**2 ANALISIS**

**2.1 Metodología de Desarrollo**

Para el desarrollo del proyecto se ha seleccionado la metodología **SCRUM**, una metodología ágil que permite una gestión eficiente y flexible del proyecto, enfocada en la entrega incremental de valor, colaboración continua con los interesados, y adaptación rápida a cambios y nuevas necesidades.

**¿Por qué SCRUM?**

* SCRUM es ideal para proyectos con requisitos dinámicos y en constante evolución, como el presente proyecto, que contempla no solo el desarrollo inicial sino también una extensión futura hacia una aplicación móvil a largo plazo (3 años después).
* Permite dividir el proyecto en iteraciones cortas llamadas **Sprints** (generalmente de 2 a 4 semanas), facilitando la entrega continua de funcionalidades parcialmente completas pero funcionales, lo que permite recibir retroalimentación frecuente y ajustar el rumbo.
* Fomenta la colaboración estrecha entre el equipo de desarrollo, clientes y stakeholders, lo que reduce riesgos y asegura que el producto final se ajuste a las expectativas y necesidades reales.

**Planificación temporal**

El desarrollo principal está planificado a una duración de **10 meses**, tiempo en el cual se llevarán a cabo múltiples Sprints, que incluyen análisis, diseño, desarrollo, pruebas y despliegue de la solución inicial. Esta solución cubrirá las funcionalidades esenciales definidas para el producto.

Posteriormente, se contempla que, tras un período de **3 años**, se emprenda el desarrollo de una **aplicación móvil** que complemente y extienda las funcionalidades del sistema original. Esta decisión estratégica responde a la necesidad de adaptarse a las tendencias tecnológicas y los hábitos de los usuarios, quienes cada vez demandan más acceso móvil.

**Integración de la visión a largo plazo en SCRUM**

* Durante los 10 meses de desarrollo inicial, se incluirán entregables y documentación que faciliten la transición y la escalabilidad futura hacia la aplicación móvil, asegurando una arquitectura modular, documentada y escalable.
* Se establecerán **backlogs diferenciados** para el producto principal y para la futura aplicación móvil, de modo que, una vez concluido el desarrollo inicial, se pueda planificar y ejecutar el proyecto móvil con un punto de partida sólido y con la experiencia acumulada.
* Se prevé la asignación de roles y equipos dedicados para el mantenimiento y evolución posterior del producto, para no afectar la estabilidad del sistema mientras se prepara el proyecto móvil.

**Roles en SCRUM para este proyecto**

* **Product Owner:** Representa los intereses del cliente y los usuarios finales, prioriza el backlog y valida los entregables.
* **Scrum Máster:** Facilita el proceso SCRUM, remueve impedimentos y asegura que el equipo siga las prácticas ágiles.
* **Equipo de desarrollo:** Grupo multidisciplinario encargado de diseñar, codificar, probar y documentar las funcionalidades en cada Sprint.

**Beneficios esperados al usar SCRUM**

* Mayor control y visibilidad del avance gracias a las reuniones diarias (Daily Stand-ups), revisiones y retrospectivas.
* Adaptabilidad a cambios en requerimientos sin perder productividad ni calidad.
* Entregas parciales funcionales que pueden ser evaluadas y usadas por los stakeholders para validar dirección.
* Facilita la incorporación de mejoras basadas en feedback real, lo que incrementa la satisfacción del usuario final.

**2.1.1 Planificación y programación de actividades (Sprint 1)**

El Sprint 1 representa la primera iteración del proyecto y está diseñado para sentar las bases fundamentales del desarrollo, definir el alcance inicial y validar los procesos de trabajo. Se establece una duración de **2 semanas** para este Sprint, dentro del marco general de 10 meses.

**Sprint 1: Configuración Inicial y Organización del Equipo**

**Objetivos**

* Presentación y alineación formal del proyecto.
* Definición de objetivos generales y visión del producto.
* Establecimiento de roles, responsabilidades y flujo de comunicación.
* Instalación y configuración de herramientas de trabajo colaborativo y gestión ágil.
* Recolección preliminar de requerimientos para formar el Product Backlog.
* Creación y priorización inicial del Product Backlog.

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Duración | Semana 1 | Semana 2 |
| Reunión inicial y Kick-off | **1 día** | **- Presentación formal del proyecto. - Exposición de visión, alcance general, metas a largo plazo. - Introducción al marco Scrum. - Asignación preliminar de roles.** |  |
| Definición de estructuras de comunicación y trabajo | **1 día** | **- Acuerdo sobre canales de comunicación (Slack, Teams, etc.). - Normas para reuniones y documentación. - Frecuencia de dailys, reviews y retros.** |  |
| Configuración de herramientas | **2 días** | **- Instalación de Jira, Trello u otro gestor ágil. - Creación y estructuración de repositorios (GitHub, GitLab). - Configuración de entornos virtuales.** |  |
| Recolección inicial de requerimientos del producto | **2 días** | **- Entrevistas o análisis documental. - Priorización con el cliente. - Revisión de funcionalidades base esperadas.** | **- Registro preliminar en backlog.** |
| Redacción y priorización inicial del Product Backlog | **3 días** | **- Conversión de requerimientos en historias de usuario con criterios de aceptación. - Aplicación de MoSCoW para definir prioridades.** | **- Validación del backlog con el equipo Scrum.** |
| Revisión y retrospectiva Sprint 1 | **1 día** |  | **- Evaluación del cumplimiento del sprint. - Ajustes a herramientas o dinámicas si hubo dificultades.** |

**Sprint 2: Análisis, Diseño de Arquitectura y Plan de Pruebas**

**Objetivos**

* Análisis detallado de requerimientos del backlog inicial.
* Identificación de flujos clave mediante casos de uso y UML.
* Diseño preliminar de la arquitectura del sistema.
* Selección de tecnologías y definición de módulos/componentes.
* Desarrollo del plan de pruebas para funcionalidades iniciales**.**

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Duración | Semana 1 | Semana 2 |
| Análisis detallado de requerimientos | **4 días** | **- Desglose de historias en tareas técnicas. - Modelado de casos de uso y flujo de datos. - Validación con stakeholders y Product Owner.** | **- Documentación de riesgos y restricciones funcionales/técnicas.** |
| Diseño preliminar de arquitectura del sistema | **3 días** | **- Esbozo de arquitectura general (cliente-servidor, n capas, etc.). - Definición de componentes y tecnologías (React, Node.js, PostgreSQL, etc.).** | **- Establecimiento de interfaces para futuras extensiones (como aplicación móvil).** |
| Planificación del plan de pruebas inicial | **2 días** | **- Definición del tipo de pruebas (unitarias, integración, funcionales, carga). - Diseño de primeros casos de prueba.** | **- Asignación de responsables de pruebas. - Estimación de esfuerzo y herramientas (Selenium, Postman, JMeter, etc.).** |
| Revisión y retrospectiva Sprint 2 | **1 día** |  | **- Evaluación del diseño y del análisis. - Revisión del plan de pruebas. - Identificación de mejoras para el Sprint 3.** |

**Sprint 3: Implementación de Base de Datos y Módulo de Autenticación**

**Objetivos**

* Establecer la base técnica del sistema mediante la creación y configuración de la base de datos.
* Diseñar e implementar el módulo de autenticación de usuarios (login/registro).
* Garantizar las conexiones iniciales entre backend y base de datos.
* Realizar pruebas unitarias sobre las funcionalidades desarrolladas.
* Iniciar procesos formales de control de calidad y revisión de código.

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Duración | Semana 1 | Semana 2 |
| Planificación del Sprint 3 | **1 día** | **- Revisión de la retrospectiva anterior. - Definición de historias a desarrollar. - Estimación en Planning Poker.** |  |
| Diseño e implementación de la BDD | **3 días** | **- Diseño lógico de la base de datos. - Diagrama Entidad-Relación. - Elección del SGBD (PostgreSQL).** | **- Creación de tablas y relaciones en el motor. - Pruebas de integridad referencial. - Documentación del modelo.** |
| Desarrollo del módulo de autenticación (Parte 1) | **3 días** | **- Diseño de interfaces de login y registro (formulario). - Configuración de rutas en back-end (API REST).** | **- Conexión al motor de base de datos. - Validación de entradas, hashing de contraseñas, control de sesiones.** |
| Pruebas unitarias iniciales | **2 días** | **- Desarrollo de pruebas para autenticación. - Configuración del entorno (Jest, Mocha o PyTest).** | **- Ejecución y análisis de resultados. - Corrección de errores detectados.** |
| Revisión y control de calidad | **1 día** |  | **- Peer review y análisis de código estático. - Documentación técnica del código escrito.** |

**Sprint 4: Desarrollo del Dashboard y Demo Funcional Inicial**

**Objetivos**

* Desarrollar la interfaz principal del sistema (dashboard).
* Conectar el dashboard con la base de datos y back-end.
* Integrar autenticación con navegación segura.
* Realizar pruebas unitarias del dashboard.
* Entregar una demo funcional inicial para revisión con el Product Owner.
* Evaluar desempeño del equipo y retroalimentar proceso.

