

Índice de Contenidos

- 1. Introducción
- 2. Modelo de Plan de Negocio
 - 2.1. Resumen Ejecutivo
 - 2.2. Entorno en el que se realizará la actividad de la empresa
 - 2.2.1. Entorno económico y sociocultural
 - 2.2.2. Entorno normativo
 - 2.2.3. Entorno tecnológico
 - 2.3. Análisis de Mercado
 - 2.3.1. Oportunidad de Mercado
 - 2.3.2. Competencia
 - 2.3.3. Análisis DAFO
 - 2.4. Modelo de negocio
 - 2.4.1. Propuesta de valor
 - 2.4.2. Segmento de clientes
 - 2.4.3. Canales de Distribución
 - 2.4.4. Fuentes de Ingresos
 - 2.5. Operaciones y Estructura Organizativa
 - 2.5.1. Estructura Organizativa
 - 2.5.2. Proceso de Desarrollo
 - 2.5.3. Localización de la Empresa: Rural vs Ciudad.
 - 2.5.3.1 Asentamiento Previsto y Criterios para su Selección
 - 2.5.3.2 Instalaciones y Comunicaciones
 - 2.5.4. Recursos Humanos
 - 2.5.4.1 Organización de Recursos y Medios Técnicos y Humanos
 - 2.5.4.2 Organigrama. Estructura de Dirección y Gestión
 - 2.6. Estrategia de marketing y ventas
 - 2.6.1. Estrategia de marketing
 - 2.6.2. Estrategia de ventas
 - 2.7. Finanzas y Proyecciones
 - 2.7.1. Inversión Inicial
 - 2.7.2. Proyecciones Financieras
 - 2.8. Aspectos Formales de la Empresa
 - 2.8.1. Elección de la Forma Jurídica
 - 2.8.1.1. Criterios de Elección para la Forma Jurídica
 - 2.8.1.2. Borrador de los Estatutos Sociales
 - 2.8.2. Protección Jurídica

- 2.8.2.1. Medidas de Protección Jurídica
- 2.8.2.2. Patentes, Marcas, Licencias y Permisos
- 2.8.3. Medios de Prevención de Riesgos Laborales
 - Legislación Aplicable en Trabajo Remoto
- 2.9. Innovación
 - 2.9.1. Aspectos Innovadores
 - 2.9.2. Elementos de Sostenibilidad de la Innovación
- 3. Planificación temporal de la puesta en marcha
 - Fase 1: Preparación (Mes 1)
 - Fase 2: Aspectos Legales y Administrativos (Mes 2)
 - Fase 3: Configuración Tecnológica (Mes 3)
 - Fase 4: Marketing y Presencia Online (Mes 4)
 - Fase 5: Inicio de Actividades (Mes 5)
 - Fase 6: Consolidación y Ajustes (Mes 6 en adelante)
- 4. Proceso productivo: Auto-repair, un proyecto tipo
 - 4.1. Captación del Cliente
 - 4.1.1. Identificación de Potenciales Clientes
 - 4.1.2. Estrategias de Marketing y Publicidad
 - 4.1.3. Presentación de Servicios y Propuestas
 - 4.2. Análisis y Requerimientos
 - 4.2.1. Reuniones con el Cliente para Comprender sus Necesidades
 - 4.2.2. Documentación de Requerimientos Funcionales y no Funcionales
 - 4.2.3. Análisis de Viabilidad del Proyecto
 - 4.2.3. Propuesta de Proyecto Cerrado o Continuo
 - 4.3. Diseño del Software
 - 4.3.1. Creación de Prototipos y Wireframes
 - 4.3.2. Diseño de la Arquitectura del Sistema
 - 4.3.3. Definición de la Interfaz de Usuario (UI) y la Experiencia del Usuario (UX)
 - 4.3.4. Elección de Tecnologías y Herramientas
 - 4.3.4.1. IDE (Entorno de Desarrollo Integrado):
 - 4.3.4.2. Frontend
 - 4.3.4.3. Backend
 - 4.3.4.4. Capa de Persistencia
 - 4.3.4.5. Control de Tareas
 - 4.3.4.6. Control de Versiones
 - 4.3.4.7. Automatización de Integración y Despliegue (CI/CD)
 - 4.3.4.8. Plataforma de Hosting
 - 4.3.4.9. Trazas, métricas y logs
 - 4.4. Desarrollo en opción de Proyecto Cerrado
 - 4.4.1 Infraestructura de Desarrollo
 - 4.4.1.1. Gestión de Tareas: GitHub Projects
 - 4.4.1.2. Control de Versiones: Git en GitHub
 - 4.4.1.3. Entornos de Desarrollo: Desarrollo, Preproducción y Producción
 - 4.4.2. Ciclo de vida del desarrollo
 - 4.4.2.1. Metodología de trabajo: Gitflow
 - 4.4.2.2. Integración continua: GitHub Actions

- 4.4.2.3. Despliegue continuo: Plataforma Render
 - 4.4.2.4. Documentación global: GitHub Pages
 - 4.4.3. Codificación
 - 4.4.3.1. Codificación del Software según Diseño
 - 4.4.3.2. Implementación de Pruebas Unitarias y de Integración
 - 4.4.4. Entrega Inicial: MVP (Minimum Viable Product)
 - 4.4.4.1. Implementación del Software en el Entorno de Producción
 - 4.4.4.2. Capacitación del Cliente en el Uso del Software
 - 4.4.4.3. Recopilación de Comentarios del Cliente
 - 4.4.4.4. Identificación de correcciones y mejoras
 - 4.4.4.5. Desarrollo e Implementación de correcciones y mejoras
 - 4.4.5. Entrega final
 - 4.4.5.1. Recopilación de Comentarios del Cliente
 - 4.4.5.2. Identificación de Oportunidades de Mejora
 - 4.4.6. Mantenimiento y Soporte
 - 4.4.6.1. Monitoreo Continuo del Rendimiento del Software
 - 4.4.6.2. Resolución de Problemas y Errores
 - 4.4.6.3. Actualizaciones de Seguridad y Mejoras Menores
 - 4.4.7 Documentación
 - 4.4.7.1. Elaboración de Manuales de Usuario
 - 4.4.7.2. Documentación Técnica y de Código
 - 4.4.7.3. Actualización Constante de la Documentación
- 5. Bibliografía y Recursos
 - Git
 - Visual Studio Code (VSCode)
 - Node.js y NestJS
 - OpenTelemetry
 - Gitflow
 - React
 - Render
 - GitHub Projects
 - GitHub Pages
 - GitHub Actions
 - PostgreSQL (PostgresDB)
 - MongoDB
 - JavaScript
- 6. Anexos
 - 6.1. Anexo 1: Código fuente
 - 6.2. Anexo 2: Entorno productivo

1. Introducción

En el dinámico escenario empresarial contemporáneo, la evolución constante de la tecnología desempeña un papel crucial en la viabilidad y competitividad de las organizaciones. En el caso concreto de las PYMES, a pesar de ser el motor vital de muchas economías, a menudo enfrentan desafíos significativos al intentar digitalizar y automatizar sus procesos internos. La falta de recursos y la complejidad percibida de la tecnología a veces actúan como barreras para su adopción plena. Esta problemática se convierte en un llamado a la acción, ya que la digitalización no solo es una ventaja competitiva, sino una necesidad imperante para el crecimiento sostenible y la eficiencia operativa.

En este contexto, se pretende explorar un enfoque empresarial minimalista: la creación de una empresa de desarrollo de software con un número de empleados mínimo (1/2), operando desde la comodidad del hogar y orientada a proporcionar un servicio de digitalización de procesos a Pequeñas y Medianas Empresas (PYMES), ayudando a superar las limitaciones tradicionales de las empresas de este tipo, ofreciendo soluciones ágiles y personalizadas basadas sobre un ciclo de vida productivo de proyectos simple y efectivo, pasando por el análisis, diseño, desarrollo, mejora y mantenimiento continuo.

Nuestro objetivo es ser el socio estratégico que simplifica la complejidad tecnológica y permite a las PYMES aprovechar al máximo las oportunidades que la digitalización brinda.

2. Modelo de Plan de Negocio

2.1. Resumen Ejecutivo

En el dinámico escenario empresarial contemporáneo, la evolución constante de la tecnología desempeña un papel crucial en la viabilidad y competitividad de las organizaciones. Las Pequeñas y Medianas Empresas (PYMES), a pesar de ser el motor vital de muchas economías, enfrentan desafíos al intentar digitalizar y automatizar sus procesos internos debido a la falta de recursos y la complejidad percibida de la tecnología.

Objetivo: Explorar un enfoque empresarial minimalista mediante la creación de una empresa de desarrollo de software, operando desde el hogar con un número mínimo de empleados. La empresa estará orientada a proporcionar servicios de digitalización de procesos a PYMES, superando las limitaciones tradicionales y ofreciendo soluciones ágiles y personalizadas.

Enfoque: Desarrollar soluciones basadas en un ciclo de vida productivo de proyectos simple y efectivo, abarcando análisis, diseño, desarrollo, mejora y mantenimiento continuo. La misión es ser el socio estratégico que simplifica la complejidad tecnológica y permite a las PYMES aprovechar al máximo las oportunidades de la digitalización.

