### Pauta de Autoevaluación de Competencias

(complemento de la Pauta de Reflexión Definición Proyecto APT)

Objetivo:

El objetivo de esta pauta de autoevaluación es que identifiques tus niveles de logro en las competencias de tu plan de estudio para que, a partir de tus fortalezas y oportunidades de mejora, puedas definir mejor tu proyecto APT. Esta pauta de autoevaluación es un complemento de las reflexiones iniciales de APT que también te ayudarán a definir tu Proyecto APT.

Instrucciones:

* Completa la tabla con las competencias de tu perfil de egreso (las puedes revisar con tu docente)
* Piensa en tu proceso de aprendizaje durante el tiempo que has estudiando en Duoc UC y evalúa el nivel de logro que alcanzaste en cada competencia de tu plan de estudio.
* Marca con una cruz el nivel de logro alcanzado para cada aprendizaje de las unidades de competencia según las siguientes categorías:

| **Categoría** | **Descripción** |
| --- | --- |
| Excelente Dominio (ED) | Tengo un excelente dominio en esta competencia y no necesito reforzarla. |
| Alto Dominio (AD) | Tengo un muy buen dominio de esta competencia, solo necesito reforzar pocos aspectos que no tengo completamente desarrollados. |
| Dominio Aceptable (DA) | Tengo un dominio básico de la competencia, que me permite lograr los aspectos centrales de ésta, pero aún tengo muchos que necesito reforzar. |
| Dominio insuficiente (DP) | Tengo un dominio muy básico de la competencia, solo manejo alguno aspectos de manera aislada. |
| Dominio no logrado (DNL) | Tengo un dominio no logrado de la competencia, no manejo casi ningún aspecto de manera clara. |

* En la columna de comentarios escribe por qué marcaste cada nivel.

| Escuela | |
| --- | --- |
| Nombre completo |  |
| Plan de Estudio |  |
| Año de ingreso |  |

| **Competencias Perfil de egreso** | **Nivel de logro** | | | | | **Comentarios** |
| --- | --- | --- | --- | --- | --- | --- |
| **Excelente Dominio** | **Alto Dominio** | **Dominio Aceptable** | **Dominio Insuficiente** | **Dominio no logrado** |
| **Dominio de tecnologías front-end: Habilidad para desarrollar interfaces de usuario interactivas y responsivas usando tecnologías clave como HTML5, CSS3 y JavaScript. Esto incluye el uso de frameworks modernos como React, Angular o Vue.js para construir Single Page Applications (SPAs).** |  |  | **x** |  |  |  |
| **Manejo de bases de datos: Capacidad para diseñar, implementar y gestionar bases de datos relacionales (como SQL Server, MySQL, PostgreSQL) y no relacionales (como MongoDB, Cassandra). Esto implica conocer el modelado de datos y el uso de lenguajes de consulta.** |  | **x** |  |  |  |  |
| **Desarrollo de APIs RESTful: Competencia en la creación y el consumo de interfaces de programación de aplicaciones (APIs) siguiendo los principios REST. Se espera que el ingeniero sepa diseñar endpoints eficientes y seguros para la comunicación entre el front-end y el back-end.** |  |  | **x** |  |  |  |
| **Gestión de servidores y back-end: Conocimiento en el desarrollo de la lógica del lado del servidor utilizando lenguajes como Python (con frameworks como Django o Flask), Node.js (con Express), Java (con Spring) o PHP.** |  |  | **x** |  |  |  |
| **Pruebas y depuración de software: Habilidad para realizar pruebas unitarias, de integración y de extremo a extremo para garantizar la calidad y fiabilidad del código. Esto incluye el uso de herramientas de depuración para identificar y corregir errores.** |  | **x** |  |  |  |  |
| **Arquitectura de software: Capacidad para diseñar arquitecturas de aplicaciones web escalables y modulares. Esto abarca la comprensión de patrones de diseño de software (MVC, microservicios) y la toma de decisiones sobre las tecnologías más adecuadas para un proyecto.** |  | **x** |  |  |  |  |
| **Control de versiones: Dominio de sistemas de control de versiones, especialmente Git, para colaborar eficientemente en equipos, gestionar cambios en el código y resolver conflictos de manera profesional.** |  |  |  | **x** |  |  |
| **Pensamiento algorítmico y resolución de problemas: Habilidad para aplicar principios de algoritmia y estructuras de datos para resolver problemas complejos de manera eficiente. Esta es una competencia fundamental que subyace a todas las demás áreas, permitiendo escribir código optimizado y de alta calidad.** |  | **x** |  |  |  |  |