

Universidad ORT Uruguay

Facultad de Ingeniería

Ingeniería de Software 2

Obligatorio 1

Joselen Cecilia (233552)

Belén Martinez (223694)

Camila Burgueño (221951)

Entregado como requisito de la materia Ingeniería de
Software 2

7 de diciembre de 2020

<https://www.overleaf.com/read/mnjzrcmgdgcj>

Declaraciones de autoría

Nosotras, Camila Burgueño, Joselen Cecilia y Belén Martínez, declaramos que el trabajo que se presenta en esa obra es de nuestra propia mano. Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos Ingeniería de Software 2 ;
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad;
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra;
- En la obra, hemos acusado recibo de las ayudas recibidas;
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros;
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.

Resumen

Para la materia Ingeniería de Software 2, obligatorio 2, se implementa el plan de mantenimiento especificado y elaborado en el obligatorio 1.

Índice general

1. Presentación del plan de mantenimiento	3
1.1. Proyecto de mantenimiento	3
1.2. Cronograma	3
2. Gestión de riesgos	4
2.1. Plan de gestión de riesgos	4
2.2. Plan de contingencia	5
3. Implementación de los cambios al software y documentación asociada	7
3.1. Implementación de cambios	7
4. Informe de avance	9
4.1. Logros obtenidos para este período de informe	9
4.2. Logros planificados, pero no completados en este período de informe .	10
4.3. Causa raíz de las variaciones	10
4.4. Impacto de la variación en los próximos hitos o en la fecha de finalización del proyecto.	10
4.5. Acciones correctivas o preventivas planificadas.	11
4.6. Fondos gastados (presupuesto ejecutado) en este período de informe.	12
4.7. Logros planificados para el próximo período de presentación de informes.	12
4.8. Costos planificados para el próximo período de presentación de informes.	13
4.9. Situación del presupuesto del proyecto.	13
4.10. Situación de los riesgos identificados	13
4.11. Evaluación de la calidad de los productos	13
5. Implementación de acciones correctivas	15
5.1. Quality control	16
5.1.1. Cobertura pruebas unitarias	16
5.1.2. CheckStyle	18
5.1.3. SonarQube	18
5.1.4. Líneas de código	19
5.1.5. Documentación	20
6. Analizar el estado de calidad final del software y productos asociados.	21

7. Informe final	23
7.1. Logros obtenidos para este período de informe	23
7.2. Logros planificados, pero no completados en este período de informe.	23
7.3. Causa raíz de las variaciones	24
7.4. Impacto de la variación en los próximos hitos o en la fecha de finalización del proyecto.	24
7.5. Acciones correctivas o preventivas planificadas.	25
7.6. Fondos gastados (presupuesto ejecutado) en este período de informe.	25
7.7. Situación del presupuesto del proyecto.	25
7.8. Situación de los riesgos identificados	25
7.9. Evaluación de la calidad de los productos	26
8. Reflexiones del equipo sobre las fortalezas y debilidades del trabajo realizado	28
Bibliografía	29

1. Presentación del plan de mantenimiento

1.1. Proyecto de mantenimiento

El plan de mantenimiento para el proyecto Alimentación Saludable consiste en mejorar la interfaz de la aplicación aumentando su usabilidad, mejorar la calidad del código para disminuir los defectos encontrados y arreglar la documentación para que se adapte al proyecto nuevo y cumpla con el estándar 302.

Por lo tanto, el mantenimiento de este software consistirá en mejorar la funcionalidad, usabilidad e interfaz de la aplicación.

1.2. Cronograma

El plan se desarrollara se entre el 16/10/2020 hasta el 7/12/2020

2. Gestión de riesgos

2.1. Plan de gestión de riesgos

Durante el desarrollo del proyecto pueden surgir problemas ya sea con los recursos, las personas o el tiempo planificado.

Estos problemas deben ser considerados antes de comenzar con el trabajo para así desarrollar para cada uno estrategias de respuestas para evitar desviaciones en el cronograma y/o en el presupuesto del proyecto

ID	Descripción	Probabilidad de ocurrencia	Impacto	Magnitud	Prioridad	Objetivos afectados	Estrategia de Respuesta	Responsable
R01	Al armar el cronograma de trabajo no se estimó correctamente el tiempo de cada tarea, lo cual puede provocar retrasos en los plazos de entrega.	1	3	3	Alta	Lograr todos los cambios propuestos en el tiempo esperado.	Revisar el cronograma diariamente para no sobrepasar el tiempo, e identificar atrasas rápidamente, destinado a cada tarea, lo que haría que las demás se atrasen. Agregar más horas de trabajo para no posponer las demás tareas.	Belén Martínez
R02	Perder recursos por asignación a otros proyectos. Puede afectar el alcance, tiempo, costo y calidad del proyecto.	0.4	2	1	Media	Lograr todos los cambios propuestos en el tiempo esperado.	Revisar el cronograma diariamente y controlar la asignación de los recursos, cambiando la distribución de estos en caso de ser necesario.	Joselen Cecilia
R03	Perder uno de los equipos de trabajo por daños. Esto puede afectar el tiempo, aumentar el costo y causar retrabajo por la pérdida de parte del proyecto.	0.2	1	0.6	Baja	Lograr todos los cambios propuestos en el tiempo esperado.	Comprar una nueva computadora, generando más gastos.	Camila Burgueño

R04	Especificar incorrectamente los requerimientos. Lo que puede llevar a que el cliente este descontento porque el producto no es el deseado. A medida que se va avanzando en el proyecto aumenta el retrabajo.	0.6	3	1.5	Alta	Se proponen/ implementan cambios que no son necesarios.	Generar mecanismos que permitan redefinir los requerimientos y modificar el plan de proyecto.	Joselen Cecilia
R05	Alguna de las desarrolladoras no conozca el lenguaje o el entorno de trabajo que se venía utilizando.	0.4	1	0.4	Media	Lograr todos los cambios propuestos en el tiempo esperado.	Enseñarle a la desarrolladora la tecnología utilizada en corto plazo.	Camila Burgueño
R06	No se aplican estándares de codificación, lo que junto con diferencias de concepto de calidad puede llevar a inconsistencias en el trabajo.	0.2	3	0.6	Alta	Mejorar la calidad del código.	Definir conceptos claros de calidad y estándares de codificación.	Belén Martínez

