

ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE (ERS)

Proyecto: TacticSphere – Consultora de Transformación Digital

Integrantes del equipo:

-Vicente Fernandez Delgado

-Samanta Navarro Peña

-Amaro Veas Quezada

Guia:

-Patricio Soto Serdio



Contenido

ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE (ERS)	1
Proyecto: TacticSphere – Consultora de Transformación Digital	1
1. Introducción.....	4
1.1 Propósito	4
1.2 Alcance del sistema	4
1.3 Definiciones, siglas y abreviaturas	4
1.4 Referencias.....	5
1.5 Visión general del documento	5
2. Descripción General del Sistema	5
2.1 Perspectiva del producto	5
2.2 Funcionalidades principales.....	6
2.3 Usuarios del sistema	6
2.4 Restricciones generales	6
2.5 Suposiciones y dependencias	7
3. Requerimientos Específicos.....	7
3.1 Requerimientos Funcionales (RF)	7
3.2 Requerimientos No Funcionales (RNF).....	11
3.3 Casos de Uso.....	12
4. Modelos del Sistema	14
4.1. Diagrama de Clases	¡Error! Marcador no definido.
4.2. Diagrama Entidad–Relación (Base de Datos)	¡Error! Marcador no definido.
4.3. Diagrama de Casos de Uso.....	15
4.4. Interfaces de Usuario (visión general).....	21
4.5. Interfaces externas.....	22
4.6. Supuestos de diseño y alineación tecnológica.....	22
4.7. Trazabilidad cruzada (resumen)	22
5. Requisitos de Rendimiento	22
5.1 Tiempo de respuesta en la interfaz	22
5.2 Procesamiento de reportes.....	22



5.3 Escalabilidad mínima	23
5.4 Disponibilidad.....	23
5.5 Respaldo y recuperación	23
6. Restricciones de Diseño	23
6.1 Lenguajes y frameworks obligatorios.....	23
6.2 Estándares de seguridad	23
6.3 Infraestructura y despliegue	23
6.4 Normativas y convenciones de desarrollo	24
6.5 Limitaciones conocidas	24
7. Validación y Criterios de Aceptación	24
7.1 Estrategias de validación	24
7.2 Criterios de aceptación generales.....	24
7.3 Aceptación de usuario (UAT)	25
7.4 Matriz de trazabilidad	25
7.5 Condiciones de aceptación final.....	26
8. Apéndices	26
8.1 Glosario de términos	26
8.2 Ejemplo de encuesta (simulada).....	26
8.3 Tablas de referencia	27
8.4 Material de apoyo.....	27



1. Introducción

1.1 Propósito

El presente documento de Especificación de Requerimientos de Software (ERS) tiene como propósito describir de manera clara, detallada y verificable las funcionalidades y restricciones que deberá cumplir el sistema TacticSphere.

Este documento está dirigido principalmente a:

- Equipo de desarrollo: como guía técnica para implementar las funcionalidades.
- Docentes evaluadores: como evidencia formal de planificación y definición de requerimientos.
- Usuarios interesados (simulados): para comprender qué hará el sistema y cómo cubrirá las necesidades planteadas.

La ERS servirá como contrato entre el equipo de trabajo y los interesados, asegurando que el producto final cumpla con las expectativas definidas en el Plan de Proyecto.

1.2 Alcance del sistema

El sistema TacticSphere permitirá a las empresas diagnosticar su nivel de madurez digital a través de nuestros software en base a encuestas estructuradas basadas en pilares de análisis.

Los principales objetivos del sistema son:

- Registrar empresas, áreas, empleados y cuestionarios de forma estructurada.
- Aplicar encuestas segmentadas a nivel de empresa, departamento o empleado.
- Procesar los resultados aplicando un sistema de puntuación y umbrales.
- Generar reportes globales y segmentados, con dashboards visuales y exportación a PDF/Excel.
- Ofrecer recomendaciones personalizadas según los resultados obtenidos.

El sistema se entregará como un MVP (producto mínimo viable) dentro del periodo académico, en un entorno de pruebas controlado.

1.3 Definiciones, siglas y abreviaturas

- API: Application Programming Interface, interfaz de programación para conectar el backend con el frontend.
- RBAC: Role-Based Access Control, control de acceso basado en roles.
- PERT: Program Evaluation and Review Technique, técnica de planificación de proyectos con tiempos probabilísticos.
- Gantt: diagrama de barras utilizado para planificar actividades en el tiempo.
- ERS: Especificación de Requerimientos de Software.
- Dashboard: panel gráfico para visualizar indicadores clave.



