# DÍGITO

## Módulo Operativo para Registro Inteligente

De 1-3 semanas de demora en registro → **Registro integrado al flujo diario**

El objetivo fue claro: convertir el registro de horas en parte natural del flujo de trabajo, no en una tarea adicional.

# Mi rol

## Research y síntesis

* Conduje 1 de 3 entrevistas cualitativas (UX Designer de Dígito)
* Estructuré vacíos de investigación y prioricé áreas de *desk research*
* Sintetizé *insights* de entrevistas en tabla comparativa + generé HMW questions
* Creé *customer journey* para validar patrones de comportamiento

## Estrategia y arquitectura

* Propuse dirección estratégica: concentrarnos en el **módulo operativo del consultor** (vs. rediseñar toda la plataforma)
* Definí 3 ejes de diseño: **Autonomía** (asistente proactivo), **Visibilidad** (Kanban↔Gantt), **Velocidad** (FAB + registro rápido)
* Prioricé *features* basándome en viabilidad técnica + impacto en adopción

## Diseño y prototipado

* Diseñé *wireframe* completo usado como base para HiFi
* Diseñé múltiples pantallas: Dashboard, Registro de horas, Calendario, Vista de Proyectos
* Armé flujos de interacción principales del prototipo
* Resolví problemas técnicos de componentes (*dropdowns* con *overlay*, *states* del *sidebar*)

## Testing y validación

* Monté test en Useberry con 4 tareas + 3 preguntas cualitativas
* Contacté usuarios para *testing*
* Iteré prototipo basándome en *feedback*

## Decisiones clave que propuse:

* Descartar **gamificación superficial** (basándome en *research* sobre "fatiga del juego")
* Priorizar **FAB siempre visible** sobre navegación profunda
* Proponer sincronización **Kanban↔Gantt en tiempo real**
* Agregar **tooltips fijos** en lugar de *wizard* intrusivo

# Contexto

Este proyecto se desarrolló como parte del curso Product Design Fundamentals de Repensar, en equipo de 4 personas durante 3.5 meses, culminando con la presentación ante el CEO de Dígito.

Dígito es una empresa de Business Intelligence y RPA. Su plataforma SaaS B2B incluía facturación, reportes y administración, pero el módulo operativo para consultores tenía **baja adopción y uso inconsistente**.

## El problema: las horas se registraban 1-3 semanas tarde

### Impacto en el negocio

Esa demora afectaba:

* Facturación
* Seguimiento operativo
* Evaluación de carga y capacidad
* Detección temprana de desvíos

### Hipótesis inicial vs. realidad

* Hipótesis del CEO: "Los usuarios se olvidan de cargar las horas."
* Pregunta crítica sin responder: **¿El registro realmente se olvida o se evita activamente?**
* Alcance definido: No modificar contratos, procesos organizacionales ni estructuras internas. Identificar qué aspectos de la experiencia generaban fricción y cómo reducirla desde diseño.  
    
  [ SCREENSHOT DEL MVP ACTUAL --- VISTA FRAGMENTADA ]

# Research: el problema real no es olvido, es depriorización activa

Combiné *desk research*, entrevistas con usuarios reales y auditoría del MVP para entender por qué el módulo operativo tenía baja adopción.

## Métodos

### Desk Research

* Validé patrones de comportamiento en literatura sobre productividad
* Respaldo teórico: "Los usuarios tienden a evitar procesos que demandan esfuerzo cognitivo extra" (Kahneman, *Thinking Fast and Slow*)
* Benchmark del mercado: 60% de empleados perciben el *time tracking* como vigilancia (Paymo, Hubstaff)

### 3 Entrevistas cualitativas

* Perfiles: PM, Desarrollador, UX Designer (todos de Dígito)
* Objetivo: observar patrones de comportamiento, no recopilar opiniones aisladas

### Auditoría heurística del MVP

* Identifiqué flujos fragmentados, interfaz centrada en administración y carga cognitiva permanente  
    
  [ CUSTOMER JOURNEY --- FLUJO ACTUAL DEL USUARIO ]

## Citas que revelaron el problema real

### Percepción del sistema:

"En una escala de 1 a 10, lo veo más como un 1: control administrativo."

