

1. Portada

- **Título del proyecto**
- **Nombre del alumno**
- **Curso y año académico**
- **Nombre del tutor o profesor**

2. Índice

- Un listado de secciones y subsecciones con sus correspondientes números de página.

3. Resumen Ejecutivo

- Una descripción breve del proyecto, sus objetivos, funcionalidades principales y tecnologías utilizadas.

4. Introducción

- Explicación del problema o necesidad que el proyecto aborda.
- Objetivos del proyecto.
- Justificación de la elección del tema o problema a resolver.

5. Análisis y Especificaciones

- **Análisis de requisitos:** Requisitos funcionales y no funcionales.
- **Especificaciones técnicas:** Herramientas y tecnologías utilizadas (lenguajes de programación, frameworks, bases de datos, etc.).
- **Diagrama de casos de uso:** Representación gráfica de las funcionalidades del sistema desde el punto de vista del usuario.
- **Diagrama de clases:** (si aplica) Descripción de la estructura de datos y relaciones entre las clases.

6. Diseño de la Aplicación

- **Arquitectura de la aplicación:** Explicación de la arquitectura adoptada (MVC, Microservicios, etc.).
- **Diseño de la base de datos:** Diagrama entidad-relación, normalización, etc.
- **Wireframes o prototipos:** Bocetos de las interfaces de usuario.
- **Flujo de navegación:** Explicación de cómo se mueven los usuarios a través de la aplicación.

7. Desarrollo

- **Implementación:** Descripción detallada del proceso de desarrollo.
- **Código fuente:** Explicación de fragmentos de código clave, estructura de carpetas, módulos principales, etc.
- **Pruebas:** Estrategia de pruebas aplicada, incluyendo pruebas unitarias, de integración y aceptación.
- **Integración continua (si aplica):** Descripción del pipeline de CI/CD.

8. Manual de Usuario

- Instrucciones para la instalación, configuración y uso de la aplicación para usuarios finales.
- Capturas de pantalla de la interfaz.

9. Conclusiones y Futuras Mejoras

- Evaluación del cumplimiento de los objetivos.
- Retos encontrados y soluciones implementadas.
- Sugerencias para posibles mejoras o extensiones futuras del proyecto.

10. Bibliografía

- Listado de fuentes y referencias utilizadas, siguiendo un formato estándar (APA, IEEE, etc.).

11. Anexos (si es necesario)

- Documentos adicionales, como especificaciones técnicas detalladas, listas de requisitos, o tablas de datos.

Formato de Entrega

1. Documento Escrito:

- Generalmente, en formato PDF.
- Estructurado de manera coherente, con una tipografía legible y bien organizado.

2. Código Fuente:

- Incluido en un repositorio (GitHub, GitLab, etc.), con enlaces adecuados.
- El repositorio debe contener un `README.md` con instrucciones de instalación y uso.

3. Presentación:

- En algunos casos, se pide una presentación (PowerPoint, PDF, etc.) que resuma el proyecto y que se utiliza durante la defensa.

4. Otros:

- Dependiendo de las exigencias, puede ser necesario entregar archivos de configuración, bases de datos, o documentación adicional como diagramas UML, etc.

Es recomendable revisar las guías específicas proporcionadas por tu institución o profesor para asegurarte de cumplir con todos los requisitos.

Algunas alternativas incluyen:

1. Carpeta ZIP o RAR

- Comprimir todos los archivos del proyecto (código fuente, documentación, etc.) en un archivo ZIP o RAR.
- Asegurarse de que la estructura de carpetas sea clara y que incluya un archivo `README.txt` o `README.md` con instrucciones para ejecutar el proyecto.

2. Plataformas de Almacenamiento en la Nube

- **Google Drive, OneDrive, Dropbox:** Subir el proyecto a una carpeta en la nube y compartir el enlace de acceso. Asegúrate de que el enlace tenga los permisos de acceso adecuados.
- La carpeta debe estar bien organizada y contener todos los archivos necesarios, incluida la documentación.

3. Plataforma de Entrega de la Institución

- Algunas instituciones tienen sus propias plataformas de gestión de aprendizaje (como Moodle, Blackboard, etc.) donde se puede subir el proyecto directamente.
- Siguiendo las instrucciones de la plataforma, normalmente se puede adjuntar el proyecto como un archivo comprimido o subirlo en varias partes.

