

# Plan de Proyecto – **TacticSphere**

## Consultora de Transformación Digital

Integrantes del equipo:

-Vicente Fernandez Delgado

-Samanta Navarro Peña

-Amaro Veas Quezada

Guia:

-Patricio Soto Serdio



## Contenido

Plan de Proyecto – <b>TacticSphere</b> .....	1
Consultora de Transformación Digital .....	1
1. Problema .....	4
Las principales problemáticas detectadas son: .....	4
2. Solución .....	4
El funcionamiento general será el siguiente: .....	5
3. Beneficiados .....	5
4. Alcance .....	6
4.1 Dentro del Alcance (Sí se incluye) .....	6
4.2 Fuera del Alcance (No se incluye) .....	7
5. Metodología .....	7
5.1. Fases del Proyecto .....	7
6. Requerimientos .....	8
Requerimientos Funcionales (RF) .....	8
Requerimientos No Funcionales (RNF) .....	11
7. DAS – Diagramas de Análisis y Diseño del Sistema .....	12
7.1. Diagrama de Clases .....	12
7.2. Diagrama Entidad–Relación (Base de Datos) .....	14
7.2.1. ERD Unificado (OLTP + Reporting) – Vista Híbrida .....	15
7.3. Diagrama de Casos de Uso .....	17
8. Tecnología .....	19
9. Cronograma del Proyecto .....	20
9.1. Fases del Proyecto .....	20
9.2 Carta Gantt .....	21
10. Costos del Proyecto .....	21
10.1. Horas de trabajo del equipo .....	21
10.2. Infraestructura tecnológica .....	22
10.3. Costos Totales Estimados .....	23
10.4. Justificación de costos .....	23



11. Riesgos del Proyecto .....	24
11.1. Riesgos Técnicos .....	24
<b>R3. Riesgos de usabilidad en la interfaz de usuario</b> .....	24
Impacto: Si la interfaz del sistema resulta poco intuitiva, los usuarios (especialmente empleados no técnicos) podrían tener dificultades para completar encuestas, reduciendo la tasa de respuesta.....	24
Mitigación: Realizar pruebas tempranas de usabilidad con un grupo reducido de usuarios de prueba, aplicar mejoras en la navegación y mantener un diseño simple con mensajes de ayuda claros. ....	24
11.2. Riesgos de Gestión .....	24
11.3. Riesgos Externos .....	25
12. Pruebas del Proyecto .....	25
12.1. Alcance y objetivos.....	25
12.2. Tipos de prueba.....	26
12.3. Entornos de prueba .....	27
12.4. Datos de prueba (semilla).....	27
12.5. Matriz de pruebas (muestra) .....	27
12.6. Criterios de aceptación (UAT).....	28
12.7. Criterios de entrada / salida .....	28
12.8. Gestión de defectos .....	28



## 1. Problema

En la actualidad, gran parte de las empresas —ya sean pequeñas, medianas o grandes— enfrentan el desafío de mantenerse competitivas en un entorno donde la transformación digital se ha vuelto un requisito indispensable. Sin embargo, muchas organizaciones presentan dificultades para adaptarse a las nuevas tecnologías, modernizar sus procesos internos o identificar las áreas en las que deben enfocar sus esfuerzos de innovación.

### Las principales problemáticas detectadas son:

#### *Falta de diagnóstico objetivo:*

La mayoría de las empresas no cuentan con herramientas que les permitan evaluar su nivel de madurez digital de manera estructurada y medible.

#### *Resistencia al cambio y desconocimiento:*

Los directivos y colaboradores suelen desconocer cuáles son las áreas críticas de la empresa que requieren priorizarse para avanzar en digitalización, lo que provoca decisiones basadas en intuición y no en datos.

#### *Ausencia de planes personalizados de modernización:*

Las soluciones que existen en el mercado suelen ser genéricas, sin adaptarse a la realidad específica de cada organización, ya sea por tamaño, rubro o recursos disponibles.

#### *Impacto en la competitividad:*

Al no contar con un análisis detallado, las empresas corren el riesgo de quedar rezagadas frente a su competencia, perder eficiencia operativa, limitar su innovación y disminuir su capacidad de crecimiento.

En este contexto, se hace evidente la necesidad de contar con una herramienta que permita evaluar el estado actual de digitalización de una empresa y sus diferentes áreas (departamentos, equipos, incluso a nivel de empleados) de forma rápida, clara y accesible, para así entregar un diagnóstico preciso y recomendaciones de mejora que apoyen la toma de decisiones estratégicas.

## 2. Solución

La solución propuesta es el desarrollo de TacticSphere, una consultora digital apoyada en un sistema de evaluación estructurado que permitirá a las empresas diagnosticar su nivel de madurez digital y recibir recomendaciones concretas de mejora.

El sistema funcionará en base a pilares de análisis, los cuales representarán las principales dimensiones de la transformación digital (ejemplo: Tecnología, Procesos, Estrategia, Talento,



Innovación, Seguridad). Cada pilar estará compuesto por un conjunto de preguntas y encuestas que se aplicarán tanto a nivel organizacional como en áreas específicas (departamentos, equipos de trabajo o incluso empleados individuales).

## El funcionamiento general será el siguiente:

### *Levantamiento de información:*

La empresa ingresa sus datos en el sistema mediante encuestas asociadas a los pilares de madurez digital.

### *Procesamiento y análisis de respuestas:*

El sistema analiza los resultados de manera automatizada, aplicando reglas de negocio y un sistema de puntuación que mide la fortaleza de cada pilar.

### *Generación de resultados:*

- Reporte global de madurez digital de la empresa.
- Análisis segmentado por área, departamento o nivel de empleado.
- Identificación de puntos fuertes y áreas críticas que requieren mejoras.

### *Recomendaciones personalizadas:*

El sistema entregará un informe con sugerencias prácticas de modernización, priorizadas según el nivel de importancia y factibilidad.

De esta manera, TacticSphere busca convertirse en un aliado estratégico que no solo entrega un diagnóstico, sino que también propone planes de acción adaptados a la realidad de cada empresa, facilitando el proceso de transformación digital y reduciendo la incertidumbre en la toma de decisiones.

## 3. Beneficiados

La implementación de TacticSphere está orientada a beneficiar a distintos actores dentro del ecosistema empresarial:

### *Empresas pequeñas y medianas (PyMEs)*

Necesidad: suelen tener recursos limitados y desconocimiento sobre cómo iniciar un proceso de transformación digital.

Beneficio: acceso a un diagnóstico rápido y accesible que les permite visualizar su estado actual y recibir recomendaciones adaptadas a sus capacidades y presupuesto.

### *Empresas grandes y corporaciones*

Necesidad: requieren evaluaciones más detalladas que permitan identificar brechas tecnológicas o de procesos en diferentes áreas de la organización.



