

## Instrucciones para la Última Entrega Documental del Proyecto

**Asignatura:** Estándares de Programación

**Docente:** Ing. Rubén Escobar Ortegón

**Universidad:** Universidad Tecnológica de El Salvador

**Ciclo:** 02-2025

**Modalidad:** Entrega digital.

### Objetivo

Esta entrega documental es la culminación del desarrollo de su aplicación web y debe consolidar todos los aspectos trabajados durante el ciclo (requerimientos, diseño, pruebas e implementación). El documento servirá como base para la **defensa final** del proyecto, donde presentarán y justificarán su trabajo. La entrega debe demostrar la aplicación de **técnicas de ingeniería de software**, incluyendo los estándares aprendidos en las Unidades de Aprendizaje 1 a 4, y preparar al equipo para responder preguntas técnicas y funcionales durante la defensa.

### Requisitos Generales

#### 1. Formato del documento:

- Archivo en formato **WORD**.
- Nombre del archivo: Equipo\_[Número o Nombre]\_EntregaFinal\_EPRO.pdf (e.g., Equipo3\_EntregaFinal\_EPRO.pdf).
- Portada con:
  - Nombre de la universidad: **Universidad Tecnológica de El Salvador**.
  - Asignatura: **Estándares de Programación**.
  - Docente: **Ing. Rubén Escobar Ortegón**.
  - Sección
  - Nombre o número del equipo.
  - Lista de integrantes (Apellidos, Nombres, Carné, en orden alfabético APELLIDO PRIMERO).
  - Fecha de entrega

- Tipografía: Times New Roman, tamaño 12, interlineado 1.5.
  - Máximo **20 páginas** (sin contar anexos, si los hay).
2. **Estructura del documento:** El documento debe incluir las siguientes secciones, alineadas con las competencias del curso:

## **Estructura y Contenido del Documento**

### **1. Introducción (1 página)**

- Describa brevemente el propósito del proyecto (qué problema resuelve la aplicación web).
- Mencione el contexto organizacional o el escenario que motivó el desarrollo (Unidad 1: Ingeniería de Requerimientos).
- Resuma los objetivos del sistema y el alcance del proyecto.

### **2. Especificación de Requerimientos (2-3 páginas)**

- Liste los **requerimientos funcionales** (e.g., "El sistema debe permitir registrar usuarios") y **no funcionales** (e.g., "El sistema debe responder en menos de 2 segundos") desarrollados en la Unidad 1.
- Incluya al menos **5 requerimientos funcionales y 3 no funcionales**, especificados claramente (pueden usar una tabla).
- Explique cómo recopilaron los requerimientos (e.g., entrevistas, análisis de sistemas legados) y cómo los validaron con los stakeholders o usuarios simulados.
- Si aplicaron conceptos como el modelo **CMMI** o ingeniería inversa, menciónenlos brevemente.

### **3. Diseño del Software (3-4 páginas)**

- Describa la arquitectura del sistema, utilizando conceptos de la Unidad 2: Ingeniería de Diseño (e.g., diseño estructurado, orientado a objetos, MVC).
- Incluya un **diagrama simple** (e.g., diagrama de arquitectura, flujo de datos, o modelo conceptual) que muestre cómo se organizaron los componentes (base de datos, interfaz, lógica, seguridad).

- Explique cómo aplicaron principios de diseño como modularidad, reusabilidad o usabilidad (e.g., "Diseñamos formularios intuitivos para mejorar la experiencia del usuario").
- Mencione al menos un **patrón de diseño** o criterio de usabilidad aplicado y cómo impactó el sistema.

#### 4. Implementación de Funcionalidades (4-5 páginas)

- Detalle las **funcionalidades implementadas**.
- Para cada funcionalidad:
  - Describa qué hace y cómo beneficia al usuario.
  - Explique cómo se diseñó (Unidad 2) y cómo se integró con otros componentes (e.g., conexión con la base de datos).
  - Incluya **capturas de pantalla** de la interfaz o resultados (máximo 2 por funcionalidad).
- Mencione las tecnologías utilizadas (e.g., HTML, CSS, JavaScript, PHP, MySQL) y justifique su elección.
- Si hubo desafíos durante la implementación (e.g., problemas de compatibilidad), descríbanlos y cómo los resolvieron.

#### 5. Plan de Pruebas y Resultados (3-4 páginas)

- Basado en la Unidad 3: Ingeniería de Pruebas, presente un **plan de pruebas simplificado** que incluya:
  - **Tipos de pruebas realizadas:** Pruebas de unidad y, si aplica, pruebas de integración.
  - **Técnicas utilizadas:** Pruebas de caja negra con datos válidos e inválidos.
  - **Casos de prueba:** Al menos **2 casos de prueba por funcionalidad** (uno exitoso y uno con error). Use una tabla como la siguiente:
  - **Resultados:** Resuma los resultados de las pruebas (e.g., "El 90% de los casos fueron exitosos; un caso falló y se corrigió").