1.4 Referencias

- Plan de Proyecto – TacticSphere (documento base que dio origen a este ERS).
- IEEE 830-1998: Recommended Practice for Software Requirements Specifications.
- Documentación oficial de tecnologías utilizadas:
 - FastAPI (<https://fastapi.tiangolo.com/>)
 - MySQL (<https://dev.mysql.com/doc/>)
 - Angular (<https://angular.dev/>)

1.5 Visión general del documento

El presente documento se organiza en las siguientes secciones:

- Sección 1: Introducción. Define propósito, alcance, términos, referencias y estructura del ERS.
- Sección 2: Descripción general. Presenta el contexto del producto, usuarios, restricciones y suposiciones.
- Sección 3: Requerimientos específicos. Lista detallada de requerimientos funcionales y no funcionales, junto con casos de uso asociados.
- Sección 4: Modelos del sistema. Incluye diagramas de clases, ERD, casos de uso e interfaces básicas.
- Sección 5: Requisitos de rendimiento. Establece métricas y criterios medibles de desempeño.
- Sección 6: Restricciones de diseño. Define tecnologías, estándares y limitaciones.
- Sección 7: Validación y criterios de aceptación. Explica cómo se verificará que el sistema cumple lo especificado.
- Sección 8: Apéndices. Contiene glosario, ejemplos de encuestas y material complementario.

2. Descripción General del Sistema

2.1 Perspectiva del producto

El sistema TacticSphere se enmarca como una plataforma web de consultoría digital cuyo propósito es ayudar a empresas a diagnosticar y mejorar su nivel de madurez digital.

- Será un sistema independiente, diseñado como un MVP académico, que funcionará en un entorno de pruebas controlado.
- Su arquitectura seguirá un enfoque de tres capas:
 - Frontend: interfaz web (Angular + TypeScript).
 - Backend/API: servicios de negocio y cálculo (FastAPI en Python).
 - Base de datos: almacenamiento persistente (MySQL).



- Permitirá la interacción directa de los usuarios mediante un navegador web, sin necesidad de instalar software adicional.
- Incluirá funciones de exportación (PDF, Excel) para facilitar la generación de reportes y presentación de resultados.

2.2 Funcionalidades principales

El sistema proporcionará, en su versión inicial, las siguientes funcionalidades:

1. Gestión de usuarios y empresas: registro de empresas, departamentos, áreas y empleados.
2. Gestión de encuestas: creación de pilares y preguntas, construcción de cuestionarios, asignación de encuestas a distintos niveles (empresa, área, empleado).
3. Aplicación de encuestas: acceso de los usuarios para responder de forma segura, con validaciones y posibilidad de respuestas anónimas.
4. Procesamiento de resultados: cálculo de puntajes por pregunta, pilar y empresa, con umbrales de colores (semáforos).
5. Visualización de resultados: dashboards ejecutivos con indicadores clave (KPI global, fortalezas y brechas principales).
6. Reportes: generación de reportes globales y segmentados, con posibilidad de exportación a PDF y Excel.
7. Seguridad y control de acceso: autenticación por credenciales, roles diferenciados y auditoría básica de actividades.

2.3 Usuarios del sistema

Se identifican los siguientes perfiles de usuario y sus necesidades:

- Administrador del sistema: gestiona empresas, usuarios y catálogos globales.
- Administrador de empresa: configura pilares, preguntas, cuestionarios, asignaciones y visualiza resultados.
- Analista/Consultor: accede a reportes y dashboards para analizar la madurez digital.
- Empleado/Respondente: responde encuestas asignadas y recibe retroalimentación indirecta de su contribución.

2.4 Restricciones generales

El sistema deberá cumplir con las siguientes restricciones:

- Tecnológicas: uso de herramientas gratuitas y open source (Python/FastAPI, MySQL, Angular, TypeScript).
- Tiempo: el desarrollo deberá completarse entre el 29 de agosto y la última semana de noviembre de 2025 (\approx 12 semanas).



- Alcance académico: el sistema funcionará en un entorno de pruebas controlado; no se desplegará a producción empresarial real.
- Infraestructura: priorización de entornos locales y servicios gratuitos en la nube (free tiers).

2.5 Suposiciones y dependencias

- Los usuarios cuentan con dispositivos con acceso a internet y navegadores modernos.
- Se asume que los empleados responden las encuestas en línea dentro de los plazos definidos.
- El sistema depende de:
 - Base de datos MySQL para almacenamiento.
 - Librerías de frontend y backend para gráficos, exportaciones y seguridad.
 - Contenedores Docker para la ejecución en entornos de desarrollo y demo.

3. Requerimientos Específicos

3.1 Requerimientos Funcionales (RF)

Los requerimientos funcionales del sistema se listan a continuación con su código identificador. Estos definen las funciones que el sistema deberá implementar obligatoriamente.

