

Título del Proyecto: Hoja de Calculo

1. Resumen Ejecutivo

Este proyecto consiste en el desarrollo de una aplicación de hojas de cálculo utilizando lenguaje ensamblador, diseñada para manejar datos numéricos y realizar diversas operaciones matemáticas y lógicas. El programa funcionará en una interfaz de línea de comandos, donde el usuario podrá ejecutar operaciones sobre celdas individuales o rangos de celdas, como sumas, restas, multiplicaciones, divisiones, operaciones lógicas (AND, OR, XOR, NOT) y cálculo de mínimos, máximos, y promedios.

Problema que aborda: El proyecto busca desafiar las habilidades del desarrollador en ensamblador, resolviendo la necesidad de manipular y procesar datos numéricos eficientemente en un entorno de bajo nivel, incluyendo el manejo de errores y el uso de un “lenguaje de comandos” propio de la aplicación para facilitar la interacción del usuario.

Solución a desarrollar: Se plantea la creación de un sistema de hojas de cálculo con una cuadrícula limitada (columnas A-K y filas 1-23), permitiendo al usuario realizar operaciones avanzadas y exportar o importar datos en distintos formatos. Los comandos específicos brindarán flexibilidad para gestionar datos y obtener resultados inmediatos en pantalla, con advertencias en casos de inconsistencias aritméticas o errores en los datos ingresados.

Este enfoque permite aplicar conocimientos de aritmética y lógica en ensamblador, fomentando la creatividad en la implementación de soluciones algorítmicas complejas en un entorno de bajo nivel.

3. Objetivos del Aprendizaje

3.1 Objetivo General

Que el desarrollador pueda aplicar los conocimientos adquiridos en el curso y que éste sea capaz de construir un sistema complejo mediante el lenguaje ensamblador haciendo uso de algoritmos creativos para solucionar los distintos requerimientos solicitados

3.2 Objetivos Específicos

Al finalizar el proyecto, los estudiantes deberán ser capaces de:

1. **Aplicar instrucciones aritméticas a la solución de algoritmos**
2. **Aplicar instrucciones lógicas a la solución de algoritmos.**
3. **Manipular errores en cualquier caso la entrada del usuario lo requiera.**

4. Enunciado del Proyecto

Diseño

En principio se definirán los elementos más importantes que deberán ser mostrados al usuario a la hora de ejecutar el programa. La cuadrícula, lo más importante, tendrá dimensiones limitadas; las columnas estarán marcadas de la letra A a la K, mientras que las filas estarán numeradas comenzando desde 1 a 23. Sobre esta se llevarán a cabo las operaciones. La última línea de la pantalla es la línea de entrada, en donde será posible instruir, mediante comandos específicos, la operación del programa.

	A	B	C	D	E	F	G	H	I	J	K
1	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
2	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
3	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
4	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
5	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
6	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
7	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
8	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
9	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
10	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
11	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
12	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
13	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
14	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
15	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
16	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
17	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
18	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
19	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
20	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
21	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
22	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
23	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000

Celdas

El programa a desarrollar deberá entender el concepto de “celdas”, las cuales están identificadas por un carácter alfabético y un número menor a 24. A la hora de ejecutar operaciones, el programa debe ser capaz de reconocer una referencia a una celda y hacerla efectiva.

Rangos

En este caso se entenderá a rango como una sucesión, vertical u horizontal, de celdas contiguas. Un rango tiene, una celda de inicio y de final. Más adelante se detallarán las operaciones que pueden hacer sobre rangos.

Lenguaje de comandos

Para interactuar con el programa se definió un lenguaje de comandos que será el principal medio de interacción con la persona que utilice el programa. Las palabras o comandos reconocidos serán descritos más adelante.

Retorno de comandos

Según el comando (se indicará más adelante), se generará un retorno. El valor de retorno podrá ser usado en los comandos que requieran una cantidad numérica como parámetro mediante el símbolo ‘*’.

