
Sistema de Nómina de Empleados

PLAN DE PROYECTO DE SOFTWARE

Sistema de Nómina de Empleados	Curso: Calidad y Pruebas de Software
Plan de Proyecto de Software	Semestre: 2019-2

Revisión histórica

Fecha	Versión	Descripción	Autor
16/08/2019	1.0	Planificación preliminar del proyecto.	Lain C. E.

Sistema de Nómina de Empleados	Curso: Calidad y Pruebas de Software
Plan de Proyecto de Software	Semestre: 2019-2

Tabla de contenido

1.	INTRODUCCIÓN	4
1.1	Propósito	4
1.2	Alcance	4
1.3	Referencias	4
2.	DESCRIPCIÓN DEL PROYECTO	4
2.1	Propósito del Proyecto, Alcance y Objetivos	4
2.2	Productos de Trabajo del Proyecto	4
3.	ORGANIZACIÓN DEL PROYECTO	5
3.1	Estructura organizacional	5
3.2	Roles y responsabilidades	5
4.	GESTIÓN DE PROCESOS	6
4.1	Estimaciones del Proyecto	6
4.2	Plan del Proyecto	6
4.3	Plan de Gestión de Riesgos	7

Sistema de Nómina de Empleados	Curso: Calidad y Pruebas de Software
Plan de Proyecto de Software	Semestre: 2019-2

Plan de Proyecto de Software

1. INTRODUCCIÓN

1.1 Propósito

La finalidad de este Plan es permitir gestionar y controlar las actividades a realizar por el equipo de desarrollo durante el semestre académico.

1.2 Alcance

Este Plan describe las iteraciones, las actividades y los artefactos que serán evaluados durante el semestre académico.

1.3 Referencias

- Especificación de Casos de Uso.
- Especificación Suplementaria.

2. DESCRIPCIÓN DEL PROYECTO

2.1 Propósito del Proyecto, Alcance y Objetivos

El propósito de este proyecto es desarrollar un Sistema de Nómina de Empleados y evaluar la calidad del producto. Su alcance es sólo la implementación de dos casos de uso: Gestionar contratos y Procesar pagos.

Los objetivos del proyecto son:

- Diseñar las clases del dominio y la base de datos en la primera iteración (I-1).
- Implementar los casos de uso con pruebas unitarias en la segunda iteración (I-2).
- Realizar pruebas funcionales y corregir defectos en la tercera iteración (I-3).
- Inspeccionar y refactorizar el software en la cuarta iteración (I-4).
- Medir la calidad del software y elaborar un paper en la quinta iteración (I-5).

2.2 Productos de Trabajo del Proyecto

Para el proyecto se ha seleccionado un conjunto de artefactos a elaborar durante cada iteración. A continuación se muestra el Marco de Desarrollo que permitirá guiar el proyecto:

(c: se comienza a elaborar el artefacto; r: se refina el artefacto y se genera nueva versión)

Sistema de Nómina de Empleados	Curso: Calidad y Pruebas de Software
Plan de Proyecto de Software	Semestre: 2019-2

Disciplina	Artefacto o entregable del producto	I-1	I-2	I-3	I-4	I-5
Requisitos	Modelo de casos de uso (proporcionado)	c				
	Especificación suplementaria (proporcionado)	c				
Diseño	Diagrama de clases con entidades	c	r			
	Diagrama de tablas de base de datos	c	r			
	Diagrama de paquetes de la arquitectura		c		r	
Implementación	Script de base de datos	c	r			
	Software		c	r	r	
	Prueba unitarias		c	r	r	
Prueba	Casos de prueba			c		
	Resultados de prueba			c	r	
Gestión de proyecto	Plan de proyecto de software (proporcionado)	c				
Gestión de calidad	Lista de control de calidad de software				c	
	Métricas de software y paper					c

