

# Algoritmos y Estructuras de Datos (2023)

[Página Principal](#) / [Mis cursos](#) / [AED \(2023\)](#) / [Ficha 19](#) / [Trabajo Práctico 03: Gestión de Cabinas de Peaje \[...\]](#)

## Trabajo Práctico 03: Gestión de Cabinas de Peaje [3.0]

✓ **Hecho:** Hacer un envío

✓ **Hecho:** Recibir una calificación

**Apertura:** viernes, 25 de agosto de 2023, 08:00

**Cierre:** domingo, 10 de septiembre de 2023, 23:59

**Aclaración importante 1:** El desarrollo de este Trabajo Práctico está pensado para poder desarrollarse en base a las herramientas y temas que se vean hasta la Ficha 20 inclusive. Entre otros, los temas abarcados por esas fichas incluyen instrucciones *arreglos de registros/objetos, ordenamiento, búsqueda y conteo/acumulación por acceso directo*. No deberían tener necesidad de utilizar temas y elementos de fichas posteriores.

**Aclaración importante 2:** Por favor, LEA **TODO** EL CONTENIDO DE ESTAS CONSIGNAS **ANTES** DE COMENZAR A TRABAJAR. No se aceptarán reclamos de ningún tipo por errores que pudiese cometer debido a no haberse tomado el trabajo de leer las consignas que se le fijaron.

a. Enunciado y Consignas<sup>[1]</sup>.

La empresa *Peajes de Sudamérica*, conociendo el éxito de las dos versiones anteriores, solicita ahora una versión 3.0 del programa que permita obtener estadísticas a partir de todas las transacciones del último mes.

En esta versión, los datos deberán ser almacenados en un arreglo de registros/objetos, y el programa debe ser gestionado a través de un menú de opciones. La corrección/calificación de este trabajo será realizada en forma manual por los profesores, así que no hay ahora restricciones en cuanto al formato y estilo de las salidas, ni hay tampoco exigencia en cuanto a cargar datos en cierto orden riguroso.

Cada registro del arreglo a crear debe mantener los datos de una operación de cobro ticket. Por cada ticket se asumen los mismos datos que para el TP2, **pero ahora se agrega también un código o número identificador del ticket** (un número entero, que se asume será diferente y único para cada ticket). En definitiva, cada registro/objeto de tipo **Ticket** deberá contener los datos siguientes:

- **Código identificador del ticket (un número entero) [NUEVO]**.
- Patente del vehículo al que se le cobra el ticket (una cadena).
- Tipo de vehículo (un entero entre 0 y 2 - (0: motocicleta, 1: automóvil, 2: camión)).
- Forma de pago (un dígito 1 o 2 que indica la forma de pago (1: manual, 2: telepeaje)).
- País de la cabina (un dígito entre 0 y 4 que indica el país donde está la cabina que hizo el cobro (0: Argentina - 1: Bolivia - 2: Brasil - 3: Paraguay - 4: Uruguay)).
- Kilómetros recorridos desde la cabina anterior (un entero que puede ser un cero para indicar que la cabina actual es la primera que ese vehículo atraviesa).

Otra vez, el programa deberá procesar e identificar datos de vehículos cuyas patentes pueden ser de cualquiera de los países del Mercosur, más Chile, de acuerdo al mismo modelo anterior. Y pueden venir patentes de otros países (en cuyo caso deberán identificarse como "Otro" cuando sea requerido).

Igual que en la versión 2.0, sólo a los efectos de este trabajo (*y sin que esto sea necesariamente real*), se asume que todos los países Mercosur cobran un **importe base** por cada peaje equivalente a 300 pesos, salvo Brasil que asumiremos que cobra peajes en una **base** de 400 pesos, y Bolivia que cobra peajes en una **base** de 200 pesos. Para simplificar, asuma que en todos los países ese monto está expresado en pesos argentinos (y no se preocupe por este detalle).

El **importe básico** que cada vehículo pagó debe calcularse nuevamente, considerando lo mismo que en la versión 2.0: si el vehículo es una motocicleta se aplica un descuento del 50% sobre el **importe base** que cobra esa cabina, y si el vehículo es un camión se aplica un recargo del 60% sobre el **importe base** de la cabina. Solo los automóviles pagan un importe básico igual al importe base. Distinga por favor entre el **importe base (400, 300 o 200)** y el **importe básico (igual al base si es auto, la mitad del base si es moto, o el base más el 60% si es camión)**.