R07	Se abarcan más tareas de las que se pueden hacer en el tiempo establecido, se subestimaron las tareas. Esto puede afectar la calidad del producto.	1	3	3	Alta	Lograr todos los cambios propuestos en el tiempo esperado. Mejorar la calidad el código.	Reducir la cantidad de cambios a implementar, sacando aquellos de baja prioridad.	Camila Burgueño
R08	Se generan objetivos que no satisfacen las necesidades del cliente.	0.2	2	0.6	Alta	Mejorar la calidad del código e implementar las solicitudes del cliente.	Generar mecanismos que permitan modificar plan de proyecto y objetivos. Hablar con el cliente.	Joselen Cecilia
R09	Mala estimaciones del costo del proyecto	1	3	3	Alta	Tareas con menos importancia.	Hablar con el cliente para aumentar presupuesto mostrando claramente las causas. Reducir las tareas, suprimir aquellas con poca importancia.	Joselen Cecilia
R10	Enfermedades que impidan por un tiempo que una de las desarrolladoras prosiga con el trabajo	0,6	3	1,5	Alta	Lograr todos los cambios propuestos en el tiempo esperado.	Dedicar más horas. Hablar con el cliente para pedir más desarrolladores	Dependiendo el caso.

2.2. Plan de contingencia

Junto con cada riesgo se propuso (en la misma tabla, columna estrategia de respuesta) un estrategia de respuesta, es decir un solución acorde a cada riesgo para

así poder tratarlo inmediatamente ocurra.

3. Implementación de los cambios al software y documentación asociada

3.1. Implementación de cambios

En la siguiente tabla se muestra un reporte de las tareas que se especificaron en el primer obligatorio indicando si se implementaron o no en esta instancia, el tiempo y esfuerzo que llevaron y los recursos requeridos para cada una.

El presupuesto planificado del proyecto era de 1578USD y hoy en día es de 2693USD, esto se debe a que se trabajó más horas de lo planificado, para corregir el aplazamiento en el cronograma, y así poder llegar al objetivo final en fecha. En la tabla adjuntada debajo se detalla en qué se gastó este presupuesto.

				PLANIFICADO			EJECUTADO			DIFERENCIA	
EDT	ACTIVIDAD	NOMBRE DE LA ACTIVIDAD	RECURSOS	ESFUERZO EN HORAS	IMPLEMENTACION	COSTO	ESFUERZO EN HORAS	IMPLEMENTACION	COSTO	DESVIACION COSTO	DESVIACION ESFUERZO
1.1	A01	Bloques de Código duplicado	1DEV 1TS	12	17/11	152	6	4/11	76	-76	-6
1.1	A02	Reducir los bugs reportados por findBugs	1DEV 1TS	6	11/11	92	6	19/11	92	0	0
1.1	A03	Mejorar cobertura de pruebas unitarias	1DEV 1TS	15	11/20	198	20	5/11	288	+90	+5
1.2	A04	Interfaz de usuario cumpla al menos 5 heurísticas	2DEV 1TS	22	9/11	332	24	20/11	368	+36	+2
1.3	A05	Contraseña alfanumérica De al menos 8 caracteres	1 Dev 1TS	4	16/10	64	8	22/11	128	+64	+4
1.4	A06	El nombre de usuario se debe mostrar en cada ventana del sistema	1 Dev 1TS	6	22/10	92	6	1/11	92	0	0
1.5	A07	Implementar las solicitudes de cambio de 10 al 15	2 Dev 1TS	33	29/10	498	20	3/11	26	-110	-13
2.1	A08	Definir lo estándares de calidad en los que nos basaremos	PM	2	15/10	40	2	18/10	40	0	0

2.1	A09	Definir los criterios de calidad en los que nos basaremos	PM	2	15/10	40	2	18/10	40	0	0
2.2	A10	Analizar código con analizadores estáticos de código	TS	1	23/11	10	8	4/12	80	+70	+7
2.3	A15	Documentar los defectos y posibles mejoras.	TS	2	30/11	20	2	5/12	20	0	0
2.4	A13	Generar casos de prueba	2TS	8	4/12	80	8	4/12	80	0	0
2.5	A16	Definir los requerimientos funcionales	PM	4	15/10	80	2	27/11	40	-40	+2
2.5	A17	Definir los requerimientos no funcionales	PM	2	15/10	40	2	27/11	40	0	0
2.6	A14	Definir las sesiones de ejecución de pruebas	TS	4	7/12	40	4	4/12	40	0	0
2.7	A12	Correr pruebas unitarias y documentar resultado	TS	2	27/11	20	2	30/11	20	0	0
2.8	A11	Documentar el análisis del código.	TS	4	2/12	40	5	5/12	50	+10	+1

Además de estas las tareas especificadas anteriormente se arreglo la documentación del producto ya que había cosas que cambiar y mejorar.

4. Informe de avance

Informe de estado del proyecto analiza el cronograma y presupuesto para el período de informe y proporciona información para el próximo período.

Periodo del informe 16/10/2020 - 9/11/2020

Equipo: Belén Martínez, Joselen Cecilia, Camila Burgueño

4.1. Logros obtenidos para este período de informe

ID	Actividad	Entregable	Fecha estimada	Fecha realización	Esfuerzo estimado	Esfuerzo en horas
1.1	A01: Bloques de código duplicados. Y Mejora de nombres de variables y métodos.	Reporte de defectos y estado del código.	17/11/2020	4/11/2020	12 horas/2 personas	12 horas
1.1	A03: Mejorar cobertura de pruebas unitarias.	Reporte de cobertura de las pruebas unitarias. Se completa en un 80% ya que no se ha terminado de implementar el resto de los cambios que involucran código.	19/11/2020	5/11/2020	15 horas/2 personas	20 horas
1.2	A04	Rediseñar la interfaz de usuario. Se comienza pero aun no se termina.	9/11/2020	6/11/2020	22 horas/ 3 personas	24 horas
1.3	A05: Implementación de la autenticación del administrador mediante una contraseña.	Interfaz con nueva funcionalidad	16/10/2020	2/11/2020	4 horas/2 personas	4 horas
1.4	A06: Mostrar en todo momento cual usuario es el que está utilizando el sistema.	Interfaz con nueva funcionalidad	16/10/2020	1/11/2020	6 horas/ 2 personas	6 horas

1.5	A07: Se implementan los cambios del 10 al 15	Interfaz con nueva funcionalidad	21/10/2020	3/11/2020	33 horas/ 3 personas	26 horas
-----	--	----------------------------------	------------	-----------	----------------------	----------

Como se ve en la comparación de la fecha estimada con la fecha de realización hay cambios que se terminan antes de la fecha esperada y otros que se atrasan esto se debe a cambios en la organización, porque creíamos que algunos cambios eran más primordiales que otros.

