New Digital Talent

TRABAJO FIN DE CICLO

CICLO DESARROLLO DE APLICACIONES MULTIPLATAFORMA

iMirly

Autores

Ana Nuñez Rejon

Ángel Millán Aragón

Eva Gallardo Romero

Nicolás Corbi de Gracia

Tutores

Guillermo Sandoval Schmidt

Javier Martín Gómez

Granada, dia mes 2026

RESUMEN EJECUTIVO

iMirly nace como una **aplicación multiplataforma** disruptiva diseñada para establecer una conexión directa y eficiente entre usuarios que demandan servicios profesionales y proveedores, abarcando desde autónomos independientes hasta empresas y constructoras. El proyecto surge tras detectar una **fragmentación crítica en el mercado actual**, donde las soluciones existentes se polarizan entre modelos de microservicios domésticos con altas comisiones y modelos de "leads" para grandes obras con costes de adquisición elevados y sin garantía de retorno.

La **oportunidad de negocio** de iMirly reside en su posicionamiento como una **plataforma híbrida "pro-profesional"** que centraliza en un único entorno digital la búsqueda, contratación, pago y valoración de servicios de sectores tan diversos como las reformas, la limpieza o el mantenimiento.

A diferencia de la competencia, iMirly elimina los intermediarios opacos y ofrece un **valor añadido** fundamental basado en la **transparencia y la confianza**. Esto se logra mediante un sistema de **verificación dual (DNI/CIF)** que garantiza la legalidad del proveedor y una infraestructura que permite la **facturación B2B automatizada**, resolviendo así uno de los mayores puntos de fricción del sector.

La **viabilidad del proyecto** se sustenta en dos pilares:

1. **Viabilidad Comercial:** Se apoya en el éxito validado de modelos de negocio similares, pero optimizados al eliminar comisiones abusivas y ofrecer mayor seguridad jurídica tanto al cliente como al profesional.
2. **Viabilidad Técnica:** El uso de tecnologías nativas y escalables de última generación, como **Kotlin con Jetpack Compose** para el desarrollo Android y **Spring Boot (Java)** bajo una **arquitectura hexagonal** para el backend, asegura un sistema robusto, un mantenimiento eficiente y una experiencia de usuario fluida con persistencia en **PostgreSQL**.

En definitiva, iMirly no es solo una herramienta de contacto, sino un **ecosistema integral** que profesionaliza la contratación de servicios digitales, garantizando rentabilidad para el proveedor y seguridad para el consumidor final.

ÍNDICE GENERAL

ÍNDICE GENERAL.... 4
ÍNDICE TABLAS.... 6
ÍNDICE FIGURAS.... 7
1. CONTEXTUALIZACIÓN EMPRESARIAL.... 8
1.1 Denominación.... 8
1.2. Justificación de la idea del proyecto.... 8
1.3. Estudio de la competencia.... 10
1.3.1. Diagnóstico del Mercado y Puntos de Fricción.... 11
1.3.2. Análisis DAFO.... 11
1.3.3. Conclusión Estratégica y Posicionamiento.... 12
1.4. Oportunidad de negocio.... 13
1.4.1. Nicho de mercado y diferenciación.... 13
1.4.2. Demostración de Viabilidad.... 13
1.5 Obligaciones fiscales.... 14
1. Régimen fiscal.... 14
2. Fiscalidad.... 14
3. Protección de datos (RGPD/LOPD).... 14
4. Gestión de pagos.... 15
1.6 Financiación, ayudas y subvenciones.... 15
1.6.1 Ayudas a través del instituto.... 15
1.6.2 Subvenciones y financiación pública local (Granada).... 15
1.6.3 Opciones para Startups o Empresas Constituidas.... 16
2. DISEÑO DEL PROYECTO.... 17
2.1. Recursos.... 17
2.1.1. Hardware.... 17
2.1.2. Software (Equipamiento Lógico).... 17
2.1.2.1 Entornos, Lenguajes y Frameworks a Emplear.... 18
2.1.2.2 Control de Versiones y Gestión.... 18
2.2. Metodología.... 19
2.3 Viabilidad.... 20
3. DESARROLLO DEL SERVICIO/PRODUCTO.... 22
3.1. Fase 1: Análisis y planificación.... 22
A. Planificación Temporal.... 22
B. Gestión de Tareas y Metodología (Trello).... 23
3.2 Fase 2: Diseño de la aplicación.... 29
3.2.1 Identidad Visual y Guía de Estilo.... 29
A. Evolución y Diseño del Logotipo.... 29
B. Desarrollo de la Identidad Cromática.... 30

C. Paleta de Colores y Tipografía Final..... 31
3.2.2 Diseño de Pantallas Principales.....32
3.2.3 Prototipo Interactiva de Alta Fidelidad.....33
3.2.4. Validación de Diseño: Análisis de la Encuesta de Usuario.....33
A. Presentación de Alternativas de Diseño.....33
B. Análisis de Resultados y Preferencias.....34
C. Disposición al Uso de la Aplicación..... 35
D. Conclusión del Diseño Final..... 35
3.3. Fase 3: Desarrollo del Sistema.....36
A. Arquitectura General y Comunicaciones.....36
B. Arquitectura del Backend y Estructura del Código.....36
C. Desarrollo Frontend (Aplicación Android Nativa)..... 38
D. Base de Datos y Persistencia..... 38
3.4. Fase 4: Pruebas y Control de Calidad..... 39
A. Pruebas Técnicas: Unitarias y de Integración.....39
B. Pruebas de Usabilidad y Validación Funcional.....39
C. Pruebas de Rendimiento y Estabilidad.....40
D. Gestión de Incidencias y Evidencia Técnica.....40
4. PROCEDIMIENTOS Y CONTROLES DE CALIDAD.....41
4.1 Control de calidad durante el desarrollo..... 41
4.2. Indicadores de calidad (KPIs)..... 42
4.3 Procedimientos de mejora continua y gestión de incidencias.....43
A. Registro y resolución de incidencias (Trello).....43
B. Ciclos de revisión y retroalimentación.....44
4.4. Documentación de control.....44
5. DOCUMENTACIÓN.....46
5.1. Manual de Usuario..... 46
5.2. Manual de Administrador.....46
5.3. Guía de Instalación y Despliegue..... 46
5.4. Documentación Técnica..... 47
5.5. Tutoriales y Material de Apoyo.....47
6. ANEXOS.....48
7. REFERENCIAS BIBLIOGRÁFICAS..... 49

ÍNDICE TABLAS

- Tabla 1: Diagnóstico de necesidades en el mercado de servicios
- Tabla 2: Análisis Competitivo del Mercado Actual
- Tabla 3: Puntos de Fricción Identificados en el Sector
- Tabla 4: Análisis DAFO (SWOT) de iMirly
- Tabla 5: Matriz de Oportunidad y Viabilidad de iMirly
- Tabla 6: Resumen de Opciones de Financiación y Ayudas
- Tabla 7: Resumen de Equipamiento Lógico (Software)
- Tabla 8: Pilares de la Metodología de Trabajo de iMirly
- Tabla 9: Justificación de la Viabilidad Técnica y de Planificación
- Tabla 10: Leyenda Diagrama de Gantt
- Tabla 11: Requisitos Funcionales (RF)
- Tabla 12: Requisitos No Funcionales (RNF)
- Tabla 13: Resumen de Feedback Cualitativo
- Tabla 14: Indicadores Clave de Desempeño (KPIs) de iMirly

ÍNDICE FIGURAS

- Figura A: Logotipo oficial de iMirly
- Figura B: Diagrama de Gantt
- Figura C: Captura del tablero Trello de iMirly
- Figura D: Diagrama Entidad-Relación de la base de datos de iMirly.
- Figura E: Diagrama de procesos y ciclo de vida del contrato.
- Figura F: Propuestas iniciales para el logotipo de iMirly.
- Figura G: Primera fase de exploración cromática.
- Figura H: Segunda fase de refinamiento (Rama de azules/turquesas).
- Figura I: Paleta de colores y guía de estilo
- Figura J: Esquemas y bocetos iniciales en papel
- Figura K: Wireframes digitales de iMirly
- Figura L: Captura de pantalla de la Opción de Diseño 1
- Figura M: Captura de pantalla de la Opción de Diseño 2
- Figura N: Arquitectura integral del sistema iMirly.
- Figura Ñ: Detalle de la Arquitectura Hexagonal del Servidor.
- Figura O: Captura del tablero Trello de iMirly con el flujo de incidencias

1. CONTEXTUALIZACIÓN EMPRESARIAL

1.1 Denominación

iMirly es el nombre comercial de una **aplicación multiplataforma** diseñada para establecer una conexión directa y eficiente entre usuarios que demandan servicios profesionales y los proveedores —tanto **autónomos** como **empresas** y constructoras— que los ofrecen.
La plataforma nace con el propósito de centralizar y simplificar todo el proceso de prestación de servicios en un entorno digital único, facilitando herramientas de **búsqueda, contratación, gestión de pagos y valoración** de servicios de forma rápida y segura. A diferencia de otras soluciones existentes, iMirly se define por su capacidad de integrar desde microservicios domésticos hasta grandes obras, unificando diferentes sectores como reformas, limpieza o mantenimiento.

Figura A: Logotipo oficial de iMirly

1.2. Justificación de la idea del proyecto

La creación de iMirly surge de la detección de una oportunidad de mercado estratégica derivada de la falta de competencia directa que integre servicios profesionales de diversa escala en una única plataforma. Actualmente, el sector de la contratación de servicios digitales presenta una marcada polarización entre modelos basados en comisiones para microservicios y modelos de "leads" para grandes obras, dejando un vacío para una solución híbrida y centralizada.
Las razones fundamentales que justifican la puesta en marcha de este proyecto son:

- **Fragmentación del Mercado**: No existe actualmente una aplicación que unifique la búsqueda y gestión de servicios tanto para profesionales autónomos como para empresas y constructoras de mayor envergadura.

- **Resolución de Puntos de Fricción:** Los modelos actuales presentan barreras críticas, como comisiones elevadas en tareas domésticas (Webel, Wolly) o costes de adquisición de clientes muy altos sin garantía de cierre en grandes reformas (Habitissimo).
- **Necesidad de Transparencia y Confianza:** Se ha identificado una demanda por parte de los clientes de contar con un sistema de verificación dual (DNI/CIF) que garantice si el proveedor es un particular o una empresa registrada.
- Eficiencia Operativa: iMirly busca eliminar la incapacidad de emitir facturación B2B formal en plataformas de comisión, integrando la gestión de pagos y la facturación automatizada en el propio ecosistema.

