

# UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO



## “PLAN DE ADMINISTRACIÓN DE PROYECTO DE SOFTWARE”

NOMBRES:

- ABASTO BERBETTY, RODRIGO
- POZADAS MAMANI, INGRID NATHALY

GRUPO: AJ

MATERIA: INGENIERÍA DE SOFTWARE II

FECHA: 16/MAR/2021

SANTA CRUZ - BOLIVIA

## Contenido

Propósito del plan .....	1
Métricas .....	2
Casos de estudio .....	2
Métricas .....	5
Métrica orientada al tamaño.....	5
Métrica orientada a la función.....	5
Estimaciones .....	8
Estimación de KLDC.....	8
COCOMO II .....	9
Puntos Objeto .....	9
Productividad .....	9
Esfuerzo.....	9
Gestión de riesgos .....	0
Planificación del tiempo.....	0
Diagrama de Gantt.....	0
Diagrama de Pert .....	1
Tabla de recursos.....	2
Organización Interna.....	3
Mecanismos de seguimiento y control .....	3
Bibliografía.....	8
Anexos .....	9

## Propósito del plan

- Definir y establecer el ámbito donde se desarrollará el proyecto.
- Planificar la gestión de los recursos que se utilizara en el proyecto.
- Planificar las actividades que se realizarán para la elaboración del proyecto.
- Analizar los riesgos y planificar las acciones preventivas a ser aplicadas.
- Proporcionar la guía de desarrollo de software a todo el personal que trabaja en el proyecto.

## Métricas

## Casos de estudio

### *Titan Fire System - Sistema de Manómetros Inalámbricos*

TITAN FIRE SYSTEM es una empresa europea ubicada en España, la cual diseña y fabrica manómetros inteligentes inalámbricos y su vez proporciona un sistema con el cual gestionarlos. Trabajan con tecnología NFC, para la configuración de los equipos, evitando la necesidad de utilizar ordenadores para su conectividad, también han diseñado para su uso exclusivo un nuevo tamper de anilla, un manómetro de entre 0 a 30 bares que aseguran que puede soportar cualquier situación climatológica y baterías de muy alta calidad para garantizar el prolongado funcionamiento de todo el equipo. (Titan Fire System, s.f.)

Entre las características del software que ellos proporcionan encontramos las siguientes:

- Tiene dos entornos diferenciados. Uno diseñado para el instalador / mantenedor. Y el otro para el cliente final.
- En caso del instalador permite control de todas las instalaciones de sus clientes.
- En caso del cliente final permite la visualización sólo de su instalación.
- Acceso a través de usuario y contraseña.
- Registro de varios niveles de alarma. Movimiento, anilla, presión y comunicación.
- Aviso de alertas vía Software, Email y SMS.

Una aplicación móvil únicamente para sistemas Android con NFC destinada para el instalador / mantenedor, login necesario, permite la instalación de los manómetros y realizar el mantenimiento de instalaciones ya creadas.

### *Imec Technologies - Fire Extinguisher Inspection Software*

IMEC Technologies es un proveedor de soluciones de software centradas en dispositivos móviles que opera en el espacio de EHS (Salud, Seguridad y Medio Ambiente), presenta un software centrado la gestión de las inspecciones a equipos de seguridad contra incendios de una manera electrónica. (IMEC, s.f.)

Características principales:

- Sistema de alertas por correo electrónico brindan visibilidad de todas las inspecciones pendientes y próximas para ubicaciones, laboratorios, incendios, seguridad humana y equipos.
- Administración de todos los puntos de inspección, por ejemplo, laboratorios, extintores.
- Usa un sistema permite que el inspector simplemente escanee un código de barras o un código QR para iniciar una inspección.
- Permite gestionar elementos como extintores de incendios, bombas contra incendios, DEA, gabinetes de líquidos inflamables, estaciones de lavado de ojos, elevadores, etc.
- Gestionar una agenda de inspecciones, generando recordatorios programados, dentro del tiempo que se asigne.

- Paneles de inspección en cumplimiento de las regulaciones, por ejemplo, NFPA.

### *Bp Srl - Smart Fire System Safety*

Denominado Smart Fire System Safety (también conocido como SFSS) es un Sistema compuesto por Hardware, Software y servicios en la nube, que propone monitorear sus parámetros y notificar rápidamente cualquier mal funcionamiento. (Tech Briefs, s.f.)

La infraestructura del sistema SFSS está compuesta por un componente Smart instalado en el mango del extintor que monitorea 24/7: presión, manguera, movimiento, pasador de seguridad, temperatura y envía estos datos, a través un Gateway o puerta de enlace que se encarga de recibir y reenviar datos con un protocolo IoT específico, a una plataforma Cloud, en caso de problemas, envía alarmas a un centro operativo o directamente en Smartphone, para asegurarse de que los técnicos relevantes puedan intervenir rápidamente en el lugar. El software del sistema SFSS también permite rastrear las instalaciones junto con la posición de la instalación del extintor, para monitorear tanto la actividad de mantenimiento ordinario como extraordinario que se realiza en los extintores.

## Métricas

### Métrica orientada al tamaño

Aplicación	LDC	Costo(\$)	Tiempo(mes)	Esfuerzo	Pág. Doc.	Gente	Errores	Defectos
Titan Fire System	175K	20M	20	5000	500	250	50	20
Imec Technologies	150K	13M	18	3780	450	210	60	23
Smart Fire System Safety	130K	10M	14	1680	400	120	55	35

Calidad[Errores+Defectos/LDC]	Productividad[LDC/Esfuerzo]
$(50+20)/175000 = 0.0004$	$175000/5000 = 35$
$(60+23)/150000 = 0.0005$	$150000/3780 = 39.7$
$(55+35)/130000 = 0.0007$	$130000/1680 = 77.4$

### Métrica orientada a la función

#### Titan Fire System

- Parámetro de medición	Cuenta	Simple	Medio	Complejo	Total
# Entradas de usuario	25	3	4	6	75
# Salidas de usuario	5	4	5	7	20
# Peticiones	5	3	4	6	20
# Archivos	30	7	10	15	300
# Interfaces externas	0	5	7	10	0
					415

Factor	No influye 0	Incidental 1	Moderado 2	Medio 3	Significativo 4	Esencial 5	F
1. ¿El sistema requiere respaldo y recuperación confiables?		x					1
2. ¿Se requieren comunicaciones de datos especializadas para transferir información hacia o desde la aplicación?				x			3
3. ¿Existen funciones de procesamiento distribuidas?	x						0
4. ¿El desempeño es crucial?			x				2
5. ¿El sistema correrá en un entorno operativo existente enormemente utilizado?			x				2
6. ¿El sistema requiere entrada de datos en línea?					x		4
7. ¿La entrada de datos en línea requiere que la transacción de entrada se construya sobre múltiples pantallas u operaciones?		x					1

8. ¿Los ALI se actualizan en línea?			x				2
9. ¿Las entradas, salidas, archivos o consultas son complejos?			x				2
10. ¿El procesamiento interno es complejo?				x			3
11. ¿El código se diseña para ser reutilizable?			x				2
12. ¿La conversión y la instalación se incluyen en el diseño?		x					1
13. ¿El sistema se diseña para instalaciones múltiples en diferentes organizaciones?		x					1
14. ¿La aplicación se diseña para facilitar el cambio y su uso por parte del usuario?			x				2
							26

$$PF = CTA \text{ TOTAL} * [0.65 + 0.01 \sum_{i=1}^{14} f_i] = 415 * (0.65 + (0.01 * 26)) = 377.65$$

Imec Technologies

- Parámetro de medición	Cuenta	Simple	Medio	Complejo	Total
# Entradas de usuario	30	3	4	6	90
# Salidas de usuario	5	4	5	7	20
# Peticiones	4	3	4	6	12
# Archivos	15	7	10	15	105
# Interfaces externas	0	5	7	10	0
					227

Factor	No influye 0	Incidental 1	Moderado 2	Medio 3	Significativo 4	Esencial 5	F
1. ¿El sistema requiere respaldo y recuperación confiables?	x						0
2. ¿Se requieren comunicaciones de datos especializadas para transferir información hacia o desde la aplicación?					x		4
3. ¿Existen funciones de procesamiento distribuidas?	x						0
4. ¿El desempeño es crucial?					x		4
5. ¿El sistema correrá en un entorno operativo existente enormemente utilizado?					x		4
6. ¿El sistema requiere entrada de datos en línea?				x			3



7. ¿La entrada de datos en línea requiere que la transacción de entrada se construya sobre múltiples pantallas u operaciones?			x				2
8. ¿Los ALI se actualizan en línea?			x				2
9. ¿Las entradas, salidas, archivos o consultas son complejos?	x						0
10. ¿El procesamiento interno es complejo?		x					1
11. ¿El código se diseña para ser reutilizable?			x				2
12. ¿La conversión y la instalación se incluyen en el diseño?			x				2
13. ¿El sistema se diseña para instalaciones múltiples en diferentes organizaciones?		x					1
14. ¿La aplicación se diseña para facilitar el cambio y su uso por parte del usuario?		x					1
							26

$$PF = CTA \text{ TOTAL} * [0.65 + 0.01 \sum_{i=1}^{14} f_i] = 227 * (0.65 + (0.01 * 26)) = 206.57$$

#### Smart Fire System Safety

- Parámetro de medición	Cuenta	Simple	Medio	Complejo	Total
# Entradas de usuario	23	3	4	6	69
# Salidas de usuario	6	4	5	7	24
# Peticiones	5	3	4	6	15
# Archivos	14	7	10	15	98
# Interfaces externas	2	5	7	10	10
					216

Factor	No influye	Incidental	Moderado	Medio	Significativo	Esencial	F
	0	1	2	3	4	5	
1. ¿El sistema requiere respaldo y recuperación confiables?	x						0
2. ¿Se requieren comunicaciones de datos especializadas para transferir información hacia o desde la aplicación?			x				2
3. ¿Existen funciones de procesamiento distribuidas?			x				2
4. ¿El desempeño es crucial?		x					1
5. ¿El sistema correrá en un entorno operativo existente enormemente utilizado?			x				2

6. ¿El sistema requiere entrada de datos en línea?				x			3
7. ¿La entrada de datos en línea requiere que la transacción de entrada se construya sobre múltiples pantallas u operaciones?			x				2
8. ¿Los ALI se actualizan en línea?			x				2
9. ¿Las entradas, salidas, archivos o consultas son complejos?			x				2
10. ¿El procesamiento interno es complejo?			x				2
11. ¿El código se diseña para ser reutilizable?			x				2
12. ¿La conversión y la instalación se incluyen en el diseño?		x					1
13. ¿El sistema se diseña para instalaciones múltiples en diferentes organizaciones?	x						0
14. ¿La aplicación se diseña para facilitar el cambio y su uso por parte del usuario?		x					1
							22

$$PF = CTA \text{ TOTAL} * [0.65 + 0.01 \sum_{i=1}^{14} f_i] = 216 * (0.65 + (0.01 * 22)) = 187.92$$

## Estimaciones

### Estimación de KDLC

$$Ve = \frac{(a)\text{Optimista} + 4(b)\text{más probable} + (c)\text{Pesimista}}{6}$$

Proyecto	KDLC		
	a	b	c
Medical-App	10	14	30

$$Ve = \frac{10 + 4 \times 14 + 30}{6} = 16 \text{ KDLC}$$

## COCOMO II

### Puntos Objeto

Tipo Objeto	Cuenta	Simple	Medio	Avanzado	Total
Pantallas	55	1	2	3	55
Reportes	6	2	5	8	12
Componentes 3GL	0			10	0
Puntos Objetos ->					67

$$PON = PO \times \frac{100 - \% \text{ Reutilizacion}}{100}$$

$$PON = 67 \times \frac{100 - 20}{100} = 67 \times 0.8 = 53.6$$

### Productividad

Proporciones de Productividad	Muy Bajo	Bajo	Normal	Alta	Muy Alta
Capacidad y Madurez del Equipo			x		
Capacidad y Estructuración del entorno			x		
PRODUCTIVIDAD	4	7	13	25	50

$$PROD = 13 + 13 = 26$$

### Esfuerzo

$$Esfuerzo = \frac{PON}{PROD} = \frac{53.6}{26} = 2.06$$

## Gestión de riesgos

Identificar Riesgo	Prob. de presencia 1-100	impacto	Reducir prob. presencia	Reducir impacto
R1. Aumento del costo del proyecto	70	Significativo	<ul style="list-style-type: none"> <li>• Verificar la estimación de costo</li> <li>• Evitar gastos innecesarios</li> </ul>	<ul style="list-style-type: none"> <li>• Tener un fondo de respaldo</li> </ul>
R2. No terminar el proyecto en la fecha prevista	80	Significativo	<ul style="list-style-type: none"> <li>• Realizar planificación de los hitos del proyecto</li> <li>• Realizar seguimiento y control a lo planeado</li> </ul>	<ul style="list-style-type: none"> <li>• Incremento en horario de trabajo</li> <li>• Procurar flexibilidad de fecha de entrega con el cliente</li> </ul>
R3. Disminución de calidad	50	Medio	<ul style="list-style-type: none"> <li>• Definición correcta de los requerimientos</li> <li>• Realizar seguimiento y control a lo planeado</li> <li>• Transferir correctamente la idea de lo que se quiere desarrollar</li> </ul>	<ul style="list-style-type: none"> <li>• Planificar mantenimientos luego de la entrega del producto</li> </ul>
R4. Perdida de RRHH	65	Significativo	<ul style="list-style-type: none"> <li>• Contrato</li> <li>• Aumentar motivación</li> </ul>	<ul style="list-style-type: none"> <li>• Adoptar un estándar de codificación</li> <li>• Tener vacantes disponibles</li> <li>• Utilizar herramientas de desarrollo sencillas</li> </ul>
R5. Sufrir daños personales o a la propiedad	30	Medio	<ul style="list-style-type: none"> <li>• Contar con un sistema de seguridad confiable</li> <li>• Evitar acciones imprudentes.</li> </ul>	<ul style="list-style-type: none"> <li>• Asegurar los equipos y al personal</li> </ul>

R6. Comprensión incorrecta de los requerimientos	55	Significativo	<ul style="list-style-type: none"> <li>Realizar entrevistas planificadas con el cliente</li> <li>Presentación de prototipos del producto al cliente previo al desarrollo</li> </ul>	<ul style="list-style-type: none"> <li>Planificar mantenimientos</li> <li>Asegurar la escalabilidad y flexibilidad en el código</li> </ul>
R7. Problemas con el Hardware	60	Medio	<ul style="list-style-type: none"> <li>Mantener al día los mantenimientos</li> <li>Comprar productos de calidad</li> </ul>	<ul style="list-style-type: none"> <li>Tener equipos de repuesto</li> <li>Tener un fondo de emergencia</li> </ul>
R8. Falta de experiencia con las herramientas software	45	Medio	<ul style="list-style-type: none"> <li>Realizar capacitaciones</li> <li>Utilizar herramientas comunes de fácil comprensión</li> </ul>	<ul style="list-style-type: none"> <li>Tener herramientas alternas a la solicitada</li> </ul>
R9. Los cambios en el desarrollo no son controlados	70	Medio	<ul style="list-style-type: none"> <li>Utilizar una herramienta de control y versionamiento</li> <li>Usar roles bien definidos</li> </ul>	<ul style="list-style-type: none"> <li>Usar backups o puntos de restauración</li> </ul>
R10. El cliente no define bien sus requerimientos	70	Significativo	<ul style="list-style-type: none"> <li>Utilizar preguntas claves y planificadas en las entrevistas</li> <li>Asesoramiento profesional para el cliente</li> </ul>	<ul style="list-style-type: none"> <li>Incluir el alcance definido en el contrato con el cliente</li> </ul>
R11. Poca comunicación en el equipo de desarrollo	70	Significativo	<ul style="list-style-type: none"> <li>Realizar actividades de integración social</li> </ul>	<ul style="list-style-type: none"> <li>Definir fechas para rendición de cuentas</li> </ul>
R12. Poco rendimiento del equipo de desarrollo	60	Significativo	<ul style="list-style-type: none"> <li>Realizar capacitaciones</li> <li>Realizar seguimiento y control a lo planeado</li> <li>Incrementar motivación</li> </ul>	<ul style="list-style-type: none"> <li>Tener personal extra o de respaldo</li> </ul>