Beneficio: obtendrán reportes segmentados por departamentos, equipos o empleados, lo que facilita la toma de decisiones estratégicas a nivel directivo.

#### *Consultores internos o externos*

Necesidad: muchas veces los equipos de consultoría carecen de herramientas estandarizadas para evaluar la madurez digital.

Beneficio: podrán utilizar TacticSphere como soporte metodológico para aplicar diagnósticos más rápidos y confiables, optimizando sus servicios y generando valor adicional a sus clientes.

#### *Colaboradores y empleados*

Necesidad: suelen no tener claridad de cómo su desempeño o área contribuye al avance digital de la organización.

Beneficio: al responder las encuestas, reciben retroalimentación que les permite identificar oportunidades de mejora y alinearse mejor con los objetivos de transformación digital de la empresa.

## 4. Alcance

El proyecto TacticSphere tendrá un alcance definido que permite desarrollar un software web (MVP – Producto Mínimo Viable) durante el periodo académico establecido. Este sistema tendrá como propósito diagnosticar la madurez digital de las empresas mediante encuestas estructuradas y reportes, asegurando que al final del periodo se cuente con un prototipo funcional que cumpla con los objetivos planteados.

### 4.1 Dentro del Alcance (Sí se incluye)

- Desarrollo de un sistema web accesible desde navegador.
- Diseño de pilares de evaluación de madurez digital y creación de encuestas asociadas.
- Gestión de usuarios y empresas (registro de empresas, departamentos, áreas y empleados).
- Módulo de aplicación de encuestas con almacenamiento de respuestas en la base de datos.
- Procesamiento automático de resultados mediante reglas de negocio y sistema de puntuación.
- Generación de reportes básicos y dashboards que muestren:
  - Estado general de la empresa.
  - Resultados por áreas/departamentos.
  - Comparación de fortalezas y debilidades.
- Exportación de resultados en formatos PDF y/o Excel.
- Documentación técnica y manual de usuario del sistema.



## 4.2 Fuera del Alcance (No se incluye)

- Consultoría presencial directa en empresas reales.
- Implementación completa de los planes de modernización sugeridos.
- Integración con sistemas empresariales externos (ERP, CRM, etc.).
- Aplicación de modelos avanzados de Machine Learning o Deep Learning (considerados para fases futuras).
- Escalabilidad del sistema a entornos de producción en la nube (se limitará a un entorno de pruebas controlado para la demo académica).

## 5. Metodología

El proyecto TacticSphere se desarrollará bajo la Metodología de Desarrollo en Cascada Tradicional, la cual plantea un avance secuencial a través de fases claramente definidas. Cada etapa debe completarse antes de iniciar la siguiente, lo que permite mantener un orden lógico, trazabilidad de entregables y una mejor gestión del tiempo dentro del periodo académico.

### 5.1. Fases del Proyecto

1. Inicio y planificación
  - Revisión de objetivos, definición del alcance y preparación del Plan de Proyecto.
  - Elaboración del Acta de Kick-Off.
2. Análisis
  - Levantamiento de requerimientos funcionales y no funcionales.
  - Modelado de procesos y definición de casos de uso.
  - Desarrollo del documento ERS.
3. Diseño
  - Diseño de la base de datos (modelo entidad-relación).
  - Elaboración de diagramas UML (clases, casos de uso).
  - Definición de la arquitectura técnica del sistema.
4. Implementación
  - Desarrollo del backend (API con FastAPI) y configuración de la base de datos MySQL.
  - Desarrollo del frontend en Angular.
  - Construcción de dashboards y módulos de reportes.
5. Pruebas e integración
  - Pruebas unitarias, de integración y de sistema.
  - Validación de flujos completos (registro → encuesta → reporte).
  - Corrección de errores y ajustes de usabilidad.
6. Documentación y entrega
  - Elaboración de manual técnico y manual de usuario.
  - Preparación de la demo final.
  - Entrega oficial el 5 de diciembre de 2025.



## 6.Requerimientos

### Requerimientos Funcionales (RF)

#### **RF-01 – Autenticación con credenciales**

El sistema deberá permitir iniciar sesión con correo y contraseña, validando credenciales contra la base de datos.

#### **RF-02 – Recuperación de contraseña por token**

El sistema deberá enviar (o generar en pantalla, si no hay correo en MVP) un token temporal para restablecer contraseña y exigir su cambio al primer ingreso.

#### **RF-03 – Gestión de roles**

El sistema deberá soportar al menos tres roles: Administrador del sistema, Administrador de empresa y Analista/Consultor, con permisos diferenciados.

#### **RF-04 – Multi-empresa (aislamiento lógico)**

El sistema deberá permitir registrar múltiples empresas y aislar sus datos (usuarios, encuestas, respuestas y reportes) por empresa.

#### **RF-05 – Alta/Baja/Modificación de empresas**

El sistema deberá permitir crear, actualizar datos básicos (razón social, rubro, país) y desactivar empresas (soft delete).

#### **RF-06 – Gestión de estructura organizacional\***

El sistema deberá permitir registrar y editar áreas/departamentos/equipos y asignar empleados a dichas unidades.

#### **RF-07 – Alta de empleados\*\***

El sistema deberá permitir crear empleados con campos obligatorios: nombre, correo (opcional en MVP), cargo, unidad organizacional.

#### **RF-08 – Definición de pilares**

El sistema deberá permitir crear y editar pilares de evaluación (ej.: Tecnología, Procesos, Estrategia, Talento, Seguridad), con descripción y peso.

#### **RF-09 – Banco de preguntas por pilar**

El sistema deberá permitir crear preguntas asociadas a un pilar con: enunciado, tipo de respuesta (Likert 1–5, sí/no, opción múltiple), obligatoriedad y peso.





**RF-10 – Versionado de cuestionarios\***

El sistema deberá permitir crear “cuestionarios” (colecciones de preguntas) con número de versión y estado (borrador/publicado/archivado).

**RF-11 – Asignación de encuestas**

El sistema deberá permitir asignar un cuestionario a: empresa completa, área/departamento o lista de empleados específicos, con fecha de inicio y cierre.

**RF-12 – Notificaciones in-app de asignación\***

El sistema deberá mostrar al usuario autenticado una notificación de encuestas pendientes al ingresar al sistema.

**RF-13 – Respuesta de encuestas**

El sistema deberá permitir responder encuestas desde la web, guardando progreso parcial (borrador) y envío final (estado: completada).

**RF-14 – Validaciones de respuesta\***

El sistema deberá validar obligatoriedad, tipos y rangos (p. ej., Likert 1–5), bloqueando el envío si hay inconsistencias.

**RF-15 – Cálculo de puntajes**

El sistema deberá calcular automáticamente puntajes por pregunta, pilar y global aplicando pesos definidos (pregunta y pilar).

**RF-16 – Umbrales y semáforos**

El sistema deberá clasificar resultados por pilar en categorías (Rojo/Amarillo/Verde) según umbrales configurables por empresa.