4.2. Logros planificados, pero no completados en este período de informe

Se esperaba tener terminado el rediseño de la interfaz, para mejorar la estética y usabilidad de esta para la fecha del 9/11/2020, pero a pesar de que se ha adelantado en un 50 % este rediseño aun no se ha terminado, por lo cual estamos atrasadas con este entregable.

Este atraso lleva a que aun no se pueda implementar en un cien por ciento el cambio de las pruebas unitarias, para mejorar la cobertura de estas.

4.3. Causa raíz de las variaciones

Las tareas anteriores al rediseño de la interfaz llevaron más tiempo del planificado, lo que junto con el hecho de no poder dedicarle tiempo extra hizo que no pudiéramos cumplir en el tiempo esperado este entregable.

Además surgieron dos factores que hicieron que este cambio, que está en proceso, nos ocupe más tiempo del planificado;

En un principio se nos dificulto entender como funcionaban las ventanas ya que el código no era de muy buena calidad, lo que nos llevo a un doble trabajo, primero entender el código y luego rehacerlo. El segundo punto que llevo a este problema fue que a medida que usábamos la interfaz, y nos fijábamos más en los detalles, surgían mas puntos que arreglar.

Como todo el tiempo se esta agregando código nuevo para la interfaz aun no se puede terminar de implementar todas las pruebas unitarias, por lo que no se llega al porcentaje de cobertura esperado.

4.4. Impacto de la variación en los próximos hitos o en la fecha de finalización del proyecto.

Para este punto vamos a analizar los valores de SPI y SV

SPI:Índice de desempeño del cronograma.
 $SPI=EV/PV$

SV:Determina en qué medida el proyecto está adelantado o retrasado en relación con la fecha de entrega, en un momento determinado.
 $SV=EV-PV$
Informe de estado del proyecto

EV:Es la medida del trabajo realizado en términos de presupuesto autorizado para dicho trabajo.
 $EV = \% \text{ Avance real } * BAC$
 $EV = 0.4 * 1578 \text{USD} = 631$

PV:Valor planificado
22 días- 655USD
53 días -1578USD
 $PV = 655 \text{USD}$

Resultados

$SV=631-655=-24$ menor a 0 lo que indica por definición que el proyecto viene atrasado con respecto al cronograma.
 $SPI=631/655=0,96$ menor 1 lo que indica por definición que estamos por encima de los costos planificados

4.5. Acciones correctivas o preventivas planificadas.

Cuando hicimos el plan de riesgos, tuvimos en cuenta que por distintas razones podían surgir retrasos en el tiempo del proyecto y definimos ciertas acciones para corregir estos problemas estas fueron:

- 1.Revisar el cronograma diariamente para no sobrepasar el tiempo destinado a cada tarea y en cada de que suceda identificarlo a tiempo.
- 2.Agregar más horas de trabajo para no aplazar a las tareas que le siguen.

En caso de que estas medidas no sean suficientes, se va a dejar de lado la implementación de tareas con prioridad baja. Siempre priorizando las solicitudes explícitas del cliente.

También tuvimos en cuenta el hecho de haber planificado mal el costo del proyecto y definimos ciertas acciones para corregir estos problemas estas fueron:

- 1.Hablar con el cliente para aumentar presupuesto mostrando claramente las causas.
- 2.Reducir las tareas, suprimir aquellas con poca importancia.

4.6. Fondos gastados (presupuesto ejecutado) en este período de informe.

Como vimos en el calculo del SPI en el punto 4, el presupuesto ejecutado en este periodo sobrepasa el panificado para este punto, vamos gastando un 46 % del costo en un 40 % de trabajo, por lo que nos lleva a pensar que el proyecto va a aumentar su costo total al terminar el proyecto.

EAC: Para calcular el costo total con los datos hasta el momento vamos a hacer la estimación del costo total previsto de completar todo el trabajo

$$EAC = BAC / CPI$$

$$EAC= BAC/ (EV/AC)$$

$$EAC= 1578/(631/655)$$

$$EAC=1578/0.96$$

Nuevo costo total previsto al completar todo el trabajo del proyecto 1644USD

4.7. Logros planificados para el próximo período de presentación de informes.

-Terminar de rediseñar la interfaz, y junto con eso terminar de implementar las pruebas unitarias.

-Analizar el código para corroborar que los cambios respectos a este hayan mejorado la calidad del mismo.

-Hacer nuevos casos de uso, probando las nuevas funcionalidades, y verificando los cambios.

-Documentar los cambios, y mejoras.

4.8. Costos planificados para el próximo período de presentación de informes.

El costo planificado para el próximo periodo de presentación va a ser (contemplando el nuevo presupuesto, calculado en el punto 6, con el aumento por el retraso=1644USD) de 986USD ya que se espera hacer un 60 % mas del trabajo, completar todas las tareas.

4.9. Situación del presupuesto del proyecto.

Como mostramos en el punto 4 el presupuesto total del trabajo aumento en un 6 % siendo ahora de 1644USD.

4.10. Situación de los riesgos identificados

Cuando hicimos el plan de riesgos, como dijimos anteriormente, identificamos dos de los riesgos que están sucediendo en este momento:

R01: al armar el cronograma de trabajo no se estimo correctamente el tiempo de cada tarea, lo cual puede provocar retraso en los plazos de entrega.

R09: mala estimaciones del costo del proyecto.

Como habíamos previsto estos tenían alta probabilidad de ocurrencia y también alto impacto ya que afectan dos de los puntos principales al planificar el proyecto, el tiempo y el costo del mismo, esto puede generar problemas con el cliente, y baja de calidad del proyecto.

Pero para que eso no sucede definimos junto con los riesgos acciones correctivas, respuestas a estos problemas con la esperanza de poder arreglarlos, sin afectar al cliente.

Los riesgos identificados se encuentran en la sección 1.plan de riesgos

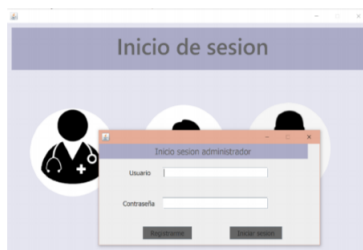
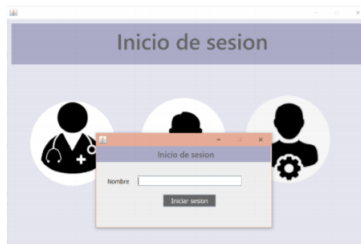
4.11. Evaluación de la calidad de los productos

Aun no se hace una nueva evaluación con los analizadores estáticos de código, ya que no se terminan de implementar los cambios en este, pero viendo el código notamos que este mejoro en gran medida, se redujeron los bloques de código duplicados que fueron posibles y se mejorar los métodos y nombres de estos facilitando el entendimiento.