Aquí se incorporan los RF-01 al RF-32 que ya definimos en el Plan de Proyecto (autenticación, gestión de empresas, cuestionarios, encuestas, reportes, exportación, auditoría, etc.).

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.

3.2 Requerimientos No Funcionales (RNF)

Los requerimientos no funcionales establecen las condiciones de calidad, seguridad, rendimiento y usabilidad del sistema.

Aquí se listan los RNF-01 al RNF-12 que ya desarrollaste en el plan (seguridad de contraseñas, control de accesos, rendimiento, disponibilidad, trazabilidad, usabilidad, mantenibilidad, etc.).

RNF-01 – Seguridad de contraseñas

Las contraseñas deberán almacenarse con hash seguro (p. ej., bcrypt/Argon2) y política de longitud mínima (≥ 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).

3.3 Casos de Uso

Los casos de uso describen la interacción entre los usuarios y el sistema. A continuación, se presentan los más relevantes:

UC-01: Autenticación de Usuario

- Actores: Administrador de sistema, Administrador de empresa, Analista, Empleado.
- Descripción: Permite a un usuario acceder al sistema mediante credenciales.
- Flujo normal:
 1. El usuario ingresa su correo y contraseña.
 2. El sistema valida credenciales en la base de datos.
 3. El sistema permite el acceso según el rol asignado.
- Flujos alternativos
 - Credenciales inválidas → mensaje de error.
 - Usuario bloqueado → acceso denegado.
- Requerimientos asociados: RF-01, RF-02, RF-03.

UC-02: Gestión de Empresas y Usuarios

- Actores: Administrador del sistema.
- Descripción: Permite registrar, actualizar y desactivar empresas, así como crear y asignar roles a usuarios.



- Flujo normal:
 1. El administrador ingresa datos de la empresa (razón social, rubro, país)
 2. El sistema guarda la información.
 3. El administrador registra usuarios asociados a la empresa y define sus roles.
- Requerimientos asociados: RF-04, RF-05, RF-06, RF-07.

UC-03: Asignación de Encuestas

- Actores: Administrador de empresa.
- Descripción: Permite asignar un cuestionario a toda la empresa, a un área o a empleados específicos.
- Flujo normal:
 1. El administrador selecciona un cuestionario publicado.
 2. Define el alcance de la asignación (empresa, área, empleado).
 3. Configura fechas de inicio y cierre.
 4. El sistema genera notificación de encuesta pendiente para los usuarios asignados.
- Flujos alternativos:
 - Fechas inválidas → error y solicitud de corrección.
- Requerimientos asociados: RF-10, RF-11, RF-12.

UC-04: Responder Encuesta

- Actores: Empleado/Respondente.
- Descripción: Permite al empleado contestar una encuesta asignada, guardar en borrador y enviar respuestas.
- Flujo normal:
 1. El empleado accede a sus encuestas pendientes.
 2. Completa las preguntas.
 3. Puede guardar como borrador o enviar definitivamente.
 4. El sistema valida respuestas obligatorias y registra el envío.
- Flujos alternativos:
 - Preguntas obligatorias sin responder → bloqueo del envío.
- Requerimientos asociados: RF-13, RF-14, RF-31.

UC-05: Generación de Reportes

- Actores: Administrador de empresa, Analista.
- Descripción: Permite generar reportes globales y segmentados con indicadores gráficos.
- Flujo normal:
 1. El usuario selecciona empresa, área o empleado.
 2. El sistema procesa puntajes y aplica umbrales.
 3. Se muestran resultados en dashboard y se permite exportación a PDF/Excel.
- Requerimientos asociados: RF-15, RF-16, RF-17, RF-18, RF-19, RF-20.



4. Modelos del Sistema

Esta sección describe los modelos que representan la estructura, los datos y el comportamiento del sistema TacticSphere. Los diagramas asociados (Clases, Entidad-Relación y Casos de Uso) se incluyen como anexos gráficos y sirven como base para la implementación y la verificación de los requerimientos.

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.