2.2. Entorno en el que se realizará la actividad de la empresa

Dado que la empresa se enfocará en la digitalización de procesos para PYMES a nivel un nivel inicial Autonómico (Galicia), podemos definir los siguientes puntos sobre esta base:

2.2.1. Entorno económico y sociocultural

Galicia, una región situada al noroeste de España, presenta un entorno económico caracterizado por una economía diversificada que abarca sectores como la pesca, la agricultura y las tecnologías emergentes. Según datos del Banco de España para el año 2023, el Producto Interno Bruto (PIB) de Galicia experimenta un crecimiento constante, con un aumento del 3.5%, reflejando la robustez de la economía regional.

En cuanto a Inflación y Costo de Vida en el mismo periodo, la inflación en Galicia se mantiene en niveles moderados, situándose en un 1.8%. Este indicador es crucial para comprender el costo de vida y los posibles impactos en los negocios locales. La estabilidad económica contribuye a un entorno propicio para el establecimiento y crecimiento de empresas, especialmente aquellas centradas en tecnologías innovadoras como la inteligencia artificial.

Por último, el entorno Sociocultural, la región se distingue por su rica herencia cultural y tradiciones arraigadas. En términos empresariales, la cultura de negocios en Galicia valora las relaciones personales y la confianza mutua. La comprensión y respeto por estas características socioculturales serán elementos esenciales para el éxito de la empresa en la interacción con clientes y colaboradores locales.

La combinación de un sólido crecimiento económico, inflación moderada y una cultura arraigada proporciona a la empresa un entorno favorable para el desarrollo de soluciones, al tiempo que se integra de manera efectiva en el tejido sociocultural de Galicia. Todo ello sin perder de vista la más que probable expansión e integración de la inteligencia artificial en el tejido empresarial de la región.

2.2.2. Entorno normativo

1. **Ley de Propiedad Intelectual (LPI):** Ley Orgánica 1/1996, de 12 de abril, de Propiedad Intelectual.
2. **Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico (LSSI-CE):** Ley 34/2002, de 11 de julio, de Servicios de la Sociedad de la Información y de Comercio Electrónico.
3. **Ley de Contratos del Sector Público (LCSP):** Real Decreto Legislativo 3/2011, de 14 de noviembre, por el que se aprueba el Texto Refundido de la Ley de Contratos del Sector Público.
4. **Reglamento General de Protección de Datos (RGPD):** Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos.
5. **Ley de Marcas:** Ley 17/2001, de 7 de diciembre, de Marcas.
6. **Ley de Competencia Desleal:** Ley 3/1991, de 10 de enero, de Competencia Desleal.
7. **Ley de Garantías en la Venta de Bienes de Consumo:** Real Decreto Legislativo 1/2007, de 16 de noviembre, por el que se aprueba el Texto Refundido de la Ley General para la Defensa de los Consumidores y Usuarios.

8. **Ley Orgánica de Protección de la Seguridad Ciudadana:** Ley Orgánica 4/2015, de 30 de marzo, de Protección de la Seguridad Ciudadana.

Futura legislación sobre AI: "La legislación futura sobre inteligencia artificial se centrará en garantizar la ética y transparencia en el desarrollo y uso de sistemas de AI, protegiendo los derechos individuales y abordando posibles impactos en la sociedad".

2.2.3. Entorno tecnológico

En un entorno marcado por rápidos avances y una demanda creciente de soluciones digitales, la convergencia de diversas tecnologías impulsa la innovación, ofreciendo oportunidades emocionantes para empresas ágiles y especializadas. Los puntos más destacados de dicho entorno son:

Inteligencia Artificial y Machine Learning: La IA y el aprendizaje automático han alcanzado un punto de inflexión, transformando la manera en que las empresas abordan la automatización, la toma de decisiones y la personalización de servicios. La capacidad de desarrollar soluciones basadas en algoritmos inteligentes ofrece un diferencial significativo en un mercado cada vez más competitivo.

Computación en la Nube: La adopción generalizada de servicios de computación en la nube ha democratizado el acceso a recursos computacionales de alta calidad. Esto permite a las pequeñas empresas de desarrollo de software competir en igualdad de condiciones con empresas más grandes al proporcionar soluciones escalables y flexibles sin inversiones masivas en infraestructura.

Desarrollo Ágil: Metodologías ágiles como Scrum y Kanban se han convertido en estándares en la industria del desarrollo de software. Estas metodologías permiten una respuesta rápida a los cambios en los requisitos del cliente, mejorando la eficiencia y la calidad del producto final. Eso sí, debemos buscar el equilibrio entre el peso de ciertos procesos y la agilidad que requiere cada proyecto.

Internet de las Cosas (IoT): La proliferación de dispositivos conectados ha creado oportunidades para el desarrollo de aplicaciones que aprovechan la interconexión de objetos cotidianos. Desde ciudades inteligentes hasta dispositivos para el hogar, el desarrollo de software centrado en IoT es esencial para mantenerse a la vanguardia.

Seguridad Cibernética: Con la creciente amenaza de ciberataques, la seguridad cibernética se ha convertido en una prioridad. Las empresas de desarrollo de software deben integrar prácticas de seguridad desde las etapas iniciales del diseño para garantizar la protección de datos y la confianza del cliente.

Realidad Aumentada y Virtual: El avance en la realidad aumentada (RA) y la realidad virtual (RV) ofrece oportunidades emocionantes para la creación de experiencias inmersivas. Desde aplicaciones de entrenamiento hasta soluciones de marketing innovadoras, estas tecnologías están redefiniendo la interacción digital.

Desarrollo Sostenible: Existe una creciente conciencia sobre la importancia de la sostenibilidad en el desarrollo de software. Las empresas deben considerar el impacto ambiental de sus soluciones y adoptar prácticas que promuevan la eficiencia y la responsabilidad social.

En este contexto, la comprensión y la incorporación de estas tendencias tecnológicas en el modelo de negocio son esenciales para destacar y prosperar en este emocionante entorno tecnológico.

2.3. Análisis de Mercado

2.3.1. Oportunidad de Mercado

La creciente necesidad de las Pequeñas y Medianas Empresas (PYMES) de digitalizar sus procesos internos representa una oportunidad estratégica en el mercado. Existe una demanda latente de soluciones que no solo mejoren la eficiencia operativa, sino que también se ajusten a las características y presupuestos específicos de las PYMES. Nuestra empresa se posiciona para capitalizar esta oportunidad al ofrecer servicios de digitalización de procesos adaptado y asequibles.

Puntos Clave:

- **Eficiencia Operativa:** Las PYMES buscan optimizar sus operaciones a través de la digitalización para competir de manera efectiva en el mercado actual.
- **Personalización y Accesibilidad:** La falta de soluciones personalizadas y asequibles crea una brecha que nuestra empresa busca llenar, proporcionando servicios adaptados a las necesidades específicas de cada cliente.

2.3.2. Competencia

El panorama actual está dominado por competidores que ofrecen soluciones con enfoques complejos y costosos. Esta dinámica crea una oportunidad significativa para nuestra empresa, que se propone diferenciarse a través de la entrega de soluciones ágiles y la minimización de costos para los clientes.

Enfoque Competitivo:

- **Soluciones Ágiles:** La agilidad en el desarrollo y la implementación de soluciones nos permitirá responder rápidamente a las necesidades cambiantes de los clientes, estableciendo una ventaja competitiva.
- **Minimización de Costos:** Al operar de manera minimalista y desde el hogar, podemos ofrecer tarifas competitivas sin comprometer la calidad del servicio.

2.3.3. Análisis DAFO

El análisis DAFO (Debilidades, Amenazas, Fortalezas y Oportunidades) nos brinda una comprensión integral de nuestra posición en el mercado y nos ayuda a desarrollar estrategias efectivas.

Debilidades:

- **Tamaño Reducido:** Al ser una empresa minimalista, nuestra capacidad de manejar múltiples proyectos simultáneamente podría ser limitada.
- **Visibilidad Inicial:** Dada nuestra naturaleza emergente, la construcción de una base de clientes puede llevar tiempo.

Amenazas:

- **Competencia Establecida:** Empresas con una presencia establecida pueden representar una amenaza inicial para nuestra cuota de mercado.
- **Fluctuaciones del Mercado:** Cambios económicos repentinos pueden afectar la demanda de servicios de desarrollo de software.

Fortalezas:

- **Agilidad Empresarial:** Nuestra estructura minimalista nos permite tomar decisiones rápidas y adaptarnos a las tendencias del mercado.
- **Costos Competitivos:** La minimización de costos nos permite ofrecer tarifas atractivas para las PYMES.

Oportunidades:

- **Crecimiento del Mercado PYME:** El continuo crecimiento del sector PYME proporciona oportunidades constantes para la expansión de nuestra clientela.
- **Innovación Tecnológica:** Mantenernos actualizados con las últimas tecnologías nos permite ofrecer soluciones innovadoras y mantenernos competitivos.

Este análisis DAFO servirá como base estratégica para el desarrollo de nuestro plan de negocio, ayudándonos a maximizar nuestras fortalezas, mitigar debilidades, capitalizar oportunidades y gestionar amenazas de manera efectiva.

2.4. Modelo de negocio

2.4.1. Propuesta de valor

Ofrecer servicios de digitalización de procesos a PYMES con soluciones ágiles, personalizadas y rentables.

2.4.2. Segmento de clientes

PYMES en diversas industrias con la necesidad de optimizar procesos y adoptar tecnologías digitales.