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Duración | Semana 1 | Semana 2 |
| Desarrollo del dashboard inicial | **3 días** | **- Maquetación de interfaz (React, Angular, HTML). - Definición de estructura de navegación y vistas.** | **- Conexión con servicios del backend (APIs). - Visualización de datos dinámicos. - Manejo de sesiones.** |
| Pruebas unitarias del dashboard | **2 días** | **- Definición de casos de prueba para el dashboard.** | **- Ejecución y documentación de pruebas. - Registro de errores e incidencias.** |
| Demo funcional inicial | **1 día** |  | **- Presentación al Product Owner. - Recolección de observaciones y mejoras.** |
| Revisión de código y documentación | **1 día** |  | **- Validación de estándares de codificación. - Actualización de la documentación técnica.** |
| Retrospectiva del Sprint 4 | **1 día** |  | **- Análisis del Sprint. - Identificación de problemas. - Plan de mejora para Sprint 5.** |

**Sprint 5**

**Objetivos del Sprint 5**

* Implementar el módulo de gestión de usuarios (CRUD).
* Añadir validaciones de seguridad en la autenticación y navegación.
* Fortalecer la documentación técnica y funcional.

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Duración | Semana 1 | Semana 2 |
| Planificación del Sprint 5 | **1 día** | **- Selección de historias de usuario. - Revisión de entregables previos. - Definición de tareas y estimación.** |  |
| Desarrollo módulo de gestión de usuarios | **4 días** | **- Maquetación de formularios (alta, baja, edición, consulta). - Implementación lógica backend para CRUD. - Integración con base de datos.** | **- Validación de inputs y restricciones (campos requeridos, formatos). - Pruebas funcionales en cada operación.** |
| Validaciones y seguridad | **2 días** | **- Validaciones frontend en formularios. - Middleware para autenticación de rutas.** | **- Prevención de errores (inyección SQL, accesos indebidos). - Encriptación de contraseñas si no se hizo antes.** |
| Documentación técnica del Sprint | **1 día** |  | **- Registro de estructuras de datos, rutas, endpoints. - Actualización de README y manual.** |

**Sprint 6**

**Objetivos del Sprint 6**

* Integrar el panel administrativo básico con control de roles.
* Realizar pruebas de integración entre vistas, lógica y base de datos.
* Refinar diseño responsivo en interfaces desarrolladas.
* Fortalecer la documentación técnica y funcional.

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Duración | Semana 1 | Semana 2 |
| Planificación del Sprint 6 | **1 día** | **- Selección de historias de usuario. - Revisión entregables del Sprint 5. - Definición de tareas y estimación.** |  |
| Panel administrativo y control de roles | **3 días** | **- Creación de vistas del panel de administración. - Gestión de acceso por tipo de usuario (admin, cliente, etc.).** | **- Integración de permisos a nivel de navegación y componentes. - Ajuste de rutas seguras.** |
| Pruebas de integración funcional | **2 días** | **- Diseño de pruebas para flujos completos (login → acceso → acción). - Verificación de funcionamiento entre vistas, lógica y base de datos.** | **- Documentación de resultados. - Corrección de errores encontrados.** |
| Refinamiento diseño responsivo | **2 días** | **- Ajustes y mejoras en interfaces para adaptabilidad en dispositivos.** |  |
| Revisión y retrospectiva Sprint 6 | **1 día** |  | **- Revisión final del Sprint. - Evaluación del equipo. - Feedback Product Owner. - Ajustes para Sprint 7.** |

**Sprint 7**

**Objetivos del Sprint 7**

* Implementar funciones especializadas: filtros y búsquedas avanzadas.
* Generar reportes exportables (PDF, Excel).
* Preparar la base para la mejora de la experiencia de usuario (UX/UI).

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Duración | Semana 1 | Semana 2 |
| Planificación del Sprint 7 | **1 día** | **- Selección de historias especializadas. - Estimación y distribución de tareas (Scrum Board).** |  |
| Desarrollo de filtros y buscador avanzado | **3 días** | **- Diseño interfaz para filtros (nombre, fecha, categoría, etc.). - Implementación lógica backend y consultas SQL optimizadas.** | **- Validación de búsquedas. - Manejo casos sin resultados y respuestas rápidas.** |
| Generación de reportes (PDF/Excel) | **3 días** | **- Integración de librerías para generación de archivos (PDFKit, jsPDF, ExcelJS). - Selección de datos exportables.** | **- Implementación del botón de descarga. - Pruebas de formato, encabezados y estilos.** |
| Ajustes iniciales de UX/UI | **1 día** | **- Revisión preliminar de experiencia de usuario.** |  |

**Sprint 8**

**Objetivos del Sprint 8**

* Desarrollar notificaciones internas y alertas básicas.
* Integrar lógica avanzada de negocio: validaciones cruzadas y criterios de ordenamiento.
* Mejorar la experiencia del usuario con microinteracciones.
* Preparar y realizar demo extendida para pruebas con usuarios finales.

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Duración | Semana 1 | Semana 2 |
| Planificación del Sprint 8 | **1 día** | **- Selección de historias y planificación.** |  |
| Notificaciones internas y alertas | **2 días** | **- Diseño visual de alertas (toast, modales). - Integración con eventos clave (creación, error, expiración).** | **- Lógica para almacenamiento y estado de notificaciones. - Pruebas de rendimiento con múltiples notificaciones.** |
| Validaciones cruzadas y lógica avanzada | **2 días** | **- Definición e implementación de reglas de negocio (validaciones entre fechas, consistencia entre formularios).** | **- Implementación de advertencias inteligentes. - Manejo de errores y retroalimentación visual.** |
| Ajustes de UX/UI y microinteracciones | **1 día** | **- Refinamiento visual de módulos previos. - Aplicación de microinteracciones para mejorar navegación.** | **- Pruebas en navegadores y dispositivos.** |
| Demo extendida y revisión Sprint 8 | **1 día** |  | **- Presentación al Product Owner. - Documentación de feedback y ajustes para próximos sprints.** |

**Sprint 9**

**Objetivos del Sprint 9**

* Ejecutar pruebas piloto controladas con usuarios reales.
* Documentar la retroalimentación técnica y funcional.
* Optimizar el rendimiento del sistema (tiempos de carga, consultas, UI).

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Duración | Semana 1 | Semana 2 |
| Planificación del Sprint 9 | 1 día | - Selección de historias de mejora. - Planeación de pruebas piloto. - Definición de métricas clave (KPI: tiempo de carga, errores, feedback). |  |
| Preparación y ejecución de prueba piloto | 3 días | - Selección grupo de prueba (usuarios internos o clientes). - Creación de guías de uso y formularios de retroalimentación. - Recolección de observaciones. | - Revisión comportamiento del sistema en condiciones reales. - Registro de bugs y oportunidades de mejora. |
| Optimización de rendimiento | 2 días | - Revisión consultas lentas a la base de datos. - Minificación de recursos (CSS, JS). - Lazy loading en vistas complejas. |  |

**Sprint 10**

**Objetivos del Sprint 10**

* Fortalecer las validaciones de seguridad adicionales (OWASP).
* Mejorar la experiencia de navegación (fluidez y accesibilidad).
* Iniciar y avanzar en la documentación para despliegue e instalación.
* Realizar demo interna y retrospectiva del Sprint.

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Duración | Semana 1 | Semana 2 |
| Optimización de rendimiento | 1 día | - Implementación de cache local. - Revisión de tiempos de respuesta y análisis de logs. |  |
| Fortalecimiento de seguridad | 3 días | - Revisión de prácticas seguras (OWASP Top 10). - Validación de entrada de datos más estricta. - Autenticación reforzada con tokens JWT / sesiones seguras. | - Protección contra CSRF/XSS. - Análisis de exposición de datos sensibles. |
| Mejoras de navegación y experiencia UX | 2 días | - Revisión de flujos de navegación confusos. - Rediseño de botones, retroalimentación visual (tooltips, loaders). | - Ajustes de accesibilidad (etiquetas, contraste, navegación por teclado). |
| Documentación para despliegue | 1 día | - Inicio del manual de instalación para entornos desarrollo, pruebas y producción. | - Definición de requerimientos mínimos del sistema. - Checklist de pre-lanzamiento. |
| Demo interna y retrospectiva | 1 día |  | - Presentación del sistema mejorado al Product Owner. - Evaluación cumplimiento de KPIs. - Retroalimentación del equipo. - Definición de objetivos para Sprint siguiente. |

**Sprint 11**

**Objetivos del Sprint 11**

* Implementar pruebas automatizadas para garantizar calidad continua.
* Configurar el entorno de preproducción para el despliegue inicial.
* Comenzar la documentación técnica básica.

