

Plan de Proyecto de Software de GesCond-247

1. Introducción

Este Plan de Desarrollo del Software es una versión preliminar preparada para ser incluida en la propuesta elaborada como respuesta al proyecto de prácticas de la asignatura de Programación de Aplicaciones en Redes de la Universidad Nacional de Ingeniería. Este documento provee una visión global del enfoque de desarrollo propuesto.

a. Propósito del plan

El propósito del Plan de Desarrollo de Software es proporcionar la información necesaria para controlar el proyecto. En él se describe el enfoque de desarrollo del software.

Los usuarios del Plan de Desarrollo del Software son:

El jefe del proyecto lo utiliza para organizar la agenda y necesidades de recursos, y para realizar su seguimiento.

Los miembros del equipo de desarrollo lo usan para entender lo qué deben hacer, cuándo deben hacerlo y qué otras actividades dependen de ello.

El proyecto busca solucionar la problemática actual que se tiene cuando en un determinado lugar hay una comisión de una persona (o más) que velan por las deudas y cuentas del edificio o lugar comunal donde se vive.

Haciendo que el sistema sea automático y genere facturas según los cálculos realizados por el sistema basados en la información de cada usuario (residente o inquilino).

Se hará mediante una interfaz de usuario (web y a futuro móvil) de tal manera que un usuario registre su fecha de pago y monto y posiblemente una foto sobre su voucher de pago, para confirmar que realmente se este haciendo la culminación del pago a su deuda o mantenimiento mensual.

b. Ambito del proyecto y objetivos

i. Declaración del ámbito

Se tiene todo tipo de empresas de condominios, residenciales, etc, que requieran un sistema automático.

GesCond 247 será un sistema que se encarga de gestionar todo tipo de transacciones de propietarios para con los pagos de mantenimiento, alquiler, o pago de su departamento, luz, agua, etc.

Por eso, para un condominio, es necesario tener un sistema que pueda manejar todos esos trámites de manera automática y sin tener que gastar demasiado esfuerzo en realizar cálculos manuales.

ii. Funciones principales

Se tiene 5 principales gestiones:

- Gestión de cronogramas de abonos

- Gestión de gastos

- Gestión de mantenimiento

Gestión de áreas comunes

Gestión de accesos

iii. Aspectos del rendimiento

* El sistema estara netamente sometido al funcionamiento del servidor remoto, el cual brindará el servicio de registro de información a los usuarios o clientes pagadores de este.

* El sistema automatizará el trabajo de dicha comisión de modo que cuando los clientes tengan que pagar la mora, no haya discusiones respecto a fechas de pago debido a que habrá fechas de registro de pagos.

* Las desiciones del sistema para los deudores sera tomado en base a su historial como pagador, puntual, moroso o variado (un mes puntual y otro mes moroso). Por lo que se otorgaría algún método de selección.

iv. Restricciones técnicas y de gestión

a) Compatibilidad:

* El sistema al ser levantado sobre un servidor que brinda hosting, será compatible en cualquier sistema operativo y en cualquier protocolo de comunicación, la unica manera de fallar es cuando no haya internet o se tenga una conexión muy pobre.

* El sistema contemplará la seguridad de información recepcionada en base a logins de usuario. Los cuales serán inquilinos, residentes y propietarios, solo el usuario propietario puede dar permisos a otro usuario (inquilino o residente) para poder realizar subida de información respecto a fecha de pagos.

b) El sistema en sí es una simulación del trámite que realiza la persona encargada (sea cálculos y poonderados e historial de pagos y selección de morosos y a su vez seguimiento de deudores) el cual se encargará de brindar información sobre dicha lista de personas y sobre sus transacciones de pago (no necesariamente pagar por el sistema, sino mediante subida de información como el vucher de pago).

2. Estimaciones del proyecto

Los hitos que marcan el final de cada fase se describen en la siguiente tabla.