Además creemos que la interfaz tenia una usabilidad muy mala, que costaba entender y este punto aunque aun no se hace el análisis heurístico notamos que mejoro

en un porcentaje muy importante.

Adjuntamos muestra de los cambios en el panel principal de la interfaz:



5. Implementación de acciones correctivas

Teniendo en cuenta el retraso en la planificación, se decide tomar ciertas acciones correctivas para lograr realizar todas las tareas para la fecha de entrega.

Se agregan mas horas de trabajo por dia para las actividades que lo requieren, y se vuelve a planificar el cronograma GANT, para registrar las fechas de las actividades y así asegurarnos de llegar a la entrega con todo implementado.

Corrección del cronograma, linea base:

		Name	Duration	Start	Finish	Predecessors	Resources	C
1		☐ IMPLEMENTACION DE CAMBIOS	24.5days?	10/16/2020	11/19/2020			
2		☐ Rehacer	17.days?	1/11/2020	20/11/2020			
3		☐ Corregir código ya existente para mejorar calidad d	4.days?	1/11/2020	19/11/2020			
4		Bloques de código duplicados	2.days?	1/11/2020	4/11/2020		Camila Burgueño,	
5		Mejorar código para reducir defectos por FindBugs	1.days?	18/11/2020	19/11/2020		Belén Martínez,Jo	
6		Mejorar cobertura de pruebas unitarias	1.days?	4/11/2020	5/11/2020		Camila Burgueño,	
7		☐ Rediseñar interfaz de usuario, mejorar usabilidad y	2.days?	17/11/2020	20/11/2020			
8		Rediseñar interfaz de usuario para que cumpla al men	2.days?	17/11/2020	20/11/2020		Camila Burgueño,	
9		☐ Nuevas funcionalidades	11.days?	27/10/2020	22/11/2020			
10		Implementación de autenticación mediante contraseña	1.days?	21/11/2020	22/11/2020		Joselen Cecilia,Be	
11		Informar sobre el usuario autenticado en la aplicación en c	3.days?	28/10/2020	1/11/2020		Belén Martínez,Ce	
12		Implementar solicitudes de cambio desde la 10 a la 15	7.days?	27/10/2020	3/11/2020		Camila Burgueño,	
13		☐ DOCUMENTACION	18.days?	16/11/2020	5/12/2020			
14		☐ Análisis de calidad	1.days?	19/11/2020	20/11/2020			
15		Definición de criterios y estándares	1.days?	19/11/2020	18/10/2020		Belén Martínez	
16		☐ Análisis de código	14.days?	19/11/2020	5/12/2020			
17		Analizar código con analizadores estáticos de código	2.days?	1/12/2020	4/12/2020		Joselen Cecilia	
18		Documentar el resultado de los análisis	1.days?	3/12/2020	5/12/2020		Camila Burgueño	
19		Correr pruebas unitarias y documentar resultado	1.days?	19/11/2020	20/11/2020		Belén Martínez	
20		☐ Alcance	2.days?	3/12/2020	5/12/2020			
21		☐ Reporte de defectos	1.days?	4/12/2020	5/12/2020			
22		Reportar los defectos y posibles mejoras del sistema	1.days?	4/12/2020	5/12/2020		Joselen Cecilia	
23		☐ Casos de uso funcionales	1.days?	3/12/2020	4/12/2020			
24		Generar casos de prueba	1.days?	3/12/2020	4/12/2020		Joselen Cecilia,Be	
25		☐ Pruebas	1.days?	3/12/2020	4/12/2020			
26		☐ Pruebas funcionales	1.days?	3/12/2020	4/12/2020			
27		Definir sesiones de ejecución de pruebas	1.days?	3/12/2020	4/12/2020		Joselen Cecilia	
28		☐ Especificación	10.days?	16/11/2020	27/11/2020			
29		☐ Especificación de requerimientos	10.days?	16/11/2020	27/11/2020			
30		Definición de requerimientos funcionales	1.days?	26/11/2020	27/11/2020		Belén Martínez	
31		Definición de requerimientos no funcionales	1.days?	26/11/2020	27/11/2020		Belén Martínez	

Se modifica también el plan de RRHH:

Joselen Cecilia, 5 horas diarias para el rol desarrollador, y 3 horas diarias para el rol tester. Esto figura en el cronograma como recurso con el nombre Joselen Cecilia.

Belén Martinez, 4 horas como tester, 2 horas como desarrollador, y 2 horas como PM. Esto figura en el cronograma como recurso con el nombre Belén Martinez.

Camila Burgueño, 4 horas como tester, 3 horas como desarrollador, 1 hora como PM. Esto figura en el cronograma como recurso con el nombre Camila Burgueño.

Todos los integrantes del equipo trabajan un promedio de 8 horas diarias es decir se aumentan dos horas diarias a cada integrante, en este ultimo periodo de trabajo (9/11/2020 - 6/12/2020), para complementar las tareas que llevaron mas tiempo del planificado, y los atrasos en el cronograma.

Mas recursos

También se ejecuto la acción correctiva de agregar mas desarrolladores a las actividades que eran mas difíciles de lo estimado, para así reducir tiempo con mas recursos.

5.1. Quality control

5.1.1. Cobertura pruebas unitarias

Se muestran los resultados finales de Jacoco para ver el analisis en detalle y la justificacion de estos resultados ir a la documentacion del producto que se adjunta en la misma carpeta.

En el primer obligatorio planteamos como uno de los objetivos mejorar la cobertura de las pruebas unitarias aumentándola a un porcentaje del 95 % en el paquete dominio del proyecto.