	A	B	C	D	E	F	G	H	I	J	K
1	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
2	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
3	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
4	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
5	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
6	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
7	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
8	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
9	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
10	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
11	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
12	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
13	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
14	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
15	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
16	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
17	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
18	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
19	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
20	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
21	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
22	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
23	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000

: GUARDAR * EN C13

Comandos y operaciones

Operaciones sobre celdas

1. GUARDAR: GUARDAR [Número o celda] EN [Celda]

Con este comando es posible colocar un valor arbitrario en una celda. No se modifica el retorno.

2. SUMA: SUMA [Número o celda] Y [Número o celda]

Esta operación ejecutará una suma y el resultado de ésta será colocado en la variable de retorno.

3. RESTA: RESTA [Número o celda] Y [Número o celda]

Esta operación ejecutará una resta y el resultado de ésta será colocado en la variable de retorno. Si la resta realizada llegara a desbordarse, es necesario que se muestre una advertencia.

4. MULTIPLICACION: MULTIPLICACION [Número o celda] Y [Número o celda]

Se ejecutará una multiplicación. El resultado será colocado en la variable de retorno. Si la operación llegara a generar un número que no puede ser representado con 5 dígitos decimales, se deberá de almacenar el número y en caso el usuario desee colocarlo en la hoja deberá de tener el siguiente formato: 12345!, donde el signo de admiración indica que el numero continua pero no es posible mostrar el numero completo.

5. DIVIDIR: DIVIDIR [Número o celda] ENTRE [Número o celda]

El comando ejecutará una división, la parte entera del resultado será colocada en la variable de retorno. Debe tenerse en cuidado de indicar, por medio de una advertencia, cuando una división no sea posible de ejecutar.

6. POTENCIAR: POTENCIAR [Número o celda] A LA [Número o celda]

Esta operación permitirá ejecutar una potencia tomando como base el primer número y el número restante como exponente. El resultado deberá ser colocado en la variable de retorno. Debe verificarse que el exponente sea positivo pues el programa no considera potencias negativas. Y de igual forma, se debe revisar si el resultado generado por esta operación no puede ser representado en 5 dígitos. De ser así se debe mostrar una advertencia.

7. OLÓGICO: OLÓGICO [Número o celda] Y [Número o celda]

Este comando ejecutará un OR a nivel de bits con las cantidades brindadas como parámetros. El resultado de esta operación se colocará en el retorno.

8. YLÓGICO: YLÓGICO [Número o celda] Y [Número o celda]

Este comando ejecutará un AND a nivel de bits con las cantidades brindadas como parámetros. El resultado de esta operación se colocará en el retorno.

9. OXLÓGICO: OXLÓGICO [Número o celda] Y [Número o celda]

Este comando ejecutará un XOR a nivel de bits con las cantidades brindadas como parámetros. El resultado de esta operación se colocará en el retorno.

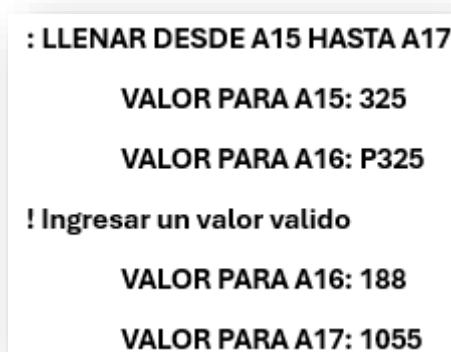
10. NOLÓGICO: NOLÓGICO [Número o celda]

Este comando ejecutará un NOT a nivel de bits con la cantidad brindada como parámetro. El resultado de esta operación se colocará en el retorno.

Operaciones sobre rangos

1. LLENAR: LLENAR DESDE [Celda de inicio] HASTA [Celda de fin]

Este comando deberá poder reconocer la orientación del rango especificado. Una vez reconocido se le solicitará a la persona que ingrese un valor numérico válido para cada celda del rango. De no ser válido lo ingresado se debe mostrar una advertencia. Una vez completado el ingreso de datos se deberá ver reflejado en la cuadrícula el cambio con los datos ingresados anteriormente colocados en su posición correcta. Esta operación no deberá modificar la variable de retorno.



```
: LLENAR DESDE A15 HASTA A17  
VALOR PARA A15: 325  
VALOR PARA A16: P325  
! Ingresar un valor valido  
VALOR PARA A16: 188  
VALOR PARA A17: 1055
```

Ejemplo de uso de comando LLENAR

2. PROMEDIO: PROMEDIO DESDE [Celda de inicio] HASTA [Celda de fin]

Este comando permitirá obtener la parte entera del promedio de un rango. El resultado se colocará en la variable de retorno para su posterior uso.