Descripción	Hito
Fase de Inicio	En esta fase desarrollará los requisitos del sistema la perspectiva del cliente. Los principales casos de uso serán identificados y se hará un refinamiento del Plan de Desarrollo del Proyecto. La aceptación del cliente / usuario del artefacto Visión y el Plan de Desarrollo marcan el final de esta fase, que culminaría con definir que necesidades tiene el cliente para gestionar un condominio.
Fase de Elaboración	En esta fase se analizan los requisitos y se desarrolla un prototipo de arquitectura. Se define el software a usar, se define los lenguajes (php, javascript, HTML, CSS, MySQL o PostgreSQL) y el servicio de hosting (google, amazon, hostinger, etc) sobre el cual estará el software
Fase de Construcción	Durante la fase de construcción se terminan de analizar y diseñar todos los casos de uso. Se tendrá en cuenta los hitos del proyecto y entregables para el cliente, cuando haya una versión estable se le denominará versión 1.0 que cumpla las funcionalidades mínimas, otra versión 2.0 que ya tenga interfaz usuario y una final versión 3.0 que sera el sistema final.
Fase de Transición	En esta fase se prepararán dos releases para distribución, asegurando una implantación y cambio del sistema previo de manera adecuada, incluyendo el entrenamiento o capacitación del software hacia los usuarios. El hito que marca el fin de esta fase incluye, la entrega de toda la documentación del proyecto con los manuales de instalación y todo el material de apoyo al usuario, la finalización del entrenamiento de los usuarios y el empaquetamiento del producto.

a. Datos históricos usados por las estimaciones

Se tiene en cuenta

b. Técnicas de estimación

Se toma en cuenta la ganancia esperada y la facilidad del sistema con el cual el usuario sienta que todo le es más sencillo, administrar toda una base de datos de números y cuentas, las cuales las tendrá a tan solo un botón o una línea de consulta, todo sera mediante la Interfaz Usuaio que será la aplicación web (ya futuro posiblemente móvil).

c. Estimaciones de esfuerzo, coste y duración

Estimación de esfuerzo:

- Como se en cuenta en el esquema de viabilidad, el esfuerzo es de aproximadamente 2 desarrolladores expertos usando 4 meses a tiempo completo.

Estimación de coste:

- Se tiene el pago de 2500 mensual a tiempo completo, entonces 2 desarrolladores a tiempo completo por 4 meses equivale a 20000.

Estimación de duración:

- 4 meses.

3. Estrategia de gestión de riesgo

a. Tabla de riesgo

Según la clasificación se tiene 4 tipos de riesgos:

Riesgo técnico: Es cuando se desarrolla el software, la falla de instalación de componentes, compatibilidad de las versiones de software (java 7 con java 8), librerías usada al momento de implementar, ausencia del personal, retraso por cuestión de bugs de sistema, no conectividad al aservidor.

Riesgo de mercado: Cuando el software falla una vez ya vendido, haya errores al momento de ejecutar o que una plataforma lo instale y justo no haya compatibilidad con dicho sistema.

Riesgo de proyecto: Conforme el proyecto avanza y el personal se vaya retirando o haya fallas internas, los mismos encargados deben tomar la responsabilidad de continuar o buscar nuev personal en el tiempo más pronto posible.

Riesgo financiero: Cuando el capital se va terminando y no haya fondos para continuar el proyecto.

Riesgos humanos: Que haya mala comunicación entre los miembros del equipo de desarrollo o entre departamentos participantes del proyecto. O que hayan miembros que por falta de responsabilidad no entreguen sus trabajos o avances a tiempo y genere la desconfianza entre el equipo.

b. Estudio de riesgos a tratar

En nuestro proyecto, se tendrá los riesgos técnicos (como cuando se caiga el servidor) y el riesgo humano (que dependerá netamente del liderazgo del jefe de proyecto).

c. Plan RSGR para cada riesgo

i. Reducción del riesgo

Se tiene los 3 riesgos que nos afectan directamente y fueron descritos anteriormente.

1.- Riesgo de Proyecto, se reduce el riesgo teniendo personal responsable.

2.- Riesgo Técnico, las personas escogidas como desarrolladoras del proyecto, deben entregar sus tareas asignadas a la fecha.

3.- Riesgo de Viabilidad, De que el software desarrollado sea adaptable a las nuevas tecnologías que emergen día a día.

ii. Supervisión del riesgo

El encargado de proyecto, y demás encargados de las áreas de desarrollo, gestión de Hitos (basados en un POP), deben velar siempre que se cumplan todas las asignaciones hasta cierto goal o meta esperada.

iii. Gestión de riesgo (planes de contingencia)

Cuando el software falla, o cuando hayan entregas que no se cumplan, o el personal se reduce de un día al otro, siempre se tendrá un backup de todo el trabajo realizado, y además, un servicio alternativo al principal, el cual se comunique al principal, obtenga los últimos datos almacenados y muestre dichos datos.