- Incluya **evidencia** (capturas de pantalla de mensajes de éxito, errores controlados, o logs).
- Explique cómo las pruebas aseguraron la calidad del sistema y cómo manejaron errores detectados.

## 6. Plan de Implementación (2-3 páginas)

- Basado en la Unidad 4: Proceso de Implementación, describa un plan para desplegar la aplicación web en un entorno real (aunque sea hipotético). Incluya:
  - **Estrategia de implementación:** Cómo se instalaría el sistema (e.g., en un servidor web, configuración de la base de datos).
  - **Roles:** Quiénes estarían involucrados (e.g., administrador de sistemas, desarrolladores).
  - **Plan de capacitación:** Cómo se entrenaría a los usuarios finales (e.g., manuales, sesiones de demostración).
  - **Presupuesto estimado:** Costos aproximados de hosting, licencias, o hardware.
  - **Criterios de aceptación:** Cómo se determinaría que el sistema está listo (e.g., pruebas finales exitosas, aprobación del cliente).
- Mencione cualquier proceso de documentación entregado al cliente (e.g., manual de usuario, guía técnica).

## 7. Conclusiones y Reflexiones (1-2 páginas)

- Resuma los logros del proyecto (e.g., funcionalidades completadas, cumplimiento de requerimientos).
- Reflexione sobre cómo aplicaron los estándares de ingeniería de software y qué aprendieron.
- Identifique **lecciones aprendidas** (e.g., importancia de pruebas tempranas, desafíos de diseño) y cómo mejorarían en futuros proyectos.
- Explique cómo el proyecto refleja los valores institucionales de la UTEC (**solidaridad y innovación permanente**).

## 8. Anexos (opcional)

- Incluya diagramas adicionales, capturas de pantalla, o fragmentos de código relevantes (máximo 5 páginas).
- Asegúrense de que los anexos estén referenciados en el texto principal.

### **Instrucciones para la Entrega**

- **Plataforma:** Suban el PDF antes de la fecha límite.
- **Confirmación:** Verifiquen que los archivos se hayan cargado correctamente y sean accesibles.
- **Integridad del equipo:** Todos los integrantes deben participar en la elaboración del documento y estar preparados para explicar cualquier sección durante la defensa.

### **Preparación para la Defensa Final**

La defensa será una presentación oral donde explicarán el proyecto y responderán preguntas del docente. Para prepararse:

- **Revise el documento:** Asegúrense de que todos los integrantes entiendan cada sección, especialmente el diseño, las pruebas y la implementación.
- **Practique la presentación:**
  - Prepare una presentación de **20-25 minutos** (en PowerPoint o similar) que resuma el proyecto (requerimientos, diseño, funcionalidades, pruebas, implementación).
  - Destaque las funcionalidades clave con una demostración en vivo de su sistema.
  - Asigne roles para que cada integrante hable sobre una sección específica.
- **Anticipe preguntas:**
  - Sobre diseño: ¿Por qué eligieron esta arquitectura? ¿Cómo garantizaron la usabilidad?
  - Sobre pruebas: ¿Qué casos de prueba fallaron y cómo los corrigieron?
  - Sobre implementación: ¿Cómo planearon el despliegue? ¿Qué harían si el sistema falla en producción?

- **Material de apoyo:** Lleven una copia digital del documento y el código fuente a la defensa.

### Criterios de Evaluación

Sección	Ponderación	Criterios
<b>Introducción y Requerimientos</b>	15%	Claridad en el propósito, especificación completa y validación de requerimientos.
<b>Diseño del Software</b>	20%	Arquitectura clara, uso de patrones de diseño, aplicación de usabilidad.
<b>Implementación de Funcionalidades</b>	25%	Funcionalidades completas, evidencia visual, tecnologías justificadas.
<b>Plan de Pruebas y Resultados</b>	20%	Casos de prueba claros, resultados documentados, manejo de errores.
<b>Plan de Implementación</b>	15%	Estrategia realista, roles definidos, criterios de aceptación claros.
<b>Conclusiones y Reflexiones</b>	5%	Resumen de logros, lecciones aprendidas, conexión con valores institucionales.

### Recomendaciones

- **Coherencia:** Asegúrense de que el documento sea consistente con el video de avance y los entregables previos. Por ejemplo, las funcionalidades descritas deben incluir las mostradas en el video.
- **Claridad:** Escriban de forma clara y eviten tecnicismos innecesarios. Usen ejemplos prácticos para explicar el diseño y las pruebas.
- **Colaboración:** Dividan las tareas entre los integrantes (e.g., uno redacta requerimientos, otro pruebas) y revisen el documento en equipo para evitar errores.
- **Pruebas finales:** Antes de entregar, ejecuten todas las funcionalidades y pruebas nuevamente para confirmar que el sistema está operativo.

- **Valores institucionales:** Destaquen cómo el trabajo en equipo reflejó **solidaridad** (e.g., apoyo mutuo) y cómo el proyecto fue **innovador** (e.g., soluciones creativas).