Planificación del tiempo

Diagrama de Gantt

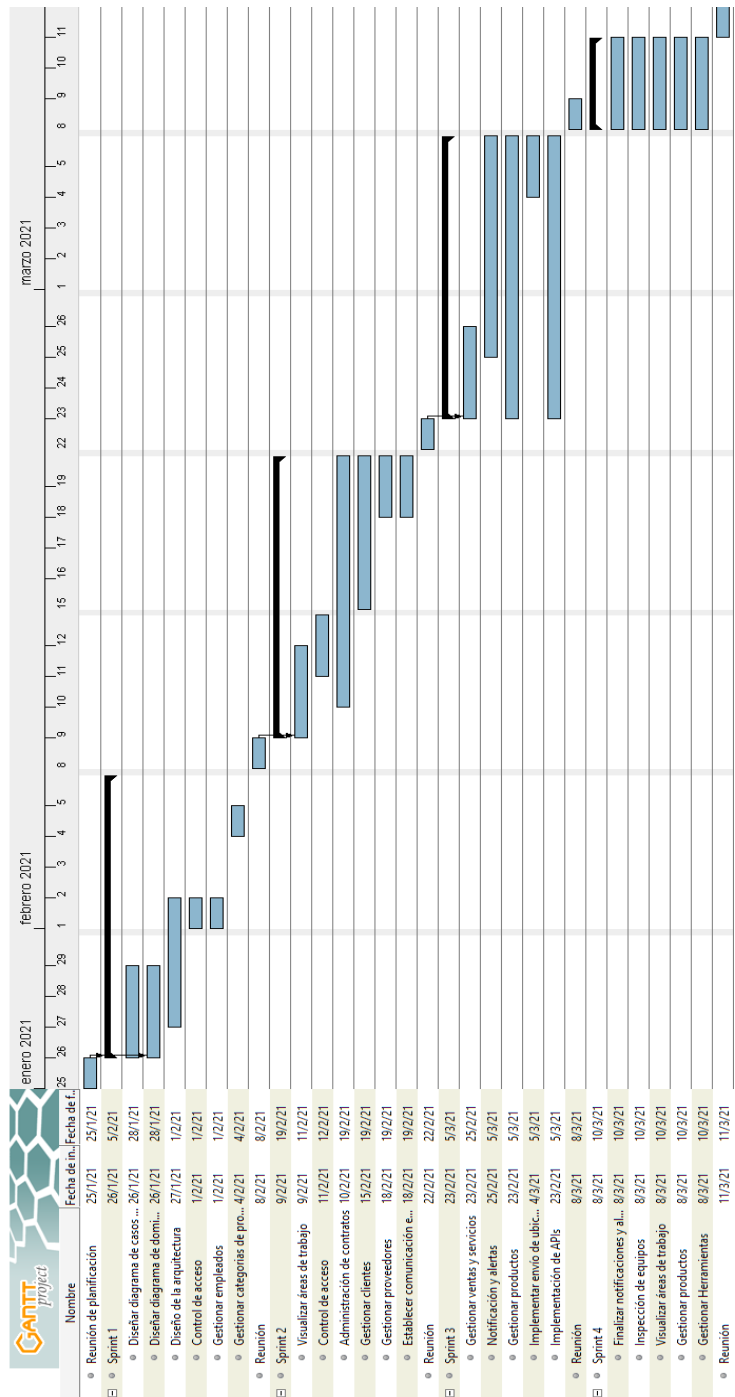


Diagrama de Pert

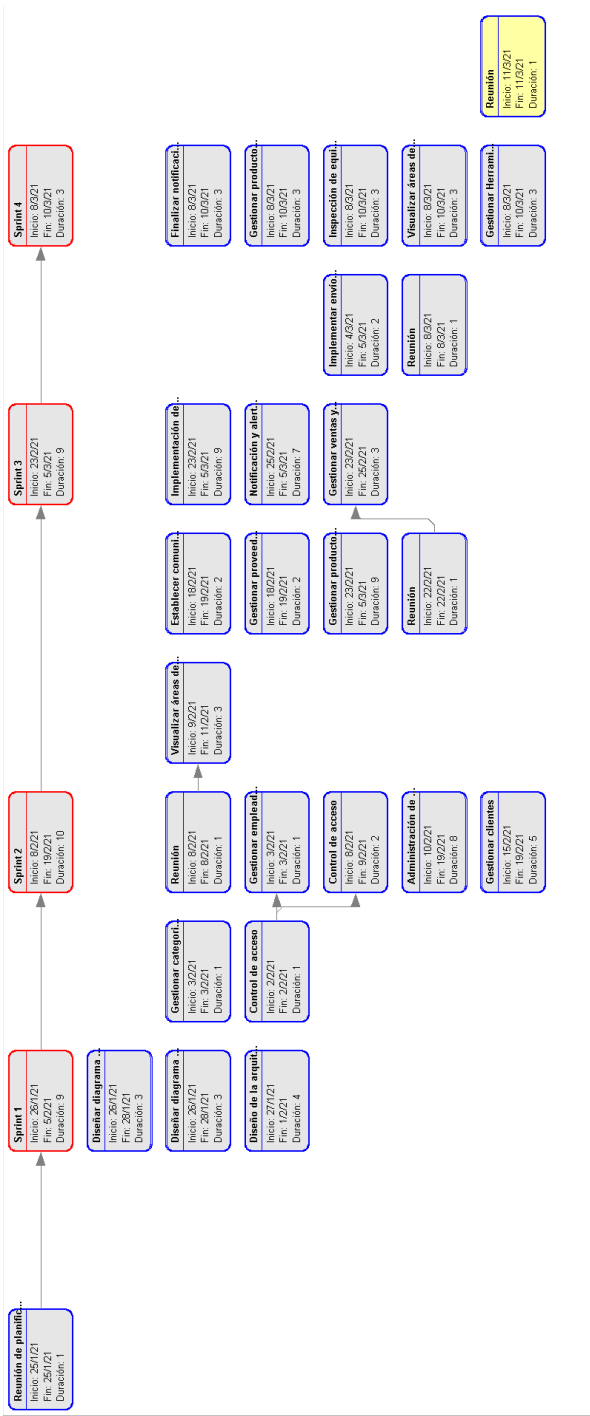


Tabla de recursos

Recurso	Fecha		Cantidad	Precio Unitario (USD)	% Depreciación	Precio Unitario Neto	Total
	Desde	Hasta					
Software							
Licencia Windows	26-ENE	10-MA	6	439	33	144.87	869.22
Licencia Avast	26-ENE	10-MA	6	7.5	33	2.475	14.85
Android Studio	26-ENE	10-MA	6	0	33	0	0
Visual Studio Code	26-ENE	10-MA	6	0	33	0	0
GitHub Team	26-ENE	10-MA	4	9	33	2.97	11.88
Licencia Office 365	26-ENE	10-MA	6	50	33	16.5	99
Architech Enterprise Unified	26-ENE	10-MA	4	500	33	165	660
Postman Pro	26-ENE	10-MA	4	8	33	2.64	10.56
MEGA Business Account	26-ENE	10-MA	4	12	33	3.96	15.84
NinjaMock Pro Team	26-ENE	10-MA	4	40	33	13.2	52.8
Taiga	26-ENE	10-MA	4	5	33	1.65	1.65
Hardware							
Workstation Precision 5820	26-ENE	10-MA	6	1430	25	429	2574
Impresora HP LaserJet Pro M281fdw	26-ENE	10-MA	1	320	25	80	80
Monitor Dell 24” E2417H	26-ENE	10-MA	12	130	25	32.5	260
UPS Forza SL-1011UL	26-ENE	10-MA	4	100	25	25	100
Proyector Epson PowerLite 1266	26-ENE	10-MA	1	650	25	162.5	162.5
SmartPhone Samsung A30	26-ENE	10-MA	8	230	25	57.8	460
Altavoces Logitech Z313	26-ENE	10-MA	1	58	25	14.5	14.5
Switch 8-Port T2500G-10TS	26-ENE	10-MA	1	100	25	25	25
Cable RJ45	26-ENE	10-MA	10	2	25	0.5	5
Arduino	26-ENE	10-MA	2	70	25	17.5	35
Case arduino	26-ENE	10-MA	2	20	20	4	8
GPRS	26-ENE	10-MA	2	120	30	3600	7200
Cargador	26-ENE	10-MA	2	45	20	9	18
Jumpers	26-ENE	10-MA	2	10	10	1	2
Módulo GPS	26-ENE	10-MA	2	70	25	17.5	35
Infraestructura							
Alquiler	26-ENE	10-MA	3	600	0	600	1800
Energía eléctrica	26-ENE	10-MA	3	120	0	120	360
Agua Potable	26-ENE	10-MA	3	20	0	20	60
Internet 50 Mbps	26-ENE	10-MA	3	100	0	100	300
Gente							
Desarrolladores	26-ENE	10-MA	6	700	0	700	4200
Logística							
Cartuchos impresora	26-ENE	10-MA	1	92	0	92	92
Papelero	26-ENE	10-MA	5	5	0	5	25
Mousepad	26-ENE	10-MA	8	2	0	2	16
Paquete hoja Bond	26-ENE	10-MA	3	4	0	4	12
Paquete Bolígrafo	26-ENE	10-MA	1	4	0	4	4



Transporte	26-ENE	10-MA	16	10	0	10	160
Refrigerio	26-ENE	10-MA	16	12	0	12	192
Viático	26-ENE	10-MA	16	15	0	15	240
TOTAL							20175.72

### Organización Interna

#### Descentralizado Controlado (DC)

- Cada área tiene un responsable
- Existe jerarquía vertical

### Mecanismos de seguimiento y control

- Realizar reuniones periódicas al finalizar cada Sprint para verificar el estado del proyecto donde todos los miembros del equipo presentarán un informe de sus progresos y dificultades.
- Evaluar los resultados de todas las revisiones realizadas a lo largo del proceso de ingeniería de software.
- Comparar la fecha real de inicio con las previstas para cada tarea del proyecto.

A continuación presentamos el formato de los reportes:

## Reporte de Tareas.

### REPORTE DE TAREAS

# de Reporte: \_\_\_\_\_

Fecha:    /    /

Lugar: \_\_\_\_\_

Hora: \_\_\_\_\_

a) Tarea a cargo: \_\_\_\_\_

b) Descripción de la funcionalidad de la tarea:

\_\_\_\_\_  
\_\_\_\_\_

c) % de avance en su desarrollo: \_\_\_\_\_

d) Control de avance:

- Retrasado:	Si	No
- Según Planificado	Si	No

e) Lista de problemas encontrados:

\_\_\_\_\_  
\_\_\_\_\_

f) Lista de posibles soluciones:

\_\_\_\_\_  
\_\_\_\_\_

g) Integrantes del equipo a cargo de la tarea:

Nombres:

\_\_\_\_\_  
\_\_\_\_\_

h) Observaciones:

\_\_\_\_\_

## Tareas en Proceso.

### Reportes de tareas en proceso

N<sup>ro</sup> de Reunión: \_\_\_\_\_ Fecha: \_\_/\_\_/\_\_  
Lugar: \_\_\_\_\_ Hora: \_\_:\_\_:\_\_

a) *RTF:*

---

---

---

---

b) *Descripción de tareas:*

---

---

---

---

Jefe de grupo

**Reporte de Tareas Finalizadas.**

**Reportes de tareas finalizadas**

Nombre de la tarea: \_\_\_\_\_ Fecha: \_\_/\_\_/\_\_

Lugar: \_\_\_\_\_ Hora: \_\_:\_\_:\_\_

*Descripción de la tarea:*

---

---

---

*Recomendación:*

---

---

---

\_\_\_\_\_  
Jefe de grupo

## Informe de Errores.

### Informes de errores

Nombre de la tarea: \_\_\_\_\_ Fecha: \_\_/\_\_/\_\_

Lugar: \_\_\_\_\_ Hora: \_\_:\_\_:\_\_

*Descripción y ubicación:*

\_\_\_\_\_  
\_\_\_\_\_

*Impacto:*

\_\_\_\_\_  
\_\_\_\_\_

*Causa:*

\_\_\_\_\_  
\_\_\_\_\_

*Solución:*

\_\_\_\_\_  
\_\_\_\_\_

\_\_\_\_\_  
Jefe de grupo

## Bibliografía

tfs. (Recuperado el 20 de enero de 2021). *Titan Fire System*.

<https://www.tfs-system.com/es/index.php>

imec technologies. (Recuperado el 20 de enero de 2021). *Titan Fire System*.

<https://imectechnologies.com/>

Tech Briefs. (Recuperado el 05 de enero de 2021). *Smart Fire System Safety*.

<https://contest.techbriefs.com/2017/entries/electronics-sensors-iot/7960>

## Anexos

### **Precision 5820 Tower Workstation**

Intel Xeon W-2123 (3.6GHz, 3.9GHz Turbo, 4C, 8.25MB Cache, HT, (120W) ) DDR4-2666

Windows 10 Pro for Workstation (up to 4 Cores) Multi - English, French, Spanish

Precision 5820 Tower 425W Chassis

Radeon Pro WX 4100, 4GB, 4 mDP (5820T)

16GB, 2x8GB, DDR4, 2666MHz, RDIMM, ECC Memory

2.5" 1TB 7200rpm SATA Hard Drive

T5820 Resource DVD

Dell KB216 Wired Keyboard

Dell MS116 Wired Mouse Black

(1400 USD)

### **LaserJet Pro M281fdw**

Funciones - Imprime, copia, escanea, envía fax

Compatibilidad con varias tareas - Sí

Volumen de páginas mensuales - 150 a 2500

Tecnología - Láser

Calidad de Impresión - Hasta 600 x 600 ppp

Pantalla - 2,7" de gráficos a color

### **UPS Forza SL-1011UL**

Capacidad: 1000VA/600W

Topología: Interactiva

Forma de onda: Onda senoidal simulada

Voltaje: 120V

Tipo de entrada: NEMA 5-15P

Tipo de salida: 8 x NEMA 5-15R

**Epson PowerLite 1266**

Projection System: 3LCD, 3-chip technology

Native Resolution: 1280 x 800 (WXGA)

Color Brightness: 3600 lumens

**Samsung A30**

Pantalla: 6.4", 1080 x 2340 pixels

Procesador: Exynos 7904 1.8GHz

RAM: 3GB/4GB

Almacenamiento: 32GB/64GB

Expansión: microSD

Cámara: Dual, 16MP+5MP

Batería: 4000 mAh

OS: Android 9.0

**Monitor Dell 24" E2417H**

Diagonal de la pantalla: 23.8pulg.

Pantalla: LCD

Tipo HD: Full HD

Resolución: 1920 x 1080 Píxeles

Tiempo de respuesta: 8 ms



# UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO



## “MANUAL DE CALIDAD”

NOMBRES:

- ABASTO BERBETTY, RODRIGO
- POZADAS MAMANI, INGRID NATHALY

GRUPO: AJ

MATERIA: INGENIERÍA DE SOFTWARE II

FECHA: 16/MAR/2021

SANTA CRUZ - BOLIVIA

## Tabla de contenido

1	Propósito .....	4
2	Documentos de referencia.....	6
3	Gestión.....	7
3.1	Organización.....	7
3.2	Tareas.....	7
3.3	Roles y responsabilidades .....	9
3.3.1	Responsabilidades de la Organización del grupo de Desarrollo: ...	9
3.3.2	Responsabilidades de la Organización del Consultor: .....	9
3.3.3	Responsabilidades de la Organización del cliente: .....	10
3.3.4	Responsabilidades de la Organización del grupo SQA.....	10
3.4	Recursos estimados de aseguramiento de calidad .....	11
4	Documentación.....	12
4.1	Propósito.....	12
4.2	Documentación mínima requerida.....	12
4.2.1	Especificación de requerimientos del software .....	12
4.2.2	Descripción del diseño del software.....	14
4.2.3	Plan de verificación y validación .....	15
4.2.4	Informe de Resultados de Verificación y Validación .....	18
4.2.5	Documentación de usuario .....	21
4.2.6	Plan de gestión de configuración del software (SCMP) .....	23
5	Estándares, prácticas, convenciones y métricas.....	27
5.1	Estándares de documentación .....	27
5.2	Estándares de codificación .....	28
5.3	Estándares de comentarios.....	28

