidad del proyecto fue necesario demostrar que los riesgos más importantes que podrían amenazar con el fracaso del proyecto fueron mitigados o medianamente mitigados.

Las Prioridades de Requerimientos que demostraban viabilidad para el Cliente y Stakeholders con sus respectivas métricas son:

1. Realizar cálculos de decaimiento radiactivo para el Radiofármaco Flúor 18 FDG.
2. Permitir ingresar datos de Dosis inicial en mCi, mL y hora.
3. Mostrar el estado de decaimiento radiactivo actual.
4. Permitir ingresar datos de Dosis suministradas en mL y hora.
5. Mostrar cálculos de Dosis en mCi, Tiempo en formato Hora:minutos y volumen en mL.
6. Generar reporte diario con las Dosis utilizadas para cada examen realizado.
7. Presentar menú de selección de alternativas para las distintas tareas.

Backlog	V0.1	V0.2	V0.3	Metrica
R1	50%	80%	100%	Generar el cálculo de decaimiento y comparar con valor real in situ
R2	20%	80%	100%	Permitir el ingreso de datos iniciales a través de interfaz gráfica en formatos correspondientes sin errores
R3	20%	80%	100%	Generar y mostrar el calculo a traves de vista en la aplicación
R4	20%	80%	100%	Permitir el ingreso de datos a través de interfaz gráfica en formatos correspondientes
R5	10%	100%	100%	Visualizar resultados en los formatos especificados en la vista de resultados
R6	10%	50%	100%	Archivo que registre las dosis iniciales y sus modificaciones a lo largo de la jornada
R7	5%	70%	100%	Generar menú que permite navegar por aplicación en vistas interfaz gráfica

Tabla 3 – Tabla de mitigacion de riesgos por cada Version

3.3. Gestión de la configuración.

Gestión de líneas base

Cada vez que se finalice una etapa será liberada una línea base del proyecto. Cada vez que se presente un Hito y este sea validado tanto por el flow Master como por el cliente será liberada una línea base del proyecto. Una vez Finalizado el Hito, se procederá a crear una nueva rama con el nombre de la versión correspondiente.

Presentación a evaluaciones (Hitos)

Durante el desarrollo del proyecto, se realizarán presentaciones ante una comisión para evaluar el avance del proyecto, para este propósito será liberada una línea base del proyecto. Cada línea base que posea el proyecto será documentada en el repositorio de la aplicación de github en la carpeta "docs".

Los Archivos serán:

- Presentación (PPTX, PDF, etc) .
- Documento de aceptación (txt, DOC, PDF, XLSX, etc).

Los Archivos serán subidos a un repositorio en GITHUB con dirección:

<https://github.com/Sebasinos/ProyectoPET.git>

The screenshot shows the GitHub repository 'Proyecto PETManager'. At the top, there are navigation links: Code, Issues (1), Pull requests (0), Projects (0), Wiki, Insights, and Settings. Below the header, it says 'Proyecto PETManager' and has an 'Edit' button. A 'Manage topics' link is also present. Key statistics are displayed: 14 commits, 1 branch, 1 release, and 1 contributor. A dropdown menu shows 'Branch: master' and a 'New pull request' button. To the right are buttons for 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. The main area lists commits from 'Sebasinos': 'Delete .DS_Store' (Latest commit 768a36a 3 hours ago), 'docs' (Realizo esquemas de carpetas y archivos solicitado 3 hours ago), 'src' (Realizo esquemas de carpetas y archivos solicitado 3 hours ago), 'tools' (Realizo esquemas de carpetas y archivos solicitado 3 hours ago), and 'README.md' (Initial commit 5 days ago). Below the commit list is a 'README.md' file with an edit icon. The title 'ProyectoPET' is visible at the bottom of the repository page.

Figura 12 - Visualización del repositorio GITHUB.

3.4. Gestión de cambios.

La Gestión de cambios será realizada con la herramienta GIT - “PullRequest”, que permite llevar un registro de todos los cambios que se han realizado en el repositorio del proyecto.

Los Commits serán obligatorios cada vez que se finalice el Hito evaluado, no impidiendo realizar avances o actualizaciones en cualquier momento de la duración del proyecto.

Cada vez que finalice un Hito se creara una nueva rama en el GITHUB con toda la información de la rama anterior y con el nombre correspondiente a la version a ejecutar.

The screenshot shows the GitHub repository 'Sebasinos / ProyectoPET'. At the top, there are navigation links: Code, Issues (1), Pull requests (0), Projects (0), Wiki, Insights, and Settings. Below the header, it says 'Branch: master' and has 'Watch' (0), 'Star' (0), and 'Fork' (0) buttons. The main area shows a list of commits: 'Delete .DS_Store' by Sebasinos committed 3 hours ago (Verified, 768a36a), 'Delete .gitignore' by Sebasinos committed 3 hours ago (Verified, beb81fd), 'Delete .DS_Store' by Sebasinos committed 3 hours ago (Verified, e65076f), and 'Realizo esquemas de carpetas y archivos solicitado' by Sebasinos committed 3 hours ago (Verified, db5ba96). On the left, there are collapsed sections for 'Commits on Dec 2, 2018' and 'Commits on Nov 30, 2018'.

Figura 13 - Visualización de cambios en GITHUB

Proyecto PETManager

Manage topics

66 commits 4 branches 1 release 2 contributors

Branch: 0.3 New pull request Create new file Upload files Find file Clone or download

This branch is even with master.

Sebasinos Create Manual_user PeTManager Beta 1 Latest commit 26d057a 4 days ago

docs update with all functionalities 6 days ago

src test changes 5 days ago

tools Create Manual_user PeTManager Beta 4 days ago

README.md Initial commit 2 months ago

README.md

ProyectoPET

Proyecto PETManager

The screenshot shows a GitHub repository interface. At the top, it displays project statistics: 66 commits, 4 branches, 1 release, and 2 contributors. Below this, there's a navigation bar with buttons for 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A message indicates that the current branch ('0.3') is even with the 'master' branch. The main content area shows a list of commits from a user named 'Sebasinos' for a 'Create Manual_user PeTManager Beta' pull request. The commits include updates to 'docs', 'src', 'tools', and the 'README.md' file. The most recent commit was made 4 days ago. A 'Pull request' and 'Compare' button are also present in this section. At the bottom of the screenshot, the repository name 'ProyectoPET' and its full name 'Proyecto PETManager' are shown.

Figura 14 - Branch o rama 0.3 GITHUB

3.5. Gestión de riesgos.

Los riesgos serán clasificados de la siguiente manera:

- Riesgo de proyecto.
- Riesgo técnico.
- Riesgo de negocio.

Después de ser clasificados se evaluará el impacto del riesgo en una escala de 0 a 1. Donde 0 es muy bajo y 1 es muy alto impacto.

Además de determinar el impacto del riesgo, hay que agregar una ponderación de probabilidad de ocurrencia que se evaluará en una escala de 0 a 1. Donde 0 es muy baja y 1 es muy alta probabilidad. El resultado nos dará una ponderación de amenaza del riesgo.

Porcentaje de amenaza = (Probabilidad * impacto del riesgo) *100

Cada riesgo incluye un plan de mitigación, un plan de contingencia y la fase del proyecto cuando se gatilla el riesgo(Versión de trabajo).

A continuación se mostraran los riesgos identificados del proyecto. Cabe destacar que la identificación de riesgos es una tarea constante a lo largo de todo el proyecto.

Riesgos de Proyecto

ID	Riesgo	Probabilidad	Impacto	% amenaza	Mitigación	Contingencia	Fecha Versión Rel.
RP1	Incumplimiento de fechas	0,8	0,2	16	Cumplir con la planificación del proyecto rigurosamente	Aumentar las horas diarias de trabajo	10-12-18
RP2	Los Requerimientos cambian en cada Versión	0,8	1	80	Establecer el Alcance junto con el cliente y respetar los ítems acordados	Por cada Cambios se deberá analizar la factibilidad y si es factible modificar los plazos	24-Dic-18
RP3	Mala planificación de Tiempo para las tareas	0,6	0,4	24	Planificar las tareas con más riesgo primero	Re-planificación de la tarea	
RP4	Mala estimación de Tarea	0,7	0,5	35	Usar técnicas de estimación validadas	Re-estimar las Tareas	
RP5	Plataforma de gestión de la configuración no está disponible, se cae.	0,2	0,1	2	Llevar una gestión de la configuración de forma manual.	Usar gestión de la config. Manual	
RP6	Plataforma de control de versiones no está disponible, se cae.	0,2	0,1	2	Llevar un control de versión local adicional.	Usar el control de versión local	
RP7	Requerimientos no cumplen con las expectativas del cliente	0,5	0,8	40	Realizar reuniones semanales para recibir feedback	Aumentar las HH para modificar aquellos requerimientos	7 Ene 19

Tabla 4- Riesgos de Proyecto

Riesgos técnicos

ID	Riesgo	Probabilidad	Impacto	% de amenaza	Mitigación	Contingencia	Fecha de Versión Relacionado
RT1	Un mal diseño de software implica volver a diseñar e implementar	0,5	0,6	30	Evaluación periódica del modelo actual	Realizar un nuevo diseño, verificarlo validar y volver a implementar	24-11-18
RT2	Metodología no acorde al proyecto	0,1	0,4	4	Evaluar la mejor metodología	Adaptar nueva metodología en base al resto de la planificación	
RT3	Inexperiencia en el lenguaje de programación o herramientas	0,8	0,9	72	Aumentar las horas de trabajo diarias dedicada al auto aprendizaje	Aumentar las horas diarias de trabajo	25-11-18
RT4	Complejidad de herramienta no permite al equipo de desarrollo avanzar	0,7	0,7	49	Realizar investigación, estudiar libros o ver tutoriales, documentaciones.	Cambiarla herramienta de desarrollo.	9-12-18

Tabla 5- Riesgos técnicos

Riesgos de negocio

ID	Riesgo	Probabilidad	Impacto	% de amenaza	Mitigación	Contingencia	Fecha Versión Relacionado
R N1	Directivos rechazan implementación de aplicación	0,2	0,8	16	Abordar las causas que provocaron rechazo.	Realizar un nuevo estudio e implementar las mejoras	04-01-19
RN 2	Incumplimiento de fecha de entrega de la aplicación.	0,5	0,8	40	Aumentar el trabajo de cada Version, solicitar reuniones extras para cumplir con los requerimientos de stakeholders	Llegar a acuerdo con directivos por nuevos plazos	04-01-19
RN 3	Fecha de presentación de la aplicación no definida por parte del cliente	0,7	0,2	14	Coordinar con cliente fecha para presentación formal de la aplicación antes de finalizar el proyecto.	Establecer fecha de presentación y documentar en proyecto antes de la versión 0.1	25-11-18

Tabla 6 - Riesgos negocio

3.6. Entorno de desarrollo.

El ambiente de desarrollo del producto software será el computador personal del desarrollador.

Hardware

PC

Modelo	MacBook Pro (Retina, 13-inch, Late 2013)
Procesador	2,4 GHz Intel Core i5
Memoria	8 GB 1600 MHz DDR3
S.O	OS X El capitan Version 10.11.6 (15G22010)

Tabla 7- Hardware de desarrollo para PET Manager

Herramientas de construcción

Tipo	Nombre
Ambiente de desarrollo integrado/Compilador	IDLE Python
Lenguaje de programación	Python 3.7.1
Gestor Grafico	Por definir
Sw Para metodología	Trello
Repositorio del proyecto	Github
Sw creador instalador	Inno Setup

Tabla 8 -Herramientas de desarrollo

Capítulo 4: Resultados

4. Resultados .

4.1. Análisis de resultados.

Se abordarán los resultados obtenidos a la fecha del proyecto, el margen de error de las estimaciones, la arquitectura y el diseño planificado inicial, versus el obtenido. Los módulos realizados hasta la fecha actual son capaces de generar la funcionalidad completa de la aplicación. Cabe destacar que se entregó, al Presentar el Hito 2, un modelo funcional por consola a 1 stakeholders con la finalidad de generar feedback de lo implementado hasta esa fecha.