Tabla 1: Diagnóstico de necesidades en el mercado de servicios

Necesidad Detectada	Situación Actual del Mercado	Impacto en el Usuario / Profesional	Propuesta de Valor de iMirly
Centralización de servicios	Mercado fragmentado entre microservicios (Webel) y grandes obras (Habitissimo).	Obliga a usar múltiples plataformas según la envergadura del proyecto.	**Plataforma híbrida** que unifica desde tareas domésticas hasta grandes reformas.
Reducción de costes operativos	Comisiones elevadas (15-25%) o pago por leads sin garantía de retorno.	Baja rentabilidad para el autónomo y encarecimiento del servicio para el cliente.	Modelo de **monetización equilibrado** (comisión/suscripción) que reduce el riesgo del profesional.
Seguridad Jurídica y Confianza	Opacidad sobre la legalidad del proveedor y falta de diferenciación clara.	Incertidumbre del cliente sobre si contrata a un particular o a una empresa registrada.	Sistema de **verificación dual (DNI/CIF)** que garantiza la transparencia.
Gestión Administrativa	Incapacidad de emitir facturación B2B formal en plataformas de microservicios.	Dificultad para empresas y profesionales de deducir gastos y gestionar cobros.	Infraestructura de **facturación B2B automatizada** integrada en el ecosistema.

Eficiencia en la Contratación	Modelos de monetización confusos (créditos/contactos) que generan fricción.	Pérdida de tiempo y dinero del profesional al pagar por contactar sin cierre asegurado.	Entorno digital único para la **búsqueda, contratación y pago seguro** sin intermediarios opacos.

1.3. Estudio de la competencia

En este apartado se procede a identificar y analizar las empresas con proyectos en marcha que representan una competencia, tanto directa como indirecta, para iMirly. El mercado actual de plataformas de servicios se encuentra fuertemente segmentado en dos modelos predominantes: el de comisión por servicio y el de pago por "lead" o contacto.
Siguiendo el estilo analítico de las fuentes, se presenta la siguiente comparativa de los principales actores del sector:

Tabla 2: Análisis Competitivo del Mercado Actual

Plataforma	Modelo de Negocio	Enfoque Principal	Análisis Crítico (Limitaciones)
Webel	Comisión por servicio (15–25%)	Tareas domésticas y microservicios.	No permite facturación B2B. Modelo limitado a profesionales individuales.
Habitissimo	Pago por lead o suscripción.	Reformas, obras y servicios B2B.	Alto coste de adquisición para el profesional, que paga sin garantía de cierre.
Cronoshare	Pago por créditos (contactos).	Amplio (clases, eventos, reformas).	Modelo de monetización confuso. Genera fricción al pagar por contactar.
Wolly	Comisión por servicio (con seguro).	Servicios del hogar.	Buen enfoque en confianza, pero carece de diferenciación empresarial clara.

1.3.1. Diagnóstico del Mercado y Puntos de Fricción

El análisis revela una clara polarización. Por un lado, los modelos basados en comisión (Webel, Wolly) ofrecen una baja barrera de entrada pero presentan comisiones elevadas y escasa diferenciación profesional. Por otro lado, los modelos basados en "leads" (Habitissimo, Cronoshare) se orientan a proyectos de mayor envergadura pero generan una alta fricción debido al pago anticipado sin garantía de retorno de la inversión (ROI).
Esta situación ha generado una serie de puntos de fricción que iMirly busca resolver, detallados en la siguiente tabla:

Tabla 3: Puntos de Fricción Identificados en el Sector

Perspectiva	Problemas Críticos Identificados
Del Profesional	Visibilidad inconsistente, gestión operativa limitada, liquidación de pagos lenta y **ausencia de facturación B2B** formal.
Del Cliente	**Déficit de confianza** (incertidumbre sobre la legalidad del proveedor), baja densidad de profesionales en ciertas zonas y opacidad en los precios finales.

1.3.2. Análisis DAFO

Para sintetizar la situación estratégica de iMirly frente a su entorno y competidores, se presenta la siguiente matriz de análisis:

Tabla 4: Análisis DAFO (SWOT) de iMirly

Fortalezas (S)	**Debilidades (W)**
Poca competencia directa especializada en	Plataforma emergente sin base de usuarios

servicios locales variados	inicial (falta de reconocimiento de marca inicial)
Alta demanda de servicios de proximidad	Brecha digital de los usuarios, sobretodo los más mayores
Digitalización de servicios tradicionales	Recursos económicos limitados
Modelo flexible ya que iMirly puede trabajar con dos tipos de clientes distintos	Poca experiencia empresarial al inicio
Oportunidades (O)	**Amenazas (T)**
Captación de profesionales insatisfechos con modelos de leads (Habitissimo) o comisiones (Webel).	Competencia consolidada (Webel) en mercados urbanos clave.
Tendencia a la economía colaborativa y local	Altos costes de adquisición de usuarios (CAC) en la fase inicial.
Diferenciación basada en la transparencia, verificación empresarial y agilidad en los pagos.	Riesgos de ciberseguridad y protección de datos a la par que riesgos legales con la prestación de servicios
Aumento del trabajo autónomo y freelance	En caso de disconformidad con algún servicio prestado, ¿quién asume la responsabilidad?

1.3.3. Conclusión Estratégica y Posicionamiento

iMirly parte con una **fuerte oportunidad de mercado** y un **bajo coste de entrada**, pero debe centrarse en **generar confianza**, **cumplir la normativa legal** y **reforzar su imagen de marca** para competir con plataformas más grandes. Su propuesta de valor central reside en ofrecer un entorno transparente donde clientes, profesionales y empresas colaboran sin intermediarios opacos, facilitando la verificación empresarial y la gestión integral de la facturación. El objetivo es establecer un entorno donde autónomos y empresas coexistan, otorgando ventajas de visibilidad claras a las entidades legalmente verificadas.

1.4. Oportunidad de negocio

En este apartado se justifica la viabilidad del proyecto basándose en la demanda actual del mercado y la propuesta de valor diferencial de iMirly. La oportunidad de negocio se fundamenta en la capacidad de la plataforma para capturar un segmento del mercado que actualmente se encuentra desatendido por las soluciones existentes.

1.4.1. Nicho de mercado y diferenciación

A diferencia de los competidores analizados, iMirly se diferencia al dar la opción de visibilizar de forma conjunta a empresas y constructoras, no solo a trabajadores autónomos. Esta integración amplía significativamente el mercado potencial, permitiendo que la aplicación escale desde pequeñas tareas domésticas hasta proyectos de edificación de gran envergadura.
Para visualizar esta oportunidad, se presenta la siguiente tabla que resume los pilares de rentabilidad del proyecto:

Tabla 5: Matriz de Oportunidad y Viabilidad de iMirly

Factor de Oportunidad	Descripción del Valor Añadido	Sustento de Viabilidad
Demanda Insatisfecha	Usuarios que buscan **seguridad legal** (empresas con CIF) en un entorno digital centralizado.	Éxito probado de modelos como Webel o Habitissimo en sectores segmentados.
Escalabilidad	Capacidad de cubrir desde **microservicios** hasta licitaciones de obras grandes.	Arquitectura modular y uso de servicios en la nube (**AWS/Google Cloud**).
Monetización Híbrida	Ingresos a través de comisiones por servicio y suscripciones para el **"Modo Pro"**.	Modelo económico equilibrado que reduce el riesgo para el profesional.

1.4.2. Demostración de Viabilidad

El proyecto es económicamente viable debido a que aprovecha el éxito ya validado de otras plataformas de contratación de servicios, pero eliminando sus principales puntos de fricción, como las comisiones abusivas o el pago por contactos sin garantía.

Técnicamente, la viabilidad está asegurada mediante el uso de **tecnologías nativas y escalables de última generación**. El sistema se fundamenta en una aplicación **Android nativa desarrollada en Kotlin con Jetpack Compose** para el frontend. Por su parte, el backend se apoya en un servidor robusto basado en **Java 21** y **Spring Boot 3** diseñado bajo una arquitectura hexagonal. Esta estructura modular separa totalmente la lógica de negocio de los detalles técnicos, facilitando un mantenimiento eficiente, una alta escalabilidad y un despliegue rápido, contando con una persistencia de datos fiable en **PostgreSQL**.

1.5 Obligaciones fiscales

La aplicación gestionará datos de clientes y empresas, cobros y contraseñas. Las obligaciones fiscales y legales en España son:

1. Régimen fiscal

- **Autónomo:** alta en Hacienda (036/037) y Seguridad Social (RETA), IRPF e IVA según ingresos.
- **Sociedad Limitada:** persona jurídica independiente, contabilidad formal, Impuesto de Sociedades y retención de dividendos.

2. Fiscalidad

- **IVA:** servicios digitales a particulares UE (OSS), exento para empresas con NIF intracomunitario; Modelo 303 trimestral y 390 anual; Modelo 349 para operaciones intracomunitarias.
- **IRPF/IS:** autónomo paga IRPF (Modelo 130), SL paga Impuesto de Sociedades (25%).
- **Seguridad Social:** autónomo cuota mínima ~400 €/mes; SL cotización según empleados/administradores.

3. Protección de datos (RGPD/LOPD)

- Registro de actividades de tratamiento.
- Política de privacidad y consentimiento de usuarios.
- Cifrado de contraseñas y datos sensibles.
- Derechos ARCO y notificación de brechas a la AEPD en 72h.

4. Gestión de pagos

- Plataformas de terceros (Stripe, PayPal, App Stores) simplifican el cumplimiento PCI-DSS.
- Pagos directos requieren cumplimiento PCI-DSS y emisión de facturas.

1.6 Financiación, ayudas y subvenciones

Como alumnos de Formación Profesional, partimos de una premisa administrativa clave: no es posible solicitar directamente la mayoría de las ayudas como particulares. Las convocatorias requieren que la solicitud sea gestionada por el instituto o que el proyecto esté vinculado a una figura jurídica formal (autónomo o empresa).

1.6.1 Ayudas a través del instituto

- Ayudas del Ministerio de Educación para Proyectos de Innovación en FP: Estas ayudas exigen que el centro educativo lidere la propuesta bajo líneas temáticas específicas. iMirly se encuadra en:
    - Innovación tecnológica y digitalización: Desarrollo de competencias en industria 4.0 y redes 5G.
    - Transferencia de conocimiento: Colaboración entre empresas y centros de FP.
    - Emprendimiento: Creación de estructuras como viveros o incubadoras para facilitar la transición al mercado laboral.
- Proyectos Colaborativos de FP (Andalucía): Convocatoria para fomentar el autoempleo. Limitación importante: Solo es aplicable a centros públicos; no puede solicitarse desde centros privados ni por sus alumnos.

