|  |  |
| --- | --- |
| ACTIVIDAD DIDÁCTICA CUESTIONARIO | |
| Generalidades de la actividad   * Las indicaciones, el mensaje de correcto e incorrecto debe estar la redacción en segunda persona. * Diligenciar solo los espacios en blanco. * El aprendiz recibe una retroalimentación cuando responde de manera correcta o incorrecta cada pregunta. * Señale en la columna Rta. Correcta con una (x) de acuerdo con las opciones presentadas. * Al final de la actividad se muestra una retroalimentación de felicitación si logra el 70 % de respuestas correctas o retroalimentación de mejora si es inferior a este porcentaje.   Para sugerir este tipo de actividad tener presente equipo de Diseño Instruccional, que solo debe haber máximo doce opciones de pregunta y que cada campo tiene un límite de palabras permitidas para garantizar el *responsive web*. | |
| Instrucciones para el aprendiz | *Esta actividad le permitirá determinar el grado de apropiación de los contenidos del componente formativo «****Desarrollo de software: programación, arquitectura y herramientas****».*  *Antes de su realización, se recomienda la lectura del componente formativo mencionado. Es opcional (no es calificable), y puede realizarse todas las veces que se desee.*  *Lea la afirmación de cada ítem y luego señale verdadero o falso según corresponda.* |
| Nombre de la Actividad | *Prueba de conocimientos sobre desarrollo de software: programación, arquitectura y herramientas.* |
| Objetivo de la actividad | *Validar el conocimiento adquirido sobre sistematización y documentación de datos masivos mediante métodos de analítica. Esto se trabaja a partir de un conjunto de preguntas con el propósito de buscar una dinámica de razonamiento ágil sobre opciones cerradas y reafirmar un conocimiento declarado dentro del componente.* |
| Texto descriptivo | *Lea cada enunciado referente a los temas desarrollados en el componente formativo y elija entre verdadero y falso según corresponda.* |