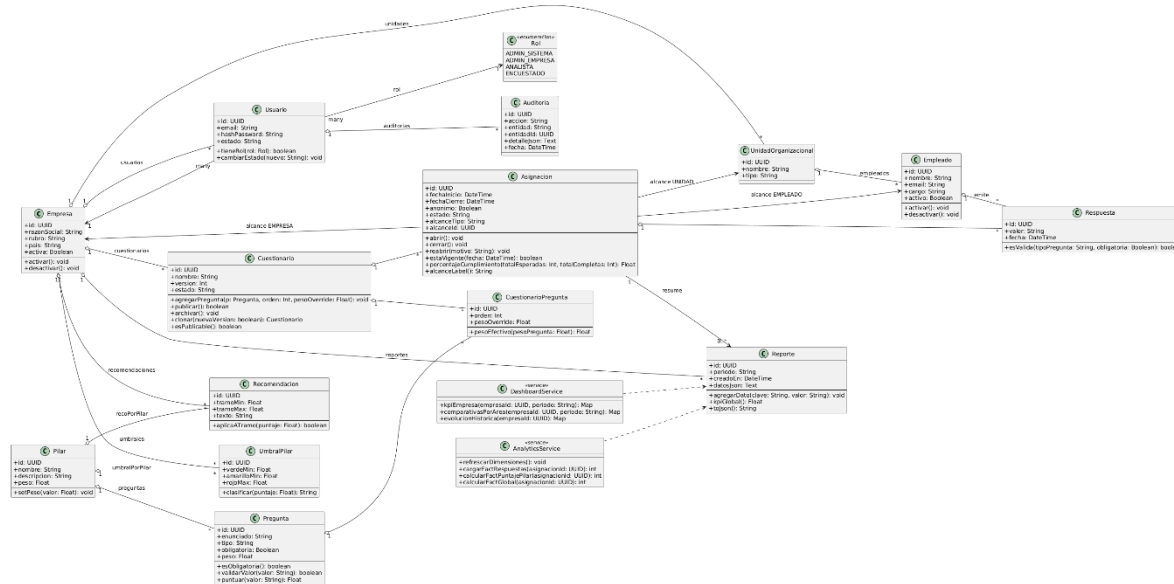
RF-01 a RF-32: autenticación/roles, gestión de empresas/usuarios/unidades, definición de pilares/preguntas, versionado y asignación de cuestionarios, validaciones, scoring, reportes, exportación, auditoría.

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

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, auditoría, 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/empleado).



Anonimato opcional: respuestas.empleado_id puede ser NULL.

Relación con requisitos.

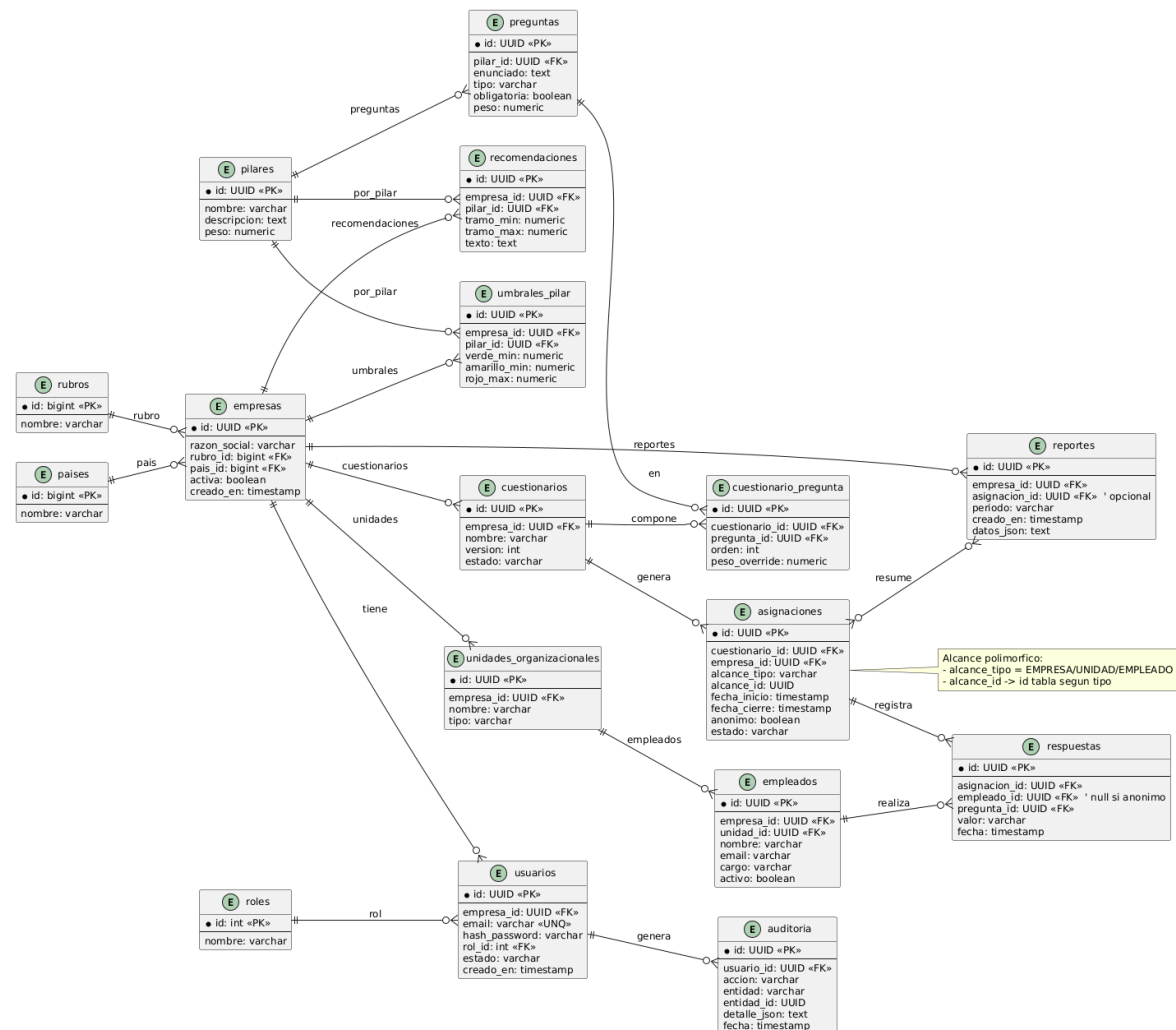
Soporta el ciclo completo: definición de instrumentos → asignación → respuesta → cálculo y reporte.