Ya que como habíamos visto el porcentaje era de 79 % ,mucho de estos test no pasaban y algunas clases estaban con cobertura 0 % es decir no habían sido probadas

JaCoCoverage analysis of project "AlimentacionSaludable" (powered by JaCoCo from EcEmma)

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Cl
interfaz	<div><div></div></div>	0%	<div><div></div></div>	0%	653	653	3.555	3.555	480	480	157	
dominio	<div><div></div></div>	79%	<div><div></div></div>	68%	91	186	151	424	77	164	2	
alimentacionsaludable	<div><div></div></div>	0%	<div><div></div></div>	n/a	2	2	10	10	2	2	1	
Total	20.183 of 22.342	10%	360 of 390	8%	746	841	3.716	3.989	559	646	160	

Created with JaCoCo 0.7.6.201602180812

dominio

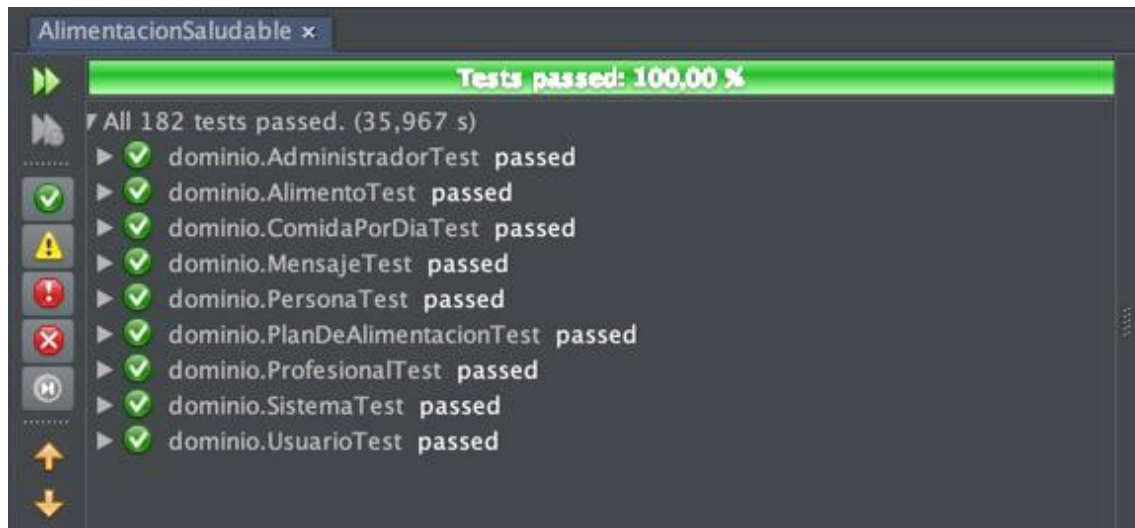
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	C
Usuario		70%		67%	16	41	18	96	12	35	0	
Sistema		63%		80%	11	25	48	108	9	20	0	
Mensaje		0%		n/a	10	10	20	20	10	10	1	
PlanDeAlimentacion		42%		n/a	16	17	24	34	16	17	0	
Administrador		0%		n/a	6	6	11	11	6	6	1	
Alimento		81%		50%	7	16	15	39	6	15	0	
ComidaPorDia		62%		50%	9	13	11	27	4	8	0	
Profesional.Pais		97%		n/a	2	4	0	5	2	4	0	
Alimento.TipoAlimento		94%		n/a	2	4	0	4	2	4	0	
Alimento.Nutrientes		91%		n/a	2	4	0	3	2	4	0	
Persona		89%		100%	3	15	3	32	3	12	0	
Usuario.Nacionalidades		98%		n/a	1	4	0	6	1	4	0	
Usuario.Restricciones		93%		n/a	1	4	0	2	1	4	0	
Usuario.Preferencias		93%		n/a	1	4	0	2	1	4	0	
Sistema.tipoUsuario		90%		n/a	1	4	0	2	1	4	0	
Profesional		99%		50%	3	15	1	33	1	13	0	
Total	567 of 2.726	79%	14 of 44	68%	91	186	151	424	77	164	2	

Ahora con los nuevos test y arreglando aquellos que estaban implementados pero no pasaban , logramos tener una cobertura de 95 , con todos los test funcionando Esto implica que todas las clases menos sistema están por encima del 90 % de cobertura, incluso tenemos algunas con cobertura del 100 % con todos los test funcionando y con sus principales funcionalidades probadas.

JaCoCoverage analysis of project "AlimentacionSaludable" (powered by JaCoCo from EcEmma)

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Cla
interfaz		0%		0%	656	656	3.674	3.674	476	476	156	
dominio		95%		77%	30	210	35	469	14	175	0	
alimentacionsaludable		0%		n/a	2	2	10	10	2	2	1	
Total	20.457 of 23.214	12%	376 of 430	13%	688	868	3.719	4.153	492	653	157	





JaCoCoverage analysis of project "AlimentacionSaludable" (powered by JaCoCo from EcEmma) > dominio [Source Files](#) [Sessions](#)

dominio

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed
Sistema	<div><div></div></div>	79%	<div><div></div></div>	81%	7	41	30	137	2	28	0
Profesional.Pais	<div><div></div></div>	97%		n/a	2	4	0	5	2	4	0
Alimento.TipoAlimento	<div><div></div></div>	94%		n/a	2	4	0	4	2	4	0
Alimento.Nutrientes	<div><div></div></div>	91%		n/a	2	4	0	3	2	4	0
Alimento	<div><div></div></div>	96%	<div><div></div></div>	100%	2	16	3	39	2	15	0
Usuario.Nacionalidades	<div><div></div></div>	98%		n/a	1	4	0	6	1	4	0
ComidaPorDia	<div><div></div></div>	94%	<div><div></div></div>	50%	5	13	2	27	0	8	0
Usuario.Restricciones	<div><div></div></div>	93%		n/a	1	4	0	2	1	4	0
Usuario.Preferencias	<div><div></div></div>	93%		n/a	1	4	0	2	1	4	0
Sistema.tipoUsuario	<div><div></div></div>	90%		n/a	1	4	0	2	1	4	0
Usuario	<div><div></div></div>	100%	<div><div></div></div>	67%	4	41	0	96	0	35	0
Profesional	<div><div></div></div>	100%	<div><div></div></div>	50%	2	15	0	33	0	13	0
PlanDeAlimentacion	<div><div></div></div>	100%		n/a	0	18	0	36	0	18	0
Persona	<div><div></div></div>	100%	<div><div></div></div>	100%	0	15	0	32	0	12	0
Administrador	<div><div></div></div>	100%	<div><div></div></div>	100%	0	13	0	25	0	8	0
Mensaje	<div><div></div></div>	100%		n/a	0	10	0	20	0	10	0
Total	160 of 2.917	95%	16 of 70	77%	30	210	35	469	14	175	0

Created with JaCoCo 0.7.6.201602180812

5.1.2. CheckStyle

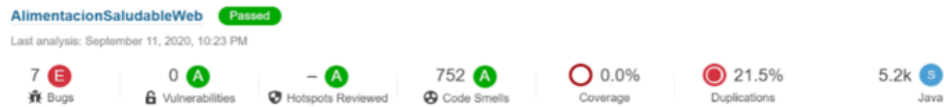
Se arreglaron los errores de indentación, espacios y nombres de variables, además se agregaron los @override que faltaban, entre otros detalles.

