|  |
| --- |

**Especificación de Requisitos de Software (SRS)**

**Plan de Gestión de Proyecto (PGP)**

Proyecto: ¡Oh my dog!

Revisión: 1



|  |  |  |
| --- | --- | --- |

**Plan de Gestión de Proyecto (PGP)**

**1) Introducción**

1. Propósito y alcance   
     
   Se debe especificar cuál es el propósito de este plan de gestión de proyecto y la audiencia a la cual está dirigido.

Este documento tiene como propósito ser el marco del plan de realización del proyecto ¡Oh my dog! Y los recursos necesarios para llevarlo a cabo. En este se especifica los plazos, roles, responsabilidades, recursos y presupuesto estimado para la realización exitosa de todos los objetivos del proyecto teniendo en cuenta como parte fundamental los riesgos presentes en este.

Este documento se encuentra dirigido a las diferentes partes interesadas en el proyecto, como lo son el equipo de desarrollo de AllaDevs y los dueños de la veterinaria Pedro y Lucia.

El proyecto ¡Oh my dog! es una aplicación web la cual servirá como servicio de gestión de turnos para la veterinaria de igual nombre, ademas este brindara la posibilidad de facilitar la comunicación y/o conexión entre diferentes individuos en contexto de posesión o no de perros, como lo es publicitar servicios de paseo y cuidado de terceros a la veterinaria, conectar perros perdidos con sus dueños, publicitación de campañas de donación, publicitación de perros en adopción y para cruza, entre otras funcionalidades más que le aportan un bien a la comunidad.

1. Definiciones, acrónimos y abreviaturas a considerar  
     
   Se debe explicar brevemente cualquier término utilizado en esta sección del documento que pueda no ser comprendido por la audiencia a la que está dirigido.

Definiciones:

* Aplicación web: programa informático diseñado para ejecutarse en navegadores con uso de internet.
* No cliente: persona que no posee cuenta en el sistema, o en su defecto que no inició sesión en este.
* Cliente: persona a la que un empleado la registró en el sistema y goza de una cuenta.
* Empleado: persona que tiene registrada en el sistema y goza de una cuenta con permisos de gestión en el sistema.
* Minimalismo: corriente artística que utiliza elementos mínimos y básicos; tendencia a reducir a lo esencial y a despojar de elementos sobrantes.
* Encriptar: la acción de transcribir un texto en otro compuesto por letras, números, signos, etc de manera que no posee ninguna relación con la versión original.
* Base de datos: es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso.
* Software: programas y aplicaciones que se ejecutan en un sistema informático, computadora o dispositivo electrónico.
* Hardware: los elementos físicos que componen un sistema informático, computadora o dispositivo electrónico.
* Scrum: metodologia agil (rapido, dinámico y efectivo) para la gestión de proyectos.
* Sprint: periodo de tiempo determinado en que se realiza trabajo determinado para cumplir metas propuestas en el proyecto con la metodología Scrum.
* Planificación Sprint: proceso de planificación de un sprint donde se determinan metas, tiempos y otros aspectos relacionados a un Sprint.
* Dominio: nombre único que identifica a una subárea de internet.
* Framework: es un conjunto de herramientas y reglas que se utilizan para desarrollar software.
* NodeJS: entorno de ejecución de código abierto para JavaScript construido en el motor JavaScript V8 de Chrome.
* JavaScript: lenguaje de programación que permite implementar características complejas e interactividad en páginas web.

1. Referencias  
     
   Se debe mencionar toda aquella documentación que está relacionada y/o es necesaria para completar esta sección del documento, indicando: nombre de la documentación, fecha de creación y autor.

Agregar referencias en cuanto a lo monetario: openqube, Glassdor, sysarmy

| **Nombre documento** | **Fecha creación** | **Autor** |
| --- | --- | --- |
| Entrevista 1 | 25/03/2023 | AllaDevs |
| Entrevista 2 | 28/03/2023 | AllaDevs |
| Epicas | 28/03/2023 | AllaDevs |
| Cuestionario “Sitio web: ¡Oh my dog!” | 28/03/2023 | AllaDevs |
| SRS | 26/04/2023 | AllaDevs |
| Full Stack Developer (argentina) | 26/04/2023 | www.salaryexpert.com |
| Registrar dominio | 26/04/2023 | www.namecheap.com |
| Servidor | 26/04/2023 | vercel.com/pricing |
| Base de datos | 26/04/2023 | www.mongodb.com/pricing |
| Cotizacion dolar | 26/04/2023 | dolarhoy.com |

**2) Planes generales**

1. Entregables del proyecto  
     
   Se deben especificar todos los elementos que se entregarán al cliente junto con las fechas de entrega (tanto de documentación como del producto).