2.4.3. Canales de Distribución

1. **Marketing Digital:** Utilizaremos estrategias online para promocionar nuestros servicios, llegar a un público más amplio y generar clientes potenciales.
2. **Alianzas Estratégicas:** Estableceremos colaboraciones con organizaciones empresariales afines para ampliar nuestra red y acceder a nuevas oportunidades de negocio.
3. **Redes Sociales y Eventos Locales:** Aprovecharemos las plataformas de redes sociales para construir una presencia online sólida y participaremos en eventos locales para generar visibilidad y establecer conexiones directas con posibles clientes.

2.4.4. Fuentes de Ingresos

1. **Desarrollo de Software Personalizado:** Generaremos ingresos a través del diseño y creación de soluciones de software adaptadas a las necesidades específicas de cada cliente.
2. **Servicios de Consultoría en Digitalización:** Ofreceremos asesoramiento especializado para guiar a las PYMES en su proceso de digitalización, proporcionando recomendaciones y mejores prácticas.
3. **Mantenimiento y Soporte Continuo:** Garantizaremos la continuidad operativa de las soluciones implementadas mediante servicios de mantenimiento y soporte técnico.

2.5. Operaciones y Estructura Organizativa

2.5.1. Estructura Organizativa

- **Propietario/Desarrollador Principal:** Encargado de la toma de decisiones estratégicas, desarrollo de software y gestión general.
- **Desarrollador Cloud:** Encargado de la consultoría, desarrollo y mantenimiento de las soluciones de infraestructura, automatización, persistencia, hospedaje y seguridad de soluciones en la nube que darán soporte a las necesidades de los proyectos y clientes.
- **Colaboración con Profesionales Especializados:** Según los proyectos, nos asociaremos con profesionales especializados en áreas específicas para garantizar la calidad y eficacia de nuestras soluciones.

2.5.2. Proceso de Desarrollo

A continuación se definen las líneas maestras del proceso de desarrollo de software que propone la empresa. En el punto 4 del proyecto se describe de forma amplia cada una de estas líneas maestras.

1. **Análisis de Necesidades:** Comprenderemos las necesidades específicas de cada cliente a través de consultas detalladas y de la elaboración.
2. **Propuesta de Solución:** Presentaremos una propuesta de solución personalizada, incluyendo un alcance detallado y dos modalidades de servicio: El proyecto cerrado y el proyecto continuo, ambos opcionalmente complementados con un plan personalizado de mantenimiento y soporte técnico.
3. **Diseño y Desarrollo:** Crearemos soluciones personalizadas, asegurando un diseño eficiente y una implementación efectiva siguiendo a las especificaciones plasmadas en la propuesta de solución.
4. **Puesta en producción y Mejora Continua:** Desplegaremos las soluciones y buscaremos continuamente formas de mejorar su rendimiento y funcionalidad.
5. **Mantenimiento y Soporte Técnico:** Servicio de carácter opcional donde proporcionaremos servicios de mantenimiento regular y soporte técnico para garantizar un funcionamiento sin problemas a lo largo del ciclo de vida de la solución.

2.5.3. Localización de la Empresa: Rural vs Ciudad.

2.5.3.1 Asentamiento Previsto y Criterios para su Selección

La empresa adopta un enfoque remoto y amigable con el trabajo a distancia, lo que elimina las limitaciones geográficas tradicionales. Aunque la elección de una ubicación física debe de ser estratégica para reuniones presenciales, eventos o necesidades específicas, se sacrifican ciertas ventajas que puedan

dar un asentamiento en grandes ciudades como Madrid o Barcelona en pro de una localización rural que permita reducir costes y ofrecer un mejor equilibrio entre vida laboral y personal para los empleados. Actualmente, la brecha entre las ciudades y el mundo rural es mucho menor, permitiendo hacer viable este enfoque.

- **Accesibilidad Digital:** Garantizar una infraestructura digital sólida para facilitar el trabajo remoto eficiente y la conectividad confiable. La proliferación de una mejor cobertura de fibra óptica en las zonas rurales proporciona dicha solidez.
- **Proximidad a Centros Empresariales:** Considerar la cercanía a áreas con presencia significativa de PYMES y eventos empresariales para oportunidades de networking y colaboración. La mejora de las infraestructuras permite de nuevo que el entorno rural pueda acceder a este tipo de eventos.
- **Calidad de Vida:** Buscar ubicaciones que ofrezcan una alta calidad de vida para atraer y retener talento, incluso si no es necesario trabajar físicamente en la oficina. Cada día más personas prefieren entornos rurales para vivir y trabajar con un menor coste de vida y una mayor calidad de vida.
- **Costo de Vida:** Evaluar el costo de vida para asegurar la viabilidad económica tanto para la empresa como para los empleados remotos. El coste de vida en entornos rurales es mucho menor que en las grandes ciudades.

2.5.3.2 Instalaciones y Comunicaciones

Dado el enfoque remoto, la empresa prioriza las instalaciones virtuales sobre las físicas. Se invertirá en plataformas de comunicación digital robustas, herramientas de colaboración en línea y sistemas de gestión de proyectos. La infraestructura digital será clave para garantizar una comunicación eficiente y efectiva entre los miembros del equipo, independientemente de su ubicación.

2.5.4. Recursos Humanos

2.5.4.1 Organización de Recursos y Medios Técnicos y Humanos

La empresa adopta un modelo de equipo flexible, aprovechando profesionales especializados según los requisitos de proyectos específicos. Se enfoca en la adquisición de talento remoto para aprovechar las ventajas de diversidad geográfica y habilidades especializadas. Los recursos técnicos y humanos se organizan de manera ágil para adaptarse a las demandas cambiantes del mercado.

2.5.4.2 Organigrama. Estructura de Dirección y Gestión

La estructura organizativa refleja la naturaleza remota de la empresa y se centra en la colaboración virtual. Las funciones directivas y de gestión se distribuyen de manera eficiente, asegurando una toma de decisiones ágil. La estructura plana fomenta la comunicación abierta y la participación de todos los miembros del equipo, independientemente de su posición. La dirección adopta un enfoque de liderazgo orientado a resultados y desarrollo profesional continuo.

2.6. Estrategia de marketing y ventas

2.6.1. Estrategia de marketing

- Enfoque en la simplicidad y eficacia de las soluciones.
- Marketing de contenidos educativos sobre digitalización para PYMES.

2.6.2. Estrategia de ventas

- Ofrecer evaluaciones gratuitas de digitalización.
- Paquetes de servicios personalizados según las necesidades del cliente.

2.7. Finanzas y Proyecciones

2.7.1. Inversión Inicial

La inversión inicial se distribuirá en áreas clave para garantizar un inicio sólido y sostenible de la empresa.

- **Equipamiento y Software:** Se estima una inversión de 2,000 euros para adquirir hardware de calidad, licencias de software esenciales y herramientas de desarrollo necesarias.
- **Marketing y Promoción Inicial:** La promoción inicial es crucial para establecer la presencia en el mercado. Se asigna un presupuesto de 1,000 euros para estrategias de marketing digital, presencia en eventos locales y creación de materiales promocionales.
- **Reserva de Emergencia:** Para cubrir posibles imprevistos y mantener la estabilidad financiera, se destina un fondo de emergencia de 1,000 euros.

2.7.2. Proyecciones Financieras

Las proyecciones financieras se basan en un enfoque realista y una gestión cuidadosa de los ingresos y gastos.

- **Ingresos Esperados basados en Proyectos:** Se proyecta alcanzar ingresos iniciales de 20,000 euros durante el primer año, considerando la adquisición de proyectos de desarrollo de software y consultoría en digitalización para PYMES.
- **Control Estricto de Costos:** Se implementará un enfoque riguroso para gestionar los costos operativos, manteniendo un equilibrio entre la calidad del servicio y la eficiencia financiera. Se espera que los gastos operativos no superen el 70% de los ingresos.
- **Escalabilidad Gradual:** La empresa tiene como objetivo una escalabilidad prudente. Se planifica reinvertir el 20% de los ingresos en la mejora de la infraestructura tecnológica, expansión del equipo y desarrollo de nuevas competencias.

Estas proyecciones están sujetas a ajustes según la evolución del mercado y las condiciones económicas, con revisiones regulares para asegurar la adaptación continua a las demandas del entorno empresarial.

2.8. Aspectos Formales de la Empresa

2.8.1. Elección de la Forma Jurídica

La empresa optará por la forma jurídica de **Empresa Individual**, dada su naturaleza minimalista y su estructura de operación desde el hogar. Esta elección se fundamenta en la responsabilidad limitada del titular y la flexibilidad en la toma de decisiones sin la obligación de un capital inicial significativo.

2.8.1.1. Criterios de Elección para la Forma Jurídica

La decisión de elegir una Empresa Individual se basa en los siguientes criterios:

- **Responsabilidad Limitada:** La forma jurídica de Empresa Individual ofrece al titular una responsabilidad limitada, lo que significa que sus activos personales están separados de los activos de la empresa.
- **Flexibilidad Operativa:** Al no requerir un capital inicial importante y permitir una toma de decisiones ágil, la Empresa Individual se adapta de manera óptima a la naturaleza minimalista y remota del negocio.
- **Sencillez Administrativa:** La carga administrativa y los requisitos legales son menos exigentes en comparación con otras formas jurídicas, lo que facilita la gestión eficiente de la empresa.