4. USB o Disco Duro Externo

- En casos donde se requiere la entrega física, puedes copiar el proyecto en un USB o un disco duro externo y entregarlo en mano.
- Asegúrate de etiquetar correctamente el dispositivo con tu nombre y el título del proyecto.

5. Correo Electrónico

- Enviar el proyecto como un archivo adjunto en un correo electrónico.
- Asegúrate de no exceder los límites de tamaño del correo y comprimir los archivos si es necesario.
- En el cuerpo del correo, describe brevemente el contenido y proporciona cualquier información adicional que sea relevante para la instalación o revisión del proyecto.

6. Otras Plataformas de Control de Versiones

- **GitLab, Bitbucket, SourceForge:** Si GitHub no es una opción, otras plataformas de control de versiones pueden ser aceptables, dependiendo de la preferencia del profesor.

7. Repositorio en el Servidor de la Escuela

- Algunas instituciones tienen sus propios servidores de control de versiones o almacenamiento donde puedes subir el proyecto.

Es importante comunicarte con tu tutor o profesor para confirmar cuál de estas alternativas es aceptable y seguir cualquier instrucción adicional que te den. Cada institución puede tener sus propios requisitos y limitaciones en cuanto a la entrega de proyectos.

Formas más comunes de entregar una base de datos:

1. Archivo de Exportación SQL

- **Descripción:** Exporta la base de datos completa a un archivo `.sql` usando herramientas como phpMyAdmin, MySQL Workbench, o comandos de terminal.
- **Instrucciones:**
 1. Accede a tu base de datos y selecciona la opción de exportación.
 2. Elige el formato SQL (normalmente se utiliza `mysqldump` para MySQL).
 3. Guarda el archivo `.sql` exportado.
 4. Incluye este archivo en la entrega del proyecto (en la carpeta comprimida, repositorio o nube).
- **Ventaja:** Este archivo permitirá que el evaluador importe fácilmente la base de datos en su propio entorno.

2. Scripts de Creación y Población

- **Descripción:** Proporciona scripts separados que contengan las instrucciones SQL para crear las tablas y para insertar los datos.
- **Instrucciones:**
 1. Crea un archivo `create_tables.sql` que incluya todas las sentencias `CREATE TABLE`.
 2. Crea un archivo `insert_data.sql` con las sentencias `INSERT INTO` para poblar las tablas con datos iniciales.
 3. Ambos archivos deben ser incluidos en la entrega.
- **Ventaja:** Facilita la personalización y ajustes en las estructuras o datos.

3. Copia de la Carpeta del Motor de Base de Datos

- **Descripción:** En algunos casos, especialmente con motores como SQLite, la base de datos puede ser entregada como un archivo `.db` o la carpeta completa del motor.
- **Instrucciones:**

1. Localiza el archivo o carpeta que contiene los datos de la base de datos.
 2. Copia el archivo o carpeta en tu proyecto y asegúrate de incluirlo en la entrega.
- **Ventaja:** No requiere configuración adicional, solo se debe asegurar que el evaluador tenga el entorno adecuado para abrirlo.

4. Docker Compose o Imágenes de Docker

- **Descripción:** Si estás utilizando contenedores Docker para gestionar la base de datos, puedes entregar un archivo `docker-compose.yml` que defina el servicio de la base de datos.
- **Instrucciones:**
 1. Asegúrate de que el archivo `docker-compose.yml` esté bien configurado para lanzar la base de datos con todos los datos necesarios.
 2. Incluye instrucciones claras sobre cómo levantar los contenedores.
- **Ventaja:** Proporciona un entorno replicable y garantiza que la base de datos se configure correctamente.

5. URL de Acceso a un Servidor Remoto

- **Descripción:** Si la base de datos está alojada en un servidor remoto (por ejemplo, en un hosting de MySQL), puedes proporcionar la URL de acceso junto con las credenciales necesarias.
- **Instrucciones:**
 1. Proporciona una URL o IP del servidor.
 2. Incluye un archivo con las credenciales necesarias (nombre de usuario y contraseña).
- **Ventaja:** No requiere que el evaluador importe la base de datos, pero puede depender de la conectividad y seguridad.

6. Enlace de Restauración Automática

- **Descripción:** Algunos frameworks permiten crear scripts que restauren la base de datos al estado original al ejecutar un comando.
- **Instrucciones:**
 1. Proporciona un script o enlace que, al ser ejecutado, restaure automáticamente la base de datos desde un archivo de respaldo.
- **Ventaja:** Simplifica la restauración, haciéndola parte del flujo de trabajo.