Del mismo modo que en la versión 2.0, debe calcularse también otra vez el **importe final del ticket** considerando que si la forma de pago fue por telepeaje se aplica un descuento del 10% al **importe básico** indicado en el párrafo anterior. Y si el **pago fue manual**, entonces **el importe final es igual al importe básico**.

Para esta versión 3.0 el programa deberá basarse en un menú de opciones (como se explica más abajo) y procesar los datos de muchos tickets. Los datos de todos esos tickets podrán ser cargados por teclado o podrán venir almacenados en un *archivo de texto* **peajes-tp3.txt** que será provisto para su procesamiento. El archivo de texto con los datos de entrada tendrá la primera línea de *timestamp* igual que en el TP2, y las líneas a partir de la segunda tendrán el **siguiente formato (recuerde que ahora se agrega el código identificador del ticket, que no estaba en el TP2)**:

- **Caracteres del 0 al 9**: El código identificador del ticket (compuesto solo por dígitos) **[NUEVO]**. El código en el archivo es una cadena de 10 caracteres numéricos, que puede tener ceros en las posiciones de la izquierda para completar la longitud de 10 caracteres.
- **Caracteres del 10 al 16**: La patente. Si el vehículo es chileno el caracter en la posición 10 será un blanco. De esa forma, en el archivo la patente siempre tendrá 7(siete) caracteres.
- **Caracter 17**: un dígito entre 0 y 2 que indica el tipo de vehículo (0: motocicleta, 1: automóvil, 2: camión).
- **Caracter 18**: un dígito 1 o 2 que indica la forma de pago (1: manual, 2 telepeaje).
- **Caracter 19**: un dígito entre 0 y 4 que indica el país donde está la cabina que hizo el cobro (0: Argentina - 1: Bolivia - 2: Brasil - 3: Paraguay - 4: Uruguay).
- **Caracteres 20 al 22**: un número entero de tres dígitos o menos, indicando la distancia en kilómetros desde la última cabina de peaje que atravesó el vehículo. Vendrá un cero para indicar que la cabina es la primera que se atraviesa. Si la distancia es menor que 100, se indicarán ceros a la izquierda para completar las tres posiciones (obviamente, entonces, un cero se representará con tres ceros...).

Se pide desarrollar un programa que en base a todo lo anterior, **defina un tipo registro/clase Ticket**, con los atributos indicados aquí, y **cargue un arreglo de registros/objetos**, de modo que cada casillero del arreglo contenga los datos de un ticket. El programa debe incluir dos módulos separados:

- En uno de esos módulos debe definirse la clase Ticket, incluyendo los métodos o funciones separadas que el grupo determine que son aplicables.
- En el otro módulo, debe desarrollarse el programa principal, con un menú de opciones para realizar cada una de las tareas solicitadas más abajo.

Los procesos que deben estar disponibles en el menú principal son los siguientes:

1. Crear el arreglo de registros de forma que contenga todos los datos de todos los tickets guardados en el archivo de texto [peajes-tp3.txt](#) que se provee junto con este enunciado (click en el nombre del archivo para descargarlo). Cada vez que se elija esta opción, **el arreglo debe ser creado de nuevo desde cero**, perdiendo todos los registros que ya hubiese contenido. Asegúrese de que antes de eliminar el viejo arreglo, se muestre en pantalla un mensaje de advertencia al usuario de forma que tenga la opción de cancelar la operación.
2. Cargar por teclado los datos de un ticket, aplicando procesos de validación para cada campo, y agregar un registro con esos datos al arreglo. Cada vez que se elija esta opción, **el nuevo registro debe agregarse al final del arreglo, sin perder ninguno de los registros que el arreglo ya contenía**. Si el arreglo no existiese, debe ser creado y luego agregar el registro cargado.
3. Mostrar todos los registros del arreglo, **ordenados por código de ticket**, de menor a mayor. Cada registro debe ocupar una sola línea en pantalla, y debe mostrarse también el nombre del país al que pertenece cada patente.
4. Buscar si existe en el arreglo un registro cuya patente sea igual a **p** y que haya pasado por una cabina del país **x**, siendo **p** y **x** dos valores que se cargan por teclado. Si existe, mostrar todos sus datos. Si no existe indicar con un mensaje. La búsqueda **debe** detenerse al encontrar el primer registro que coincida con el criterio pedido.
5. Buscar si existe en el arreglo un registro cuyo código de ticket sea igual a **c**, siendo **c** una valor que se carga por teclado. Si existe, cambiar el valor del campo **forma de pago**, de forma que pase a valer el valor contrario (si valía 1, que pase a valer 2, y viceversa), y luego mostrar el registro completo modificado. Si no existe indicar con un mensaje. La búsqueda **debe** detenerse al encontrar el primer registro que coincida con el criterio pedido.
6. Determinar la cantidad de vehículos de cada país que pasaron por las cabinas, contando también los vehículos que no son de ninguno de los seis países básicos. **Observación: ni siquiera se les ocurra plantear un esquema de 7 condiciones y siete contadores separados... Esto se resuelve con un vector de conteo o nada... [Sugerencia: mantener los nombres de los países y la palabra "Otro" en una tupla, y que luego la función/método que detecta el país a partir de la patente no retorne el nombre de país sino el índice para entrar en esa tupla... lo demás, sale solo. No duden en consultar si tienen problemas con esto].**
7. Determinar el importe acumulado por pagos de tickets, por cada uno de los posibles **tipos de vehiculo** (un acumulador por cada uno de los tres tipos). **De nuevo: esto también se hace con un vector de acumuladores, o nada.**
8. En base al resultado obtenido en el punto 7, determinar y mostrar cuál fue cuál fue el tipo de vehículo con mayor monto acumulado, e indicar además qué porcentaje representa ese monto mayor sobre el monto total.
9. Calcular y mostrar la distancia promedio desde la última cabina recorrida entre todos los vehículos del arreglo, e informar además cuántos de los vehículos recorrieron una distancia mayor a ese promedio.

## **b. Reglas de Entrega y Formación de Grupos.**

1.) El trabajo **debe** ser realizado en grupos, y debe ser subido en la fecha indicada para ello en el aula virtual. Es suficiente con que UN alumno del grupo suba el trabajo, y con eso quedará entregado para todos los miembros del grupo. El práctico debe entregarse en un archivo comprimido (zip, rar o el tipo que dispongan los alumnos) cuyo nombre respete el siguiente formato: **2023\_AED\_TP3\_Apellido\_Legajo[Curso]\_Apellido\_Legajo[Curso].zip** (En donde se debe reemplazar *Apellido* por el apellido de cada integrante y *Legajo* por el número de legajo de cada uno, hasta completar al grupo, además de indicar a qué curso pertenece cada integrante. Obviamente, la extensión zip puede ser rar u otra). **Los grupos serán de entre tres y cinco alumnos del mismo curso y/o de cursos diferentes de la Cátedra (NO SERÁN ACEPTADOS BAJO NINGUNA CIRCUNSTANCIA GRUPOS CON MENOS DE TRES INTEGRANTES o CON MÁS DE 5).** Es muy importante, para facilitar el trabajo de corrección, que los alumnos indiquen claramente el curso al cual pertenecen en el nombre del archivo que suben.

2.) Solo podrán subir un trabajo los alumnos que formen parte de un grupo oficialmente formado y notificado por los medios que se hayan indicado en el aula virtual. Ningún estudiante podrá subir un trabajo sin pertenecer previamente a un grupo (no dispondrá de botones o links para subir su trabajo en el aula virtual si no pertenece a un grupo). Y todo grupo que no llegue al menos a tres integrantes, será desactivado. Están avisados.

3.) La nota que obtengan en este trabajo, será promediada en forma ponderada con el resto de las notas que se obtengan en la parte práctica de la materia, según fórmula indicada en el documento *Reglas del Juego* que figura en la Zona Cero del aula virtual del curso, para formar así la nota de trabajos prácticos que empezará a definir la regularidad.

## **c. Posibles Sanciones.**

1.) Si en los trabajos prácticos se detectan resoluciones duplicadas (dos o más grupos con trabajos duplicados o demasiado similares) se asumirá que fueron copiadas y TODOS los alumnos de todos los grupos que estén involucrados serán aplazados.