Se decidió con el equipo de proyecto priorizar el funcionamiento de la aplicación y luego de desarrollar todos los módulos funcionales agregar una interfaz que pueda ser aceptada por los stakeholders.

Se inició la programación del código realizando el cálculo de decaimiento radiactivo para el radiofármaco Flúor-18 FDG. logrando el funcionamiento del código con parámetros ingresados de manera manual.

```
import math

Act_ini = float(raw_input("Ingrese Actividad Inicial: "))
tiempo = float(raw_input("Ingrese tiempo: "))
des_fluor =float(109.771)
Act_fin =Act_ini*(math.exp(-(0.693*float(tiempo)/des_fluor)))

print "El decaimiento es", Act_fin
```

Figura 15- Código fórmula decaimiento radiactivo.

```
=====
RESTART: /Users/Sebas/Desktop/Codigo PET/Decaimiento.py =====
Ingrese Actividad Inicial: 120
Ingrese tiempo: 30
El decaimiento es 99.29522252044228
>>> |
```

Figura 16 - resultado formula decaimiento radiactivo.

El código fue actualizado para poder ser implementado en conjunto con otros módulos. Los **módulos de cálculo** se presentan de la siguiente manera:

- Modulo **cal_decay**:

```
def cal_decay(dose_act,minut_decay):  
    dose_act=float(dose_act)  
    des_fluor =float(109.771)  
    minutos=float(minut_decay)  
    act_fin=dose_act*(math.exp(-(0.693*minutos/des_fluor)))  
    return act_fin
```

Figura 17 - Módulo actualizado fórmula decaimiento radiactivo.

Éste Módulo retorna el valor de actividad restante luego de ingresar la actividad inicial y el tiempo transcurrido en formato de minutos, valor que es ocupado en conjunto con otros módulos según el requerimiento solicitado.

- Módulo **dif_min**: que calcula la diferencia de tiempo entre una hora determinada por el usuario y la hora actual del sistema, dando un resultado en minutos.

```
def dif_min(time_x):  
    now= time.strftime('%H:%M')  
    start_dt = dt.datetime.strptime(now, '%H:%M')  
    end_dt = dt.datetime.strptime(time_x, '%H:%M')  
    diff = (start_dt - end_dt)  
    minutos= int(diff.seconds/60)  
    return minutos
```

Figura 18 - Módulo dif_min .

- Módulo **dif_min_proy**: que calcula la diferencia de tiempo entre dos horas determinadas por el usuario, dando el resultado en minutos. En otras palabras es la cantidad de minutos que han transcurrido entre una hora y otra.

```
def dif_min_proy(time_1,time_2):
    start_dt = dt.datetime.strptime(time_2, '%H:%M')
    end_dt = dt.datetime.strptime(time_1, '%H:%M')
    diff = (start_dt - end_dt)
    minutos= int(diff.seconds/60)
    print (minutos)
    return minutos
```

Figura 19- Módulo dif_min_proy

Para la implementación del **Módulo Chequeo de datos** (Check_output) se analizó crear restricciones individuales para cada tipo de dato que debe ingresar el usuario, con el fin de disminuir la probabilidad de fallo de la aplicación.

Los módulos implementados se presentan a continuación:

- **Módulo check_input_dose** : Encargado de controlar el ingreso adecuado de datos de tipo dosis, limitando los datos a tipo numérico con dos decimales. Si los valores son ingresado de manera incorrecta, se retornara una opción de ingreso nuevamente de datos con la restricción solicitada.

```
def check_input_dose():
    while True:
        dosis_u = input("Ingrese Dosis con PUNTO(.) y solo 2 decimales\n")
        try:
            dosis_u = float(dosis_u)
            dosis_u = "%.*f" % (2, dosis_u)
            return dosis_u

        except ValueError:
            print ("Debe ingresar valores separador por PUNTO(.) y solo 2 decimales")
```

Figura 20 - Módulo Check_input_dose.

- **Módulo check_input_ml**: Encargado de controlar el ingreso adecuado de datos de tipo ml, limitando los datos a tipo numérico con dos decimales. Si los valores son ingresados de manera incorrecta, se retornará una opción de ingreso nuevamente de datos con la restricción solicitada.

```
def check_input_ml():
    while True:
        ml_u = input("Ingrese cantidad de ml con PUNTO(.) y solo 2 decimales\n")
        try:
            ml_u = float(ml_u)
            ml_u = "%.*f" % (2, ml_u)
            return ml_u

        except ValueError:
            print ("Debe ingresar valores separador por PUNTO(.) y solo 2 decimales.")
```

Figura 21 - Módulo Check_input_ml.

- Modulo **check_input_hora**: Encargado de controlar el ingreso adecuado de datos de tipo Hora, limitando los datos a tipo numérico con formato HH:MM (Hora:minuto). Si los valores son ingresados de manera incorrecta, se retornará una opción de ingreso nuevamente de datos con la restricción solicitada.

```
def check_input_hora():
    while True:
        data = input("Ingrese hora en formato HH:MM \n")
        try:
            time.strptime(data, "%H:%M")
            hour=time.strptime(data, "%H:%M")
            hour_min=time.strftime('%H:%M', hour)
            return hour_min
        except ValueError:
            print ("Debe ingresar formato valido HH:MM ")
```

Figura 22 - Módulo Check_input_hora.

Para los **Módulos de Controladores (Controllers)** se desarrollaron:

- Módulo **dose_inicio** : Encargado de solicitar los datos iniciales de Dosis, tiempo y cantidad de ml.

```
def dose_inicio():
    dose=check_input_dose()
    hora=check_input_hora()
    ml=check_input_ml()
    tupla=(dose,hora,ml)
    lista.append(tupla)
    print ("Datos ingresados con éxito")
```

Figura 23 - Módulo dose_inicio.

- Modulo **dose_now** : Encargado de entregar la dosis en tiempo Real.

```
def dose_now(lista):
    act_ini=dose_last(lista)
    time=time_last(lista)
    minutos=dif_min(time)
    dose_now=cal_decay(act_ini,minutos)
    dose_now=round(dose_now,3)
    return dose_now
```

Figura 24 - Módulo dose_now.

- Modulo **dose_proy**: Encargado de entregar la dosis en un tiempo proyectado.

```
def dose_proy(lista):
    hora_x=check_input_hora()
    act_ini=dose_last(lista)
    time=time_last(lista)
    minutos=dif_min_proy(time,hora_x)
    dose=cal_decay(act_ini,minutos)
    dose=round(dose,3)
    print ("la dosis proyectada para las",hora_x," es", dose, "mCi")
    return dose
```

Figura 25 - Módulo dose_proy.

- Módulo **dose_ml**: Encargado de entregar la cantidad de mL necesarios para cierta actividad en el instante en que se hace la consulta.

```
def dose_ml(lista):
    dose_req=float(check_input_dose())
    dose_act=float(dose_now(lista))
    ml_act=float(ml_last(lista))
    ml= (dose_req*ml_act)/dose_act
    ml= round(ml,1)
    print ("Los mL necesarios para tener", dose_req, "mCi, son: ", ml, "mL")
```

Figura 26 - Módulo dose_ml.

- Módulo **dose_last**: Encargado de entregar el último valor de dosis

```
def dose_last(lista):
    lista=lista[-1]
    dose_last=lista[0]
    return dose_last
```

Figura 27 - Módulo dose_last.

- Módulo **time_last**: Encargado de entregar el último valor de tiempo

```
def time_last(lista):
    lista=lista[-1]
    time_last=lista[1]
    return time_last
```

Figura 28 - Módulo time_last.

- Módulo **ml_last**: Encargado de entregar el último valor de mL

```
def ml_last(lista):
    lista=lista[-1]
    ml_last=lista[2]
    return ml_last
```

Figura 29 - Módulo ml_last.

- Módulo **create_report**: Encargado de generar un reporte de los pacientes con los siguientes datos: Número de paciente, dosis en mCi, hora y ml. Para la realización se necesitan 2 módulos:
 - Modulo **gen_info**: Encargado de generar una lista con toda la información que se incorporará en el reporte

```
def gen_info(lista2):
    d=1
    a=0
    for i in lista2:
        listadosis=[]
        listadosis.append(str(d))
        c=0
        while c < 3:
            listadosis.append(lista2[a][c])
            c = c+1

        listafinal.append(listadosis)
        a=a+1
        d=d+1
```

Figura 30 - Módulo gen_info.

- Modulo **create_file**: Encargado de generar el archivo excel en formato csv.

```
def create_file(listafinal):
    archivo="lista_Pacientes_Diario.csv"
    csv=open(archivo,"w")
    titulo= "Paciente,Dosis,Hora,mL\n"
    csv.write(titulo)

    s=0
    for linea in listafinal:
        pkte,dosis,hora,ml=listafinal[s][0],listafinal[s][1],listafinal[s][2],listafinal[s][3],
        filas=pkte+","+dosis+","+hora+","+ml+"\n"
        csv.write(filas)
        s=s+1
    csv.close()
    print ("Reporte generado con Exito")
```

Figura 31 - Módulo create_file.

- Módulo **mod_last_dat**: Encargado de modificar los datos del último registro de paciente ingresado.

```
def mod_last_dat(lista2,lista):
    if (lista2 == []):
        print("No hay pacientes para modificar")

    else:
        opc=0
        while opc != "s":
            print("Ingrese una opción valida")
            menuPrincipal = menu_mod()
            opc = opciones()
            if(opc == "1"):
                dose=check_input_dose()
                lis=list(lista2[-1])
                lis[0]=dose
                hour=lis[1]
                ml=lis[2]
                listanew=tuple(lis)
                lista2.pop(-1)
                lista2.append(listanew)
                lista.pop(-1)
                input_data_mod(dose,hour,ml)
                print ("Regresando al Menú")
            if(opc == "2"):
                hour=check_input_hora()
                lis=list(lista2[-1])
                lis[1]=hour
                dose=lis[0]
                ml=lis[2]
                listanew=tuple(lis)
                lista2.pop(-1)
                lista2.append(listanew)
                lista.pop(-1)
                input_data_mod(dose,hour,ml)
                print ("Regresando al Menú")
            if(opc == "3"):
                ml=check_input_ml()
                lis=list(lista2[-1])
                lis[2]=ml
                dose=lis[0]
                hour=lis[1]
                listanew=tuple(lis)
                lista2.pop(-1)
                lista2.append(listanew)
                lista.pop(-1)
                input_data_mod(dose,hour,ml)
                print ("Regresando al Menú")
```

Figura 32 - Módulo gen_info.

Para los **Módulos de datos (colectores)** se desarrollaron:

- Módulo **input_data**: Encargado de guardar los datos de ingreso de cada nuevo paciente y generar la modificación de los nuevos datos calculados para el radiofármaco.

```
def input_data(lista):
    act_dose=dose_last(lista)
    act_ml=ml_last(lista)
    act_hour=time_last(lista)
    new_dose=check_input_dose()
    new_hour=check_input_hora()
    new_ml=check_input_ml()
    minutos=dif_min_proy(act_hour,new_hour)
    doserefresh=cal_decay(act_dose,minutos)
    doserefresh=round(float(doserefresh)-float(new_dose),2)
    mlrefresh= float(ml_last(lista)) - float(new_ml)
    tupla=(new_dose,new_hour,new_ml)
    lista2.append(tupla)
    tupla_act=(str(doserefresh), new_hour, str(mlrefresh))
    lista.append(tupla_act)
    print ("Datos ingresados con éxito")
```

Figura 33 - Módulo input_data.

- Módulo **input_data_mod**: Encargado de guardar y generar los datos de la lista principal que almacena los valores autogenerados del decaimiento de radiofármaco, luego de haber realizado una modificación.

```
def input_data_mod(dose,hour,ml):
    act_dose=dose_last(lista)
    act_ml=ml_last(lista)
    act_hour=time_last(lista)
    new_dose=dose
    new_hour=hour
    new_ml=ml
    minutos=dif_min_proy(act_hour,new_hour)
    doserefresh=cal_decay(act_dose,minutos)
    doserefresh=round(float(doserefresh)-float(new_dose),2)
    mlrefresh= float(ml_last(lista)) - float(new_ml)
    tupla_act=(str(doserefresh), new_hour, str(mlrefresh))
    lista.append(tupla_act)
    print ("Datos ingresados con éxito")
```

Figura 34 - Módulo input_data_mod.

Por Último, se implementaron dos menús provisoriamente cada uno con un selector de opciones para la realización de pruebas del código expuesto anteriormente, con el fin de comprobar que los módulos se han conectado de manera correcta entre sí y funcionarán como una interfaz provisoria para el usuario.

Módulos Interfaz para navegabilidad del usuario:

- Módulo **menú** y **opciones** : Permiten la navegabilidad dentro del código y actúa como herramienta provisoria antes de la interfaz gráfica a realizar.

```
def menu():

    #MENU DE OPCIONES
    print("** ::::::::::::::::::::: **")
    print(":: Seleccione una Opción ::")
    print("** ::::::::::::::::::::: **")
    print("-----")
    print("I Ingresar Dosis Inicial : -> 1 |")
    print("I Calcular Dosis Proyectada: -> 2 |")
    print("I Calcular Dosis Real-Time : -> 3 |")
    print("I Calcular mL para X dosis : -> 4 |")
    print("I Ingreso dato nuevo Pcte : -> 5 |")
    print("I Generar Reporte : -> 6 |")
    print("I Modificar ultimo Pcte : -> 7 |")
    print("I Salir : -> s |")
    print("-----")
    print("** ::::::::::::::::::::: **")
    print("\n")

#OPCIÓN DEL MENU

def opciones(opc=0):
    opcion =input("Seleccione una Opción... ")
    return opcion

opc=0
while opc != "s":
    print("Ingrese una opción valida")
    menuPrincipal = menu()
    opc = opciones()
#OPCIÓN DEL MENU
```

Figura 35 - Módulo menú y opciones.

- Módulo **menú_mod** : Permite la navegabilidad dentro del código para el módulo de modificación de datos y actúa como herramienta provisoria antes de la interfaz gráfica a realizar.

```
def menu_mod():
    clean(var)
    print("** Importante **")
    print("Los datos modificados solo son aplicables al ULTIMO paciente ingresado")
    print("Selecciona Opción:")
    print("** ::::::::::::::::::::: **")
    print("Modificar Dosis :> 1")
    print("Modificar Hora :> 2")
    print("Modificar mL :> 3")
    print("Volver al menú :> s")
    print("** ::::::::::::::::::::: **")
```

Figura 36 - Módulo menu_mod.

La aplicación con el archivo instalable se encuentra actualmente en el repositorio de Github del proyecto con el nombre PetManager_Setup.exe , este instalador es compatible con windows para cumplir con unos de los requisitos no funcionales.

Además se encuentra disponible el manual de usuario con el nombre User_Manual_PetManager.pdf Aquí se encuentran las instrucciones de instalación y funcionamiento de la aplicación.

Sebasinos / ProyectoPET

Code Issues Pull requests Projects Wiki Insights Settings

Branch: master → **ProyectoPET / tools / Install_files_PETManager_Beta /**

Create new file Upload files Find file History

Sebasinos Create Manual_user PeTManager Beta Latest commit 260d57a 2 minutes ago

..

Licencia de uso del software PETMAN... Create Licencia de uso del software PETMANAGER.txt 3 minutes ago

PetManager_Setup.exe files_release Version Beta- SETUP 4 minutes ago

Petmanager_Install_Scripts.iss Create Petmanager_Install_Scripts.iss 2 minutes ago

User_Manual_PetManager.pdf Create Manual_user PeTManager Beta 2 minutes ago

Figura 37 - Github ProyectoPET carpeta de instaladores.

La Aplicación tuvo fecha de lanzamiento al cliente el dia 4 de Enero y fue instalada el dia 7 de enero en el servicio de PET. El computador ocupado posee un SO windows 7 con 4 gb ram y disco duro de 500Gb. El Software no tuvo problemas de compatibilidad.

Su versión 0.3 se encuentra disponible en versión para SO windows y mac, la versión mac fue desarrollada por los developers para testeos del código.

Sebasinos / ProyectoPET

Code Issues Pull requests Projects Wiki Insights Settings

Branch: master → **ProyectoPET / tools /**

Create new file Upload files Find file History

Sebasinos Create Manual_user PeTManager Beta Latest commit 260d57a 5 hours ago

..

Install_files_PETManager_Beta Create Manual_user PeTManager Beta 5 hours ago

petmanager-vbeta -mac-files files from release 20 hours ago

petmanager-vbeta -win-files files from release 20 hours ago

release_vbeta -petManager - mac release PetManger Beta mac/win 20 hours ago

release_vbeta -petManager - windows release PetManger Beta mac/win 20 hours ago

Figura 38 - Github ProyectoPET carpeta de archivos instalables para win y mac.

Pruebas de desarrollador

A continuación se presentan los resultados con evidencia generados en la actualidad:

- Menú y selector de opciones ingresadas.

```
Ingrese una opción valida
** ::::::::::::::::::::: **
:: Seleccione una Opción ::
** ::::::::::::::::::::: **
-----
| Ingresar Dosis Inicial : -> 1      |
| Calcular Dosis Proyectada: -> 2     |
| Calcular Dosis Real-Time : -> 3     |
| Calcular mL para X dosis : -> 4     |
| Ingreso dato nuevo Pcte : -> 5     |
| Generar Reporte          : -> 6     |
| Modificar ultimo Pcte   : -> 7     |
| Salir                     : -> s     |
-----
** ::::::::::::::::::::: **

Seleccione una Opción...
```

Figura 39 - Ejecución del menu y selector de opciones.

- Ingreso de Dosis inicial en conjunto con chequeo de inputs.

```
Seleccione una Opción... 1
Ingrese Dosis con PUNTO(.) y solo 2 decimales
120
Ingrese hora en formato HH:MM
10:00
Ingrese cantidad de ml con PUNTO(.) y solo 2 decimales
10
Datos ingresados con exito
Regresando al Menu
```

Figura 40 - Ejecución de check input e ingreso de dosis inicial.

- Conocer la dosis proyectada en un tiempo específico.

```
Seleccione una Opción... 2
Ingrese hora en formato HH:MM
23:00
la dosis proyectada para las 23:00 es 0.872 mCi
Regresando al Menu
```

Figura 41 - Ejecución de Dosis en tiempo real.

- Conocer la dosis en tiempo real.

```
Seleccione una Opción... 3
la Dosis actual es: 0.889 mCi, en: 10.00 mL
Actualmente se han realizado 0 pacientes
Regresando al Menu
```

Figura 42 - Ejecución de Dosis en tiempo real.

- Cálculo de mL para cierta actividad requerida.

```
Seleccione una Opción... 4
Ingrese Dosis con PUNTO(.) y solo 2 decimales
1
Los ml necesarios para tener 1.0 mCi, son: 12.9 mL
Regresando al Menu
```

Figura 43 - Ejecución de cálculo de mL para cierta actividad.

- Ingreso de datos nuevos para ingreso de paciente.

```
Seleccione una Opción... 5
Ingrese Dosis con PUNTO(.) y solo 2 decimales
10
Ingrese hora en formato HH:MM
11:00
Ingrese cantidad de mL con PUNTO(.) y solo 2 decimales
0.5
Datos ingresados con éxito
```

Figura 44 - Ejecución de ingreso de datos nuevos por paciente.

- Generación de reporte.

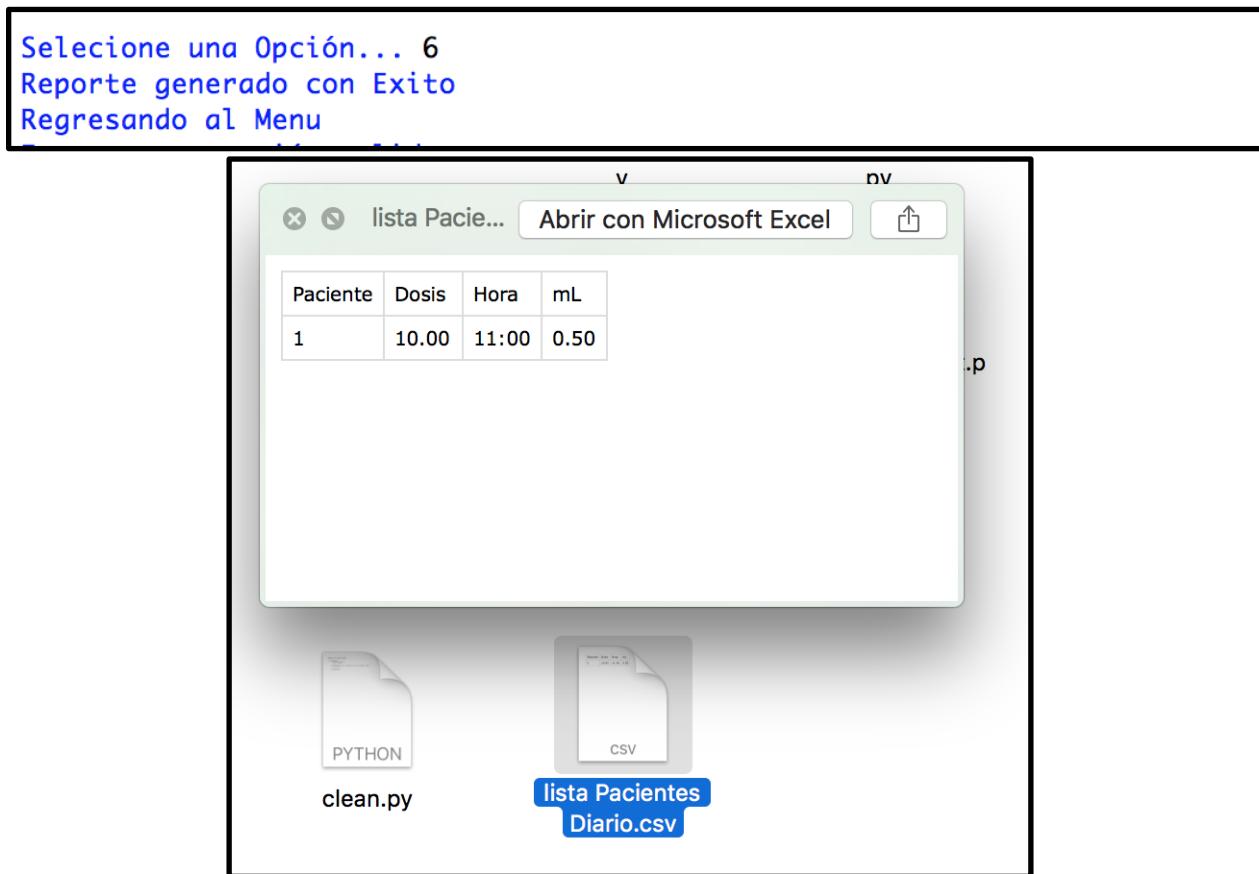


Figura 45 - Generación de reporte.

- Modificar ultimo paciente.

```
Seleccione una Opción... 7
Ingese una opción valida
** Importante **
Los datos modificados solo son aplicables al ULTIMO paciente ingresado
Selecciona Opción:
** ::::::::::::::::::::: **
Modificar Dosis  :-> 1
Modificar Hora   :-> 2
Modificar mL     :-> 3
Volver al menú  :-> s
** ::::::::::::::::::::: **
Seleccione una Opción... |
```

Figura 46 - Modificación datos de ultimo paciente.

Dentro de los principales problemas que se fueron dilucidando a la fecha están:

- Falta de conocimientos para programar interfaces gráficas modernas y avanzadas.
 - Se está realizando cursos de interfaces gráficas compatibles con python 3 acordes con la necesidad del cliente.
- Los stakeholders han realizado observaciones en el prototipo que han generado nuevos módulos y modificación del código.
 - Se realizaron módulos adicionales que serán detallados más adelante.
- Problemas para la realización de las pruebas in-situ, debido a que la máquina sufrió un desperfecto y el trabajo fue detenido durante una semana, por lo que hubo que reagendar el periodo de pruebas.
 - Se lograron realizar una serie de pruebas que lograron cumplir con ciertos requisitos.

Cabe destacar que los resultados han sido expuestos al cliente el dia viernes 7 de diciembre 2018, en una reunión de avance que solicito de manera extraordinaria. Los avances han generado un impacto positivo, por lo cual tenemos la aprobación de los módulos implementados por parte del cliente en fase teórica hasta la actualidad.

El dia 24 de diciembre del 2018 se solicitó la autorización para el inicio de pruebas in-situ de la aplicación, ésta fue realizada en conjunto con el stakeholder Jose Perez Astorga Tecnólogo Médico a cargo esa semana del funcionamiento del PET.

Dentro de las pruebas realizadas por la aplicación directamente desde la shell de python (consola). Se logró la aprobación del funcionamiento de todos los módulos implementados por parte del stakeholder. La aplicación fue capaz de ingresar datos, procesar datos, entregar resultados, generar reporte. Ciertas consideraciones fueron realizadas para mejorar la aplicación.

El dia 4 de Enero del 2018 se realizó la primera entrega funcional del Software, versión 0.3 de PET Manager al Servicio de PET de la clínica Reñaca. La Entrega se realizó en una reunión con el Cliente y el Stakeholder Jose Perez, donde realizó una serie de pruebas de verificación de funcionalidades y Requisitos establecidos.

La aplicación paso por un proceso de observaciones y se evaluaron varios ítems mostrados en la siguiente pauta:

HOJA DE ACEPTACION				
Proyecto	PETManager			
Entregable	Software			
Autor	Sebastian Inostroza H.			
Versión/Edición	Beta	Fecha Versión	04/01/2019	
Aprobado por	Felipe Vera Muñoz	Fecha Aprobación	04/01/2019	
		Nº Total de pruebas	6	
Caso de Prueba	Malo	Regular	Bueno	Observación
Instalación correcta			X	S/O
Visualización de la Aplicación		X		Pendiente mejorar GUI
Funcionalidad de Aplicación			X	Realizar más test
Generación de Archivos			X	S/O
Facilidad de uso		X		Podría mejorar

Aprobado para la utilización en Marcha blanca del Servicio de PET CT Clínica Bupa Reñaca.

TM Felipe Vera Muñoz
Jefe Tecnólogos Medica

Figura 47 – Formato Aceptación version liberada.

Dentro de las observaciones generadas se encuentran:

Cuadro de Observaciones propuestas por los Stakeholders en ambiente de pruebas.

Modulo	Stakeholder	Observaciones
Interfaz Usuario	F.Vera	Solo mejora estética y escalabilidad.
Interfaz Usuario	J.Perez	Detalle de presentación y visual.
Check_inputs	F.Vera	Funcional
Check_inputs	J.Perez	Funcional
Ingreso de Dosis	F.Vera	Ajustado a lo requerido
Ingreso de Dosis	J.Perez	Funcional
Dosis Proyectada	F.Vera	-
Dosis Proyectada	J.Perez	Valores aceptados
Dosis Tiempo Real	F.Vera	Valor aceptado
Dosis Tiempo Real	J.Perez	Tener un contador de Pcte. y mL junto a este resultado
mL para dosis específica	F.Vera	-
mL para dosis específica	J.Perez	Funcional
Ingreso de nuevo dato pcte	F.Vera	Cumple lo solicitado
Ingreso de nuevo dato pcte	J.Perez	Funcional
Generar reporte	F.Vera	Acorde a los solicitado
Generar reporte	J.Perez	Cumple lo solicitado
Modificar ultimo dato	F.Vera	Funcional
Modificar ultimo dato	J.Perez	Funcional

Tabla 9 – Observacion de Stakeholders

Objetivos específicos vs resultados pruebas in-Situ versión 0.3

Respecto a los resultados obtenidos vs los objetivos específicos tenemos a la fecha:

Objetivo Específico	Metrica	Unidad	Criterio de Éxito	Método
1	Margen de error	%mCi	<5	Comparar con activímetro
2	Disminución Dosimetría	%mSv	<5	Revisión Informe dosimétricos CCHEN
3	Margen de error	%mL	<10	Comparar con jeringas medidoras
4	Comprobación valores	Segundo	<1	Comprobador de valores de tiempo.
5	Tiempo Procesamiento	Segundo	<1	Medidor de tiempo de Procesamiento.
6	Margen de error	%mCi	<20	Comparar actividad dosificada con activímetro.

Tabla 10 - Obj.Específicos Avances

Objetivos Específicos	% Avance en ambiente de pruebas
1	80%
2	Sin medición
3	Sin medición
4	100%
5	100%
6	100%

Tabla 11 -Porcentaje cumplimiento Obj.Específicos.

Evidencia

Objetivo específico 1:

Se realizaron mediciones en tiempo real de la aplicación y el activímetro.

Para cierta hora al finalizar la jornada se midió directamente en el activímetro la cantidad de radiofármaco sobrante. El resultado fue el siguiente:

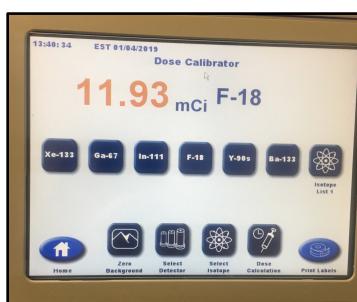


Figura 48 - Medición de prueba con activímetro.

Medición de la aplicación:

Seleccione una Opción... 3
la Dosis para este momento es; 13.151 mCi
Regresando al Menu

Figura 49- Medición de prueba con aplicación.

Según los cálculos existe una diferencia de un margen de error del 10,2%.

Esta variación, hemos concluido, se debe principalmente a que los relojes del activímetro y del computador personal no se encontraban en sincronía. El activímetro registraba 5 minutos adicionales de diferencia, por lo que este cambio podría acercar más el valor al margen de error propuesto.

Se establecieron más periodos de prueba para poder modificar los relojes del activímetro o Pc.

Objetivo específico 2:

El objetivo específico 2 necesita de un tiempo adicional para ser evaluado, pues los informes dosimétricos de la cchen (comisión chilena de energía nuclear) son trimestrales y deberían medirse a partir de la implementación diaria de la aplicación.

Objetivo específico 3:

Se necesitan permisos adicionales para mediciones de fármaco restante en el contendor. Pues implica sacar de la fuente principal la cantidad restante de mL. Se están gestionando los permisos para tales pruebas

Objetivo específico 4:

El objetivo puede ser demostrado a través de código de la aplicación con los módulos calculadores de tiempo, su formato es en minutos como unidad mínima y la actualización por ende de los cálculos también aplica como mínimo al minuto como tiempo de medición.

```
def dif_min_proy(time_1,time_2):
    start_dt = dt.datetime.strptime(time_2, '%H:%M')
    end_dt = dt.datetime.strptime(time_1, '%H:%M')
    diff = (start_dt - end_dt)
    minutos= int(diff.seconds/60)
    return minutos
```

```
def dif_min(time_x):
    now= time.strftime('%H:%M')
    start_dt = dt.datetime.strptime(now, '%H:%M')
    end_dt = dt.datetime.strptime(time_x, '%H:%M')
    diff = (start_dt - end_dt)
    minutos= int(diff.seconds/60)
    return minutos
```

Figura 50 - Modulos calculadoras.

Objetivo específico 5:

Para la demostración de los tiempos utilizados se modificará el código y se expondrá a continuación:

```
#FUNCION DE DOSIS PROYECTADA
def dose_proy(lista):
    import time
    hour_x=check_input_hora()
    time_ini=time.time()
    act_ini=dose_last(lista)
    time=time_last(lista)
    minutos=dif_min_proy(time,hour_x)
    dose=cal_decay(act_ini,minutos)
    dose=round(dose,3)
    print ("la dosis proyectada para las",hour_x," es", dose, "mCi")
    import time
    time_fin=time.time()
    time_tot=time_fin - time_ini
    print("El tiempo total de ejecucion del modulo es:", time_tot, "segundos")
    return dose
```

Figura 51- Modificación de código para prueba.

Se utilizó la función time.time() y el resultado se presentó por consola.

```
Seleccione una Opción... 2
Ingrese hora en formato HH:MM
11:00
la dosis proyectada para las 11:00 es 82.163 mCi
El tiempo total de ejecucion del modulo es: 0.19755101203918457 segundos
Regresando al Menu
```

Figura 52 - Resultado por consola de prueba.

Se acepta resultado para criterio de 1 segundo de ejecución.

Objetivo específico 6:

Se realizó una prueba específica de cálculo de mL expuesta a continuación:

Se solicitó una cantidad de 9 mCi para que la aplicación calculará cuántos mL eran necesarios.

```
Seleccione una Opción... 4
Ingrese Dosis con punto y solo 2 decimales9
9.00
Los ml necesarios para tener 9.0 mCi, son: 0.8 ml
Regresando al Menu
** :::::::::::::::::::: **
```

Figura 53 -Solicitud de mCi para prueba.

El software calculo 0,8 mL, los cuales fueron sacados del contenedor para ser medidos y corroborados.



Figura 54 -Evidencia de los mL recolectados.

Se procede a la medición en el activímetro.

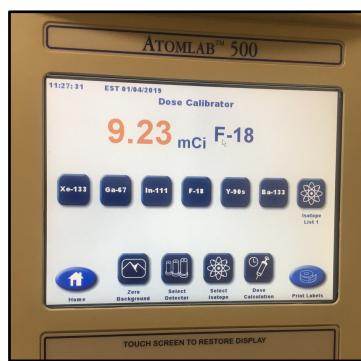


Figura 55 -Medición de los mL recolectados.

Según los cálculos realizados el margen de error es 2.5% para esta prueba, siendo aceptado como margen un 20%.

Debido a la radiación adicional que produce realizar las pruebas adicionales, éstas sólo han sido realizadas en momentos específicos donde la radiación ha decaído más de la mitad. Además se han establecido nuevas fechas para testear la aplicación durante la jornada con los distintos stakeholders.

4.2. Diseño detallado.

Diagrama Módulo ingreso de dosis inicial.

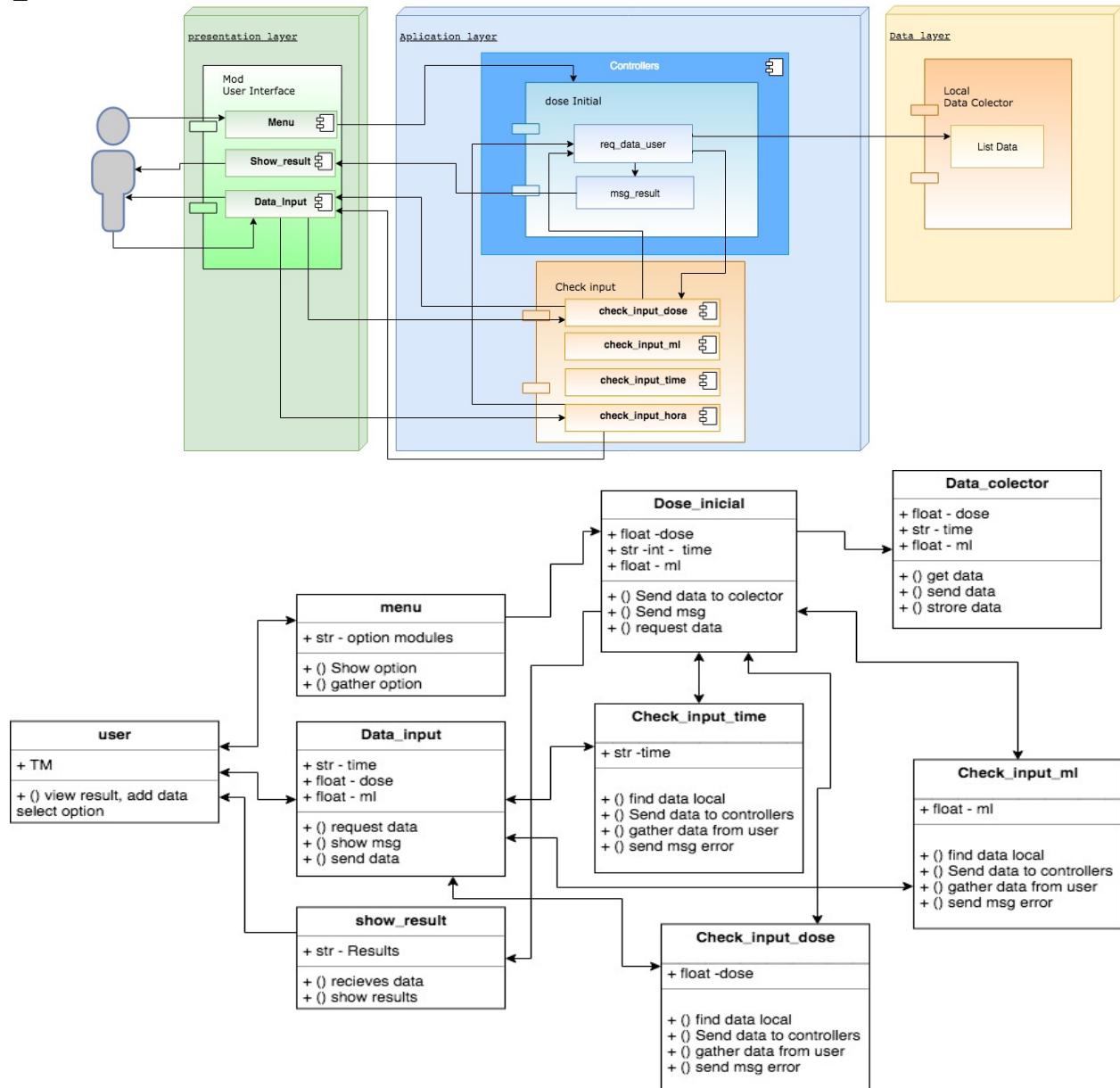


Figura 56. Diseño detallado para módulo ingreso de dosis Inicial.

Diagrama Módulo Dosis en Tiempo Real.

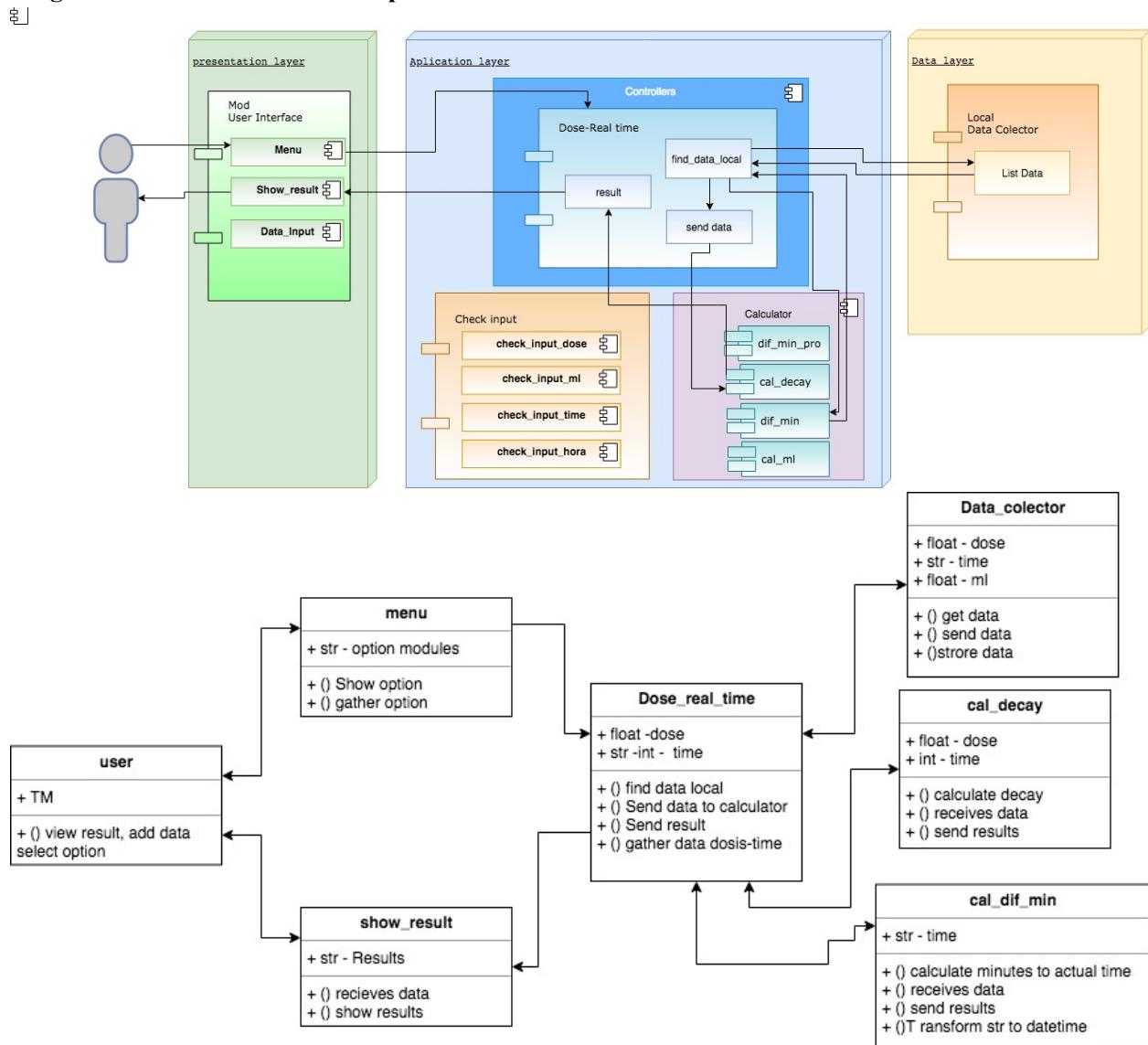


Figura 57. Diseño detallado para módulo Dosis en tiempo real.

Diagrama Módulo Dosis Proyectada.

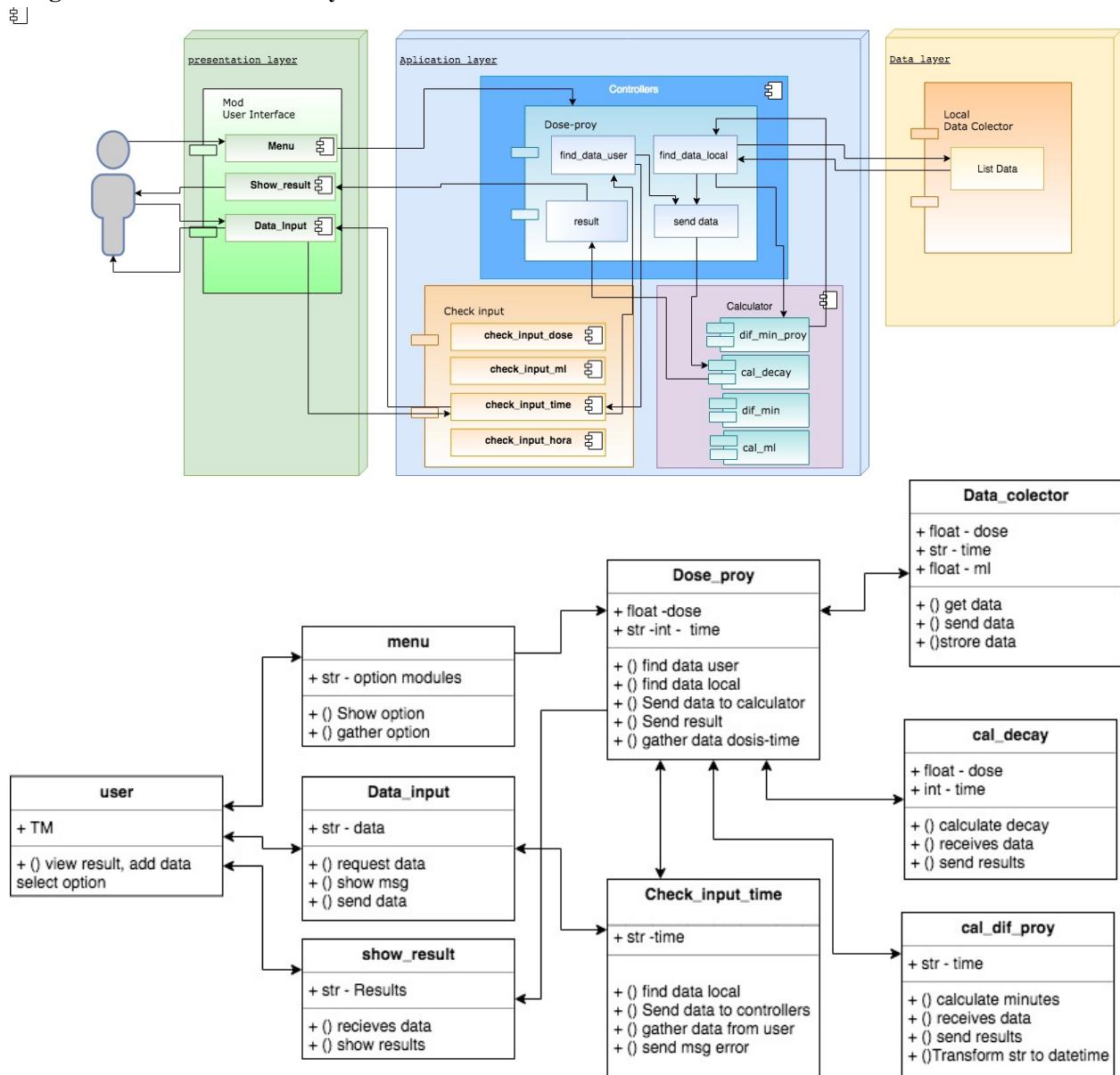


Figura 58 . Diseño detallado para módulo Dosis proyectada en el tiempo.

Diagrama Módulo cálculo mL para actividad específica.

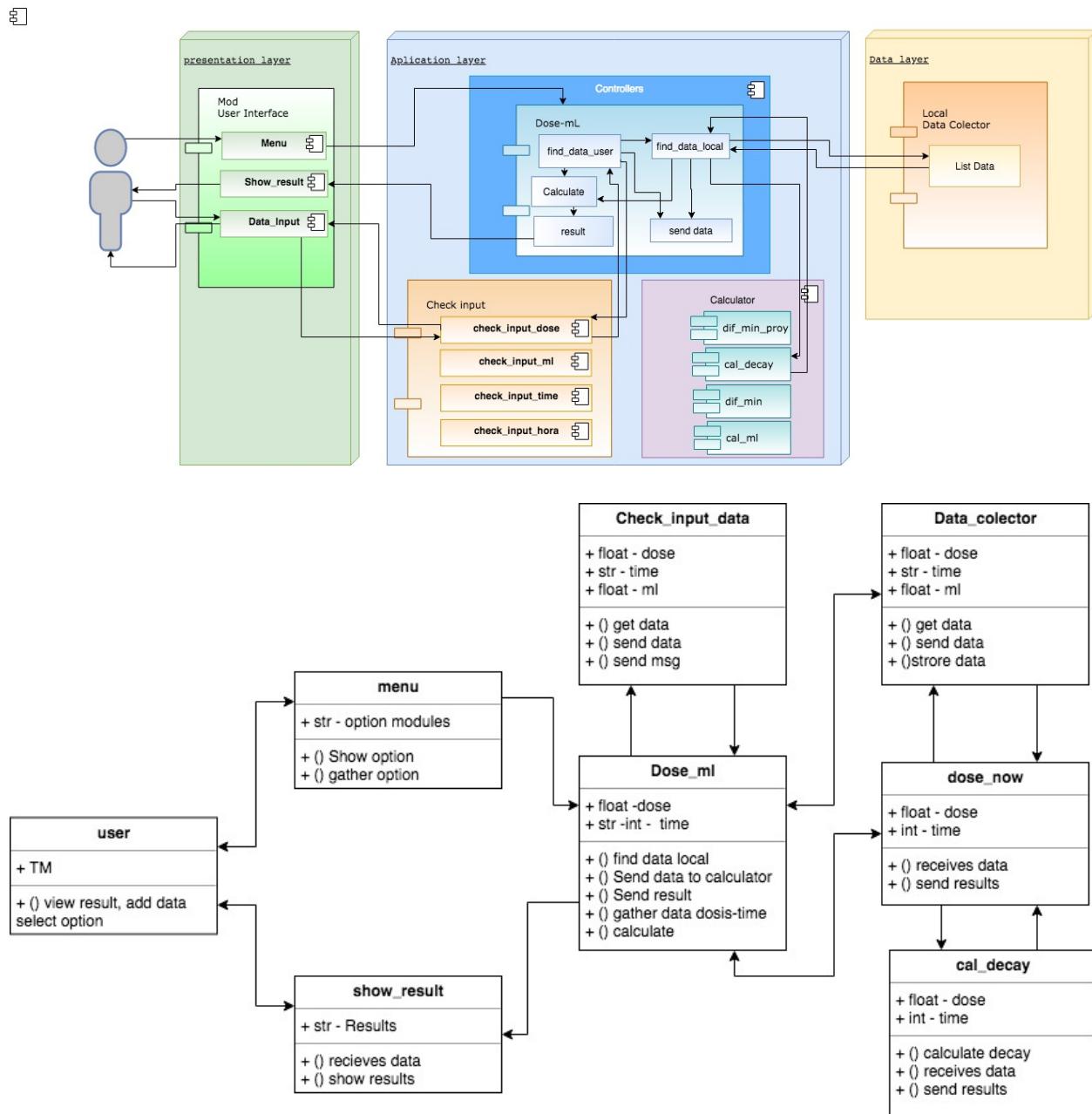


Figura 59. Diseño detallado para módulo de cálculo de mL para actividad específica.

Diagrama Módulo Ingreso de dato actualizado(Nuevo paciente).

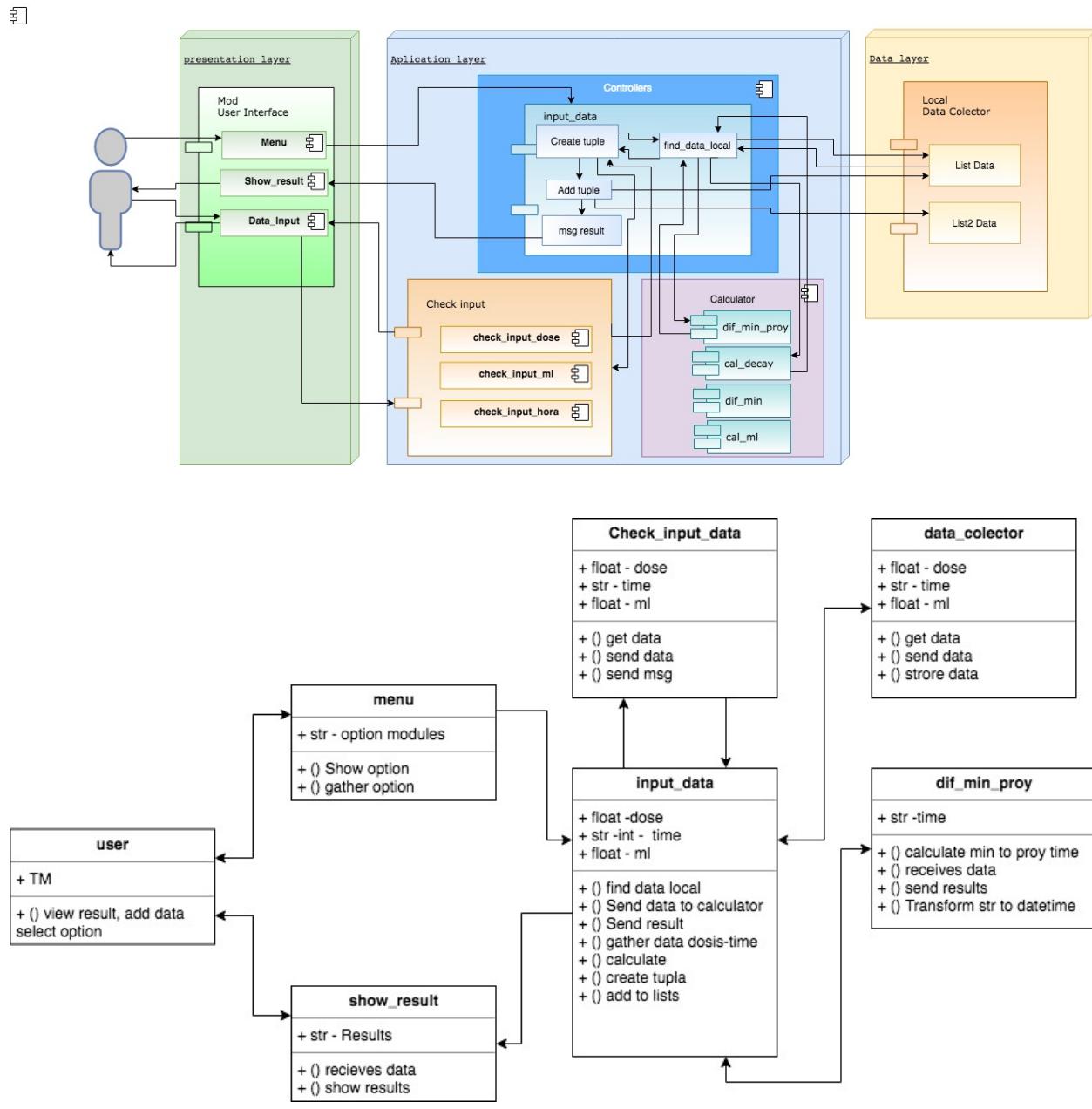


Figura 60. Diseño detallado para módulo de ingreso de nuevo dato.

Diagrama Módulo Generador de reporte.

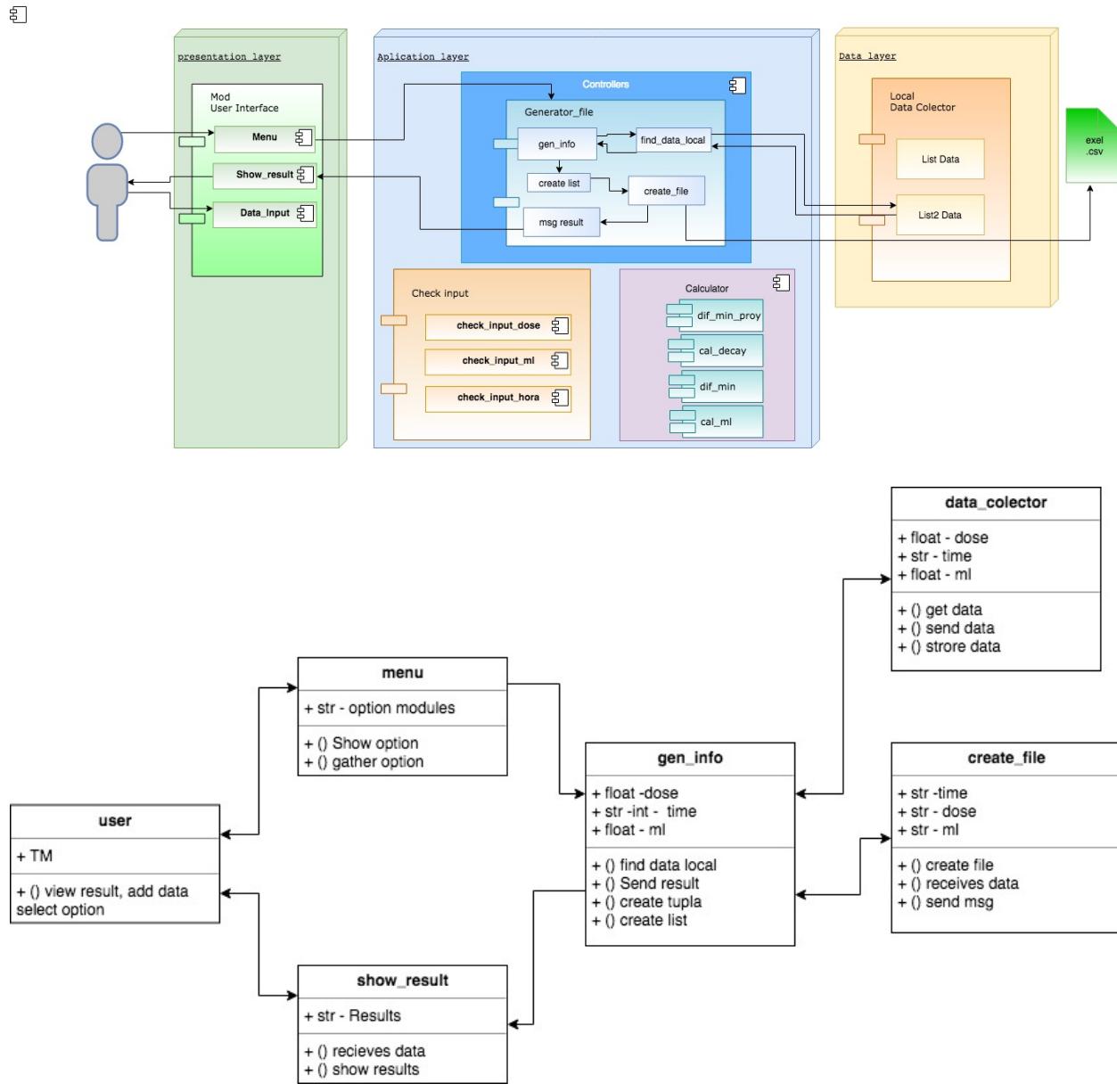


Figura 61. Diseño detallado para generador de reporte.

Diagrama Módulo Modificación de último dato.

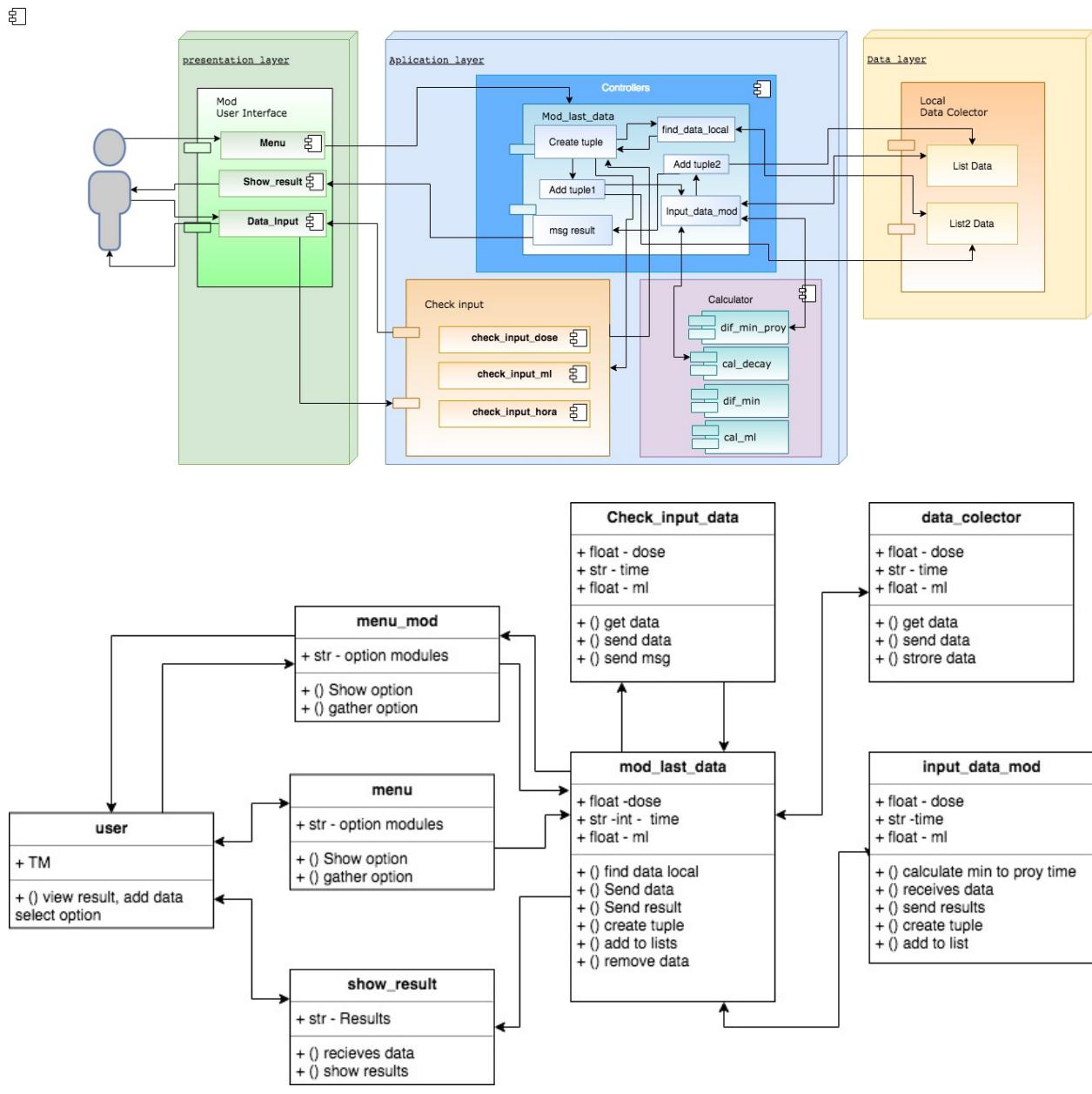


Figura 62. Diseño detallado para modificador de último dato paciente.

4.3. Evidencia de control de cambios. .

Una de las herramientas de Github para el control de cambios, es la utilización de gráficos para visualizar la frecuencia y cantidad de cambios que se han realizado al proyecto.

Se muestran algunas evidencias a continuación:

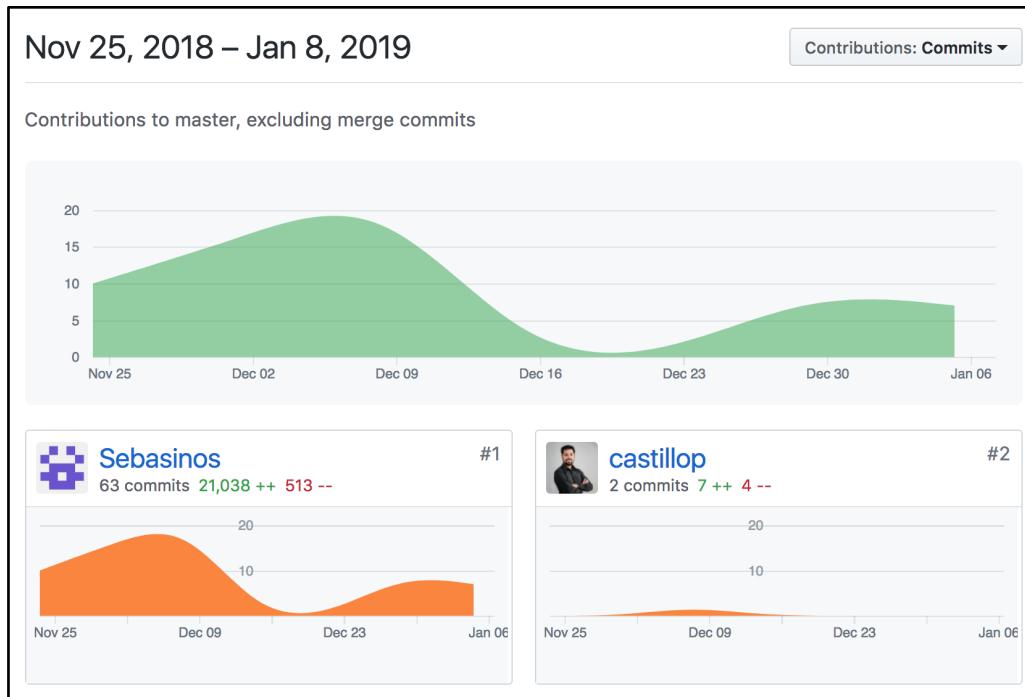


Figura 63. Evidencia control de cambios realizado al proyecto PETManager - Github.

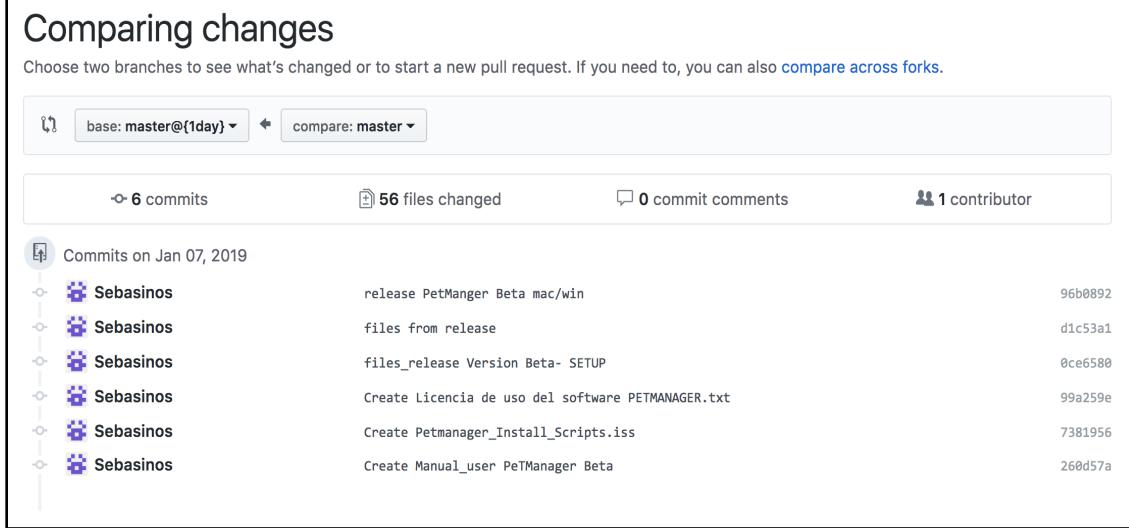


Figura 64. Evidencia control de cambios realizado el dia 7 enero 2019 - Github.

	Showing 56 changed files with 19,333 additions and 0 deletions.	Unified	Split
	tools/install_files_PETManager_Beta/Licencia de uso del software PETMANAGER.txt	+1 -0	
	tools/install_files_PETManager_Beta/PetManager_Setup.exe	BIN	
	tools/install_files_PETManager_Beta/Petmanager_Install_Scripts.iss	+44 -0	
	tools/install_files_PETManager_Beta/User_Manual_PetManager.pdf	BIN	
	tools/petmanager-vbeta -mac-files/_pycache_/petbeta.cpython-37.pyc	BIN	
	tools/petmanager-vbeta -mac-files/build/petbeta/Analysis-00.toc	+542 -0	
	tools/petmanager-vbeta -mac-files/build/petbeta/BUNDLE-00.toc	+1 -0	
	tools/petmanager-vbeta -mac-files/build/petbeta/EXE-00.toc	+163 -0	
	tools/petmanager-vbeta -mac-files/build/petbeta/PKG-00.pkg	BIN	
	tools/petmanager-vbeta -mac-files/build/petbeta/PKG-00.toc	+156 -0	
	tools/petmanager-vbeta -mac-files/build/petbeta/PYZ-00.pyz	BIN	
	tools/petmanager-vbeta -mac-files/build/petbeta/PYZ-00.toc	+403 -0	
	tools/petmanager-vbeta -mac-files/build/petbeta/base_library.zip	BIN	
	tools/petmanager-vbeta -mac-files/build/petbeta/localpycos/pyimod01_os_path.pyc	BIN	

Figura 65. Evidencia archivos modificados - Github.

4.4. Situación futura.

- El primer prototipo entregado a los stakeholders permite la realización del ingreso de datos iniciales, realizar calculo de dosis en tiempo real y cálculo de dosis proyectada. Avalado por la teoría de las fórmulas de decaimiento, ha tenido resultados positivos, los cuales han logrado mitigar en parte la probabilidad de no lograr el funcionamiento básico de la aplicación.
- El Segundo prototipo entregado fue el dia 24 de diciembre, este permitió la realización de todas las funcionalidades propuestas en el inicio, se realizaron pruebas de software para medir los objetivos específicos y se produjo feedback para la implementación de nuevos requisitos.
- La liberación del producto V 0.3 fue realizada el día 4 de Enero del 2019, el cliente Felipe Vera aceptó la versión con ciertas observaciones a implementar para la siguiente versión.
- El Software operará desde el 7 de enero en ambiente de pruebas por todos los stakeholders.

Para la realización de las futuras tareas se ha realizado una nueva Planificación.

- Los Objetivos de los futuros entregables son:
 - **V0.4 (spring 4)**:Modificaciones de código para disminuir margen de error en mediciones.
 - **V0.5 (spring 5)**:Iniciar el trabajo de interfaz gráfico para aplicación de escritorio-SO.windows (Definido por cliente).
 - **V0.6 (spring 6)**:Envio de Documento generado por correo electrónico.

	Nombre	Inicio	Vencimiento	Duraci...
	PET Manager	6 Nov 2018	18 Mar 2019	132d
1	Incepcion	6 Nov 2018	20 Nov 2018	15d
5	Iteración 1	26 Nov 2018	11 Dic 2018	16d
12	Iteración 2	11 Dic 2018	26 Dic 2018	16d
25	Iteración 3	25 Dic 2018	7 Ene 2019	14d
39	Iteración 4	4 Ene 2019	21 Ene 2019	18d
45	Iteración 5	22 Ene 2019	18 Feb 2019	27d
47	Iteración 6	19 Feb 2019	18 Mar 2019	27d

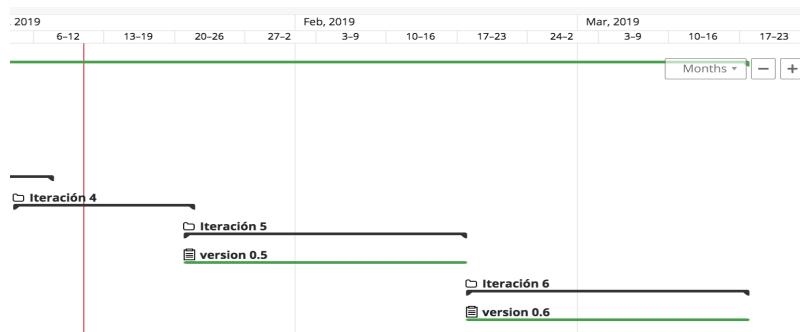


Figura 66. Planificación nuevos entregables.

4.4.1. Resultados esperados.

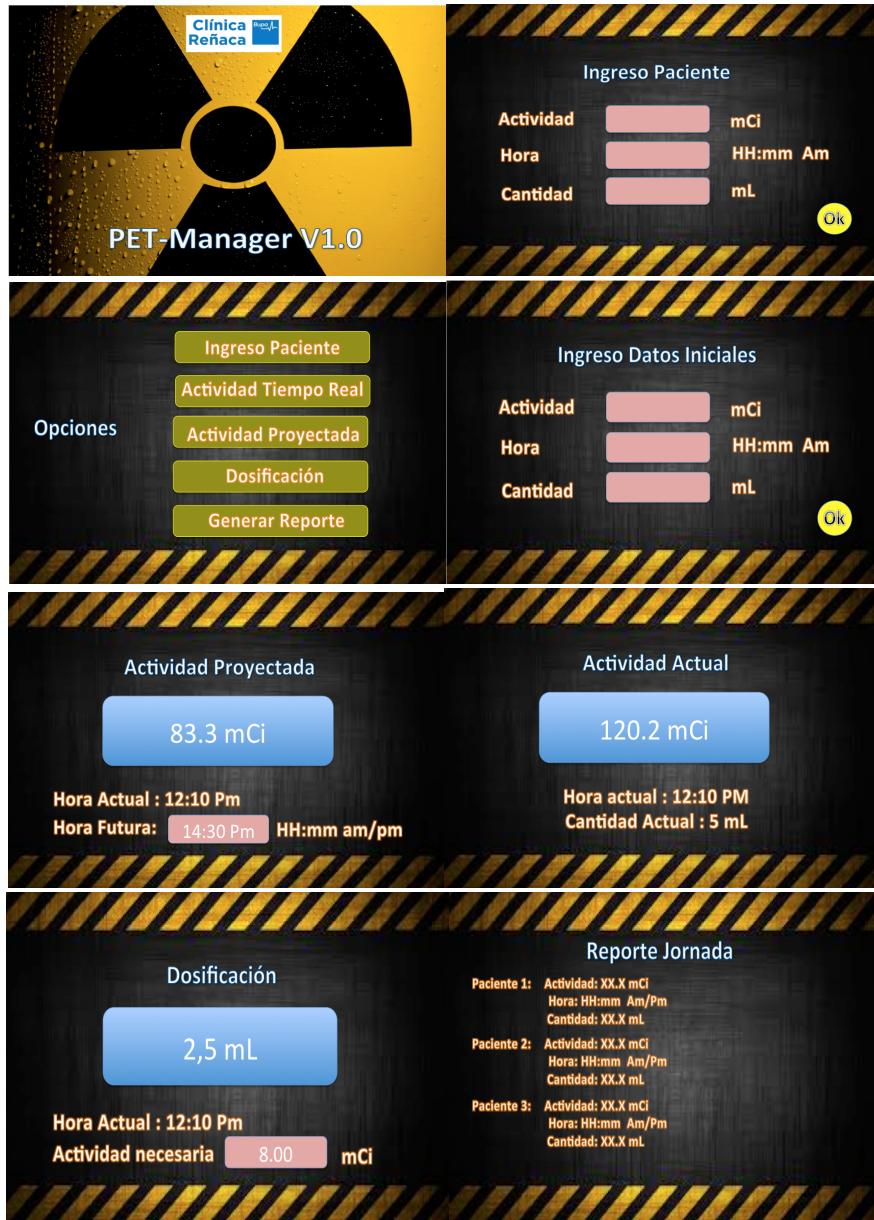


Figura 67 - Cuadro App PET Manager esperados

Capítulo 5: Conclusiones

5. Conclusiones.

5.1. Conclusión.

Conclusiones de la primera parte del proyecto:

- La planificación del Proyecto vs los Objetivos propuestos han sido aceptados por el cliente y por el equipo de trabajo.
- Se han respetado los tiempos de planificación y los entregables al término de los hitos, para generar el feedback con los stakeholders.
- Se ha realizado cambios a lo largo del proyecto que incluyen la creación de módulos sugeridos por los stakeholders y profesores en la presentación de los hitos.

El proyecto ha evolucionado acorde a la metodología ágil establecida. Se ha logrado liberar el producto en las fechas establecidas.

El cliente ha aceptado la entrega formal de la versión 0.3 de la aplicación, el objetivo actual será probar con esta versión todas las funcionalidades implementadas.

Dentro de las pruebas de aceptación realizadas con las versiones anteriores y las mitigaciones de los riesgos, se ha demostrado que la aplicación es funcional y tendrá la capacidad de apoyar el trabajo de los tecnólogos médicos del área de PET.

La utilización de esta herramienta busca cumplir con todos los objetivos propuestos y además los nuevos objetivos que comenzarán a dilucidar por parte del cliente y los usuarios finales.

El equipo de PET clínica reñaca está muy optimista en conseguir una aplicación que cumpla con las expectativas generadas, y así en conjunto, realizar un trabajo más prolífico y con menos incertidumbre. Para los usuarios es muy importante entregar un examen de calidad y óptimo para los pacientes. Este tipo de atención con estándares de calidad y leyes de protección hacia el personal y los pacientes debe tener un apoyo importante, el cual puede mejorar y aumentar todos los parámetros positivos de trabajo para los Tecnólogos Médicos y los pacientes.

El proyecto de PET Manager está en constante evolución, y se espera tenga una mayor proyección para el trabajo médico actual en el área de Medicina Nuclear y Tomografía computada.

5.2. Problemas abiertos.

A la fecha con la entrega de la primera versión funcional de PET Manager, podemos mencionar los siguientes problemas abiertos:

- Modificación de Código para ser implementado con los gestores de interfaz gráfica solicitados.
- Estudio especializado por parte del equipo para implementar interfaces gráficas de tipo app desktop.
- Problemas para realizar mayor cantidad de pruebas de aceptación relacionadas con la radiación adicional que recibe el personal.
- La interfaz gráfica de la versión es funcional, pero está lejos de cumplir con varios de los requerimientos no funcionales propuestos.

5.3. PostMortem.

Luego de la entrega funcional del producto y liberación de la versión 0.3 el cliente y Stakeholders han realizado observaciones a considerar para la nueva versión, además se han llegado a conclusiones por parte del equipo completo de trabajo las cuales podemos plasmar en esta sección:

¿Qué es lo que funcionó?

- Las funcionalidades propuestas están operativas y funcionando.
- El código compila sin errores.
- Las pruebas realizadas fueron logradas en conjunto con los stakeholders.
- Las entregas se realizaron en los plazos estipulados.
- La metodología actual del proyecto es cómoda y efectiva para el equipo de trabajo.

¿Qué es lo que resultó mal?

- No se logró Implementar una interfaz gráfica acorde a lo propuesto en el proyecto.
- Las pruebas no fueron realizadas en los períodos programados.
- Las pruebas no dieron resultados precisos y no se cumplieron algunos objetivos
- La funcionalidad de envío de correo no fue gestionada en esta parte del proyecto, dejando un objetivo sin cumplir.

¿Cómo respondió el equipo a los problemas Generales?

- Problemas del desarrollador:
 - Falta de conocimientos: Cursos y tutoriales en la materia.
 - Falta de Tiempo: Mejor gestión del tiempo, sacrificar horas no planificadas.
 - Falta de Orden: Cambio de metodología en el proyecto.

- Problemas de Cliente y Stakeholders:

- No se respetan horas de reunión: Reprogramacion.
- Stakeholders no puede poner atención en Proyecto : Reprogramacion jornada.
- Equipo PET falla: Reprogramación de las tareas de prueba.

Respuesta a los problemas planteados:

- No se logró Implementar una interfaz gráfica acorde a lo propuesto en el proyecto./ R. Se aplazó la implementación para tener un mayor conocimiento por parte del desarrollador y especificación del requisito por parte del cliente.
- Las pruebas no fueron realizadas en los periodos programados./
R. Se reprogramaron las pruebas para ser realizadas lo más pronto posible y no alterar las fechas de entrega.
- Las pruebas no dieron resultados precisos y no se cumplieron algunos objetivos./
R. Se crearon nuevas tareas para modificar y evaluar los módulos involucrados.
- La funcionalidad de envío de correo no fue gestionada en esta parte del proyecto, dejando un objetivo sin cumplir.
R. Se planificó para ser implementada en la versión posterior, resolviendo primero algunos objetivos y requisitos más críticos.

¿Qué aprendimos?:

- La buena planificación de las tareas permite una trabajo más fluido.
- Realizar entregas periódicas es una buena forma de evaluar el Software para el cliente.
- El Software debe tener la capacidad de ser modificable, pues el cliente aspira a nuevas funcionalidades para las futuras entregas.
- El Cliente y los stakeholders son un buen equipo de trabajo, responsable y dan soluciones a las problemáticas que surgen.

5.4. Trabajo futuro.

Se deberá realizar una nueva planificación para los proximos eventos relacionados.

Puesto que la Primera parte del proyecto dio como resultado un SW operacional basado en la resolución de cálculos para la dosificación del radiofármaco Flúor FDG.

Dentro de la primera parte del Proyecto se centraron los esfuerzos en desarrollar los módulos capaces de generar información válida para trabajar con una lista de pacientes, y poder cumplir los objetivos propuestos.

Sobre la Metodología, se decidió establecer métricas acompañadas para generar información adicional para el proyecto.

Se establecerá como base las métricas:

Cycle Time: Es la métrica que registra el tiempo que sucede entre el inicio y el final del proceso, para un ítem de trabajo dado. Se suele medir en días de trabajo.

Lead Time: Es la métrica que registra el tiempo que sucede entre el momento en el cual se está pidiendo un ítem de trabajo y el momento de su entrega (el final del proceso). Se suele medir en días de trabajo.

Sobre la última versión, la entrega se realizó sin mayores observaciones, nuestro cliente puso énfasis en la capacidad que deberá tener la aplicación de ser escalable.

Desea para desarrollar como trabajo futuro:

- Permitir desarrollar las funciones con otro tipo de Radiofármaco.
- Establecer directrices de obligatoriedad en el uso de la aplicación, siendo la aplicación la responsable de exigir los datos de entrada.
- Ser capaz de planificar y notificar ciertos eventos relacionados con el flujo de los pacientes.
- Utilización desde cualquier dispositivo personal (Celulares, tablet, pc) de los operadores.