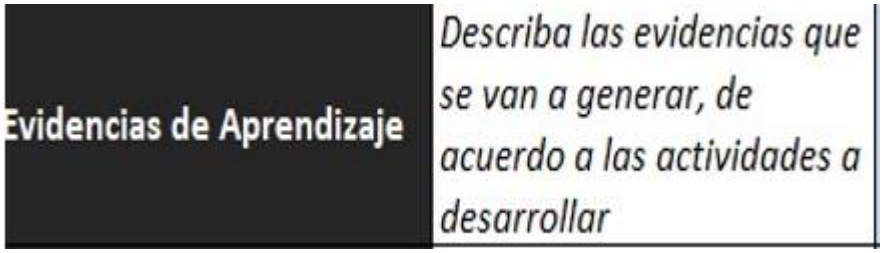
Actividades a Desarrollar:	*Relacione las actividades que el(la) aprendiz va a realizar según lo acordado con el ente coformador. (Estas deben corresponder al Perfil del egresado establecido en el programa de formación que el aprendiz está*

Desarrollo del Sitio Web Institucional	Implementar un sitio web moderno y responsivo utilizando Django (backend) + Bootstrap 5 (frontend) para mostrar los servicios de Personal Technology (asesoría en TI, diseño web, configuración de servidores, mantenimiento, etc.). Incluir secciones como: catálogo de servicios, formulario de contacto, panel de clientes y blog técnico.
Sistema de Gestión de Servicios Técnicos	Desarrollar una aplicación web para administrar servicios técnicos (revisión/diagnóstico, instalación, mantenimiento, diseño web, etc.), con: • Registro de clientes (TIPO CLIENTE: contrato TI o servicio puntual). • Formulario técnico digitalizado (similar al formato físico proporcionado). • Generación automática de reportes en PDF (diagnósticos, órdenes de servicio).
Autenticación y Control de Accesos	Implementar un sistema de login seguro (Django-auth) para técnicos y clientes. Roles diferenciados: administrador, técnico, cliente.
Integración con Base de Datos PostgreSQL	Diseñar un esquema de base de datos relacional para gestionar: • Servicios realizados (tipo, cliente, productos entregados). • Inventario de equipos (modelo, número de serie). • Historial de visitas técnicas.
Dockerización y Despliegue en Azure	Crear Dockerfiles y docker-compose.yml para empaquetar la aplicación (backend + frontend + PostgreSQL). Configurar despliegue automático en Azure App Services (con CI/CD mediante GitHub Actions o Azure DevOps). Asegurar escalabilidad y alta disponibilidad en la nube.
API REST para Integraciones	Desarrollar una API en Django REST Framework para: • Sincronizar datos con herramientas externas (ej: envío de diagnósticos por email). • Conectar con sensores/equipos técnicos (ej: routers, servidores).

Panel de Reportes y Dashboard	Visualizar métricas clave: servicios más solicitados, tiempo de respuesta, clientes frecuentes. Exportar datos a CSV/PDF (ej: listado de equipos con mantenimiento pendiente).
Documentación Técnica	Manual de usuario para técnicos y clientes. Documentación de la API (Swagger/OpenAPI).



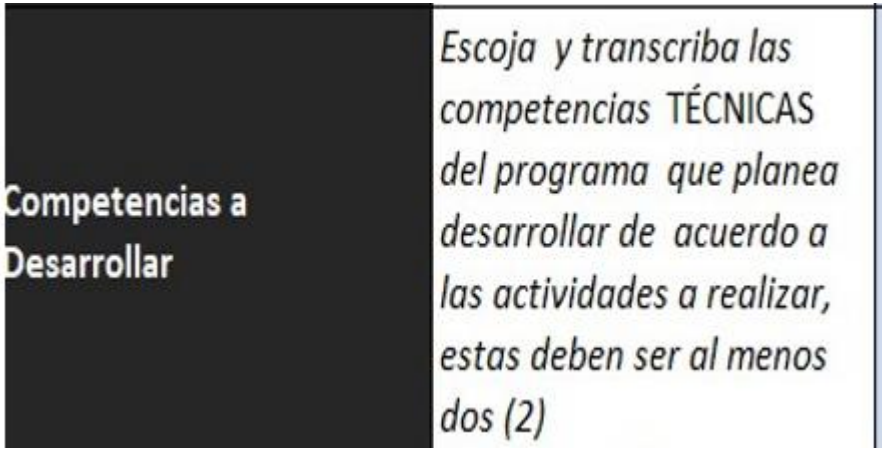
Repositorio de Código (GitHub/GitLab)	Código fuente completo del proyecto (backend en Django y frontend con Bootstrap 5). Estructura organizada en carpetas: • backend/ (models, views, API endpoints, settings). • frontend/ (templates HTML, CSS/JS personalizados, assets). • docker/ (archivos de configuración para contenedores). README.md con: • Instrucciones de instalación y despliegue. • Diagrama de la arquitectura (flujo de datos entre componentes).
Sitio Web en Producción (Azure)	URL pública de la aplicación desplegada en Azure App Services. Funcionalidades accesibles: • Página principal con servicios ofrecidos. • Formulario de solicitud de servicios técnicos. • Panel de administración (ej: admin/ de Django). Certificado SSL activo (HTTPS)
Contenedores Docker	Evidencia técnica: • Dockerfile para el backend (Django + Gunicorn). • Dockerfile para el frontend (servidor estático, ej: Nginx).