### Postergación activa:

"Si estoy en varios proyectos al mismo tiempo y surge una urgencia, el registro queda para después." "Yo lo hago al final del día, a las seis de la tarde, donde registro todo junto."

### Fricción física:

"Si cierro la *laptop* en la oficina, es engorroso volver a encenderla solo para registrar."

### Dependencias operativas:

"Si el CEO no revisa ese día, la aprobación queda en pausa. A veces podemos avanzar, pero otras veces tenemos que esperar."

## Insight emergente

**El registro no se hace porque no aporta valor al flujo real de trabajo.**

El sistema está construido desde la lógica del dato, no desde la lógica de la acción. Los consultores priorizan lo que mueve la operación y descartan lo que no devuelve utilidad inmediata.

### Pivot del diseño:

* De: "¿Cómo logramos que la gente recuerde cargar horas?"
* A: "¿Cómo logramos que registrar sea un resultado natural del trabajo, no una tarea adicional?"  
    
  [ TABLA COMPARATIVA --- DESK RESEARCH + ENTREVISTAS + AUDITORÍA ]

# Estrategia: módulo operativo, no rediseño total

Tras analizar los *insights*, quedó claro que mejorar el registro no implicaba rediseñar toda la plataforma, sino construir un módulo operativo cohesivo alineado con el flujo real de los consultores.

## Alternativa evaluada desde negocio: penalizar el registro tardío

Además de las direcciones de UX, surgió desde negocio: forzar el registro mediante sanciones.

### Camino descartado por:

* Genera cumplimiento, no adopción
* Degrada la calidad del dato (incentiva registros genéricos)
* Castiga al usuario por una fricción no resuelta
* Contradice una cultura basada en autonomía

Conclusión: La vía sostenible es lograr que registrar sea **rápido, útil y parte del ritmo de trabajo.**

## Alcance: concentrarnos en el módulo operativo del consultor

El *brief* original del CEO incluía 8 áreas de mejora. Rediseñar toda la plataforma era inviable en 3.5 meses y estratégicamente ineficiente: nuestra investigación mostró que la fricción crítica estaba concentrada en el módulo operativo, no en los módulos administrativos.

### Nuestra decisión:

Concentrarnos en donde registra horas, gestiona tareas y ve su carga de trabajo.

Esto permitió:

* Atacar directamente el problema detectado (registro tardío de 1-3 semanas)
* Proponer una solución completa y validable en el plazo académico
* Mantener compatibilidad con módulos administrativos que ya funcionaban
* Sentar bases para futuras integraciones (Reportes, Tickets, *mobile*)

### Por qué no fue solo una limitación de ***scope***:

El *research* validó que los consultores no se quejaban de facturación ni reportes gerenciales. Se quejaban del flujo diario: registro tardío, herramientas paralelas (Freedcamp, Teams, Excel), falta de visibilidad del equipo.

## Principios que guiaron las decisiones

* **Relevancia:** responder a la causa raíz, no solo el síntoma
* **Impacto en el usuario:** disminuir fricción, aumentar continuidad del flujo
* **Diferenciación:** ofrecer experiencia más coherente que herramientas fragmentadas
* **Viabilidad técnica:** soluciones implementables dentro del ecosistema actual
* **Escalabilidad:** bases sólidas para incorporar automatización o IA

El principio rector: **Acompañar el ciclo completo del trabajo, no simplemente documentarlo.**

# Insights y decisiones de diseño: 5 fricciones críticas

Tras analizar entrevistas, *benchmark* y auditoría del MVP, identifiqué cinco patrones consistentes que explicaban por qué el registro fallaba.

## 1. El usuario piensa en tareas, no en horas

* **Dolor:** Los consultores estructuran su día en base a **qué tienen que hacer**, no cuánto tiempo dedicarán.
* **Decisión:** Asistente proactivo + autocompletado desde historial
* **Por qué funciona:** El sistema sugiere actividades basadas en calendario y patrones previos. El usuario confirma en lugar de reconstruir memoria.
* **Se tradujo en UI como:** FAB permanente con sugerencias contextuales, opción "Duplicar día anterior" y autocompletado basado en historial de tareas recurrentes.