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.

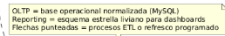
Acelera RF-17/18/19/20 (reportes y dashboard), ayuda a cumplir RNF-04/05 (rendimiento).

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.



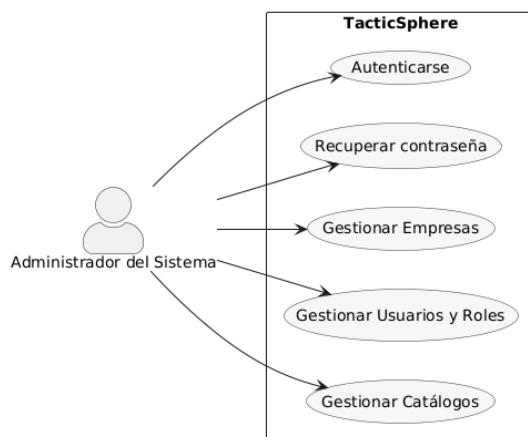


4.3. Diagrama de Casos de Uso

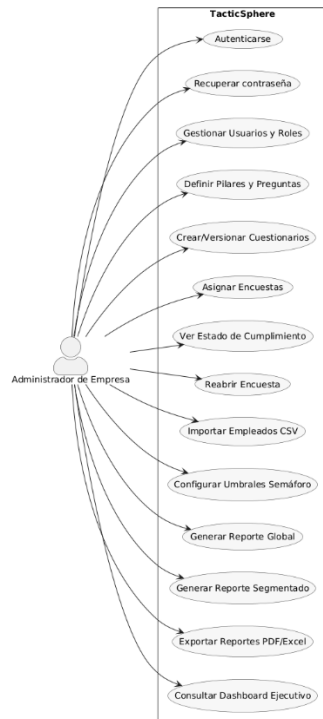
Propósito. Describir la interacción entre actores y el sistema, mostrando los servicios que ofrece y sus límites.

Actores.

- Administrador del Sistema: opera empresas, usuarios y catálogos globales.

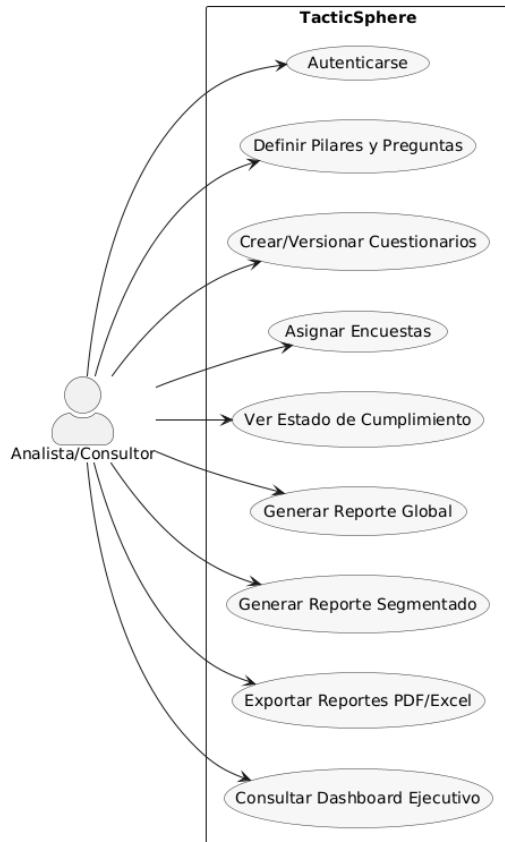


- Administrador de Empresa: configura pilares, cuestionarios, asignaciones y umbrales; consulta reportes.