Esto hizo que los errores marcados por CheckStyle se redujeran al 0 % en los paquetes dominio, alimentacionSaludable y test y al 20 % en el paquete interfaz.

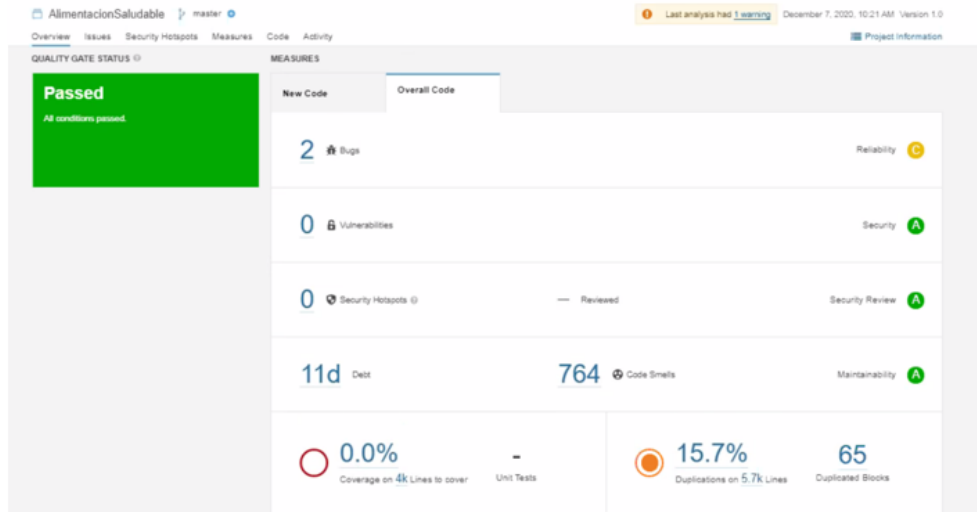
5.1.3. SonarQube

Se muestran los resultados finales de sonarQube para ver el analisis en detalle y la justificacion de estos resultados ir a la documentacion del producto que se adjunta en la misma carpeta.

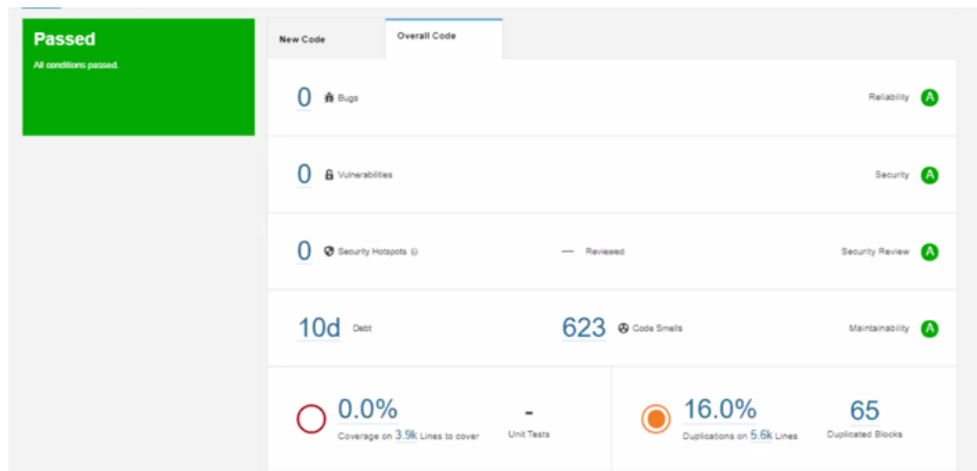
16/10/2020



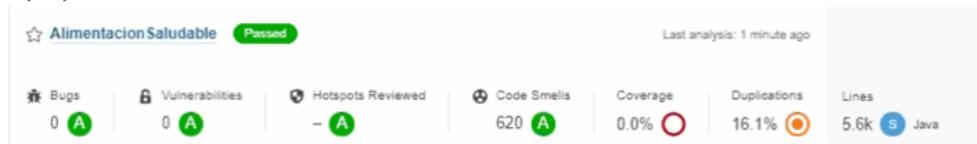
23/11/2020



29/11/2020



6/12/2020



5.1.4. Lineas de código

Al comienzo del obligatorio contábamos con 5181 lineas de código, en este momento tenemos 5582 este aumento se debe a las nuevas funcionalidades implementadas, en un principio habíamos pensado reducir esta cantidad pero luego nos dimos

cuneta que la mayoría de estas lineas son código auto generado.

	Lines of Code	Bugs	Vulnerabilities	Code Smells	Security Hotspots	Coverage	Duplications
src	5,582	0	0	623	0	0.0%	16.0%
alimentacionsaludable	17	0	0	0	0	0.0%	0.0%
dominio	498	0	0	25	0	0.0%	0.0%
interfaz	5,067	0	0	598	0	0.0%	19.2%

	Lines of Code	Bugs	Vulnerabilities	Code Smells	Security Hotspots	Coverage	Duplications
AlimentacionSaludable							
src	5,628	2	0	755	0	0.0%	15.8%

	Lines of Code	Bugs	Vulnerabilities	Code Smells	Security Hotspots	Coverage	Duplications
dominio	613	7	0	126	0	0.0%	0.0%
Alimento.java	100	1	0	28	0	0.0%	0.0%
ComidaPorDia.java	54	2	0	8	0	0.0%	0.0%
Mensaje.java	45	0	0	1	0	0.0%	0.0%
Persona.java	65	2	0	1	0	0.0%	0.0%
PlanDeAlimentacion.java	73	0	0	22	0	0.0%	0.0%
Profesional.java	87	0	0	37	0	0.0%	0.0%
Sistema.java	189	2	0	29	0	0.0%	0.0%
Usuario.java	0	0	0	0	0	—	0.0%

5.1.5. Documentación

La documentación se modifiko para que se correspondiera con el sistema actual, además se agregaron partes faltantes y se hizo que esta cumpliera con el formato de escritura 302.

