Práctica 1. Inforgrafía:
Los 8 Pilares de un Proyecto de Software

Instrucciones: Con la investigación grupal y con los siguientes elementos, ensambla una infografia de la pregunta 1, llamada:

Los 8 Pilares de un Proyecto de Software

- Saber que hacer
- Diseñar bien el sistema
- Elegir las herramientas
- Probar todo
- Controlar los cambios
- Documentar todo
- Proteger el software
- Organizar el trabajo

Información complementaria:

Requisitos Técnicos Básicos para un Proyecto de Software

- **Saber qué hacer:** Antes de empezar, hay que entender bien qué debe hacer el software. Esto incluye sus funciones, límites y cómo saber si funciona bien.
- **Diseñar bien el sistema:** Hay que planear cómo se va a construir el software, pensando en que sea fácil de mejorar, mantener y seguro. Esto incluye cómo se organizarán las partes del programa.
- **Elegir las herramientas correctas:** Escoger los lenguajes de programación, bases de datos y programas que ayudarán a crear el software. Por ejemplo, usar Git para guardar cambios en el código.
- **Probar todo:** Hacer pruebas para asegurarse de que el software funciona bien. Esto incluye probar partes pequeñas, cómo funcionan juntas y si cumple con lo que se necesita.
- **Controlar los cambios:** Usar herramientas como Git para guardar y organizar los cambios en el código, así no se pierde nada y se puede volver atrás si es necesario.
- **Documentar todo:** Escribir guías claras sobre cómo funciona el software, cómo usarlo y cómo está hecho. Esto ayuda a que otros lo entiendan y lo mantengan.
- **Proteger el software:** Desde el principio, hay que pensar en la seguridad. Esto incluye proteger los datos, verificar quién puede usar el software y hacer pruebas para encontrar fallos de seguridad.
- **Organizar el trabajo:** Usar métodos como Scrum o herramientas como Trello para planear y seguir el progreso del proyecto. Esto ayuda a que todo salga a tiempo y sin problemas.

Práctica 2. Comparte en equipo.

Cómo Asegurar que el Software Funcione en el Mundo Real

Instrucciones: Con la respuesta grupal, de la siguiente pregunta:

1. ¿Cómo se aseguran de que el software sea compatible con el entorno productivo?

Ensambla una infografia llamada:

Cómo Asegurar que el Software Funcione en el Mundo Real

1. 🔍 **Entender el Entorno:** Investiga cómo es el lugar donde se usará el software (hardware, sistemas operativos, redes).
2. 🧪 **Pruebas en un Ambiente Similar:** Crea un entorno de prueba idéntico al real para probar el software.
3. 💻 **Compatibilidad con Sistemas Operativos:** Asegúrate de que funcione en Windows, Linux, macOS, etc.
4. 🖥️ **Compatibilidad con Hardware:** Prueba el software en equipos con las mismas especificaciones que los del entorno real.
5. 🔗 **Integración con Otros Sistemas:** Verifica que funcione bien con bases de datos, APIs y otros programas.
6. ⚡ **Pruebas de Rendimiento:** Prueba cómo se comporta el software con muchos usuarios o datos.
7. 🔄 **Actualizaciones y Parches:** Asegúrate de que el software se pueda actualizar sin problemas.
8. 📄 **Documentación y Soporte:** Crea guías claras para instalar, configurar y usar el software.
9. 📊 **Monitoreo Post-Implementación:** Supervisa el software después del lanzamiento para detectar errores.

FRASE O CONSEJO: _____

Instrucciones: Con la investigación grupal, sobre: ¿Qué consideraciones son importantes al configurar *hardware* y *software*?, realiza un póster estilo cómic con un personaje guía (un robot o un científico loco) que explique cada paso.

Diseño: Cada paso se presenta como un nivel de un videojuego, donde el personaje avanza superando desafíos. Incluye iconos, colores llamativos y frases motivadoras.

Toques Creativos:
• Personaje Guía: Un robot o científico que aparezca en cada nivel dando consejos.
• Colores: Usa una paleta vibrante (azules, verdes, naranjas) para que sea atractivo.
• Iconos: Iconos grandes y divertidos para cada paso.
• Fondo: Un estilo futurista o de laboratorio.