3. **MÍNIMO: MÍNIMO DESDE [Celda de inicio] HASTA [Celda de fin]**

Por medio de este comando se podrá encontrar el valor mínimo en el rango especificado. Tal valor será el resultado que deberá ser colocado en el retorno.

4. **MÁXIMO: MÁXIMO DESDE [Celda de inicio] HASTA [Celda de fin]**

Por medio de este comando se podrá encontrar el valor máximo en el rango especificado. Tal valor será el resultado que deberá ser colocado en el retorno.

Entrada y Salida

1. **IMPORTAR [Nombre del archivo] SEPARADO POR [COMA | TABULADOR]**

Permitirá importar los datos numéricos de un archivo separado por comas o tabuladores. La primera línea del archivo tendrá los nombres de los encabezados, después de que se haya verificado que el archivo exista el programa pedirá la letra de la columna a la que los datos correspondientes de cada encabezado irán. Teniendo un archivo como el siguiente:

Carne,Tarea 1,Parcial 1,Tarea 2
19001,90,70,85
19002,90,70,85
19003,90,70,85
19004,90,70,85

El programa debe operar de la siguiente forma:

8	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
9	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
10	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
11	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
12	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
13	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
14	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
15	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
16	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
17	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
18	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
19	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
20	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
21	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
22	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
23	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000

: IMPORTAR entrada.csv SEPARADO POR COMAS

Letra de la columna para "Carne": B

Letra de la columna para "Tarea 1": C

Letra de la columna para "Parcial 1": E

Letra de la columna para "Tarea 2": D

Los datos importados se colocarán en la columna indicada partiendo desde la primera fila hacia abajo.

2. EXPORTAR sintaxis: EXPORTAR [Número] DESDE [Letra de columna] HACIA [Nombre de archivo]

Este comando generará un archivo .html que contendrá los datos tabulados. Específicamente se exportarán [Número] columnas desde la columna [Letra de columna]. Debe colocarse la fecha y hora en que se exportaron los datos en alguna parte del archivo generado. Se debe tener cuidado de que la cantidad de columnas no sobrepase el número de columnas disponibles y otras verificaciones que se consideren pertinentes. Para cada columna se pedirá a quien use el programa el encabezado que esa columna tendrá en el archivo generado. Como lo muestra el ejemplo:

8	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
9	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
10	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
11	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
12	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
13	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
14	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
15	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
16	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
17	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
18	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
19	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
20	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
21	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
22	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
23	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000

: EXPORTAR 3 DESDE D HACIA salida.html

Encabezado para columna D: Codigo

Encabezado para columna E: Precio minorista

Encabezado para columna F: Precio mayorista

Puntos Importantes

- Los números a tratar dentro del programa son enteros con signo de 64 bits.
- Las advertencias en cuanto a inconsistencias aritméticas deben estar presentes para que se otorguen los puntos completos.
- Los estudiantes de carné PAR deben desarrollar la operación IMPORTAR tomando en cuenta archivos separados por COMAS.
- Los estudiantes de carné IMPAR deben desarrollar la operación IMPORTAR tomando en cuenta archivos separados por TABULADORES.

4.1 Descripción del problema a resolver

Se le solicita el desarrollo de una aplicación de hojas de cálculo en Ensamblador. Dicho software permitirá manejar datos numéricos por medio de diversas operaciones matemáticas y lógicas, operando sobre datos puntuales o rangos. Se interactuará con el programa a través de una interfaz de línea de comandos por la cual se instruirá al programa sobre las acciones que debe realizar. El

programa también tendrá la capacidad de generar salidas en varios formatos para diversos propósitos.

