

GUÍA DE ESTRUCTURA DEL INFORME TFT

Carrera: Carrera de Ingeniería en Tecnologías de la
Información

Semestre:
2026-I

Estudiante: [Nombre del estudiante]

Propósito de este documento. Esta plantilla define la estructura mínima del informe de Trabajo de Fin de Titulación (TFT), indicando qué contenido debe presentar el estudiante en cada apartado y cómo será evaluado mediante rúbrica.

1 Introducción

La introducción debe explicar **qué problema aborda el TFT, por qué es relevante y en qué contexto se aplicará la solución**. Esta sección enmarca el trabajo para que un lector no técnico comprenda su propósito antes de revisar detalles de implementación.

Qué debe incluir

- Contexto institucional, académico o empresarial donde surge la necesidad.
- Problema central y evidencia de su existencia (indicadores, observaciones, entrevistas o reportes).
- Justificación del TFT: impacto esperado, beneficiarios y valor agregado.
- Delimitación inicial del proyecto (qué parte del problema se atenderá en el periodo del TFT).

Cómo redactarla

- Inicie con el contexto general y cierre con el propósito específico del proyecto.
- Use datos verificables para justificar la necesidad; evite afirmaciones sin respaldo.
- Mantenga enfoque conceptual: no incluya aquí detalles técnicos profundos.

Evidencias sugeridas

- Referencias a normativa, artículos, reportes internos o documentos de levantamiento.
- Línea base del problema (tiempos, costos, errores, nivel de servicio u otro indicador pertinente).

Errores comunes a evitar

- Confundir introducción con antecedentes teóricos extensos.
- Presentar objetivos o solución técnica sin haber definido claramente el problema.
- Redactar una introducción genérica que no describa el contexto real del estudiante.

2 Objetivos

Los objetivos definen los logros verificables del TFT y sirven como base para evaluar resultados. Deben tener relación directa con el problema planteado y con la rúbrica de evaluación.

Estructura mínima

- **Objetivo general:** un enunciado que describa el logro principal del proyecto.
- **Objetivos específicos:** entre dos y cinco enunciados que desglosen el objetivo general en resultados parciales medibles.

Plantilla de redacción

Verbo en infinitivo + objeto de intervención + propósito + contexto/condiciones.

Criterios de calidad

- Utilizar verbos observables: *analizar, diseñar, implementar, validar, evaluar, optimizar*.
- Evitar verbos ambiguos: *aprender, entender, conocer*.
- Asegurar que cada objetivo específico produzca una evidencia concreta.
- Mantener coherencia entre objetivos, alcance, desarrollo y pruebas.

Evidencias sugeridas

- Tabla de trazabilidad objetivo-evidencia (por ejemplo: objetivo, actividad, entregable, criterio de aceptación).
- Relación explícita entre objetivos y criterios de rúbrica.

3 Alcance

El alcance establece los límites técnicos y funcionales del TFT para evitar interpretaciones ambiguas sobre lo que el sistema entrega en la versión final.

Qué debe incluir

- Funcionalidades o componentes **incluidos** en el proyecto.
- Funcionalidades o componentes **excluidos** (fuera del periodo o recursos del TFT).
- Restricciones relevantes: tiempo, datos disponibles, infraestructura, permisos o presupuesto.
- Supuestos de trabajo que condicionan la implementación.

Formato recomendado

- Presentar una tabla con tres columnas: *Elemento, Incluye/No incluye, Justificación*.
- Complementar con un diagrama simple de frontera del sistema cuando aplique.

Evidencias sugeridas

- Lista priorizada de requerimientos dentro del alcance del TFT.
- Declaración de versión objetivo (MVP o entrega final definida).

Errores comunes a evitar

- Definir un alcance demasiado amplio sin respaldo en tiempo y recursos.
- Omitir exclusiones, generando expectativas no evaluables.

4 Requisitos del sistema

Esta sección debe especificar las condiciones técnicas necesarias para ejecutar y evaluar la solución. El objetivo es permitir replicabilidad y despliegue consistente.

Clasificación recomendada

- 1. Requisitos de software** Lenguajes, frameworks, librerías, sistema operativo, versiones mínimas, gestor de paquetes y servicios requeridos.
- 2. Requisitos de hardware** Capacidad mínima de procesador, memoria, almacenamiento y dispositivos adicionales (si aplica).
- 3. Dependencias externas** APIs, servicios en la nube, motores de base de datos, credenciales y políticas de acceso.
- 4. Requisitos no funcionales** Criterios de rendimiento, seguridad, disponibilidad, mantenibilidad y compatibilidad.

Formato recomendado

- Tabla por categorías con: *requisito, versión o umbral, obligatorio/opcional, justificación*.
- Incluir comandos de verificación de versiones para los componentes críticos.

Evidencias sugeridas

- Archivo de dependencias del proyecto (por ejemplo: `requirements.txt`, `package.json`, `pom.xml`).
- Evidencia de instalación de dependencias clave y conectividad con servicios externos.