|  |
| --- |
| PREGUNTAS |

| # | Pregunta | V | F | Retroalimentación Correcta | Retroalimentación Incorrecta |
| --- | --- | --- | --- | --- | --- |
| 1 | Los lenguajes de programación de alto nivel están más cerca del lenguaje de máquina que los de bajo nivel. |  | X | ¡Correcto! Los lenguajes de alto nivel están más alejados del lenguaje de máquina, siendo más cercanos al lenguaje humano. | Incorrecto. Los lenguajes de bajo nivel están más cerca del lenguaje de máquina, mientras que los de alto nivel son más abstractos. |
| 2 | Git es un sistema de control de versiones distribuido. | X |  | ¡Correcto! Git es efectivamente un sistema de control de versiones distribuido que permite trabajo colaborativo. | Incorrecto. Git es un sistema distribuido, lo que significa que cada desarrollador tiene una copia completa del repositorio. |
| 3 | Una API RESTful no mantiene estado entre peticiones. | X |  | ¡Correcto! REST es stateless, lo que significa que cada petición debe contener toda la información necesaria. | Incorrecto. La ausencia de estado (stateless) es uno de los principios fundamentales de REST. |
| 4 | El staging area en Git es lo mismo que el repositorio local. |  | X | ¡Correcto! El staging area es un área intermedia donde se preparan los cambios antes de confirmarlos al repositorio. | Incorrecto. El staging area y el repositorio local son áreas diferentes en Git con distintos propósitos. |
| 5 | Los microservicios son siempre mejor opción que las arquitecturas monolíticas. |  | X | ¡Correcto! La elección entre microservicios y monolito depende del contexto y necesidades específicas del proyecto. | Incorrecto. No existe una solución universal; cada arquitectura tiene sus propios casos de uso apropiados. |
| 6 | El *debugging* solo debe realizarse cuando el código tiene errores evidentes. |  | X | ¡Correcto! El *debugging* es una práctica continua que también se usa para optimización y comprensión del código. | Incorrecto. El *debugging* es una herramienta de desarrollo que se usa tanto para corrección como para análisis del código. |
| 7 | Los principios SOLID se aplican únicamente a la programación orientada a objetos. |  | X | ¡Correcto! Aunque surgieron en POO, los principios SOLID pueden aplicarse a cualquier paradigma de programación. | Incorrecto. Los principios SOLID son aplicables a diversos paradigmas de programación, no solo a POO. |
| 8 | Una rama (*branch*) en Git es una copia independiente del código. | X |  | ¡Correcto! Una rama permite desarrollo independiente sin afectar otras líneas de trabajo. | Incorrecto. Las ramas son fundamentales para el desarrollo paralelo y la gestión de características. |
| 9 | El *testing* automático solo es necesario en proyectos grandes. |  | X | ¡Correcto! El *testing* automático es valioso en proyectos de cualquier tamaño para garantizar calidad. | Incorrecto. El *testing* automático es importante en cualquier proyecto, independientemente de su tamaño. |
| 10 | La integración continua requiere *commits* frecuentes al repositorio compartido. | X |  | ¡Correcto! La integración continua se basa en la integración frecuente de cambios al repositorio principal. | Incorrecto. La frecuencia de *commits* es fundamental para la integración continua efectiva. |
| 11 | Los *arrays* son estructuras de datos de tamaño fijo en la mayoría de los lenguajes. | X |  | ¡Correcto! En la mayoría de los lenguajes, los *arrays* tienen un tamaño fijo una vez creados. | Incorrecto. Los *arrays* tradicionales tienen un tamaño fijo, a diferencia de estructuras dinámicas como las listas. |
| 12 | El *deployment* solo debe realizarse cuando todas las pruebas están perfectas. |  | X | ¡Correcto! El *deployment* puede realizarse en diferentes ambientes para pruebas y validación continua. | Incorrecto. Existen diferentes ambientes de *deployment* que permiten validación progresiva. |
| 13 | Los pull requests son exclusivamente para fusionar código. |  | X | ¡Correcto! Los pull requests también sirven para revisión de código y discusión de cambios. | Incorrecto. Los pull requests son herramientas de colaboración que incluyen revisión y discusión. |
| 14 | La seguridad en el desarrollo es responsabilidad exclusiva del equipo de seguridad. |  | X | ¡Correcto! La seguridad es responsabilidad de todos los desarrolladores desde el inicio del desarrollo. | Incorrecto. La seguridad debe ser considerada por todos los desarrolladores durante todo el proceso. |
| 15 | Un IDE siempre es mejor que un editor de código simple. |  | X | ¡Correcto! La elección entre IDE y editor depende del contexto y necesidades específicas. | Incorrecto. Tanto IDEs como editores tienen sus casos de uso apropiados. |
| 16 | Las variables deben tener nombres descriptivos que indiquen su propósito. | X |  | ¡Correcto! Los nombres descriptivos mejoran la legibilidad y mantenibilidad del código. | Incorrecto. Los nombres de variables deben ser claros y describir su propósito. |
| 17 | La documentación del código solo es necesaria en proyectos complejos. |  | X | ¡Correcto! La documentación es importante en cualquier proyecto para facilitar mantenimiento y colaboración. | Incorrecto. Todo proyecto debe estar adecuadamente documentado, independiente de su complejidad. |
| 18 | El versionamiento semántico (SemVer) ayuda a gestionar dependencias. | X |  | ¡Correcto! SemVer permite gestionar versiones y dependencias de manera consistente. | Incorrecto. El versionamiento semántico es crucial para la gestión efectiva de dependencias. |
| 19 | Los *commits* deben contener múltiples cambios no relacionados. |  | X | ¡Correcto! Cada *commit* debe representar un cambio lógico y cohesivo. | Incorrecto. Los *commits* deben ser atómicos y representar cambios relacionados. |
| 20 | Las estructuras de control son específicas de cada lenguaje de programación. |  | X | ¡Correcto! Las estructuras de control son conceptos fundamentales presentes en la mayoría de los lenguajes. | Incorrecto. Aunque la sintaxis varía, los conceptos de estructuras de control son universales. |

|  |  |
| --- | --- |
| MENSAJE FINAL ACTIVIDAD | |
| Mensaje cuando supera el 70 % de respuestas correctas | *¡Excelente! Ha superado la actividad demostrando sólidos conocimientos sobre el componente formativo.* |
| Mensaje cuando el porcentaje de respuestas correctas es inferior al 70 % | *No ha superado la actividad. Se le recomienda volver a revisar el componente formativo e intentar nuevamente la actividad didáctica.* |

|  |  |  |
| --- | --- | --- |
| **CONTROL DE REVISIÓN** | | |
|  | **Responsable** | **Fecha** |
|  |  |  |