**RF-17 – Reporte global por empresa**

El sistema deberá generar un reporte consolidado por empresa con: puntaje global, puntajes por pilar, ranking de pilares y top brechas.

**RF-18 – Reporte segmentado**

El sistema deberá generar reportes por área/departamento y por empleado (si corresponde), con comparativas respecto al promedio de empresa.

**RF-19 – Dashboard ejecutivo**

El sistema deberá mostrar un dashboard con tarjetas/resúmenes (KPI global, top 3 fortalezas y top 3 brechas) y gráficos (barras/radar).



**RF-20 – Exportación de reportes**

El sistema deberá exportar reportes a PDF y Excel, preservando tablas y gráficos principales.

**RF-21 – Registro de actividad (auditoría)**

El sistema deberá registrar acciones clave (creación/edición de empresas, cuestionarios, asignaciones, envíos de encuestas, cambios de umbrales).

**RF-22 – Búsqueda y filtrado**

El sistema deberá permitir buscar y filtrar empresas, empleados, encuestas y reportes por texto, estado, pilar, fecha y unidad organizacional.

**RF-23 – Gestión de catálogos\***

El sistema deberá permitir administrar catálogos auxiliares (rubro, tamaños de empresa, países) usados en formularios y reportes.

**RF-24 – Gestión de sesiones y cierre\***

El sistema deberá invalidar sesiones tras cierre de sesión o inactividad configurable (ej. 30 minutos), requiriendo reautenticación.

**RF-25 – Reapertura controlada**

El sistema deberá permitir al Administrador de empresa reabrir una encuesta cerrada, registrando motivo en bitácora.

**RF-26 – Duplicación de cuestionarios**

El sistema deberá permitir clonar cuestionarios existentes (nueva versión o copia) para agilizar iteraciones.

**RF-27 – Importación básica de empleados\***

El sistema deberá permitir cargar empleados mediante archivo CSV con formato validado y reporte de errores.

**RF-28 – Control de acceso por rol a datos**

El sistema deberá impedir que usuarios de una empresa vean datos de otra, y restringir vistas según rol (ej. Analista no puede cambiar umbrales).

**RF-29 – Histórico de resultados**

El sistema deberá mantener histórico de mediciones por periodo (ej. por mes/trimestre) para comparar evolución.



### **RF-30 – Soporte para anonimato opcional\***

El sistema deberá permitir definir encuestas anónimas (respuestas no vinculadas a identidad), manteniendo solo agregados.

### **RF-31 – Estado de cumplimiento de encuestas**

El sistema deberá mostrar indicadores de avance (% respondidas) por asignación y enviar recordatorios in-app antes del cierre.

### **RF-32 – Plantilla de recomendaciones**

El sistema deberá generar un listado de recomendaciones predefinidas por pilar según tramos de puntaje (sin ML), editable por el Administrador de empresa.

## **Requerimientos No Funcionales (RNF)**

### **RNF-01 – Seguridad de contraseñas**

Las contraseñas deberán almacenarse con hash seguro (p. ej., bcrypt/Argon2) y política de longitud mínima ( $\geq 10$  caracteres).

### **RNF-02 – Control de accesos**

Se deberá aplicar control de acceso basado en roles (RBAC) en backend, validando cada endpoint.

### **RNF-03 – Protección de datos**

Toda comunicación deberá usar HTTPS en entornos de demo/producción; en desarrollo, TLS opcional documentado.

### **RNF-04 – Rendimiento (respuesta UI)**

El tiempo de respuesta percibido en vistas principales (dashboard, listado encuestas) deberá ser  $\leq 2s$  para 95% de solicitudes con 10k respuestas.

### **RNF-05 – Rendimiento (cálculo de puntajes)**

El cálculo de puntajes y generación de reporte global por empresa ( $\leq 5k$  respuestas) deberá completarse en  $\leq 10s$ .

### **RNF-06 – Disponibilidad (demo)**

La disponibilidad objetivo del entorno de demo durante el periodo de evaluación deberá ser  $\geq 95\%$  mensual.

### **RNF-07 – Respaldo y recuperación**

Se deberán realizar respaldos automáticos de base de datos al menos 1 vez al día, con retención mínima de 7 días y prueba de restauración documentada.



#### **RNF-08 – Trazabilidad**

Todas las operaciones críticas deberán registrar: usuario, fecha/hora (UTC), entidad afectada y valores clave antes/después cuando aplique.

#### **RNF-09 – Usabilidad**

La interfaz deberá cumplir con navegación consistente (menú lateral + breadcrumbs), formularios con validación en línea y mensajes de error claros.

#### **RNF-10 – Internacionalización básica**

La UI deberá estar en español por defecto y permitir parametrización de textos (archivos o i18n) para futura traducción.

#### **RNF-11 – Escalabilidad horizontal inicial**

La arquitectura deberá separar frontend, backend y base de datos, permitiendo desplegar servicios por separado (contenedores).

#### **RNF-12 – Mantenibilidad**

El código deberá seguir convenciones (linters), pruebas unitarias mínimas ( $\geq 60\%$  de cobertura en servicios críticos) y documentación de endpoints (OpenAPI/Swagger).

## **7. DAS – Diagramas de Análisis y Diseño del Sistema**

Con el objetivo de representar gráficamente la arquitectura y funcionamiento del sistema TacticSphere, se elaboraron tres diagramas principales: Diagrama de Clases, Diagrama Entidad–Relación (Base de Datos) y Diagrama de Casos de Uso. Estos permiten visualizar la estructura de datos, las interacciones del usuario y la organización del sistema a nivel conceptual.

### **7.1. Diagrama de Clases**

Este diagrama muestra cómo el software estructura su lógica en objetos (clases), sus atributos y métodos. Representa el diseño de software (no la base de datos). Aquí se definen las entidades del dominio (Empresa, Usuario, Pilar, Pregunta, Cuestionario, Asignación, Respuesta, etc.) y los servicios de aplicación que implementan los casos de uso (por ejemplo, creación y publicación de cuestionarios, asignación de encuestas, cálculo de puntajes, generación de reportes y dashboards).

#### **Cómo leerlo.**

Las clases definen datos y comportamiento: por ejemplo, Cuestionario.publicar(), Asignacion.reabrir(), UmbralPilar.clasificar().

Las asociaciones representan relaciones del negocio: una Empresa tiene Usuarios, Unidades y Cuestionarios; un Cuestionario contiene Preguntas (vía CuestionarioPregunta) y genera Asignaciones; una Asignación registra Respuestas.



Los servicios (<<service>>) encapsulan la lógica de casos de uso (AuthService, CuestionarioService, ScoringService, ReporteService, etc.) y orquestan a las entidades.

## Decisiones de diseño clave.

Separación clara entre modelo de dominio (entidades con reglas) y servicios de aplicación (coordinan flujos).