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Duración | Semana 1 | Semana 2 |
| Planificación del Sprint 11 | **1 día** | **- Revisión del backlog técnico y ajustes finales. - Definición de historias relacionadas con QA, pruebas y documentación.** |  |
| Diseño e implementación de pruebas automáticas | **4 días** | **- Selección de framework (PyTest, Selenium, JUnit, etc.). - Creación de scripts para pruebas unitarias y funcionales.** | **- Cobertura de pruebas para módulos críticos (login, CRUD, reportes, filtros). - Ejecución automática con CI/CD (GitHub Actions, GitLab CI, etc.).** |
| Configuración del entorno de preproducción | **2 días** | **- Instalación y despliegue en entorno espejo a producción. - Configuración de base de datos de pruebas.** | **- Verificación de conectividad, seguridad y rendimiento en preproducción.** |

**Sprint 12**

**Objetivos del Sprint 12**

* Establecer mecanismos de monitoreo y logging del sistema.
* Completar documentación técnica y manuales de usuario.
* Realizar revisión final y retrospectiva de la fase intensiva de desarrollo web.

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Duración | Semana 1 | Semana 2 |
| Integración de herramientas de monitoreo y logging | **2 días** | **- Configuración de Grafana, Kibana, Prometheus, etc. - Activación de logs en puntos críticos.** | **- Pruebas de alertas automáticas ante errores críticos (email, logs, Slack).** |
| Documentación técnica y manuales de usuario | **3 días** | **- Redacción manual de instalación, actualización y mantenimiento. - Diagramas de arquitectura y flujos técnicos.** | **- Manual de usuario para personal operativo o clientes. - Plantilla de FAQs y resolución de errores comunes.** |
| Revisión final y retrospectiva del Sprint 12 | **1 día** |  | **- Revisión general del sistema con stakeholders. - Verificación de KPIs. - Cierre oficial de la fase web. - Planificación preliminar para versión móvil.** |

**Sprint 13**

**Objetivos del Sprint 13**

* Realizar el despliegue final del sistema en ambiente de producción.
* Activar el sistema de monitoreo y soporte post-lanzamiento.
* Comenzar a recopilar feedback real de usuarios finales mediante encuestas y canales de atención.

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Duración | Semana 1 | Semana 2 |
| Planificación del Sprint 13 | **1 día** | **- Revisión del backlog post-productivo. - Definición de entregables de transición. - Alineación con partes interesadas.** |  |
| Despliegue a producción | **2 días** | **- Migración de datos finales. - Configuración de DNS/SSL y servicios en la nube. - Activación de la aplicación al público.** | **- Pruebas en producción (smoke tests, monitoreo de carga).** |
| Habilitación de monitoreo y soporte activo | **2 días** | **- Activación de alertas en tiempo real. - Integración con canales de soporte (email, formulario, chat). - Definición de tiempos de respuesta (SLA).** | **- Revisión de logs y performance real. - Revisión de errores reportados y gestión de incidencias tempranas.** |
| Diseño y publicación de encuestas | **1 día** | **- Diseño de encuestas de satisfacción (CSAT, NPS). - Publicación mediante la plataforma.** |  |

**Sprint 14**

**Objetivos del Sprint 14**

* Recopilar y analizar feedback real de usuarios.
* Evaluar cumplimiento de KPIs operativos y de satisfacción.
* Documentar lecciones aprendidas, incidencias y planificar próximas iteraciones.
* Establecer estrategia de mantenimiento correctivo y evolutivo.

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Duración | Semana 1 | Semana 2 |
| Recolección y análisis de encuestas | **2 días** | **- Recolección y análisis inicial de respuestas. - Identificación de puntos de fricción en la experiencia de usuario. - Recomendaciones preliminares.** |  |
| Evaluación de KPIs de sistema | **2 días** | **- Evaluación de: tiempo de respuesta, errores críticos, uptime, interacción por módulo. - Comparación contra línea base definida en sprints anteriores.** | **- Identificación de brechas y propuestas de solución para próximos sprints o actualizaciones.** |
| Lecciones aprendidas y plan de mejora | **1 día** |  | **- Reunión final de sprint con stakeholders. - Presentación de logros, errores y acciones de mejora. - Identificación de tareas para roadmap evolutivo y posible integración móvil.** |

**Sprint 15**

**Objetivos del Sprint 15**

* Aplicar mejoras funcionales menores derivadas del feedback recibido.
* Ajustar y mejorar la experiencia de usuario en formularios y flujos problemáticos.
* Realizar pruebas de usabilidad (UX testing) para validar los cambios.
* Planificar y priorizar optimizaciones de rendimiento para módulos clave.

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Duración | Semana 1 | Semana 2 |
| Planificación del Sprint 15 | **1 día** | **- Revisión del backlog evolutivo. - Selección y priorización de mejoras funcionales menores. - Ajuste de objetivos conforme roadmap.** |  |
| Implementación de mejoras funcionales menores | **3 días** | **- Ajustes en formularios y validaciones. - Corrección de flujos detectados como problemáticos. - Mejoras en navegación.** | **- Ejecución de pruebas con usuarios (UX testing).** |

**Sprint 16**

**Objetivos del Sprint 16**

* Optimizar el rendimiento general del sistema.
* Mejorar tiempos de carga con técnicas de caching y lazy loading.
* Ajustar consultas a la base de datos para mejorar eficiencia.
* Establecer monitoreo de rendimiento con métricas y reportes.

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Duración | Semana 1 | Semana 2 |
| Optimización de rendimiento | **3 días** | **- Minimización y compresión de archivos estáticos (JS, CSS). - Implementación de caching local y lazy loading.** | **- Revisión y ajuste de consultas a base de datos. - Monitoreo de mejora de tiempos comparado con línea base.** |

**Sprint 17**

**Objetivos del Sprint 17**

* Refactorizar el código base para mejorar la escalabilidad y mantenibilidad.
* Modularizar componentes y limpiar código obsoleto.
* Actualizar la documentación técnica para reflejar la nueva estructura.

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Duración | Semana 1 | Semana 2 |
| Refactorización de código | **2 días** | **- Análisis y mejora de estructuras internas. - Modularización del código y eliminación de componentes obsoletos.** | **- Actualización completa de la documentación técnica.** |

**Sprint 18**

**Objetivos del Sprint 18**

* Implementar mejoras en accesibilidad y experiencia de usuario (UX/UI).
* Asegurar el cumplimiento de estándares WCAG 2.1.
* Realizar pruebas A/B para validar los cambios.
* Fortalecer la seguridad del sistema con prácticas actualizadas.
* Preparar la infraestructura para futuras integraciones, especialmente la versión móvil.

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Duración | Semana 1 | Semana 2 |
| Mejoras de accesibilidad y UX/UI | **2 días** | **- Ajustes en contraste, tipografías y navegación por teclado. - Simplificación y mejora de formularios. - Verificación con estándares WCAG 2.1.** | **- Realización de pruebas A/B con usuarios. - Documentación de resultados y recomendaciones.** |
| Fortalecimiento de seguridad | **2 días** | **- Revisión y ajuste de roles y permisos. - Implementación y actualización de headers de seguridad. - Auditoría con OWASP Top 10.** | **- Pruebas automatizadas de penetración (Zap Proxy, SonarQube, etc.).** |
| Retrospectiva y planificación futura | **1 día** |  | **- Evaluación del impacto de las mejoras. - Identificación de tareas para backlog de mantenimiento y preparación para aplicación móvil.** |

**Sprint 19**

**Objetivos del Sprint 19**

* Iniciar desarrollo e integración de la versión móvil del sistema.
* Adaptar funcionalidades clave para la plataforma móvil.
* Diseñar interfaces responsivas y amigables para dispositivos móviles.
* Establecer comunicación y sincronización con backend existente.
* Probar conectividad y funcionalidades básicas en entornos móviles.