2.8.1.2. Borrador de los Estatutos Sociales

A continuación, se presenta un resumen del borrador de los Estatutos Sociales de la Empresa Individual:

- **Denominación Social:** [Nombre del titular] Desarrollo de Software.
- **Domicilio Social:** [Dirección del domicilio del titular].
- **Objeto Social:** Desarrollo de software, consultoría en digitalización y servicios relacionados con la tecnología.
- **Duración:** Indefinida, sujeta a la normativa vigente.
- **Capital Social:** No aplica, dado que la Empresa Individual no requiere un capital social mínimo.
- **Órganos de Administración:** El titular será el único responsable de la administración y toma de decisiones.
- **Responsabilidad Limitada:** El titular no responde con sus bienes personales, limitando su responsabilidad al capital de la empresa.

Este borrador se ajustará según la normativa legal específica y los requisitos formales al momento de la constitución de la empresa.

2.8.2. Protección Jurídica

2.8.2.1. Medidas de Protección Jurídica

Dada la importancia de asegurar la protección legal de la empresa, se implementarán las siguientes medidas:

- **Contratos y Acuerdos:** Se establecerán contratos claros y detallados con clientes, colaboradores y cualquier entidad con la que la empresa tenga relación comercial. Estos contratos especificarán los términos, condiciones y responsabilidades de ambas partes, proporcionando una base legal sólida.
- **Política de Privacidad y Protección de Datos:** Se implementará una política de privacidad que cumpla con la normativa de protección de datos vigente. Esto garantizará el manejo adecuado y seguro de la información confidencial de clientes y colaboradores.
- **Asesoría Legal Continua:** La empresa contará con servicios de asesoría legal para mantenerse actualizada sobre los cambios en la legislación y recibir orientación en aspectos legales relevantes para su operación.

2.8.2.2. Patentes, Marcas, Licencias y Permisos

Para resguardar la propiedad intelectual y asegurar el cumplimiento normativo, se llevarán a cabo las siguientes acciones:

- **Registro de Propiedad Intelectual:** Se evaluará la posibilidad de registrar el software desarrollado y cualquier propiedad intelectual asociada para proteger los derechos de autor y la innovación.
- **Registro de Marca:** Se realizará el registro de la marca comercial de la empresa para proteger su identidad y diferenciar sus servicios en el mercado.
- **Licencias de Software:** En caso de utilizar software de terceros, se verificará y adquirirá las licencias necesarias para garantizar un uso legal y evitar posibles litigios.
- **Permisos y Autorizaciones:** Se obtendrán los permisos y autorizaciones necesarios para operar la empresa de acuerdo con la normativa local y sectorial.

Estas medidas asegurarán una base jurídica sólida y contribuirán a la protección de los activos de la empresa.

2.8.3. Medios de Prevención de Riesgos Laborales

Aunque la empresa opera de manera remota, se implementarán medidas para prevenir riesgos laborales, priorizando la salud y bienestar de los empleados.

Legislación Aplicable en Trabajo Remoto

La empresa se registrará por la legislación vigente en materia de trabajo remoto en España. Algunos aspectos clave incluirán:

- **Evaluación de Riesgos:** Se realizará una evaluación de riesgos laborales adaptada al trabajo remoto, identificando posibles peligros en el entorno doméstico de los autónomos.

- **Equipamiento Ergonómico:** La empresa proporcionará orientación sobre la configuración ergonómica de los espacios de trabajo en casa y, en la medida de lo posible, facilitará el suministro de equipos ergonómicos para garantizar condiciones laborales seguras.
- **Horarios y Descansos:** Se establecerán pautas claras respecto a horarios laborales, pausas y períodos de descanso para evitar la fatiga y promover un equilibrio adecuado entre trabajo y vida personal.
- **Comunicación Continua:** Se fomentará la comunicación constante entre la empresa y los autónomos para abordar cualquier preocupación relacionada con el bienestar laboral.
- **Formación en Prevención:** Se proporcionará formación específica en prevención de riesgos laborales, adaptada a las circunstancias del trabajo remoto.

Estas medidas garantizarán el cumplimiento normativo y contribuirán a un entorno laboral remoto seguro y saludable.

2.9. Innovación

2.9.1. Aspectos Innovadores

La empresa se destacará por:

- **Flexibilidad Laboral:** La estructura remota y la autonomía ofrecen flexibilidad a los empleados, fomentando un ambiente innovador y adaptativo.
- **Adopción de Nuevas Tecnologías:** La constante actualización en las tendencias tecnológicas garantizará soluciones innovadoras y competitivas.

2.9.2. Elementos de Sostenibilidad de la Innovación

La sostenibilidad de la innovación se promoverá mediante:

- **Formación Continua:** Incentivando a los empleados a mantenerse actualizados con las últimas tecnologías a través de formación continua.
- **Cultura Colaborativa:** Fomentando la colaboración y el intercambio de ideas para impulsar la innovación constante en el desarrollo de soluciones digitales.

3. Planificación temporal de la puesta en marcha

Fase 1: Preparación (Mes 1)

- **Semana 1-2:** Definición del plan de negocio, objetivos y estrategias.
- **Semana 3-4:** Investigación y selección de herramientas y tecnologías necesarias.

Fase 2: Aspectos Legales y Administrativos (Mes 2)

- **Semana 1-2:** Registro como trabajador autónomo y elección de la forma jurídica.
- **Semana 3-4:** Establecimiento de medidas de protección jurídica y trámites legales.

Fase 3: Configuración Tecnológica (Mes 3)

- **Semana 1-2:** Adquisición y configuración de equipos y software.
- **Semana 3-4:** Implementación de medidas de ciberseguridad y protección de datos.

Fase 4: Marketing y Presencia Online (Mes 4)

- **Semana 1-2:** Desarrollo de estrategia de marketing digital.
- **Semana 3-4:** Creación de perfiles en redes sociales y sitio web.

Fase 5: Inicio de Actividades (Mes 5)

- **Semana 1-2:** Lanzamiento oficial y promoción en redes sociales.
- **Semana 3-4:** Oferta de servicios y evaluación de retroalimentación inicial.

Fase 6: Consolidación y Ajustes (Mes 6 en adelante)

- **Mes 6-12:** Evaluación continua, ajustes en la estrategia según la retroalimentación.
- **Mensualmente:** Revisión de objetivos y proyecciones financieras.

Nota: Dado que es una empresa individual, las fases pueden solaparse y ajustarse según las necesidades y circunstancias.

4. Proceso productivo: auto-repair, un proyecto tipo

4.1. Captación del Cliente

4.1.1. Identificación de Potenciales Clientes

- Utilización de investigaciones de mercado para identificar talleres mecánicos interesados en Auto-repair, centrándonos en aquellos que buscan una gestión integral de su negocio con un desembolso monetario adaptado a las posibilidades de las PYMEs.

- Campañas de marketing dirigidas a talleres específicos de tamaño pequeño/mediano, enfatizando la asequibilidad y adaptabilidad de Auto-repair para las PYMEs, destacando que satisface las necesidades sin exceder presupuestos.

4.1.2. Estrategias de Marketing y Publicidad

- Desarrollo de campañas publicitarias que resalten las ventajas de Auto-repair para los talleres, centrándonos en su flexibilidad y adaptabilidad a las necesidades de cada negocio.
- Participación activa en eventos locales de la industria automotriz para aumentar la visibilidad y establecer conexiones directas con los propietarios de talleres.

4.1.3. Presentación de Servicios y Propuestas

- Programación de reuniones detalladas para presentar los servicios de Auto-repair, destacando cómo puede beneficiar a los talleres de forma específica y adaptada a sus requerimientos.
- Elaboración de propuestas personalizadas que resalten la relación costo-beneficio, haciendo hincapié en cómo Auto-repair puede cubrir las necesidades específicas de cada taller.

4.2. Análisis y Requerimientos

4.2.1. Reuniones con el Cliente para Comprender sus Necesidades

- Establecimiento de reuniones presenciales o virtuales con los clientes, priorizando la cercanía y la comprensión directa de sus necesidades.
- Participación activa en sesiones de lluvia de ideas para captar los requisitos específicos de cada taller, aprovechando la experiencia en el sector de desarrollo y de la automoción de la mano de Auto-repair.
- De estas reuniones se obtendrá la información necesaria para la elaboración de la documentación de los requisitos en forma de informes y actas de reunión.

Ejemplo de acta de reunión:

Informe de Reuniones con Cliente – Aplicación de Gestión para Talleres Mecánicos

Resumen Ejecutivo

Este informe resume las discusiones y conclusiones de las primeras reuniones con el cliente con el objetivo de identificar las necesidades para el desarrollo de una aplicación de gestión mecánicos de pequeño y mediano tamaño.

Contexto

El cliente expresó la necesidad de mejorar la eficiencia operativa y la administración interna. Actualmente, enfrentan desafíos en la organización de tareas, seguimiento de inventario, y comunicación con clientes y empleados.

Reunión 1 – Exploración de Requerimientos Iniciales (Fecha: DD/MM/AAAA)

Participantes:

– Representantes del Cliente

– Auto-repair

- Equipo de Desarrollo

Temas Discutidos:

1. **Procesos Operativos Actuales:**

- Se revisaron los procesos internos del taller, desde la recepción de clientes hasta la reparación de vehículos.
- Se identificaron áreas de mejora en la eficiencia y la comunicación interna.

2. **Gestión de Citas:**

- Se discutió la necesidad de un sistema de gestión de citas eficiente para evitar conflictos de tiempo de trabajo del personal.

3. **Control de Inventario:**

- El cliente expresó preocupaciones sobre el seguimiento de piezas y repuestos, así como la automatización de reabastecimiento.

