FASES DE DESARROLLO - INE STRUCTUM

PROYECTO COMPLETO ESTRUCTURADO PARA POWER PLANNER

INFORMACIÓN GENERAL DEL PROYECTO

Nombre del Proyecto: INE STRUCTUM - Software de Verificación Estructural para STAAD.Pro

Objetivo: Desarrollar una aplicación Windows que automatice verificaciones de servicio (deflexiones, derivas, desplazamientos) y genere reportes ejecutivos a partir de modelos STAAD.Pro, con gestión de proyectos, productos y combinaciones de carga según códigos ASCE 7-22 y Eurocode.

Tecnologías:

- Frontend: Python (tkinter/CustomTkinter) o C# WPF
- Backend: Python con OpenSTAADPy
- Base de datos: SQLite o PostgreSQL
- Reportes: ReportLab/openpyxl o Word automation

Logo oficial: ✅ Estructura 3D isométrica con gradiente naranja-azul

FASE 1: PLANIFICACIÓN Y ARQUITECTURA (2 semanas)

1.1 Definición de Requerimientos Finales

- ✅ Revisar documento completo de especificaciones funcionales
- ✅ Validar todas las reglas de negocio acordadas
- ✅ Confirmar flujo de módulos: Proyecto → Producto → Verificación
- ✅ Documentar casos de uso detallados

Entregables:

- Documento de Especificación Funcional (SRS)
- Diagrama de flujo de navegación entre módulos
- Casos de uso documentados con screenshots

1.2 Diseño de Base de Datos

- Diseñar esquema completo de base de datos
- Definir relaciones entre tablas (Proyecto, Producto, Casos de Carga, Combinaciones)
- Crear script de inicialización de BD
- Definir estrategia de versionado de datos

Entregables:

- Diagrama Entidad-Relación (ERD)
- Script SQL de creación de tablas
- Diccionario de datos

1.3 Arquitectura de Software

- Definir patrón arquitectónico (MVC/MVVM)
- Diseñar estructura de carpetas del proyecto
- Establecer convenciones de código
- Configurar control de versiones (Git)

Entregables:

- Documento de Arquitectura de Software
- Repositorio Git inicializado
- Plantilla de estructura de carpetas

1.4 Diseño de UI/UX

- Crear wireframes de todas las pantallas
- Definir paleta de colores corporativa (Inelectra)
- Diseñar sistema de iconografía
- Crear mockups de alta fidelidad

Entregables:

- Wireframes en Figma/Adobe XD
- Guía de estilo visual
- Mockups interactivos

FASE 2: CONFIGURACIÓN DEL ENTORNO (1 semana)

2.1 Configuración de Desarrollo

- Instalar y configurar IDEs (VS Code/Visual Studio)
- Configurar entorno virtual Python
- Instalar OpenSTAADPy y dependencias
- Configurar STAAD.Pro Connect Edition para testing

Entregables:

- Documento de instalación de entorno
- Requirements.txt con todas las dependencias
- Script de configuración automática

2.2 Estructura del Proyecto

- Crear estructura de carpetas definitiva
- Configurar módulos Python base
- Implementar sistema de logging
- Configurar manejo de excepciones global

Entregables:

- Proyecto base funcional
- Sistema de logs operativo
- Manejador de errores implementado

2.3 Integración con STAAD.Pro

- Desarrollar módulo de conexión con OpenSTAAD
- Crear funciones de lectura de modelo
- Implementar extracción de casos de carga
- Desarrollar pruebas de conexión

Entregables:

- Módulo `staad_connector.py` funcional
- Suite de pruebas unitarias
- Documentación de API interna

□ FASE 3: MÓDULO PROYECTOS (3 semanas)

3.1 Gestión de Proyectos Base

- Crear interfaz de listado de proyectos
- Implementar creación de nuevo proyecto
- Desarrollar edición de proyectos existentes
- Implementar eliminación con confirmación

Entregables:

- Pantalla de gestión de proyectos funcional
- CRUD completo de proyectos
- Validaciones implementadas

3.2 Configuración de Códigos de Diseño

- Implementar selector ASCE 7-22 vs Eurocode
- Desarrollar interfaz de parámetros sísmicos (ASCE)
- Crear interfaz de parámetros Eurocode
- Implementar carga automática de valores típicos