1.6.2 Subvenciones y financiación pública local (Granada)

En caso de formalizar la actividad económica, existen opciones en el entorno provincial:

- Diputación de Granada – Jóvenes Emprendedores (2025): Dirigida a personas de 18 a 35 años empadronadas en municipios de menos de 30.000 habitantes. Requiere alta en el RETA o constitución de empresa.
- Subvención para desarrollo socioeconómico: Orientada a la innovación empresarial. Limitación: Dado que el centro de formación se ubica en Granada capital, no se cumplen los requisitos de ubicación geográfica para ciertas líneas de estas ayudas locales.

1.6.3 Opciones para Startups o Empresas Constituidas

Si iMirly se lanza como una iniciativa empresarial real, las opciones son más ambiciosas:

- Innovactiva 2025 (Junta de Andalucía): Ofrece hasta 9.000 € para poner en marcha el proyecto.
    - Requisitos: Tener entre 18 y 35 años, poseer titulación de FP y tener domicilio fiscal en Andalucía.
    - Gastos subvencionables: Solo gastos de inversión (infraestructura, equipamiento, desarrollo), no gastos corrientes.
- Acelera Startups Andalucía: Programa de mentorización, talleres y networking para proyectos con intención real de lanzamiento. No es una subvención directa, sino apoyo al crecimiento y acceso a inversores.

Tabla 6: Resumen de Opciones de Financiación y Ayudas

Programa	Beneficio	Requisito Crítico	Viabilidad Real
Ministerio de Educación	Recursos técnicos y apoyo a la innovación digital.	Solicitud tramitada obligatoriamente vía Instituto.	**Media:** Sujeta a la apertura de convocatorias específicas.
Innovactiva 2025	Subvención directa de hasta **9.000 €** para gastos de inversión.	Poseer título de FP, entre 18-35 años y domicilio fiscal en Andalucía.	**Alta:** El perfil de los autores encaja tras la obtención del título.
Diputación de Granada	Ayuda económica para jóvenes emprendedores.	Empadronamiento en municipios de menos de 30.000 habitantes.	**Baja:** La ubicación en Granada capital actúa como criterio de exclusión.
Acelera Startups Andalucía	Programa de mentoría, talleres y redes de inversión.	Contar con una estructura formal o un MVP en fase de despliegue.	**Media:** Factible una vez lanzada la fase de consumo real de la app.

2. DISEÑO DEL PROYECTO

En este apartado haremos un breve recorrido por los recursos que utilizaremos a nivel software y hardware, ya que a nivel de recursos humanos compuestos por el equipo de cuatro estudiantes(con la supervisión de los tutores); mediante diagramas de Gantt veremos la planificación temporal del proyecto; hablaremos de la metodología de desarrollo escogida; y finalmente comentaremos la viabilidad del proyecto.

2.1. Recursos

El proyecto iMirly, siendo una **aplicación multiplataforma enfocada al entorno digital**, no requiere instalaciones físicas permanentes, aunque sí se necesitaría una oficina técnica o un espacio de *coworking* en el futuro para gestionar las tareas de soporte y atención al cliente.

2.1.1. Hardware

Los recursos de hardware previstos para el desarrollo del proyecto son los siguientes:

- **Equipo principal: Equipos espacio de trabajo**
    - CPU: Intel(R) Core(TM) i7-7700K 4.2GHz. 1 *sockets* con 4 núcleos/*socket*.
    - GPU:NVDIA GeForce RTX 4060
    - RAM: 16 GB

Además de los equipos de cómputo, se requiere equipamiento físico adicional, como dispositivos móviles Android e iOS para realizar pruebas de compatibilidad, y mobiliario básico (mesas, sillas ergonómicas, pantallas adicionales).

2.1.2. Software (Equipamiento Lógico)

El proyecto iMirly requiere un conjunto robusto de equipamiento lógico para garantizar su rendimiento, escalabilidad y mantenimiento a largo plazo. El desarrollo se ha centrado en tecnologías nativas de alto rendimiento que aseguran una experiencia de usuario fluida y una arquitectura de servidor profesional.

2.1.2.1 Entornos, Lenguajes y Frameworks a Emplear

El software de desarrollo se basa en herramientas que proporcionan flexibilidad y total control sobre el hardware:

- **Sistemas Operativos y IDEs**: El trabajo se realiza sobre sistemas Windows, utilizando principalmente **Android Studio** para el cliente y **IntelliJ IDEA** o **VS Code** para el servidor.
- **Frontend (App)**: La aplicación móvil es nativa para Android, desarrollada íntegramente en **Kotlin** utilizando el toolkit declarativo **Jetpack Compose**. Esta combinación permite una UI moderna y una gestión eficiente de estados mediante ViewModels y Corrutinas.
- **Backend (Lógica)**: El servidor API se ha implementado con **Java 21** y **Spring Boot 3**, siguiendo una **Arquitectura Hexagonal (Clean Architecture)**. Este enfoque separa las reglas de negocio de la infraestructura, facilitando el testeo y la escalabilidad.
- **Infraestructura de Datos y Seguridad**: La persistencia de datos se gestiona mediante **PostgreSQL**. La seguridad y el control de acceso se implementan con **Spring Security**, utilizando **Tokens JWT** para una comunicación stateless y segura entre el cliente y el servidor.
- **Consumo de API y Servicios**: Se utiliza **Retrofit** y **OkHttp** para la comunicación en red. Para el despliegue y desarrollo se contemplan servicios en la nube como AWS o Google Cloud Platform.
- **Diseño:** Se emplearán herramientas de diseño y prototipado como **Figma, Adobe XD o Canva**.

2.1.2.2 Control de Versiones y Gestión

Para el control del código y la facilitación del trabajo colaborativo (Requisito RNF-5.2), se emplean:

- **Git y GitHub**: Como pilar central del control de versiones y revisiones de código (code reviews).
- **Trello**: Para la gestión ágil del flujo de trabajo, registro de incidencias y seguimiento de tareas.

Tabla 7: Resumen de Equipamiento Lógico (Software)

Categoría	Herramientas Seleccionadas	Propósito y Justificación
Sistemas Operativos	Windows	Entornos base para la ejecución de los IDEs y simuladores.
Entornos de Desarrollo (IDE)	Android Studio / IntelliJ IDEA	Herramientas principales para la codificación del ecosistema.
Frontend / App	**Kotlin (Jetpack Compose)**	Desarrollo nativo Android para una experiencia fluida y moderna.
Backend / Lógica	**Java 21 / Spring Boot 3**	Arquitectura Hexagonal para máxima escalabilidad y mantenimiento.
Base de Datos y Auth	**PostgreSQL / JWT**	Gestión robusta de persistencia y seguridad de accesos.
Control de Versiones	Git / **GitHub**	Garantía del trabajo colaborativo y cumplimiento del requisito **RNF-5.2**.
Gestión y Diseño	Trello / Figma / Canva	Coordinación de tareas, prototipado UI/UX y comunicación

2.2. Metodología

El modelo de trabajo del proyecto iMirly se basa en una metodología **estructurada en fases** que facilita un enfoque **incremental y modular**. Este enfoque permite que las distintas funcionalidades del sistema sean desarrolladas, implementadas y testeadas de manera constante a lo largo del tiempo.

Los pilares de esta metodología son:

- **Modularidad:** El sistema se estructura en **módulos independientes** para facilitar futuras actualizaciones del software.
- **Control de Versiones y Colaboración**: Se empleará exclusivamente el control de versiones mediante **Git y GitHub**. El uso de **GitHub** es un requisito no funcional (**RNF-5.2**) diseñado explícitamente para facilitar el trabajo colaborativo, la gestión de ramas y el seguimiento del histórico de cambios.

- **Aseguramiento de Calidad:** La metodología exige que **cada módulo desarrollado sea revisado mediante *code reviews*** en GitHub antes de su integración. También se

implementarán **pruebas automatizadas** para las principales funciones, y el **registro de incidencias y mejoras en Trello** permitirá la mejora continua.

Esta metodología incremental asegura la obtención de versiones funcionales del producto (MVP inicial en Fase 3) que permiten realizar pruebas y mejoras durante todo el desarrollo.

Tabla 8: Pilares de la Metodología de Trabajo de iMirly

Pilar Metodológico	Descripción y Aplicación	Herramientas / Requisitos
Enfoque Incremental	Estructura en cinco fases estratégicas para asegurar la obtención de productos mínimos viables (MVP).	Diagrama de Gantt.
Modularidad	Desarrollo de funciones en módulos independientes para facilitar futuras actualizaciones y escalabilidad.	Arquitectura de software modular (Hexagonal)
Desarrollo Colaborativo	Gestión del código fuente para permitir el trabajo en equipo y mantener un histórico de cambios detallado	Git y GitHub (**RNF-5.2**).
Control de Calidad	Revisiones de código (code reviews) obligatorias antes de la integración y ejecución de pruebas automatizadas	**GitHub**
Mejora Continua	Registro sistemático de incidencias, errores (bugs) y propuestas de mejora.	**Trello** (Tablero de incidencias).

2.3 Viabilidad

A la vista de la planificación y los recursos disponibles, el proyecto iMirly es **viable** en la forma y plazos propuestos.

La viabilidad se respalda en:

- Recursos (Hardware y Software): Se cuenta con recursos materiales y lógicos más que suficientes, incluyendo ordenadores adecuados, dispositivos móviles para pruebas, entornos de desarrollo modernos (Android Studio, VS Code) y frameworks de desarrollo eficiente (Kotlin y Spring Boot)". La infraestructura en la nube (AWS, Google Cloud Platform o Render) está prevista para el desarrollo y producción.

- **Planificación y Metodología:** La implementación de un **Diagrama de Gantt** y una **planificación clara** dividida en cinco fases, junto con una metodología incremental y modular, facilita un **desarrollo constante del proyecto** y permite la introducción de cambios con facilidad en la aplicación, lo que asegura el cumplimiento de los plazos.

Tabla 9: Justificación de la Viabilidad Técnica y de Planificación

Pilar de Viabilidad	Elementos Clave	Garantía de Éxito
Infraestructura de Hardware	Equipos de alto rendimiento (RTX 4060, 16GB RAM) y dispositivos móviles para pruebas,.	Capacidad de procesamiento suficiente para el desarrollo, emulación y testeo sin cuellos de botella técnicos.
Stack Tecnológico Moderno	Uso de Kotlin, Spring Boot y servicios en la nube (AWS/Google Cloud)...	Asegura un mantenimiento eficiente, portabilidad multiplataforma y una experiencia de usuario fluida,.
Gestión Cronológica	Planificación detallada en **5 fases** y un cronograma de **30 semanas.**	El seguimiento mediante el **Diagrama de Gantt** asegura el cumplimiento de los hitos y la entrega del MVP.
Metodología y Control	Desarrollo incremental, modular y gestión de tareas/incidencias en **Trello**.	Permite la introducción de cambios con facilidad y garantiza una mejora continua basada en el testeo real,.