2.) Como consecuencia, esos alumnos *perderán la posibilidad de promocionar o aprobar en forma directa la materia y la condición de regularidad quedará sujeta a la decisión del docente adjunto y/o al coordinador de la cátedra.*

3.) Por lo tanto CADA GRUPO ES RESPONSABLE DE SU TRABAJO. No habrá disculpa incluso si lo comparten como ayuda a otros compañeros y estos traicionan su confianza copiándolo.

**[1]** El enunciado y planteo general de este *Trabajo Práctico 3* fue desarrollado por los ingenieros *Marcela Tartabini* y *Valerio Frittelli*.

## Estado de la entrega

<b>Grupo</b>	TP3-G182
<b>Estado de la entrega</b>	Enviado para calificar
<b>Estado de la calificación</b>	Publicada
<b>Tiempo restante</b>	La tarea fue enviada 46 segundos antes

**Criterio de  
calificación**
**Criterios de Evaluación para Trabajo Práctico 3.**
**a.) Programa entregado.**

Entrega un archivo que no es Python, o es Python pero no ejecuta (Esta casilla implica reprobar directamente el TP1, marcando en el mínimo posible el resto de las casillas de esta tabla).  
**-1 puntos**

Entrega un programa ejecutable en Python.  
**0 puntos**

**b.) Entrega en fecha.**

Entrega tardía entre las 48hs y la 72hs luego del vencimiento.  
**-3 puntos**

Entrega tardía entre las 24hs y las 48hs luego del vencimiento.  
**-2 puntos**

Entrega tardía dentro de las 24hs luego del vencimiento.  
**-1 punto**

Entrega en término.  
**0 puntos**

c.) Entrega del archivo comprimido.	No cumple convenciones solicitadas para el nombre del archivo comprimido. <b>-1 puntos</b>	Cumple convenciones solicitadas para el nombre del archivo comprimido. <b>0 puntos</b>	
d.) Uso de al menos dos módulos (uno para la clase y otro para el programa principal).	Todo el programa se realiza en un único módulo. <b>0 puntos</b>	Divide correctamente en módulos e importa correctamente. <b>1 puntos</b>	
e.) Definición de la clase requerida.	Mal o muy mal. No define correctamente la clase o no directamente no la define. <b>0 puntos</b>	Regular. <b>1 puntos</b>	Correcto. <b>2 puntos</b>
f.) Estilo de escritura de código fuente (Es simple: prestar atención a los subrayados en gris que el IDE remarque en el código fuente).	Muy mal. Muy improlijo. No respeta convenciones. <b>0 puntos</b>	Regular. <b>1 puntos</b>	Muy bien. <b>2 puntos</b>
g.) Menú de opciones.	Mal o muy mal. No realizado, o no funciona. <b>0 puntos</b>	Regular. <b>1 puntos</b>	Muy bien. <b>2 puntos</b>
h.) Interfaz de usuario (se controla también aquí la ortografía en los mensajes de salida).	Muy pobre. <b>0 puntos</b>	Regular. <b>1 puntos</b>	Muy bien. <b>2 puntos</b>



i.) Opción 1: Carga correcta del archivo de texto y generación correcta del arreglo.	Mal o muy mal. No genera el arreglo o no lo hace con objetos. <b>0 puntos</b>	Regular. <b>1 puntos</b>	Muy bien. <b>2 puntos</b>	
j.) Opción 2: Carga manual del arreglo (incluye procesos de validación).	Mal o muy mal. No realizado o incompleto o no funciona. <b>0 puntos</b>	Regular. <b>1 puntos</b>	Muy bien. <b>2 puntos</b>	
k.) Opción 3: Ordenar y mostrar el arreglo (incluye una línea por cada objeto, y que aparezca correctamente el nombre del país original de la patente).	Muy mal. No realizado o incompleto o no funciona, o no muestra el nombre del país. <b>0 puntos</b>	Mal. Muestra pero no ordena, ordena mal. <b>1 puntos</b>	Regular. Ordena y muestra pero ocupa más de dos líneas por objeto. <b>2 puntos</b>	Muy bien. Ordena y muestra ocupando hasta dos líneas por objeto. Muestra el nombre del país. <b>3 puntos</b>