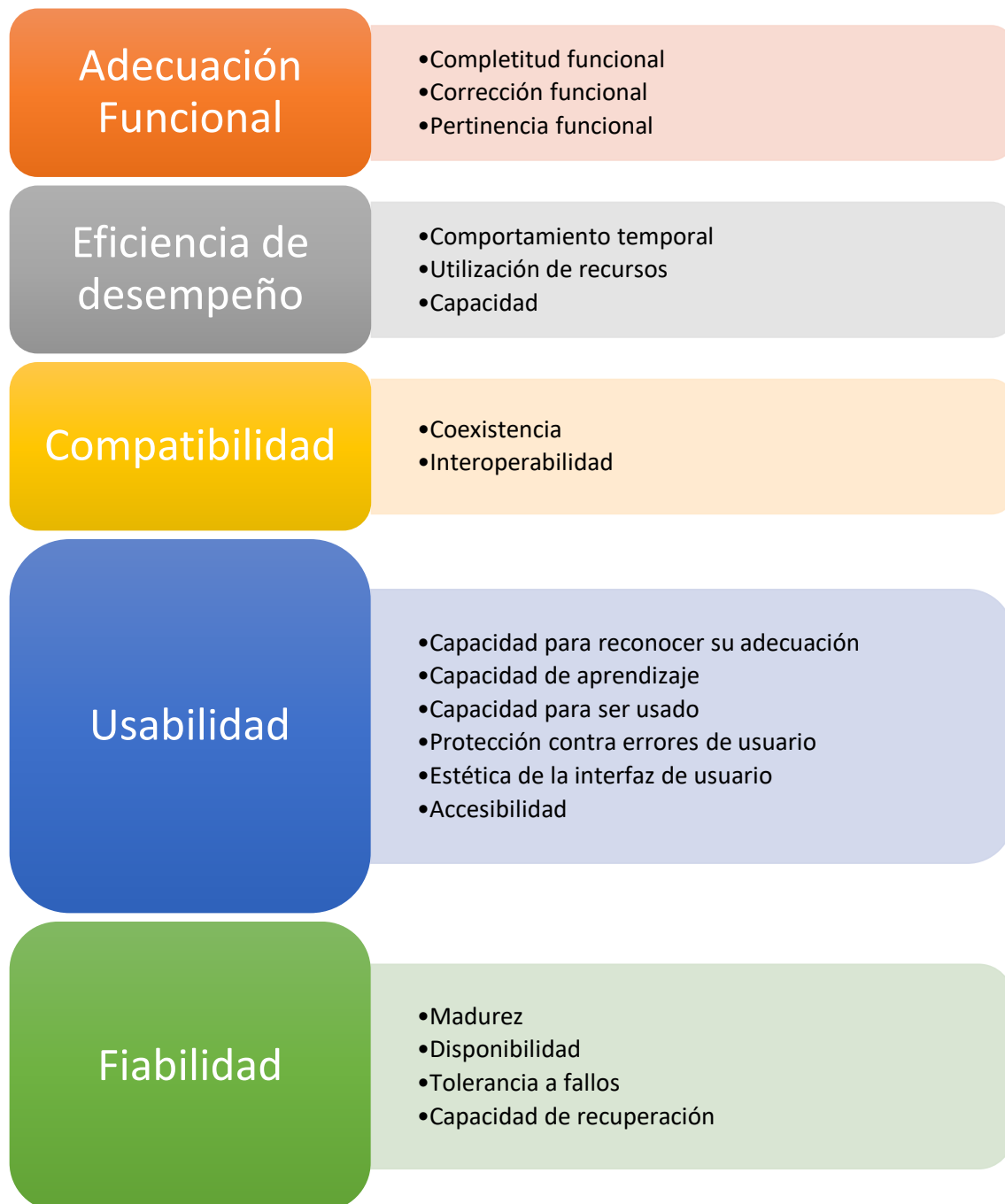
5.4	Estándares de verificación y prácticas .....	28
5.5	Métricas .....	28
6	Revisiones y Auditorias .....	30
6.1	Propósito.....	30
6.2	Requerimientos mínimos.....	30
6.2.1	Revisión de requerimientos .....	30
6.2.2	Revisión de diseño preliminar .....	31
6.2.3	Revisión de diseño crítico .....	32
6.2.4	Revisión de plan de verificación y validación .....	32
6.2.5	Auditoria funcional .....	32
6.2.6	Auditoria física .....	33
6.2.7	Auditorías internas al proceso .....	33
6.2.8	Revisiones gerenciales.....	33
6.2.9	Revisiones del plan de gestión de configuración .....	33
6.2.10	Revisión posterior a la implementación .....	34
7	Pruebas .....	34
8	Reportes de Problemas y Acciones Correctivas .....	39
9	Herramientas, Técnicas y Metodologías .....	44
10	Control de Código .....	45
11	Control de Medios .....	45
12	Control de Proveedores.....	47
13	Recolección de Documentos, mantenimiento y almacenado .....	47
14	Capacitación .....	48
14.1	Objetivos organizacionales .....	48
14.2	Objetivos específicos .....	49

14.3	Cursos impartidos .....	50
14.4	Desarrollo de cursos de capacitación .....	51
15	Gestión de riesgos .....	53

## 1 Propósito

Definir y describir las pautas a seguir durante el proceso de desarrollo, con el fin de poder asegurar la calidad del mismo y del producto a elaborar, detallando todo lo referente a la planificación del seguimiento de la calidad en el proyecto.

Con el seguimiento a las pautas detalladas de este manual se desea que el producto software cumpla con las características definidas por la ISO/IEC 25010:





*Ilustración 1. Factores de Calidad según ISO/IEC 25010*

El manual cubre los siguientes flujos de trabajo que se detallan dentro del Proceso Unificado de Desarrollo de Software: Captura de Requisitos, Análisis, Diseño, Implementación y Prueba.

## 2 Documentos de referencia

- ISO/IEC 25010 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.
- IEEE 730 IEEE Standard for Software Quality Assurance Plans.
- IEEE 860 Guide for Software Requirements Specifications.
- IEEE 1016 Recommended Practice for Software Design Descriptions
- IEEE 1063 Standard for Software User Documentation
- IEEE 1028 Standard for Software Reviews and Audits
- ISO/IEC 12207 - Information Technology / Software Life Cycle Processes

### 3 Gestión

#### 3.1 Organización

Las siguientes organizaciones son las que influirán y se harán cargo de controlar la calidad del software

- Organización del Grupo de Desarrollo de Software
- Organización del Consultor o Especialista en SQA
- Organización del Cliente.
- Organización del Grupo SQA

#### 3.2 Tareas

Las tareas asociadas con el ciclo de vida de desarrollo de software y las actividades de la SQA son las siguientes:

FT. PUDS	TAREAS Y ACTIVIDADES
Requerimientos	<ul style="list-style-type: none"> <li>• Captura de requisitos del software</li> <li>• Generación de especificaciones</li> <li>• Revisión de especificaciones</li> <li>• Revisión de las especificaciones de software</li> <li>• Modelo Conceptual</li> </ul>
Análisis	<ul style="list-style-type: none"> <li>• Definir los casos esenciales de uso</li> <li>• Perfeccionar los diagramas de casos de uso</li> <li>• Perfeccionar el modelo conceptual</li> <li>• Definir los contratos de operaciones</li> <li>• Definir los diagramas de comunicación y de estados</li> </ul>
Diseño	<ul style="list-style-type: none"> <li>• Diseño preliminar</li> <li>• Generación de especificaciones de diseño preliminar</li> <li>• Revisión del diseño preliminar</li> <li>• Diseño detallado</li> <li>• Generación de especificaciones de diseño detallado</li> </ul>



	<ul style="list-style-type: none"> <li>• Revisión del diseño detallado</li> <li>• Revisión del diseño preliminar</li> <li>• Revisión del diseño detallado del software</li> </ul>
Implementación	<ul style="list-style-type: none"> <li>• Codificación</li> <li>• Generación de código</li> <li>• Revisión de código</li> <li>• Revisión de código versus la documentación generada</li> </ul>
Pruebas	<ul style="list-style-type: none"> <li>• Elaboración de pruebas de unidad y generación de resultados</li> <li>• Revisión de resultados</li> <li>• Elaboración de las pruebas de unidad y de integración del software</li> <li>• Revisión de los resultados de las pruebas</li> <li>• Revisión de las pruebas funcionales y evaluación de los resultados</li> </ul>
Instalación y Prueba Final	<ul style="list-style-type: none"> <li>• Instalación del producto software</li> <li>• Prueba final bajo ambiente real</li> <li>• Generación de resultados de prueba</li> <li>• Revisión de resultados</li> <li>• Revisión de la instalación del software y evaluación de resultados</li> </ul>

Actividades y sus entregables:

Actividad	Entregable
Elaboración del Plan de SQA	Plan de SQA
Identificar propiedades de Calidad	Plan de SQA
Evaluación de la calidad de los productos	Informe de revisión de SQA
Revisar el ajuste al proceso	Informe de revisión de SQA
Realizar Revisión Técnica Formal	Informe de Revisión Técnica Formal
Evaluar y ajustar el Plan de SQA	Documento de Evaluación y Ajustes al Plan de SQA
Evaluación final de SQA	Informe final de SQA
Revisar la entrega semanal	Entrega semanal de SQA

### 3.3 Roles y responsabilidades

#### 3.3.1 Responsabilidades de la Organización del grupo de Desarrollo:

- Desarrollar un producto de software en base a lo definido en el SQAP y los contratos establecidos con el cliente.
- Generar la debida documentación definida en la SQAP acerca de cada una de sus actividades con el fin de llevar un control de las mismas.
- Entregar la documentación de desarrollo que se exige en el plan (SQAP).

#### 3.3.2 Responsabilidades de la Organización del Consultor:

- Establecer normas definidas por el cliente.
- Supervisar el desarrollo del software.
- Hacer cumplir a cabalidad las cláusulas del contrato.

### 3.3.3 Responsabilidades de la Organización del cliente:

- Proveer la información necesaria para el desarrollo del software con el fin de satisfacer sus necesidades.
- Brindar los recursos y condiciones necesarias para elaborar el software.
- Participar activamente en la organización del SQA para obtener óptimos resultados.

### 3.3.4 Responsabilidades de la Organización del grupo SQA

- Garantizar la calidad del producto de software desarrollado.
- Implantar normas y actividades para el desarrollo del software.
- Realizar reuniones para resolver los posibles conflictos durante el desarrollo del software.
- Aprobar y publicar el SQAP.
- Observar las deficiencias en el SQAP.
- Mejorar el SQAP, recomendando modificaciones o correcciones con el fin de obtener resultados óptimos.
- Autorizar la implantación del software.
- Enfoque de gestión de calidad.
- Tecnologías (métodos y herramientas).
- Revisiones Técnicas Formales.
- Estrategia de pruebas.
- Control de la documentación y de cambios.
- Procedimientos que aseguren ajustes a los estándares.
- Mecanismos de medición y generación de informes.

### 3.4 Recursos estimados de aseguramiento de calidad

El esfuerzo requerido para las actividades de SQA asciende a aproximadamente el 25~30% del esfuerzo total requerido en el proyecto.

Para llevar a cabo el plan de aseguramiento de calidad, se hace necesario contar con un grupo especializado de SQA, quienes desempeñarán distintas funciones para el cumplimiento de los objetivos.

Las actividades de SQA requieren de aproximadamente el 25~30% del tiempo total requerido en el proyecto.

SQA necesita tener acceso a equipos de trabajo que le permitan realizar sus labores de evaluación y auditoría eficazmente. Por ello dispondrá de la infraestructura y equipamiento destinados en el plan de proyecto:

Hardware	Cantidad	Precio Unitario (USD)	Total (USD)
Workstation Precision 5820	6	1430	8580
Impresora HP LaserJet Pro	1	320	320
Monitor Dell 24" E2417H	12	130	1560
UPS Forza SL-1011UL	4	100	400
Proyector Epson 1266	1	650	650
SmartPhone Samsung A30	8	230	1840
Altavoces Logitech Z313	1	58	58
Switch 8-Port T2500G-10TS	1	100	100
Cable RJ45	10	2	20
			13.528

## 4 Documentación

### 4.1 Propósito

Identificar toda la documentación que gobernará el desarrollo, validación y verificación, mantenimiento y uso del software.

### 4.2 Documentación mínima requerida

La documentación mínima que exige IEEE 730 para garantizar que la implementación del software satisface los requisitos es la siguiente:

- Especificación de Requisitos de Software
- Descripción del Diseño del Software
- Plan de Verificación y Validación
- Informe de Resultados de Verificación y Validación
- Documentación de Usuario

#### 4.2.1 Especificación de requerimientos del software

Esta documentación está basada en el contenido definido por IEEE 860 “Guide for Software Requirements Specifications”.

Las especificaciones de requerimientos del software describir claramente y de forma precisa cada uno de los requerimientos del Software, tal como: funciones, rendimiento, restricciones de diseño y atributos.

1. INTRODUCCION
  - 1.1 Objetivo
  - 1.2 Alcance
  - 1.3 Definiciones, acrónimos y abreviaciones
  - 1.4 Referencias
  - 1.5 Revisión
2. DESCRIPCION GENERAL
  - 2.1 Perspectiva del producto
  - 2.2 Funciones del producto
  - 2.3 Características de los usuarios

- 2.4 Restricciones generales
- 2.5 Asunciones y dependencias
- 3. ESPECIFICACION DE REQUERIMIENTOS
  - 3.1 Requerimiento Funcional
    - 3.1.1. Introducción
    - 3.1.2. Entradas
    - 3.1.3. Procesos
    - 3.1.4. Salidas
    - 3.1.5. Interfaces externas
      - 3.1.5.1. Interfaces del usuario
      - 3.1.5.2. Interfaces del hardware
      - 3.1.5.3. Interfaces del software
    - 3.1.6 Requerimientos de rendimiento
    - 3.1.7 Representación del diseño
    - 3.1.8 Cumplimientos con estándares
    - 3.1.9 Limitaciones del hardware
    - 3.1.10 Atributos
      - 3.1.10.1. Disponibilidad
      - 3.1.10.2. Seguridad
      - 3.1.10.3. Mantenibilidad
      - 3.1.10.4. Transferencia / Conversión
      - 3.1.10.5 Prevenciones
    - 3.1.11 Otros requerimientos
      - 3.1.11.1. Base de datos
      - 3.1.11.2. Operaciones
      - 3.1.11.3. Adaptaciones
- 4. APENDICES
- 5. INDICE
- 6. ANEXOS

La especificación debe:

- Ser completa:
- Externa, respecto al alcance acordado.
- Internamente, no deben existir elementos sin especificar.
- Ser consistente, no puede haber elementos contradictorios.
- Ser no ambigua, todo término referido al área de aplicación debe estar definido en un glosario.
- Ser verificable, debe ser posible verificar siguiendo un método definido, si el producto final cumple o no con cada requerimiento.
- Estar acompañada de un detalle de los procedimientos adecuados para verificar si el producto cumple o no con los requerimientos.
- Incluir requerimientos de calidad del producto a construir.

#### 4.2.2 Descripción del diseño del software

Esta documentación está basada en el contenido definido por IEEE 1016 “Recommended Practice for Software Design Descriptions”

Vista de Diseño	Alcance	Atributos de Entidad	Ejemplos de Representaciones
Descripción de descomposición	Partición del sistema dentro de entidades de diseño.	Identificación, tipo, objetivo, función, subordinación.	Diagrama de descomposición, jerarquía y lenguaje natural.
Descripción de dependencia	Descripción de las relaciones entre entidades y recursos del sistema.	Identificación, tipo, objetivo, dependencias, recursos.	Diagrama de estructura.
Descripción de interfaces	Lista de cada interfaz de diseñador, programador, o pruebas necesarias para conocer el uso de la entidad de diseño que componen el sistema.	Identificación, función, interfaces.	Tablas de parámetros.
Descripción de detalle	Descripción de los detalles de diseño internos en una entidad.	Identificación, procesamiento, datos.	Diagrama de flujos.

#### 4.2.3 Plan de verificación y validación

En el proceso de verificación se evalúa si los requerimientos cumplen con las características establecidas durante todo el ciclo de vida y sus procesos, además revisa el cumplimiento de estándares y define criterio para determinar el éxito de cada actividad durante todo el ciclo de vida del software.

Los procesos V&V apoyan a los 6 principales procesos del estándar ISO/IEC 12207:

- Administración
- Adquisición
- Suministro
- Desarrollo
- Operación
- Mantenimiento

#### **Procesos de Administración**

- La actividad de administración de V&V monitorea y evalúa todas las salidas V&V.
- Cuando existe un cambio, la administración de V&V identifica el requerimiento afectado y planea las tareas de V&V para añadir el cambio.
- Se analiza los riesgos y peligros introducidos en el software o sistema. Posteriormente se revisa el plan de tareas de V&V añadiendo nuevas tareas o cambiando el alcance o cambiar la intensidad de las tareas existentes de V&V si el nivel de integridad, peligro o riesgo cambio.

#### **Proceso de Adquisición**

El proceso de adquisición contiene las actividades y las tareas del adquisidor. El proceso comienza con la identificación de la necesidad de adquirir un sistema, un producto software o un servicio software. El proceso continúa con la preparación y publicación de una solicitud de propuestas, la selección de un proveedor y la gestión del proceso de adquisición hasta la aceptación del sistema, del producto software o del servicio software.



Incluye las actividades de:

- Inicio
- Preparación para la solicitud de propuestas.
- Preparación y actualización del contrato.
- Seguimiento del proveedor.
- Aceptación y finalización.

### **Proceso de Suministro**

El proceso de suministro se inicia como una propuesta en respuesta a la solicitud de una adquirente, o bien, mediante la negociación de un contrato con el adquirente para proporcionar un sistema, producto de software o servicio de software.

Este proceso continúa con la determinación de los procedimientos y recursos necesarios para administrar el proyecto, incluido el desarrollo de planes de proyecto y la ejecución de los planes mediante la entrega del sistema al adquirente.

El esfuerzo de V & V de suministro utiliza los productos del proceso de suministro para confirmar que los requisitos son consistentes y satisfacen las necesidades del usuario antes de que se finalice el contrato.