Entregables:

- Selector de código funcional
- Formularios de parámetros sísmicos
- Base de datos de valores por defecto

3.3 Identificación de Casos de Carga

- Desarrollar interfaz de selección de sismo (X, Y, Z)
- Implementar selector de casos de viento (±X, ±Z)
- Crear lógica de validación de casos únicos
- Implementar restricción de edición desde Producto

Entregables:

- Interfaz de identificación de cargas
- Lógica de bloqueo de casos críticos
- Validaciones de unicidad

3.4 Agrupación de Casos por Tipo

- Crear interfaz de agrupación de Dead loads
- Implementar agrupación de Live loads
- Desarrollar selector de tratamiento (suma/alternativas)
- Implementar agrupación de Temperature, Fluid, etc.

Entregables:

- Sistema de agrupación completo
- Interfaz de checkboxes y opciones
- Resumen de agrupaciones definidas

3.5 Tabla de Límites de Deflexión

- Crear Tabla 1 (3 columnas con selector tipo límite)
- Implementar Tabla 2 (1 columna con selector tipo límite)
- Desarrollar toggle L/denom vs mm absoluto
- Implementar agregar/editar/eliminar filas

Entregables:

- Tablas de deflexiones funcionales
- Selector dinámico de tipo límite
- Sistema de validación de valores

3.6 Configuración de Derivas Sísmicas

- Implementar configuración según ASCE (% con Cd)
- Crear configuración según Eurocode (SLS y ULS)
- Desarrollar validación de límites

Entregables:

- Interfaz de configuración de derivas
- Lógica condicional por código
- Valores por defecto cargados

□ FASE 4: MÓDULO PRODUCTOS (4 semanas)

4.1 Conexión con Archivo STAAD

- Desarrollar selector de archivo .std
- Implementar lectura de modelo completo
- Crear visualización de conexión activa
- Implementar desconexión manual

Entregables:

- Selector de archivo funcional
- Indicador visual de archivo conectado
- Sistema de reconexión automática

4.2 Importación Automática de Casos de Carga

- Leer todos los casos de carga del modelo
- Clasificar automáticamente por tipo
- Implementar detección de tipo (Dead, Live, Wind, etc.)
- Crear tabla de casos importados

Entregables:

- Algoritmo de clasificación automática
- Tabla de casos con tipo detectado
- Sistema de corrección manual de tipos

4.3 Sistema de Clasificación de Grupos

- Leer grupos definidos en STAAD
- Implementar clasificación automática por prefijo
- Crear lógica de detección de tipo estructural
- Desarrollar override manual de clasificación

Entregables:

- Tabla de grupos clasificados
- Algoritmo de detección por nombre
- Interfaz de reclasificación

4.4 Generación Automática de Combinaciones

- Implementar generador según ASCE 7-22
- Desarrollar generador según Eurocode
- Crear combinaciones ULS (resistencia)
- Generar combinaciones SLS (servicio)

- Implementar combinaciones de viento direccionales
- Desarrollar regla 100-30-30% para sismo
- Crear combinaciones con sismo vertical
- Implementar combinaciones para conexiones (con Ω_0)

Entregables:

- Motor de generación de combinaciones
- Lista completa de combos generadas
- Nomenclatura clara y descriptiva

4.5 Clasificación Automática de Combinaciones

- Etiquetar combos como ULS/SLS/Viento/Sismo/Conexiones
- Implementar filtros por categoría
- Crear interfaz de visualización clasificada

Entregables:

- Sistema de etiquetado automático
- Filtros funcionales
- Visualización por categorías

4.6 Edición Manual de Combinaciones

- Permitir editar factores de carga
- Implementar agregar/eliminar combos personalizadas
- Crear validación de sintaxis de combos
- Desarrollar preview de combo antes de guardar

Entregables:

- Editor de combinaciones
- Validador de sintaxis
- Sistema de guardado seguro

□ FASE 5: MÓDULO VERIFICACIÓN (5 semanas)

5.1 Selección de Combinaciones por Tipo

- Crear interfaz de selección para deflexiones (por columna)
- Implementar selector de combos para desplazamientos viento
- Desarrollar selector de combos para derivas sismo
- Crear selector de combos para diseño de resistencia

- Implementar selector de combos para conexiones

Entregables:

- Interfaz de selección completa
- Checkboxes por categoría
- Sistema de aplicar a todos

5.2 Verificación de Deflexiones

- Implementar lectura de deflexiones desde STAAD
- Desarrollar algoritmo de verificación Tabla 1 (3 columnas)
- Crear algoritmo de verificación Tabla 2 (1 columna)
- Implementar cálculo de ratios
- Desarrollar identificación de elementos que no cumplen

Entregables:

- Módulo de verificación de deflexiones
- Tabla de resultados con ratios
- Marcado visual de incumplimientos (rojo)

5.3 Verificación de Desplazamientos por Viento

- Leer desplazamientos laterales de pisos
- Implementar cálculo de drift por piso
- Verificar contra límite h/400 (o configurado)
- Generar tabla de resultados por nivel y dirección

Entregables:

- Módulo de desplazamientos por viento
- Tabla de drifts por piso
- Identificación de pisos críticos

5.4 Verificación de Derivas Sísmicas

- Leer desplazamientos sísmicos por nodos
- Implementar amplificación con Cd (ASCE) o v (EC8)
- Calcular deriva entre pisos
- Verificar contra límite configurado (2%, 0.5%, etc.)
- Generar tabla de derivas amplificadas

Entregables:

- Módulo de derivas sísmicas

- Cálculo con factor de amplificación
- Tabla de derivas por piso y combo

5.5 Extracción de Ratios de Diseño

- Leer ratios DCR de diseño de STAAD
- Extraer máximos por grupo
- Identificar elementos críticos
- Crear tabla de resumen de diseño

Entregables:

- Extractor de ratios de diseño
- Tabla de elementos críticos
- Visualización de capacidad utilizada

5.6 Interfaz de Resultados

- Crear tabla maestra de resultados
- Implementar filtros por grupo/combo/cumplimiento
- Desarrollar exportación a Excel
- Crear visualización gráfica de resultados

Entregables:

- Interfaz de resultados completa
- Sistema de filtros avanzados
- Exportador a Excel funcional

FASE 6: GENERACIÓN DE REPORTES (3 semanas)

6.1 Plantilla de Reporte Ejecutivo

- Diseñar plantilla profesional en Word/PDF
- Incluir portada con logo INE STRUCTUM
- Crear secciones estructuradas
- Implementar tablas formateadas

Entregables:

- Plantilla de reporte .docx
- Diseño visual corporativo
- Estructura modular

6.2 Generador Automático de Reportes

- Implementar llenado automático de datos
- Desarrollar generación de tablas de resultados
- Crear inserción de gráficos y diagramas
- Implementar numeración automática de secciones

Entregables:

- Motor de generación de reportes
- Reportes PDF/Word funcionales
- Sistema de plantillas personalizables

6.3 Contenido del Reporte

- Incluir datos del proyecto
- Agregar resumen de combinaciones usadas
- Insertar tablas de verificación de deflexiones
- Agregar tablas de derivas sísmicas
- Incluir tabla de desplazamientos por viento
- Insertar ratios de diseño críticos
- Crear sección de conclusiones automáticas

Entregables:

- Reporte completo con todas las secciones
- Conclusiones auto-generadas
- Formato profesional

6.4 Exportación y Distribución

- Implementar exportación a PDF
- Desarrollar exportación a Word editable
- Crear exportación a Excel de datos crudos
- Implementar envío por email (opcional)

Entregables:

- Múltiples formatos de exportación
- Sistema de guardado automático
- Integración con email (si aplica)

□ FASE 7: TESTING Y VALIDACIÓN (2 semanas)

7.1 Pruebas Unitarias

- Crear suite de pruebas para cada módulo
- Implementar pruebas de funciones críticas
- Desarrollar pruebas de validación de datos
- Alcanzar 80%+ cobertura de código

Entregables:

- Suite de pruebas unitarias completa
- Reporte de cobertura de código
- Documentación de casos de prueba

7.2 Pruebas de Integración

- Probar flujo completo Proyecto→Producto→Verificación
- Validar generación de combinaciones
- Verificar cálculos contra casos manuales
- Probar integración con STAAD.Pro

Entregables:

- Suite de pruebas de integración
- Casos de prueba documentados
- Validación matemática de cálculos

