

Desarrollo de soluciones informáticas aplicando inteligencia artificial

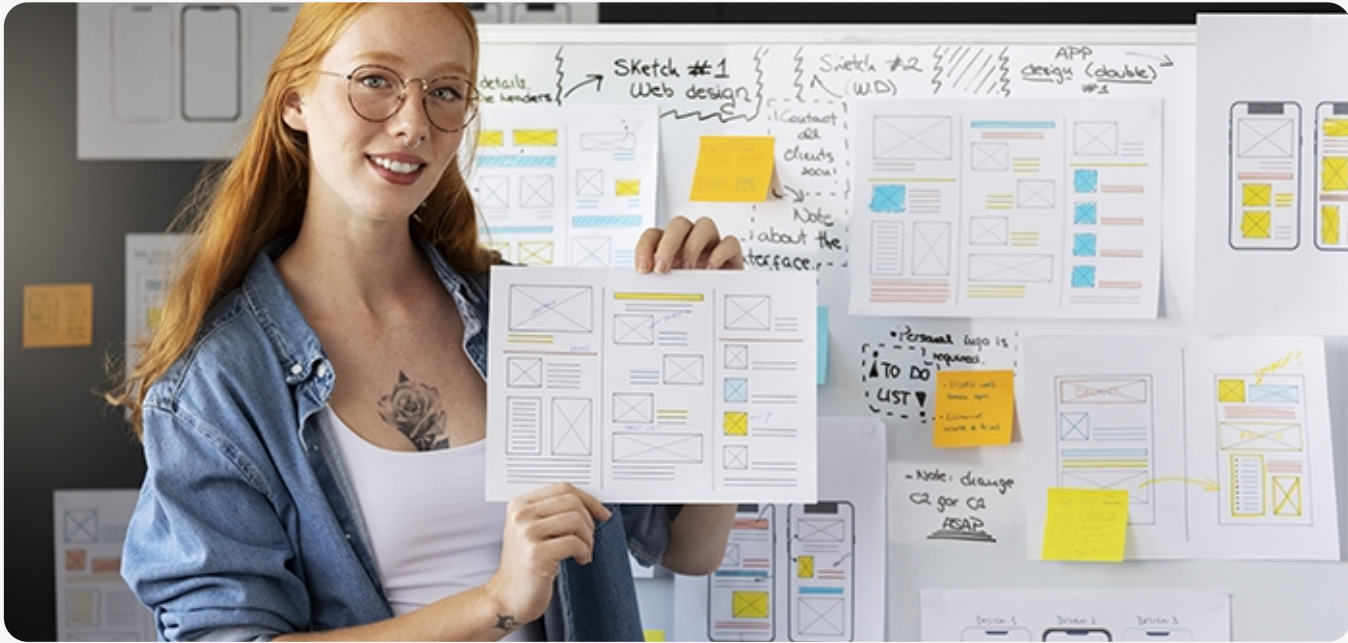
Síntesis: Desarrollo de software: programación, arquitectura y herramientas.



El diagrama representa la estructura integral del componente sobre desarrollo de *software*, centrado en los fundamentos, arquitectura, control de versiones y entornos de desarrollo. Partiendo del concepto central del desarrollo de *software*, se ramifica en cuatro áreas esenciales: fundamentos de lenguajes de programación, arquitectura de *software*, control de versiones y entornos de desarrollo. Cada una de estas áreas incorpora subtemas específicos que conforman los elementos fundamentales para comprender y aplicar eficazmente las prácticas modernas de desarrollo de *software*.



Esta organización ilustra el flujo lógico del proceso de desarrollo de *software*. Comienza con la comprensión de los fundamentos de programación, proporcionando una base sólida en sintaxis, control de flujo y estructuras de datos. Luego, profundiza en la arquitectura de *software*, abordando patrones de diseño y principios fundamentales que son esenciales para construir aplicaciones robustas y escalables. A continuación, se enfoca en el control de versiones con Git y las prácticas de desarrollo colaborativo, herramientas estratégicas para el trabajo en equipo y la gestión eficiente del código. Finalmente, aborda los entornos de desarrollo y las herramientas modernas, destacando la importancia del *testing*, *debugging* y las consideraciones de seguridad en el despliegue de aplicaciones.



El diagrama funciona como una hoja de ruta visual para comprender la estructura y el alcance del componente, permitiendo al estudiante visualizar rápidamente la progresión del aprendizaje y las conexiones entre los diferentes temas. Los elementos transversales (desarrollo colaborativo, buenas prácticas y herramientas modernas) se entrelazan con todas las áreas, enfatizando su importancia en todo el proceso de desarrollo. Se sugiere utilizarlo como referencia para organizar el estudio y entender cómo se integran los diversos aspectos en el desarrollo de *software* moderno, garantizando la creación de aplicaciones mantenibles y escalables en entornos profesionales.