El desarrollo de la aplicación web consta de cinco entregas de las cuales dos son de documentos y las tres restantes siendo demos del estado actual del desarrollo, resaltando que en la última demo la aplicación se encontrará finalizada.

Se detallan los entregables en la tabla siguiente.

| **Entregable** | **Fecha de entrega** |
| --- | --- |
| Entrevista 1, Entrevista 2, Cuestionario y Épicas | 31/03/2023 |
| SRS, PGP y Pila de producto | 28/04/2023 |
| Demo 1 | 02/06/2023 |
| Demo 2 | 23/06/2023 |
| Demo 3 | 14/07/2023 |

1. Calendario y resumen del presupuesto  
     
   Se debe indicar el tiempo proyectado para entregar el producto final y el valor del presupuesto final (calculado en el punto 3c). En caso de que el cliente haya indicado alguna restricción con respecto a alguno de estos dos tópicos, debe aclararse en este apartado.

El desarrollo se realiza en el periodo comprendido entre el 10/03/2023 y el 14/07/2023. En la tabla siguiente se detalla un calendario con las etapas de desarrollo.

| **Etapa** | **Accion/es** | **Fecha** |
| --- | --- | --- |
| Elicitación de requerimientos | Entrevista 1 | 10/03/2023 |
| Entrevista 2 | 17/03/2023 |
| Especificación de requerimientos | Entrega de Entrevistas 1 y 2, Cuestionario y Épicas | 31/03/2023 |
| Entrega de SRS, PGP y Pila de producto | 28/04/2023 |
| Sprint 1 | Planeacion Sprint 1 | 05/05/2023 |
| Scrum diario 1-1 | 12/05/2023 |
| Scrum diario 1-2 | 19/05/2023 |
| Demo 1 | 02/06/2023 |
| Sprint 2 | Planeación Sprint 2 | 02/06/2023 |
| Scrum diario 2-1 | 09/06/2023 |
| Scrum diario 2-2 | 16/06/2023 |
| Demo 2 | 23/06/2023 |
| Sprint 3 | Planeación Sprint 3 | 23/06/2023 |
| Scrum diario 3-1 | 30/06/2023 |
| Scrum diario 3-2 | 07/07/2023 |
| Demo 3 | 14/07/2023 |

El presupuesto final es de 1239000 Pesos Argentinos para el desarrollo completo.

1. Plan del personal  
     
   Se debe identificar cantidad y tipo de personal requerido, como así también la duración para la cual son necesarios.

Para llevar a cabo el proyecto serán necesarios tres programadores con conocimiento, experiencia y determinación. Estos mismo deben ser capaces de realizar tanto la elicitación de los requerimientos, como el diseño lógico, arquitectura y gráfico producto de los requerimientos, como también la realización efectiva de los previos en código para la aplicación web. No es requerido especificación en ninguna área, solo una correcta organización de esfuerzos y tiempos mediante comunicación. Tres desarrolladores fullstack serían pertinentes en este proyecto.

El personal será contratado durante las 18 semanas, aprox. 4 meses, que se estimó el desarrollo completo.

**3) Presupuesto**

1. Principales actividades del proyecto  
     
   Se deben especificar todas las actividades a realizar para el proyecto. Esto incluye desde la elicitación de requerimientos hasta el mantenimiento (si lo hubiese). Para lograr una mejor estimación, la etapa referida al desarrollo podría desglosarse en submódulos.

El proyecto consiste en el desarrollo de una aplicación web desde cero para la veterinaria ¡Oh my dog!, para lo cual se requieren las siguientes actividades:

1. Elicitar requerimientos: elaboración y realización de entrevistas y cuestionario.
2. Especificar requerimientos: elaboración de especificaciones de requerimientos de manera formal, describiendo las funcionalidades y restricciones del sistema a desarrollar. Estos documentos además tienen la importancia de ser contratos con el cliente sobre el sistema solicitado.
3. Diseño conceptual: elaboración de diseños abstractos en cuanto al manejo de la información del sistema y bocetos gráficos de la aplicación en base a los requerimientos.
4. Desarrollo: la aplicación será en un entorno pertinente como lo es la plataforma NodeJS que facilita el desarrollo de aplicaciones web, donde se usará el conjunto de tecnologías/frameworks SvelteKit, tailwindcss, Prisma, MongoDB y TypeScript ya que estas brindan la adaptabilidad, agilidad y efectividad para realizar un proyecto de esta envergadura. Para lo cual se gestionará el proceso con la metodología de desarrollo Scrum, para lo cual se gestiona el desarrollo completo en los submódulos (no se requiere la completitud del submódulo para que funciones determinados elementos del mismo):
   1. Sistema de usuarios: contiene las funcionalidades respectivas a autenticación, manejo de cuentas.
   2. Sistema de turnos: contiene las funcionalidades respectivas a la gestión de turnos por parte de los empleados y la solicitud de turnos de los clientes.
   3. Sistema de servicios de terceros: contiene las funcionalidades respectivas a la publicitación de servicio de cuidado y paseo de perros.
   4. Sistema de donaciones: contiene las funcionalidades respectivas a la gestión de campañas de donación, descuentos por donaciones y donaciones libres de los clientes y no clientes.
   5. Sistema de sucursales: contiene las funcionalidades respectivas a la gestión de sucursales de la veterinaria por parte de los empleados así como la visualización por parte de clientes y no clientes.
   6. Sistema de cruzas: contiene las funcionalidades respectivas a gestión de perros para cruza por parte de los clientes.
   7. Sistema de encuentros: contiene las funcionalidades respectivas a la publicación de perros perdidos, encontrados tanto así como para ver los casos resueltos.
   8. Sistema de adopciones: contiene las funcionalidades respectivas a la gestión de dar en adopción un perro por parte de un cliente o no cliente, como también ver los disponibles a ser adoptados por los clientes y no clientes.
5. Despliegue efectivo: se realizará el despliegue final efectivo del sistema para que pueda ser usado en la veterinaria como consumido por los clientes de la misma de manera que funcione adecuadamente y de la manera esperada.
6. Mantenimiento: se realiza el monitoreo del funcionamiento del sistema luego que se encuentre activo para verificar su correcto funcionamiento.
7. Asignación de esfuerzo  
     
   Por cada una de las actividades definidas en el punto 3a se debe indicar: cantidad de personas destinadas a esa actividad, esfuerzo en horas unitario y esfuerzo en horas total (cantidad de personas \* esfuerzo unitario).   
     
   Al sumar los esfuerzos en horas totales, nos va a determinar la cantidad de horas destinadas al proyecto completo.

| **Actividad** | **Responsable/s (cantidad)** | **Esfuerzo unitario (hs)** | **Esfuerzo total (hs)** |
| --- | --- | --- | --- |
| Elicitar requerimientos | 3 | 5 | 15 |
| Especificar requerimientos | 3 | 8 | 24 |
| Diseño conceptual | 3 | 4 | 12 |
| Sistema de usuarios | 3 | 18 | 54 |
| Sistema de turnos | 3 | 7 | 21 |
| Sistema de servicios de terceros | 3 | 4 | 12 |
| Sistema de donaciones | 3 | 10 | 30 |
| Sistema de sucursales | 3 | 6 | 18 |
| Sistema de cruzas | 3 | 8 | 24 |
| Sistema de encuentros | 3 | 14 | 42 |
| Sistema de adopciones | 3 | 12 | 36 |
| Despliegue efectivo | 3 | 6 | 18 |
| Mantenimiento | 3 | 8 | 24 |

Total de horas entre todas las actividades: 330 hs.

1. Presupuesto final  
     
   El valor del presupuesto total se obtiene calculando:   
   **(cantidad de horas del proyecto \* precio por hora) + recursos adicionales**Cantidad de horas del proyecto: Obtenidas en el punto 3b  
   Precio por hora: Lo determina la empresa (es decir, ustedes)  
   Recursos adicionales: Se agrega si existe algún recurso adicional, el cual es necesario considerar en el presupuesto. En caso de que sí, detallar el nombre y precio del recurso. Ejemplos: dominio, hosting, etc.

El presupuesto se calculará en base a la suma de los presupuestos para los salarios de los desarrolladores más los costos de los recursos. El sueldo de cada desarrollador será el producto entre las horas empleadas y el salario por hora de la tabla a continuación:

| **Especialidad** | **Sueldo base** | **Extra** | **Total (salario por hora)** |
| --- | --- | --- | --- |
| Fullstack | 5 USD | 2 USD | 7 USD |

Precio total del personal = 7 USD/hs \* 330 hs = 2310 USD

Los recursos utilizados y su precio se describen en la tabla a continuación:

| **Nombre** | **Tiempo** | **Costo** |
| --- | --- | --- |
| Dominio ohmydog.vet | 1 año | 32 USD |
| Servidor | 1 mes | 20 USD |
| Base de datos | 1 mes | 5 USD |

Precio total de recursos a 1 año = 1\* 31.98 USD + 12 \* 20 USD + 12 \* 25 USD = 332 USD

Costo total en USD = 2310 USD + 332 USD = 2642 USD

Costo total en Pesos Argentinos (27/04/2023) = 2642 USD \* 469 Pesos Argentinos/USD = 1239000 Pesos Argentinos

**4) Riesgos**

Se deben mencionar al menos 6 riesgos indicando quién es el responsable de su tratamiento (integrante del equipo), probabilidad de que ocurra, impacto, mitigación (tareas a realizar para evitar que el riesgo ocurra) y plan de contingencia (tareas a realizar después de que el riesgo ocurrió).