### **Proceso de Desarrollo**

Este proceso comprende las siguientes actividades:

- |                              |                                 |
|------------------------------|---------------------------------|
| • Análisis de Requerimientos | • V&V de Concepto               |
| • Diseño                     | • V&V de Requerimientos         |
| • Codificación               | • V&V de Diseño                 |
| • Integración                | • V&V de Implementación         |
| • Pruebas                    | • V&V de Pruebas                |
| • Instalación y soporte      | • V&V de Instalación y revisión |

## **Proceso de Operación**

El proceso de operación implica el uso del sistema de software por parte del usuario final en un entorno operativo.

La actividad de V & V de operación evalúa el impacto de los cambios en el funcionamiento, el efecto en el sistema ante cualquier cambio propuesto y los procedimientos operativos para cumplir con el uso previsto, además analiza los riesgos que afectan al usuario y al sistema.

El objetivo de la V&V de operación es:

- Evaluar nuevas restricciones en el sistema.
- Evaluar los cambios propuestos en el sistema y su impacto en el software.

Evaluar los procedimientos operativos para verificar su corrección y usabilidad.

## **Proceso de Mantenimiento**

El proceso de mantenimiento se activa cuando el sistema de software o la documentación asociada debe cambiarse en respuesta a una necesidad de mantenimiento del sistema.

La actividad de mantenimiento de V & V aborda:

- Modificación del sistema
- Migración del sistema
- Retiro de soporte (El reemplazo parcial o total por un nuevo sistema o la instalación de un sistema actualizado)

El objetivo de la V&V de mantenimiento es evaluar los cambios propuestos en el sistema de software y su impacto, evaluar las anomalías que se descubren durante la operación, evaluar los requisitos de migración, evaluar los requisitos de retiro de soporte y volver a realizar las tareas de V & V.

#### 4.2.4 Informe de Resultados de Verificación y Validación

En esta sección se enumerará los documentos, herramientas e informes que se crearán, por quien, para quién y cuándo serán liberados. Para cada entregable deberá indicar las fechas en que son liberadas todas las versiones del mismo.

Se realizarán los informes de resultados siguiendo los siguientes modelos:

##### Modelo de Casos de Prueba

Documento	Modelo de Casos de Prueba
Creado por	El Responsable de verificación, [nombre del responsable de verificación].
Para quien	Es la guía para realizar las pruebas del sistema y lo usarán los Asistentes de verificación y el Responsable de verificación cuando se ejecuten las pruebas del sistema.
Fecha de liberación	Será liberado el [fecha de primera liberación].

##### Informes de Verificación

Documento	Se genera un documento <b>Informe de Verificación Unitaria</b> por cada prueba unitaria que se realice al sistema.
Creado por	Las personas que ejecutan las pruebas.
Para quien	Es el retorno para los implementadores de la tarea de verificación, que detalla los errores encontrados para que puedan ser corregidos.
Fecha de liberación	Será liberado luego de cada verificación unitaria. [Indique la versión y la fecha de liberación de todas las versiones de este informe.]

Documento	Se genera un documento <b>Informe Consolidación</b> por cada consolidación que se realice al sistema.
Creado por	Las personas que ejecutan las pruebas.
Para quien	Es el retorno para los implementadores de la tarea de consolidación, que detalla los errores encontrados para que puedan ser corregidos.
Fecha de liberación	Será liberado luego de cada consolidación. [Indique la versión y la fecha de liberación de todas las versiones de este informe.]

Documento	Se genera un documento <b>Informe de Verificación de Integración</b> por cada prueba de integración que se realice al sistema.
Creado por	Las personas que ejecutan las pruebas.
Para quien	Es el retorno para los implementadores de la tarea de verificación, que detalla los errores encontrados para que puedan ser corregidos.
Fecha de liberación	Será liberado luego de cada verificación de integración. [Indique la versión y la fecha de liberación de todas las versiones de este informe.]

Documento	Se genera un documento <b>Informe de Verificación de Sistema</b> por cada prueba de sistema que se realice.
Creado por	Las personas que ejecutan las pruebas.
Para quien	Es el retorno para los implementadores de la tarea de verificación, que detalla los errores encontrados para que puedan ser corregidos.

Fecha de liberación	Será liberado luego de cada verificación de sistema. [Indique la fecha de liberación de este informe.]
---------------------	--

### Evaluación de la verificación

Documento	Se genera un documento <b>Evaluación de la verificación</b> por cada prueba que se realice al sistema. Este documento contiene las fallas encontradas en el sistema, la cobertura de la verificación realizada y el estado del sistema.
Creado por	El Responsable de verificación, que toma como fuente de su trabajo los Informes de verificación.
Para quien	Es el resumen de la tarea de verificación y es el retorno para todo el equipo de trabajo del estado del sistema.
Fecha de liberación	Será liberado luego de cada verificación, unitaria, de integración y de sistema. [Indique la versión y la fecha de liberación de todas las versiones de este informe.]

### Informe final de verificación

Documento	El documento <b>Informe final de verificación</b> es el resumen de la verificación final del sistema antes de que sea liberado al entorno del usuario.
Creado por	El Responsable de verificación, que toma como fuente de su trabajo los Informes de verificación.
Para quien	Indica el estado del sistema.
Fecha de liberación	Será liberado luego de la verificación final del sistema.

#### 4.2.5 Documentación de usuario

Esta descripción de documentación de usuario (UD) se basa en el estándar IEEE 1036 “Standard for Software User Documentation”.

La información especificada debe ser incluida en la documentación del usuario, esta documentación de usuario comprenderá de un conjunto. En cada documento se debe tomar en cuenta y describir los siguientes puntos.

Los documentos de usuario serán presentados en dos modos: instruccional y de referencia.

Los usuarios del software utilizarán los documentos ya sea para aprender acerca del software (modo instruccional) o para refrescar su memoria acerca del software (modo de referencia).

##### **Modo Instruccional**

Un modo instruccional de documento debe:

- Proveer el ambiente y la información necesaria para entender el sistema.
- Proveer la información necesaria para aprender lo que puede hacer con el software y como lo puede usar.
- Proveer ejemplos para reforzar el proceso de aprendizaje.

##### **Modo de Referencia**

Un documento de modo de referencia debe:

- Organizar y proveer información necesaria.
- Facilitar accesos aleatorios a la información.

Los documentos de modo de referencia que debe ser incluido son:

- a) Manual de comandos.
- b) Manual de mensajes de error.
- c) Manual de llamadas de programas.
- d) Guía de referencia rápida.
- e) Manual de Herramientas del software.
- f) Manual de utilitarios.

**Modelo a usar para el Documento de Usuario**

1. TITULO DE LA PAGINA
2. RESTRICCIONES
3. GARANTIAS Y OBLIGACIONES CONTRACTUALES
4. TABLA DE CONTENIDO
5. LISTA DE ILUSTRACIONES
6. INTRODUCCION
  - Descripción de audiencia
  - Declaración de aplicación
  - Declaración de objetivos
  - 6.4 Descripción del uso de documentos
  - 6.5 Documentos relacionados
  - 6.6 Convenciones
    - 6.6.1 Símbolos
    - 6.6.2 Convenciones de estilo
    - 6.6.3 Convenciones de sintaxis de comandos
  - 6.7 Instrucciones de reportes de problemas
7. CUERPO DEL DOCUMENTO
  - 7.1. Cuerpo del documento en modo instruccional
    - 7.1.1 Alcance
    - 7.1.2 Materiales
    - 7.1.3 Preparaciones
    - 7.1.4 Precauciones y prevenciones
    - 7.1.5 Métodos
    - 7.1.6 Información relacionada
  - 7.2. Cuerpo del documento en modo de referencia
    - 7.2.1 Objetivo
    - 7.2.2 Materiales
    - 7.2.3 Preparaciones
    - 7.2.4 Entradas
    - 7.2.5 Precauciones y prevenciones

7.2.6 Invocación

7.2.7 Operaciones de suspensión

7.2.8 Operaciones de terminación

7.2.9 Salidas

7.2.10 Condiciones de error

7.2.11 Información relacionada

8. MENSAJES DE ERROR, CONOCIMIENTO DE PROBLEMAS,  
RECUPERACION DE ERROR

9. ANEXOS

10. BIBLIOGRAFIA

11. GLOSARIO

#### 4.2.6 Plan de gestión de configuración del software (SCMP)

##### **Propósito**

El propósito está puesto en tener control sobre cada una de las iteraciones y de las fases de forma de identificar los componentes y las versiones que constituyen el producto de cada iteración, y rastrear los cambios y las razones que los justifican, al pasar de una iteración a otra. De esta forma se pretende lograr un desarrollo incremental en el que cada iteración se apoya sobre los resultados de la anterior en lugar de ser una construcción completamente nueva, lo que permite ganar en productividad, facilitar la verificación y generar un producto más estable.

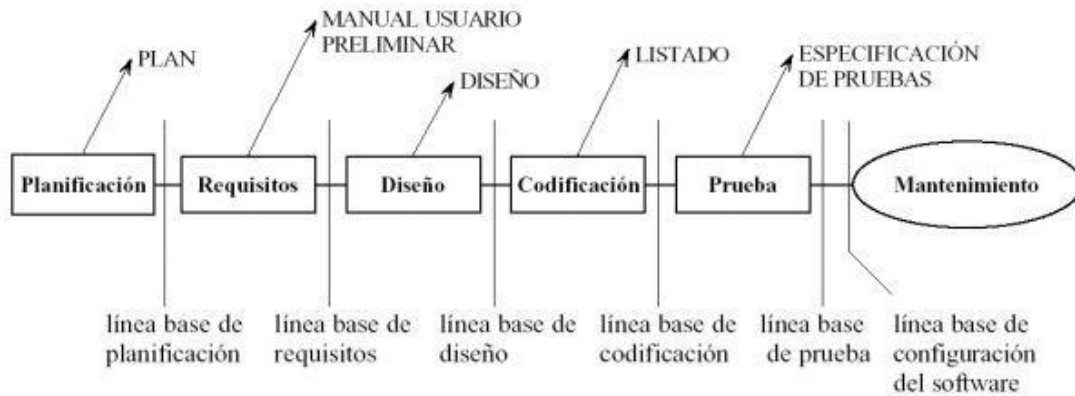
##### **Alcance**

En el Plan de Gestión de Configuración se han establecido, para cada uno de los procesos de desarrollo, los productos sobre los que se va a aplicar la gestión de configuración. Además, también se considera producto a mantener en el sistema de gestión de configuración el producto global resultante en cada proceso.



## Gestión de Configuración del Software

La gestión de configuración de software se especifica de la siguiente manera:



*Ilustración 2 Gestión de Configuración en el ciclo de vida del software*

## Roles y Responsabilidades

Roles	Responsabilidades
Gestor de configuración	<ul style="list-style-type: none"> <li>• Gestionar la planificación, identificación, control, seguimiento y auditoría de todos los elementos de configuración en la base de datos de configuración.</li> <li>• Desarrollar el plan de gestión de configuración.</li> <li>• Promover el uso efectivo de la base de datos de configuración dentro de la organización.</li> <li>• Monitorear y reportar los cambios no autorizados sobre los elementos de configuración.</li> <li>• Asegurar la consistencia e integridad de los datos de la base de datos de configuración a través de la ejecución de</li> </ul>

	<p>procedimientos de verificación y auditoría.</p> <ul style="list-style-type: none"> <li>• Liderar las actividades de evaluación del proceso: revisar tipos de elementos de configuración, relaciones, atributos y valores asociados, estructura de la base de datos, derechos de acceso.</li> <li>• Aprobar cambios estructurales en la base de datos de configuración.</li> </ul>
Coordinador de configuración	<ul style="list-style-type: none"> <li>• Asegurar que todos los elementos de configuración están registrados de forma adecuada en la base de datos de configuración.</li> <li>• Asegurar la consistencia e integridad de los datos de la base de datos de configuración y la estructura del sistema a través de la ejecución de procedimientos de verificación y auditoría.</li> <li>• Reportar cualquier discrepancia o no conformidad en los elementos de configuración al gestor de configuración. Participar en la mejora continua del proceso de gestión de configuración.</li> </ul>
Responsable de elementos de configuración	<ul style="list-style-type: none"> <li>• Asegurar que los elementos de configuración de los que es responsable están registrados en la base de datos de configuración con el estado y datos de configuración apropiados.</li> </ul>

	<ul style="list-style-type: none"> <li>• Verificar que los cambios sobre los elementos de configuración siguen el proceso de cambios definido.</li> <li>• Asegurar la idoneidad e integridad de los elementos de configuración de los que es responsable.</li> <li>• Trabajar conjuntamente con el gestor de configuración para identificar las causas de cualquier discrepancia identificada en las auditorías e implementar las acciones correctivas.</li> </ul>
Gestor de cambio	<ul style="list-style-type: none"> <li>• Evaluar el impacto y riesgo de los cambios. Asegurar que los responsables de los elementos de configuración actualizan los históricos de estos elementos con los cambios implementados.</li> </ul>

### Actividades de la Gestión de Configuración del Software

Actividad	Rol Responsable	Entradas	Salidas
Gestión del proceso de gestión de configuración.	Gestor de configuración.	Necesidades del proyecto. Plan de proyecto.	Plan de gestión de configuración aprobado.
Identificación de elementos de configuración.	Gestor de configuración.	Productos del proyecto.	Elementos de configuración identificados. Línea base. Estructura del directorio de gestión de configuración.
Mantenimiento y control de la gestión de configuración.	Responsable del elemento de configuración.	Peticiones de cambio.	Registro de solicitud de cambio. Solicitud de cambio aprobada. Línea base.

Informe de estado de la configuración.	Gestor de configuración.	Elementos de configuración.	Informe de estado de elementos de configuración.
Verificación y auditoría.	Gestor de configuración.	Registros de la gestión de configuración. Línea base. Registros de cambios.	Informe de auditoría de gestión de configuración.

## 5 Estándares, prácticas, convenciones y métricas

### 5.1 Estándares de documentación

#### Documentación en General

Para la elaboración del documento se han definido plantillas para todos los documentos a realizar. En ellos se definen:

- Fuente: Arial - Tamaño: 12 - Color: Negro - Estilos: Normal, Titulo1, Titulo2, Titulo3, Titulo4 hasta el nivel que sea necesario.
- Cada documento debe contar con una carátula al principio que debe contener:
  - Título explicativo del contenido del documento
  - Índice del contenido del documento y por consiguiente todas las páginas deben estar numeradas.
- Es deseable, que incluya al comienzo cual es el objetivo del documento.

#### Documentación Técnica

Por documentación Técnica se entiende Modelo de Dominio, Diagrama de Casos de Uso, Diagramas de Sistema en general.

La documentación técnica del producto debe:

- Ser adecuada para que un grupo independiente del de desarrollo pueda encarar el mantenimiento del producto.
- Incluir fuentes de los modelos.

## 5.2 Estándares de codificación

El proyecto realizado en PHP deberá seguir lo establecido por “PHP Framework Interop Group (PHP-FIG)” en los documentos “PHP Standard Recommendations (PSR)”.

- PSR-1: “Basic Coding Standard”
- PSR-12: “Extended Coding Style”

## 5.3 Estándares de comentarios

Se usará la siguiente norma para realizar comentarios dentro del código:

- Un comentario debe explicar porque se realiza alguna acción.
- Los comentarios dentro de un módulo deben estar separados del código.
- Utilizar comentarios de más de una línea para realizar descripciones, y comentarios de una línea para realizar especificaciones.
- Utilizar {} para comentarios de bloque
- Utilizar // para comentarios de línea

## 5.4 Estándares de verificación y prácticas

Se utilizarán las prácticas definidas en el documento "Plan de Verificación y Validación" basado en el estándar IEEE 1012 “Standard for Software Verification and Validation Plans”

## 5.5 Métricas

Se espera que los siguientes indicadores se cumplan en un cien por ciento, de lo contrario, se solicitará acciones correctivas para alcanzar el nivel de satisfacción deseado.

**Métrica de Adecuación a los estándares de documentación**

Sea N la cantidad total de identificadores definidos en la elaboración de documentos y llamemos C a la cantidad de identificadores que cumplen con los estándares definidos en el punto 5.1 del documento.

Entonces,

$$\text{Porcentaje de Adecuación a los estándares de documentación} = (C / N) * 100.$$

**Métrica de adecuación a los estándares de codificación**

Sea N la cantidad total de identificadores definidos en el desarrollo del software y llamemos C a la cantidad de identificadores que cumplen con los estándares definidos en el punto 5.2 del documento.

Entonces,

$$\text{Porcentaje de Adecuación a los estándares de codificación} = (C / N) * 100.$$

**Métrica del Cubrimiento de las Pruebas**

Se realizarán pruebas Unitarias y de Sistema por lo que este punto tiene el objetivo de cuantificar la cantidad de pruebas realizadas sobre el total, lo cual nos sugiere una noción de verificación del sistema.

Sea N la cantidad total de pruebas a desarrollar y llamemos C a la cantidad de pruebas efectivamente realizadas.

Entonces,

$$\text{Porcentaje de Cubrimiento de las Pruebas} = (C / N) * 100$$

## 6 Revisiones y Auditorias

### 6.1 Propósito

Los responsables de estas revisiones es la organización del SQA, con la participación de todo elemento de la organización que tengan que ver con los requerimientos, tales como: los diseñadores del software, agentes de pruebas.