**Actividades y Duración**

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Duración | Semana 1 | Semana 2 |
| Planificación Sprint 19 | 1 día | - Selección de funcionalidades prioritarias para móvil. - Definición de tareas y estimación. |  |
| Diseño UI/UX móvil | 3 días | - Creación de wireframes y prototipos responsivos. - Revisión de patrones de navegación móvil. | - Validación y ajustes con equipo y stakeholders. |
| Desarrollo inicial versión móvil | 4 días | - Adaptación de módulos clave (login, dashboards, formularios). - Implementación de sincronización con backend. | - Pruebas funcionales básicas en dispositivos reales o emuladores. |

**Sprint 20**

**Objetivos del Sprint 20**

* Continuar y finalizar desarrollo de versión móvil.
* Realizar pruebas integrales y corrección de errores.
* Mejorar rendimiento y experiencia móvil.
* Preparar documentación y guía para usuarios móviles.
* Establecer plan de soporte y mantenimiento específico para móvil.

**Actividades y Duración**

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Duración | Semana 1 | Semana 2 |
| Desarrollo avanzado y ajustes | 4 días | - Finalización de funcionalidades móviles restantes. - Optimización de rendimiento y consumo de recursos. | - Corrección de bugs y ajustes de UI/UX según pruebas. |
| Pruebas integrales y QA | 3 días | - Pruebas funcionales, de compatibilidad y usabilidad. - Revisión de feedback de testers y usuarios piloto. |  |
| Documentación y soporte móvil | 2 días | - Creación de manuales y guías de usuario para móvil. - Definición de flujos de soporte técnico. |  |
| Retrospectiva y planificación futura | 1 día |  | - Evaluación general. - Planificación para mantenimiento y nuevas mejoras. |

**ORGANIGRAMA FUNCIONAL (SCRUM ORIENTADO A DESARROLLO WEB-MÓVIL)**

**FUNCIONES Y RESPONSABILIDADES**

|  |  |  |
| --- | --- | --- |
| Rol | Funciones Principales | Sprints donde interviene |
| Product Owner | - Define visión del producto - Prioriza historias de usuario - Valida entregables - Recoge feedback | Todos los sprints |
| Scrum Máster | - Facilita las ceremonias Scrum - Remueve impedimentos - Vela por cumplimiento de metodología ágil | Todos los sprints |
| Líder de Proyecto | - Coordina recursos - Define cronograma - Controla entregables - Supervisa cumplimiento técnico | Sprints 1–20 |
| Desarrollador Front-End | - Implementa UI en web - Aplica diseño responsivo - Maneja validaciones, formularios, filtros | Sprints 2, 3, 4, 5, 6, 9, 10, 19 |
| Desarrollador Back-End | - Crea APIs - Administra base de datos - Gestiona roles, seguridad, lógica de negocio | Sprints 2, 3, 4, 5, 6, 9, 10, 20 |
| Desarrollador Móvil | - Implementa versión móvil - Sincroniza con backend - Adapta funcionalidades clave a móvil | Sprints 19, 20, 21+ (evolutivos) |
| QA Tester | - Diseña y ejecuta pruebas - Automatiza tests - Evalúa usabilidad y accesibilidad | Sprints 4, 5, 6, 7, 9, 10, 19, 20 |
| Soporte Técnico / TI | - Configura entornos (pre-prod, prod) - Da soporte post-lanzamiento - Monitorea servicios y logs | Sprints 6, 7, 10, 20 |

**Tabla de Identificación de Partes Interesadas (Stakeholders)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nombre del Stakeholder | Rol / Cargo | Interés Principal en el Proyecto | Nivel de Influencia | Nivel de Interés | Forma de Participación / Involucramiento |
| Product Owner | Representante del cliente | Quiero asegurar que el producto cumpla con los requerimientos funcionales y objetivos del negocio | Alto | Alto | Priorización del backlog, validación de entregables, feedback |
| Scrum Máster | Facilitador del equipo ágil | Garantizar cumplimiento de la metodología Scrum y eliminar obstáculos del equipo | Medio | Alto | Facilita reuniones, gestiona impedimentos, asegura buenas prácticas |
| Líder de Proyecto | Coordinador general del proyecto | Supervisar el cumplimiento del cronograma, alcance, calidad y costo | Alto | Alto | Planificación, monitoreo, toma de decisiones críticas |
| Equipo de Desarrollo | Devs Frontend, Backend, Móvil | Entregar funcionalidades completas y de calidad dentro de cada Sprint | Medio | Alto | Participación en desarrollo, pruebas, documentación |
| QA Tester / Equipo de QA | Control de calidad | Validar que el sistema funcione correctamente y sin errores | Medio | Alto | Diseño y ejecución de pruebas manuales y automatizadas |
| Usuarios Finales (Piloto) | Clientes reales / internos | Usabilidad del sistema, facilidad de acceso, rendimiento, experiencia de usuario | Bajo | Alto | Participación en pruebas piloto, encuestas, entrevistas |
| Soporte Técnico / TI | Administradores de sistemas | Garantizar el correcto funcionamiento de servidores, despliegue y monitoreo | Medio | Medio | Configuración de entornos, respaldo, seguridad y logs |
| Alta Dirección / Directivos | Gerentes, inversionistas | Retorno de inversión, eficiencia operativa, valor agregado | Alto | Medio | Revisión de KPIs, resultados de impacto, decisiones estratégicas |
| Clientes Institucionales | Empresas u organizaciones aliadas | Acceso al sistema, integración con otros servicios, personalización de funcionalidades | Medio | Alto | Reuniones de validación, levantamiento de requerimientos específicos |
| Equipo Legal | Abogados, auditores | Cumplimiento de normativas legales, protección de datos, accesibilidad y términos de uso | Medio | Medio | Revisión de políticas, términos, avisos de privacidad |
| Comunidad de Usuarios (post) | Usuarios externos post lanzamiento | Satisfacción con el sistema, calidad del servicio, experiencia en diversos dispositivos | Bajo | Alto | Retroalimentación por encuestas, soporte, foros o reportes |
| Partners Tecnológicos | Proveedores de servicios TI | Integración de APIs, soporte de herramientas de monitoreo, hosting, librerías externas | Bajo | Medio | Acuerdos técnicos, soporte, actualizaciones de compatibilidad |

**Recursos del Proyecto**

**1. Hardware**

El hardware requerido se compone de computadoras de desarrollo, un servidor físico para el procesamiento principal, y un servidor en la nube encargado de la distribución de datos a los usuarios finales. Esta arquitectura híbrida garantiza eficiencia, escalabilidad y alta disponibilidad del sistema.

**🔹 1.1 Estaciones de Trabajo (Computadoras de Desarrollo)**

|  |  |
| --- | --- |
| Característica | Detalle técnico |
| Cantidad | 18 unidades |
| Procesador | Intel Core i7 12700H (12 núcleos, 20 hilos, hasta 4.7GHz) |
| Memoria RAM | 32 GB DDR4 |
| Almacenamiento | SSD NVMe 1 TB |
| Sistema Operativo | Windows 11 Pro 64 bits / Ubuntu 22.04 LTS |
| Tarjeta gráfica (opcional) | NVIDIA RTX 3050 (para pruebas de interfaces gráficas exigentes) |
| Conectividad | Wi-Fi 6, Ethernet Gigabit |
| Pantalla | 15.6” FHD IPS |
| Periféricos | Teclado mecánico, ratón óptico, audífonos con micrófono |

Estas estaciones de trabajo serán utilizadas por los desarrolladores frontend, backend y testers, optimizadas para entornos de desarrollo complejos (IDEs, contenedores, entornos virtualizados).

**🔹 1.2 Servidor Físico (Back-End de Procesamiento)**

|  |  |
| --- | --- |
| Característica | Detalle técnico |
| Ubicación | Centro de datos local / oficina técnica |
| Procesador | AMD EPYC 7402P, 24 núcleos, 48 hilos, hasta 3.35GHz |
| Memoria RAM | 256 GB DDR4 ECC |
| Almacenamiento | RAID 10 – 4x SSD de 2 TB NVMe |
| Sistema Operativo | Linux CentOS / Ubuntu Server 22.04 LTS |
| Fuente de poder redundante | Sí |
| Sistemas de respaldo | UPS de respaldo y respaldo automático semanal |
| Uso principal | Procesamiento de lógica del sistema, cálculos, consultas complejas, API REST, tareas en segundo plano |
| Seguridad local | Firewall perimetral, segmentación de red y autenticación por VPN |

Este servidor asume las tareas pesadas del sistema, como operaciones de bases de datos, procesamiento de grandes volúmenes de datos y tareas automatizadas, asegurando bajo tiempo de respuesta al servidor web.