Soporte explícito para multiempresa y alcance polimórfico de las asignaciones (empresa/unidad/empleado).

Métodos de dominio que facilitan invarianza y validaciones (por ejemplo, `Pregunta.validarValor()`, `Asignacion.estaVigente()`).

**Relación con requisitos.**

RF-01 a RF-32: autenticación/roles, gestión de empresas/usuarios/unidades, definición de pilares/preguntas, versionado y asignación de cuestionarios, validaciones, scoring, reportes, exportación, auditoría.

RNF-01 a RNF-12: seguridad, mantenibilidad (servicios cohesionados), rendimiento (scoring/ETL dedicados), trazabilidad (Auditoria).

**Qué no muestra.**

No es un esquema de tablas. La persistencia se detalla en el ERD de la base de datos (sección 7.2).



## 7.2. Diagrama Entidad–Relación (Base de Datos)

Este diagrama representa la base de datos transaccional (OLTP) en tercera forma normal ( $\approx 3FN$ ). Es la fuente de verdad donde se almacenan empresas, usuarios, unidades, empleados, pilares, preguntas, cuestionarios, asignaciones y respuestas, además de umbrales, recomendaciones, reportes (metadatos) y auditoría.

### Cómo leerlo.

Tablas principales: empresas, usuarios, unidades\_organizacionales, empleados, pilares, preguntas, cuestionarios, asignaciones, respuestas.

Tablas de relación/soporte: cuestionario\_pregunta (N:M entre cuestionarios y preguntas), umbrales\_pilar, recomendaciones, auditoria, catálogos (rubros, países).

Claves foráneas (FK) aseguran integridad referencial; por ejemplo, preguntas.pilar\_id  $\rightarrow$  pilares.id.

### Decisiones de diseño clave.

Normalización para evitar redundancia y facilitar consistencia (preguntas separadas de cuestionarios, N:M resuelta con cuestionario\_pregunta).

Multiempresa: todas las entidades operativas referencian a empresas.

Alcance polimórfico de asignaciones: se parametriza con alcance\_tipo y alcance\_id (empresa/unidad/empleador).

Anonimato opcional: respuestas.empleado\_id puede ser NULL.

### Relación con requisitos.

Soporta el ciclo completo: definición de instrumentos  $\rightarrow$  asignación  $\rightarrow$  respuesta  $\rightarrow$  cálculo y reporte.

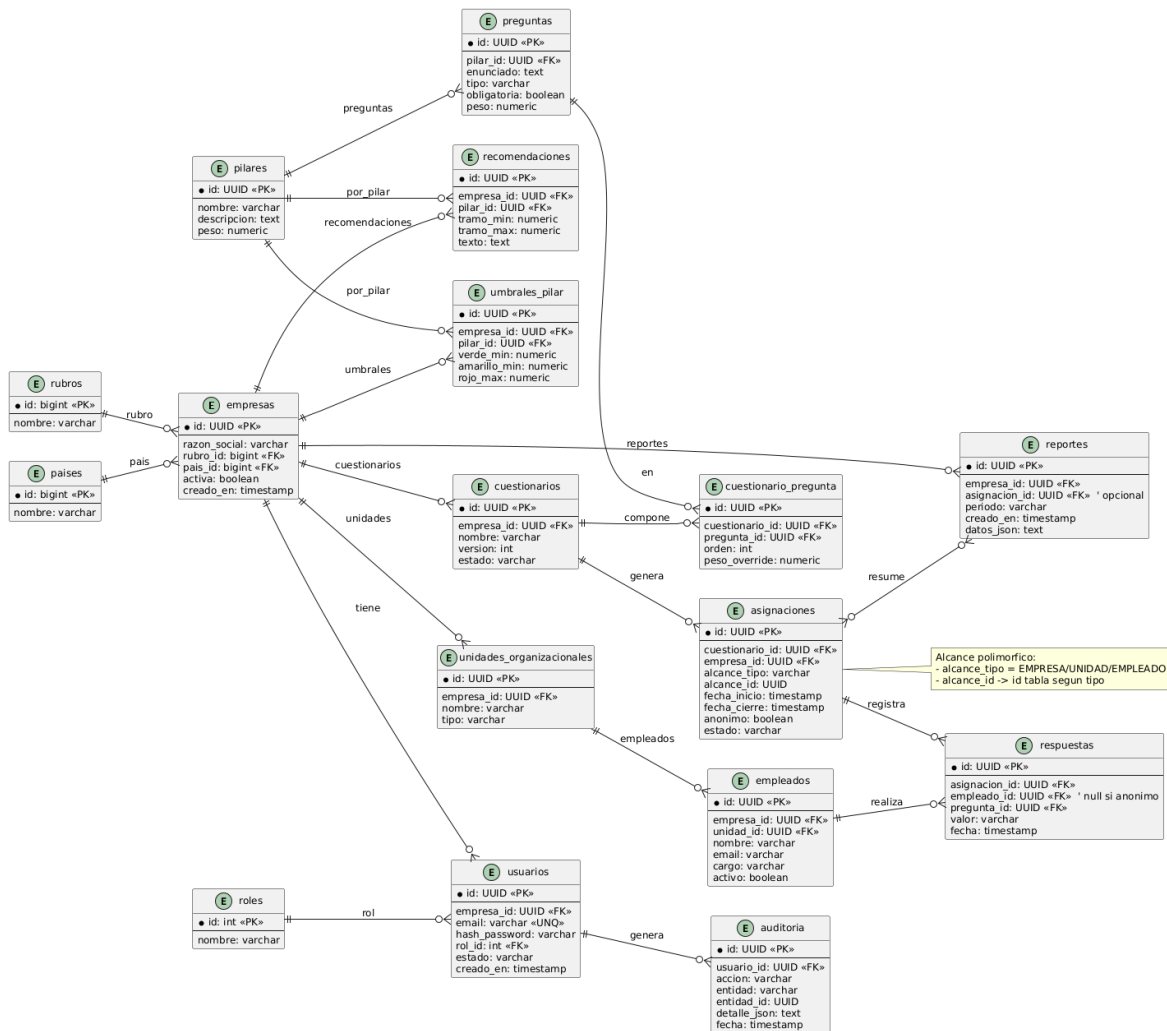
Facilita auditoría (RNF-08) y respaldo/recuperación (RNF-07).

Base adecuada para implementar validaciones (RF-14), históricos (RF-29) y aislamiento de datos por empresa (RF-04).

### Qué no muestra.

No refleja optimizaciones de consulta para dashboards; esas se contemplan en la vista híbrida (7.2.1) como capa derivada opcional.





### 7.2.1. ERD Unificado (OLTP + Reporting) – Vista Híbrida

Presenta, en una sola figura, la base OLTP normalizada y una capa de reporting (esquema estrella liviano con dimensiones y hechos) destinada a acelerar dashboards y consultas agregadas. Las flechas punteadas indican el flujo de datos (ETL/refresh) desde las tablas OLTP hacia las tablas de reporting.