Las revisiones y auditorias de los resultados del desarrollo se realizan a medida que se terminan cada una de las fases del ciclo de vida de desarrollo de software, con el fin de:

- Conocer el progreso alcanzado en el desarrollo.
- Evaluar el ajuste a los requerimientos del sistema.
- Evaluar la eficiencia en el trabajo.

### 6.2 Requerimientos mínimos

Cómo mínimo deberán revisarse todas las entregas semanales, basado en los estándares definidos anteriormente. Estas revisiones serán realizadas por el Responsable de SQA y/o el Asistente de SQA.

#### 6.2.1 Revisión de requerimientos

Esta revisión se realiza para:

- Evaluar las especificaciones de requerimientos del software (SRS).
- Asegurar que los requerimientos establecidos en la SRS, sean los correctos y estén completos.
- Garantizar la calidad, viabilidad e integridad de los requerimientos establecidos.

### 6.2.2 Revisión de diseño preliminar

La Revisión de diseño preliminar es realizada para evaluar la suficiencia técnica del DDS preliminar, antes de comenzar con el diseño detallado, define los siguientes puntos:

- Evaluar el progreso, consistencia y suficiencia técnica del alcance de diseño con los requerimientos funcionales de la ERS.
- Verificar la existencia y compatibilidad de las interfaces entre el software, el hardware y los usuarios finales.
- Determinar un diseño de software que cumpla con los requerimientos.

Para la Revisión de diseño preliminar se toman como requerimientos de revisiones los siguientes puntos:

- Revisar que se detallen todas las interfaces con otro software, sistemas de comunicación, etc.
- Para una adecuada identificación de interfaces y de un diseño óptimo.
- Revisar que exista un análisis del diseño para verificar la compatibilidad con los requerimientos críticos.
- Revisar que se establece los requerimientos del factor humano.

Se toman como requerimientos de revisiones los siguientes puntos:

- Revisar que se detallen todas las interfaces con otro software, sistemas de comunicación, etc.
- Para una adecuada identificación de interfaces y de un diseño óptimo.
- Revisar que exista un análisis del diseño para verificar la compatibilidad con los requerimientos críticos.
- Revisar que se establece los requerimientos del factor humano.



### 6.2.3 Revisión de diseño crítico

La Revisión de diseño crítico es generada para determinar la aceptabilidad de cómo la Descripción de diseño de software cumple con la Revisión de los Requisitos de Software. Evalúa la suficiencia técnica, integridad del diseño detallado del software, antes de comenzar a codificar para establecer que el diseño detallado satisface los requerimientos de la especificación de requerimientos del software.

Para la Revisión de diseño crítico se toman como requerimientos de revisión los siguientes puntos:

- Evaluar la compatibilidad del diseño detallado con la Revisión de los Requisitos de Software.
- Examinar la representación de datos en forma de diagramas lógicos, algoritmos, almacenamiento y representación de datos.
- Determinar la compatibilidad e integridad de requerimientos de interfaces.
- Establecer que todas las interfaces internas y externas incluyendo interacciones con la base de datos sean expresadas.

### 6.2.4 Revisión de plan de verificación y validación

Esta revisión se realiza para asegurar la consistencia y completitud de los métodos especificados en el Plan de Verificación & Validación.

### 6.2.5 Auditoría funcional

Esta auditoría se realiza previa a la liberación del software, para verificar que todos los requerimientos especificados en el documento de requerimientos fueron cumplidos.

La verificación funcional compara el código con los requerimientos documentados del software, como se estableció en SRS. Su propósito es asegurar que el código hace todo y solo lo que se indica en la documentación establecida por la SRS.

Se definen los siguientes puntos:

- Nomenclatura
- Número de identificación de la especificación
- Número de ítem de configuración
- La especificación de requerimientos de software
- Copia de código objeto
- Listado actualizado de ítems de configuración especificados
- El reporte de verificación y validación del software
- Listado del cumplimiento exitoso de pruebas funcionales
- Listado de todo lo planificado y pruebas que no fueron ejecutadas
- Actualizaciones para la documentación previamente liberada deberá ser revisada para asegurar su exactitud y consistencia.

#### 6.2.6 Auditoria física

Esta revisión se realiza para verificar que el software y la documentación son consistentes y están aptos para la liberación.

#### 6.2.7 Auditorías internas al proceso

Estas auditorías son para verificar la consistencia: del código versus el documento de diseño, especificaciones de interface, implementaciones de diseño versus requerimientos funcionales, requerimientos funcionales versus descripciones de testeo.

#### 6.2.8 Revisiones gerenciales

Estas revisiones se realizan periódicamente para asegurar la ejecución de todas las actividades identificadas en este Plan. Deben realizarse por una persona ajena al grupo de trabajo (en caso de que sea posible).

#### 6.2.9 Revisiones del plan de gestión de configuración

Esta revisión se realiza para asegurar la consistencia y completitud de los métodos especificados en el Plan de gestión de configuración.

#### 6.2.10 Revisión posterior a la implementación

Esta revisión se lleva a cabo al final del proyecto para evaluar las actividades de desarrollo de ese proyecto y proporcionar recomendaciones para las acciones apropiadas.

### 7 Pruebas

Esta sección identificará todas las pruebas no incluidas en el plan de verificación y validación del software para el software cubierto por el SQAP y deberá indicar los métodos que se utilizarán.

Se debe tener en cuenta, a la hora de definir un plan de prueba, que es lo que se va a probar, es decir describir el objeto de prueba y que es lo que se va a probar del mismo (una lista de las características y descripción del sistema, sus componentes tomados por separado, etc).

Se debe establecer qué estrategias se van a utilizar para las mismas (técnica, patrones y/o herramientas a utilizar). Por ejemplo, en el caso de pruebas unitarias de un procedimiento, esta sección podría indicar: «Se aplicará la estrategia caja-negra de fronteras de la precondition» o «Ejercicio de los caminos ciclomáticos válidos». En lo posible, la estrategia debe precisar el número mínimo de casos de prueba a diseñar; por ej. 100% de las fronteras, 60% de los caminos ciclomáticos, etc. La estrategia también explicita el grado de automatización que se exigirá, tanto para la generación de casos de prueba como para su ejecución.

Se deberán establecer las secuencias de actividades, como la definición de los tiempos para la preparación de las pruebas, la ejecución de las pruebas hasta el análisis de resultados de las pruebas en el contexto de las fases previstas del desarrollo. Además, se deberá disponer de toda la documentación necesaria, la plataforma de pruebas preparada (banco de pruebas) y la integridad del desarrollo de la funcionalidad requerida.

Se deberá categorizar la configuración, es decir explicitar las condiciones bajo las cuales, el plan debe ser:

- Suspendido.
- Repetido.
- Culminado.

En algunas circunstancias (las cuales deben ser explicitadas) el proceso de prueba debe suspenderse en vista de los defectos o fallas que se han detectado. Al corregirse los defectos, el proceso de prueba previsto por el plan puede continuar, pero debe explicitarse a partir de qué punto, ya que puede ser necesario repetir algunas pruebas. Los criterios de culminación pueden ser tan simples como aprobar el número mínimo de casos de prueba diseñados o tan complejos como tomar en cuenta no sólo el número mínimo, sino también el tiempo previsto para las pruebas y la tasa de detección de fallas.

Se tendrán que explicitar los documentos a entregarse al culminar el proceso previsto por el plan de prueba, por ejemplo: sub-planes, especificación de pruebas, casos de prueba, resumen gerencial del proceso y bitácora de pruebas.

Este plan de pruebas soporta los siguientes objetivos:

- Identificar los ítems a probar
- Describe, en términos generales, el enfoque de pruebas a ser usado
- Identifica los recursos requeridos y provee un estimado de sus esfuerzos
- Lista los entregables de las pruebas del proyecto
- Identificar los tipos de pruebas a utilizar en la ejecución de las mismas.
- Diseñar cada una de las pruebas de cada una de las iteraciones a probar.

## **Tipos de prueba**

### **Pruebas de la implementación y de unidad**

Será objeto de estas pruebas verificar la funcionalidad y estructura de cada componente individualmente, una vez que estos han sido codificados. Para poder llevar a cabo estas pruebas es preciso:

- Garantizar que el proceso de codificación, las revisiones asociadas y la prueba de unidad sean conducidos de acuerdo a los estándares y procedimientos establecidos en el plan de proyecto.
- Asegurar la incorporación de los resultados de las revisiones en los entregables de esta fase.
- Verificar la implementación de las acciones correctivas derivadas de la prueba de unidad.
- Comprobar la utilización de la especificación de procedimientos y casos de prueba durante la prueba de unidad.
- Corroborar la documentación del código y de los resultados de la prueba de unidad.

Objetivo de la Prueba:	Se focaliza en ejecutar cada módulo (o unidad mínima a ser probada, ej = una clase) lo que provee un mejor modo de manejar la integración de las unidades en componentes mayores. Busca asegurar que el código funciona de acuerdo con las especificaciones y que el módulo lógico es válido.
Responsable:	El encargado de diseñar y realizar esta prueba puede ser uno de los desarrolladores del sistema.
Pasos:	<ol style="list-style-type: none"> <li>1. Determinar el ambiente de prueba.</li> <li>2. Particionar el sistema en módulos de unidades lógicas fáciles de probar.</li> <li>3. Por cada unidad hay que definir los casos de prueba (pruebas de caja blanca).</li> <li>4. Para esto los casos de prueba deben diseñarse de forma tal que se recorran todos los caminos de ejecución posibles dentro del código bajo prueba; por lo tanto, el diseñador debe construirlos con acceso al código fuente de la unidad a probar.</li> <li>5. Los aspectos a considerar son los siguientes: Rutinas de excepción, Rutinas de error, Manejo de parámetros, Validaciones, Valores válidos, Valores límites, Rangos, Mensajes posibles.</li> </ol>
Técnica:	<ul style="list-style-type: none"> <li>• Comparar el resultado esperado con el resultado obtenido.</li> <li>• Si existen errores, reportarlos.</li> </ul>
Criterio de Completitud:	<ul style="list-style-type: none"> <li>• Todas las pruebas planeadas han sido ejecutadas.</li> <li>• Todos los defectos que se identificaron han sido tenidos en cuenta.</li> </ul>

### Pruebas de Stress

Será objeto de estas pruebas conocer y mitigar los riesgos relacionados con el mal desempeño de las aplicaciones en el entorno, y efectuar las correcciones necesarias.

Para poder llevar a cabo estas pruebas es preciso:

- Cuantificar la capacidad de la infraestructura.
- Validar los requerimientos de performance.
- Validar la escalabilidad de la plataforma.

Con la definición de estos criterios, se podrá conocer qué cantidad de clientes simultáneos soporta el producto desarrollado y si el hardware empleado es suficiente para soportar el nivel de transacciones propuesto. Además, se podrá evaluar qué expectativa de crecimiento soporta el producto.

### **Criterios de aceptación**

- No se acepta documentación con errores ortográficos.
- No se acepta documentación con un grado de completitud menor al 70% con respecto a los issues del producto.
- No se aceptan las pruebas unitarias que no cumplan los objetivos esperados.
- No se aceptan las pruebas de integración en las que alguna de las pruebas unitarias falle o como resultado final, no se cumpla la funcionalidad establecida en el requerimiento.
- No se aceptan las pruebas de regresión las cuales luego de corregir/agregar una funcionalidad arrojen un resultado distinto.

## 8 Reportes de Problemas y Acciones Correctivas

En esta sección se describen las prácticas y procedimientos que se van a utilizar para la notificación, seguimiento y resolución de problemas de software, así como las responsabilidades organizativas.

Se tiene el siguiente propósito:

- Asegurar que todos los problemas de documentan, se corrigen y no caen en el olvido.
- Asegurar que se evalúa la validez de los informes de problemas.
- Realimentar al desarrollador y el usuario sobre el estado de los problemas.
- Proporcionar datos para medir y predecir la calidad y fiabilidad del software.

Antes de la identificación de la presencia de un problema se debe buscar los posibles orígenes del mismo sin desechar ninguna de las posibilidades, para esto se debe tomar dos rutas para generar el reporte de problema completo y consistente.

- Registrar las causas sospechosas del origen del problema, esto asegura que no se descarta ninguna situación posible.
- Registrar causas adicionales
- Registrar causas plenamente identificadas, esto es, causas verificadas por detectores del problema
- Registrar otras causas externas o relacionadas a las identificadas anteriormente
- Generar el reporte del problema, este debe estar completamente detallado y de acuerdo a los puntos que contiene el mismo, este reporte debe ser entregado a la organización responsable por los problemas.
- La organización responsable por los problemas, convocan a una reunión técnica, en la cual participarán además de la organización responsable, los elementos de las organizaciones afectadas por el problema, quienes



describirán el problema y darán recomendaciones necesarias para solucionar el problema.

Cualquier problema en el producto de software que sea encontrado durante el ciclo de vida de desarrollo de software, debe ser reportado a través de un reporte en el cual se detalla la fecha de cuando fue encontrado el problema, una identificación preliminar del mismo, descripción, etc., este reporte debe ser firmado por los que identificaron el problema, debe ser entregado a la organización responsable de los problemas.

<b><u>REPORTE DE ANOMALÍAS</u></b>		<b>Pág.</b> .....								
FASE: .....										
# De Reporte: .....	Fecha: .. / .. / ..									
Lugar: .....	Hora: .....									
<p>a) Descripción y Ubicación</p> <p>.....</p> <p>.....</p> <p>b) Impacto</p> <p>.....</p> <p>.....</p> <p>c) Causa</p> <p>.....</p> <p>.....</p> <p>d) Criticidad</p> <p>.....</p> <p>.....</p> <p>d) Recomendaciones:</p> <p>.....</p> <p>.....</p>										
<div style="border: 1px solid black; padding: 10px; min-height: 100px;"> <p><b>Equipo de Trabajo:</b></p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center; vertical-align: top;"><b>Nombre:</b></td> <td style="width: 50%; text-align: center; vertical-align: top;"><b>Firma</b></td> </tr> <tr> <td style="text-align: center;">.....</td> <td></td> </tr> <tr> <td style="text-align: center;">.....</td> <td></td> </tr> <tr> <td style="text-align: center;">.....</td> <td></td> </tr> </table> </div>			<b>Nombre:</b>	<b>Firma</b>	.....		.....		.....	
<b>Nombre:</b>	<b>Firma</b>									
.....										
.....										
.....										

<b><u>REPORTE SUMARIO DE FASE DE VERIFICACIÓN Y VALIDACIÓN</u></b>		<b>Pág.</b> .....								
<b>FASE:</b> .....										
<b># De Reporte:</b> .....	<b>Fecha:</b> ..../.../.....									
<b>Lugar:</b> .....	<b>Hora:</b> .....									
<p><i>a) Descripción de las tareas de V&amp;V que se realizaron</i></p> <p>.....</p> <p>.....</p> <p><i>b) Sumario de resultados de las tareas:</i></p> <p>.....</p> <p>.....</p> <p><i>c) Sumario de anomalías y resolución:</i></p> <p>.....</p> <p>.....</p> <p><i>d) Evaluación de calidad del software:</i></p> <p>.....</p> <p>.....</p> <p><i>e) Recomendaciones:</i></p> <p>.....</p> <p>.....</p>										
<p><b>Equipo de Trabajo:</b></p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center; padding: 5px;"><b>Nombre:</b></td> <td style="width: 50%; text-align: center; padding: 5px;"><b>Firma</b></td> </tr> <tr> <td style="text-align: center; padding: 5px;">.....</td> <td style="text-align: center; padding: 5px;">.....</td> </tr> <tr> <td style="text-align: center; padding: 5px;">.....</td> <td style="text-align: center; padding: 5px;">.....</td> </tr> <tr> <td style="text-align: center; padding: 5px;">.....</td> <td style="text-align: center; padding: 5px;">.....</td> </tr> </table>			<b>Nombre:</b>	<b>Firma</b>	.....	.....	.....	.....	.....	.....
<b>Nombre:</b>	<b>Firma</b>									
.....	.....									
.....	.....									
.....	.....									

<b><u>REPORTE FINAL DE PROBLEMAS</u></b>		<b><i>Pág.</i></b> .....								
<b># De Reporte:</b> ..... <b>Lugar:</b> .....		<b>Fecha:</b> ..... / ..... / ..... <b>Hora:</b> .....								
<p><i>a) Identificación del problema:</i></p> <p>.....</p> <p>.....</p> <p><i>b) Descripción:</i></p> <p>.....</p> <p>.....</p> <p><i>c) El evento ejecutado cuando se presentó el problema es:</i></p> <p>.....</p> <p>.....</p> <p><i>f) Posibles orígenes del problema:</i></p> <p>.....</p> <p>.....</p>										
<div style="border: 1px solid black; padding: 10px;"> <p><b><i>Equipo de Trabajo:</i></b></p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center; vertical-align: top;"><b><i>Nombre:</i></b></td> <td style="width: 50%; text-align: center; vertical-align: top;"><b><i>Firma</i></b></td> </tr> <tr> <td>.....</td> <td>.....</td> </tr> <tr> <td>.....</td> <td>.....</td> </tr> <tr> <td>.....</td> <td>.....</td> </tr> </table> </div>			<b><i>Nombre:</i></b>	<b><i>Firma</i></b>	.....	.....	.....	.....	.....	.....
<b><i>Nombre:</i></b>	<b><i>Firma</i></b>									
.....	.....									
.....	.....									
.....	.....									