## 2. Falta de visibilidad del equipo

* **Dolor:** Los usuarios usaban Freedcamp y Teams porque el sistema no mostraba "qué está haciendo el equipo" ni "qué sigue después de esta tarea".
* **Decisión:** Panel unificado con Kanban (flujo diario) + Gantt (dependencias y planificación macro) + Vista General (actividad del equipo)
* **Por qué funciona:**
  + Kanban = "qué hacer hoy" con estados visibles
  + Gantt = "qué viene después" con jerarquías claras
  + Vista General = contexto del proyecto + actividad reciente sin salir del sistema
* **Se tradujo en UI como:** Integración **Kanban↔Gantt↔Calendario** en un mismo componente con *tabs* persistentes. Todas las vistas consumen la misma fuente de datos: mover una *card* en Kanban actualiza automáticamente el Gantt y el *dashboard* personal en tiempo real.

## 3. Plataforma percibida como control

* **Dolor:** "Para mí es un 1. Es solo control administrativo, no me organiza nada."
* **Decisión:** Estados visibles + progreso personal + señales de avance
* **Por qué funciona:** El sistema pasa de "registro para facturación" a "herramienta que muestra mi progreso". El usuario ve valor inmediato: qué completó, qué avanzó, qué falta.
* **Se tradujo en UI como:** Dashboard con indicadores de progreso del día, estados de completitud por proyecto y *feedback* ligero al registrar ("Día completo registrado", sin trofeos ni *streaks* infantiles).

## 4. Tareas repetitivas

* **Dolor:** "Yo lo hago al final del día, a las seis de la tarde, donde registro todo junto."
* **Decisión:** "Duplicar día anterior" + sugerencias desde calendario
* **Por qué funciona:** Los usuarios tienen rutinas predecibles. Prefieren confirmar patrones en lugar de cargar todo manual. Validado en entrevistas (mencionaron usar Excel y copiar filas anteriores).
* **Se tradujo en UI como:** Botón destacado "Duplicar día anterior" en el FAB y sugerencias automáticas extraídas de eventos del calendario sincronizado (Google/Microsoft).

## 5. Trabajo en equipo dependiente de terceros

* **Dolor:** "Si el CEO no revisa ese día, la aprobación queda en pausa. A veces podemos avanzar, pero otras veces tenemos que esperar."
* **Decisión:** Transparencia en estados + permisos claros + notas públicas
* **Por qué funciona:** Los usuarios ven el estado sin *pings* innecesarios. Las notas públicas funcionan como *knowledge base* ligero y evitan preguntas repetidas.
* Se tradujo en UI como: Tooltips explicativos sobre permisos, iconografía diferenciada para tareas propias vs. del equipo, y sección de notas públicas por proyecto visible para todo el equipo.  
    
  [ CARDS INTERACTIVAS --- INSIGHT → DECISIÓN → UI ]

# Arquitectura, flujos y UI: tres ejes de diseño

La solución se tradujo en flujos y componentes organizados en tres ejes: estructural, funcional y motivacional.

[ DIAGRAMA DE ARQUITECTURA --- TRES EJES ]

## Panel Operativo Unificado (Eje estructural)

* **Problema detectado:** Fragmentación y navegación confusa entre vistas. Algunos roles dependían del CEO para confirmar estados por falta de visibilidad del trabajo del equipo.
* **Solución:** Integrar todas las vistas operativas en un único espacio coherente.

### Componentes:

* **Kanban:** ejecución rápida, estados visibles, foco en "qué sigue"
* **Vista General:** *brief*, información del proyecto y actividad reciente del equipo (cambios en Kanban, notas públicas, acceso al perfil de cada miembro)
* **Gantt:** dependencias, jerarquías claras, planificación macro y visualización del equipo

### Sincronización Kanban ↔ Gantt:

Todas las vistas consumen la misma fuente de datos. Mover una card en Kanban actualiza automáticamente el Gantt general, el dashboard personal y la visibilidad del equipo en tiempo real.