4.2 Alcance del proyecto

- **Alcance obligatorio:** Se deberá de poder ver visualmente la hoja de calculo y poder ingresar valores a cada una de las celdas. Se deberá de poder implementar al menos un comando.
- **Alcance opcional:** De forma opcional los estudiantes podrán implementar los comandos por rangos, sin embargo de no tener estos comandos se perderán puntos.

4.3 Requerimientos técnicos

Los estudiantes deberán de usar una distribución de Linux para poder emular la arquitectura ARM y se usara QEMU. Estas herramientas fueron definidas en clase.

4.4 Entregables

Tipo	Descripción
Documentación Técnica	Incluye manuales de usuario, guías de instalación, diagramas y descripciones de la arquitectura o flujo del sistema desarrollado.
Código Fuente	El código completo del proyecto, con una estructura clara y comentada, que refleja las buenas prácticas de programación y uso de control de versiones.
Lista de Tareas	Un resumen de las tareas realizadas durante el proyecto, con detalles sobre el progreso, los responsables y el cumplimiento de los plazos.
Manual de Usuario	Documento que explica cómo usar el sistema desarrollado, incluyendo capturas de pantalla, pasos detallados y resolución de problemas comunes.

- **Prototipo funcional de la aplicación:** Se debe entregar un código fuente de la aplicación funcional.
- **Informe técnico:** Un informe de no más de 10 páginas que describa el proceso de desarrollo, los retos encontrados y cómo se resolvieron.
- **Documentación del usuario:** Manual de uso de la aplicación con capturas de pantalla.

5. Metodología

1. Planificación

Objetivo: Entender los requisitos del proyecto y organizar las tareas.

- **Analizar el enunciado:** Leer detenidamente los objetivos y especificaciones, identificando los comandos y operaciones que se deben implementar.
- **Dividir en módulos:** Separar el proyecto en componentes funcionales (e.g., manipulación de celdas, operaciones aritméticas, operaciones lógicas, importación y exportación de datos).
- **Establecer un cronograma:** Asignar tiempo para cada módulo según la dificultad estimada y la fecha límite. Esto ayuda a gestionar el tiempo y evitar retrasos.

2. Diseño

Objetivo: Planificar la estructura del código y la lógica de cada función.

- **Definir la estructura de datos:** Diseñar cómo se almacenarán las celdas y sus valores en memoria, considerando las limitaciones del ensamblador.
- **Especificar la lógica de cada comando:** Escribir pseudocódigo o diagramas de flujo para cada operación. Esto incluye validar los datos ingresados y manejar errores.
- **Establecer el formato de comandos:** Planificar cómo se interpretarán y procesarán los comandos ingresados por el usuario en la línea de comandos.

3. Implementación

Objetivo: Codificar el proyecto en ensamblador, integrando cada módulo.

- **Desarrollar módulos básicos:** Iniciar con las funciones de manipulación de celdas y operaciones aritméticas. Implementar primero los comandos esenciales, como **GUARDAR** y **SUMA**.
- **Codificar manejo de errores:** Incluir validaciones para operaciones como división por cero y límites de representación numérica.

- **Avanzar con operaciones complejas:** Continuar con operaciones de rangos y lógicas (e.g., OR, AND) y la lógica de comandos de importación y exportación.
- **Revisar y optimizar:** Optimizar el código para reducir el uso de recursos, manteniendo el funcionamiento correcto.

4. Pruebas y Documentación

Objetivo: Verificar que el programa funcione correctamente y preparar los manuales requeridos.

- **Realizar pruebas unitarias:** Probar cada comando y operación de forma aislada para asegurar su correcto funcionamiento.
- **Pruebas de integración:** Ejecutar el sistema completo con diferentes secuencias de comandos para verificar la interacción entre módulos.
- **Documentación:** Crear el manual técnico y el manual de usuario en Markdown. Estos deben incluir:
 - **Manual técnico:** Explicación del diseño y estructura del código, descripción de cada comando y manejo de errores.
 - **Manual de usuario:** Instrucciones detalladas para usar el sistema de hojas de cálculo, con ejemplos prácticos de cada comando.