La organización responsable de los problemas del software, es la organización del SQA, comandada por la organización del consultor, estas organizaciones son las encargadas de determinar el cronograma, lugar y temario, para llevar a fijar la acción correctiva del problema.

La especificación de acciones correctivas generada en la reunión técnica, será entregada a los elementos organizacionales afectados por el problema para que estos implementen las acciones correctivas respectivas.

## 9 Herramientas, Técnicas y Metodologías

Para la realización de pruebas dentro de un proyecto web, realizado en PHP se utilizarán las siguientes herramientas:

- Bugzilla
- Atoum
- Behat
- Codeception
- OpenSTA

Las herramientas utilizadas son generalmente herramientas de prueba en donde una aplicación se ejecuta a través de una serie de pruebas para medir el rendimiento de la aplicación. Estas herramientas se emplean para probar la aplicación y producir números y estadísticas sobre la aplicación real. A través de estos números, el equipo de SQA y sus desarrolladores se sabe si la solicitud ha cumplido de acuerdo a los resultados específicos.

Las técnicas que se emplearán para el aseguramiento de la calidad son:

- Auditorias PPQA (Process and Product Quality Assurance)
- Pruebas de Validación
- Comparación de datos
- Revisión por Pares
- Revisión Técnica Formal

## 10 Control de Código

Se utilizará Git como el Sistema de Control de Versiones.

La rama principal o main, será accedida únicamente por el jefe del grupo de desarrollo quién controlará, verificará y validará todos los cambios realizados por los desarrolladores en sus propias ramas antes de modificar la rama principal.

Se seguirán las siguientes normas básicas:

- No se podrá hacer Commit si el proyecto no compila exitosamente.
- No se podrá hacer Commit si tiene errores de ejecución conocidos.
- No se podrá hacer Commit si algún documento de código fuente tiene implementaciones incompletas.
- Ningún documento de código puede tener más de 500 líneas, a menos que sea autogenerada.
- Ningún método puede tener más de 50 líneas de código a menos que sea autogenerado.

## 11 Control de Medios

En esta sección se definen los métodos y facilidades que se van a utilizar para proteger el medio físico de accesos no autorizados y daños y degradaciones inesperadas, y las organizaciones responsables para realizar este control.

La organización responsable por esta tarea es la organización de desarrollo, con la supervisión de la organización de la SQA.

Se debería asegurar que:

- Está garantizado el almacenamiento y recuperación de software.
- El software está accesible únicamente para aquellos que lo necesitan.
- Se controla el entorno para que no se degrade el medio físico en el que se almacena el software.
- Se almacenan copias del software crítico y del código en línea base fuera de las instalaciones de la organización.

### **Medio de Almacenamiento**

El medio del programa de computadora se define como aquellos medios sobre los cuales los datos son almacenados.

Se utilizarán los siguientes medios:

- Los discos duros como dispositivos primarios.
- Los CD's como almacenamiento secundario, para guardar las copias de seguridad.
- La documentación respectiva sobre el desarrollo de software (papel).

### **Proceso de copias de seguridad**

Las copias de seguridad serán realizadas a la finalización de cada sesión de trabajo, registrándose la fecha y hora de copia de seguridad.

### **Puntos de Control**

Para el acceso no autorizado, se debe asignar cuentas privilegiadas, cada usuario que interactúa con el software tendrá su propia cuenta de acuerdo al cargo que desempeñe.

Se cuenta con la integridad de la Base de Datos para la protección de los datos. Se realizará una revisión periódica del software con el fin de que funcione óptimamente.

## 12 Control de Proveedores

Esta sección deberá establecer las disposiciones para asegurar que el software proporcionado por los proveedores cumpla con los requisitos establecidos. Además, esta sección deberá establecer los métodos que se utilizarán para asegurar que el proveedor de software reciba los requisitos adecuados y completos. Para el software desarrollado previamente, esta sección deberá establecer los métodos que se utilizarán para garantizar la idoneidad del producto para su uso con los elementos de software cubiertos por el SQAP. Para el software que se va a desarrollar, se requerirá que el proveedor prepare e implemente un SQAP de acuerdo con este estándar. Esta sección también deberá establecer los métodos que se emplearán para asegurar que los proveedores cumplan con los requisitos de esta norma. Si el software se va a desarrollar bajo contrato, se describirán los procedimientos para la revisión y actualización del contrato.

## 13 Recolección de Documentos, mantenimiento y almacenado

Las organizaciones responsables por las tareas de esta sección, es la organización del consultor, en coordinación con la organización de la SQA.

Se identifica aquella documentación que se debe retener, y se especifican los métodos y facilidades que se utilizarán para recolectar, proteger y mantener esta documentación.

También se especificará el período de retención para cada tipo de registro.

Los documentos que son requeridos son los siguientes:

- Plan de garantía de calidad del software.
- Especificación de requerimientos del software.
- Descripción del diseño del software.
- Plan de verificación y validación del software.
- Documentación del usuario.



El mantenimiento de los registros del software, será realizado por versiones sucesivas de actualizaciones de las mismas, para esto se lleva un registro de actualizaciones de la documentación.

Los documentos verificados y validados, deben ser documentados en libros impresos, con tres copias de cada documento y almacenado en lugares diferentes y ambiente adecuados.

La retención de registros se realizará en cada finalización de las fases del ciclo de vida de desarrollo de software y según los puntos de verificación y validación.

## 14 Capacitación

El presente documento describe los objetivos relacionados con la capacitación y las actividades que se realizan para alcanzarlos, teniendo presente que el primer propósito de la capacitación es desarrollar y potenciar las capacidades y el conocimiento de los individuos para expandir las áreas de experticia de la institución al desarrollo, la implementación y el soporte de los sistemas de información; y para obtener profesionales de excelencia que ejecuten sus roles más eficiente y eficazmente.

### 14.1 Objetivos organizacionales

La actividad de capacitación es realizada para alcanzar los siguientes objetivos organizacionales:

- Producir software de calidad eficientemente.
- Introducir rápidamente las prácticas, metodologías y tecnologías, relacionadas con el proceso de SQA, en el proceso de desarrollo.
- Promover el mejoramiento continuo del proceso de desarrollo.

## 14.2 Objetivos específicos

### **Grupo de SQA**

- Difundir la importancia y los beneficios de SQA.
- Conocer con detenimiento todas las áreas de prácticas relacionadas con SQA.
- Comprender el proceso de desarrollo.
- Definir los recursos y la infraestructura apropiada para la implementación del proceso de SQA.
- Apoyar y dirigir la definición y el mejoramiento del proceso y las prácticas de SQA.
- Documentar el proceso de SQA.
- Dirigir el proceso de implantación de las prácticas de SQA.
- Respalda el soporte de las actividades de SQA, especialmente aquellas relacionadas con la capacitación.
- Asegurar la capacitación del personal del grupo de SQA.

### **Nivel de gestión**

- Comprender los conceptos fundamentales asociados a las prácticas de SQA.
- Establecer formalmente y fortalecer la política para SQA.
- Establecer e incrementar la infraestructura requerida para las prácticas asociadas a SQA.

### **Gerente técnico**

- Difundir y apoyar la implementación de la política para SQA.
- Comprender los conceptos fundamentales asociados a las prácticas de SQA.
- Monitorear el cumplimiento de las prácticas de SQA en los diferentes proyectos de desarrollo.
- Establecer, implementar y monitorear nuevas iniciativas con relación a las prácticas de SQA y al mejoramiento continuo de este proceso.

### ***Jefes de proyectos***

- Difundir y apoyar la implementación de la política para SQA.
- Comprender los conceptos fundamentales asociados a las prácticas de SQA.
- Planificar las actividades relacionadas con las prácticas de SQA e incluirlas en el plan de proyecto.
- Establecer y procurar los recursos necesarios para las actividades de SQA.
- Establecer los mecanismos de control y seguimiento para las actividades de SQA.
- Gestionar y apoyar las actividades de SQA.
- Guiar al equipo desarrollador en la implementación adecuada de las prácticas, metodologías y tecnologías asociadas al proceso de SQA.

### ***Desarrolladores***

- Comprender la importancia del proceso de SQA.
- Comprender los conceptos fundamentales asociados a las prácticas de SQA.
- Participar y apoyar las actividades de SQA.
- Participar en la implementación de mejoras a las actividades de SQA.

### 14.3 Cursos impartidos

<b><i>Curso</i></b>	
<b><i>Propósito</i></b>	
<b><i>Objetivos</i></b>	
<b><i>Audiencia</i></b>	
<b><i>Prerrequisitos</i></b>	
<b><i>Temario</i></b>	
<b><i>Duración</i></b>	
<b><i>Instructores</i></b>	

Como mínimo deben incluirse los siguientes cursos:

- Introducción al proceso de SQA.
- Introducción a las prácticas implantadas de SQA.
- Cursos sobre prácticas específicas de SQA (actividades, procedimientos, guías, uso de informes, roles y responsabilidades)
- Herramientas específicas para el proceso de SQA y sus prácticas

#### 14.4 Desarrollo de cursos de capacitación

##### **Especificación de necesidades**

- Analizar necesidades: identificarlas, priorizarlas y proponer soluciones.
- Determinar la audiencia: roles y responsabilidades, nivel educacional, experiencia y conocimientos.
- Fijar los objetivos educacionales y contrastarlos con los objetivos de la capacitación.
- Seleccionar el tipo de presentación y los materiales.
- Identificar restricciones.
- Establecer las actividades y los recursos asociados.
- Definir el temario.
- Describir el curso.

##### **Diseño del curso de capacitación**

- Diseñar los puntos contenidos en el temario.
- Incorporar el conocimiento de los expertos en el tema.
- Determinar el o los instructores.
- Desarrollar los instructivos y el material de apoyo.
- Planificar la capacitación.
- Re-estimar los costos y recursos.

**Curso de capacitación**

- Conducir el curso.
- Evaluar el curso.

**Revisión del curso**

- Analizar la evaluación del curso.
- Tras el análisis, incorporar las observaciones y mejoras propuestas.
- Una vez aprobado, incorporarlo a los cursos impartidos.

## 15 Gestión de riesgos

Identificar Riesgo	Prob.	Impacto	Reducir prob. presencia	Reducir impacto
R1. Aumento del costo del proyecto	70	Significativo	<ul style="list-style-type: none"> <li>• Verificar la estimación de costo</li> <li>• Evitar gastos innecesarios</li> </ul>	<ul style="list-style-type: none"> <li>• Tener un fondo de respaldo</li> </ul>
R2. No terminar el proyecto en la fecha prevista	80	Significativo	<ul style="list-style-type: none"> <li>• Realizar planificación de los hitos del proyecto</li> <li>• Realizar seguimiento y control a lo planeado</li> </ul>	<ul style="list-style-type: none"> <li>• Incremento en horario de trabajo</li> <li>• Procurar flexibilidad de fecha de entrega con el cliente</li> </ul>
R3. Disminución de calidad	50	Medio	<ul style="list-style-type: none"> <li>• Definición correcta de los requerimientos</li> <li>• Realizar seguimiento y control a lo planeado</li> <li>• Transferir correctamente la idea de lo que se quiere desarrollar</li> </ul>	<ul style="list-style-type: none"> <li>• Planificar mantenimientos luego de la entrega del producto</li> </ul>
R4. Pérdida de RRHH	65	Significativo	<ul style="list-style-type: none"> <li>• Contrato</li> <li>• Aumentar motivación</li> </ul>	<ul style="list-style-type: none"> <li>• Adoptar un estándar de codificación</li> <li>• Tener vacantes disponibles</li> <li>• Utilizar herramientas de desarrollo sencillas</li> </ul>
R5. Sufrir daños personales o a la propiedad	30	Medio	<ul style="list-style-type: none"> <li>• Contar con un sistema de seguridad confiable</li> <li>• Evitar acciones imprudentes.</li> </ul>	<ul style="list-style-type: none"> <li>• Asegurar los equipos y al personal</li> </ul>
R6. Comprensión incorrecta de los requerimientos	55	Significativo	<ul style="list-style-type: none"> <li>• Realizar entrevistas planificadas con el cliente</li> </ul>	<ul style="list-style-type: none"> <li>• Planificar mantenimientos</li> <li>• Asegurar la escalabilidad y</li> </ul>

			<ul style="list-style-type: none"> <li>• Presentación de prototipos del producto al cliente previo al desarrollo</li> </ul>	flexibilidad en el código
R7. Problemas con el Hardware	60	Medio	<ul style="list-style-type: none"> <li>• Mantener al día los mantenimientos</li> <li>• Comprar productos de calidad</li> </ul>	<ul style="list-style-type: none"> <li>• Tener equipos de repuesto</li> <li>• Tener un fondo de emergencia</li> </ul>
R8. Falta de experiencia con las herramientas software	45	Medio	<ul style="list-style-type: none"> <li>• Realizar capacitaciones</li> <li>• Utilizar herramientas comunes de fácil comprensión</li> </ul>	<ul style="list-style-type: none"> <li>• Tener herramientas alternas a la solicitada</li> </ul>
R9. Los cambios en el desarrollo no son controlados	70	Medio	<ul style="list-style-type: none"> <li>• Utilizar una herramienta de control y versionamiento</li> <li>• Usar roles bien definidos</li> </ul>	<ul style="list-style-type: none"> <li>• Usar backups o puntos de restauración</li> </ul>
R10. El cliente no define bien sus requerimientos	70	Significativo	<ul style="list-style-type: none"> <li>• Utilizar preguntas claves y planificadas en las entrevistas</li> <li>• Asesoramiento profesional para el cliente</li> </ul>	<ul style="list-style-type: none"> <li>• Incluir el alcance definido en el contrato con el cliente</li> </ul>
R11. Poca comunicación en el equipo de desarrollo	70	Significativo	<ul style="list-style-type: none"> <li>• Realizar actividades de integración social</li> </ul>	<ul style="list-style-type: none"> <li>• Definir fechas para rendición de cuentas</li> </ul>
R12. Poco rendimiento del equipo de desarrollo	60	Significativo	<ul style="list-style-type: none"> <li>• Realizar capacitaciones</li> <li>• Realizar seguimiento y control a lo planeado</li> <li>• Incrementar motivación</li> </ul>	<ul style="list-style-type: none"> <li>• Tener personal extra o de respaldo</li> </ul>

# UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO



## “MODELOS DE DESARROLLO DEL SOFTWARE”

NOMBRES:

- ABASTO BERBETTY, RODRIGO
- POZADAS MAMANI, INGRID NATHALY

GRUPO: AJ

MATERIA: INGENIERÍA DE SOFTWARE II

FECHA: 16/MAR/2021

SANTA CRUZ - BOLIVIA



## Contenido

Diagrama de Casos de Uso .....	2
Diseño de Arquitectura .....	4
Arquitectura Lógica .....	4
Arquitectura Física .....	6
Modelo de Dominio.....	7