#### Cómo leerlo.

OLTP (normalizado): paquete izquierdo con las tablas operacionales (empresa, cuestionarios, asignaciones, respuestas, etc.).

Reporting (dim/fact): paquete derecho con dimensiones (dim\_empresa, dim\_unidad, dim\_empleado, dim\_pilar, dim\_pregunta, dim\_tiempo) y hechos (fact\_respuesta, fact\_puntaje\_pilar, fact\_puntaje\_global).

Flujo ETL: se extraen datos del OLTP, se transforman (p. ej. mapeo de Likert a números, aplicación de pesos/umbrales) y se cargan en las tablas fact\_\*



## Por qué esta capa (opcional).

Los dashboards suelen requerir agregaciones frecuentes por periodo, empresa, pilar, área o pregunta. Hacer esto directamente sobre el OLTP con múltiples JOIN puede ser costoso.

La capa de reporting guarda pre-agregados indexados, mejorando rendimiento sin alterar la base oficial.

Se puede refrescar al cerrar una asignación o mediante jobs periódicos.

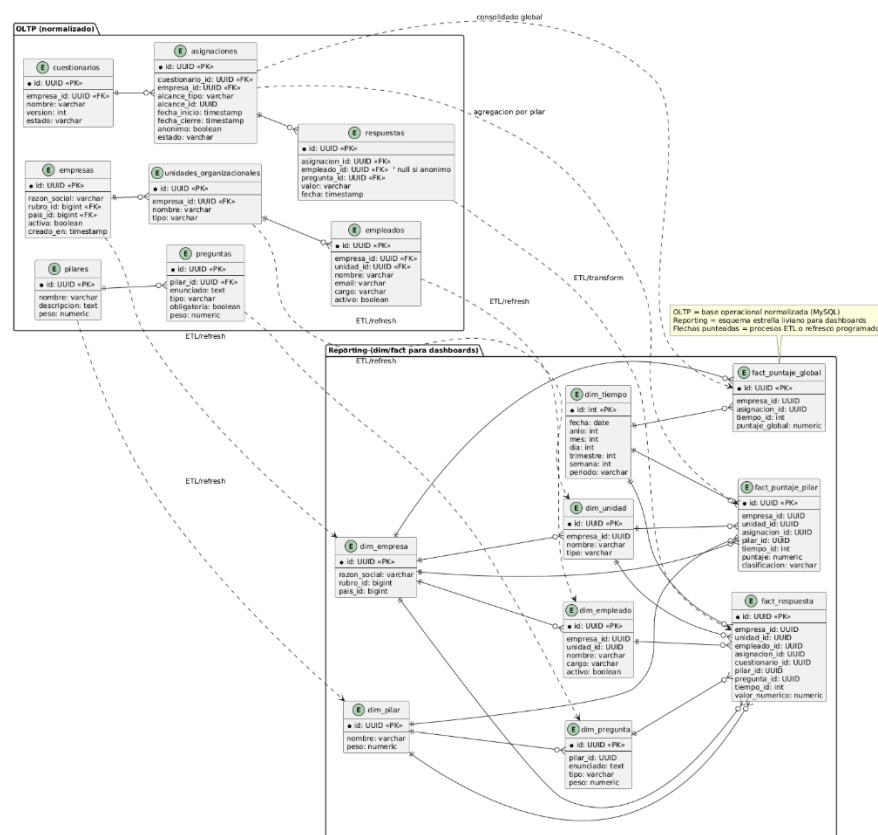
## Relación con requisitos.

Acelera RF-17/18/19/20 (reportes y dashboard), ayuda a cumplir RNF-04/05 (rendimiento).

Mantiene aislamiento multiempresa y soporte a históricos (RF-29), con dim\_tiempo y fact\_\*..

## Qué no es.

No sustituye al OLTP ni crea “otra base independiente”. Es una vista analítica derivada, útil para desempeño del front y reporting.

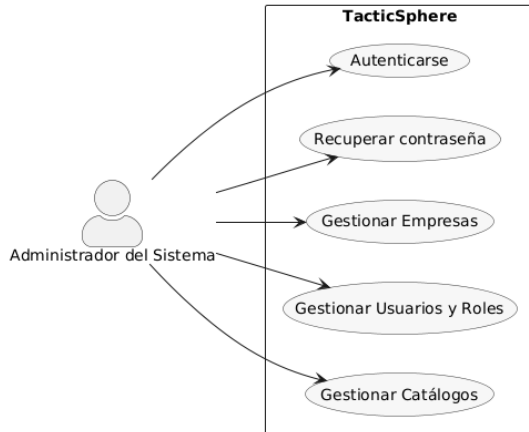




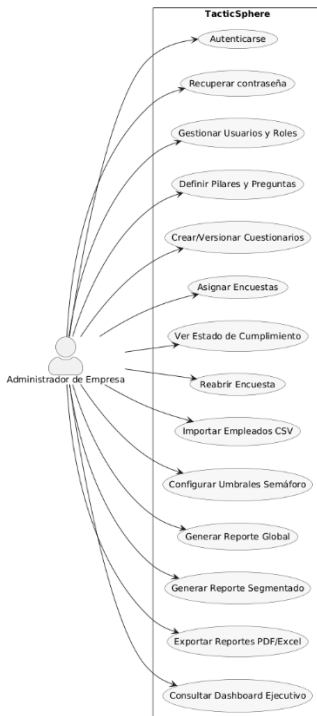
### 7.3. Diagrama de Casos de Uso

El Diagrama de Casos de Uso representa las interacciones entre los diferentes actores del sistema y sus funcionalidades principales.

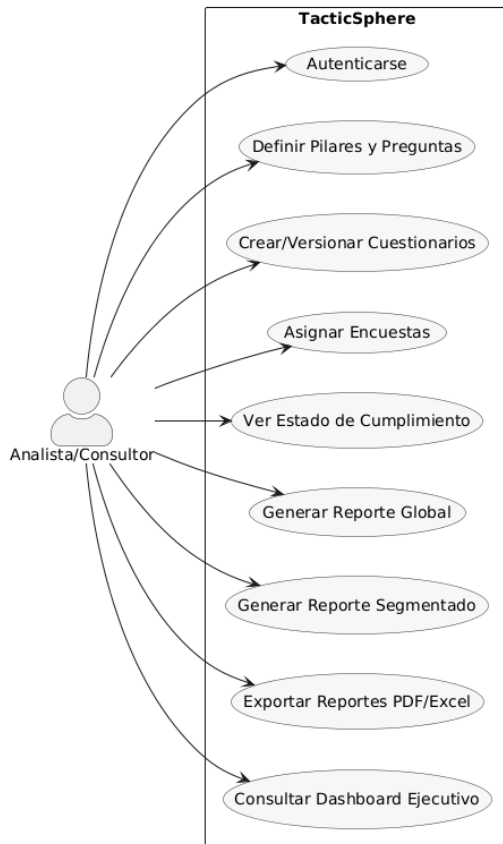
**Administrador del Sistema:** gestiona empresas, usuarios y catálogos generales.