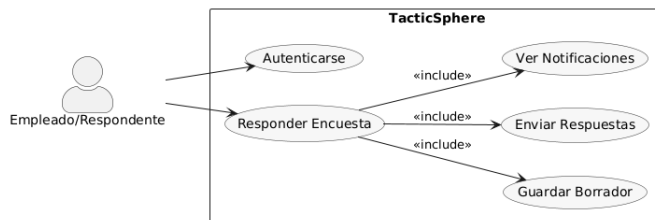


- Analista/Consultor: consume dashboards y reportes; exporta resultados.





- Empleado/Respondente: responde encuestas asignadas.



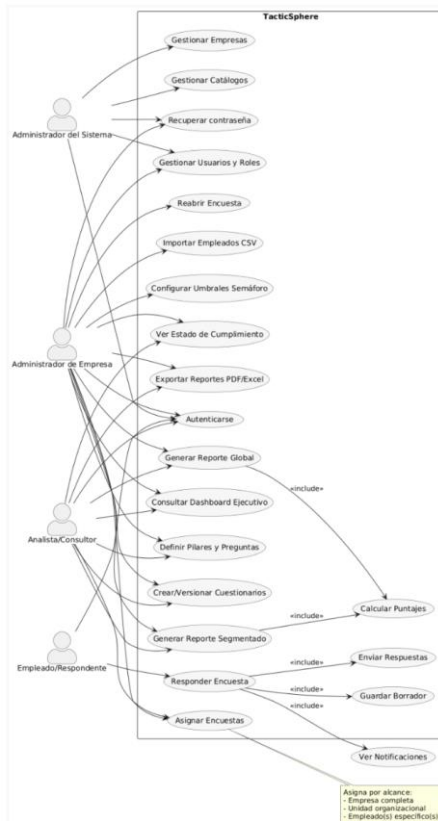
Casos principales.

- Autenticarse / Recuperar contraseña: control de acceso y continuidad.
- Gestionar Empresas, Usuarios, Catálogos: preparación del entorno de medición.
- Definir Pilares y Preguntas / Crear/Versionar Cuestionarios: modelado de instrumentos.
- Asignar Encuestas / Ver Estado de Cumplimiento / Reabrir Encuesta / Importar Empleados: operación del ciclo de levantamiento.
- Responder Encuesta (borrador/envío): captura de datos con validaciones.
- Calcular Puntajes / Configurar Umbrales: procesamiento y clasificación.
- Generar Reporte Global/Segmentado / Exportar PDF-Excel / Consultar Dashboard Ejecutivo: análisis y comunicación de resultados.



Relaciones. Inclusiones entre Responder Encuesta ↔ Borrador/Enviar; Generar Reporte ↔ Calcular Puntajes refuerzan la cohesión entre funcionalidades.

Trazabilidad. Cobertura directa de RF-01-02/05-06/08-20/22/25-27/31-32 y RNF-09.



4.4. Interfaces de Usuario (visión general)

Objetivo. Alinear la experiencia con los requerimientos de usabilidad y flujo operativo.

Pantallas previstas (baja fidelidad).

- Inicio de sesión (credenciales, recuperación).
- Panel del Administrador de Empresa (KPI global, estado de cumplimiento, accesos a encuestas y reportes).
- Gestión de cuestionarios (pilares, preguntas, versiones).
- Asignación de encuestas (alcance, fechas, anonimato).
- Respuesta de encuestas (validaciones, borrador/envío).
- Dashboards (semáforos por pilar, comparativas por área, top brechas/fortalezas).
- Exportaciones (descarga PDF/Excel).



Criterios de usabilidad. Navegación consistente, formularios con validación en línea, mensajes claros (RNF-09), soporte básico (RNF-10).

4.5. Interfaces externas

- Exportación de resultados: generación de PDF y Excel desde el backend, consumidos desde la UI (RF-20).
- Servicios de autenticación interna: manejo de JWT y roles (RNF-01/02).
- Herramientas de administración de base de datos: uso de clientes estándar (Adminer/DBeaiver) en entornos de desarrollo y demo (RNF-07).

4.6. Supuestos de diseño y alineación tecnológica

- Arquitectura cliente-servidor (Angular ↔ FastAPI ↔ MySQL).
- Despliegue en contenedores con separación de servicios (RNF-11).
- Documentación de endpoints mediante OpenAPI/Swagger para facilitar pruebas e integración (RNF-12).

4.7. Trazabilidad cruzada (resumen)

- Modelos → Requerimientos:
 - Clases Asignacion/Respuesta/Reporte ↔ RF-11/13/17-20/29/31.
 - UmbralPilar/Recomendacion ↔ RF-16/32.
 - Auditoria ↔ RF-21 y RNF-08.
- Modelos → Pruebas:
 - Casos de uso UC-03/UC-04/UC-05 ↔ pruebas de integración, aceptación y rendimiento (Sección 12).