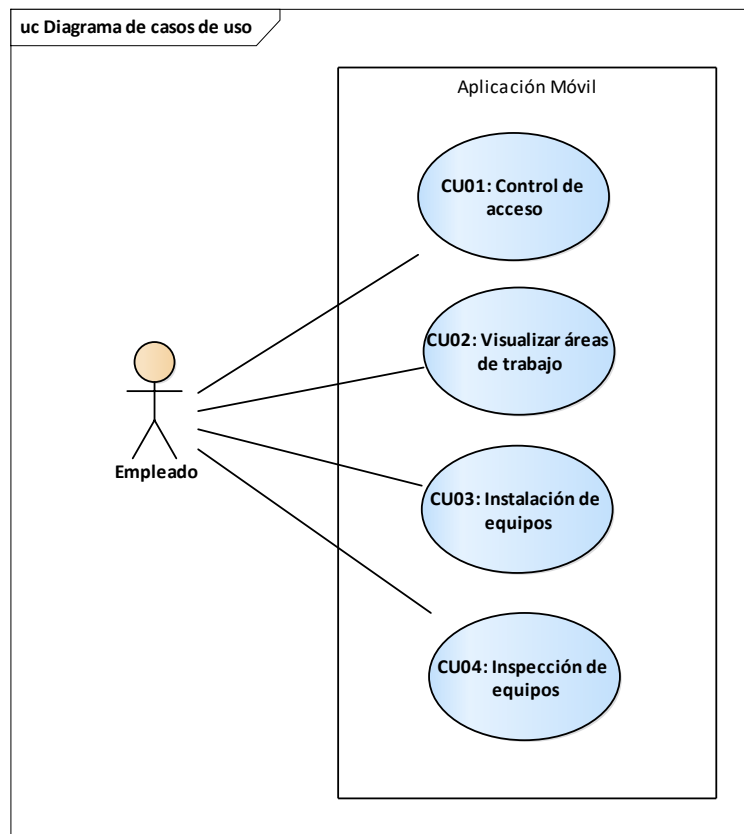
## Diagrama de Casos de Uso

Figura 1

*Diagrama de Casos de Uso de la aplicación web*

**Figura 2**

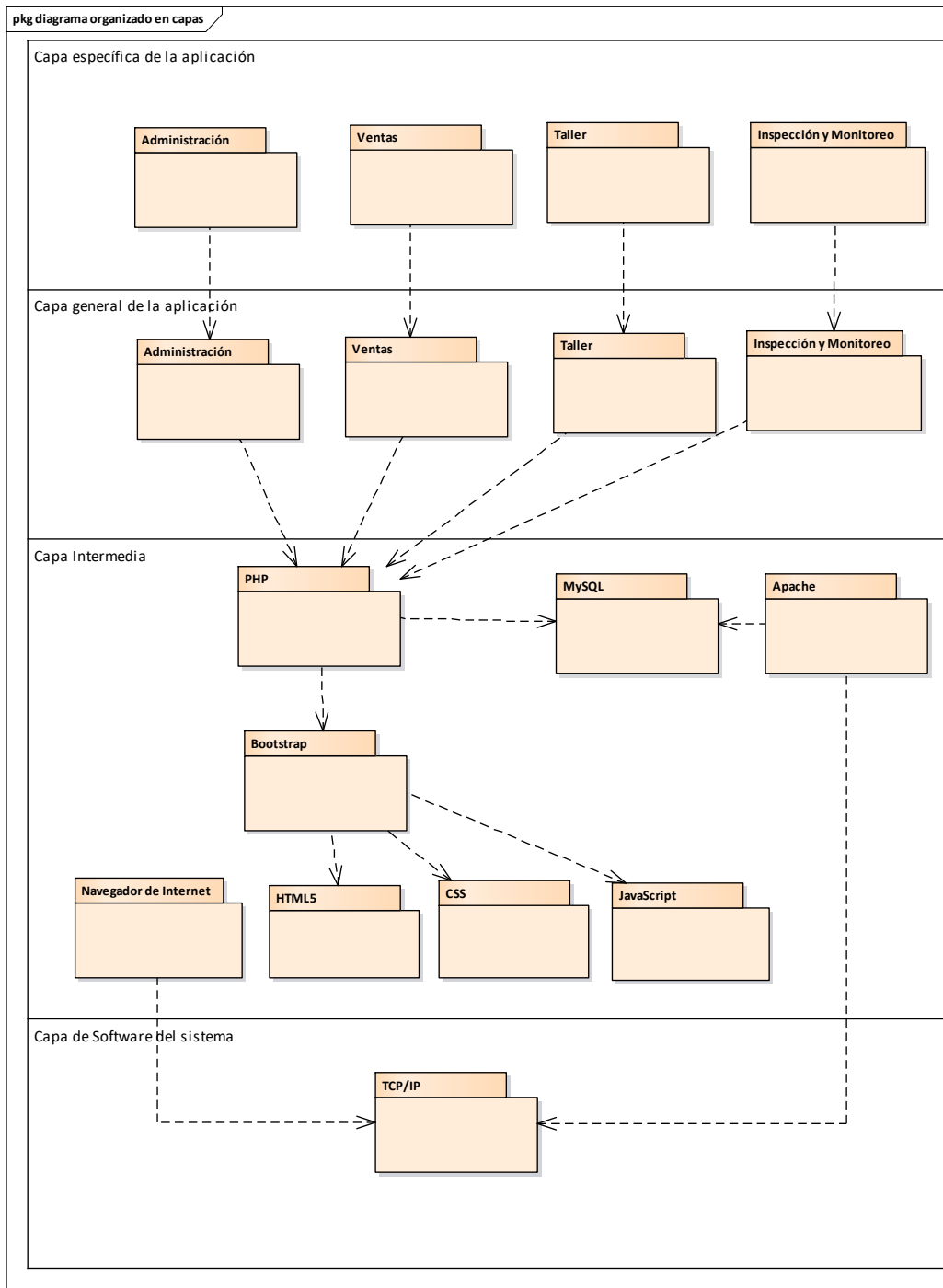
*Diagrama de Casos de Uso de la aplicación móvil*

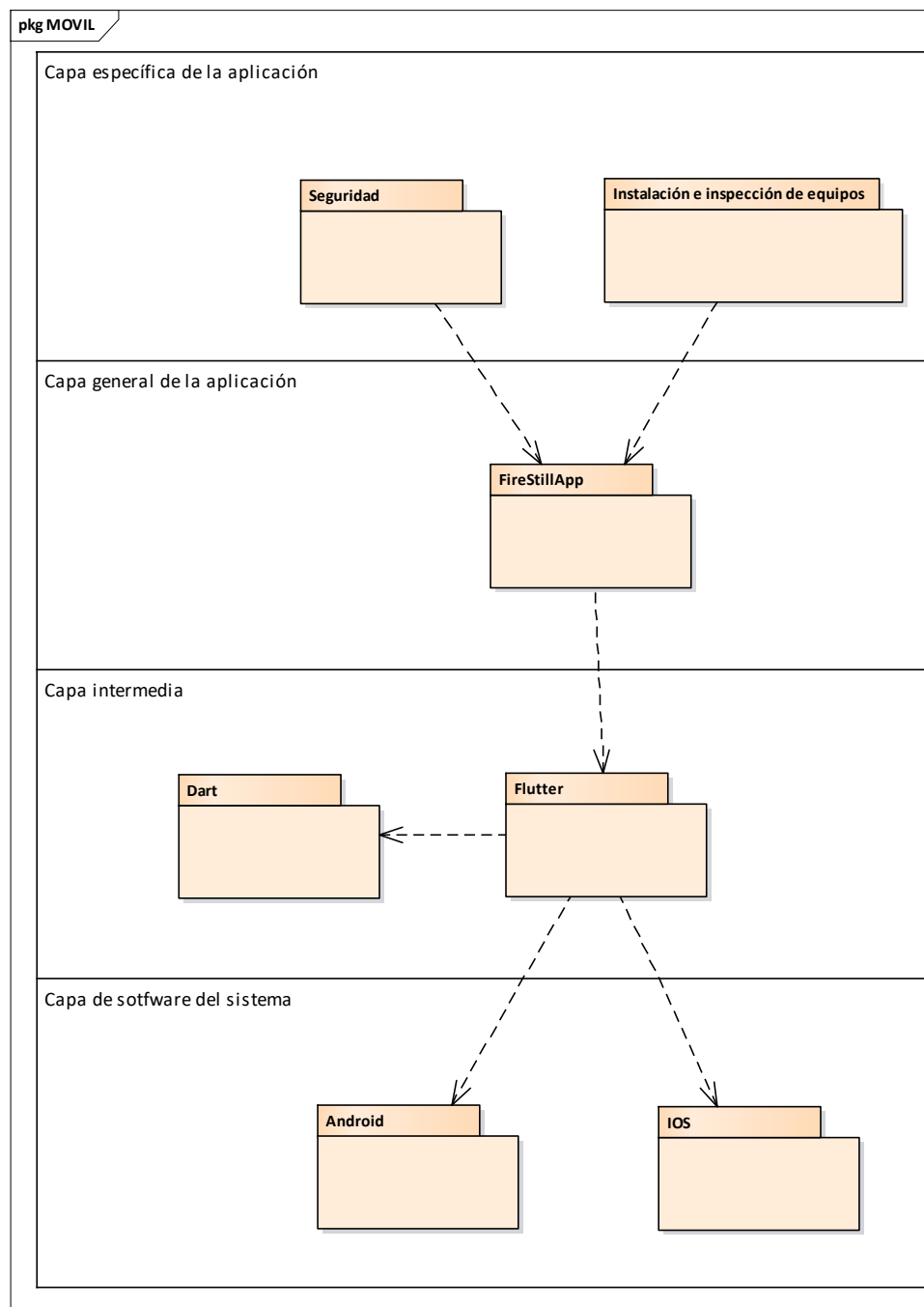


## Diseño de Arquitectura

## Arquitectura Lógica

Figura 3

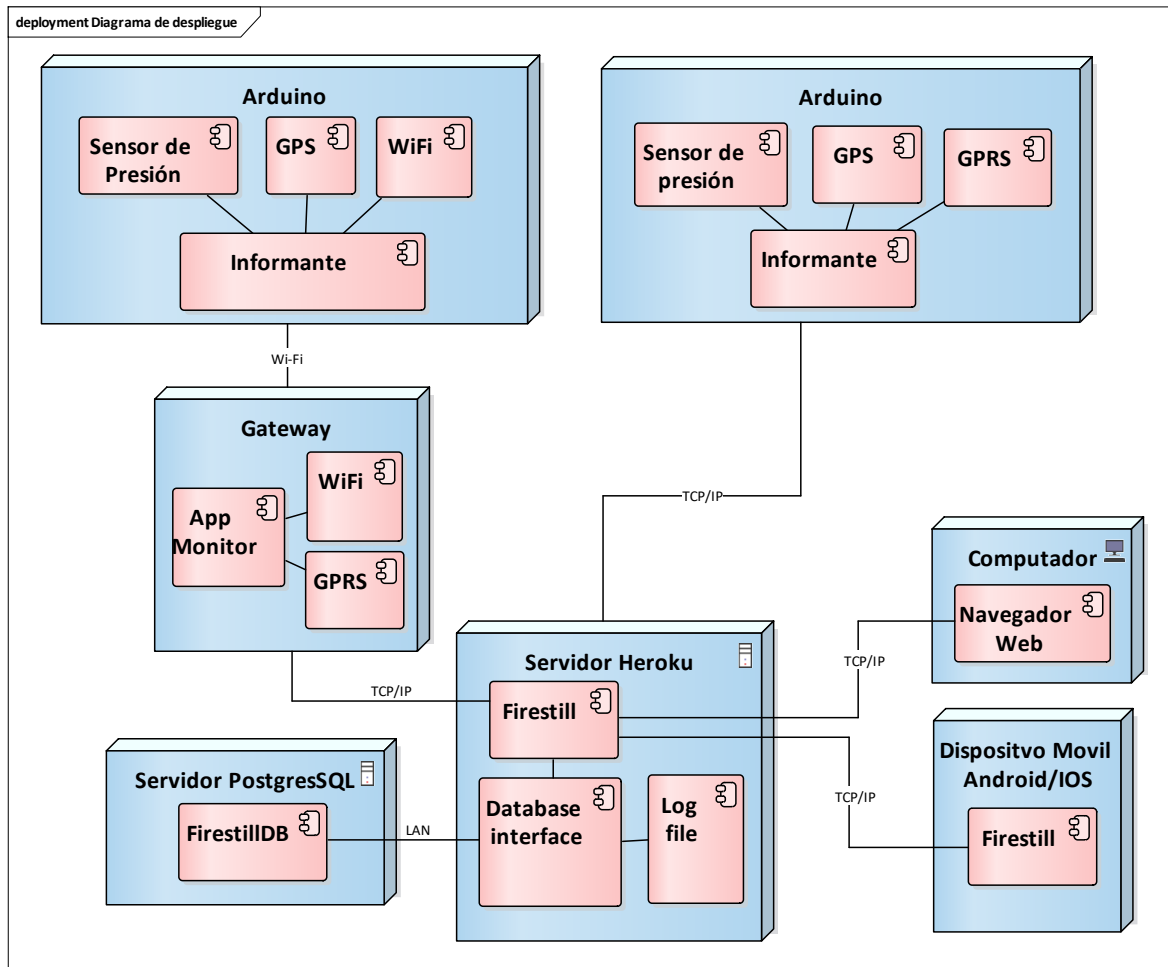
*Arquitectura lógica de la aplicación web*

**Figura 4***Arquitectura lógica de la aplicación móvil*

## Arquitectura Física

Figura 5

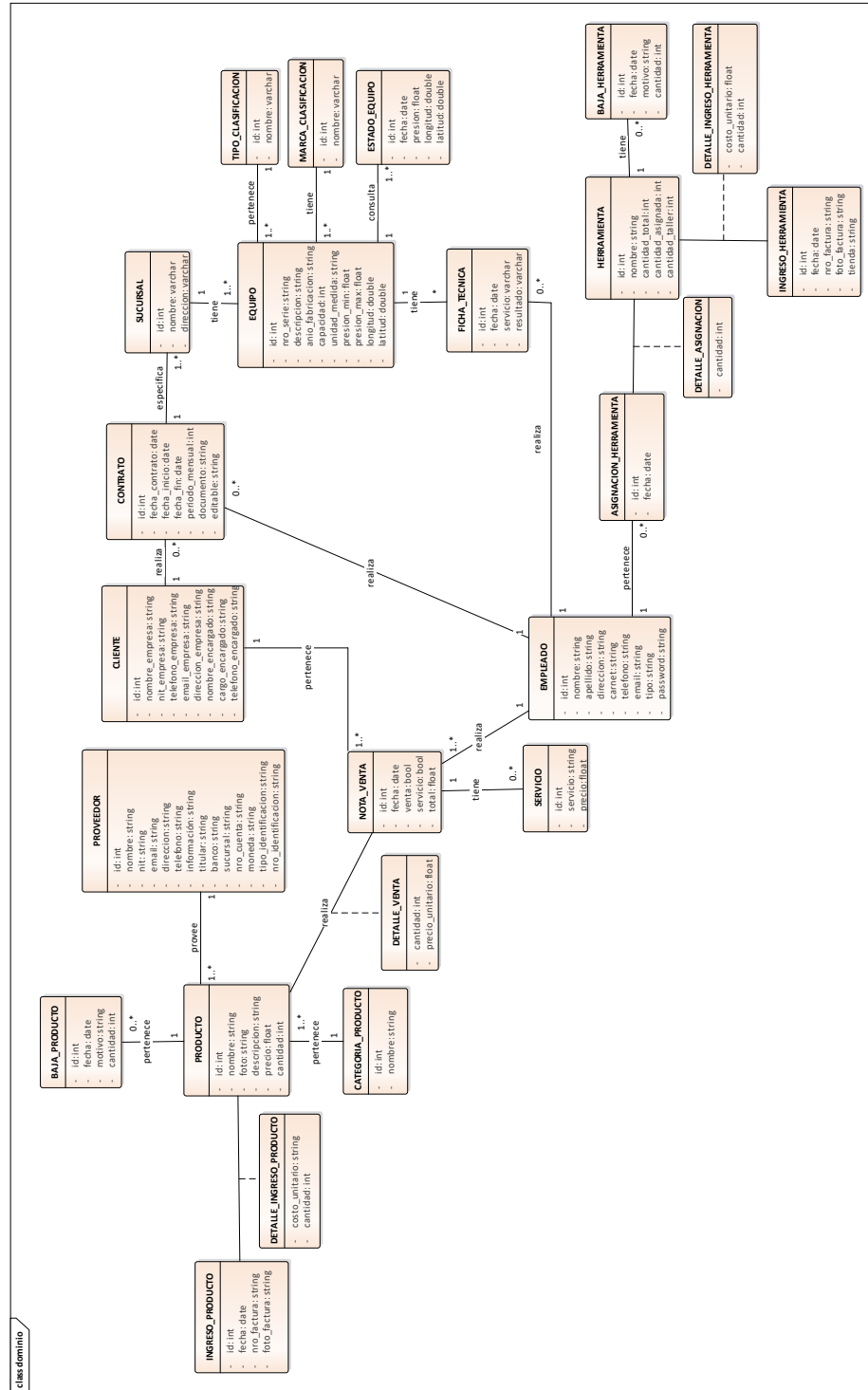
## Arquitectura física



## Modelo de Dominio

Figura 6

## Diagrama de clases



# UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO



“MODELOS FORMATO DIGITAL – HERRAMIENTA CASE”

NOMBRES:

- ABASTO BERBETTY, RODRIGO
- POZADAS MAMANI, INGRID NATHALY

GRUPO: AJ

MATERIA: INGENIERÍA DE SOFTWARE II

FECHA: 16/MAR/2021

SANTA CRUZ - BOLIVIA



## Contenido

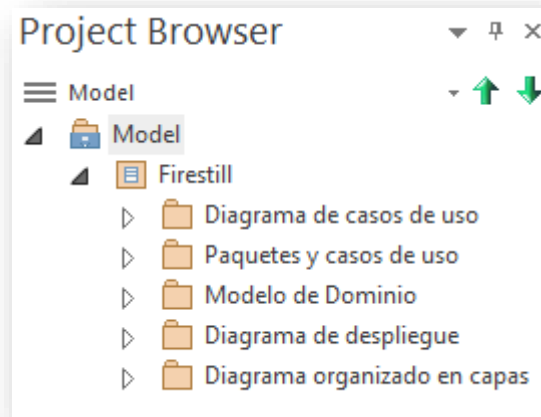
Herramienta CASE .....	2
Características de Enterprise Architect.....	3

## Herramienta CASE

La herramienta CASE que se utilizó para el desarrollo del software es “Enterprise Architect”.