3. ORGANIZACIÓN DEL PROYECTO

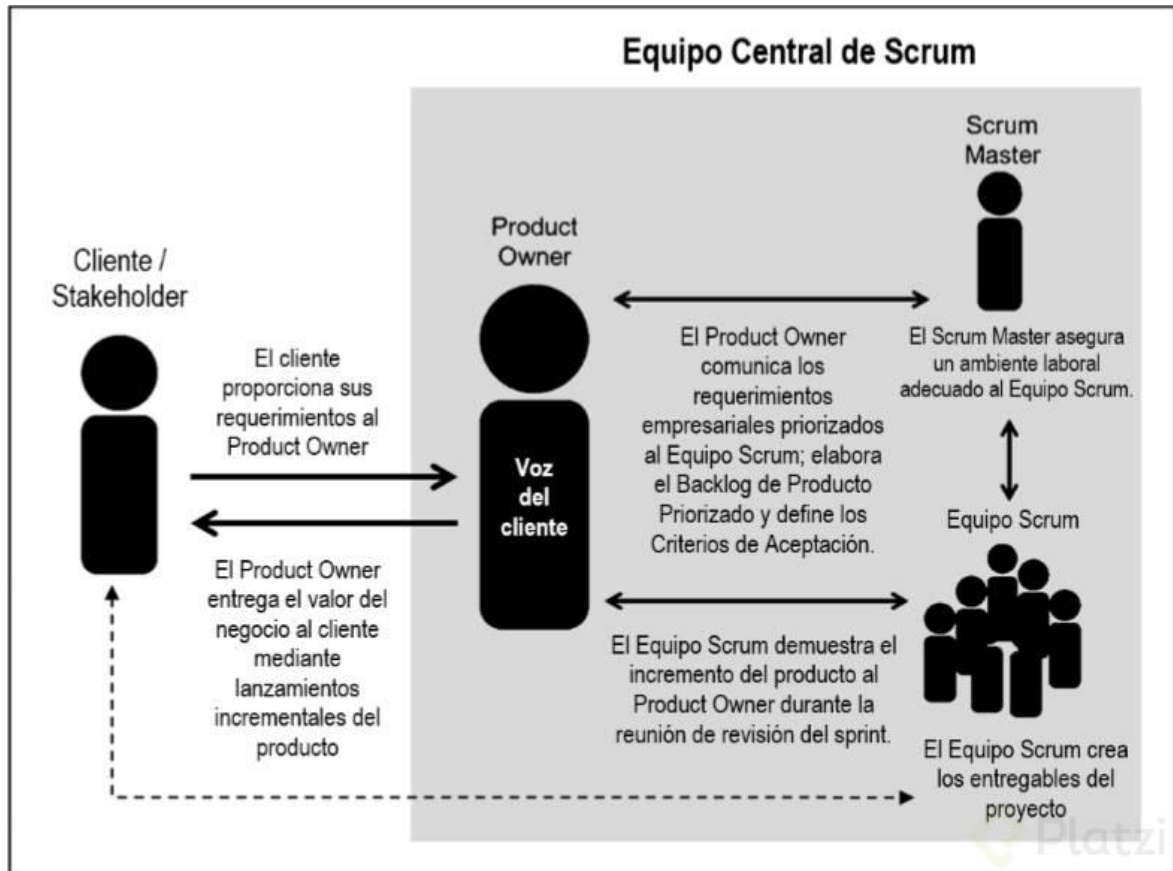
3.1 Estructura organizacional

El proyecto estará formado por tres roles: Scrum Master, Development Team y Product Owner.

3.2 Roles y responsabilidades

Rol	Responsabilidad
Scrum Master	<p>Es el coordinador de las tareas del equipo de desarrollo. Es responsable de asegurarse que el trabajo del equipo vaya bien. Debe coordinar y presentar los entregables al Product Owner.</p> <p>Al final de cada iteración o sprint, debe realizar una reunión de retrospectiva con el equipo de desarrollo con el objetivo de mejorar de manera continua la productividad y la calidad del producto que está desarrollando, la motivación del equipo, cómo están engranando entre ellos, como fue la última iteración o cómo está yendo el proyecto. El alumno que asuma este rol, también debe cumplir el rol de Development Team.</p>
Development Team	<p>Son los miembros del equipo de desarrollo encargados de programar, realizar pruebas unitarias y pruebas funcionales. Además, son responsables de medir, inspeccionar y refactorizar el código para asegurar la calidad del software.</p>
Product Owner	<p>Es la “voz del cliente” y el responsable de elaborar, mantener y priorizar los requisitos, así como de hacer las pruebas de aceptación.</p> <p>Este rol será realizado por el docente del curso quien evaluará los entregables del producto a través de revisiones semanales y al final de cada iteración o sprint.</p>

Sistema de Nómina de Empleados	Curso: Calidad y Pruebas de Software
Plan de Proyecto de Software	Semestre: 2019-2



4. GESTIÓN DE PROCESOS

4.1 Estimaciones del Proyecto

Para el desarrollo del proyecto se tomará en cuenta un total de 16 semanas correspondiente al semestre académico.