**🔹 1.3 Servidor en la Nube (Interfaz de Usuario)**

|  |  |
| --- | --- |
| Característica | Detalle técnico |
| Proveedor | Google Cloud Platform (GCP) |
| Instancia recomendada | 4 vCPU, 16 GB RAM, SSD 50 GB |
| Sistema Operativo | Ubuntu Server 22.04 LTS |
| Servicios utilizados | Computer Engine (VM), Cloud Storage, Load Balancer |
| Uso principal | Servir vistas al usuario, entregar datos procesados, administrar el tráfico web |
| Redundancia / Alta disponibilidad | Sí (multi-zona) |
| Escalabilidad | Horizontal (autoescalado por tráfico) |
| Seguridad en la nube | HTTPS con certificado SSL, autenticación OAuth2, firewalls GCP |

Este servidor está diseñado para distribuir contenido a los usuarios de forma eficiente, segura y escalable. Toda la carga intensiva se delega al servidor físico, lo que permite mantener este servidor ligero y receptivo.

**2. Software**

**🔹 2.1 Software de Desarrollo**

|  |  |  |
| --- | --- | --- |
| Tipo | Herramienta | Función principal |
| IDE / Editor | Visual Studio Code, IntelliJ, PyCharm | Desarrollo del código |
| Control de versiones | Git + GitHub / GitLab | Versionado y colaboración |
| Contenedores | Docker + Docker Compose | Entornos reproducibles |
| Integración continua | GitHub Actions / GitLab CI/CD | Automatización de pruebas y despliegues |
| Frameworks Web | React.js (frontend), Node.js / Django | Interfaces y lógica del sistema |

**🔹 2.2 Software de Bases de Datos**

|  |  |
| --- | --- |
| Base de Datos | PostgreSQL / MySQL |
| Uso | Almacenamiento de datos relacionales, consultas y transacciones seguras |

**🔹 2.3 Herramientas de Pruebas**

|  |  |
| --- | --- |
| Tipo | Herramienta |
| Pruebas unitarias | PyTest, JUnit, Mocha |
| Pruebas funcionales | Selenium, Cypress |
| Pruebas de carga | Apache JMeter |

**🔹 2.4 Software de Seguridad**

|  |  |
| --- | --- |
| Herramienta | Uso |
| OWASP ZAP | Análisis de vulnerabilidades |
| SonarQube | Análisis de calidad de código estático |
| Fail2Ban, UFW | Protección contra accesos indebidos |

**🔹 2.5 Herramientas de Monitoreo**

|  |  |
| --- | --- |
| Herramienta | Función |
| Grafana | Monitoreo de rendimiento en tiempo real |
| ELK Stack (Elasticsearch, Logstash, Kibana) | Logging estructurado y análisis de eventos |
| Google Cloud Monitoring | Métricas y alertas en entornos cloud |

**2.1.5 Identificación de Requerimientos y Alcance del Proyecto**

**Descripción General del Proyecto**

El proyecto consiste en el desarrollo e implementación de una plataforma digital híbrida (web y futura versión móvil) destinada a ofrecer un servicio integral que incluye funcionalidades clave como gestión de datos, visualización de información, procesamiento automatizado, interacción con usuarios y escalabilidad para futuras integraciones. Está orientado a resolver una necesidad específica del usuario final mediante una experiencia segura, eficiente y accesible.

**Requerimientos Funcionales (RF)**

|  |  |
| --- | --- |
| Código | Requerimiento Funcional |
| RF01 | El sistema debe permitir a los usuarios autenticarse mediante un formulario de login seguro. |
| RF02 | El usuario debe poder consultar información personalizada a través de un panel dinámico. |
| RF03 | El administrador podrá gestionar usuarios, roles y permisos desde un módulo de administración. |
| RF04 | El sistema debe registrar logs de acciones críticas del usuario y del sistema. |
| RF05 | Se deben realizar validaciones automáticas en formularios y flujos de datos. |
| RF06 | La plataforma debe emitir alertas o notificaciones ante eventos predefinidos. |
| RF07 | El usuario podrá acceder a reportes descargables en PDF o Excel. |
| RF08 | Debe existir una interfaz para recopilar y visualizar retroalimentación del usuario final. |
| RF09 | El sistema debe poder integrarse en un futuro con una aplicación móvil nativa. |

**Requerimientos No Funcionales (RNF)**

|  |  |
| --- | --- |
| Código | Requerimiento No Funcional |
| RNF01 | El sistema debe responder en un tiempo inferior a 2 segundos para consultas estándar. |
| RNF02 | La disponibilidad del sistema debe ser del 99.9% mensual. |
| RNF03 | La plataforma debe cumplir con estándares de accesibilidad WCAG 2.1 nivel AA. |
| RNF04 | La seguridad debe ajustarse a los lineamientos del OWASP Top 10. |
| RNF05 | Debe soportar al menos 500 usuarios concurrentes en condiciones estándar. |
| RNF06 | El sistema debe estar construido bajo principios de escalabilidad y modularidad. |
| RNF07 | Toda la comunicación entre cliente y servidor debe estar cifrada mediante HTTPS. |
| RNF08 | Los datos sensibles deben almacenarse utilizando cifrado AES de 256 bits. |
| RNF09 | Debe integrarse con herramientas de monitoreo para detección de errores y alertas tempranas. |

**Alcance del Proyecto**

|  |  |
| --- | --- |
| Categoría | Descripción del Alcance |
| Cobertura funcional | Plataforma web con módulo de autenticación, visualización de datos, administración de usuarios, reportes y dashboard. |
| Plataformas objetivo | Web (responsive) como prioridad. Versión móvil en planeación para fases posteriores. |
| Usuarios finales | Administradores, operadores internos y usuarios externos con perfiles diferenciados. |
| Interoperabilidad | Conectividad mediante API RESTful, arquitectura preparada para integración con otros sistemas o aplicaciones móviles. |
| Seguridad | Cifrado en tránsito y en reposo, control de accesos, autenticación reforzada, validación de entradas. |
| Despliegue | Servidor físico para procesamiento interno + servidor cloud para disponibilidad pública. |
| Límites del alcance | No contempla desarrollo de aplicación móvil en esta fase. No incluye soporte multilenguaje ni multimoneda inicialmente. |

**Alcance fuera del proyecto (Exclusiones)**

|  |  |
| --- | --- |
| Exclusión | Justificación |
| Desarrollo de aplicación móvil nativa | Será desarrollado en una fase futura tras validación del sistema web. |
| Integración con sistemas externos de facturación | No es requerido en esta etapa del proyecto. |
| Soporte multilenguaje | Se enfocará en idioma español en esta versión inicial. |
| Desarrollo offline (sin conexión) | La plataforma requiere conexión a internet para funcionar correctamente. |

**2.1.6 Entregables EDT / WBS (Estructura de Desglose del Trabajo)**

La **EDT (Estructura de Desglose del Trabajo)** o **WBS (Work Breakdown Structure)** es una representación jerárquica del alcance del proyecto, organizada en paquetes de trabajo que facilitan la gestión, asignación y seguimiento de cada componente del desarrollo. Los entregables se derivan directamente de esta estructura y permiten establecer hitos clave, dependencias y cronograma.

|  |  |  |  |
| --- | --- | --- | --- |
| Nivel | Código WBS | Entregable | Descripción |
| 1 | 1.0 | Inicio del proyecto | Documentación base para formalizar y arrancar el proyecto. |
| 2 | 1.1 | Acta constitutiva del proyecto | Documento formal que autoriza el inicio del proyecto, define objetivos y alcance. (“Plantilla del acta de constitución de proyecto (Project Charter)”) |
| 2 | 1.2 | Designación de roles y equipo | Asignación de miembros del equipo y responsabilidades. |
| 1 | 2.0 | Planeación | Actividades y documentos para guiar la ejecución técnica y administrativa. |
| 2 | 2.1 | Plan de trabajo general (Scrum/Gantt) | Calendario, sprints, dependencias y línea base. |
| 2 | 2.2 | Identificación de stakeholders | Tabla de interesados con influencia y participación. |
| 2 | 2.3 | Matriz RACI | Distribución clara de roles: Responsable, Aprobador, Consultado, Informado. |
| 1 | 3.0 | Ejecución técnica | Desarrollo del sistema y sus funcionalidades principales. |
| 2 | 3.1 | Arquitectura del sistema | Documento y diagrama técnico del diseño estructural. |
| 2 | 3.2 | Módulos funcionales (login, dashboard, reportes, filtros) | Codificación de cada módulo en back y front. |
| 2 | 3.3 | Pruebas técnicas y automatizadas | QA funcional, pruebas unitarias e integración continua. |
| 1 | 4.0 | Despliegue y documentación | Puesta en marcha del sistema y documentación de usuario. |
| 2 | 4.1 | Manual de instalación y configuración | Instrucciones técnicas para servidores físico y en la nube. |
| 2 | 4.2 | Manual de usuario | Guía operativa para usuarios finales y administrativos. |
| 2 | 4.3 | Presentación y cierre de fase | Evaluación de KPIs, demo funcional y cierre de desarrollo web. |