5 Arquitectura del sistema

La arquitectura describe cómo se organizan los componentes del sistema y cómo fluye la información entre ellos. Debe mostrar decisiones de diseño, no solo listar tecnologías.

Contenido mínimo

- Vista general de la solución (capas, módulos o microservicios).
- Responsabilidad de cada componente principal.
- Flujo de datos de entrada, procesamiento y salida.
- Integraciones externas y puntos de fallo relevantes.

Representación recomendada

- Un diagrama de arquitectura de alto nivel.
- Si aplica, diagramas complementarios: despliegue, componentes o secuencia para casos críticos.

Justificación técnica

- Explicar por qué la arquitectura elegida es adecuada para el alcance y requisitos.
- Indicar compromisos de diseño (trade-offs): complejidad vs. mantenibilidad, costo vs. rendimiento, entre otros.

Evidencias sugeridas

- Relación entre componentes del diagrama y directorios reales del repositorio.
- Descripción de interfaces, contratos o APIs internas/externas.

6 Instalación y configuración

Esta sección debe permitir que un tercero despliegue la solución **sin asistencia adicional**. El procedimiento debe ser reproducible en un entorno limpio.

Estructura recomendada

- **Prerrequisitos:** herramientas y versiones necesarias.
- **Instalación:** descarga del código, instalación de dependencias, preparación de base de datos o servicios.
- **Configuración:** variables de entorno, archivos de parámetros y secretos.
- **Inicialización:** carga de datos base, migraciones y verificación de estado.

Buenas prácticas de documentación

- Numerar pasos en orden estricto y usar comandos copiables.
- Incluir resultado esperado por paso (salida mínima de validación).
- Separar claramente configuración local, pruebas y producción.
- Nunca incluir credenciales reales; usar plantillas tipo `.env.example`.

Evidencias sugeridas

- Capturas o logs de instalación exitosa.
- Checklist de verificación post-instalación.

7 Uso del sistema

Aquí se explica cómo operar la solución una vez desplegada: flujos principales, entradas requeridas, salidas esperadas y manejo básico de errores.

Contenido mínimo

- Procedimiento para ejecutar el sistema (modo local y/o servidor).
- Casos de uso principales vinculados con los objetivos del TFT.
- Ejemplos de entrada y salida representativos.
- Parámetros de ejecución, opciones de configuración y límites de uso.

Formato recomendado

- Para CLI: comandos de ejemplo con contexto y explicación breve.
- Para web/API: flujo de navegación o consumo de endpoints con ejemplos de petición/respuesta.
- Incluir condiciones de error comunes y su resolución inicial.

Evidencias sugeridas

- Escenarios de uso validados con datos reales o simulados.
- Registro de resultados obtenidos y comparación con comportamiento esperado.

8 Estructura del repositorio

La estructura del repositorio debe facilitar lectura, mantenimiento y colaboración. Esta sección documenta la organización del código y el propósito de cada carpeta clave.

Qué debe incluir

- Árbol resumido de directorios principales.
- Función de cada carpeta y convenciones de nombres.
- Ubicación de documentación, scripts de automatización y pruebas.
- Archivos de configuración críticos y cómo se usan.

Formato recomendado



Ejemplo de organización

```
proyecto/  
  src/           # código fuente principal  
  tests/         # pruebas automáticas  
  docs/          # documentación técnica  
  scripts/       # automatizaciones de setup/build/deploy  
  config/        # archivos de configuración  
  README.md      # guía rápida de uso
```

Evidencias sugeridas

- Correspondencia entre arquitectura lógica y estructura física del código.
- Justificación breve cuando la estructura difiere de convenciones estándar.

9 Pruebas

Las pruebas demuestran que la solución cumple su propósito bajo criterios verificables. La sección debe cubrir estrategia, ejecución y evidencia de resultados.

Contenido mínimo

- Tipos de prueba aplicados: unitarias, integración, sistema, aceptación o rendimiento.
- Casos de prueba trazables a objetivos y requisitos.
- Procedimiento para ejecutar pruebas (comandos, entorno y datos).
- Resultado esperado, resultado obtenido y análisis de brechas.

Indicadores recomendados

- Tasa de pruebas aprobadas.
- Cobertura de funcionalidades críticas.
- Incidencias detectadas, severidad y estado de corrección.

Evidencias sugeridas

- Reportes automáticos (logs, cobertura, capturas de CI/CD).
- Matriz de trazabilidad *requisito -¿ prueba -¿ resultado*.

Errores comunes a evitar

- Reportar solo pruebas exitosas y omitir fallos relevantes.
- Presentar capturas sin contexto del caso de prueba.

10 Mantenimiento y contribución

Esta sección define cómo evoluciona el proyecto después de la entrega del TFT: prácticas de calidad, control de cambios y colaboración técnica.