4. Planificación temporal

a. Estructura de descomposición del trabajo del proyecto

A continuación se describen las principales responsabilidades de cada uno de los puestos en el equipo de desarrollo durante las fases de Inicio y Elaboración, de acuerdo con los roles que desempeñan.

Puesto	Responsabilidad
Jefe de Proyecto	El jefe de proyecto asigna y decide los recursos, gestiona las tareas principales, coordina las reuniones con los clientes y usuarios, mantiene los objetivos del proyecto. Establece el estandar de programación junto al medio de comunicación (como Trello o SCRUM).
Analista de Sistemas	Captura, especificación y validación de requisitos, se reúne con el cliente y los usuarios mediante entrevistas. Elabora el Modelo de Análisis y Diseño. Colabora en las pruebas funcionales del sistema.
Programador	Programa e implementa el sistema.
Ingeniero de Software	Gestión de requisitos, gestión de configuración y cambios, elaboración del modelo de datos, preparación de las pruebas funcionales, elaboración de la documentación. Elaborar modelos de implementación y despliegue.

b. Red de tareas

En el proyecto se tiene las siguientes tareas:

- Análisis.
- Diseño.
- Código.
- Prueba.

Aplicando este modelo de ciclo de vida a un software de nuevo desarrollo podemos plantear las siguientes tareas:

- Entrevista con el cliente.
- Entrega de una propuesta inicial.
- Hacer los cambios.
- Seleccionar miembros del equipo.
- Investigar recursos.
- Diseñar la interface de usuario.
- Desarrollar el plan del proyecto.
- Programar la aplicación.
- Integración de módulos.
- Depurar la aplicación.
- Pruebas.
- Calendario de entregas finales (aplicación, documentación, entrenamiento, etc.)

Es importante notar que:

1. Cada tarea produce a su vez documentación específica.
2. se pueden programar uno o mas incrementos, cada uno con sus cuatro fases de analisis, desarroll, código y pruebas.

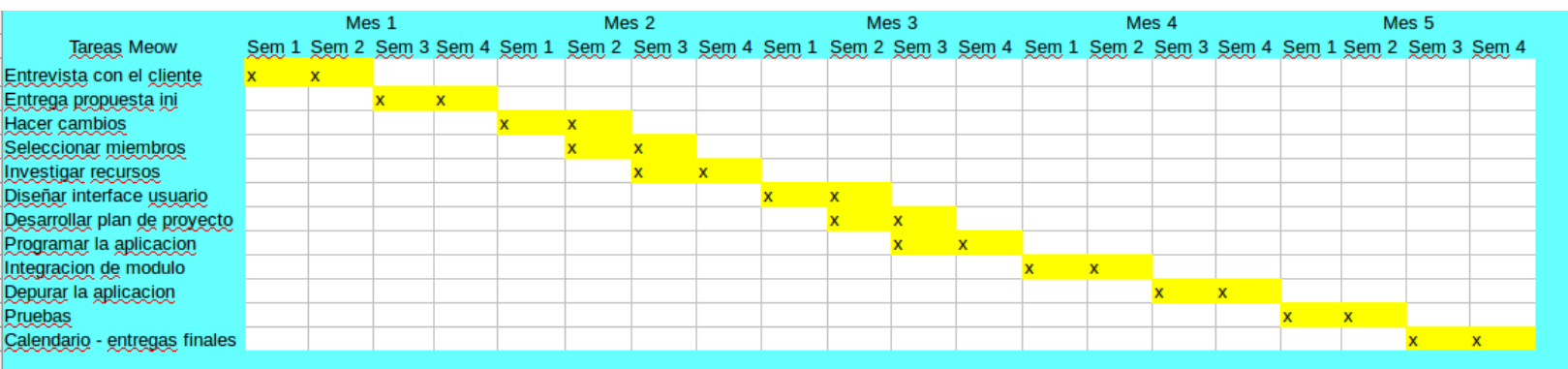
Refinamiento de las tareas

Las tareas generales pueden emplearse para definir una planeación temporal a nivel macro, por ejemplo, para una red de tareas, y luego refinarse para crear una planeación temporal detallada. Cada tarea general se descompone en subtareas, cada una con entregas y metas intermedias.