3. DESARROLLO DEL SERVICIO/PRODUCTO

El desarrollo de **iMirly** se articula como un proceso integral que abarca desde la concepción analítica hasta el despliegue y mantenimiento del software. Siguiendo una **metodología incremental y modular**, el proyecto se divide en cinco fases estratégicas que aseguran la obtención de productos mínimos viables (MVP) y permiten una mejora continua basada en el testeo real.

3.1. Fase 1: Análisis y planificación

Refleja el tiempo previsto para definir los requerimientos funcionales y técnicos de la aplicación, incluyendo el análisis del mercado, la definición de los tipos de usuarios (empresas, autónomos, clientes) y la elaboración de casos de uso y funcionalidades principales.
Esta fase constituye el cimiento técnico del proyecto, donde se definen los alcances y la hoja de ruta temporal.

A. Planificación Temporal

Para garantizar el cumplimiento de los hitos establecidos, se ha diseñado un cronograma detallado que distribuye las tareas en un horizonte temporal de 30 semanas. Como se observa en la **Figura B**, este esquema permite visualizar los solapamientos entre las tareas de investigación, desarrollo y documentación.

Figura B: Diagrama de Gantt

Para facilitar la lectura del cronograma, la **Tabla 10** detalla la correspondencia entre las siglas del diagrama y los periodos temporales reales:

Tabla 10: Leyenda Diagrama de Gantt

Fase / Tarea	Color en Diagrama	Descripción y Alcance Técnico
Análisis	**Cian**	Definición de requerimientos funcionales (RF) y no funcionales (RNF).
Planificación	**Verde**	Establecimiento de la hoja de ruta temporal y alcances del proyecto.
Investigación	**Rosa**	Análisis del mercado de servicios y diagnóstico de la competencia.
Diseño	**Amarillo**	Creación de la identidad visual, wireframes y prototipo interactivo en Figma.
Desarrollo	**Morado**	Implementación del Frontend (Kotlin/Compose) y Backend (Spring Boot/JPA).
Documentación	**Naranja**	Elaboración de manuales técnicos, de usuario y registros de control de calidad.
Presentación	**Azul**	Preparación de materiales para la defensa final y cierre de hitos.
Estado: Planificado	**Tono Oscuro**	Representa la estimación de tiempo inicial prevista para la tarea.
Estado: Real	**Tono Claro**	Representa el tiempo efectivo de ejecución y seguimiento del trabajo.

B. Gestión de Tareas y Metodología (Trello)

La organización operativa se apoya en el uso de la herramienta **Trello**, la cual permite una gestión ágil de los flujos de trabajo mediante tarjetas y tableros. En la **Figura C** se muestra una captura del estado del tablero, donde se centraliza el registro de incidencias, mejoras y el seguimiento de las tareas en curso.

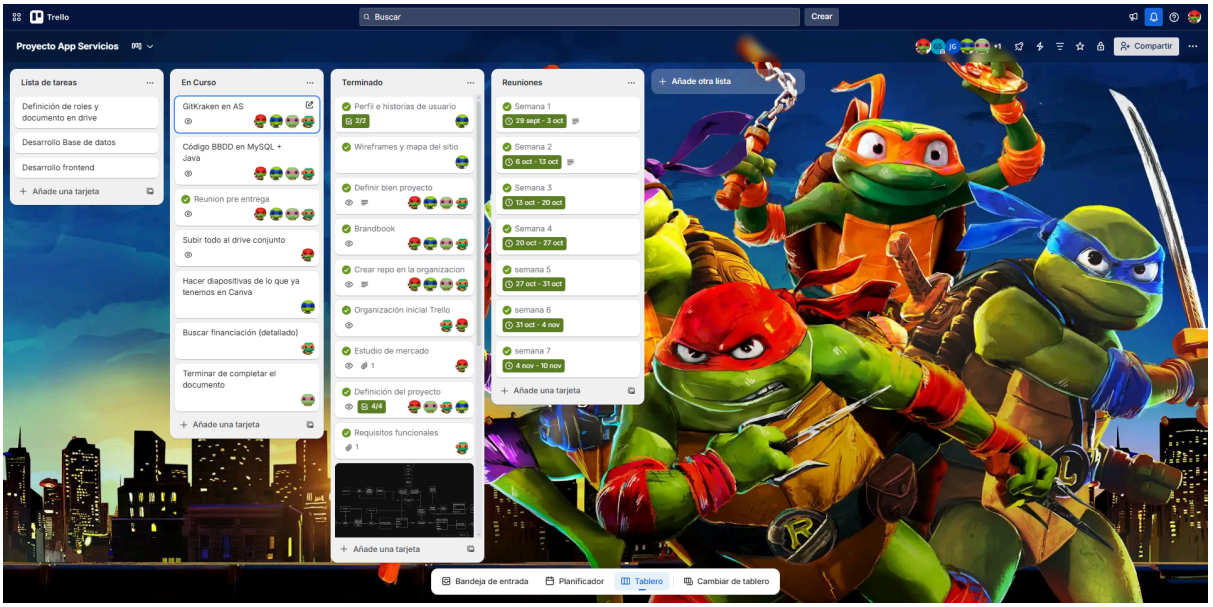

Figura C: Captura del tablero Trello de iMirly

C. Definición de Requerimientos (RF y RNF) La identificación de las necesidades funcionales y técnicas es obligatoria para establecer el comportamiento del sistema. A continuación, se detallan los requisitos que rigen el ecosistema iMirly:

Tabla 11: Requisitos Funcionales (RF)

Nº Requisito	Descripción	Prioridad
RF-1.1	Registrar usuario (cliente, autónomo o empresa).	Alta
RF-1.2	Iniciar y cerrar sesión.	Alta
RF-1.3	Editar y eliminar perfil.	Media
RF-2.1	Publicar servicio con título, descripción y precio.	Alta
RF-2.2	Modificar o eliminar servicio.	Alta
RF-2.3	Buscar y filtrar servicios por categoría o ubicación.	Alta
RF-3.1	Enviar solicitud de contratación.	Alta
RF-3.2	Aceptar o rechazar solicitud.	Media
RF-3.3	Actualizar estado del servicio (pendiente, en curso, completado)	Baja
RF-4.1	Valorar un servicio recibido.	Alta

RF-4.2	Consultar valoraciones de un profesional.	Alta
RF-5.1	Enviar mensajes entre cliente y profesional.	Alta
RF-5.2	Consultar historial de mensajes.	Baja
RF-6.1	Mostrar ubicación aproximada de servicios cercanos.	Alta
RF-7.1	Ver y eliminar usuarios o servicios inactivos.	Media
RF-7.2	Consultar estadísticas básicas del sistema.	Media
RF-8.1	Mostrar aviso al recibir solicitudes o mensajes.	Media
RF-9.1	Acceder a una sección de preguntas frecuentes.	Alta
RF-9.2	Enviar sugerencias o incidencias.	Alta

Tabla 12: Requisitos No Funcionales (RNF)

Nº Requisito	Descripción	Prioridad
RNF-1.1	La aplicación deberá responder a las acciones del usuario en un tiempo razonable (menos de 3 segundos por operación).	Media / Alta
RNF-1.2	El sistema deberá permitir el uso fluido en dispositivos Android de gama media.	Baja
RNF-1.3	La carga de listas o búsquedas no deberá bloquear la interfaz principal.	Alta
RNF-2.1	La interfaz será intuitiva, con iconos y menús fácilmente reconocibles.	Baja
RNF-2.2	El diseño seguirá una estructura consistente en todas las pantallas.	Alta
RNF-2.3	El usuario deberá poder realizar cualquier acción principal en un máximo de tres pasos.	Baja
RNF-2.4	La aplicación incluirá mensajes de error y confirmación claros.	Alta
RNF-3.1	El sistema deberá poder ejecutarse sin errores críticos durante las pruebas de uso.	Alta
RNF-3.2	En caso de fallo, la aplicación deberá mostrar un mensaje informativo sin cerrarse abruptamente.	Baja
RNF-3.3	La base de datos deberá mantener la integridad de los datos tras reinicios o cierres inesperados.	Baja

RNF-4.1	Las contraseñas se almacenarán en formato cifrado (por ejemplo, con hash).	Alta
RNF-4.2	El acceso a las funcionalidades estará restringido según el tipo de usuario.	Baja
RNF-4.3	Los datos personales solo podrán ser modificados por el propio usuario.	Alta
RNF-4.4	No se gestionarán datos bancarios reales; los pagos estarán simulados.	Baja
RNF-5.1	El código se documentará adecuadamente mediante comentarios y convenciones de nombres.	Alta
RNF-5.2	Se empleará control de versiones (GitHub) para facilitar el trabajo colaborativo.	Alta
RNF-5.3	El sistema se estructurará en módulos independientes para facilitar futuras actualizaciones.	Baja
RNF-6.1	La aplicación será compatible con dispositivos Android a partir de la versión 8.0 (Oreo).	Baja
RNF-6.2	La interfaz se adaptará a distintos tamaños de pantalla (móviles y tablets).	Baja
RNF-6.3	El backend y la base de datos estarán diseñados para poder integrarse con versiones web en el futuro.	Baja
RNF-7.1	El proyecto podrá ejecutarse en distintos entornos de desarrollo (Android Studio y VS Code).	Baja
RNF-7.2	La estructura de carpetas y dependencias estará documentada para poder importar fácilmente el proyecto.	Alta
RNF-8.1	La aplicación usará colores contrastados para mejorar la visibilidad.	Media
RNF-8.2	Se procurará que los textos sean legibles con un tamaño mínimo de 14 px.	Baja

D. Lógica de Datos y Navegación (E-R y Flujo)

Para garantizar que la base de datos soporte eficientemente la relación entre clientes y profesionales, se ha diseñado el Diagrama Entidad-Relación (E-R). Este esquema define la persistencia de la información en un entorno PostgreSQL y asegura la integridad de los datos mediante el uso de identificadores únicos (UUID).