Reunión 2 – Detalle de Funcionalidades (Fecha: DD/MM/AAAA)

Participantes:

- Representantes del Cliente
- Equipo de Desarrollo

Temas Discutidos:

1. **Inicio de Sesión y Perfiles:**

- Se acordó la importancia de distintos roles (administrador, mecánico, recepcionista) para asegurar la seguridad y privacidad de la información.

2. **Gestión de Clientes:**

- Se detallaron los datos necesarios para el registro de clientes y la importancia de mantener actualizados los servicios.

3. **Facturación y Contabilidad:**

- Se identificaron las funciones necesarias para crear facturas, registrar transacciones y generar reportes.

4. **Comunicación Cliente-Taller:**

- Se discutieron herramientas de comunicación interna y externa para mantener a los clientes informados sobre el progreso de las reparaciones.

Conclusiones y Próximos Pasos

1. **Priorización de Funcionalidades:**

- Se establecieron prioridades para las funcionalidades identificadas, con un enfoque inicial en la gestión de citas, control de inventario y comunicación cliente-taller.

2. **Fases del Desarrollo:**

- Se acordó una aproximación por fases en el desarrollo de la aplicación, comenzando con las funcionalidades fundamentales y expandiéndose según las necesidades evolucionen.

3. **Próxima Reunión:**

- Se programó una próxima reunión para revisar los primeros prototipos y ajustar los requisitos del desarrollo.

Este informe servirá como guía para el equipo de desarrollo al comenzar el diseño y la implementación de la aplicación de gestión para talleres mecánicos. Se espera que esta solución mejore significativamente la eficiencia y la experiencia operativa del taller.

4.2.2. Documentación de Requerimientos Funcionales y no Funcionales

- Detallada documentación de los requisitos funcionales y no funcionales identificados durante las reuniones con el cliente. Se crearán actas de cada reunión para garantizar la correcta comprensión de los requisitos y dejar constancia de los mismos.
- Utilización de un lenguaje claro y accesible para garantizar la comprensión mutua y evitar malentendidos.

Sobre estas bases, se definen los siguientes requisitos para nuestra app auto-repair:

Existirán tres niveles de usuarios: administrador , empleado y cliente :

- El administrador será el encargado de gestionar los datos de los clientes y de los empleados, así como d

4.2.2.1 Requisitos Funcionales

Funcionalidades Generales:

1. Inicio de Sesión:

- Los usuarios pueden iniciar sesión como administrador, empleado o cliente.

2. Panel de Control:

- Resumen de actividades recientes.
- Acceso rápido a funciones clave.

3. Gestión de Perfiles:

- Crear, editar y eliminar perfiles de empleados y clientes.
- Asignar roles y permisos.

4. Notificaciones:

- Notificaciones en tiempo real para eventos importantes.

5. Registro de Actividades:

- Registro de cambios y acciones realizadas por usuarios.

Funcionalidades para Administradores:

1. Gestión de Empleados:

- Crear, editar y eliminar perfiles de empleados.
- Asignar roles y permisos específicos.

2. Inventario:

- Seguimiento de existencias de piezas y repuestos.

- Alertas de reabastecimiento.

3. Gestión de Clientes:

- Visualizar y editar información de clientes.
- Historial de servicios y transacciones.

4. Programación de Citas:

- Crear citas para clientes.
- Asignar tareas a empleados.

5. Facturación y Contabilidad:

- Crear y gestionar facturas.
- Seguimiento de transacciones y gastos.

Funcionalidades para Empleados:

1. Agenda de Trabajo:

- Visualizar tareas asignadas y citas programadas.
- Confirmar la finalización de tareas.

2. Historial de Servicios:

- Registrar detalles de los servicios realizados.
- Actualizar el estado de reparación.

3. Gestión de Inventario:

- Solicitar piezas y repuestos.
- Marcar productos como utilizados.

4. Comunicación Interna:

- Mensajería interna para colaboración entre empleados.

Funcionalidades para Clientes:

1. Solicitud de Citas:

- Solicitar citas para servicios de reparación o mantenimiento.

2. Historial de Servicios:

- Visualizar historial de servicios y reparaciones anteriores.

3. Presupuestos:

- Solicitar y revisar presupuestos para servicios.

4. **Comunicación con el Taller:**

- Mensajería para consultas y actualizaciones.

5. **Pagos en Línea:**

- Facilitar pagos de servicios en línea.

Funcionalidades Compartidas:

1. **Seguimiento en Tiempo Real:**

- Clientes y empleados pueden seguir el progreso de las reparaciones.

2. **Documentación Digital:**

- Almacenar y compartir documentos importantes como facturas y garantías.

3. **Reportes y Estadísticas:**

- Generar informes sobre desempeño, ingresos y tendencias.

4. **Integración de Calendario:**

- Sincronizar citas con calendarios externos (Google Calendar, etc.).

5. **Seguridad:**

- Seguridad robusta con autenticación de dos factores y encriptación.

6. **Soporte Técnico:**

- Sistema de tickets para solicitar ayuda y soporte técnico.

4.2.2.2 Requisitos no Funcionales

1. **Usabilidad:**

- La interfaz de usuario debe ser intuitiva y fácil de usar.

2. **Rendimiento:**

- Respuesta rápida a las solicitudes de los usuarios.
- Capacidad para manejar múltiples usuarios concurrentes.

3. **Seguridad:**

- Autenticación segura.
- Encriptación de datos sensibles.

4. **Disponibilidad:**

- La aplicación debe estar disponible el mayor tiempo posible.

5. **Escalabilidad:**

- Capacidad para crecer y adaptarse a un aumento en el volumen de datos o usuarios.

6. **Mantenibilidad:**

- Facilidad para realizar actualizaciones y correcciones.
- Documentación clara y completa.

7. **Compatibilidad:**

- Compatibilidad con diferentes navegadores web.
- Interoperabilidad con sistemas externos si es necesario.

8. **Cumplimiento Normativo:**

- Cumplimiento con regulaciones y normativas de seguridad y privacidad.

9. **Desempeño:**

- Tiempo de carga rápido para evitar la frustración de los usuarios.

10. **Backup y Recuperación:**

- Sistemas de respaldo regulares y capacidad de recuperación de datos en caso de fallos.

Estos requisitos funcionales y no funcionales ayudarán a guiar el desarrollo de la aplicación web para garantizar que cumpla con las expectativas y necesidades de los usuarios y de la empresa.

4.2.3. **Análisis de Viabilidad del Proyecto**

- Evaluación cuidadosa de la viabilidad técnica y financiera del proyecto en colaboración estrecha con el cliente.
- Proporcionar análisis transparente sobre la capacidad de Auto-repair para cumplir con los requisitos identificados.

4.2.3. **Propuesta de Proyecto Cerrado o Continuo**

- Presentación de opciones claras de proyecto cerrado o continuo, destacando los beneficios de ambas y adaptándolas a las necesidades específicas de cada cliente.
- Proyecto Cerrado: Desarrollo de un proyecto único con un presupuesto y plazo de entrega fijos.
- Proyecto Continuo: Desarrollo de un proyecto a largo plazo con un presupuesto y plazo de entrega flexibles, con iteraciones continuas y adaptación a los cambios de requisitos, próximo a metodologías ágiles.

4.3. Diseño del Software

4.3.1. Creación de Prototipos y Wireframes

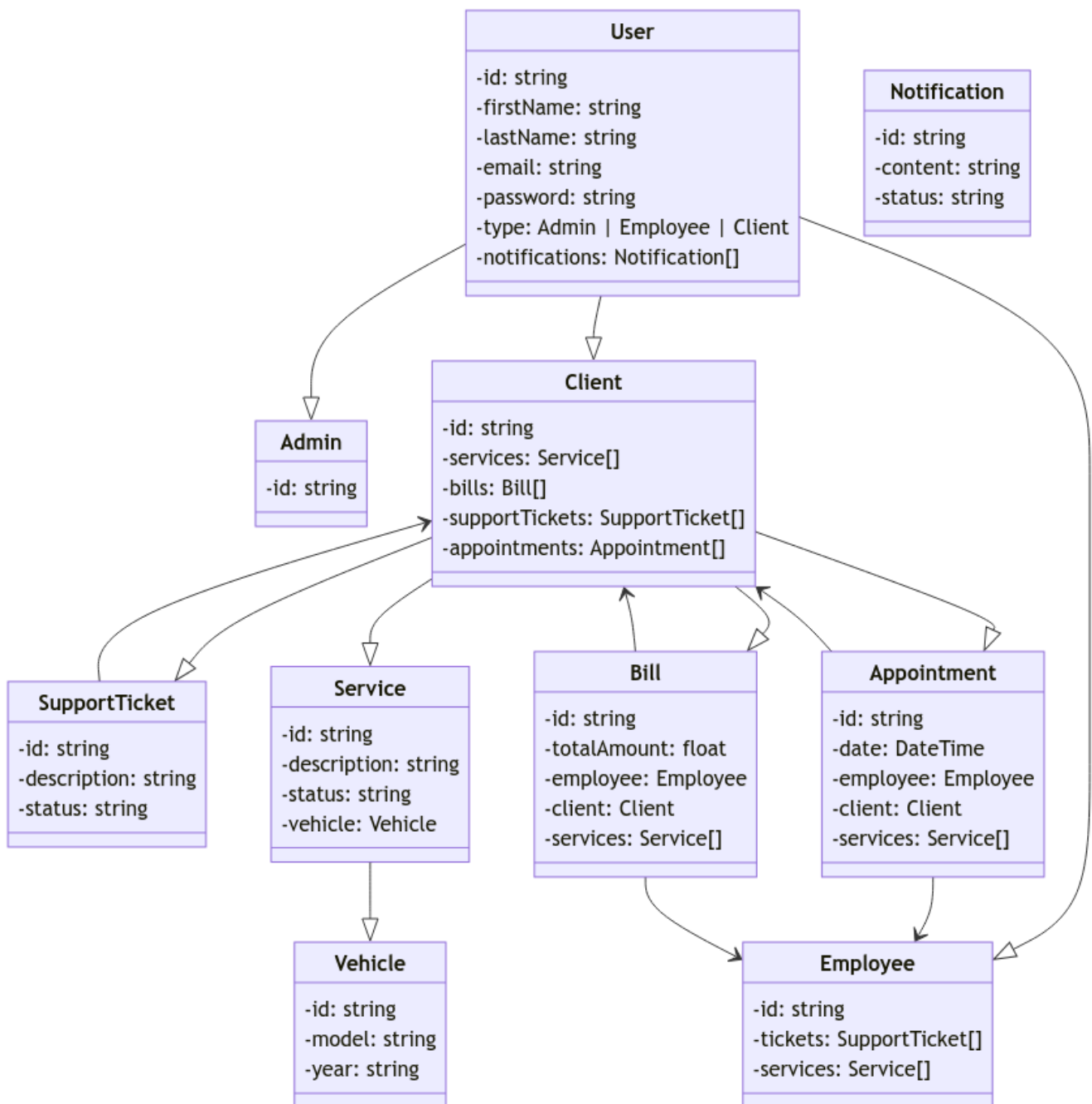
- Desarrollo de prototipos visuales y wireframes interactivos que representen la estructura y flujo de Auto-repair.
- Iteración constante en colaboración con el cliente para ajustar y mejorar el diseño según sus comentarios y necesidades.