Por ejemplo, la tarea de entrevista con el cliente se puede descomponer en:

- Identificar las necesidades y limitaciones del producto. Meta intermedia: definir el producto, vistas, diseño, servicios.
- Definir las OCI (entradas, control y salidas) del sistema. Meta intermedia: documento OCI.
- Definir las funciones principales. Meta intermedia: propuesta inicial.

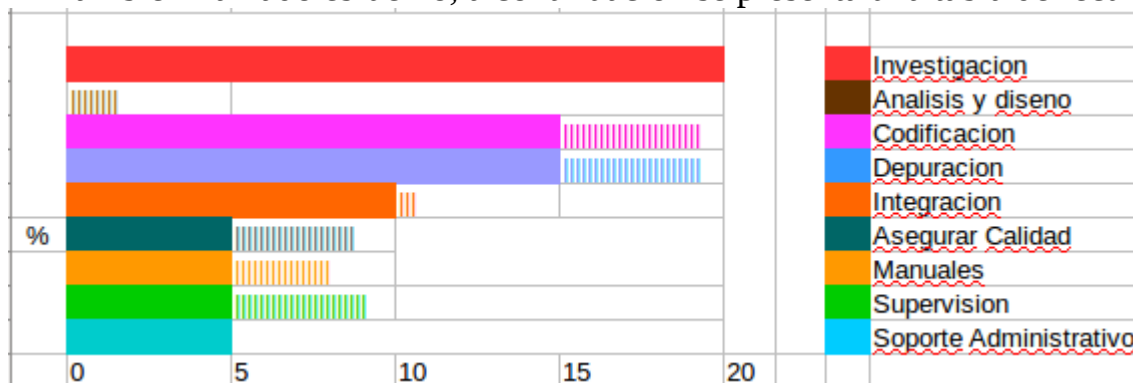
c. Gráfico de tiempo (Gráfico Gantt)



Fase	Nro. Iteraciones	Duración
Fase de Inicio	1	1 mes
Fase de Elaboración	1	1 mes y 2 semanas
Fase de Construcción	2	2 meses y 2 semanas
Fase de Transición	-	-

d. Tabla de recursos

También llamado esfuerzo, a continuación se presenta una tabla de recursos.



5. Recursos del proyecto

a. Personal

GesCond247 definirá los participantes del proyecto que proporcionarán los requisitos del sistema, y entre ellos quiénes serán los encargados de evaluar los artefactos de acuerdo a cada subsistema y según el plan establecido.

El equipo de desarrollo interactuará activamente con los participantes de GesCond247 para especificación y validación de los artefactos generados.

Tendrá que seguir los siguientes pasos:

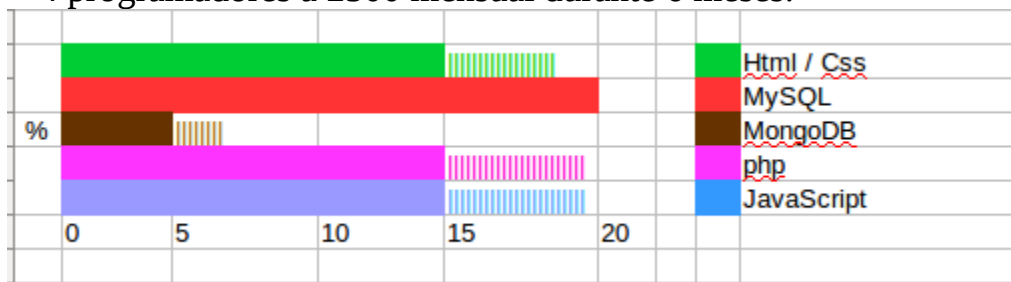
1. Gestión de programadores: (Que conozcan el software LAMP).
2. Verificación de la adquisición: (Compatibilidad del sistema con LAMP, servidores web).
3. Validación de la adquisición: (Comprobar el hosting,y servicios).
4. Evaluación de la adquisición: (Pruebas realizadas al software).

b. Hardware y software

El software requerido, según las especificaciones del esquema de viabilidad es: LAMP o MEAN, que son diverss softwares para preparar el entorno de desarrollo desde tu computadora portátil.

c. Tabla de recursos

4 programadores a 2500 mensual durante 6 meses.