6. Analizar el estado de calidad final del software y productos asociados.

El objetivo general del proyecto es la mejora y aseguramiento de la calidad, junto con la gestión del mantenimiento del mismo.

Para cumplir con esto, intentamos cumplir con los objetivos SMART propuestos en el obligatorio 1, ya que todos estos trataban sobre las mejoras que necesitaba el sistema.

Para evaluar la calidad del producto hicimos análisis con distintas herramientas, este análisis se puede ver detallado en la documentación del producto que se adjunta en la carpeta del obligatorio.

Cobertura de las pruebas unitarias

Se propuso como primer objetivo el aumentar el porcentaje de cobertura de las pruebas unitarias, ya que en el análisis inicial este era muy bajo.

Resultados iniciales: paquete dominio 81 % y paquete AlimentacionSaludable 0 %

Resultado esperado: objetivo logrado cuando se haya llegado en los resultado del analisis de jacoco a una cobertura del 95 % en el paquete dominio.

Resultado finales: paquete dominio con cobertura del 95 %

Bloques de código duplicado

Se propuso como segundo objetivo el disminuir el porcentaje de bloques de código duplicado, ya que en los resultados del análisis de sonarqube hay un porcentaje alto de código duplicado.

Resultados iniciales: 21,5 % de código duplicado en el paquete interfaz.

Resultados esperados: el objetivo se habrá logrado cuando se haya bajado, en los resultados de código duplicado de SonarQube, a un porcentaje de a lo sumo 10 %

Resultado finales : 16 % de código duplicado en el paquete interfaz, no se pudo llegar al resultado esperado ya que la mayor parte el código duplicado era código auto generado en la interfaz que no se podía modificar, este objetivo no se analizo tan

en profundidad al plantearlo lo que hizo que se propusiera un porcentaje difícil de cumplir.

Cumplimiento de algunas heurísticas de Nielsen

Se propuso como tercer objetivo el mejorar el cumplimiento de las heurísticas de Nielsen, y así mejorar la usabilidad.

Resultados iniciales: muchos puntos de las heurísticas no se cumplen lo que hace que sea más difícil de entender y usar.

Resultados esperados: el objetivo se habrá logrado cuando se haya cumplido con el 50 % de las heurísticas

Resultado finales: Se cumple con el 80 % de las heurísticas, superando los resultados esperados.

Lograr todos los cambios propuestos en el tiempo esperado.

Se propuso como cuarto objetivo, el cumplir con todos los cambios propuestos en secciones anteriores.

Resultados iniciales: se proponen 16 cambios.

Resultado finales: se implementa los 16 cambios con éxitos.

Calidad del proyecto en general

Se propuso como quinto y último objetivo, pero no menos importante, el mejorar la calidad del código en general en tiempo y costos estimados, ya que como vimos en los resultados del análisis con los distintos analizadores estáticos de código este tiene muchos puntos que mejorar.

Resultados iniciales: considerando solo aquellos resultados que para nosotras eran relevantes se encontraron 56 defectos.

Resultados esperados: el objetivo se habrá logrado cuando se haya disminuido la cantidad de defectos en un 50 por ciento en los resultados de los analizadores estáticos de código.

Resultado finales: se reportan tan solo 19 defectos sin solucionar por lo tanto se cumple con el objetivo.

Estos 14 defectos siguen estando presentes en el análisis de checkStyle y sonarqube.

Como dijimos anteriormente para contar la cantidad de defectos que se arreglaron nos fijamos en el análisis hecho en la nueva documentación del producto que se adjunta en la misma carpeta.

7. Informe final

Informe de estado del proyecto analiza el cronograma y presupuesto para el período de informe.

Periodo del informe 9/11/2020 - 6/12/2020

Equipo: Belén Martínez, Joselen Cecilia, Camila Burgueño

7.1. Logros obtenidos para este período de informe

ID	Actividad	Fecha estimada	Fecha realización	Esfuerzo estimado	Esfuerzo en horas
1.1	A03: Mejorar cobertura de pruebas unitarias.	20/11	5/11	198	288
1.1	A02: Reducir los bugs reportados por FindBugs	11/11	19/11	92	92
1.2	A04: Rediseñar interfaz de usuario	9/11	20/11	332	368
2.1	A08: Definir los estándares de calidad en los que nos basaremos	15/10	18/10	40	40
2.1	A09: Definir los criterios de calidad en los que nos basaremos	15/10	18/10	40	40
2.2	A10: Analizar código con analizadores estáticos de código.	23/11	4/12	10	80
2.7	A12: Correr pruebas unitarias y documentar resultado	27/11	30/11	20	20
2.8	A11: Documentar el resultado de los análisis	2/12	5/12	40	50
2.3	A15: Generar casos de prueba	30/11	5/12	20	20
2.4	A13: Generar casos de prueba	4/12	4/12	80	80
2.5	A16: Definir los requerimientos funcionales	15/10	27/11	80	40
2.5	A17: Definir los requerimientos funcionales	15/10	27/11	40	40
2.6	A14: Definir las sesiones de ejecución de pruebas	7/12	4/12	40	40

Como se ve en la comparación de la fecha estimada con la fecha de realización hay cambios que se terminan antes de la fecha esperada y otros que se atrasan esto se debe a cambios en la organización, porque creíamos que algunos cambios eran más primordiales que otros.

7.2. Logros planificados, pero no completados en este período de informe.

Para el informe de cierre se han completado todas las tareas propuestas.

7.3. Causa raíz de las variaciones

Se hicieron variaciones en el cronograma ya que se decidió dar prioridad a las tareas que tenían mayor importancia, es decir aquellas tareas que formaban parte de los objetivos SMART o parte de los cambios pedidos por el cliente.

Además hubo unas semanas en las que no se trabajó las horas esperadas sobre el proyecto ya que los integrantes de este tenían otras entregas por lo que se decidió luego de estas aumentar las horas por días para recompensar el trabajo perdido y llegar así en tiempo y costo al final del proyecto

7.4. Impacto de la variación en los próximos hitos o en la fecha de finalización del proyecto.

Para este punto vamos a analizar los valores de SPI y SV

SPI:Índice de desempeño del cronograma.