5. Requisitos de Rendimiento

Esta sección define los criterios medibles de desempeño que el sistema TacticSphere deberá cumplir para garantizar una experiencia de usuario fluida y confiable.

5.1 Tiempo de respuesta en la interfaz

Las vistas principales (login, dashboard, listado de encuestas, reportes) deberán cargarse en ≤ 2 segundos en al menos el 95% de los casos bajo condiciones normales de uso.

El tiempo de validación de formularios y guardado de respuestas en borrador no deberá exceder 1 segundo desde el envío.

5.2 Procesamiento de reportes

El cálculo de puntajes y la generación de reportes globales de una empresa con hasta 5.000 respuestas deberá completarse en ≤ 10 segundos.

Los reportes segmentados por área/empleado deberán generarse en ≤ 5 segundos en escenarios de carga estándar.



5.3 Escalabilidad mínima

El sistema deberá soportar hasta 50 usuarios concurrentes en la demo académica sin degradación significativa del rendimiento.

La arquitectura (frontend, backend y base de datos desacoplados en contenedores) permitirá una futura escalabilidad horizontal.

5.4 Disponibilidad

Durante el periodo de evaluación, la disponibilidad del entorno de demo deberá ser de al menos 95% mensual, considerando que se usará infraestructura gratuita (free tiers y entornos locales).

5.5 Respaldo y recuperación

La base de datos deberá contar con respaldos diarios automáticos y la posibilidad de restauración en un tiempo máximo de 30 minutos, con una pérdida máxima de datos de 24 horas.

6. Restricciones de Diseño

Esta sección define las limitaciones técnicas, normativas y de entorno que condicionan el diseño del sistema TacticSphere.

6.1 Lenguajes y frameworks obligatorios

- Backend: Python con FastAPI como framework para servicios REST.
- Frontend: Angular y CSS para la interfaz gráfica.
- Base de datos: MySQL como motor de almacenamiento relacional.
- Contenedores: Docker para empaquetar y desplegar servicios en entornos controlados.

Justificación: Todas son tecnologías gratuitas, de código abierto y con amplia comunidad, lo que facilita su uso en el proyecto académico y asegura mantenibilidad.

6.2 Estándares de seguridad

- Las contraseñas deberán almacenarse con algoritmos de hash seguros (Argon2 o bcrypt).
- El sistema usará JWT (JSON Web Tokens) para autenticación y gestión de sesiones.
- Toda comunicación en la demo deberá implementarse bajo HTTPS.

6.3 Infraestructura y despliegue

- Se utilizarán principalmente entornos locales y servicios gratuitos de la nube (free tiers) para la demo.



- No se contempla despliegue en entornos productivos empresariales (solo académico).
- Los contenedores deberán separar el frontend, backend y base de datos, asegurando modularidad.

6.4 Normativas y convenciones de desarrollo

- El código seguirá convenciones de estilo establecidas por cada lenguaje.
- Se deberá documentar la API con OpenAPI/Swagger.
- Cobertura mínima de 60% en pruebas unitarias para servicios críticos.

6.5 Limitaciones conocidas

- El sistema funcionará como MVP en un entorno de pruebas controlado, no preparado aún para uso masivo en producción.
- No se implementarán en esta versión inicial algoritmos avanzados de Machine Learning o Deep Learning (previstos para fases futuras).
- La escalabilidad horizontal solo será conceptual, dado que los recursos se limitan a free tiers y contenedores locales.

7. Validación y Criterios de Aceptación

La validación del sistema TacticSphere se llevará a cabo mediante diferentes estrategias de prueba y verificación, asegurando que cada requerimiento funcional y no funcional quede cubierto. Esta sección define los mecanismos de validación y los criterios objetivos de aceptación.

7.1 Estrategias de validación

- Revisión de requerimientos: verificación cruzada entre RF/RNF y modelos del sistema (clases, ERD, casos de uso).
- Pruebas de software: ejecución de pruebas unitarias, de integración, de sistema y de aceptación de usuario.
- Demostraciones parciales (sprints): presentación de avances funcionales al finalizar cada sprint quincenal.
- Revisión documental: validación de que cada entregable (Plan de Proyecto, ERS, Kick-Off, manuales) está alineado con lo solicitado en la rúbrica académica.

7.2 Criterios de aceptación generales

1. Requerimientos funcionales: cada RF será validado con al menos un caso de prueba exitoso (ejemplo: asignación de encuestas → encuesta visible al empleado → respuestas registradas).
2. Requerimientos no funcionales: se validarán con métricas cuantitativas:
 - Tiempo de carga en dashboard ≤ 2 segundos en 95% de solicitudes.
 - Reporte global ($\leq 5k$ respuestas) generado en ≤ 10 segundos.