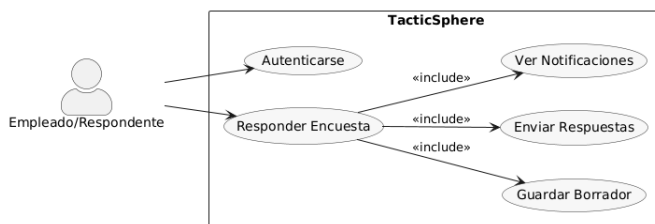
**Administrador de Empresa:** configura pilares, preguntas, cuestionarios, asignaciones, umbrales y genera reportes.



**Analista/Consultor:** accede a la visualización de reportes y dashboards para análisis estratégico.

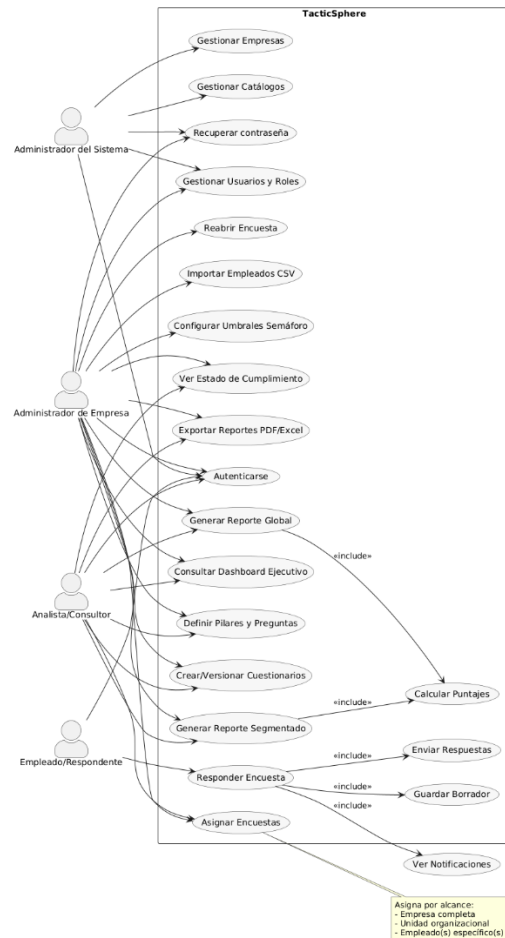


**Empleado/Respondente:** interactúa con el sistema respondiendo encuestas asignadas.



Los casos de uso principales incluyen: gestión de encuestas, cálculo de puntajes, generación de reportes y exportación de resultados.

Este diagrama permite validar que el sistema cubra las necesidades de todos los actores involucrados y que los requerimientos funcionales estén debidamente contemplados.



## 8. Tecnología

Para el desarrollo de TacticSphere se seleccionaron tecnologías gratuitas, con amplia comunidad de soporte y librerías, lo que facilita tanto la implementación como la futura escalabilidad del sistema.

### Backend (lógica del sistema y API)

Python con FastAPI: lenguaje sencillo y muy utilizado en la industria, con soporte para crear APIs rápidas y seguras que conecten el sistema con la base de datos y el frontend.



### *Base de Datos*

MySQL: motor de base de datos gratuito, robusto y ampliamente utilizado, que permitirá almacenar de manera segura la información de empresas, usuarios, encuestas y resultados.

### *Frontend (interfaz de usuario)*

Angular + TypeScript: tecnologías gratuitas que permiten construir páginas modernas, atractivas y fáciles de usar. Se prioriza la estética visual para que la experiencia sea clara e intuitiva.

### *Dashboards y visualizaciones*

Se utilizarán librerías de gráficos gratuitas que permitirán mostrar resultados en tablas, gráficos y paneles ejecutivos de manera clara.

### *Despliegue y control de versiones*

GitHub para control de versiones y trabajo colaborativo.

Docker para empaquetar la aplicación y facilitar su instalación en distintos entorno.

## 9. Cronograma del Proyecto

El cronograma de TacticSphere se desarrollará en 16 semanas bajo la metodología en cascada tradicional, avanzando fase por fase de forma secuencial.

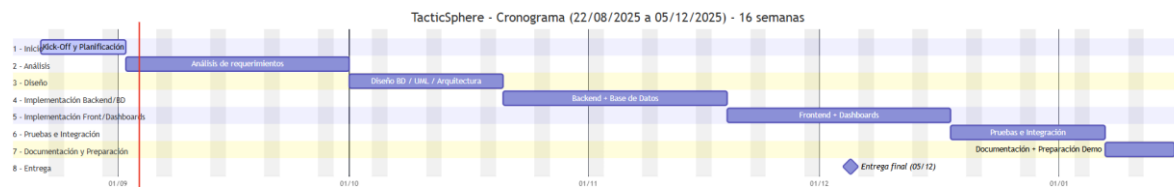
### 9.1. Fases del Proyecto

1. Inicio y Planificación (Semana 1)
  - Reunión inicial y revisión de objetivos.
  - Elaboración del Acta de Kick-Off y Plan de Proyecto.
2. Análisis (Semanas 2–4)
  - Definición de requerimientos funcionales y no funcionales.
  - Modelado de procesos y alcance del sistema.
  - Elaboración del documento ERS.
3. Diseño (Semanas 5–6)
  - Diseño de la base de datos (ERD).
  - Diagramas UML (clases, casos de uso).
  - Definición de arquitectura técnica.
4. Implementación Backend y Base de Datos (Semanas 7–9)
  - Desarrollo de API con FastAPI.
  - Configuración de la base de datos MySQL.
  - Primeras pruebas de integración de datos.
5. Implementación Frontend y Dashboards (Semanas 10–12)
  - Desarrollo de interfaces en Angular
  - Implementación de dashboards y reportes básicos.
  - Validación de conexión frontend–backend.
6. Pruebas e Integración (Semanas 13–14)

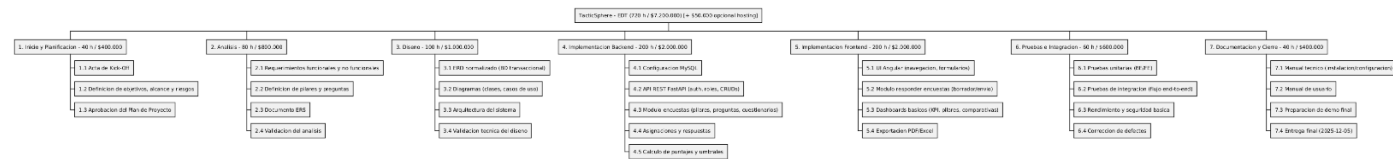


- Pruebas unitarias e integración del flujo completo.
- Ajustes de usabilidad y corrección de errores.
- 7. Documentación y Preparación Final (Semana 15)
  - Elaboración de manual de usuario y documentación técnica.
  - Preparación de demo final.
- 8. Entrega y Presentación (Semana 16 – 5 de diciembre)
  - Presentación oficial del proyecto.
  - Entrega de entregables finales.