**2.1.6.1 Acta Constitutiva del Proyecto**

|  |  |
| --- | --- |
| Elemento | Contenido |
| Nombre del Proyecto | Plataforma Digital Integral (Web + Proyección Móvil) |
| Propósito del Proyecto | Desarrollar una solución tecnológica que automatice procesos clave, mejore la experiencia de usuario y permita futuras integraciones móviles. |
| Product Owner (PO) | Bicho |
| Scrum Máster / PM | Bicho |
| Equipo de desarrollo | Backend, Frontend, QA, UI/UX, Seguridad, Infraestructura |
| Alcance Inicial | Desarrollo completo de plataforma web operativa, con arquitectura preparada para versión móvil futura. |
| Presupuesto estimado | 5,000,000 |
| Duración estimada | 10 meses divididos en 18 sprints |
| Criterios de éxito | Cumplimiento funcional, usabilidad validada, tiempo de respuesta <2s, disponibilidad >99.9% |
| Fecha de aprobación | 1 junio del 2025 |
| Firmas de aprobación | Product Owner, Scrum Máster, Representantes legales |

**2.1.6.2 Matriz de Roles y Responsabilidades (RACI)**

La matriz RACI permite asignar claramente quién es **Responsable** (R), **Aprobador** (A), **Consultado** (C) e **Informado** (I) en cada actividad clave del proyecto:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Actividad / Entregable | Product Owner | Scrum Máster / PM | Desarrollador Backend | Frontend / UI-UX | QA / Testing | Infraestructura |
| Planificación de Sprints | C | A/R | I | I | I | I |
| Desarrollo de Módulos | I | C | R | R | C | I |
| Diseño UX/UI | I | C | I | R | C | I |
| Implementación de seguridad | I | C | R | C | C | R |
| Pruebas funcionales y automatizadas | I | C | C | C | R | I |
| Configuración de entorno y despliegue | I | C | C | I | C | R |
| Documentación técnica y manuales | I | R | C | C | C | C |
| Demo y validación de entregables | A/R | R | I | I | I | I |

**Leyenda:**

* **R**: Responsable (quien ejecuta)
* **A**: Aprobador (quien valida)
* **C**: Consultado (quien da opinión experta)
* **I**: Informado (quien debe conocer el avance)

**3. Requerimientos**

**3.1 Requerimientos funcionales**

Los requerimientos funcionales describen las funcionalidades que el sistema debe ofrecer para satisfacer las necesidades del usuario y cumplir los objetivos del proyecto. (“Requisitos funcionales - Requisitos para”)

**Principales requerimientos funcionales:**

1. **Gestión de usuarios**
   * Registro, autenticación y administración de perfiles.
   * Gestión de roles y permisos para control de acceso (administradores, usuarios finales, etc.).
   * Recuperación y restablecimiento seguro de contraseñas.
2. **Módulo de autenticación y seguridad**
   * Autenticación basada en tokens JWT.
   * Implementación de sesiones seguras con control de expiración.
   * Validación de entrada de datos para evitar inyección y ataques XSS.
3. **Gestión de contenido y datos**
   * CRUD (Crear, Leer, Actualizar, Eliminar) para entidades principales del sistema.
   * Filtros y búsquedas avanzadas con criterios múltiples (fecha, categoría, usuario).
   * Generación de reportes exportables en formatos PDF.
4. **Notificaciones y alertas**
   * Sistema de notificaciones internas con estados (vistas/no vistas).
   * Alertas visuales para eventos críticos, errores y expiraciones.
   * Configuración y personalización de notificaciones por usuario.
5. **Optimización y rendimiento**
   * Implementación de caching y lazy loading para mejorar tiempos de carga.
   * Minimización y compresión de recursos estáticos (JS, CSS).
   * Monitoreo de performance y análisis continuo para identificar cuellos de botella.
6. **Pruebas y calidad**
   * Pruebas unitarias y funcionales automatizadas integradas en pipeline CI/CD.
   * Pruebas piloto controladas con usuarios reales para validar funcionalidades.
   * Pruebas de penetración para detectar vulnerabilidades.
7. **Despliegue y mantenimiento**
   * Configuración de entorno preproducción para pruebas finales.
   * Despliegue en ambiente productivo con monitoreo activo.
   * Documentación técnica y manuales de usuario actualizados.
8. **Interfaz y experiencia de usuario**
   * Diseño responsivo y accesible, compatible con múltiples dispositivos y navegadores.
   * Microinteracciones para mejorar la navegación y feedback visual.
   * Ajustes de accesibilidad conforme a estándares WCAG 2.1. (Anexos, Seccion 2)
9. **Integración futura**
   * Preparación de infraestructura y arquitectura para la versión móvil.
   * Modularidad del código para facilitar integración y escalabilidad.

**3.2 Requerimientos no funcionales**

Los requerimientos no funcionales definen criterios y restricciones sobre el desempeño, usabilidad, confiabilidad y otros atributos cualitativos del sistema.

**Principales requerimientos no funcionales:**

1. **Rendimiento**
   * Tiempo máximo de respuesta para operaciones críticas no mayor a 2 segundos.
   * Capacidad para manejar al menos 500 usuarios concurrentes sin degradación significativa.
   * Disponibilidad del sistema mínima del 99.9% en producción.
2. **Seguridad**
   * Protección contra ataques comunes según OWASP Top 10. (Anexos, Sección 3)
   * Encriptación segura de datos sensibles tanto en tránsito (TLS) como en reposo. (Anexos, Sección 4)
   * Control de acceso granular basado en roles y permisos.
3. **Escalabilidad**
   * Arquitectura modular para facilitar la incorporación de nuevas funcionalidades.
   * Capacidad para escalar horizontalmente en servidores físicos y en la nube.
4. **Mantenibilidad**
   * Código documentado y estructurado según estándares de desarrollo.
   * Uso de herramientas de control de versiones y seguimiento de incidencias.
   * Documentación técnica clara y actualizada.
5. **Usabilidad**
   * Interfaz intuitiva y accesible para usuarios con diferentes niveles de conocimiento.
   * Soporte para navegación por teclado y compatibilidad con lectores de pantalla.
   * Feedback inmediato para acciones de usuario (errores, confirmaciones, cargas).
6. **Compatibilidad**
   * Soporte para los principales navegadores web actuales (Chrome, Firefox, Edge, Safari).
   * Adaptabilidad a dispositivos móviles y tabletas.
7. **Portabilidad**
   * Capacidad para desplegarse en diferentes entornos: desarrollo, pruebas, producción.
   * Compatibilidad con plataformas de nube (AWS, Azure, GCP) y servidores físicos. (CHECAR)
8. **Confiabilidad**
   * Mecanismos de respaldo y recuperación ante fallos.
   * Registro detallado de logs para auditoría y diagnóstico.

**3.3 Reglas de negocio**

Las reglas de negocio determinan las políticas, restricciones y procedimientos que deben cumplirse en el funcionamiento del sistema para garantizar coherencia y alineación con objetivos empresariales o institucionales.

**Ejemplos de reglas de negocio:**

1. **Gestión de usuarios**
   * Solo los usuarios con rol administrador pueden modificar roles y permisos.
   * El bloqueo de usuarios se activa tras 5 intentos fallidos consecutivos de inicio de sesión.
2. **Validaciones de datos**
   * Los campos de fecha deben respetar la relación lógica: fecha inicio ≤ fecha fin.
   * Los reportes no podrán incluir datos de usuarios inactivos por más de 6 meses.
3. **Notificaciones**
   * Las notificaciones críticas deben mostrarse inmediatamente y repetirse hasta que sean leídas.
   * Los usuarios pueden configurar qué tipos de alertas desean recibir por correo electrónico.
4. **Seguridad**
   * Todas las contraseñas deben cumplir con una longitud mínima de 10 caracteres y complejidad (mayúsculas, minúsculas, números y símbolos).
   * Las sesiones expiran tras 30 minutos de inactividad.
5. **Acceso a datos**
   * El acceso a información sensible está restringido según el nivel de permiso asignado.
   * Los datos de auditoría deben mantenerse almacenados al menos 2 años.
6. **Despliegue**
   * Los despliegues en producción solo se realizarán tras aprobación del Product Owner.
   * Cada despliegue debe ir acompañado de un plan de rollback para restaurar el sistema en caso de fallo.