4.2 Plan del Proyecto

Iteración o Sprint	Inicia	Finaliza	Evaluación
1	Semana 1	Semana 4	T1
2	Semana 5	Semana 8	T2
3	Semana 9	Semana 11	T3
4	Semana 12	Semana 14	T4
5	Semana 15	Semana 16	EF

Sistema de Nómina de Empleados	Curso: Calidad y Pruebas de Software
Plan de Proyecto de Software	Semestre: 2019-2

Semana	Actividad	Artefacto o entregable del producto	Semana de presentación
1	Formar el equipo de desarrollo y presentar el Formato de Proyecto de Software con las siguientes partes: título, integrantes y características técnicas del producto software a desarrollar.	Formato de Proyecto de Software con las siguientes partes: título, integrantes y características técnicas del producto software a desarrollar.	2
2	Elaborar el diagrama de clases de diseño con Entidades de negocio en base a la especificación de los casos de uso.	Diagrama de clases.	3
3	Diseñar e implementar la base de datos en SQL Server.	Diagrama de tablas y Script de la base de datos en SQL Server.	4
4	Elaborar el diagrama de paquetes que represente el modelo arquitectónico del Software.	Diagrama de paquetes de la arquitectura.	5
5	Implementar el Software para un caso de uso principal a un 50%.	Software en versión 1.1	6
6	Implementar el Software para un caso de uso principal al 100%.	Software en versión 1.2	7
7	Implementar las pruebas unitarias del Software para un caso de uso principal.	Software con pruebas unitarias en versión 1.3	8
8	Depurar el Software para pasar con éxito las pruebas unitarias.	Software en versión 1.4	9
9	Diseñar casos de prueba para evaluar la funcionalidad del Software.	Casos de prueba.	10
10	Ejecutar los casos de prueba y depurar el Software para pasar con éxito las pruebas funcionales.	Software en versión 1.5	11
11	Inspeccionar el Software.	Lista de Control de Calidad.	12
12	Refactorizar el Software.	Software en versión 2.1	13
13	Refactorizar el Software. Inspeccionar el Software.	Lista de Control de Calidad Software en versión 2.2	14
14	Realizar mediciones al Software con herramientas de métricas estáticas. Elaborar un avance del PAPER.	Avance del PAPER	15
15	Corregir y completar el PAPER en base a las observaciones del docente.	PAPER.	16

4.3 Plan de Gestión de Riesgos

Los riesgos son problemas potenciales que pueden afectar el proyecto. Estos riesgos deben ser identificados y listados para describir sus acciones de mitigación que permitan reducir la probabilidad de ocurrencia o sus acciones de contingencia para solucionarlos si llegan a ocurrir.

A continuación se listan cuatro riesgos considerados los de mayor probabilidad de ocurrencia e impacto en el proyecto:

Sistema de Nómina de Empleados	Curso: Calidad y Pruebas de Software
Plan de Proyecto de Software	Semestre: 2019-2

Riesgo	Acción de mitigación	Acción de contingencia
El equipo de desarrollo podría tener retraso en la culminación de las actividades y elaboración de los artefactos programados por cada iteración.	Mantener un equipo motivado y comprometido. Hacer reuniones diarias con el equipo para conocer el avance y dar solución a los problemas. Trabajar de forma colaborativa.	El Scrum Master debería informar oportunamente al Product Owner (docente) los motivos del retraso y asumir compromisos para terminar las actividades pendiente y elaboración de artefactos.
Podría existir falta de comunicación y entendimiento entre los miembros del equipo.	Crear un grupo privado para el proyecto en redes sociales o WhatsApp. Hacer uso del correo electrónico. Respetar y valorar los aportes que pueda dar cada miembro del equipo.	El Scrum Master debería informar oportunamente al Product Owner (docente) los problemas internos que pueda tener el equipo para que apoye en la solución.
El equipo de desarrollo podría perder todos o parte de los artefactos elaborados en la iteración.	Guardar semanalmente todos los artefactos del proyecto en Google Drive u otro servicio similar.	El Scrum Master debería coordinar oportunamente con el Product Owner (docente) un plazo adicional a la fecha de presentación de los artefactos.
El equipo de desarrollo podría realizar una entrega Incorrecta de versiones del software.	Usar un sistema de control de versiones como Git u otro servicio similar.	El Scrum Master debería justificar el error con el Product Owner (docente) y recuperar la versión correcta del software de algún repositorio donde esté guardado.