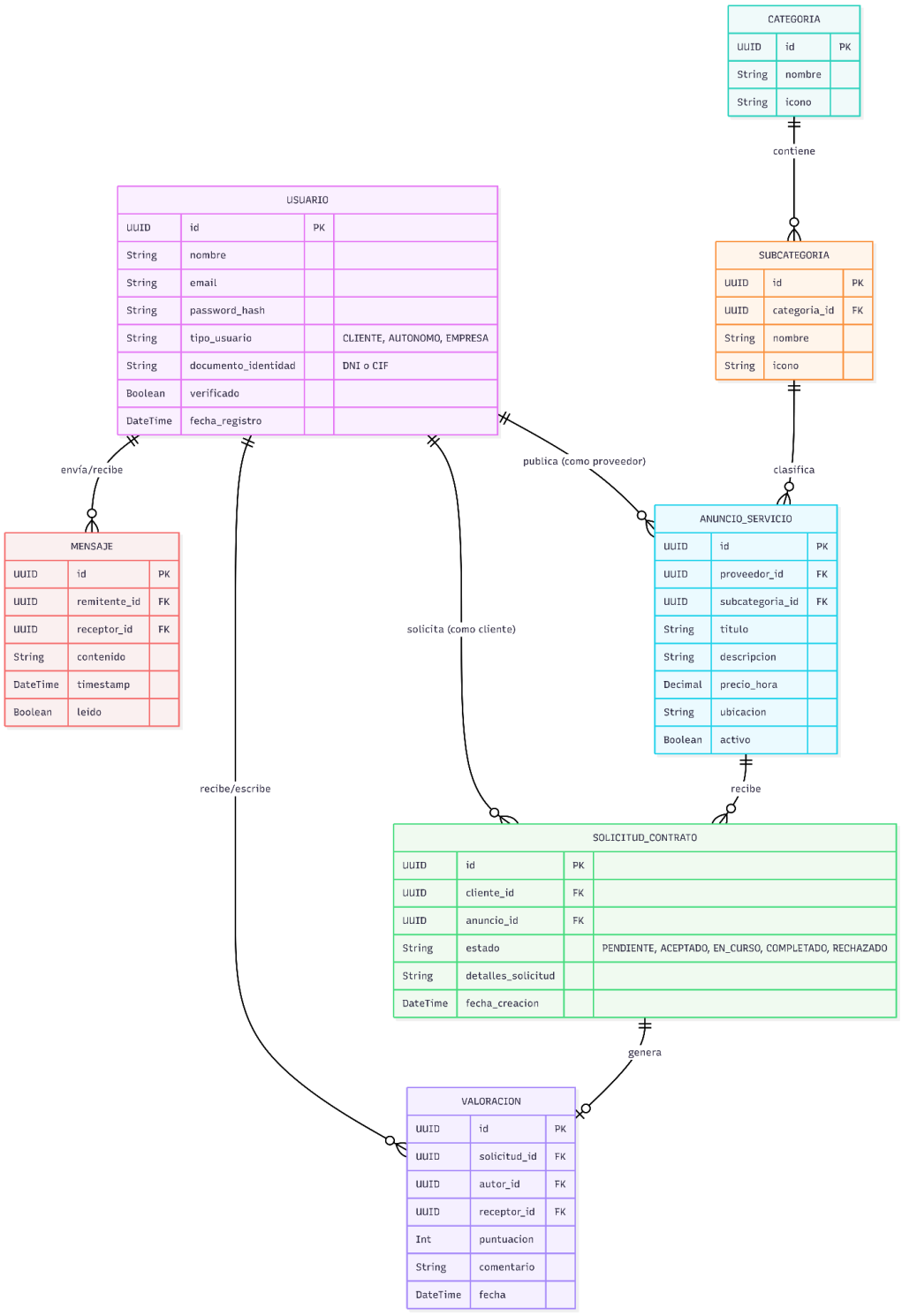

Figura D: Diagrama Entidad-Relación de la base de datos de iMirly.

El diagrama modela el ecosistema completo, destacando la entidad Usuario con sus roles (Cliente, Autónomo y Empresa) y su relación con los Anuncios de Servicio. Se definen tablas

críticas para la trazabilidad, como Solicitud de Contrato (con estados de gestión), Mensajería y el sistema de Valoración cruzada.

Complementariamente, para visualizar la experiencia del usuario y los puntos críticos de decisión, se ha elaborado el Diagrama de Flujo del Servicio. Este mapa marca el recorrido lógico desde el inicio del contacto hasta la finalización efectiva de la transacción.

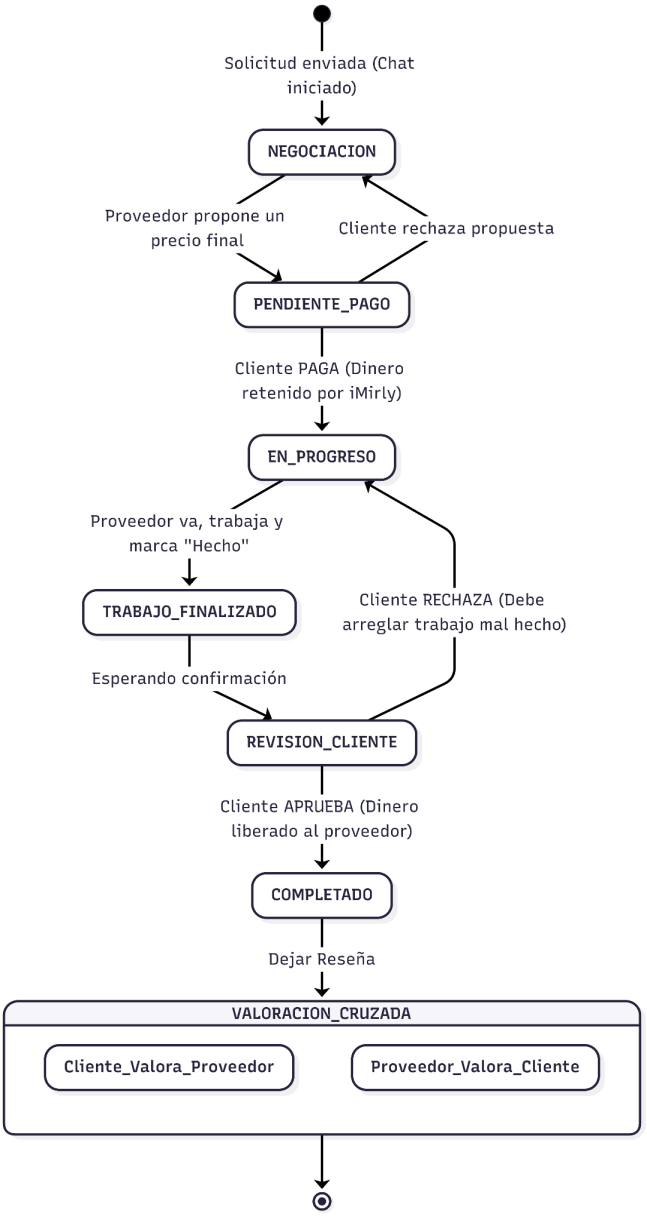

Figura E: Diagrama de procesos y ciclo de vida del contrato.

El flujo detalla la lógica de negocio de iMirly, haciendo hincapié en la seguridad financiera: el dinero es retenido por la plataforma durante la fase de "En Progreso" y solo se libera al

proveedor tras la "Revisión y Aprobación" del cliente. Este proceso garantiza el cumplimiento de la propuesta de valor basada en la confianza y la transparencia.

3.2 Fase 2: Diseño de la aplicación

En esta fase se transforma el análisis de requisitos en una **experiencia visual e interactiva**, definiendo la **Interfaz de Usuario (UI)** y la **Experiencia de Usuario (UX)**. El objetivo es validar la usabilidad y los flujos de navegación antes de iniciar la etapa de codificación para reducir errores de diseño.

3.2.1 Identidad Visual y Guía de Estilo

La identidad de iMirly busca transmitir profesionalidad y transparencia. Para llegar al resultado final, se siguió un proceso de diseño iterativo centrado en la legibilidad y la psicología del color, evaluando múltiples alternativas antes de consolidar la marca definitiva.

A. Evolución y Diseño del Logotipo

El diseño del logotipo no fue un proceso inmediato, sino el resultado de la exploración de diversas variantes tipográficas y simbólicas.

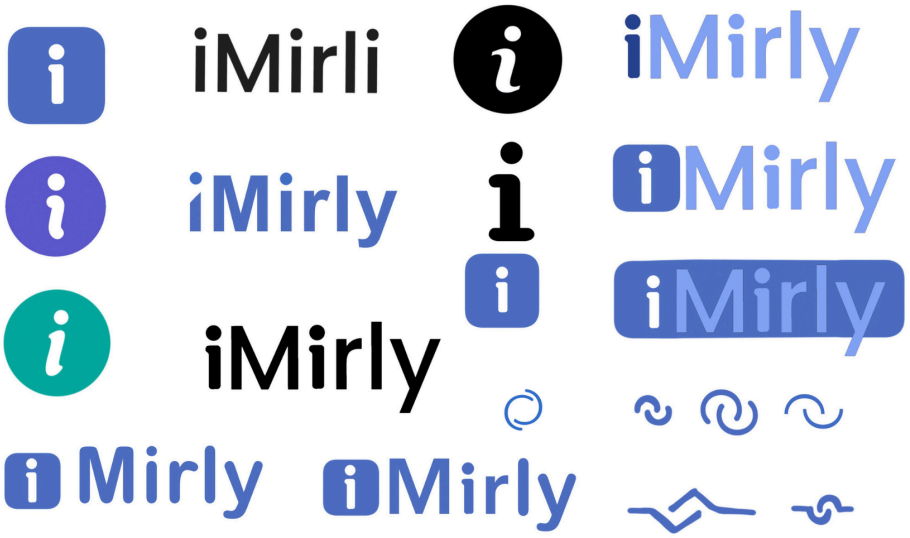

Figura F: Propuestas iniciales para el logotipo de iMirly.

Como se observa en la **Figura F**, se experimentó con diferentes pesos visuales para la tipografía "iMirly", probando desde estilos más minimalistas hasta otros más robustos. Asimismo, se testearon distintas formas de integrar el isotipo (la letra "i"), evaluando su presencia dentro de

contenedores circulares, cuadrados con bordes redondeados y de forma independiente, buscando siempre que el elemento central representara la unión entre la oferta y la demanda profesional.

B. Desarrollo de la Identidad Cromática

La selección de la paleta de colores se dividió en dos fases de filtrado para asegurar que el entorno visual fuera coherente con los valores de confianza de la aplicación.

1. Propuesta General de Colores: En una primera etapa, se exploraron cuatro ramas cromáticas distintas (Azul, Púrpura, Naranja/Dorado y Rojo) para evaluar qué sensaciones transmitía cada una en la interfaz de la aplicación.

Figura G: Primera fase de exploración cromática.

2. Refinamiento de la Gama Preferida: Tras el análisis inicial, se decidió apostar por la gama de los azules y turquesas, por ser los que mejor proyectaban una imagen de seguridad, tecnología y profesionalidad. En esta segunda fase, se refinaron las tonalidades dentro de esa misma familia para encontrar el equilibrio exacto de contraste y accesibilidad.