- Disponibilidad del demo $\geq 95\%$ en periodo de evaluación.
- 3. Seguridad: validación de autenticación con credenciales, cifrado de contraseñas y control de accesos por roles.
- 4. Usabilidad: pruebas con usuarios de prueba (compañeros/docentes) que deberán completar tareas clave (crear cuestionario, responder encuesta, generar reporte) en tiempos razonables (< 3 min).
- 5. Compatibilidad: el sistema deberá funcionar en navegadores modernos (Chrome, Edge, Firefox).
- 6. Documentación: entrega de manual de usuario, manual técnico y Swagger de la API.

7.3 Aceptación de usuario (UAT)

- Los casos de uso principales (UC-01 a UC-05) deberán ser ejecutados por usuarios de prueba y aceptados sin errores críticos.
- El sistema deberá demostrar:
 - Gestión de empresas y usuarios.
 - Creación y asignación de encuestas.
 - Respuesta de encuestas por empleados.
 - Cálculo automático de puntajes y semáforos.
 - Visualización de dashboards y generación de reportes exportables.

7.4 Matriz de trazabilidad

Cada requerimiento tendrá asociado al menos un caso de uso y uno o más casos de prueba.
Ejemplo:

Requerimiento	Caso de Uso	Caso de Prueba	Validación
RF-11 (Asignación de encuestas)	UC-03	TC-PI-03	Encuesta asignada se muestra en panel de empleado y se puede responder



Requerimiento	Caso de Uso	Caso de Prueba	Validación
RF-19 (Dashboard ejecutivo)	UC-05	TC-UAT-07	KPI global y comparativa de pilares visibles en dashboard
RNF-04 (Rendimiento UI $\leq 2s$)	UC-05	TC-PR-02	Tiempo de carga del dashboard medido en pruebas de rendimiento

7.5 Condiciones de aceptación final

El proyecto se considerará aceptado cuando:

- Todos los casos de prueba críticos se ejecuten con éxito.
- No existan defectos de severidad alta o crítica pendientes.
- Los criterios de rendimiento y seguridad estén validados.
- La demo del sistema (MVP) muestre un flujo completo Empresa → Encuesta → Respuesta → Reporte funcionando correctamente.

8. Apéndices

8.1 Glosario de términos

- MVP (Minimum Viable Product): Producto mínimo viable; versión inicial del sistema con las funciones esenciales.
- RBAC (Role-Based Access Control): Control de acceso basado en roles; restringe funcionalidades según el perfil del usuario.
- KPI (Key Performance Indicator): Indicador clave de rendimiento usado en dashboards para medir resultados.
- Dashboard: Panel visual que muestra métricas e indicadores en forma de gráficos y tablas.
- Free tier: Plan gratuito ofrecido por plataformas de nube, con recursos limitados.
- API (Application Programming Interface): Interfaz de programación que permite la comunicación entre backend y frontend.
- Swagger/OpenAPI: Herramienta para documentar y probar APIs REST.
- Semáforo (Rojo/Amarillo/Verde): Representación visual usada en los reportes para clasificar el nivel de madurez en cada pilar.

8.2 Ejemplo de encuesta (simulada)

- Pilar: Tecnología
- Preguntas de ejemplo:
- ¿Su empresa cuenta con sistemas de gestión en la nube? (Sí/No)



- ¿Con qué frecuencia se actualizan los sistemas internos? (Likert 1–5)
- ¿Existen protocolos documentados de respaldo de información? (Sí/No)
- Pilar: Talento
- ¿La empresa ofrece capacitación en herramientas digitales a los empleados? (Likert 1–5)
- ¿Existen programas de innovación liderados por equipos internos? (Sí/No)

8.3 Tablas de referencia

- Estados de cuestionarios: BORRADOR, PUBLICADO, ARCHIVADO.
- Estados de asignaciones: ABIERTA, CERRADA, REABIERTA.
- Tipos de preguntas: LIKERT, SI/NO, OPCIÓN MÚLTIPLE.

8.4 Material de apoyo

- PlantUML de diagramas: Se adjuntan en anexos los códigos fuente de diagramas de clases, ERD y casos de uso para permitir su regeneración.
- Manual preliminar de usuario: Documento separado con capturas de mockups básicos (login, encuesta, dashboard).
- Referencias académicas y técnicas:
- IEEE 830-1998 – Recommended Practice for Software Requirements Specifications.
- Documentación de FastAPI, MySQL, Angular.
- Rúbrica del Capstone y guías entregadas por la escuela.