7.3 Pruebas con Modelos Reales

- Probar con 5+ modelos STAAD reales
- Validar resultados contra cálculos manuales
- Identificar casos edge
- Corregir bugs encontrados

Entregables:

- Reporte de pruebas con modelos reales
- Lista de bugs corregidos
- Validación de precisión

7.4 Pruebas de Usuario (UAT)

- Realizar sesiones con ingenieros estructurales
- Recopilar feedback de usabilidad
- Identificar mejoras de UX
- Implementar correcciones críticas

Entregables:

- Reporte de UAT
- Lista de mejoras implementadas
- Feedback documentado

FASE 8: DOCUMENTACIÓN (2 semanas)

8.1 Manual de Usuario

- Crear guía paso a paso
- Incluir capturas de pantalla
- Documentar todos los flujos de trabajo
- Crear sección de troubleshooting

Entregables:

- Manual de usuario en PDF
- Versión digital interactiva
- Videos tutoriales (opcional)

8.2 Documentación Técnica

- Documentar arquitectura del software
- Crear guía de instalación
- Documentar API interna
- Crear guía de mantenimiento

Entregables:

- Documentación técnica completa
- Diagramas de arquitectura
- Guía de desarrollo futuro

8.3 Documentación de Códigos

- Documentar interpretación de ASCE 7-22
- Crear guía de Eurocode implementado
- Documentar reglas de combinaciones
- Crear tabla de referencia de factores

Entregables:

- Guía de códigos implementados
- Referencia técnica de normativas
- Justificación de fórmulas

FASE 9: DEPLOYMENT Y DISTRIBUCIÓN (1 semana)

9.1 Empaquetado de Aplicación

- Crear instalador Windows (.exe)
- Incluir todas las dependencias
- Configurar instalación de BD
- Crear desinstalador limpio

Entregables:

- Instalador funcional
- Script de instalación silenciosa
- Desinstalador

9.2 Configuración de Licenciamiento

- Implementar sistema de licencias (si aplica)
- Crear validación de activación
- Desarrollar sistema de trial (opcional)

Entregables:

- Sistema de licencias funcional
- Documentación de activación

9.3 Distribución Inicial

- Preparar paquete de instalación
- Crear guía de instalación rápida
- Configurar servidor de actualizaciones (si aplica)

Entregables:

- Paquete de distribución completo
- Quick Start Guide
- Sistema de updates

FASE 10: CAPACITACIÓN Y SOPORTE (1 semana)

10.1 Capacitación de Usuarios

- Realizar sesiones de entrenamiento
- Crear material de capacitación
- Grabar videos demostrativos

Entregables:

- Sesiones de capacitación realizadas
- Material de entrenamiento
- Videos tutoriales

10.2 Configuración de Soporte

- Establecer canal de soporte técnico
- Crear sistema de tickets
- Definir SLA de respuesta

Entregables:

- Sistema de soporte operativo
- Documentación de soporte
- SLA definido

RESUMEN DE FASES Y DURACIÓN

Fase	Descripción	Duración	Dependencias
1	Planificación y Arquitectura	2 semanas	-
2	Configuración del Entorno	1 semana	Fase 1

Fase	Descripción	Duración	Dependencias
3	Módulo Proyectos	3 semanas	Fase 2
4	Módulo Productos	4 semanas	Fase 3
5	Módulo Verificación	5 semanas	Fase 4
6	Generación de Reportes	3 semanas	Fase 5
7	Testing y Validación	2 semanas	Fase 6
8	Documentación	2 semanas	Fase 7
9	Deployment	1 semana	Fase 8
10	Capacitación y Soporte	1 semana	Fase 9

DURACIÓN TOTAL: 24 semanas (6 meses)

HITOS PRINCIPALES

1. ✅ **H1:** Arquitectura aprobada (Fin Fase 1)
2. ✅ **H2:** Módulo Proyectos funcional (Fin Fase 3)
3. ✅ **H3:** Generación automática de combinaciones (Fin Fase 4)
4. ✅ **H4:** Verificaciones completas operativas (Fin Fase 5)
5. ✅ **H5:** Reportes generables (Fin Fase 6)
6. ✅ **H6:** Software validado y listo (Fin Fase 7)
7. ✅ **H7:** Primera versión distribuida (Fin Fase 9)

¿Listo para plasmar esto en Power Planner?