Herramientas para Crearlo: Canva: Para diseñar la infografía con plantillas modernas.

Nivel 1: 🧩 Compatibilidad
• Frase: ¡Encuentra las piezas que encajan!
• Descripción: Asegúrate de que el hardware y el software sean compatibles entre sí y con el sistema operativo.

Nivel 2: 📋 Requisitos del Sistema
• Frase: ¡Revisa la lista antes de empezar!
• Descripción: Verifica los requisitos mínimos de hardware y software para que todo funcione bien.

Nivel 3: 🛠️ Configuración del Hardware
• Frase: ¡Conecta todo como un experto!
• Descripción: Ensambla correctamente los componentes y configura el BIOS/UEFI.

Nivel 4: ⚙️ Configuración del Software
• Frase: ¡Ajusta el software a tu medida!
• Descripción: Instala y configura el software, activa medidas de seguridad y personaliza las opciones.

Nivel 5: 🚀 Optimización del Rendimiento
• Frase: ¡Haz que vuele!
• Descripción: Ajusta el hardware y el software para que funcionen a máxima velocidad.

Nivel 6: 🧪 Pruebas y Verificación
• Frase: ¡Prueba antes de usar!
• Descripción: Verifica que todo funcione correctamente con herramientas de diagnóstico y pruebas.

Nivel 7: 📚 Documentación y Soporte
- **Frase**: ¡Guarda tus secretos!
- **Descripción**: Documenta la configuración y guarda manuales, licencias y copias de seguridad.

Práctica 4. Cuadro compartivo

1. **Instrucciones:** Con la investigación grupal, sobre: ¿Qué sistemas operativos son más comunes en servidores y por qué?

Completa la siguiente tabla:

Sistemas Operativos para Servidores	Versiones populares	Principales característcias	Logo
Linux	Ubuntu Server, CentOS, RHEL	Gratuito, estable y seguro	
Windows Server	Windows Server 2019, 2022	Fácil de usar, ideal para empresas	
Unix	IBM AIX, Oracle Solaris	Estable y seguro para entornos críticos	

La elección del sistema operativo depende de las necesidades del servidor:

- Si buscas **gratuidad y flexibilidad**, elige _____
- Si prefieres **facilidad de uso y soporte profesional**, elige _____
- Si necesitas **máxima estabilidad y seguridad**, considera _____

Práctica 5. Cuadro comparativo.

Instrucciones: Con la investigación grupal, sobre ¿Qué herramientas o métodos se utilizan para asegurar que un software funcione correctamente en diferentes entornos?

Completa la siguiente tabla:

Métodos	Qué es	Herramientas
1. Pruebas de Compatibilidad		
2. Pruebas Unitarias		
3. Pruebas de Integración		
4. Pruebas de Rendimiento		
5. Pruebas de Seguridad		
6. Pruebas de Usabilidad		
7. Automatización de Pruebas		
8. Monitoreo Continuo		

Consejo

La clave es combinar **pruebas manuales** y **automatizadas**, y usar herramientas adecuadas para cada tipo de prueba. Así te aseguras de que el software funcione correctamente en cualquier entorno.

Práctica 6:
Informe de configuración del software

En equipos de 3 a 4 personas, investigan cómo se configuró el entorno para una aplicación famosa como Netflix o Spotify. Escribirán un breve informe en un documento colaborativo, explicando:

Portada

Introducción: Generalidades de la aplicación y resumen del contenido.

Desarrollo

- Requisitos de Software
  - Los requisitos técnicos de software.
  - Los requisitos técnicos para el desarrollo de Software.
  - Cómo se configuró el hardware y el software paso a paso en forma de tutorial.
  - Tabla de ejemplo de compatibilidad del software (apoyo docente).
  - Configuración de entorno virtual en Python

Conclusiones:
De forma individual responder a lo siguiente:
¿Qué aprendieron sobre la importancia de los requisitos técnicos?
¿Cómo se sintieron al configurar un entorno desde cero?
¿Qué habilidades creen que necesitan desarrollar para configurar software adecuadamente?

Anexos: Glosario de términos