$SPI = EV/PV$

SV:Determina en qué medida el proyecto está adelantado o retrasado en relación con la fecha de entrega, en un momento determinado. $SV = EV - PV$

EV:Es la medida del trabajo realizado en términos de presupuesto autorizado para dicho trabajo. $EV = \% \text{ Avance real} * BAC$ $EV = 0,6 * 1578USD = 946,8$ $0,6 = 60\%$ del Proyecto que faltaba implementar complementario al 40

PV:Valor planificado

28 días- 883,6USD

48 días -1578USD

$PV = 920,5USD$

Resultados $SV = 946,8 - 920,5 = 26,1$ -mayor a 0 lo que indica por definición que el ritmo del proyecto es más rápido que lo presupuestado pero en una mínima medida.

$SPI = 920,5 / 946,8 = 0,97$ -menor a 1 lo que indica por definición que estamos por encima de los costos planificados, pero en una muy pequeña medida.

7.5. Acciones correctivas o preventivas planificadas.

Cuando hicimos el plan de riesgos, tuvimos en cuenta que por distintas razones podían surgir retrasos en el tiempo del proyecto y definimos ciertas acciones para corregir estos problemas estas fueron:

- 1.Revisar el cronograma diariamente para no sobrepasar el tiempo destinado a cada tarea y en cada de que suceda identificarlo a tiempo.
- 2.Agregar más horas de trabajo para no aplazar a las tareas que le siguen.
- 3.Volver a planificar el proyecto, cambiando las fechas de las tareas, para esto se modifíco el GANT.

7.6. Fondos gastados (presupuesto ejecutado) en este período de informe.

Para calcular el costo total con los datos hasta el momento vamos a calcular el EAC(estimación del costo del proyecto).

$$EAC= AC+(BAC-EV) = 542+1578-946,8=1173$$

Costos de este periodo de informe 1173USD

7.7. Situación del presupuesto del proyecto.

El presupuesto planificado del proyecto era de 1578USD y hoy en día es de 2693USD, esto se debe a que se trabajo mas horas de lo planificado, para corregir el aplazamiento en el cronograma, y así poder llegar al objetivo final en fecha.

En la sección “3. Cambios implementados” de la documentación se detalla en que se gasto este presupuesto.

7.8. Situación de los riesgos identificados

Cuando hicimos el plan de riesgos, como dijimos anteriormente, identificamos dos de los riesgos que están sucediendo en este momento:

R01: al armar el cronograma de trabajo no se estimo correctamente el tiempo de cada tarea, lo cual puede provocar retraso en los plazos de entrega.

R09: mala estimaciones del costo del proyecto. Como habíamos previsto estos tenían alta probabilidad de ocurrencia y también alto impacto ya que afectan dos de los puntos principales al planificar el proyecto, el tiempo y el costo de este, esto puede generar problemas con el cliente, y baja de calidad del proyecto.

Para evitar que estos riesgos empeoraran se aplicaron las estrategias de respuestas mencionadas anteriormente:

- 1.Revisar el cronograma diariamente para no sobrepasar el tiempo destinado a cada tarea y en cada de que suceda identificarlo a tiempo.

- 2.Agregar más horas de trabajo para no aplazar a las tareas que le siguen.

- 3.Volver a planificar el proyecto, cambiando las fechas de las tareas, para esto se modificó el GANT.

7.9. Evaluación de la calidad de los productos

Calidad de código

Podemos decir que se mejoro el código en cuanto a bugs, líneas de código, nombres de variables y métodos, también se arregló la indentación, imports no utilizados y se agregaron @override faltantes. Estos cambios en conjunto hacen que la calidad del código mejore ampliamente. La cantidad de líneas de código creció ya que se implementaron nuevas funcionalidades las cuales llevaron a escribir nueva lógica, crear nuevas ventanas y remodelar la interfaz lo que genero mas código en el paquete interfaz y dominio.

Calidad de la documentación

En la documentación del proyecto se mide cuanto mejoro la calidad del código comparando con las métricas obtenidos en el obligatorio anterior.

Calidad de la interfaz

Actualmente la interfaz cumple con 8 de las 10 heurísticas de Nielsen esto se logró, agregando nuevas validación, funcionalidades y mejorando la estética de la misma.

Calidad de las pruebas unitarias

Se aumenta la cobertura de las pruebas unitarias, mejorando los test e implementando nuevos generando una cobertura de 95

Calidad de la documentación

Se actualizo la documentación haciendo que esta cumpla con el estándar 302, y agregando o modificando las partes que se consideraron necesarias para que corresponda con el obligatorio realizado por nuestro equipo.

8. Reflexiones del equipo sobre las fortalezas y debilidades del trabajo realizado

Gracias al uso de la de un repositorio GIT pudimos trabajar con el equipo de forma eficiente y ordenada, ya que esta herramienta nos permitió ir haciendo distintos cambios en el proyecto simultáneamente.

Este proyecto fue un desafío para todas las integrantes del equipo ya que ninguna había realizado anteriormente un mantenimiento de software, lo que dificultó la parte de generar un plan de proyecto adecuado y eficiente.

Creemos que los errores en la estimación tanto del costo, como del tiempo en determinadas tareas se debe a esta falta de experiencia en el tema, ya que siempre actuábamos solamente como desarrolladoras pero no como, project manager.

Esto sumado a que la documentación del obligatorio era escasa y poco informativa, junto con la mala calidad del código de la aplicación llevo a una mayor dificultad, ya que en un comienzo costaba entender el código, lo que genero una mayor demanda de tiempo del esperando.

Creemos que este obligatorio nos familiarizo de un forma muy directa con esta tarea, que de forma inevitable nos vamos a topar en nuestro futuro profesional.

Este obligatorio nos enseñó y nos mostró lo importante que es la planificación de un proyecto, ya que la mala estimación tanto de costos como de tiempo puede generar problemas.

Entendimos que un buen plan de proyecto con buenas acciones correctivas frente a riesgos planeados facilita la corrección de las desviaciones en el cronograma y en las estimaciones.

Creemos que se cumplió con los objetivos planteados en el obligatorio 1 pero que aun hay cosas que se pueden seguir mejorando, también creemos que al haber mejorado la documentación y la calidad del código en caso de querer hacer un nuevo mantenimiento a esta aplicación sera mas fácil.

Bibliografía

- [1] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, eight ed. McGraw-Hill, 2014.
- [2] I. Sommerville, *Software Engineering*, 10th ed. Pearson, 2015.
- [3] Universidad ORT Uruguay. (2013) Documento 302 - Facultad de Ingeniería. [Online]. Available: <https://www.ort.edu.uy/fi/documentos/normas-especificas-para-la-presentacion-de-trabajos-finales-de-carrera.pdf>