**3.4 Requisitos de usuario**

Los requisitos de usuario reflejan las expectativas, necesidades y preferencias de los usuarios finales para la correcta interacción con el sistema.

**Principales requisitos de usuario:**

1. **Facilidad de uso**
   * La interfaz debe ser sencilla, con un diseño limpio y organizado.
   * Las operaciones básicas deben realizarse con un mínimo número de pasos.
2. **Accesibilidad**
   * El sistema debe ser usable por personas con discapacidades visuales, auditivas o motoras.
   * Soporte para navegación mediante teclado y lectores de pantalla.
3. **Disponibilidad**
   * Los usuarios deben poder acceder al sistema en cualquier momento, salvo en mantenimientos programados.
4. **Personalización**
   * Posibilidad de configurar notificaciones, temas visuales y preferencias personales.
5. **Ayuda y soporte**
   * Disponibilidad de manuales, tutoriales y FAQs integrados.
   * Canales de soporte accesibles (chat, correo, formularios).
6. **Feedback y transparencia**
   * El sistema debe proporcionar retroalimentación clara tras cada acción (éxito, error, proceso en curso).
   * Visualización clara de estados, errores y alertas.
7. **Seguridad y privacidad**
   * Confianza en que sus datos personales están protegidos y no se compartirán sin consentimiento.
   * Control sobre la visibilidad de su información y actividad dentro del sistema.

**Consideraciones para elegir la arquitectura:**

| **Criterio** | **Detalle relevante para el proyecto** |
| --- | --- |
| **Carga de procesamiento** | Tienes un servidor físico que realizará el procesamiento fuerte. |
| **Interacción de usuarios** | Servidor en la nube para mostrar datos, acceso simultáneo desde web. |
| **Escalabilidad** | Se debe poder crecer en usuarios y funcionalidades sin impacto fuerte. |
| **Seguridad** | Autenticación, roles, protección OWASP, encriptación. |
| **Mantenibilidad** | Código modular, con facilidad para futuras actualizaciones o versiones móviles. |
| **Disponibilidad** | Sistema con alta disponibilidad y monitoreo. |
| **Tecnologías actuales** | Uso de frameworks modernos, integración con CI/CD, pruebas automatizadas. |

**Opciones comunes y evaluación**

|  |  |  |  |
| --- | --- | --- | --- |
| Arquitectura | Descripción breve | Ventajas relevantes | Desventajas / Consideraciones |
| Monolítica tradicional | Todo en un solo servidor y aplicación integrada. | Fácil de desarrollar al inicio. | Poco escalable, difícil mantenimiento y despliegue. |
| Cliente-Servidor (2 capas) | Servidor para lógica y base de datos, cliente para UI. | Simple, divide responsabilidades básicas. | Escalabilidad limitada para muchos usuarios. |
| 3 capas (Presentación, Lógica, Datos) | Separación clara en UI, lógica y base de datos. | Mejor organización, fácil mantenimiento. | Requiere infraestructura y coordinación. |
| Microservicios | Módulos independientes desplegados y escalables por separado. | Escalabilidad, resiliencia, despliegue ágil. | Mayor complejidad, requiere orquestación y monitoreo. |
| Arquitectura basada en la nube híbrida | Procesamiento pesado en servidor físico local + interfaz en la nube. | Optimiza recursos y balancea cargas. | Necesita sincronización y gestión de latencia. |

**Arquitectura para el Proyecto**

**Arquitectura híbrida de 3 capas con servidor físico para procesamiento y nube para presentación:**

* **Capa de presentación:** Aplicación web alojada en la nube, accesible desde cualquier dispositivo, gestionando la interfaz de usuario, validaciones básicas y consultas livianas.
* **Capa de lógica / procesamiento:** Servidor físico local encargado de la lógica compleja, cálculos, procesamiento de datos, seguridad y acceso a la base de datos.
* **Capa de datos:** Base de datos centralizada en el servidor físico, con backups regulares y replicación si se requiere.
* **Monitoreo y logging:** Integrados con herramientas que corren en la nube para supervisar el estado del servidor físico y la aplicación web.
* **CI/CD y pruebas:** Integradas para permitir despliegues ágiles y control de calidad automatizado.

**Justificación**

* Aprovecha la potencia del servidor físico para procesos intensivos.
* Utiliza la nube para garantizar alta disponibilidad y acceso rápido a la UI.
* Facilita escalabilidad gradual, manteniendo el control sobre datos sensibles localmente.
* Mejora seguridad al segmentar funciones y limitar accesos.
* Es compatible con las mejores prácticas modernas (modularidad, pruebas, despliegue continuo).
* Permite evolución hacia microservicios o versión móvil en el futuro.

**3.5 Fases de la Metodología Design Thinking**

La **metodología Design Thinking** es un enfoque centrado en el usuario que permite resolver problemas complejos mediante la comprensión profunda de las necesidades reales, la experimentación iterativa y el diseño colaborativo de soluciones. Su estructura se basa en cinco fases secuenciales pero flexibles: **Empatizar, Definir, Idear, Prototipar y Evaluar**.

Esta metodología resulta especialmente valiosa en proyectos tecnológicos e innovadores, ya que facilita la **alineación entre los requerimientos técnicos y humanos**, asegurando que el producto o sistema desarrollado **responda a necesidades auténticas y verificadas** por los usuarios.

**3.5.1 Comprensión y aplicación del paso “Empatizar”**

**Descripción general:**

La fase de **Empatizar** tiene como objetivo principal **conocer al usuario en profundidad**, no solo desde el punto de vista funcional, sino también emocional, conductual y aspiracional. Implica una inmersión profunda en su realidad para identificar **deseos, frustraciones, motivaciones y contextos de uso**.

Para lograrlo, se utilizan técnicas cualitativas como entrevistas, observación directa, encuestas, sesiones de co-creación y pruebas preliminares.

**Herramientas aplicadas:**

A continuación, se detallan las herramientas utilizadas durante la aplicación de esta fase en el contexto del sistema en desarrollo:

**Mapa de necesidades**

|  |  |
| --- | --- |
| Elemento | Descripción |
| ¿Qué es? | Una herramienta que organiza las necesidades explícitas e implícitas de los usuarios detectadas en entrevistas, encuestas o pruebas de campo. |
| ¿Cómo se aplicó? | Se elaboró a partir de encuestas a usuarios finales, entrevistas con profesionales del sector y análisis de plataformas similares. |
| Resultado | Identificación de necesidades clave como: agilidad de acceso, centralización de datos, notificaciones personalizadas y facilidad de navegación. |

**Mapa de empatía**

|  |  |
| --- | --- |
| Dimensión | Aplicación específica |
| ¿Qué dice? | “Necesito un sistema que no sea complicado y que me diga cuándo actuar.” |
| ¿Qué piensa? | “Ojalá no tenga que depender de soporte técnico cada vez que ingreso.” |
| ¿Qué siente? | Frustración por sistemas anteriores lentos o con errores frecuentes. |
| ¿Qué hace? | Usa plataformas limitadas o poco intuitivas. |
| ¿Qué ve? | Comparaciones con otros sistemas, referencias por otros usuarios o empresas. |
| ¿Qué escucha? | Opiniones de colegas o experiencias previas negativas. |

**Buyer Persona**

|  |  |
| --- | --- |
| Perfil | Descripción |
| Nombre ficticio: | Laura Rodríguez |
| Edad: | 34 años |
| Ocupación: | Responsable de gestión administrativa en PyME |
| Tecnología: | Usuario medio, maneja Excel, correo, CRM básicos |
| Necesidades clave: | Facilidad de acceso, claridad en notificaciones, reportes exportables |
| Puntos de dolor: | Tiempos de carga largos, sistemas confusos, soporte lento |

**Test de usuario preliminar**

|  |  |  |
| --- | --- | --- |
| Fase del prototipo | Interacción observada | Feedback recibido |
| Primer wireframe | Dificultades para encontrar botones clave, flujo confuso | “No sé a dónde debo ir para generar un reporte.” |
| Prototipo funcional simple | Mejora en navegación, pero dificultad en filtros avanzados | “¿Y si quiero filtrar por fecha exacta y por categoría a la vez?” |
| Versión con mejoras | Fluidez de navegación, comprensión de jerarquías, claridad en resultados visuales | “Ahora se siente más intuitivo, parece que entiende cómo trabajo.” |

El desarrollo de esta fase permitió construir una **visión empática, validada y centrada en el usuario final**, lo que asegura una solución **funcionalmente relevante y emocionalmente aceptable**. Se definieron claramente las necesidades principales, los perfiles prioritarios y las condiciones del entorno en el que operará el sistema.