Figura H: Segunda fase de refinamiento (Rama de azules/turquesas).

C. Paleta de Colores y Tipografía Final

Finalmente, para cumplir con el requisito de accesibilidad RNF-8.1, se seleccionó la paleta definitiva (**Figura I**), que utiliza los tonos púrpura y azul profundo que se visualizan en el prototipo interactivo, garantizando colores contrastados que mejoran la visibilidad.

Figura I: Paleta de colores y guía de estilo

Como recurso adicional para la defensa del proyecto, se ha elaborado una presentación detallada de la identidad de marca:

- **Enlace a la presentación del diseño:** Guía de Estilo iMirly (Canva).

3.2.2 Diseño de Pantallas Principales

El proceso de creación de la interfaz ha seguido una evolución desde lo conceptual hasta lo técnico, asegurando que el usuario pueda realizar cualquier acción principal en un **máximo de tres pasos**, cumpliendo con el requisito **RNF-2.3**.

1. **Esquemas de página en papel:** En la etapa inicial se realizaron bocetos manuales (**Figura J**) para estructurar la jerarquía de información de las pantallas de búsqueda y perfil.

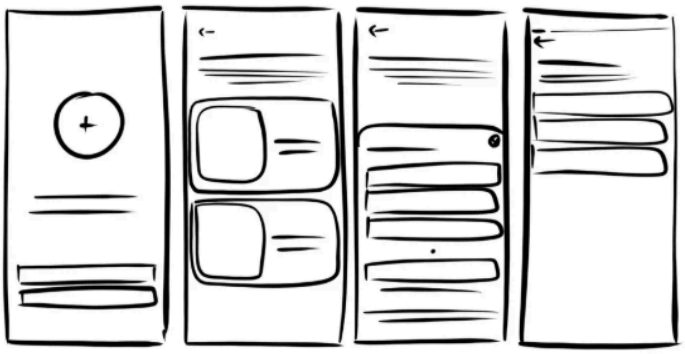

Figura J: Esquemas y bocetos iniciales en papel

2. **Prototipo de baja fidelidad (Wireframes):** Estos esquemas se digitalizaron para definir los flujos de navegación sin distracciones visuales, centrándose en la funcionalidad de contratación y registro de CIF/DNI.

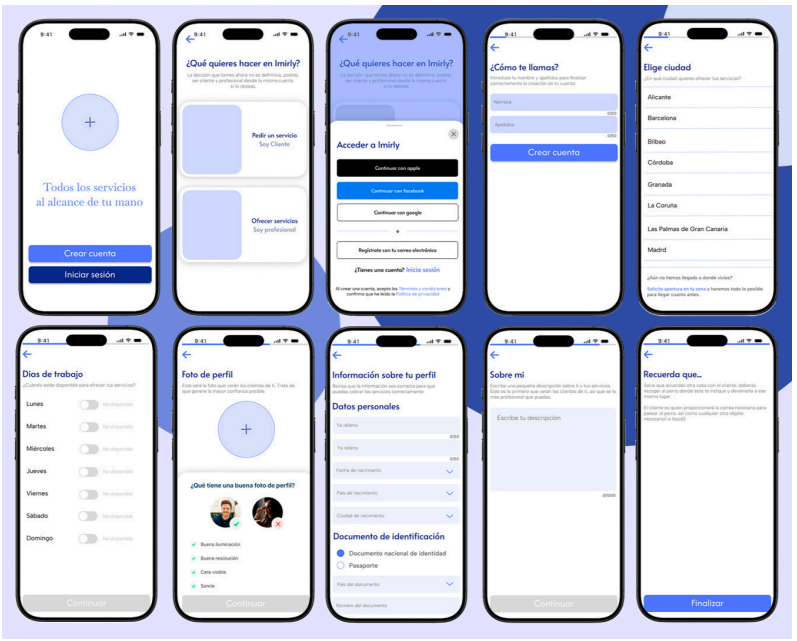

Figura K: Wireframes digitales de iMirly

3.2.3 Prototipo Interactiva de Alta Fidelidad

Finalmente, utilizando la herramienta **Figma**, se ha desarrollado un prototipo interactivo que simula el comportamiento real de la aplicación. Este recurso permite validar el **flujo de navegación completo**, desde el registro de un nuevo profesional hasta la valoración final de un servicio.

A diferencia de los esquemas estáticos, este entorno permite al tribunal experimentar la usabilidad de la plataforma en tiempo real:

• **Acceso al prototipo interactivo:** Prototipo Navegable iMirly (Figma).

3.2.4. Validación de Diseño: Análisis de la Encuesta de Usuario

Con el objetivo de validar la **usabilidad y el impacto visual** de la interfaz antes de la fase de desarrollo, se realizó una consulta a una muestra de usuarios potenciales de diversos rangos de edad. En esta encuesta se presentaron dos alternativas de diseño: la **Opción 1** y la **Opción 2**.

A. Presentación de Alternativas de Diseño

Para el estudio de mercado, se propusieron las siguientes dos líneas gráficas:

Opción 1

Opción 2

Figura L: Capturas de pantalla de la Opción de Diseño 1 y 2

La opción 1, sería una propuesta basada en la simplicidad, con un estilo "clean" y profesional, centrada en la eficiencia operativa. Y la opción 2 una propuesta con un alto componente visual, utilizando ilustraciones descriptivas y colores más vivos para generar un entorno más cercano y moderno.

B. Análisis de Resultados y Preferencias

Tras analizar las respuestas, la **Opción 2 resultó ser la preferida** por la mayoría de los encuestados, basándose en los siguientes criterios extraídos de la encuesta:

1. **Impacto Visual y Cercanía**: Los usuarios destacaron que la Opción 2 es "más visual", "dinámica" y "creativa". Varios encuestados señalaron que este diseño genera una mayor sensación de **confianza y cercanía**, factores críticos para una aplicación de contratación de servicios.
2. **Intuitividad y Accesibilidad**: Un hallazgo relevante fue la percepción de los usuarios de mayor edad (55+ años) y perfiles que buscan rapidez. Estos indicaron que los iconos e ilustraciones de la Opción 2 son "**más claros y diferenciables**", facilitando la identificación de categorías con una "vista rápida" sin necesidad de leer detenidamente los títulos.
3. **Experiencia de Usuario (UX)**: Mientras que la Opción 1 fue valorada positivamente por su "minimalismo" y por ser "menos cargada", la Opción 2 fue descrita como "más amigable para el target" y con "mejor visibilidad", lo que incentiva la interacción.

C. Disposición al Uso de la Aplicación

La encuesta también validó el interés comercial en iMirly. La gran mayoría de los participantes afirmaron que **usarían la aplicación** para anunciarse o buscar servicios, lo que refuerza la viabilidad del proyecto planteada en la sección 1.4.2.

Tabla 13: Resumen de Feedback Cualitativo

Perfil de Usuario	Preferencia	Justificación Clave (Feedback de la Encuesta)
Nativo Digital (18-34 años)	**Opción 2**	Destacan que es un diseño "más dinámico", "creativo" y con "mejor visibilidad", lo que incentiva la interacción.

Senior (55+ años)	**Opción 2**	Valoran que los iconos e ilustraciones son "más claros y diferenciables", facilitando la identificación de categorías de forma rápida.
Público General / Profesional	**Opción 2**	El estilo visual genera una mayor sensación de **"confianza y cercanía"**, factores considerados críticos para una app de contratación de servicios.

D. Conclusión del Diseño Final

Basándonos en estos resultados, se ha decidido adoptar la **Opción 2** como base para el prototipo de alta fidelidad. Sin embargo, atendiendo a las recomendaciones de los usuarios para evitar el exceso de scroll y la saturación visual, se integrarán elementos de la Opción 1 para mantener un entorno **limpio, profesional y fácil de entender**.

3.3. Fase 3: Desarrollo del Sistema

En esta fase se ha llevado a cabo la implementación técnica de los dos grandes componentes que integran el ecosistema iMirly: una aplicación móvil nativa y un servidor API robusto, orquestados para lograr alta escalabilidad y seguridad.

A. Arquitectura General y Comunicaciones

Como se observa en el esquema integral del sistema, la comunicación entre ambos extremos se realiza mediante el protocolo JSON/HTTPS, garantizando la privacidad y la integridad de la información en tránsito.

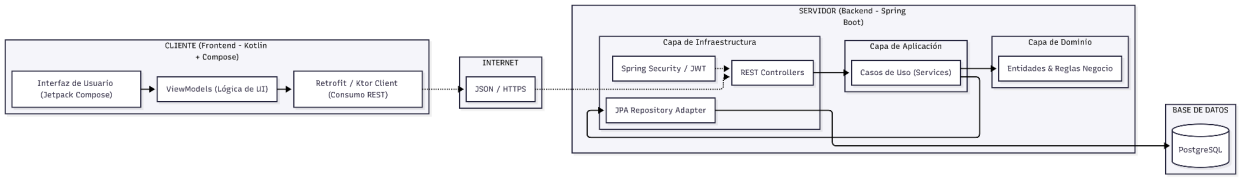

Figura N: Arquitectura integral del sistema iMirly.

El diagrama ilustra el flujo de datos bidireccional. En el extremo del Cliente (Kotlin + Jetpack Compose), la lógica de interfaz se gestiona mediante ViewModels, que realizan peticiones asíncronas a través de Retrofit. En el extremo del Servidor (Spring Boot 3), cada petición es interceptada por un filtro de Spring Security que valida el token JWT antes de permitir el acceso a

los controladores y la lógica de negocio, finalizando con la persistencia en la base de datos PostgreSQL.

B. Arquitectura del Backend y Estructura del Código

Para asegurar la escalabilidad y el mantenimiento a largo plazo, el backend se ha organizado siguiendo una Arquitectura Hexagonal (Clean Architecture). Este patrón permite una separación total entre la lógica de negocio y los detalles de implementación técnica (como la base de datos o frameworks externos) mediante la definición de Puertos y Adaptadores.

La estructura se divide en tres niveles de abstracción:

1. **Capa de Dominio**: Contiene las entidades esenciales (User, ServiceAd) y las reglas de negocio puras, aisladas de cualquier librería de terceros.
2. **Capa de Aplicación**: Implementa los casos de uso concretos que ejecutan las directrices del negocio a través de servicios especializados.
3. **Capa de Infraestructura**: Contiene los adaptadores reales, incluyendo los controladores REST (Spring MVC) para la entrada de datos y los repositorios JPA (PostgreSQL) para la salida y persistencia.