6. Organización del personal

a. Estructura de equipo (si procede)

El equipo final del proyecto ya aprobado para la realización completa y sus respectivas responsabilidades son:

Puesto	Responsabilidad	Conocimiento	Habilidades
Lider de Proyecto	Llevar a cabo a coordinación y gestión de fechas de los entregables e hitos, haciendo seguimiento al desarrollo del software.	Sabe todos los temas relacionados al proyecto incluyendo posibles soluciones alternativas para cuando hayan problemas de compatibilidad de software	Toma desiciones en base a sus amplios conocimientos y experiencia dirigiendo y/o desarrollando.
Analista de Sistemas	Hace un análisis sobre el proyecto, también sobre el sistema final a crear, la compatibilidad, el enfoque.	Conoce el flujo del desarrollo del software, conoce sus limites y alcances.	Habilidad de comunicación con lo desarrolladores y lider de proyecto.

Programador	Se encarga de programar el sistema usando los estándares ya definidos por el lider de desarrolladores y el analista de sistemas.	Lenguaje de programación como php, MySQL, manejo de servidores locales (como Apache) y conexión a servidores externos(usando FileZilla)	Facilidad de entendimiento de código, adaptación a un sistema de codificación y uso de lenguajes.
Ingeniero de Software	Planificar el desarrollo del software, desde el procesado al apretar un botón de login hasta la impresión de una factura.	Modelado de procesos, como el login, proceso de impresión, cálculo de deudas.	Conoce el flujo de proceso, por lo que puede modificar el software facilmente.
Lider de desarrolladores	Supervisa el trabajo de los programadores a su cargo.	Conoce todo sobre los lenguajes, servicios y requerimientos a usar en el sistema, y aparte de dominar otras alternativas en caso haya problemas.	Reconocimiento de problemas y errores de código, asi como mucha facilidad de legibilidad de código.
Capacitador	Imparte conocimientos o una capacitación del software ya desarrollado a los usuarios.	Funcionalidad del sistema desarrollado, y conocimientos sobre las herramientas usadas.	Comprender las necesidades de las personas, y rapidez de solución.
Administrador del Sistema	Realiza pruebas de superusuario, encontrando bugs del sistema y reportándolos	Funcionalidad del sistema, así como su gestión.	Organizar al equipo e identificar vacios en el flujo del sistema.

b. Informes de gestión

Se tiene planeado que cada miembro del proyecto presente un informe mensual en donde indique su avance, que aportó al proyecto y que tareas tiene aún, hasta el siguiente mes o hito de proyecto.

7. Mecanismos de seguimiento y control

a. Garantía de calidad y control

En base a los enfoques tratados anteriormente, la empresa tratará con el **Enfoque Practico** el cual es mejorar en corto plazo.

1. Verificar los problemas de implementación y la aceptación del mercado
2. La necesidad de gestión y manejo de cuentas de deudores.

b. Gestión y control de cambios

Cuando haya un cambio en el proceso de realización del proyecto se reevaluará lo siguiente:

I. Gestión de Requisitos:

Los requisitos del sistema son especificados en el entregable de especificación de requisitos. Si en caso haya que cambiar de software, se instalará o preparará el entorno necesario para utilizar dicho software o librería o Framework.

II. Control de Plazos:

El calendario del proyecto será modificado debido al cambio del software que haya o a las negociaciones o requerimientos(nuevos o mejorados) del cliente, coordinando con dicho cliente los nuevos plazos de tiempo y entrega.

III. Control de Calidad:

Los defectos encontrados ya sea en la compatibilidad o posibles bugs del sistema (siempre hay) se pueden enlistar y hacer una garantía o solución para cuando sucedan dichos bugs.

IV. Gestión de Riesgos:

A partir de la lista de riesgos, se tendrá en cuenta el riesgo de modificación haciendo backups de las versiones y estrategias para mitigar dichos problemas asociados a los cambios.

V. Gestión de Configuración:

Se realizará una gestión de configuración, las cuales serán estables y serán de uso y mientras no, serán sometidas a pruebas y debugeo hasta que sea estable, a cada entregable estable se le denominará versión.

8. Apéndices

1. Contrato de prestación de servicios:

Se adjunta el contrato con cláusulas de confidencialidad para los programadores.

2. Manual de instalación:

Se adjunta un manual de instalación del software necesario (en este caso LAMPP server).