4.3.2. Diseño de la Arquitectura del Sistema

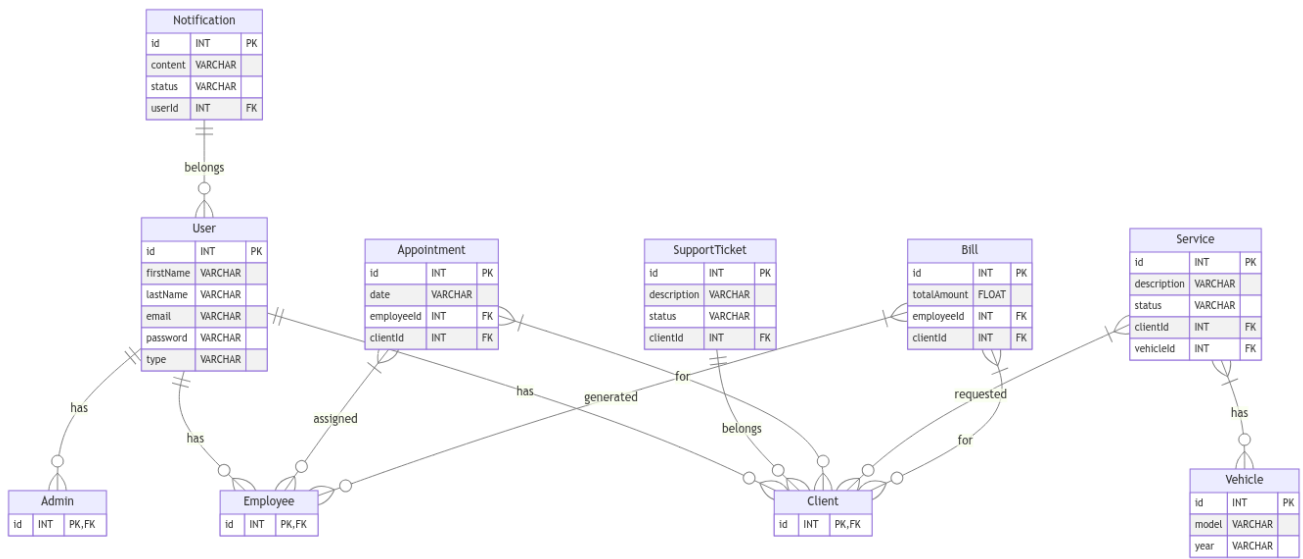
- Definición de una arquitectura de sistema tradicional con cuatro componentes esenciales: el frontend web, el backend con una API basada en el patrón MVC (Modelo-Vista-Controlador), la capa de persistencia y una posible capa de caché opcional.
- El frontend web proporcionará la interfaz de usuario que interactuará directamente con los usuarios de los talleres mecánicos, permitiéndoles gestionar sus actividades de manera eficiente y sencilla.
- El backend, implementado con un enfoque basado en el patrón MVC, se encargará de procesar las solicitudes del frontend, gestionar la lógica de negocio y acceder a la capa de persistencia para almacenar y recuperar información relevante.
- La capa de persistencia constará de dos subpiezas: una base de datos tradicional relacional, que almacenará información crítica y estructurada del sistema, y otra de tipo no relacional, destinada al almacenamiento de información secundaria que no requiere la misma estructura rigurosa.
- Se priorizará la escalabilidad y la redundancia en el diseño de la capa de persistencia para garantizar la integridad y disponibilidad de los datos.
- La capa de caché opcional podría implementarse más adelante para mejorar la velocidad de acceso a datos frecuentemente utilizados, optimizando así el rendimiento del sistema.
- La comunicación entre el frontend y el backend, así como entre el backend y la capa de persistencia (y la posible capa de caché), se llevará a cabo a través de interfaces bien definidas y protocolos seguros.

4.3.2.1. Diseño backend

- Entidades principales del sistema:

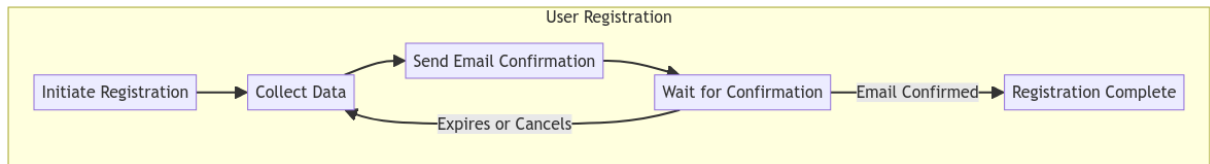


- Diagrama base de la base de datos:

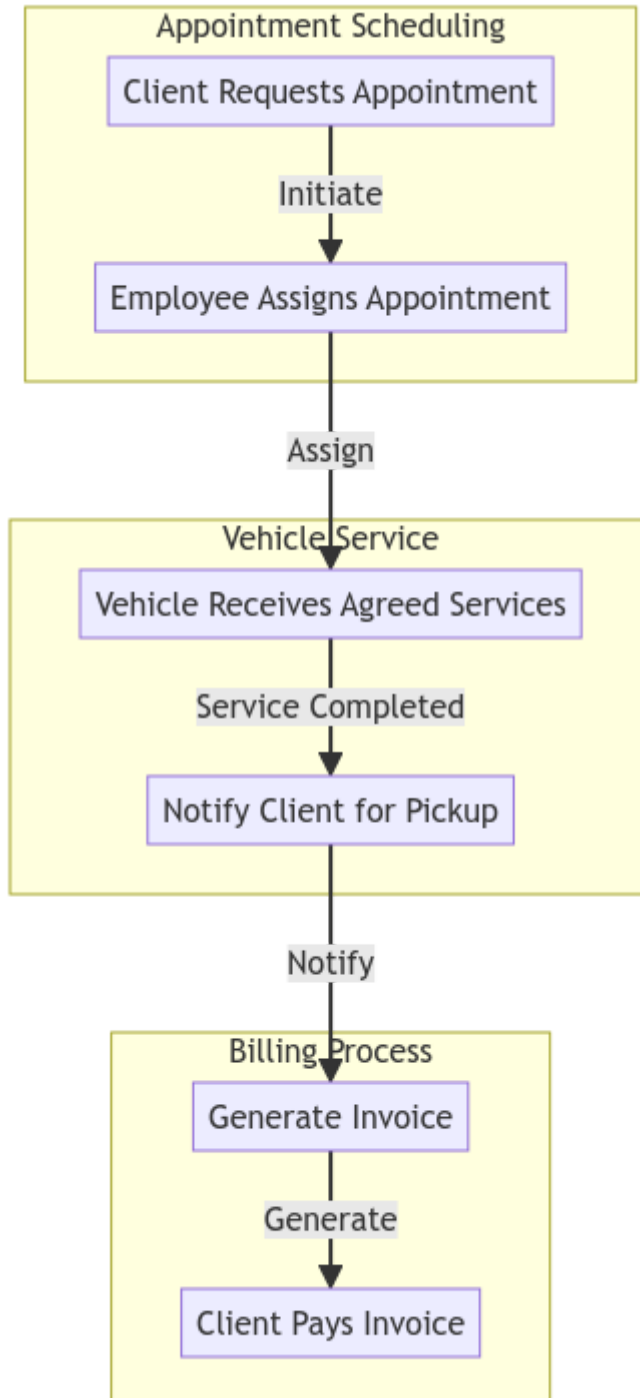


- Ejemplos de algunos flujos de la aplicación:

- Registro de usuario:



- Prestación de servicio a un vehículo:



4.3.2.2. Diseño frontend

4.3.2.2.1. Definición de la Interfaz de Usuario (UI) y la Experiencia del Usuario (UX)

- Diseño de una interfaz de usuario (UI) basada en componentes tipo tabla, formularios y dashboards, permitiendo una experiencia visual clara y eficiente para los usuarios de Auto-repair.
- Implementación de visualizaciones personalizadas por usuario, adaptadas a sus roles y responsabilidades en el taller mecánico, brindando información relevante y funcionalidades específicas.
- Desarrollo de una UI responsive, optimizada para versiones desktop y dispositivos móviles, garantizando la accesibilidad y usabilidad en diferentes plataformas.
- Integración de elementos de diseño moderno y atractivo, priorizando la legibilidad, la consistencia y el contraste para mejorar la experiencia del usuario y reducir la curva de aprendizaje.
- Colaboración estrecha con los usuarios finales para recopilar comentarios sobre la UI/UX durante el desarrollo, asegurando ajustes continuos y mejoras basadas en la retroalimentación directa.

4.3.4. Elección de Tecnologías y Herramientas

El lenguaje de programación elegido para el desarrollo de Auto-repair es **JavaScript**, un lenguaje de programación interpretado, dinámico y de alto nivel, que se ejecuta en el lado del cliente y del servidor. JavaScript es el lenguaje de programación más popular del mundo, utilizado por más del 95% de los sitios web y por la mayoría de los navegadores web modernos.

4.3.4.1. IDE (Entorno de Desarrollo Integrado)

- Visual Studio Code (VSCode): Un IDE ligero, potente y de código abierto que ofrece funcionalidades avanzadas para el desarrollo web.

4.3.4.2. Frontend

- React: Una biblioteca de JavaScript para construir interfaces de usuario interactivas y eficientes.

4.3.4.3. Backend

- Node.js: Un entorno de ejecución para JavaScript del lado del servidor.
- NestJS: Un framework de desarrollo para construir aplicaciones eficientes en Node.js.

4.3.4.4. Capa de Persistencia

- PostgreSQL: Un sistema de gestión de bases de datos relacional de código abierto y potente.
- MongoDB: Una base de datos NoSQL orientada a documentos, escalable y flexible.

4.3.4.5. Control de Tareas

- GitHub Projects: Una herramienta de gestión de proyectos integrada en GitHub que facilita el seguimiento y la organización de tareas.

4.3.4.6. Control de Versiones

- GitHub: Una plataforma de desarrollo colaborativo que utiliza Git para el control de versiones.

4.3.4.7. Automatización de Integración y Despliegue (CI/CD):

- GitHub Actions: Permite la automatización de tareas como la integración continua y el despliegue continuo directamente desde GitHub.

4.3.4.8. Plataforma de Hosting:

- Render: Una plataforma de alojamiento que soporta el despliegue de aplicaciones web, proporcionando servicios para frontend, backend y bases de datos.

4.3.4.9. Trazas, métricas y logs:

- OpenTelemetry: Una plataforma de observabilidad de código abierto que permite la generación de trazas, métricas y logs para el análisis y la monitorización de aplicaciones en contexto backend y frontend.

4.4. Desarrollo en opción de Proyecto Cerrado

4.4.1 Infraestructura de Desarrollo

4.4.1.1. Gestión de Tareas: GitHub Projects

- GitHub Projects se utilizará para organizar y gestionar las tareas del proyecto. Se crearán tableros con columnas que representen los diferentes estados de las tareas, desde la planificación hasta la implementación y la revisión.

4.4.1.2. Control de Versiones: Git en GitHub

- Git, integrado con GitHub, será la herramienta principal para el control de versiones. Se seguirá el flujo de trabajo de Gitflow para gestionar las ramas y las versiones del código fuente.

4.4.1.3. Entornos de Desarrollo: Desarrollo, Preproducción y Producción

- Se establecerán tres entornos distintos:
 - i. **Desarrollo:** Utilizado por los desarrolladores para trabajar en nuevas características y mejoras. En el se pueden desplegar ramas **develop** y **feature**.
 - ii. **Preproducción:** Donde se realizarán pruebas finales antes de la implementación en producción. En el se pueden desplegar ramas **release**.
 - iii. **Producción:** Entorno en vivo utilizado por los usuarios finales para acceder y utilizar la aplicación Auto-repair. En el se despliega la rama **main**. Todos los entornos se desplegarán en Render, una plataforma de alojamiento que soporta el despliegue de aplicaciones, servicios y bases de datos.

4.4.2. Ciclo de vida del desarrollo

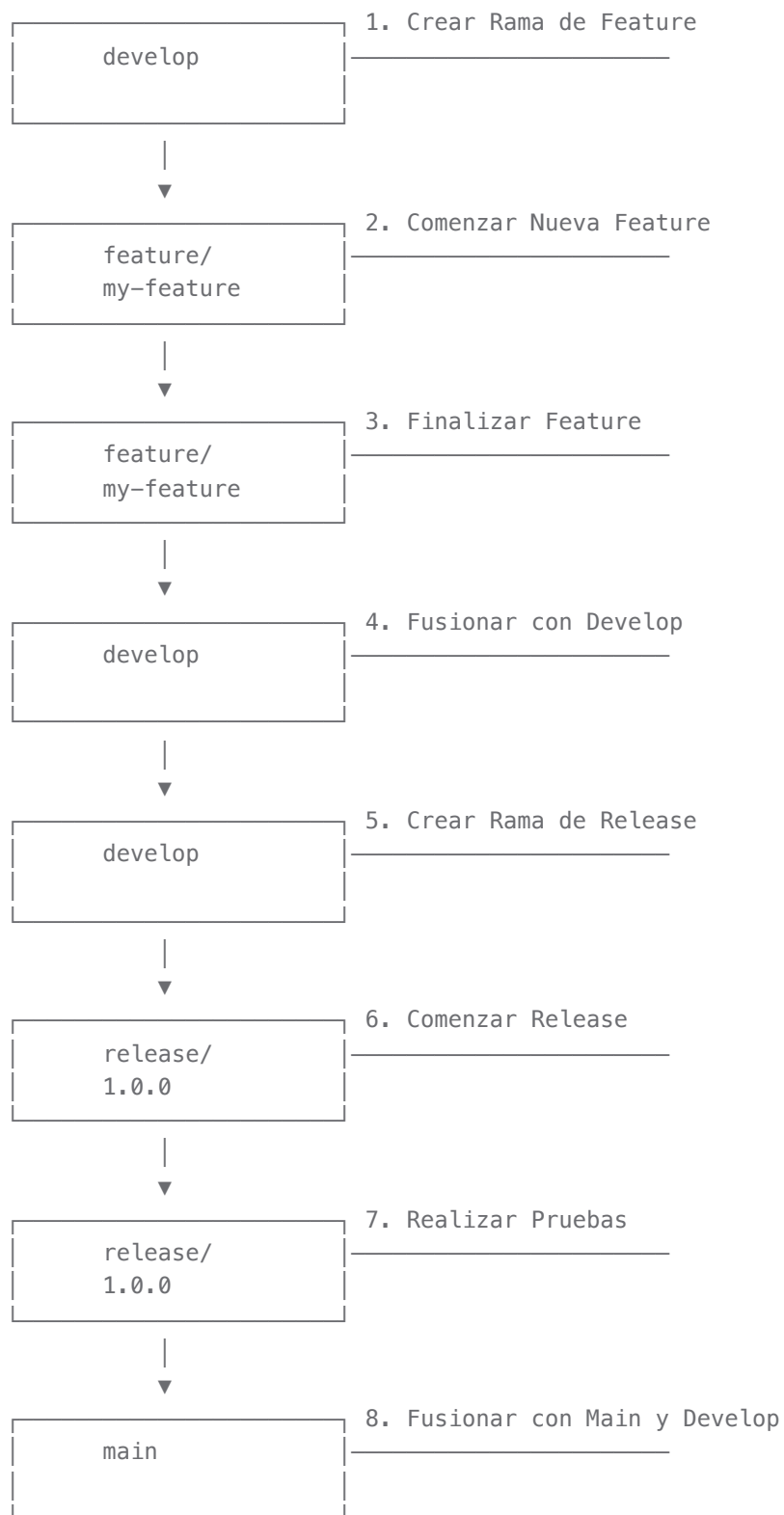
4.4.2.1. Metodología de trabajo: Gitflow

Gitflow es una metodología de flujo de trabajo para proyectos gestionados con Git. Propuesto por Vincent Driessen, se centra en organizar el proceso de desarrollo de software en un conjunto de ramas bien definidas para facilitar la colaboración y la entrega de versiones estables. Aquí hay una breve descripción de las ramas clave y cómo se utilizan:

1. **Develop (Desarrollo):** Esta rama es la rama principal donde se integran todas las características y correcciones de errores. Representa la última versión estable del proyecto.
2. **Feature (Característica):** Se crean ramas de características a partir de la rama de desarrollo para implementar nuevas funcionalidades o mejoras. Una vez que se completa la característica, se fusiona de nuevo en la rama de desarrollo.
3. **Release (Lanzamiento):** Cuando se prepara una nueva versión para su lanzamiento, se crea una rama de lanzamiento a partir de la rama de desarrollo. En esta rama, se pueden realizar pruebas finales y correcciones de errores antes de la versión final. Luego, se fusiona en la rama de desarrollo y se etiqueta como una versión.
4. **Main (Principal):** Esta es la rama principal del proyecto. Las versiones estables se fusionan aquí desde la rama de desarrollo. Cada fusión en la rama principal también se etiqueta con la versión.