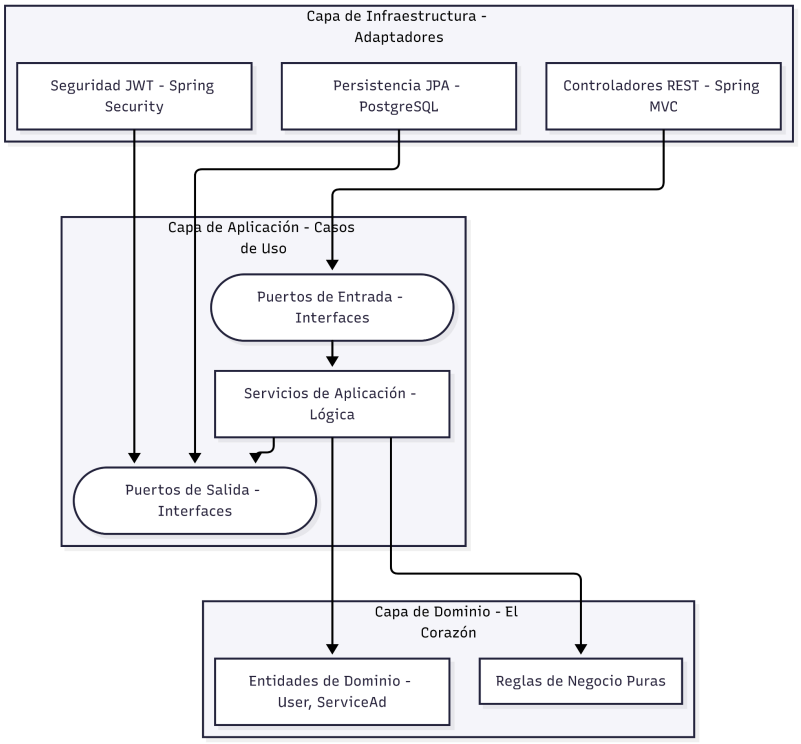

Figura Ñ: Detalle de la Arquitectura Hexagonal del Servidor.

El diagrama ilustra la jerarquía de dependencias del backend, donde la Capa de Infraestructura gestiona la seguridad JWT y la persistencia en PostgreSQL, rodeando el núcleo del sistema compuesto por la lógica de aplicación y el corazón del dominio (entidades y reglas puras).

Para optimizar la gestión del código, el servidor se ha modularizado en cuatro bloques estratégicos que agrupan funcionalidades relacionadas:

- **`auth`**: Encargado de la gestión de perfiles, el registro de usuarios con validación dual (DNI/CIF) y la administración del monedero virtual.
- **`catalog`**: Motor de búsqueda y gestión dinámica de los anuncios de servicios y sus subcategorías.
- **`contracting`**: Módulo crítico que gestiona el ciclo de vida del contrato, desde la negociación hasta el sistema de pagos retenidos.
- **`messages`**: Sistema de chat dinámico que integra el envío de ofertas interactivas y la gestión de valoraciones finales.

C. Desarrollo Frontend (Aplicación Android Nativa)

La interfaz de usuario de iMirly se ha construido de forma 100% declarativa utilizando Kotlin y Jetpack Compose. Esta elección tecnológica garantiza un desarrollo ágil, una UI moderna y una separación clara entre la lógica de presentación y la vista.
Características técnicas del Frontend:

- **Gestión de Estado:** Se ha implementado el patrón de diseño MVVM (Model-View-ViewModel). El uso de Corrutinas y viewModelScope permite gestionar procesos asíncronos sin bloquear la interfaz principal, cumpliendo con el requisito de fluidez RNF-1.3.
- **Consumo de API:** La comunicación con el backend se realiza mediante Retrofit y OkHttp. Se han integrado interceptores que adjuntan automáticamente el token JWT en las cabeceras HTTP de cada petición, garantizando una navegación segura tras el inicio de sesión.
- Funcionalidades Implementadas:
    - **Chat en Tiempo Real:** Comunicación fluida que permite la negociación directa.

- **Sistema Inteligente de Presupuestos:** Gestión de ofertas y acuerdos mediante burbujas dinámicas integradas en el propio flujo del chat.
- **Monedero Virtual (Escrow):** Sistema de pago retenido donde la plataforma custodia el saldo hasta que el servicio es aprobado por ambas partes, asegurando el cobro para el profesional y la calidad para el cliente.

D. Base de Datos y Persistencia

Siguiendo el diseño analítico del Diagrama E-R, se ha implementado la persistencia de datos utilizando PostgreSQL como motor principal.
Detalles de la implementación:

- **Integridad de Datos:** La relación entre las entidades (Usuarios, Anuncios, Contratos y Valoraciones) se gestiona mediante Spring Data JPA e Hibernate. El uso de identificadores UUID asegura la escalabilidad y seguridad de los registros.
- **Estructura Jerárquica:** La base de datos organiza los servicios en Categorías y Subcategorías, permitiendo un filtrado eficiente según los requisitos funcionales RF-2.3.
- **Población Automática (Database Seeder):** Para facilitar el despliegue en entornos de desarrollo y pruebas, se ha programado un DatabaseSeeder. Al inicializar el servidor por primera vez, este inyecta automáticamente categorías reales (como Reformas, Educación, Estética y Salud) y datos de prueba (mockups), permitiendo disponer de un entorno funcional de forma inmediata.

3.4. Fase 4: Pruebas y Control de Calidad

Esta fase constituye el filtro de seguridad crítico previo al lanzamiento comercial, cuyo objetivo es garantizar que el ecosistema iMirly cumpla con los estándares de robustez, fluidez y seguridad definidos en los requisitos no funcionales. El trabajo se divide en cuatro ejes de validación técnica:

A. Pruebas Técnicas: Unitarias y de Integración

Siguiendo los pilares de la metodología de trabajo, se ejecutan tests automatizados en el entorno de Testing (Staging) para validar los módulos de forma aislada y conjunta antes de su integración definitiva en la rama principal de GitHub.

- **Pruebas Unitarias**: Validación de la lógica pura en controladores y servicios del backend desarrollados en Java 21 y Spring Boot 3. Se busca una cobertura ≥ 80% del código lógico, priorizando funciones críticas como el registro de usuarios con validación dual (DNI/CIF) y el motor de búsqueda del módulo `catalog`.
- **Pruebas de Integración**: Verificación de la comunicación fluida entre el cliente Android (Kotlin) y el servidor a través de peticiones JSON/HTTPS. Se testea exhaustivamente el flujo de negocio crítico: la retención de saldo en el monedero virtual (Escrow) gestionado por el módulo contracting y el envío de presupuestos inteligentes en el módulo messages.

B. Pruebas de Usabilidad y Validación Funcional

Se retoman los hallazgos de la fase de diseño para confirmar que la aplicación es intuitiva para los perfiles de usuario identificados (Nativo Digital y Senior).

- **Navegación Ágil**: Se comprueba mediante pruebas manuales que cualquier acción principal, como la búsqueda o contratación, se realice en un máximo de tres pasos, cumpliendo con el requisito RNF-2.3.
- **Accesibilidad Visual**: Basándose en la preferencia por la Opción de Diseño 2, se valida que los iconos e ilustraciones permitan identificar categorías de forma rápida ("vista rápida"), facilitando el uso para usuarios mayores de 55 años.

C. Pruebas de Rendimiento y Estabilidad

Se monitoriza el comportamiento del sistema para asegurar que la experiencia de uso sea instantánea y fiable bajo las métricas de calidad establecidas.

- Tiempo de Respuesta: Optimización de la carga de listas y procesos asíncronos para que el tiempo medio de respuesta sea inferior a 2 segundos, superando el límite de 3 segundos definido inicialmente en el RNF-1.1.
- Tasa de Fallos: El sistema debe operar con una estabilidad que limite los cierres inesperados (crashes) a menos del 1% mensual en condiciones de uso real.

D. Gestión de Incidencias y Evidencia Técnica

Cualquier error detectado durante estas pruebas se registra de forma sistemática en el tablero de Trello, permitiendo una resolución ágil y documentada. Como resultado final de esta fase, se genera el Informe de pruebas y validación, un entregable que recopila los resultados de todos los tests y actúa como evidencia del cumplimiento del requisito RNF-3.1 de estabilidad operativa.

- **Lanzamiento y Mantenimiento (Fase 5):** Incluye el tiempo para la publicación de la aplicación en Google Play y App Store, la implementación de sistemas de *feedback* y el seguimiento de métricas.

4. PROCEDIMIENTOS Y CONTROLES DE CALIDAD

El aseguramiento de la calidad en iMirly se concibe como un **proceso transversal** que abarca todas las fases del ciclo de vida del software, desde la codificación inicial hasta la interacción final del usuario con la plataforma. El objetivo fundamental de este bloque es definir los procedimientos técnicos de seguimiento y control necesarios para garantizar que el producto final cumpla con los estándares de **rendimiento, seguridad y usabilidad** esperados por los clientes y profesionales del sector.

Para ello, se establece un marco de trabajo que combina la verificación preventiva durante el desarrollo, el uso de indicadores métricos de rendimiento (KPIs) y un sistema robusto de gestión de incidencias apoyado en herramientas de colaboración ágil como **Trello**.

4.1 Control de calidad durante el desarrollo

Durante la fase de implementación de iMirly, se aplican diversas medidas preventivas y correctivas para asegurar la integridad del código y la funcionalidad de los módulos independientes. Los pilares de este control son:

- **Revisiones de código (Code Reviews)**: Siguiendo un modelo de trabajo colaborativo, cada módulo desarrollado en el frontend (Kotlin) o backend (Java) será revisado mediante code reviews en GitHub antes de su integración definitiva en la rama principal del proyecto. Este procedimiento garantiza la detección temprana de errores lógicos y asegura el cumplimiento del requisito **RNF-5.1** sobre convenciones de nombres y documentación.
- **Pruebas automatizadas**: Se implementarán tests unitarios y de integración para las funciones críticas del sistema, como la pasarela de pagos simulada, el registro de usuarios con validación DNI/CIF y el motor de búsqueda del módulo `catalog`. Gracias a la **Arquitectura Hexagonal**, estas pruebas permiten verificar la lógica de negocio de forma aislada de la base de datos PostgreSQL.
- **Validación funcional**: Se llevan a cabo comprobaciones manuales para asegurar que cada funcionalidad implementada se ajusta estrictamente a los **Requisitos Funcionales (RF)** definidos, como la publicación de anuncios o la gestión del chat en tiempo real.