## 9.2 Carta Gantt



## 9.3 EDT



# 10. Costos del Proyecto

El desarrollo del proyecto TacticSphere contempla dos grandes categorías de costos: horas de trabajo del equipo e infraestructura tecnológica.

Dado que se trata de un proyecto académico, se busca optimizar recursos utilizando principalmente herramientas gratuitas y equipos propios, asegurando la viabilidad dentro del periodo de ejecución.

## 10.1. Horas de trabajo del equipo

El equipo está conformado por tres integrantes, quienes dedicarán en promedio 15 horas semanales cada uno durante 16 semanas (22 de agosto – 5 de diciembre).

- Horas totales por integrante: 15 h/semana × 16 semanas = 240 h
- Horas totales equipo: 240 h × 3 integrantes = 720 h

La mayor parte de estas horas se concentrará en la fase de desarrollo (backend y frontend), considerando que es la más demandante y crítica para cumplir con los objetivos del MVP. La distribución estimada es la siguiente:

- Inicio y planificación: 40 h
- Análisis: 80 h
- Diseño: 100 h
- Desarrollo backend: 200 h



- Desarrollo frontend: 200 h
- Pruebas e integración: 60 h
- Documentación y cierre: 40 h

Total: 720 h

Si se estimara un valor referencial de \$10.000 CLP por hora (simulación de mercado laboral), el costo total de horas de trabajo equivaldría a:

$$720 \text{ h} \times \$10.000 = \$7.200.000 \text{ CLP}$$

En este contexto académico, estas horas no representan un gasto real, pero sirven para dimensionar el valor del esfuerzo invertido por el equipo y reflejar el énfasis en la fase de desarrollo, que concentra más del 60% de los recursos humanos.

## 10.2. Infraestructura tecnológica

El proyecto se apoyará en equipos propios y en tecnologías gratuitas de código abierto, con costos reales mínimos.

- Equipos de trabajo:
  - Notebook personal por integrante → costo \$0 (ya adquirido, recurso esencial en programación).
- Software y frameworks (todos gratuitos):
  - Base de datos: MySQL (open source).
  - Backend: Python + FastAPI.
  - Frontend: Angular.
  - Dashboards: librerías como Recharts o ECharts.
  - Control de versiones: GitHub (plan educativo gratuito).
  - Contenedores: Docker.
- Servicios en la nube (demo y pruebas):
  - Uso de free tiers en plataformas como Render, Railway o Vercel → costo \$0.
- Otros recursos:
  - Conexión a internet (hogar, ya contratada) → costo \$0.
  - Energía eléctrica (ya cubierta por el entorno académico/hogar) → costo \$0.
  - Posible adquisición de dominio web y hosting básico para presentación → \$50.000 CLP (opcional).



### 10.3. Costos Totales Estimados

Categoría	Estimación Referencial
Horas de trabajo (720 h)	\$7.200.000 CLP (valor de mercado)
Notebooks y equipos	\$0 CLP (ya adquiridos, recursos propios)
Software y frameworks	\$0 CLP (open source)
Servicios en la nube (free tier)	\$0 CLP
Internet y energía eléctrica	\$0 CLP (ya contratados)
Dominio/Hosting (opcional)	\$50.000 CLP
<b>Total estimado</b>	<b>\$7.250.000 CLP</b>

### 10.4. Justificación de costos

- Se incluyen todos los recursos empleados, incluso aquellos ya disponibles (notebooks, internet, energía), registrados como costo \$0 para reflejar su uso.
- Se prioriza el uso de herramientas open source y planes gratuitos para evitar gastos innecesarios.
- El único gasto opcional considerado es la compra de un dominio/hosting para demostrar profesionalismo en la entrega final.
- El mayor valor del proyecto está en las 600 horas de trabajo del equipo, lo que representa el esfuerzo y dedicación del grupo más que un gasto monetario directo.



## 11. Riesgos del Proyecto

En todo proyecto tecnológico existen factores que pueden afectar su desarrollo. Para TacticSphere, se identificaron los principales riesgos en tres dimensiones: técnicos, de gestión y externos. Cada riesgo incluye su posible impacto y la estrategia de mitigación propuesta.

### 11.1. Riesgos Técnicos

#### **R1. Complejidad en la definición de encuestas y pilares**

**Impacto:** Una mala estructuración de las encuestas o de los pilares podría generar resultados poco claros, reduciendo la calidad de los reportes y dificultando la interpretación por parte de las empresas.

**Mitigación:** Definir desde el inicio una guía estandarizada de pilares y preguntas, validada con bibliografía y ejemplos, y limitar el alcance a un número controlado de preguntas para el MVP.

#### **R2. Limitaciones en la infraestructura gratuita**

**Impacto:** servicios free tier pueden tener límites de uso (tiempo de ejecución, memoria, conexiones simultáneas).

**Mitigación:** pruebas de carga con datos simulados; si es necesario, migración temporal a servidores locales para la demo.

#### **R3. Riesgos de usabilidad en la interfaz de usuario**

**Impacto:** Si la interfaz del sistema resulta poco intuitiva, los usuarios (especialmente empleados no técnicos) podrían tener dificultades para completar encuestas, reduciendo la tasa de respuesta.

**Mitigación:** Realizar pruebas tempranas de usabilidad con un grupo reducido de usuarios de prueba, aplicar mejoras en la navegación y mantener un diseño simple con mensajes de ayuda claros.

### 11.2. Riesgos de Gestión

#### **R4. Retrasos en la planificación**

**Impacto:** acumulación de tareas críticas en las últimas semanas del proyecto.

**Mitigación:** aplicación de metodología ágil con entregas parciales (sprints quincenales) y reuniones semanales de seguimiento.

#### **R5. Sobrecarga de trabajo académico**

**Impacto:** la coincidencia con otras asignaturas puede reducir las horas reales dedicadas.





Mitigación: distribución de tareas equitativa entre los tres integrantes y planificación de actividades con anticipación.

#### **R6. Alcance excesivo para el tiempo disponible**

Impacto: intentar incluir funcionalidades avanzadas (ej. machine learning) podría retrasar el MVP.

Mitigación: priorización del MVP con encuestas, reportes y dashboards básicos, dejando mejoras futuras como iteraciones posteriores.

### **11.3. Riesgos Externos**

#### **R7. Dependencia de librerías externas**

Impacto: actualizaciones o bugs en librerías podrían generar incompatibilidades.

Mitigación: congelar versiones de dependencias en los entornos de desarrollo y usar librerías con amplia comunidad.