Contenido mínimo

- Estrategia de versionado (por ejemplo, SemVer) y política de releases.
- Flujo de trabajo de ramas y revisiones (`main`, `develop`, `feature/*`, `hotfix/*`).
- Estándares de código: formato, linting, estilo y documentación.
- Proceso para reportar incidencias y proponer mejoras (plantillas de issue/PR).

Prácticas recomendadas

- Definir convenciones de commits y criterios de aprobación de PR.
- Automatizar validaciones mínimas en integración continua.
- Mantener changelog y registro de decisiones técnicas relevantes.

Evidencias sugeridas

- Archivos `CONTRIBUTING.md`, `CODE_OF_CONDUCT.md` y plantillas en `.github/` (si aplica).
- Ejemplo de incidencia cerrada y mejora integrada con trazabilidad.

11 Rúbrica de evaluación del informe TFT

La rúbrica evalúa la calidad técnica y documental del informe entregado por el estudiante. Cada criterio se alinea con una sección obligatoria del documento y debe valorarse con evidencia verificable.

Escala de desempeño

- **Alcanzado (90–100%)**: evidencia completa, consistente y técnicamente sólida.
- **Medianamente (80–89%)**: evidencia adecuada con observaciones menores.
- **Parcialmente (60–79%)**: evidencia incompleta o con debilidades relevantes.
- **No alcanzado (0–59%)**: ausencia de evidencia suficiente o errores críticos.

Table 1:

Criterio de evaluación	Peso (pts.)	Alcanzado (90–100)%	Medianamente (80–89)%	Parcialmente (60–79)%	No alcanzado (0–59)%
[RdA1] Introducción y contexto del problema	10	Contexto, problema y justificación claramente sustentados con evidencia.	Contexto y problema definidos; sustento parcial o poco profundo.	Descripción general sin delimitación clara del problema.	No define contexto ni problema de forma verificable.

Continued on next page

Table 1: (Continued)

[RdA1] Formulación de objetivos	10	Objetivo general y específicos medibles, coherentes y trazables.	Objetivos mayormente claros con leves am- bigüedades.	Objetivos incompletos o parcialmente medibles.	Objetivos ausentes o no evaluables.
[RdA1] Delimitación del alcance	8	Incluye/excluye con criterios claros y restricciones explícitas.	Delimitación adecuada con omisiones menores.	Alcance ambiguo o poco justificable.	No define límites del proyecto.
[RdA2] Requisitos del sistema	12	Requisitos fun- cionales/no funcionales y dependen- cias detallados y verificables.	Requisitos suficientes, con detalle técnico parcial.	Requisitos incompletos o sin criterios de verificación.	No identifica requisitos críticos.
[RdA2] Arquitectura del sistema	12	Arquitectura bien estruc- turada, justificada y consistente con el repositorio.	Arquitectura correcta con justificación limitada.	Arquitectura superficial o inconsis- tente.	No presenta arquitectura útil para comprensión técnica.
[RdA2] Instalación y configuración	12	Procedimiento reproducible, ordenado y validado en entorno limpio.	Procedimiento funcional con ajustes menores no críticos.	Pasos incompletos o poco re- producibles.	No permite desplegar el sistema.
[RdA2] Uso del sistema	10	Flujos de uso claros, ejemplos representa- tivos y manejo básico de errores.	Uso descrito correcta- mente con ejemplos limitados.	Descripción de uso parcial o confusa.	No explica operación del sistema.
[RdA3] Estructura del repositorio	8	Organización clara, coherente y alineada a la arquitectura.	Estructura comprensible con leves inconsisten- cias.	Organización débil o poco documen- tada.	Repositorio desordenado o sin docu- mentación estructural.

Continued on next page

Table 1: (Continued)

[RdA3] Pruebas y evidencias de calidad	10	Casos de prueba trazables, resultados analizados y métricas reportadas.	Pruebas suficientes con análisis parcial.	Pruebas limitadas o sin trazabilidad completa.	No presenta pruebas verificables.
[RdA3] Mantenimiento y contribución	8	Define versionado, flujo de trabajo y estándares de contribución aplicables.	Define prácticas básicas con faltantes menores.	Prácticas de manten- imiento incompletas o ambiguas.	No define lineamientos de evolución del proyecto.

Instrucciones para evaluación docente

- Calificar cada criterio según la evidencia presentada en el informe y anexos.
- Registrar observaciones concretas por criterio para retroalimentación formativa.
- La nota final corresponde a la suma ponderada de los criterios (100 puntos).

ELABORADO POR:	REVISADO POR:	APROBADO POR:
Firma:	Firma:	Firma:
Nombre: Mgt. Velasteguí Izurieta Homero Javier	Nombre: Mgt. Carvajal Carvajal José Luis	Nombre: Mgt. Puente Holgún Washington David
Cargo: Coordinador de Carrera	Cargo: Responsable de Aseguramiento de la Calidad	Cargo: Director de Escuela
Fecha: 23/02/2026	Fecha: 25/02/2026	Fecha: 27/02/2026

A Anexos