Para estandarizar la medida de los riesgos se usará la métrica de impacto que describe como un riesgo impacta en el sistema. El impacto se medirá en los valores 1, 2, 3 y 4 y cada valor se describe en la tabla a continuación:

| **Valor** | **Alias** | **Interpretacion** |
| --- | --- | --- |
| 1 | Catastrofico | Cancelacion o suspension del proyecto |
| 2 | Serio | Reducción de rendimiento, retrasos en la entrega, excesos importante en costo |
| 3 | Tolerable | Reducciones mínimas de rendimiento, posibles retrasos, exceso en costo |
| 4 | Insignificante | Incidencia mínima en el desarrollo |

En el desarrollo y posterior a él se encuentran riesgos que pueden afectar en el sistema, para lo cual a continuación se detallan con un nombre, probabilidad de ocurrir, impacto en el sistema si sucede, las medidas mitigadoras para que no ocurra, estrategias de contingencia en caso de que suceda para minimizar el impacto inevitable y el miembro del equipo de desarrollo designado como responsable del riesgo.

* **Riesgo:** Menos código reusable del previsto.

**Probabilidad:** 30%.

**Impacto:** 3.

**Medidas mitigadoras:** Cumplir con buenas prácticas de programación como los criterios y heurísticas de diseño, para lograr un software modularizado y reusable.

**Plan de contingencia:** Aplicar técnicas de refactoring sobre el código.

**Responsable:** Bonoris Nicolas.

* **Riesgo:** Cambios en el equipo de desarrollo.

**Probabilidad:** 10%.

**Impacto:** 3.

**Medidas mitigadoras:** Fomentar un ambiente de trabajo agradable y alentar a los miembros del equipo a comunicarse abiertamente.

**Plan de contingencia:** : Evaluar la necesidad de traer nuevos miembros al equipo o redistribuir las responsabilidades actuales.

**Responsable:** Gallardo Lucas.

* **Riesgo:** Subestimación del tamaño del sistema.

**Probabilidad:** 40%.

**Impacto:** 2.

**Medidas mitigadoras:** Realizar la elicitación de requerimientos habiendo analizado el negocio y mercado. Analizar detalladamente la información recaudada y definir correctamente los requerimientos subyacentes. Investigar profundamente de ser necesario.

**Plan de contingencia:** Pautar con el cliente una nueva fecha de entrega del producto.

**Responsable:** Gallardo Lucas.

* **Riesgo:** Enfermedad del personal.

**Probabilidad:** 20%.

**Impacto:** 4.

**Medidas mitigadoras:** Organizar y gestionar una buena comunicación entre los miembros del equipo de desarrollo para comprender las tareas que los demás desarrolladores tienen que cumplir. Fomentar un ambiente de trabajo saludable y flexible, en el que se promueva el equilibrio entre la vida laboral y personal.

**Plan de contingencia:** Enviar un médico a domicilio, para procurar su pronta recuperación.

**Responsable:** Suarez Josue.

* **Riesgo**: Usuario no se adapta a la UI.

**Probabilidad:** 25%.

**Impacto:** 2.

**Medidas mitigadoras:** Pruebas de usabilidad, diseño y desarrollo iterativo, documentación clara de la interfaz de usuario.

**Plan de contingencia:** Capacitación adicional para el usuario, posible revisión del diseño de la UI.

**Responsable:** Bonoris Nicolas.

* **Riesgo:** Cambios de licencias de software..

**Probabilidad:** 10%.

**Impacto:** 1.

**Medidas mitigadoras:** Investigación exhaustiva de las licencias de software antes de su adopción, implementación de alternativas de software si es necesario.

**Plan de contingencia:** Cambio a alternativas de software, reasignación de recursos para adaptarse al nuevo software, posible retraso en el proyecto.

**Responsable:** Suarez Josue.

* **Riesgo:** Rendimiento del sistema no es el esperado.

**Probabilidad:** 20 %.

**Impacto:** 2.

**Medidas mitigadoras:** Pruebas de rendimiento durante desarrollo, diseño y desarrollo iterativo, revisión y optimización del código.

**Plan de contingencia:** Revisión y optimización del código, posible reasignación de recursos para mejorar el rendimiento.

**Responsable:** Gallardo Lucas.