Recomendaciones Adicionales:

- **Documentación:** Incluye un documento o sección en la documentación del proyecto con instrucciones claras sobre cómo importar o configurar la base de datos.
- **Pruebas:** Verifica que la base de datos sea fácilmente importable o accesible desde otro entorno antes de la entrega.

Esquema de los pasos iniciales que deberías seguir:

1. Definición de Requisitos

- **Requisitos Funcionales:** Especifica qué funciones y características debe tener la aplicación (por ejemplo, autenticación de usuarios, gestión de productos, etc.).
- **Requisitos No Funcionales:** Define aspectos como rendimiento, seguridad, escalabilidad, y compatibilidad.

2. Investigación y Selección de Tecnologías

- **Frameworks y Lenguajes:** Decide qué lenguajes de programación, frameworks y herramientas utilizarás (por ejemplo, HTML/CSS/JavaScript para el frontend, Django/Laravel para el backend).
- **Bases de Datos:** Selecciona la base de datos adecuada (MySQL, PostgreSQL, MongoDB, etc.).
- **Herramientas de Desarrollo:** Configura tu entorno de desarrollo integrado (IDE), repositorio de control de versiones (Git), y cualquier otra herramienta que necesites.

3. Planificación del Proyecto

- **Tareas y Cronograma:** Desglosa el proyecto en tareas más pequeñas y crea un cronograma o roadmap que guíe el desarrollo.
- **Prioridades:** Identifica qué funcionalidades son críticas y deben implementarse primero (MVP - Producto Mínimo Viable).
- **Asignación de Recursos:** Si trabajas en equipo, asigna tareas a cada miembro según sus habilidades.

4. Diseño de la Arquitectura

- **Diagrama de Arquitectura:** Dibuja un diagrama que muestre la estructura general del proyecto, incluyendo la separación entre frontend y backend, cómo se comunica cada componente, y la interacción con la base de datos.
- **Modelado de Datos:** Crea el esquema de la base de datos y los modelos que representan las entidades clave.
- **Diseño de Interfaces:** Realiza wireframes o prototipos de las pantallas principales de la aplicación.

5. Configuración del Entorno de Desarrollo

- **Repositorio de Código:** Crea un repositorio en GitHub, GitLab o Bitbucket y configura el control de versiones.

- **Entorno Local:** Configura el entorno de desarrollo local, instalando dependencias y preparando las herramientas necesarias (por ejemplo, Node.js, Docker, etc.).
- **Entorno de Pruebas:** Configura un entorno de pruebas (si es necesario) donde puedas desplegar versiones previas de la aplicación.

6. Desarrollo Iterativo

- **Módulos Iniciales:** Comienza a trabajar en los componentes fundamentales (como la autenticación de usuarios o la configuración de la base de datos).
- **Iteraciones Cortas:** Implementa funcionalidades en pequeñas iteraciones, haciendo commits regulares al repositorio.
- **Pruebas Continuas:** Prueba cada módulo a medida que lo desarrollas para detectar errores temprano.

7. Documentación del Progreso

- **Registros:** Lleva un registro de las decisiones de diseño y las modificaciones realizadas.
- **Manual de Instalación y Uso:** Comienza a documentar cómo instalar, configurar y usar la aplicación desde el principio para evitar acumulaciones al final.

8. Pruebas y Refinamiento

- **Pruebas Unitarias y de Integración:** Asegúrate de que cada parte de la aplicación funcione correctamente tanto de manera aislada como en conjunto.
- **Corrección de Errores:** Resuelve cualquier bug encontrado y realiza ajustes según sea necesario.

9. Preparación para la Entrega

- **Revisión Final:** Realiza una revisión integral del código, la funcionalidad y la documentación.
- **Optimización:** Haz ajustes finales en cuanto a rendimiento, seguridad y usabilidad.
- **Preparación de la Documentación:** Completa toda la documentación necesaria, incluyendo el manual del usuario y la guía técnica.

10. Despliegue y Presentación

- **Deploy:** Despliega la aplicación en un entorno de producción o prepara un entorno donde el proyecto pueda ser evaluado.
- **Presentación:** Prepara una presentación para explicar el proyecto, sus características, y el proceso de desarrollo.

Consejo Final

Siempre comienza con una base sólida en la planificación y diseño, ya que esto te ahorrará tiempo y problemas en las etapas posteriores del desarrollo. Mantén una comunicación continua si trabajas en equipo y asegúrate de hacer revisiones regulares para mantener el proyecto en el buen camino.