Sparx Systems Enterprise Architect es una herramienta de diseño y modelado visual basada en OMG UML. La plataforma admite: el diseño y la construcción de sistemas de software; modelado de procesos comerciales; y modelado de dominios basados en la industria.



*Ilustración 1 Estructura del proyecto creado con Enterprise Architect*

## Características de Enterprise Architect

### Interfaz de usuario intuitiva

- Amplio rango de barras de herramientas, ventanas acoplables, y estilos visuales.
- Guarde y restaure disposiciones de ventanas personalizadas.
- Modifique y personalice las barras de herramientas y menús.
- Cree sus propios aceleradores.
- Puede "Desplazar" las ventanas acopladas para maximizar el espacio de las ventanas y mejorar la eficiencia de su trabajo.
- Amplio sistema de menús para tener control de su modelo.
- Los accesos rápidos permiten la creación de elementos de diagramas y conexiones sensitivas al contexto

### Soporta los 13 diagramas UML 2.1:

- Los diagramas de comportamiento incluyen: Casos de Uso, Actividades, Estado, Interacción, Secuencia y Comunicación
- Los diagramas de Estructura Incluyen: Paquetes, Clases, Objetos, Composición, Componentes y Despliegue.

### Soporte para Transformaciones MDA

- La Arquitectura Dirigida por Modelos permite transformar elementos simples del modelo en complejos.
- Transformaciones Completas conducidas por plantillas.
- Fácil para escribir y modificar transformaciones de plantillas.
- Realizar transformaciones para DDL, Java, C#, EJB, XSD.
- Generar y sincronizar su Modelo Específico de Plataforma del Modelo Independiente de Plataforma.
- Cada PIM puede soportar múltiples PSMs.

### **Salida en formato de texto enriquecido**

- Opciones simplificadas con filtros y criterios de selección.
- Guarde las plantillas reportadas para una reutilización posterior
- Puede crear documentos maestros personalizados con vínculos sostenibles a las porciones generadas de la salida de EA
- Enterprise Architect permite exportar un modelo completo o una rama simple del modelo a páginas web en HTML
- Fije sus modelos en Internet o en su Intranet usando el generador de reportes HTML
- Los reportes adicionales incluyen para el uso de métricas de casos de uso, pruebas y más.

### **Documentación flexible y comprensible.**

- Plantillas conducidas por generador RTF
- Las Plantillas soportan todas las características de los elementos del modelo de EA y datos extendidos (tales como Pruebas, Riesgos, Recursos, Cambios, etc.)
- Las plantillas soportan encabezados, pies, tablas de contenidos, imágenes embebidas, índices, títulos de páginas, tablas jerarquizadas y más.

### **Ingeniería de Código Directa e Inversa**

- Las plantillas completas encaminan al generador de ingeniería de código - modifique las plantillas incorporadas o empiece a escribirlas de cero
- Agregue lenguajes adicionales
- Sintaxis resaltada del editor del código fuente con capacidad para "guardar y sincronizar" con rapidez
- Puede generar código fuente C++, Java, C#, VB.Net, Visual Basic, Delphi, PHP, Python y ActionScript
- Soporte para Corba también disponible como "plug-in" libre

# UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO



## “INFRAESTRUCTURA PARA EL DESARROLLO DEL SOFTWARE”

NOMBRES:

- ABASTO BERBETTY, RODRIGO
- POZADAS MAMANI, INGRID NATHALY

GRUPO: AJ

MATERIA: INGENIERÍA DE SOFTWARE II

FECHA: 16/MAR/2021

SANTA CRUZ - BOLIVIA

## Contenido

Control de versiones.....	2
Desarrollo colaborativo.....	2

## Control de versiones

Se optó por utilizar “GitHub” que es un proveedor de alojamiento de Internet para el desarrollo de software y el control de versiones mediante “Git”.



Los repositorios creados son los siguientes:

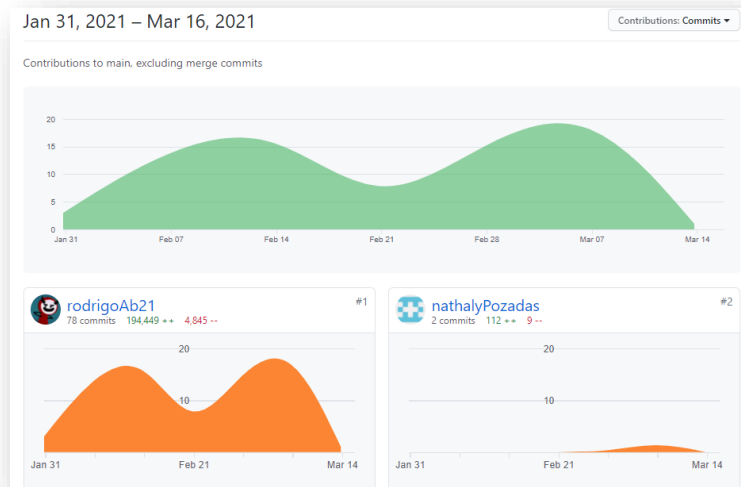
- Aplicación Web: <https://github.com/rodrigoAb21/firestill22020>
- Aplicación Móvil: <https://github.com/nathalyPozadas/tallerMovil>
- Aplicación Arduino: <https://github.com/rodrigoAb21/ArduinoFireStill>
- Documentación: <https://github.com/rodrigoAb21/DocumentacionFirestill>

## Desarrollo colaborativo

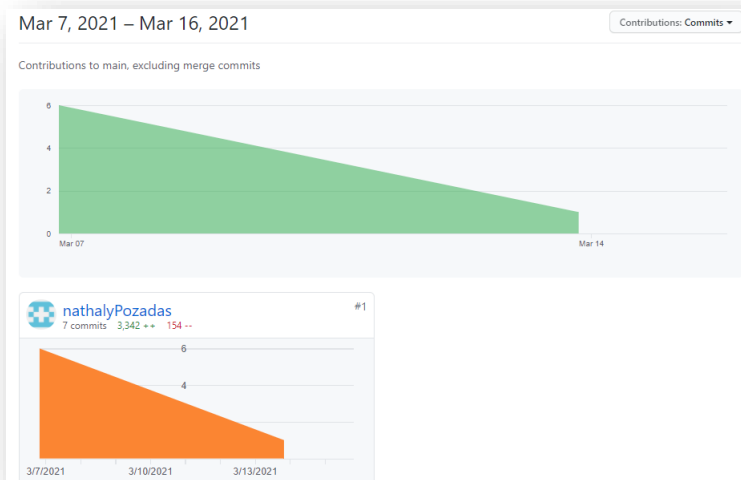
Las únicas personas autorizadas para realizar commits a los proyectos son:

N°	Nombre	Cuenta GitHub
1	Abasto Berbetty Rodrigo	rodrigoAb21
2	Pozadas Mamani Ingrid Nathaly	nathalyPozadas

### Gráfica de contribuciones - Proyecto Web

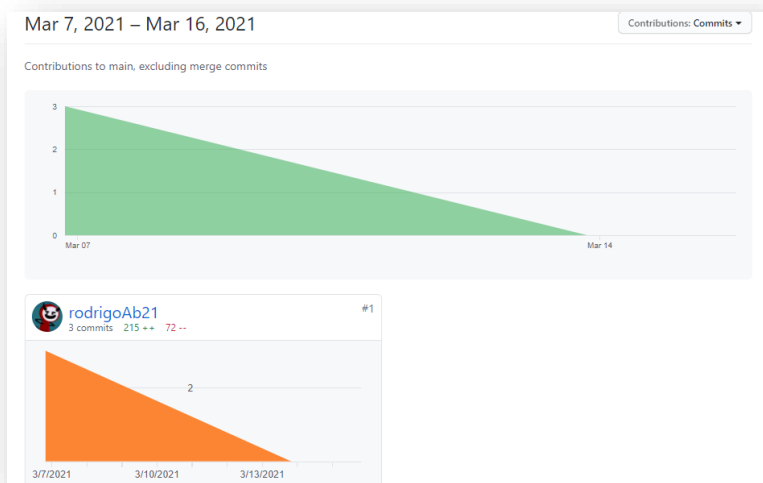


### Gráfica de contribuciones - Proyecto Móvil





### Gráfica de contribuciones - Proyecto Arduino



# UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO



“PRUEBAS”

NOMBRES:

- ABASTO BERBETTY, RODRIGO
- POZADAS MAMANI, INGRID NATHALY

GRUPO: AJ

MATERIA: INGENIERÍA DE SOFTWARE II

FECHA: 16/MAR/2021

SANTA CRUZ - BOLIVIA

## Contenido

Caja Blanca: Prueba del camino básico .....	2
Prueba: Actualizar una categoría de producto .....	2
Prueba: Nuevo Contrato .....	3
Prueba: Eliminar una asignación de herramientas .....	5

## Caja Blanca: Prueba del camino básico

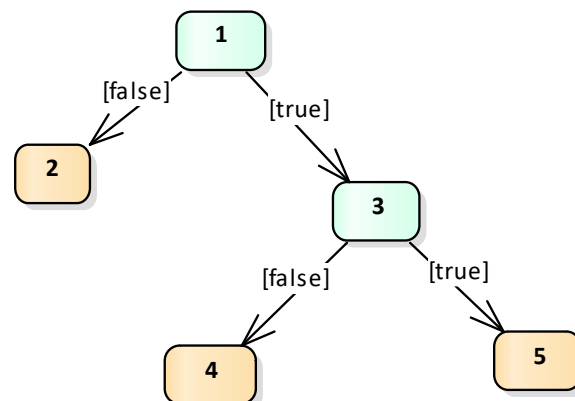
### Prueba: Actualizar una categoría de producto

```
public function update(Request $request, $id)
{
    $this->validate($request, [
        'nombre' => 'required|max:255',
    ]); (1)(2)
    $categoria = Categoria::findOrFail($id); (3)(4)
    $categoria->nombre = $request['nombre'];
    $categoria->update();
    return redirect('categorias'); (5)
}
```

$$V(G) = P + 1$$

$$V(G) = 2 + 1$$

$$V(G) = 3$$



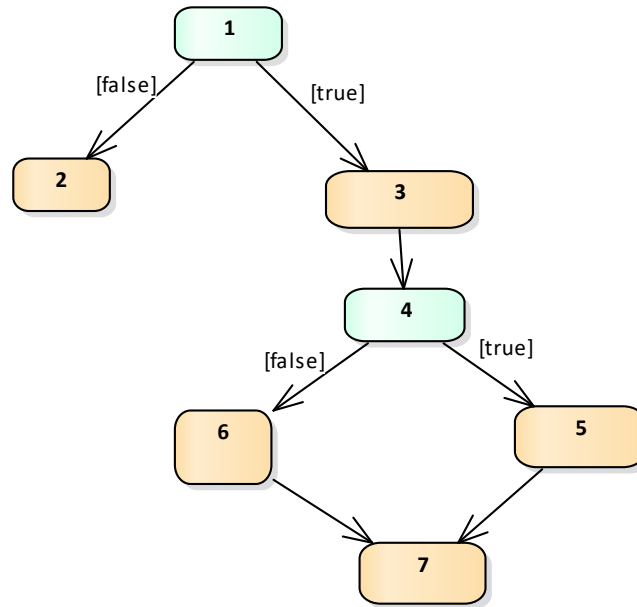
#	Camino	Entrada	Salida
1	1,3,5	{'id':5,'nombre':'mangueras'}	redirect('categorias')
2	1,2	{'id':5, 'nombre':125} (tipo de dato incorrecto)	redirect('categorias/edit/5') ->error('message')
3	1,3,4	{'id':10, 'nombre':'manguera'} (id inexistente)	Página de Error

### Prueba: Nuevo Contrato

```
public function guardarContrato(Request $request){
    $this->validate($request, [
        'fecha_inicio' => 'required|date',
        'fecha_fin' => 'required|date',
        'periodo' => 'required|numeric|min:1',
        'cliente_id' => 'required|numeric|min:1',
        'empleado_id' => 'required|numeric|min:1',
        'documento' => 'nullable|file|mimes:doc,docx,pdf,txt',
    ]); (1)(2)

    $contrato = new Contrato();
    $contrato->fecha_inicio = $request['fecha_inicio'];
    $contrato->fecha_fin = $request['fecha_fin'];
    $contrato->estado = "Vigente";
    $contrato->edicion = true;
    $contrato->periodo = $request['periodo']; (2)
    if (Input::hasFile('documento')) {(4)
        $file = Input::file('documento');
        $file->move(public_path().'/contrato/', $file-
>getClientOriginalName());
        $contrato->documento = $file->getClientOriginalName();(5)
    }else{
        $contrato->documento = 'default.png';(6)
    }
    $contrato->cliente_id = $request['cliente_id'];
    $contrato->empleado_id = $request['empleado_id'];
    $contrato->save();

    return redirect('imonitorio/listaContratos'); (7)
}
```



$$V(G) = P + 1$$

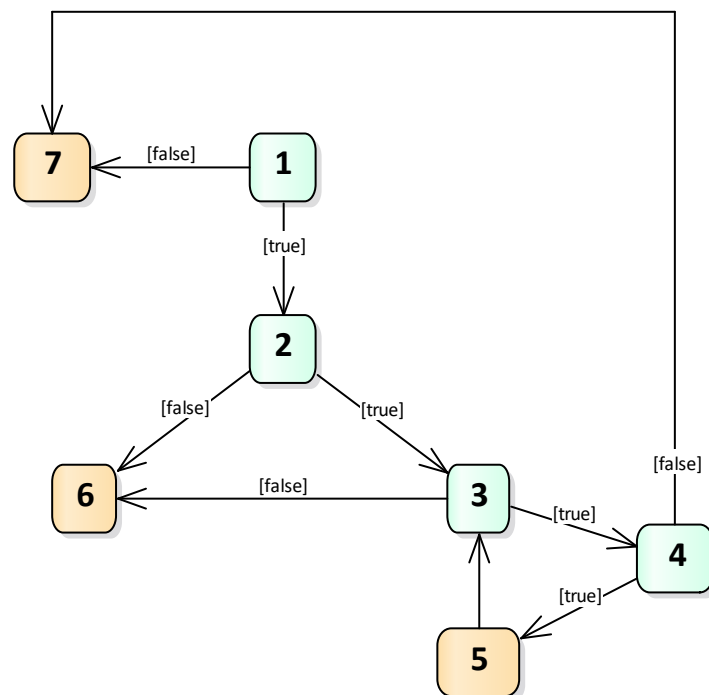
$$V(G) = 2 + 1$$

$$V(G) = 3$$

#	Camino	Entrada	Salida
1	1,2	{'fecha_inicio' : '54.3', 'fecha_fin' : 'abril', 'periodo' : 5, 'cliente_id' : 1, 'empleado_id': 2, 'documento': 'contx.pdf'} (Tipos de datos erroneos)	Redirect('imonitorio/nuev oContrato') ->error('message')
2	1,3,4,6,7	{'fecha_inicio' : '02/02/21', 'fecha_fin' : '02/02/22', 'periodo' : 5, 'cliente_id' : 1, 'empleado_id': 2, 'documento': null} (No se está enviando el file)	redirect('imonitorio/ listaContratos')
3	1,3,4,5,7	{'fecha_inicio' : '02/02/21', 'fecha_fin' : '02/02/22', 'periodo' : 5, 'cliente_id' : 1, 'empleado_id': 2, 'documento': 'contx.pdf'} (Correcto)	redirect('imonitorio/ listaContratos')

## Prueba: Eliminar una asignación de herramientas

```
public function eliminarAsignacion($id)
{
    $asignacion = AsignacionHerramienta::findOrFail($id); (1)(7)
    if ($asignacion->estado == 'Activa'){ (2)
        foreach ($asignacion->detalles as $detalle) {(3)
            $herramienta =
                Herramienta::withTrashed()->findOrFail($detalle-
>herramienta_id);(4)(7)
            $herramienta->cantidad_asignada =
                $herramienta->cantidad_asignada - $detalle->cantidad;
            $herramienta->cantidad_taller =
                $herramienta->cantidad_taller + $detalle->cantidad;
            $herramienta->update(); (5)
        }
    }
    $asignacion->delete();
    return redirect('herramientas/listaAsignaciones'); (6)
}
```



$$V(G) = P + 1$$

$$V(G) = 4 + 1$$

$$V(G) = 5$$

#	Camino	Entrada	Salida
1	1,7	{id:999999999} (id inexistente)	Página Error
2	1,2,6	{id:1} (asignación finalizada)	View("herramientas/listaAsignaciones")
3	1,2,3,6	{id:4} (asignación sin detalle)	View("herramientas/listaAsignaciones")
4	1,2,3,4,7	{id: 3} (asignación con detalle erróneo)	Página Error
5	1,2,3,4,5,3,6	{id:1} (asignación con detalle correcto)	View("herramientas/listaAsignaciones")