

Característica	Muy mal (0-2 puntos)	Mal (2-4 puntos)	Correcto (4 -6 puntos)	Bien (6-8 puntos)	Muy bien (8-10 puntos)
<b>Elementos a estar en la codificación (45%)</b>					
<b>Tipos de Datos (5%)</b>	Los tipos de datos utilizados no son correctos o faltan; además hay error en la sintaxis de definición de los datos	Los tipos de datos utilizados no son correctos o faltan; pero no hay error en la sintaxis	Los tipos de datos utilizados son correctos pero no los más adecuados. Puede faltar alguno, Debido a que no son los adecuados para poder resolver el problema se deben hacer transformaciones.	Los tipos de datos utilizados son los más adecuados pero hay error en la sintaxis de definición de los datos.	Los tipos de datos utilizados son los más adecuados y no hay error de sintaxis
<b>Definición de función (5%)</b>	No hay declaración de la función	Hay declaración de la función pero no es completa	Hay declaración de la función, tiene todas las partes pero el paso de parámetros no es correcto y hay error en la sintaxis	Hay declaración de la función, tiene todas las partes pero el paso de parámetros no es correcto. No hay error en la sintaxis	La declaración de la función es completa, los parámetros son pasados de forma correcta y no hay errores de sintaxis.
<b>Entrada/Salida de Datos (10%)</b>	No ha identificado la entrada ni salida de datos	Falta o la entrada o la salida y hay errores	Están la entrada y salida de datos pero hay errores de lógica de programación pero ninguno de sintaxis	Están la entrada y salida de datos pero hay mínimos errores de lógica y ninguno de sintaxis	La entrada y salida de datos es correcta.
<b>Utilización de Bucles</b>	No identifica ningún bucle, o identifica alguno pero la lógica del bucle no es correcta y hay errores de sintaxis.	Identifica todos los bucles pero hay errores de lógica y sintaxis.	Identifica todos los bucles. Los bucles utilizados no son los más adecuado y falta alguna de las partes funcional de los bucles por lo que falla la lógica y hay error de sintaxis	Identifica todos los bucles. El bucle utilizado es el más adecuado pero falta alguna de las partes funcional de los bucles.	Identifica todos los bucles, son los adecuados son correctos desde el punto de vista de la lógica de programación y no hay errores de sintaxis.
<b>Utilización de condicionales (15%)</b>	No identifica ninguna condición, o identifica alguno pero la lógica de la condición no es correcta.	Identifica todas las condiciones. Falta alguna parte lógica de la condición.	Identifica todas las condicionales pero no están bien utilizadas, la lógica es correcta pero la sintaxis no.	Identifica todas las condicionales, la lógica es correcta pero la sintaxis no.	Identifica todas las condiciones y tanto la lógica como la sintaxis son totalmente correctos
<b>Lógica de la Programación (45%)</b>					
<b>Lógica de los bucles (15%)</b>	La lógica de los bucles es incorrecta	La lógica de los bucles es incorrecta por no tener todas las partes funcionales	La lógica de los bucles es incorrecta porque el código del bucle no es correcto	La lógica de los bucles no es perfecta, hay errores mínimos pero no hay errores en las partes funcionales del bucle	La lógica de los bucles es perfecta, ha utilizado cada bucle correctamente
<b>Lógica de condicionales (5%)</b>	La lógica de las condicionales es incorrecta	La lógica de las condicionales es incorrecta por no tener todas las partes funcionales	La lógica de las condicionales es incorrecta porque el código de las condicionales no es correcto	La lógica de las condicionales no es perfecta, hay errores mínimos pero no hay errores en las partes funcionales de las condicionales	La lógica de las condicionales es perfecta
<b>Lógica de Variables Booleanas (5%)</b>	La lógica de las variables booleanas en bucles, condicionales es incorrecta	La lógica de las variables booleanas es incorrecta por no tener todas las partes funcionales	La lógica de las variables booleanas es incorrecta porque el código no es correcto	La lógica de las variables booleanas no es perfecta, hay errores mínimos pero no hay errores en las partes funcionales	La lógica de las variables booleanas es perfecta
<b>Resultado (10%)</b>	A través de la lógica no se obtiene ningún resultado pedido	A través de la lógica se obtiene alguno de los resultados pero no como se han pedido	A través de la lógica se obtiene alguno de los resultados no todos, pero los que obtiene son tal y como se han pedido	A través de la lógica se obtienen todos los resultados pero no como se han pedido	A través de la lógica se obtienen todos los resultados tal y como se han pedido
<b>Eficiencia (10%)</b>	El código planteado no tiene ninguna lógica	Sólo el 40% del código planteado tiene una lógica correcta	Más del 60 % del código tiene lógica	Más del 80% del código tiene lógica	El código/ algoritmo planteado es perfecto

**Formato (10%)**

<b>Llaves (3%)</b>	No hay llaves	Sólo están las llaves de inicio y fin	Falta alguna llave	Están las llaves pero hay alguna mal puesta	No falta ninguna llave, donde se necesitan las llaves están
<b>Tabulación (5%)</b>	Código no está tabulado	Sólo está tabulado al inicio y fin	Está tabulado en un solo nivel (datos, bucles principales)	Está tabulado en dos niveles	Está perfectamente tabulado
<b>Claridad (2%)</b>	No hay claridad en el código, no se entiende	La claridad es mínima, hay que hacer grandes esfuerzos para poder seguir el código. Hay tachones, no está estructurado, letra difícil de entender.	La claridad es mínima, hay que hacer esfuerzos para poder seguir el código. No hay tachones, pero no está muy estructurado, letra difícil de entender.	Hay claridad, se entiende el código aunque no está muy estructurado y la lectura no es del todo fácil. .	Es limpio, sin tachones, estructurado, organizado, letra clara, sin problemas para entender el código
<b>Complejidad Alcanzada</b>	Es un código totalmente inmaduro desde el punto de vista sintáctico, lógico y resolutivo	Es un código maduro desde el punto de vista sintáctico, pero inmaduro desde el punto de vista lógico y resolutivo	Es un código maduro desde el punto de vista sintáctico y lógico pero no desde un punto de vista resolutivo	Es un código maduro desde el punto de vista sintáctico, lógico y resolutivo pero tiene fallos	Es un código maduro desde el punto de vista sintáctico, lógico y resolutivo sin fallos.