Proyecto Final

Estefanía García González, Sebastián Mora Sabogal

11 de mayo de 2019

Índice general

I PROYECTO **7**

1. Caso de Estudio **9**
1.1. Introducción 9
1.2. Objetivo General 9
1.3. Objetivos Específicos 9
1.4. Descripción del problema 10
1.5. Alcance 10

2. Metodología **11**
2.1. Introducción 11
2.2. Proceso de Software 12
2.2.1. Metodología de implementación 13
2.3. Open Source 13

II DISEÑO **15**

3. Requerimientos **17**
3.1. Introducción 17
3.2. Requerimientos del Cliente 18
3.3. Casos de uso 19
3.4. Diagramas de secuencia 21
3.5. Diagramas de comunicación 22

4. Interacción **29**
4.1. Introducción 29

5. Clases **31**
5.1. Introducción 31

6. Patrones **33**
6.1. Introducción 33
6.2. Prototipo 34

7. Estados **35**
7.1. Introducción 35

8. Componentes **37**
8.1. Introducción 37

9. Nodos **39**
9.1. Introducción 39

10. Actividades **41**
10.1. Introducción 41

III REFLEXIONES **43**

11. Conclusiones **47**
11.1. Introducción 47

Índice de figuras

2.1. 13

3.1. Primer diagrama de caso de uso 19
3.2. Segundo diagrama de caso de uso 20
3.3. Tercero diagrama de caso de uso 20
3.4. Cuarto diagrama de caso de uso 21
3.5. Diagrama de secuencia caso de uso 14. 21
3.6. Diagrama de secuencia caso de uso 15. 21
3.7. Diagrama de secuencia caso de uso 16. 21
3.8. Diagrama de secuencia caso de uso 19. 21
3.9. Diagrama de comunicación caso de uso 14. 22
3.10. Diagrama de comunicación caso de uso 15. 22
3.11. Diagrama de comunicación caso de uso 16. 22
3.12. Diagrama de comunicación caso de uso 19. 22

Parte I

PROYECTO

Capítulo 1

Caso de Estudio

1.1. Introducción

Desde que el ser humano cuenta con raciocinio , ha buscado organizarse, desarrollar metodologías y nuevas tecnologías que faciliten su diario vivir. Ha habido un recorrido histórico en el cual las necesidades humanas de optimización de tiempo y recursos han ido en aumento, así mismo las soluciones a éstas. En los últimos años se ha podido apreciar una constante migración al uso de tecnologías de la información que permiten realizar a cabo tareas en todos los ámbitos de forma óptima. Uno de los actores que más se han visto inmersos en la revolución digital son los estudiantes, pero en su contexto universitario, hace falta desarrollar estrategias que le permitan mejorar la gestión de tiempo de sus actividades académicas; por lo cual se buscará una solución tecnológica que se adapte a las necesidades de los universitarios.

1.2. Objetivo General

Desarrollar un software que gestione actividades y tiempos de las asignaciones académicas a estudiantes universitarios, utilizando los modelos y metodologías de ingeniería de software para mejorar la productividad del universitario.

1.3. Objetivos Específicos

1. Analizar el problema teniendo en cuenta la observación de las necesidades del estudiante, para así enfocarse en estos elementos primordiales a la hora de desarrollar el software.

2. Presentar una solución a nivel de software a partir del previo análisis del problema para finalmente implementarlo.

1.4. Descripción del problema

La vida universitaria y académica suele ser difícil de manejar debido a la cantidad de trabajos que se deben entregar diariamente, a la prioridad que cada una es para el usuario y a la gestión de tiempo para poder realizarlos. Tareas, trabajos, talleres y grandes proyectos son algunas de las actividades que un estudiante realiza durante su semestre; además de que cada uno tiene complejidad y tiempo de realización diferentes estimados por el estudiante. Una solución factible es la utilización de un software gestor de tareas orientado a la organización y optimización de actividades académicas.

1.5. Alcance

Este software tendrá la capacidad de gestionar los horarios de los estudiantes, añadir recordatorios de trabajos próximos a presentar y ofrecer el servicio de organizar en horarios la realización de las tareas pendientes. Esto se llevará a cabo de acuerdo a la complejidad de la actividad a realizar, en la cual se tomará en cuenta el nivel de dificultad, si se puede desarrollar en diferentes etapas y la fecha de entrega.

El estudiante estará en la capacidad de añadir actividades, determinar la complejidad de éstas y asignarles un horario de realización que puede ser repartido en varios bloques cuando la tarea requiere de mucho tiempo. Adicionalmente, las actividades podrán personalizarse añadiendoles objetivos a cumplir o subactividades.

Capítulo 2

Metodología

2.1. Introducción

La metodología del proceso de software que se debe seguir, es fundamental pues define las acciones generales que se deben llevar, a modo de conseguir un desarrollo del proyecto optimo, pasando por cada una de las fases del proceso elegido.

2.2. Proceso de Software

Parte importante de un proyecto de software es definir el, o los ciclos de vida que se manejarán dentro del proyecto, ya que estos determinarán estrategias para planificar,desarrollar y mantener el software. Por esta razón, se definirá el modelo de procesos a utilizar, tomando en cuenta los siguientes criterios:

- Es necesaria una metodología que sea pertinente para un proyecto de software pequeño con pocos desarrolladores.
- Se considera importante la verificación en cada fase del ciclo de vida, ya que permite sentar buenas bases dentro del proyecto y reducir el riesgo.
- Además de la verificación, es necesaria una retroalimentación constante, ya que es posible ver con mayor claridad las falencias y carencias del proyecto.
- Como último criterio fundamental, se contempla la necesidad de desarrollar algunas partes de software de forma rápida, ya que esto facilitaría la retroalimentación del sistema.

Para cumplir con las pautas anteriormente mencionadas, los ciclos de vida que se elegirán son prototipo y V. Cada uno de estos modelos obedece solo a algunas de las especificaciones, pero juntos se complementan de la siguiente manera:

- El modelo V es perfecto para equipos de trabajo pequeños, ya que es sencillo, de fácil aprendizaje, robusto e incluye pruebas en cada fase, lo que facilita el trabajo cuando hay pocas personas.
- Gracias a los dos ciclos de vida, es posible hacer una verificación y retroalimentación de forma efectiva, ya que con el modelo en V se hacen pruebas en cada fase y con el prototipo es posible obtener resultados a corto plazo que se pueden ir revisando y evaluando.
- El modelo de prototipo brinda la posibilidad de construir partes del proyecto de forma prematura, por lo que es posible realizar pruebas y verificar qué cosas es necesario cambiar o añadir.

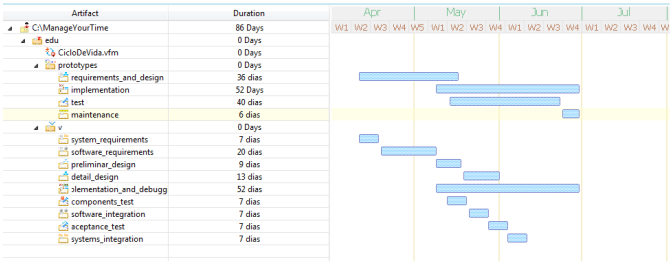


Figura 2.1:

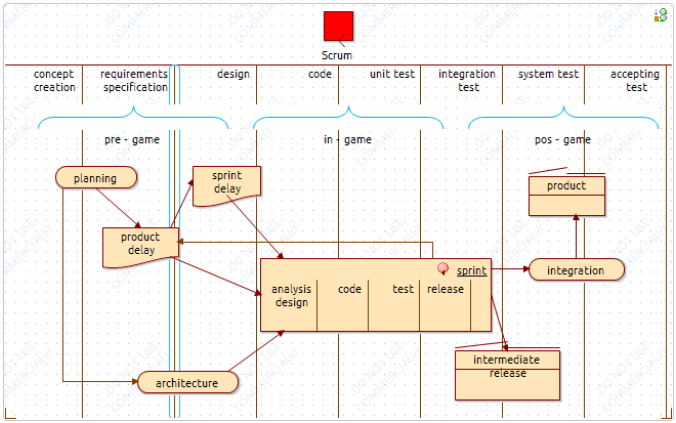
2.2.1. Metodología de implementación

Los criterios que se establecieron al momento de justificar la elección los procesos de software prototipo y V, cuentan con la misma validez para determinar la metodología de implementación, debido a que para esta etapa también es necesario tener un plan de acción que beneficie la gestión de tiempos del proyecto, complemente los procesos de software, cumpliendo un proceder de forma organizada.

2.3. Open Source

Desde que las personas empezaron a desarrollar software, han empezado a indagar en diferentes formas de realizar las cosas, a fin de obtener la solución computacional que solucione su necesidad. Con el tiempo estos pensamientos han devenido en ideologías que orientan la variedad de metodologías disponibles para desarrollar software.

El pensamiento o filosofía que entra en cuestión, es la del software libre, donde uno de sus principios, consiste en la reutilización del conocimiento, en este caso, el código. Es aquí donde entra el Open Source, que se relaciona con el código abierto, y con su revisión por parte de una comunidad de desarrolladores externos. Siguiendo el principio de filosofía libre, se pretende utilizar el concepto O.S con la intención de obtener una ayuda en momentos donde la implementación se torne complicada, llegandose a extrapolar a diversos casos en los que se necesite la apreciación del problema que se está trabajando por parte de un externo el cual ya lo haya desarrollado.


Scrum
concept creation
requirements specification
design
code
unit test
integration test
system test
accepting test
pre - game
in - game
pos - game
planning
sprint delay
product delay
product
sprint
analysis design
code
test
release
integration
intermediate release
architecture

Parte II

DISEÑO

Capítulo 3

Requerimientos

3.1. Introducción

Para cualquier proyecto de software, es un punto fundamental conocer cuál es la necesidad y el problema que el cliente desea resolver. Para tener una visión holística del problema, se hace necesario definir los requerimientos que satisfagan al cliente y resuelvan el problema.

3.2. Requerimientos del Cliente

Se entiende como lo que el cliente espera encontrar cuando interactúe con la aplicación. Bajo la anterior premisa, se definieron los siguientes requerimientos:

1. Añadir una tarea.
2. Añadir subtareas para una tarea.
3. Añadir un horario universitario.
4. Añadir un horario de descanso (dormir).
5. Añadir un horario de transporte.
6. Añadir una tarea a una materia.
7. Mostrar todas las tareas pendientes.
8. Mostrar las tareas pendientes por materia.
9. Mostrar las tareas pendientes por tipo.
10. Mostrar las tareas pendientes para una fecha.
11. Mostrar las tareas pendientes por dificultad.
12. Mostrar el horario general del usuario.
13. Mostrar los horarios asignados para las tareas pendientes.
14. Modificar horario.
15. Modificar tarea.
16. Sugerir horarios para realizar tareas.
17. Sugerir cuánto tiempo podría tomar una tarea.
18. Sugerir tiempos de pausas activas durante la realización de una tarea.
19. Alertar de la próxima entrega de una tarea.
20. Advertir si se debe sacrificar algún espacio de descanso.

3.3. Casos de uso

Los casos de uso describen la interacción del usuario con las diversas funcionalidades planteadas, permitiendo obtener una forma de comunicar los requerimientos de tal forma que sea entendida tanto por usuario como por desarrolladores. Como se podrá observar a continuación, serán cuatro diagramas de caso de uso los que se presentan, donde el motivo por el cual los casos de uso comparten diagrama, es porque se considera que existe cierta relación entre ellos.

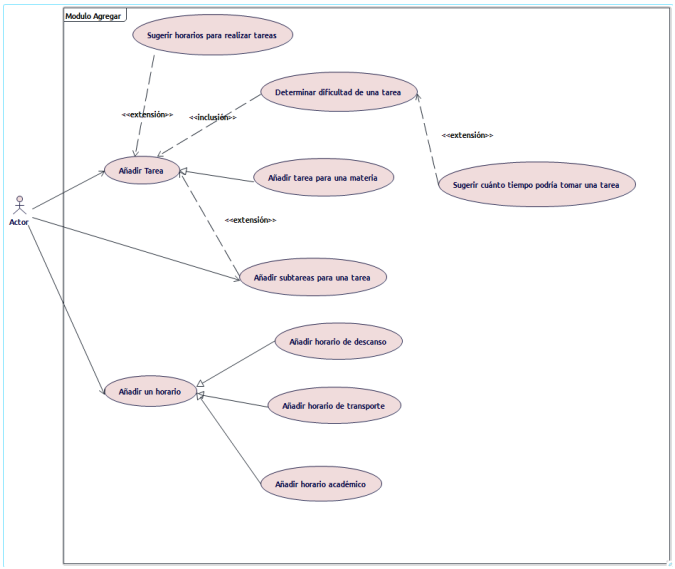


Figura 3.1: Primer diagrama de caso de uso

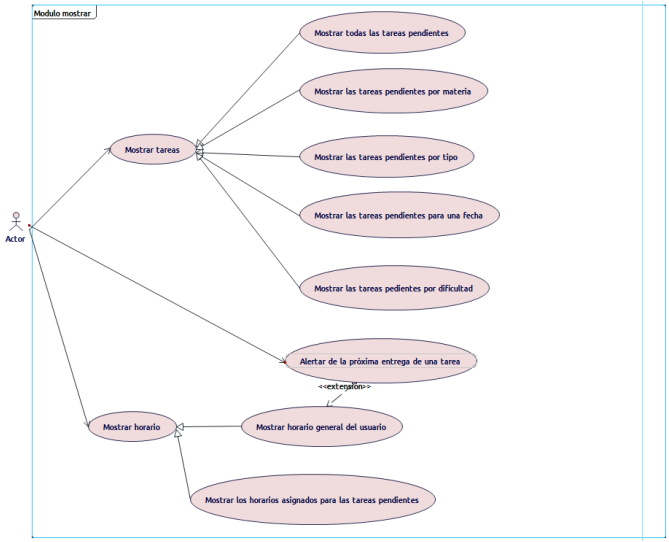


Figura 3.2: Segundo diagrama de caso de uso

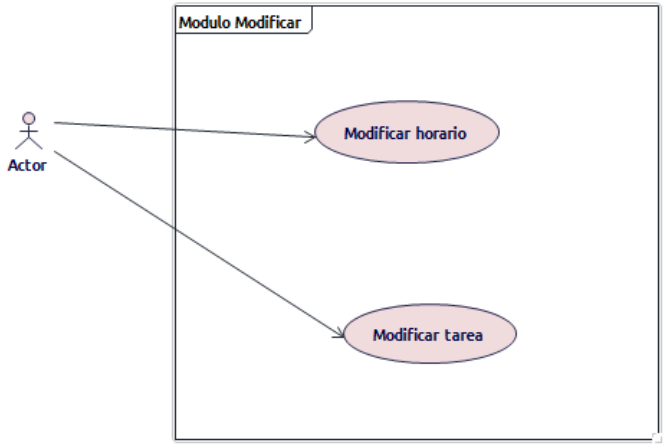


Figura 3.3: Tercero diagrama de caso de uso

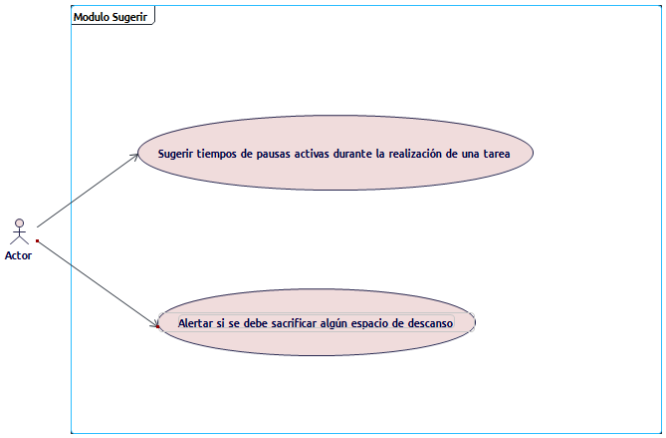


Figura 3.4: Cuarto diagrama de caso de uso

Los diagramas a continuación representan gran importancia complemetando la definición de los requerimientos.

3.4. Diagramas de secuencia

Los diagramas de secuencia permiten obervar la realización del caso de uso, responden el como se va a hacer el requerimiento. Los siguientes son los diagramas de secuencia de 4 casos de uso que se consideran de mayor importancia.

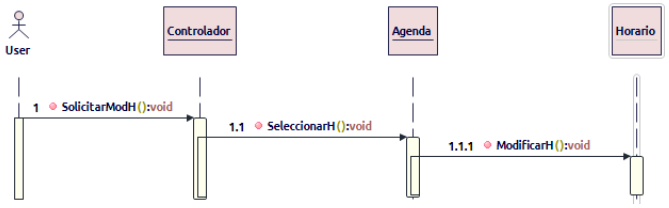


Figura 3.5: Diagrama de secuencia caso de uso 14.

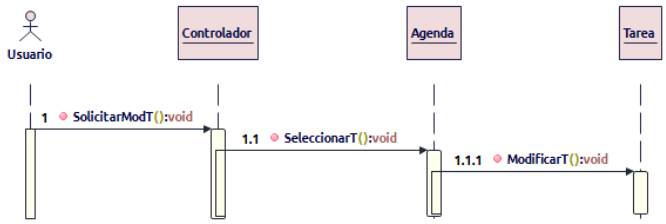


Figura 3.6: Diagrama de secuencia caso de uso 15.

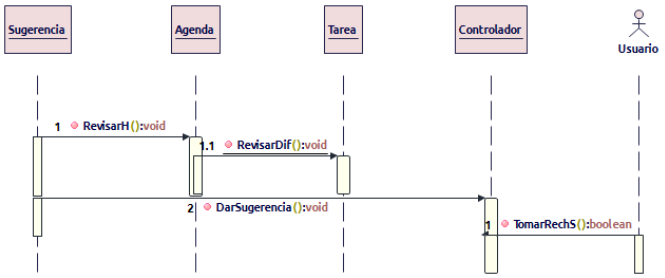


Figura 3.7: Diagrama de secuencia caso de uso 16.

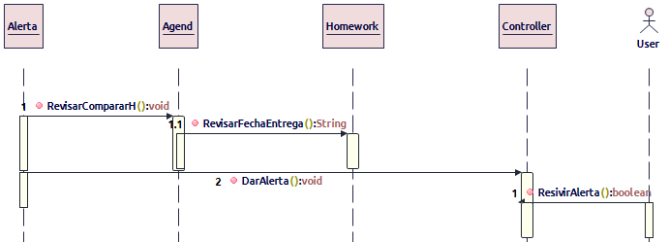


Figura 3.8: Diagrama de secuencia caso de uso 19.

3.5. Diagramas de comunicación

Igualmente relacionados con los diagramas anteriores, principalmente con el diagrama de secuencia. Su función como su nombre lo indica, consiste en detallar en como se comunican los objetos que solucionan el requerimiento.

Se presentan los diagramas correspondientes a los ya expuestos diagramas de secuencia:

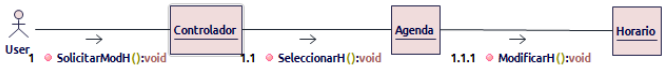


Figura 3.9: Diagrama de comunicación caso de uso 14.

Figura 3.10: Diagrama de comunicación caso de uso 15.

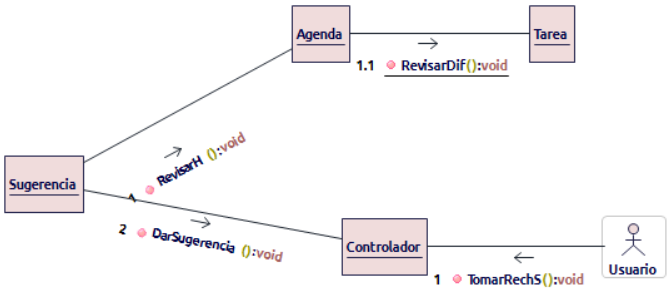


Figura 3.11: Diagrama de comunicación caso de uso 16.

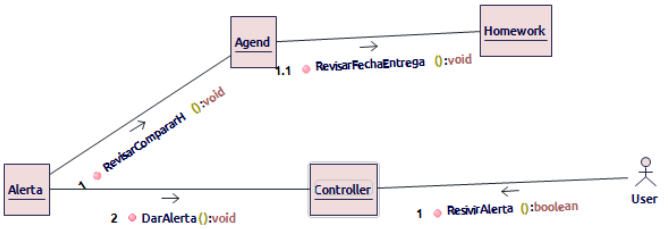


Figura 3.12: Diagrama de comunicación caso de uso 19.

Las siguientes tablas, son la especificación de los requerimientos que se considera que tienen un nivel de importancia alta.

RF-01	Añadir una tarea	
Descripción	El usuario añade una tarea pendiente por desarrollar.	
Precondición	El usuario debe tener un horario	
Secuencia	Paso	Acción
	1	El usuario selecciona la opción de crear tarea.
	2	El usuario proporciona la información requerida (nombre de la tarea, tipo, materia a la que pertenece)
	3	El usuario verifica la información registrada.
	4	El usuario hace selecciona el botón aceptar.
Postcondición	El sistema muestra la tarea recién asignada con sus especificaciones y su recomendación de tiempo de realización y de horario	
Excepciones	Paso	Acción
	4	Se añade una tarea que requiere urgencia (Imprevisto). El usuario elige que horario sacrificará para realizar la tarea.
	4	Se añade una tarea que es imposible de realizar debido al tiempo u horario. Es necesario modificar los tiempos u horarios en los que se realizará la tarea o elegir si sacrificar una frnaja de horario.
Rendimiento	Paso	Cota de tiempo
	1	1 segundo
	2	40 segundos
	3	5 segundos
	4	1 segundo
Importancia	Muy importante	
Urgencia	urgente	

RF-02	Añadir subtareas para una tarea.	
Descripcion	Se añade una subtarea a una tarea.	
Precondición	Debe existir alguna tarea pendiente.	
Secuencia	Paso	Acción
	1	El usuario selecciona la tarea a la que desea añadirle una subtarea.
	2	Seleccionar la opción de añadir subtarea.
	3	Se añade la subtarea como una tarea (RF-01)
	4	El usuario verifica la información.
	5	El pulsa la opción de aceptar.
Postcondición	El sistema añadirá la subtarea a la tarea, mostrará sus especificaciones y recomendación de tiempo de realización y de horario	
Excepciones	Paso	Acción
	1	No existe una tarea para añadirle una subtarea.
	3	La subtarea es de carácter urgente.
	4	Se añade una subtarea y esta hace que la tarea sea imposible de terminar debido al tiempo u horario.
Rendimiento	Paso	Cota de tiempo
	1	5 segundos
	2	1 segundo
	3	40 segundos
	4	5 segundos
	5	1 segundo
Importancia	Importante	
Urgencia	No urgente	
Comentarios	No.	Descripción
	1	Añadir una subtarea es lo mismo que añadir una tarea, la deferencia es que está anidada dentro de una tarea general.

RF-14	Modificar horarios	
Descripcion	Se selecciona y modifica una franja del horario.	
Precondición	El horario que lo que se desea modificar debe estar asignado.	
Secuencia	Paso	Acción
	1	Se selecciona la opción de modificar horario.
	2	Se selecciona el horario de la tarea que se desea cambiar.
	3	Se selecciona la nueva franja de horario en la que se acomodara la tarea.
	4	El cambio de horario se ha realizado.
Postcondición	El horario es modificado y el sistema puede ofrecer sugerencia de tiempo de realización, o incluso si se debe sacrificar algún espacio de descanso.	
Excepciones	Paso	Acción
	1	No hay ningún horario para seleccionar, en este caso el caso de uso acaba.
	3	No existe ninguna franja disponible para cambiar, en este caso el caso de uso acaba.
Rendimiento	Paso	Cota de tiempo
	1	1 segundo
	2	1 segundo
	3	10 segundos
	4	1 segundo
Importancia	Importante	
Urgencia	Hay presión	

RF-15	Modificar Tareas.	
Descripcion	Se permite modificar los diferentes campos de una tarea.	
Precondición	Debe existir alguna tarea para modificar.	
Secuencia	Paso	Acción
	1	Se solicita modificar una tarea.
	2	El usuario selecciona la tarea que desea modificar.
	3	El usuario selecciona el campo de la tarea que desea modificar.
	4	Se modifica el campo de la tarea.
	5	Se permite elegir realizar otro cambio o terminar.
Postcondición	La tarea tiene un campo modificado, según el tipo podría haber una sugerencia, como en el caso de dificultad, o cambio de horario para realizarse.	
Excepciones	Paso	Acción
	3	Si la tarea solo tiene los campos minimos se puede agregar el cambio, así el caso de uso continua.
Rendimiento	Paso	Cota de tiempo
	1	1 segundo
	2	1 segundo
	3	1 segundo
	4	5 segundos
	5	1 segundo
Importancia	vital	
Urgencia	Hay presión	

RF-16	Sugerir horarios para realizar tareas.	
Descripcion	Según los horarios disponibles, al momentos de adicionar una tarea se hace una sugerencia de horario para realizarla.	
Precondición	Se debe haber agregado una tarea, y deben haber horarios disponibles para hacer la recomendación.	
Secuencia	Paso	Acción
	1	Al momento de agregar una tarea, se revisan horarios disponibles.
	2	Se revisan las variables de la tarea (dificultad, fecha de entrega).
	3	Se hace la recomendación.
Postcondición	La recomendación se dará al usuario dandole la posibilidad de tomarla o dejarla.	
Excepciones	Paso	Acción
	1	Puede pasar que no haya horarios disponibles para hacer la recomendación, así el caso de uso termina.
	2	Si no hay variables de tarea definidos, se pasa al siguiente paso.
Rendimiento	Paso	Cota de tiempo
	1	5 segundos
	2	5 segundos
	2	1 segundo
Importancia	Normal	
Urgencia	Puede esperar	
Comentarios	No.	Descripción
	1	El caso de uso esta bastante ligado a otros casos de uso, como puede apreciarse en el diagrama, sin embargo su relevancia no es la misma como la de los casos a los que apoya.

RF-19	Alertar de la próxima entrega de una tarea.	
Descripcion	Se pretende avisar con tiempo prudencial que el ciclo de una tarea esta por finalizar, lo cual significa que debe ser terminada para ser entregada.	
Precondición	La existencia de la tarea.	
Secuencia	Paso	Acción
	1	Se revisa el tiempo para que la tarea deba ser terminada comparandolo con el prudencial.
	2	Se realiza el aviso.
Postcondición	El usuario será avisado sobre la proximidad de su tarea.	
Excepciones	Paso	Acción
	1	Si la tarea no cuenta con tiempo de realización se considera indefinida, así el caso de uso termina.
Rendimiento	Paso	Cota de tiempo
	1	1 segundo
	2	1 segundo
Importancia	Importante	
Urgencia	Puede esperar	

Capítulo 4

Interacción

4.1. Introducción

cntenido...

Capítulo 5

Clases

5.1. Introducción

cntenido...

Capítulo 6

Patrones

6.1. Introducción

cntenido...

6.2. Prototipo

Capítulo 7

Estados

7.1. Introducción

cntenido...

Capítulo 8

Componentes

8.1. Introducción

cntenido...

Capítulo 9

Nodos

9.1. Introducción

cntenido...

Capítulo 10

Actividades

10.1. Introducción

cntenido...

Parte III

REFLEXIONES

k

Capítulo 11

Conclusiones

11.1. Introducción

cntenido...

Bibliografía