6. Desarrollo de Habilidades Blandas

Para complementar el desarrollo técnico, esta sección se centra en las habilidades blandas que los estudiantes deberán mejorar a lo largo del proyecto, como la comunicación, el liderazgo, y la colaboración en equipo.

6.2 Proyectos Individuales

Los proyectos individuales permiten a los estudiantes desarrollar autonomía y responsabilidad sobre su propio trabajo. En este tipo de proyectos, cada estudiante debe gestionar su tiempo, investigar, planificar y ejecutar todas las fases del proyecto de manera independiente.

6.2.1 Autogestión del Tiempo

Los estudiantes deben crear un cronograma personal para cumplir con los plazos establecidos. Esto les ayuda a mejorar su disciplina y capacidad de priorización, habilidades esenciales en cualquier entorno profesional.

6.2.2 Responsabilidad y Compromiso

En un proyecto individual, el estudiante asume la totalidad de las responsabilidades, desde la

investigación hasta la entrega final. Esto fomenta el sentido de compromiso y permite una mayor personalización en la solución del problema planteado.

6.2.3 Resolución de Problemas

Trabajar de manera independiente impulsa a los estudiantes a buscar soluciones de forma creativa y a enfrentarse a los desafíos sin depender del apoyo constante de otros. Esto fortalece su capacidad para resolver problemas de manera autónoma.

6.2.4 Reflexión Personal

Al concluir el proyecto, el estudiante realiza una autoevaluación, reflexionando sobre sus decisiones, lo aprendido y las áreas en las que podría mejorar. Esta práctica promueve el desarrollo continuo y el autoaprendizaje.

7. Cronograma

El cronograma describe las etapas clave del proyecto, los plazos estimados para cada una, y el proceso de asignación, elaboración y calificación de las tareas. Los estudiantes deberán seguir este plan para asegurar que el proyecto avance de manera organizada y cumpla con los plazos establecidos. Cada fase incluye la asignación de tareas, el tiempo estimado para su elaboración, y el momento de su calificación.

Tipo	Fecha Inicio	Fecha Fin
Asignación de Proyecto	Por definir	Por definir
Elaboración	Por definir	Por definir
Calificación	Por definir	Por definir

8. Evaluación

La evaluación del proyecto se realizará en función de varios criterios clave, teniendo en cuenta tanto los aspectos técnicos como las habilidades blandas demostradas a lo largo del desarrollo. A continuación, se presenta un **ejemplo** de cómo podría distribuirse la puntuación, aunque esto puede cambiar según sea necesario y las particularidades del proyecto.

Resumen de Puntuaciones

Área	Puntos Totales	Puntos Obtenidos
Funcionalidad del Proyecto	80	
Procedimiento y Desarrollo	10	
Innovación y Originalidad	5	
Habilidades Blandas y Presentación	5	
Total	100	

Detalle de la Calificación

DESCRIPCIÓN DE PONDERACIÓN	VALOR	OBSERVACIÓN	PUNTEO
Diseño profesional y entendible de la hoja de calculo	2		
Se logra apreciar de forma clara el encabezado con los datos solicitados al iniciar su aplicación.	2		
Operaciones sobre Celdas			
Comando GUARDAR	4		
Sintaxis correcta del comando	0.5		
Cumple con la tarea propuesta	2		
No modifica el retorno	0.5		
Para el primer parámetro acepta un numero o una celda	0.5		
Verificación de cualquier error en el comando	0.5		
Comando SUMA	4		
Sintaxis correcta del comando	0.5		
Cumple con la tarea propuesta	2		
Modifica el retorno	0.5		
Para ambos parámetros acepta un numero o una celda	0.5		
Verificación de cualquier error en el comando	0.5		
Comando RESTA	4		

Original Estudiante

Copia tutor Académico(a entregar a Coordinación DTT-ECYS)