#### **R8. Problemas en la presentación final (demo)**

Impacto: fallas de conexión a internet o límites en servicios en la nube durante la entrega.

Mitigación: preparar un plan B con despliegue en servidor local y base de datos precargada para asegurar la demo.

#### **R9. Factores personales o de equipo**

Impacto: ausencias por motivos de salud o personales podrían afectar entregas.

Mitigación: documentación clara del avance y repositorio compartido en GitHub para permitir continuidad por cualquier integrante.

## **12. Pruebas del Proyecto**

El objetivo del plan de pruebas es validar que el MVP de TacticSphere cumple con los requerimientos funcionales y no funcionales definidos, asegurando calidad, confiabilidad y una demo estable.

### **12.1. Alcance y objetivos**

- Verificar funcionalidades clave: encuestas, cálculo de puntajes, dashboards, exportaciones, multiempresa y control de acceso.
- Garantizar estabilidad y rendimiento en datos simulados realistas.
- Asegurar usabilidad y claridad para usuarios no técnicos (administradores y encuestados).



## 12.2. Tipos de prueba

### 1. Pruebas Unitarias (PU)

- Cobertura mínima:  $\geq 60\%$  en servicios críticos (auth, cálculo de puntajes, exportación, endpoints de encuestas).
- Enfoque: funciones puras y servicios de dominio (validaciones, pesos, umbrales).
- Herramientas: PyTest (backend), Vitest/RTL (frontend).

### 2. Pruebas de Integración (PI)

- Flujo completo: registro empresa  $\rightarrow$  cuestionario  $\rightarrow$  asignación  $\rightarrow$  respuestas  $\rightarrow$  reporte  $\rightarrow$  exportación.
- Incluye base de datos MySQL y API real en entorno de prueba (Docker).
- Verifica contratos Swagger/OpenAPI.

### 3. Pruebas de Sistema (PS)

- Verifica el sistema end-to-end con datos y roles reales (Admin Sistema, Admin Empresa, Analista, Respondente).
- Cubre RBAC, multiempresa y auditoría.

### 4. Pruebas de Rendimiento (PR)

- Objetivos RNF:
- Respuesta UI  $\leq 2s$  en 95% de solicitudes (listados/dashboards).
- Generar reporte global ( $\leq 5k$  respuestas)  $\leq 10s$ .
- Carga simulada: 50 usuarios / 200–1.000 encuestas.
- Herramientas: locust/k6 (opcional).

### 5. Pruebas de Seguridad (SEG)

- Contraseñas hash (Argon2/bcrypt), JWT con expiración, CORS restringido.
- Intentos de acceso fuera de rol (RBAC), fuerza bruta con bloqueo temporal.
- Verificación de transporte HTTPS en demo.

### 6. Pruebas de Usabilidad (UX)

- Tareas: “crear cuestionario”, “asignar encuesta”, “ver avance”, “descargar PDF”.
- Métrica: éxito de tarea y tiempo promedio ( $< 3$  min por tarea clave).

### 7. Pruebas de Datos/Integridad (DATA)

- Validación de obligatoriedad y rangos (Likert 1–5), consistencia de pesos, anonimato opcional.
- Verificación de histórico y no duplicidad en respuestas.

### 8. Respaldo y Recuperación (BKP)



- Prueba de backup diario y restauración: recuperar sistema en < 30 min con pérdida ≤ 24h.

#### 9. Regresión (REG)

- Re-ejecución de casos críticos al cerrar cada sprint y antes del feature freeze.

#### 12.3. Entornos de prueba

- DEV: desarrollo local (hot reload, datos sintéticos pequeños).
- QA: Docker Compose (frontend, backend, MySQL, Adminer), datos de prueba medianos.
- DEMO: despliegue estable (free tier / local), datos realistas, solo lectura para demo.

#### 12.4. Datos de prueba (semilla)

- 3 empresas (pequeña, mediana, grande).
- 6–9 unidades organizacionales por empresa.
- 60–120 empleados simulados.
- 5 pilares, 40–60 preguntas total.
- 3 cuestionarios (borrador/publicado), 200–1.000 respuestas distribuidas.

#### 12.5. Matriz de pruebas (muestra)

ID	Tipo	Reqs asociados	Descripción	Resultado esperado
TC-PU-01	PU	RF-15	Calcula puntaje pilar con pesos y Likert 1–5	Puntaje correcto y redondeo consistente
TC-PI-03	PI	RF-11/13/14	Asignar encuesta → responder → validar obligatorias	Envío bloquea faltantes; guarda borrador; marca completada
TC-PS-05	PS	RF-04/28	Aislamiento multiempresa y RBAC	Usuario ve solo sus datos; accesos denegados fuera de rol
TC-UAT-07	UAT	RF-17/19	Generar y visualizar reporte global en dashboard	KPI global, top 3 brechas/fortalezas visibles
TC-PR-02	PR	RNF-05	Reporte global ≤ 10s con 5k respuestas	Tiempo ≤ 10s y sin errores 5xx
TC-SEG-04	SEG	RNF-01/02/03	Recuperación de contraseña + políticas + HTTPS	Hash seguro; token válido; HTTPS activo en demo



ID	Tipo	Reqs asociados	Descripción	Resultado esperado
TC-UX-06	UX	RNF-09	Crear y asignar cuestionario en < 3 min	Tarea se completa sin ayuda
TC-DATA-08	DATA	RF-30	Encuesta anónima: no persiste empleado_id	Respuestas sin vínculo personal
TC-BKP-01	BKP	RNF-07	Restaurar backup diario	Sistema operativo con pérdida ≤ 24h
TC-REG-09	REG	Todos críticos	Revisión post-cambios en exportación	Export PDF/Excel sigue funcionando

## 12.6. Criterios de aceptación (UAT)

- Autenticación y roles: acceso según perfil; bloqueo de acciones no autorizadas.
- Encuestas: asignación por empresa/unidad/empleado; validaciones; envío final.
- Resultados: cálculo de puntajes y semáforos con umbrales configurables.
- Dashboards: KPI global, comparativas por área; filtros básicos.
- Exportaciones: PDF y Excel con tablas/gráficos legibles.
- Estabilidad: sin errores críticos en demo (0 fallos bloqueantes).

## 12.7. Criterios de entrada / salida

- Entrada a QA: endpoints clave implementados, Swagger actualizado, datos semilla listos.
- Salida de QA / Feature freeze: 100% de casos críticos OK, sin defectos severos abiertos, rendimiento cumplido.
- Aprobación UAT: todos los criterios de aceptación OK y demo lista.

## 12.8. Gestión de defectos

- Registro en Issues (GitHub): ID, severidad, pasos, esperado/obtenido, adjuntos.
- SLA interno: críticos en 48 h, altos en 72 h, medios en 1 semana.
- Etiquetas por área (frontend / backend / datos / infra).