- **Segregación de entornos de despliegue**: Para evitar que errores en fase de pruebas impacten en la experiencia del usuario, se utilizan tres entornos independientes:
    1. **Desarrollo:** Para la codificación y pruebas iniciales.
    2. **Testing (Staging):** Para la validación funcional completa y control de calidad antes de la entrega.
    3. **Producción:** Entorno final operativo donde los clientes consumen el servicio.
- **Gestión de incidencias técnica:** Se utilizará la herramienta **Trello** como software centralizado para el registro, evaluación y seguimiento de cualquier error o incidencia detectada durante la ejecución de las tareas. Esto permite una **resolución ágil y documentada** de los problemas técnicos, facilitando la transparencia en el proceso de desarrollo.

4.2. Indicadores de calidad (KPIs)

Para medir de manera objetiva el éxito técnico y la satisfacción del usuario tras el despliegue de iMirly, se han establecido una serie de Indicadores Clave de Desempeño (KPIs). Estas métricas permiten evaluar si el producto cumple con los Requisitos No Funcionales (RNF) definidos en la fase de análisis y garantizan que la experiencia de uso sea fluida, segura y fiable.
A continuación, se detallan los indicadores que regirán la evaluación del sistema:

Tabla 14: Indicadores Clave de Desempeño (KPIs) de iMirly

Indicador	Objetivo (Target)	Relación Técnica
Resolución de incidencias críticas	≥ **95%**	Garantiza que casi la totalidad de los errores que impiden la contratación o el pago sean solventados antes de 48h.
Tiempo medio de respuesta	< **2 segundos**	Mejora el **RNF-1.1** (límite de 3s) para asegurar una navegación instantánea en dispositivos Android.
Satisfacción del usuario	≥ **4/5 estrellas**	Métrica basada en el feedback directo de clientes y profesionales tras completar un servicio.
Cobertura de pruebas automatizadas	≥ **80%**	Asegura que la mayor parte del código lógico haya sido validado mediante tests unitarios en el entorno de desarrollo.
Estabilidad del sistema	< **1% mensual**	Límite máximo de cierres inesperados (crashes) reportados en la versión de producción.

La monitorización constante de estos valores permitirá identificar áreas de mejora y priorizar las actualizaciones en el sistema de gestión de incidencias de **Trello**, asegurando así un ciclo de **mejora continua** para la plataforma.

4.3 Procedimientos de mejora continua y gestión de incidencias

La plataforma iMirly adopta una filosofía de mejora continua basada en ciclos de retroalimentación constantes y una gestión ágil de las desviaciones técnicas. Este enfoque asegura que el producto no solo cumpla con los requisitos iniciales, sino que evolucione según las necesidades del mercado y mantenga los estándares de calidad esperados.

A. Registro y resolución de incidencias (Trello)

La organización operativa se apoya en el uso de la herramienta **Trello**, la cual permite una gestión ágil de los flujos de trabajo mediante tarjetas y tableros. Este sistema funciona como el núcleo del control de calidad y seguimiento técnico, donde cada tarea o error detectado se documenta para permitir un seguimiento en tiempo real desde su detección hasta su validación definitiva.

El tablero de iMirly está organizado en las siguientes columnas, que reflejan nuestra metodología de trabajo colaborativo:

- **Lista de tareas:** Columna central donde se crean, definen y asignan las tareas pendientes a los miembros del equipo, o donde se eligen las nuevas funciones a desarrollar según la prioridad del proyecto.
- **En curso:** Espacio destinado a las tareas que están siendo ejecutadas activamente en ese momento por los desarrolladores.
- **Terminado:** Repositorio de tareas finalizadas. Para garantizar el control de calidad, cada tarjeta en esta columna incluye adjunto el contenido o la evidencia técnica del trabajo realizado antes de su integración definitiva en GitHub.
- **Reuniones:** Espacio dedicado a la gestión de las sesiones grupales que el equipo realiza al inicio de cada semana. En estas tarjetas se registra el seguimiento de los avances logrados y la planificación de los objetivos para la semana siguiente.

Figura O: Captura del tablero Trello de iMirly con el flujo de incidencias

B. Ciclos de revisión y retroalimentación

Para complementar el uso de herramientas digitales, se establecen los siguientes protocolos de actuación:

- **Reuniones semanales de seguimiento:** Sesiones de planificación para analizar el avance del proyecto, evaluar el volumen de incidencias registradas en Trello y priorizar las tareas del siguiente ciclo de desarrollo.
- **Feedback del usuario final:** Una vez que la aplicación esté en fase de consumo por parte de los clientes, se implementarán encuestas de satisfacción y buzones de sugerencias integrados. Este feedback directo será la fuente principal para planificar actualizaciones y nuevas funcionalidades que resuelvan puntos de fricción reales detectados en el mercado.
- **Monitorización post-lanzamiento:** Se establecerá un sistema de seguimiento para detectar errores críticos en el entorno de producción, asegurando que cualquier fallo sea devuelto al ciclo de desarrollo de forma inmediata para su corrección.

4.4. Documentación de control

Como garantía final del proceso de calidad, el proyecto iMirly genera una serie de entregables técnicos que actúan como evidencia del cumplimiento de los estándares establecidos. Estos

documentos no solo sirven para la validación interna del equipo, sino que son instrumentos esenciales para el **mantenimiento futuro y la escalabilidad** del ecosistema.

A continuación, se describen los cuatro documentos de control que se elaboran durante la ejecución del proyecto:

1. **Informe de seguimiento del desarrollo:** Este documento registra el **progreso real frente a la planificación** temporal establecida en el Diagrama de Gantt. Permite identificar desviaciones en las fases de análisis, diseño e implementación, justificando técnicamente cualquier ajuste realizado en el cronograma original de 30 semanas.
2. **Registro de incidencias y resolución:** Vinculado directamente con la gestión operativa en **Trello**, este documento recopila de forma detallada todos los errores detectados durante el desarrollo. Cada entrada incluye la descripción del fallo, el impacto en los Requisitos Funcionales y la solución técnica aplicada. Se pone especial énfasis en la columna "Terminado" del tablero, donde se adjuntan las evidencias del trabajo realizado antes de su cierre definitivo.
3. **Informe de pruebas y validación:** Constituye la certificación técnica de que el software es estable antes de su puesta en producción. Incluye los resultados de las **pruebas unitarias y de integración** realizadas sobre los módulos de backend (Java/Spring Boot) y frontend (Kotlin), asegurando el cumplimiento estricto del requisito **RNF-3.1** de estabilidad operativa.
4. **Manual de control de calidad:** Es el documento normativo donde se estandarizan los procedimientos para futuras actualizaciones. En él se definen las **convenciones de nombres** y la estructura de comentarios en **GitHub** (según el requisito **RNF-5.1**), así como los protocolos de testeo que deben seguirse para asegurar que las nuevas funcionalidades no degraden la calidad del Producto Mínimo Viable (MVP) inicial.

5. DOCUMENTACIÓN

Se han elaborado los siguientes documentos técnicos y de usuario, diseñados para cubrir todo el ciclo de vida del producto:

5.1. Manual de Usuario

Este manual constituye la guía fundamental para que clientes y profesionales operen en la plataforma de forma autónoma. Incluye instrucciones detalladas sobre el proceso de registro con verificación dual (DNI/CIF), la búsqueda de servicios mediante filtros de categoría y ubicación, y la gestión de contratos a través del chat en tiempo real. También describe el funcionamiento del monedero virtual (Escrow) para asegurar que los pagos se realicen con total confianza.

- Acceso al manual:

5.2. Guía de Instalación y Despliegue

Manual técnico que describe los requisitos previos y los pasos necesarios para poner en marcha el sistema en un entorno de desarrollo o producción. Especifica la configuración obligatoria de Java JDK 21, PostgreSQL y Android Studio, así como los pasos para clonar el repositorio de GitHub, configurar el archivo application.properties e inicializar el Database Seeder para inyectar automáticamente los datos de prueba y categorías.

- Acceso al manual: Guía de Instalación y Despliegue

5.3. Manual de Administrador

Documento técnico-operativo destinado al personal encargado de la supervisión de iMirly. Detalla los procedimientos para la gestión y mantenimiento de usuarios y empresas, la moderación del catálogo dinámico de servicios y los protocolos de supervisión del marketplace. Incluye también herramientas para la resolución de incidencias y la monitorización de métricas globales del sistema.

- Acceso al manual:

5.4. Documentación Técnica

Compendio exhaustivo de la ingeniería del proyecto para futuros desarrolladores. Explica detalladamente la arquitectura hexagonal (capas de Dominio, Aplicación e Infraestructura), la estructura de la API REST, el diagrama entidad-relación de la base de datos y la lógica de los cuatro módulos estratégicos: auth, catalog, contracting y messages.

- Acceso al manual: Documentación Técnica

5.5. Tutoriales y Material de Apoyo

Recursos multimedia y guías rápidas diseñados para facilitar la curva de aprendizaje inicial en la primera versión del producto. Incluye vídeos explicativos sobre el flujo completo del servicio (desde la oferta inicial hasta la valoración bidireccional de 1 a 5 estrellas) y esquemas visuales de la interfaz para usuarios finales.

- Acceso al manual:

6. ANEXOS

Este apartado está destinado a incluir todo aquello que queráis añadir de manera adicional al trabajo. Imágenes, glosarios, tablas grandes, etc.

7. REFERENCIAS BIBLIOGRÁFICAS

Andalucía Emprende. (s. f.). *Guía de ayudas y subvenciones para emprender*.

https://www.andaluciaemprende.es/guia-ayudas

Diputación de Granada. (s. f.-a). *Ayudas a jóvenes emprendedores*.

https://www.dipgra.es/ayudas-jovenes-emprendedores

Diputación de Granada. (s. f.-b). *Subvenciones*. https://www.dipgra.es/subvenciones

e-Autónomos. (s. f.). *Los costes de vender tu app*.

https://e-autonomos.es/los-costes-de-vender-tu-app

Junta de Andalucía. (s. f.-a). *Proyectos colaborativos de formación profesional*.

https://www.juntadeandalucia.es/proyectos-colaborativos-fp

Junta de Andalucía. (s. f.-b). *Innovactiva 2025*. https://www.juntadeandalucia.es/innovactiva2025

Ministerio de Educación, Formación Profesional y Deportes. (s. f.). *Proyectos de innovación de formación profesional*. https://www.educacion.gob.es/proyectos-innovacion-fp

Oficina de Transferencia de Resultados de Investigación (OTRI). (s. f.). *Portal de la OTRI*. Universidad de Granada. https://otri.ugr.es

webel. (2023, octubre). *Términos y condiciones webel* (Documento 679070515). Scribd.

https://es.scribd.com/document/679070515/Terminos-y-condiciones-webel

Sandoval, G., & Martín, J. (2025). *Guía proyecto integrado*. C.D.P. - Escuela Técnica Audiovisual de Andalucía.