	• docker-compose.yml con servicios: Django, PostgreSQL, Nginx. Capturas de pantalla de: • Containers corriendo en local (docker ps). • Logs de ejecución sin errores.
Base de Datos PostgreSQL	Scripts SQL de: • Creación de tablas (schema). • Datos de prueba (ej: 10 clientes, 5 servicios registrados). Diagrama ER (Entity-Relationship) generado con herramientas como dbdiagram.io.
API Documentada (Swagger/OpenAPI)	Endpoint /api/docs/ con: • Listado de rutas (ej: GET /api/servicios/). • Ejemplos de requests/responses. Colección Postman exportable (JSON).
Vídeo Demostrativo (3-5 min)	Demo del sitio web (público y panel administrativo). Flujo completo: desde solicitud de servicio hasta generación de reporte. Explicación técnica breve (ej: cómo se dockeriza el proyecto). Plataforma: Subido a YouTube (enlace privado) o Google Drive.
Documentación Técnica	Manual de Usuario (PDF): • Guía para técnicos (cómo registrar servicios). • Instrucciones para clientes (cómo solicitar soporte). Manual de Despliegue (Azure + Docker): • Pasos para replicar el entorno en producción.

Formatos Específicos de Evidencia

Evidencia	Formato	Herramienta Utilizada
Código fuente	.py, .html, .sql	Django, Bootstrap, PostgreSQL
Docker	.yml, .Dockerfile	Docker, Azure Container Registry
Base de datos.sql	.png (ER)	pgAdmin, dbdiagram.io

API	.json (Postman)	Swagger, Postman
Reportes	.pdf, .csv	ReportLab, Pandas



1. DESARROLLAR LA SOLUCIÓN DE SOFTWARE DE ACUERDO CON EL DISEÑO Y METODOLOGÍAS DE DESARROLLO (Código: 220501096)

Relación con las actividades:	• Desarrollo del sitio web institucional con Django y Bootstrap 5. • Implementación del sistema de gestión de servicios técnicos (formularios digitales, autenticación, roles). • Creación de la API REST para integraciones.
Saberes aplicados:	• Metodologías ágiles (SCRUM). • Patrones de diseño MVC (Django). • Frameworks frontend (Bootstrap).

2. IMPLEMENTAR LA SOLUCIÓN DE SOFTWARE DE ACUERDO CON LOS REQUISITOS DE OPERACIÓN Y MODELOS DE REFERENCIA (Código: 220501097)

Relación con las actividades:	• Dockerización de la aplicación (contenedores para backend, frontend y PostgreSQL). • Despliegue en Azure (App Services, CI/CD con GitHub Actions). • Configuración de entornos (desarrollo, producción).
Saberes aplicados:	• DevOps básico (gestión de contenedores, despliegue en la nube). • Seguridad en despliegues (variables de entorno, HTTPS).

3. CONTROLAR LA CALIDAD DEL SERVICIO DE SOFTWARE DE ACUERDO CON LOS ESTÁNDARES TÉCNICOS (Código: 220501098) (Opcional adicional)

Relación con las actividades:	• Pruebas unitarias y de integración (Django pytest). • Generación de reportes sin errores (PDF/CSV). • Optimización del rendimiento (caching, queries SQL eficientes).

Resultados de Aprendizaje Asociados

Para 220501096:

"Diseña interfaces de usuario responsivas usando frameworks frontend (Bootstrap 5)."

"Aplica metodologías ágiles en el desarrollo de software."

Para 220501097:

"Implementa soluciones en la nube usando servicios como Azure App Services."

"Configura pipelines CI/CD para despliegues automatizados."