Estos son los conceptos básicos de Gitflow. Facilita la gestión de versiones y el trabajo colaborativo, ya que proporciona una estructura clara para la integración de características y lanzamientos.

Ejemplo de flujo gitflow para el desarrollo de una nueva funcionalidad:



4.4.2.2. Integración continua: GitHub Actions

La rama `develop` será gestionada a través de Pull Requests (PR), donde se aplicarán verificaciones automáticas como linting, formateo y ejecución de pruebas antes de que el código sea fusionado. GitHub Actions se encargará de ejecutar estos procesos automáticamente, asegurando la calidad del código antes de su integración.

En el caso de las ramas `release` se desplegarán en el entorno de preproducción de forma automática y pasando los mismos procesos de verificación que en la rama `develop`. Una vez que se compruebe que la versión es estable, se fusionará con la rama `main`, la cual volverá a pasar por los mismos procesos de verificación y se desplegará también de forma automática en el entorno de producción.

Ejemplo de fichero de configuración de GitHub Actions para la rama `develop` :

```
name: Continuous Integration

on:
  pull_request:
  branches:
    - develop

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout Repository
        uses: actions/checkout@v2

      - name: Set Up Node.js
        uses: actions/setup-node@v3
        with:
          node-version: 14

      - name: Install Dependencies
        run: npm install

      - name: Linting and Formatting
        run: npm run lint && npm run format:check

      - name: Run Tests
        run: npm test
```

4.4.2.3. Despliegue continuo: Plataforma Render

Una vez que las Pull Requests son aceptadas y se fusiona código en la rama `develop` , render se encargará de desplegar automáticamente la aplicación en el entorno de desarrollo. Este proceso garantiza una rápida visualización de los cambios implementados y facilita la colaboración en un entorno controlado.

Este mismo proceso se aplicará a las ramas `release` y `main` , desplegando la aplicación en los entornos de preproducción y producción respectivamente.

4.4.2.4. Documentación global: GitHub Pages

La documentación del proyecto estará alojada en GitHub Pages, proporcionando una fuente centralizada de información para desarrolladores, colaboradores y usuarios finales. La actualización de la documentación será parte del flujo de trabajo automatizado para reflejar los cambios realizados en el código y garantizar que la información esté siempre sincronizada con la versión más reciente del software.

4.4.3. Codificación

4.4.3.1. Codificación del Software según Diseño

El proceso de codificación seguirá estándares y buenas prácticas de programación para garantizar la legibilidad, mantenibilidad y eficiencia del código. Algunos principios clave incluirán:

- **Claridad y Consistencia:** Se buscará que el código sea claro y coherente, siguiendo convenciones de nomenclatura comprensibles.
- **Modularidad:** El código se organizará en módulos y componentes independientes para facilitar la comprensión y posibilitar futuras expansiones.
- **Comentarios y Documentación:** Se incluirán comentarios explicativos en el código, y se mantendrá una documentación técnica actualizada que describa la estructura y funcionalidad del sistema.

4.4.3.2. Implementación de Pruebas Unitarias y de Integración

Se adoptará una estrategia integral de pruebas para asegurar la fiabilidad y la robustez del software. Esto incluirá:

- **Pruebas Unitarias:** Se desarrollarán pruebas unitarias para evaluar la funcionalidad de las unidades individuales de código. Se utilizarán herramientas como Jest para el frontend y el backend.
- **Pruebas de Integración:** Se realizarán pruebas de integración para verificar la correcta interacción entre los distintos componentes del sistema. Se emplearán herramientas como Supertest para el backend y Cypress para el frontend.
- **Automatización de Pruebas:** Se implementará un flujo de trabajo que ejecute automáticamente las pruebas con cada cambio de código. GitHub Actions se utilizará para facilitar la integración continua y despliegue seguro.

Este enfoque garantizará la calidad del software y permitirá una identificación temprana de posibles problemas, facilitando la corrección oportuna.

4.4.4. Entrega Inicial: MVP (Minimum Viable Product)

El concepto de **MVP (Minimum Viable Product)** se refiere a la entrega inicial de un producto que incluye solo las características esenciales necesarias para satisfacer las necesidades básicas de los usuarios. En lugar de desarrollar un producto completo con todas las funciones desde el principio, se identifican y implementan las características mínimas que permitirán que el producto sea funcional y útil.

4.4.4.1. Implementación del Software en el Entorno de Producción

Una vez completada la fase de desarrollo y superadas las pruebas, el software se implementará en el entorno de producción. Este despliegue se realizará de forma automatizada mediante GitHub Actions y se verificará para garantizar una transición suave al ambiente de producción.

4.4.4.2. Capacitación del Cliente en el Uso del Software

Se llevará a cabo una sesión de capacitación con el cliente para familiarizarlo con la interfaz y funcionalidades del software. Se proporcionará documentación detallada y soporte para asegurar una adopción exitosa.

4.4.4.3. Recopilación de Comentarios del Cliente

Después de la implementación inicial, se recopilarán los comentarios del cliente sobre su experiencia con el software. Se utilizarán encuestas y sesiones de retroalimentación para entender sus percepciones y necesidades.

4.4.4.4. Identificación de correcciones y mejoras

Con base en los comentarios del cliente, se identificarán posibles correcciones y mejoras. Se categorizarán según su prioridad y complejidad para su inclusión en futuras actualizaciones.

4.4.4.5. Desarrollo e Implementación de correcciones y mejoras

Se procederá con el desarrollo de las correcciones y mejoras identificadas. Estos cambios se integrarán en el código mediante un enfoque de desarrollo ágil y serán desplegados en el entorno de producción de manera continua.

Este ciclo permitirá una evolución constante del software, asegurando que se adapte a las necesidades cambiantes del cliente y proporcionando una experiencia de usuario mejorada.

4.4.5. Entrega final

4.4.5.1. Recopilación de Comentarios del Cliente

Se llevará a cabo una sesión de retroalimentación con el cliente para recopilar comentarios finales sobre la versión entregada y se documentará cuidadosamente todas las sugerencias, observaciones y requisitos adicionales proporcionados por el cliente.

4.4.5.2. Identificación de Oportunidades de Mejora

Se analizarán los comentarios del cliente para identificar oportunidades de mejora y nuevas características, evaluando la viabilidad de implementar las mejoras sugeridas y realizar ajustes según sea necesario.

4.4.6. Mantenimiento y Soporte

4.4.6.1. Monitoreo Continuo del Rendimiento del Software

En el caso de frontend como backend utilizaremos la herramienta de OpenTelemetry para monitorear el rendimiento del software, recopilando trazas, métricas y registros.

4.4.6.2. Resolución de Problemas y Errores

A través de Github Projects, se destinará un tablero para el cliente donde podrá reportar informes de errores, que permitirá a los desarrolladores responder y solucionar rápidamente los problemas y errores identificados y mantener al cliente informado sobre el estado de las correcciones.

4.4.6.3. Actualizaciones de Seguridad y Mejoras Menores

Implementar actualizaciones regulares para abordar problemas de seguridad y realizar mejoras menores, manteniendo actualizaciones de paquetes y dependencias de forma regular y planificada. Garantizar que el software esté siempre actualizado y cumpla con los estándares de seguridad.

4.4.7 Documentación

4.4.7.1. Elaboración de Manuales de Usuario

Crear manuales de usuario detallados que guíen a los usuarios en la operación y utilización efectiva del software. Todo ello se centralizará a través de Github Pages.

4.4.7.2. Documentación Técnica y de Código

- Generar documentación técnica que describa la arquitectura, los componentes y las tecnologías utilizadas en el desarrollo.
- Documentar el código fuente para facilitar la comprensión y el mantenimiento futuro.

De nuevo sacaremos partido de Github Pages para centralizar toda la documentación técnica y de código.

4.4.7.3. Actualización Constante de la Documentación

Se mantendrá la documentación actualizada a medida que se realizan cambios en el software, proporcionando un recurso de referencia completo para el equipo de desarrollo y los usuarios.

5. Bibliografía y Recursos

Git

- [Pro Git](#) de Scott Chacon y Ben Straub.
- [Documentación oficial de Git](#).
- [GitHub Learning Lab](#).

Visual Studio Code (VSCode)

- [Documentación oficial de Visual Studio Code](#).

Node.js y NestJS

- [Documentación oficial de Node.js](#).
- [Documentación oficial de NestJS](#).

OpenTelemetry

- [Documentación oficial de OpenTelemetry](#).

Gitflow

- [Blog del creador de Gitflow, Vincent Driessen](#).

React

- [Documentación oficial de React](#).

Render

- [Documentación oficial de Render](#).

GitHub Projects

- [Documentación oficial de GitHub Projects](#).

GitHub Pages

- [Documentación oficial de GitHub Pages.](#)

GitHub Actions

- [Documentación oficial de GitHub Actions.](#)

PostgreSQL (PostgresDB)

- [Documentación oficial de PostgreSQL.](#)

MongoDB

- [Documentación oficial de MongoDB.](#)

JavaScript

- [Mozilla Developer Network \(MDN\) - JavaScript Guide.](#)

6. Anexos

6.1. Anexo 1: Código fuente

- [frontend](#)
- [backend](#)

6.2. Anexo 2: Entorno productivo

- [Producción](#)