[ MOCKUP --- PANEL UNIFICADO CON TABS KANBAN/GANTT/CALENDARIO ]

## Automatización Ligera (Eje funcional)

* **Problema detectado:** Al final del día, los usuarios están saturados y registrar exige reconstruir memoria, cambiar de contexto y retomar la *laptop*.
* **Solución:** Un asistente "ligero" que sugiere actividades basadas en agenda, historial y patrones de trabajo, transformando el registro en una confirmación rápida en lugar de una tarea de memoria.

### Funcionalidades principales:

* "Duplicar día anterior"
* Sugerencias desde calendario (Google/Microsoft)
* Autocompletado basado en historial
* Stopwatch  
    
  [ MOCKUP --- FAB CON SUGERENCIAS CONTEXTUALES ]

## Capa Social y Transparencia (Eje motivacional)

* **Problema detectado:** El *brief* sugería gamificación mediante puntajes y recompensas. El *research* mostró que sistemas de puntos generan fatiga del juego cuando no están conectados con motivación intrínseca ni valor real.
* **Solución:** Mantener elementos sociales valiosos, pero redirigidos a productividad:
  + Reconocimiento público ligado a consistencia, no trofeos
  + Notas públicas para fomentar conocimiento compartido
  + Perfiles visibles que facilitan colaboración
  + Señales de avance que refuerzan autonomía, no vigilancia

### Resultado:

Una capa motivacional adulta, centrada en progreso, transparencia operativa y colaboración, sin caer en fatiga de gamificación superficial.

[ MOCKUP --- DASHBOARD CON INDICADORES DE PROGRESO ]

## Dashboard Refinado (Componente que articula los tres ejes)

El *dashboard* funciona como puerta de entrada al módulo operativo. Su diseño se centró en dar contexto inmediato sin sobrecargar al usuario.

### Incluye:

* Tareas del día
* Reuniones diarias
* Últimos proyectos con modificaciones propias
* Widget de notas pendientes  
    
  [ MOCKUP --- DASHBOARD COMPLETO ]

# Testing: validación con 4 usuarios reales

Realicé pruebas exploratorias de usabilidad con prototipo HiFi en Useberry mediante tareas guiadas.

* **Participantes:** 4 usuarios reales de Dígito (Desarrollador, Diseñador UX, Líder de Proyecto, Analista PowerBI)
* **Método:** Observación del comportamiento + *feedback* cualitativo
* **Objetivo:** Identificar fricciones críticas en navegación, comprensión de vistas y continuidad entre planificación-ejecución.

## Fricciones detectadas

* **Dificultad para relacionar Gantt ↔ Kanban:** Los usuarios entendían cada vista aislada, pero no cómo se conectaban. El cambio entre modos no preservaba suficiente contexto.
* **Falta de claridad en roles, permisos y visibilidad:** Los participantes no tenían claro qué podían ver o modificar según su rol. Esto reforzaba la percepción de "control" detectada en investigación.

## Iteraciones aplicadas

### Reorganización profunda del Dashboard

* **Ajustes:**
  + Bifurcación de "Proyectos" en Proyectos Recientes (últimos modificados) y Tareas Recientes (obtenidas del Kanban)
  + Logros movidos al Perfil
  + Calendario + notas reubicados en columna lateral más estrecha
* Resultado: El dashboard dejó de percibirse "plano" y pasó a funcionar como punto de orientación diario.  
    
  [ ANTES/DESPUÉS --- DASHBOARD ITERADO ]

### Clarificación de visibilidad y señales contextuales

* **Ajustes:**
  + Iconografía para distinguir tareas propias y del equipo
  + *Tooltips* en el Gantt que aclaran que su progreso se calcula a partir del estado de las tareas en Kanban
  + *Tooltips* en el Kanban que indican que ese avance es visible para todo el equipo
* Aprendizaje: Algunas funciones (notas públicas, sincronización Gantt-Kanban) requieren educación inicial, no solo señalización. De ahí surge la idea de sumar posteriormente un Wizard de onboarding orientado a los flujos más complejos.  
    