Esta comprensión profunda **fundamentó todo el ciclo de desarrollo posterior** y reforzó la aplicación de principios de accesibilidad, seguridad, claridad y eficiencia en el diseño y arquitectura general del sistema.

**3.5.3 Comprensión del paso de “Idear”**

La etapa de “**Idear**” dentro del proceso de diseño centrado en el usuario representa el momento en que, con base en los hallazgos obtenidos durante las fases de *Empatizar* y *Definir*, se generan soluciones creativas y viables a los problemas detectados. Esta fase requiere un enfoque divergente, donde se prioriza la cantidad y variedad de ideas, antes de converger hacia aquellas más factibles, innovadoras y alineadas con las necesidades reales de los usuarios.

**En el contexto del proyecto:**

Durante esta fase, el equipo interdisciplinario involucrado en el desarrollo de la plataforma web para el cuidado de mascotas participó en sesiones estructuradas de lluvia de ideas, análisis comparativo de plataformas similares (benchmarking) y ejercicios de co-creación con usuarios clave (dueños de mascotas, veterinarios, y personal de tiendas especializadas).

Se exploraron distintos enfoques para resolver los retos más críticos identificados en las fases anteriores, tales como:

* La dificultad de encontrar profesionales veterinarios certificados cercanos.
* El seguimiento desorganizado del historial clínico de las mascotas.
* La falta de integración entre servicios de comercio electrónico y atención médica veterinaria.
* El bajo nivel de accesibilidad en plataformas existentes.
* La fragmentación de servicios en aplicaciones distintas.

**Metodologías empleadas:**

* **Brainstorming guiado**, orientado a generar ideas sin restricciones.
* **Mapa de Afinidad**, para clasificar las ideas según funcionalidad, valor al usuario y viabilidad técnica.
* **Técnica SCAMPER**, que permitió reformular soluciones existentes y adaptarlas al contexto del sistema.
* **Matriz de Impacto/Esfuerzo**, para priorizar las funcionalidades y características a implementar.

**Resultados clave:**

La fase de ideación permitió delimitar un conjunto de funcionalidades innovadoras y técnicamente alcanzables para la primera versión del sistema, incluyendo:

* Un **buscador filtrado de clínicas y servicios veterinarios** según localización y especialidad.
* Un **módulo de historial clínico digital compartida** entre usuario y profesional.
* Un **carrito de compras inteligente** que recomienda productos según el perfil de la mascota.
* La futura **integración de geolocalización, escaneo QR** y **notificaciones personalizadas** en la aplicación móvil.

**Referencias**

* **Instituto Nacional de Estadística y Geografía (INEGI). (2021). *Estadísticas a propósito de los animales de compañía*.** [**https://www.inegi.org.mx/contenidos/saladeprensa/aproposito/2021/animales2021\_Nal.pdf**](https://www.inegi.org.mx/contenidos/saladeprensa/aproposito/2021/animales2021_Nal.pdf)
* **Euromonitor International. (2022). *Pet Care in Mexico*.** [**https://www.euromonitor.com/pet-care-in-mexico/report**](https://www.euromonitor.com/pet-care-in-mexico/report)
* **Grand View Research. (2023). *Pet Tech Market Size, Share & Trends Analysis Report by Product, By Application, By Region, And Segment Forecasts, 2023*. (“Pet Tech Market Size, Statistics - Share Forecast 2028”)** [**https://www.grandviewresearch.com/industry-analysis/pet-tech-market**](https://www.grandviewresearch.com/industry-analysis/pet-tech-market)
* **Asociación Mexicana de Médicos Veterinarios Especialistas en Pequeñas Especies (AMMVEPE). (2020). *Retos y oportunidades del sector veterinario en México*.** [**https://www.ammvepe.com.mx/**](https://www.ammvepe.com.mx/)
* **Pew Research Center. (2019). *"Smartphone Ownership Is Growing Rapidly Around the World, but Not Always Equally*." (“Smartphone Ownership Is Growing Rapidly Around the World, but Not ...”)** [**https://www.pewresearch.org/global/2019/02/05/smartphone-ownership-is-growing-rapidly-around-the-world-but-not-always-equally/**](https://www.pewresearch.org/global/2019/02/05/smartphone-ownership-is-growing-rapidly-around-the-world-but-not-always-equally/)
* **International Data Corporation (IDC). (2023). *Latin America Mobile Device Forecast*.** [**https://www.idc.com/**](https://www.idc.com/)
* **Secretaría de Agricultura y Desarrollo Rural (SADER). (2020). *Bienestar animal y tenencia responsable de mascotas*. Gobierno de México.** [**https://www.gob.mx/agricultura/es/articulos/bienestar-animal-y-tenencia-responsable-de-mascotas**](https://www.gob.mx/agricultura/es/articulos/bienestar-animal-y-tenencia-responsable-de-mascotas)
* **Instituto Nacional de Estadística y Geografía (INEGI). (2022). Encuesta Nacional de los Hogares: Tenencia de mascotas en México. Recuperado de** [**https://www.inegi.org.mx**](https://www.inegi.org.mx/)
* **Petco México. (2024). Tienda en línea de productos para mascotas. Recuperado de** [**https://www.petco.com.mx**](https://www.petco.com.mx/)
* **DogHero. (2024). Servicios de cuidado y hospedaje para mascotas. Recuperado de** [**https://www.doghero.com.mx**](https://www.doghero.com.mx/)
* **Vets Smart. (2023). Software de gestión clínica para veterinarias. Recuperado de** [**https://www.vetsmart.com.mx**](https://www.vetsmart.com.mx/)
* **Mercado Libre México. (2024). Sección mascotas. Recuperado de** [**https://www.mercadolibre.com.mx/mascotas**](https://www.mercadolibre.com.mx/mascotas)
* **PetDesk. (2024). Plataforma de gestión veterinaria y recordatorios de salud. Recuperado de** [**https://www.petdesk.com**](https://www.petdesk.com/)
* **Pawtrack. (2024). Rastreador GPS para mascotas con aplicación integrada. Recuperado de** [**https://www.pawtrack.com**](https://www.pawtrack.com/)
* **Rover. (2024). Servicios de hospedaje y cuidado de mascotas. Recuperado de** [**https://www.rover.com**](https://www.rover.com/)
* **Whistle Labs. (2023). Monitor de salud para mascotas con tecnología wearable. Recuperado de** [**https://www.whistle.com**](https://www.whistle.com/)
* **Vetster. (2024). Plataforma de telemedicina veterinaria. Recuperado de** [**https://www.vetster.com**](https://www.vetster.com/)
* **Statista Research Department. (2023). Growth of pet care market worldwide 2023-2030. Recuperado de** [**https://www.statista.com**](https://www.statista.com/)
* **Deloitte. (2022). Digital trends in pet care and e-commerce. Recuperado de** [**https://www2.deloitte.com**](https://www2.deloitte.com/)
* **García, M., & López, J. (2021). Humanización de mascotas y su impacto en la demanda de servicios digitales. *Revista Mexicana de Estudios Socioculturales*, 15(3), 45-62.**
* **Pérez, A. (2020). Adopción tecnológica en servicios veterinarios: retos y oportunidades en Latinoamérica. *Tecnología y Sociedad*, 8(2), 101-115.**
* W3C. (2018). *Web Content Accessibility Guidelines (WCAG) 2.1*. World Wide Web Consortium (W3C). <https://www.w3.org/TR/WCAG21/>
* OWASP Foundation. (2021). *OWASP Top 10 - 2021: The Ten Most Critical Web Application Security Risks*. <https://owasp.org/Top10/>
* ISO/IEC. (2013). *ISO/IEC 27001:2013 — Information technology — Security techniques — Information security management systems — Requirements*. (“INTERNATIONAL ISO/IEC STANDARD 27001”) (“INTERNATIONAL ISO/IEC STANDARD 27001”)
* National Institute of Standards and Technology. (2019). *Security Requirements for Cryptographic Modules (FIPS PUB 140-3)*. U.S. Department of Commerce.
* Mozilla. (2024). *Security/Server-Side TLS* – <https://wiki.mozilla.org/Security/Server_Side_TLS>