Sintaxis correcta del comando	0.5		
Cumple con la tarea propuesta	2		
Modifica el retorno	0.5		
Para ambos parámetros acepta un numero o una celda	0.5		
Verificación de cualquier error en el comando	0.5		
Comando MULTIPLICACION	4		
Sintaxis correcta del comando	0.5		
Cumple con la tarea propuesta	2		
Modifica el retorno	0.5		
Para ambos parámetros acepta un numero o una celda	0.5		
Verificación de cualquier error en el comando	0.5		
Comando DIVIDIR	4		
Sintaxis correcta del comando	0.5		
Cumple con la tarea propuesta	2		
Modifica el retorno	0.5		
Para ambos parámetros acepta un numero o una celda	0.5		
Verificación de cualquier error en el comando	0.5		
Comando POTENCIAR	4		
Sintaxis correcta del comando	0.5		
Cumple con la tarea propuesta	2		
Modifica el retorno	0.5		
Para ambos parámetros acepta un numero o una celda	0.5		
Verificación de cualquier error en el comando	0.5		
Comando OLOGICO	4		
Sintaxis correcta del comando	0.5		
Cumple con la tarea propuesta	2		
Modifica el retorno	0.5		
Para ambos parámetros acepta un numero o una celda	0.5		
Verificación de cualquier error en el comando	0.5		
Comando YLOGICO	4		
Sintaxis correcta del comando	0.5		
Cumple con la tarea propuesta	2		
Modifica el retorno	0.5		
Para ambos parámetros acepta un numero o una celda	0.5		
Verificación de cualquier error en el comando	0.5		
Comando OXLÓGICO	4		

Sintaxis correcta del comando	0.5		
Cumple con la tarea propuesta	2		
Modifica el retorno	0.5		
Para ambos parámetros acepta un numero o una celda	0.5		
Verificación de cualquier error en el comando	0.5		
Comando NOLÓGICO	4		
Sintaxis correcta del comando	0.5		
Cumple con la tarea propuesta	2		
Modifica el retorno	0.5		
Para el parámetro acepta un numero o una celda	0.5		
Verificación de cualquier error en el comando	0.5		
Operaciones sobre rangos			
Comando LLENAR DESDE	6		
Sintaxis correcta del comando	0.5		
Cumple con la tarea propuesta	2		
Reconoce la orientación del rango especificado	1		
Vuelve a pedir el valor de un numero ingresado incorrectamente para una posición de la celda	0.5		
Se refleja en la cuadrícula los números ingresados correctamente	1		
Le indica al usuario la celda que se está llenando actualmente	1		
Comando PROMEDIO DESDE	6		
Sintaxis correcta del comando	0.5		
Cumple con la tarea propuesta	2		
Modifica el retorno	1		
Para los parámetros acepta únicamente celdas	0.5		
Reconoce la orientación del rango especificado	1		
Verificación de cualquier error en el comando	1		
Comando MINIMO DESDE	6		
Sintaxis correcta del comando	0.5		
Cumple con la tarea propuesta	2		
Modifica el retorno	1		
Para los parámetros acepta únicamente celdas	0.5		
Reconoce la orientación del rango especificado	1		
Verificación de cualquier error en el comando	1		
Comando MAXIMO DESDE	6		

Sintaxis correcta del comando	0.5		
Cumple con la tarea propuesta	2		
Modifica el retorno	1		
Para los parámetros acepta únicamente celdas	0.5		
Reconoce la orientación del rango especificado	1		
Verificación de cualquier error en el comando	1		
Entrada y Salida			
Comando IMPORTAR	6		
Sintaxis correcta del comando	0.5		
Cumple con la tarea propuesta	2		
Le indica al usuario que columna del archivo se está procesando.	1		
Se reflejan de forma correcta los valores en la cuadrícula desde la primera fila.	2		
Verificación de cualquier error en el comando	0.5		
Comando EXPORTAR	6		
Sintaxis correcta del comando	0.5		
Cumple con la tarea propuesta	2		
Archivo generado es visualmente atractivo y comprensible para el usuario	1		
Al generar el archivo pide los encabezados para cada columna.	2		
Verificación de cualquier error en el comando	0.5		
Se muestra el formato especificado para los números que no se pueden representar en su totalidad en pantalla.	4		
Manejo de números negativos para todos los comandos.	6		
Manual Técnico	2.5		
Manual de Usuario	2.5		
PREGUNTAS	5		

Pregunta 1	2.5		
Pregunta 2	2.5		
TOTAL	100		

Comentarios Generales: