Identificación aprendiz: C.C : 1077723426

Nombre aprendiz: Brayan Santiago Guerrero Mendez

Sección 1: Verdadero o Falso

1. Alistar herramientas TIC implica únicamente instalarlas sin configurar parámetros.

RTA/ Falso

1. Optimizar los resultados requiere verificar el funcionamiento de las herramientas.

RTA/ Verdadero

1. Git es una herramienta útil para el trabajo colaborativo porque permite llevar control de versiones.

RTA/ Verdadero

1. Es irrelevante documentar los cambios cuando se trabaja en equipo usando un sistema de control de versiones.

RTA/ Falso

Sección 2: Múltiple Opción – Respuesta Única (Marca solo una opción correcta)

1. ¿Cuál es el propósito de aplicar funcionalidades de herramientas TIC?  
   a) Evitar el uso de manuales  
   b) Seguir buenas prácticas y procedimientos establecidos  
   c) Instalar sin verificar funcionamiento  
   d) Usarlas de forma intuitiva
2. Optimizar los resultados en herramientas TIC implica:  
   a) Agregar más funciones sin verificar  
   b) Eliminar procesos  
   **c) Ajustar configuración tras verificación**  
   d) Reemplazar software continuamente
3. ¿Qué se hace en la etapa de evaluación de resultados?  
   a) Se ignora la documentación  
   **b) Se comparan con los requerimientos**  
   c) Se crean nuevas herramientas  
   d) Se eliminan datos innecesarios
4. ¿Qué es Git en el contexto del desarrollo de software?  
   a) Un editor de texto  
   b) Un lenguaje de programación  
   **c) Un sistema de control de versiones**  
   d) Un entorno de ejecución

Sección 3: Múltiple Opción – Respuesta Múltiple (Marca todas las respuestas correctas)

1. ¿Qué elementos deben considerarse al aplicar funcionalidades de herramientas TIC?  
   [ x ] Manuales de uso  
   [ ] Experiencia del usuario  
   [ x ] Buenas prácticas  
   [ ] Adivinación del funcionamiento
2. ¿Qué acciones se asocian a la optimización de resultados?  
   [ x ] Verificación del sistema  
   [ ] Comparación con requerimientos  
   [ ] Reinicio forzado  
   [ x ] Ajustes de configuración
3. ¿Qué aspectos se deben evaluar en una herramienta informática?  
   [ x ] Rendimiento frente a requerimientos  
   [ ] Interfaz gráfica únicamente  
   [ x ] Cumplimiento de objetivos del proceso  
   [ ] Color del diseño
4. Al alistar herramientas TIC, ¿qué debe considerarse?  
   [ x ] Compatibilidad con el sistema  
   [ x ] Necesidades de procesamiento de información  
   [ ] Preferencias personales sin análisis  
   [ ] Capacidades de comunicación
5. ¿Cuáles de los siguientes son principios correctos para el uso eficiente de herramientas informáticas y trabajo colaborativo?  
   [ x ] Aplicar buenas prácticas  
   [ ] Selección según necesidades  
   [ ] Configuración al azar  
   [ x ] Evaluación basada en requerimientos

Trabajo practico:

1. Trabajo practico Excel

* Creación de un libro de Excel y guardar el documento.
* Crear y eliminar hojas en el libro de Excel
* Cambio de nombre del libro.
* Crear de Hoja:
  + Estudiante
  + Curso
  + Nota
* Crear 10 Registros

|  |  |  |
| --- | --- | --- |
| Hoja1: Estudiante | Curso | Hoja1: Nota |
| Código Estudiante | Código Curso | Código Estudiante |
| Nombre Estudiante | Nombre Curso | Nombre Estudiante |
| Teléfono |  | Teléfono |
|  |  | Código Curso |
|  |  | Curso |
|  |  | Nota 1 |
|  |  | Nota 2 |
|  |  | Nota 3 |
|  |  | Promedio |

* Diseño tabla
* Crear fórmulas de promedio
* Símbolo ‘
* Manejo del carácter $
* Crear encabezado y diseño de impresión
* Generar Filtros.
* Ordenamiento tabla por nombre estudiante y promedio.
* Contar número de participantes.g
* Promedio del curso.
* Calculo mínimo y máximo.
* Formato condicional en rojo cuando la nota sea inferior a 3 y verde si es mayor o igual a 3.

1. Trabajo practico kanban

Crear el tablero Kanban

Seleccionen una herramienta para su tablero (elige una):

Y crear un repositorio GIT.

Configuren las siguientes columnas mínimas:

* Backlog
* To Do
* In Progress
* Code Review / Testing
* Done

2. Definir tareas del proyecto

Describir las siguientes tareas del proyecto en las anteriores listas:

Planificación y configuración inicial

* Configurar el repositorio Git y agregar el archivo .gitignore (real)
* Crear estructura básica del proyecto (carpetas, paquetes) (simulado)
* Definir requerimientos funcionales y no funcionales (simulado)
* Crear el archivo README.md con descripción inicial (trabajo real)
* Crear diagrama de arquitectura (opcional: usar Lucidchart o Draw.io) (simulado)

🧑‍💻 Desarrollo Backend (simulado)

* Configurar base de datos en entorno local
* Crear entidad "Usuario" con autenticación básica
* Crear API REST para entidad "Tarea"
* Implementar validaciones en los endpoints
* Configurar manejo de errores global

🎨 Desarrollo Frontend (simulado)

* Diseñar layout principal con menú y navegación
* Implementar formulario de inicio de sesión
* Crear componente para mostrar tareas por estado
* Conectar frontend con backend (API)
* Agregar estilos con TailwindCSS o Bootstrap

🧪 Pruebas y aseguramiento de calidad (simulado)

* Escribir pruebas unitarias para servicios
* Realizar pruebas de integración con Postman
* Hacer pruebas manuales del frontend en varios navegadores
* Documentar casos de prueba encontrados
* Corregir bugs detectados durante testing

🔄 Gestión de versiones y colaboración (simulado)

* Crear ramas por funcionalidad (feature/login, feature/task-crud, etc.)

git Branch feature

git Branch login

git Branch task-crud

git Branch testing

* Realizar y revisar Pull Requests (PR)
* Etiquetar versión estable del proyecto (v1.0.0)

🚀 Despliegue (simulado)

* Configurar archivo Dockerfile y docker-compose
* Desplegar backend en servicio gratuito (ej. Render, Railway)
* Desplegar frontend en Vercel o Netlify
* Configurar variables de entorno para producción
* Verificar funcionamiento en entorno en línea

📚 Documentación y cierre (simulado)

* Escribir instrucciones de instalación en el README.md
* Documentar arquitectura del sistema y decisiones técnicas
* Registrar lecciones aprendidas y feedback del equipo
* Preparar presentación del proyecto (video o PDF)
* Archivar o cerrar tareas en el tablero Kanban

Cada tarea debe tener:

* Título
* Descripción
* Asignado a un miembro del equipo
* Fecha estimada de entrega
* Etiquetas (si aplica)

3. Simular el flujo de trabajo

* Muevan tareas a medida que progresan (ej., de "To Do" a "In Progress")
* Use checklists dentro de cada tarjeta si es necesario
* De feedback en tareas (especialmente en la etapa de revisión)
* Realizar commit and push con archivos varios simulando el trabajo anterior.

1. shot con los movimientos de Kanban y git.
2. Crear archivo de Word, con capturas de pantalla o screen:
   1. Encabezado, pie de página con enumeración.
   2. Portada.
   3. Tabla de contenido.
   4. Capturas de pantalla.
   5. Examen.
3. Mínimo 8 commits

7. CONTROL DEL DOCUMENTO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia | Fecha |
| Autor (es) | Carlos Julio Cadena Sarasty | Instructor | CIES | 01/06/2025 |

8. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia | Fecha | Razón del Cambio |
| Autor (es) | Carlos Julio Cadena | Instructor | Tecnologías de la información, diseño y desarrollo de software | 01/06/2025 | Creación. |