  [ ANTES/DESPUÉS --- TOOLTIPS Y SEÑALES CONTEXTUALES ]

# Resultados, aprendizajes y próximos pasos

## Validación cualitativa

* **Antes del rediseño:** "Para mí es un 1. Es solo control administrativo, no me organiza nada."
* **Después del prototipo:** "Lo veo sencillo de manejar, así que sí me ayudaría bastante."

### Impacto real:

El sistema pasó de ser un registrador de horas a funcionar como una **herramienta operativa.**

## Aprendizajes críticos

### Qué hubiera hecho distinto

La funcionalidad de sugerencias inteligentes se diseñó con posibilidad de incorporar IA, pero quedó fuera del alcance porque no teníamos validación directa de gerencia.

En la presentación final, el CEO mostró interés fuerte por soluciones basadas en IA, interés que no había sido evidente durante el proceso.

**Aprendizaje:** Validar la ambición tecnológica con dirección, incluso cuando el acceso es limitado, puede desbloquear oportunidades que no surgen únicamente desde la investigación con usuarios.

De haber tenido esa confirmación temprana, habría planteado desde el inicio un **enfoque híbrido**: IA ligera como complemento (no como dependencia), con reglas tradicionales como base.

### Cómo evolucionaría hacia IA (Fase 2)

El módulo quedó diseñado para escalar sin reescritura. Si el negocio decidiera invertir, la evolución hacia IA sería directa y controlada.

### Componentes necesarios:

* API de Azure OpenAI
* *Dataset* existente: historial de horas, calendario sincronizado, tipos de tareas y patrones de uso
* Lógica híbrida: reglas simples como *fallback*
* IA como capa de contextualización, no como motor principal

### Beneficios esperados:

* Sugerencias predictivas ("Probablemente trabajaste en X de 11-13 h")
* Recomendación automática de carga o replanificación
* Precisión creciente a medida que la IA aprende patrones reales
* Datos más coherentes para facturación y seguimiento

Esta funcionalidad quedó planteada como **Fase 2**, condicionada a validar adopción del sistema base.

## Próximos pasos estratégicos

### Onboarding: pieza crítica para acelerar adopción

El sistema incorpora conceptos potentes (registro contextual, automatización ligera, sincronización entre vistas), pero requieren comprensión inicial.

Diseñé un *onboarding* guiado basado en cuatro flujos clave:

1. *Dashboard* como punto de partida
2. Relación Kanban ↔ Gantt
3. Perfil y capa social
4. Registro sin fricción

El *wizard* aparece solo en la primera sesión y luego queda accesible desde un punto discreto de ayuda.

### Impacto esperado:

Menos soporte, incorporación más rápida, adopción más consistente.

### Mobile-first: extensión natural del sistema

El patrón fue claro en la investigación: cuando la *laptop* se cierra, el registro se posterga. El mercado refuerza esta dirección: más del 60% del *time-tracking* ligero ocurre en *mobile*.

La decisión de priorizar *desktop* en el MVP fue deliberada: primero había que resolver la raíz del problema (flujo, coherencia entre vistas y valor percibido) antes de expandir a otros dispositivos.

Con esa base ya resuelta, *mobile* se convierte en la evolución lógica para reducir retroactividad y sostener adopción en contextos fuera de escritorio.

### Principios que guían la Fase 2

* **Adopción ocurre cuando hay valor inmediato:** El usuario registra cuando el sistema devuelve algo útil en tiempo real. Cualquier *feature* futuro debe aumentar utilidad, no suma de pasos.
* **El ecosistema importa más que cada vista:** Kanban, Calendario y Gantt funcionan cuando actúan como un único modelo mental. Mantener esa coherencia es clave para escalar.
* **Automatización ligera es la base para IA:** El diseño actual ya opera con reglas que permiten evolucionar hacia sugerencias predictivas sin reescribir el sistema. La transición a IA es una ampliación, no un reemplazo.

# Cierre

Este proyecto no buscó mejorar una tarea: buscó redefinir el rol del sistema dentro del día laboral. El resultado es un módulo operativo coherente que reduce esfuerzo, anticipa fricción y acompaña el trabajo real.

No diseñé pantallas. Diseñé un comportamiento nuevo.

[ CTA --- VER OTROS PROYECTOS ]