<b>l.) Opción 4: Búsqueda de primera ocurrencia, por patente.</b>	Muy mal. No desarrolla do o no funciona. <b>0</b> <b>puntos</b>	Mal. Busca pero aplica el algoritmo equivocado. <b>1</b> <b>puntos</b>	Regular. No corta cuando encuentra lo que buscaba o resuelve mal cuando no existe. <b>2</b> <b>puntos</b>	Muy bien. <b>3</b> <b>puntos</b>
<b>m.) Opción 5: Búsqueda de primera ocurrencia, por código (incluye actualización del contenido del objeto encontrado).</b>	Muy mal. No desarrolla do o no funciona. <b>0</b> <b>puntos</b>	Mal. Busca pero aplica el algoritmo equivocado, o, o no actualiza el contenido al encontrarlo. <b>1</b> <b>puntos</b>	Regular. No corta cuando encuentra lo que buscaba o resuelve mal cuando no existe. <b>2</b> <b>puntos</b>	Muy bien. <b>3</b> <b>puntos</b>

<p><b>n.) Opción 6: Conteo por país de origen (vector de conteo, o nada: fueron avisados).</b></p>	<p>Muy mal. No realizado o incompleto o no funciona, o no usó un vector de conteo. <b>0 puntos</b></p>	<p>Mal. Cuenta pero en forma incorrecta o usa mal los índices. <b>1 punto</b></p>	<p>Regular (a criterio del profesor). <b>2 puntos</b></p>	<p>Muy bien. <b>3 puntos</b></p>
<p><b>ñ.) Opción 7: Acumulación de importes por tipo de vehículo (vector de acumulación, o nada: fueron avisados).</b></p>	<p>Muy mal. No realizado o incompleto o no funciona, o no usó un vector de acumulación. <b>0 puntos</b></p>	<p>Mal. Cuenta pero en forma incorrecta o usa mal los índices. <b>1 punto</b></p>	<p>Regular (a criterio del profesor). <b>2 puntos</b></p>	<p>Muy bien. <b>3 puntos</b></p>

**o.) Opción 8: Tipo de vehículo con mayor monto acumulado, y porcentaje sobre el total. (a realizar sobre el vector obtenido en la Opción 7).**

Muy mal.  
No  
realizado o  
incomplet  
o o no  
funciona.  
**0  
punto  
s**

Mal. No  
trabaja  
sobre el  
vector de  
la Opción  
7.  
**1  
punto  
s**

Regular (a  
criterio del  
profesor).  
**2  
punto  
s**

Muy bien.  
**3  
punto  
s**

**p.) Opción 9: Distancia promedio y cantidad de distancias mayores a ese promedio.**

Muy mal.  
No  
realizado o  
incomplet  
o o no  
funciona.  
**0  
punto  
s**

Mal. Calcula  
mal el  
promedio  
o cuenta  
mal.  
**1  
punto  
s**

Regular (a  
criterio del  
profesor).  
**2  
punto  
s**

Muy bien.  
**3  
punto  
s**

**Última  
modificación**

domingo, 10 de septiembre de 2023, 23:58

**Archivos  
enviados**



[2023\\_TP3\\_G182\\_Rodriguez\\_401737\[1k13\]\\_Hidalgo\\_89334\[1k9\]\\_Coro\\_96111\[1k6\]\\_Velo\\_401396\[1k6\]\\_Diez\\_402015\[1k13\].rar](#)



10 de septiembre de 2023, 23:58

[Exportar al portafolios](#)

**Comentarios  
de la entrega**

► [Comentarios \(0\)](#).

# Comentario

## Calificación

Criterios de Evaluación para Trabajo Práctico 3.

<b>a.) Programa entregado.</b>	Entrega un archivo que no es Python, o es Python pero no ejecuta (Esta casilla implica reprobar directamente el TP1, marcando en el mínimo posible el resto de las casillas de esta tabla). <b>-1 puntos</b>	Entrega un programa ejecutable en Python. <b>0 puntos</b>	
--------------------------------	---	--	--

b.) Entrega en fecha.	Entrega tardía entre las 48hs y la 72hs luego del vencimiento o. <b>-3 puntos</b>	Entrega tardía entre las 24hs y las 48hs luego del vencimiento o. <b>-2 puntos</b>	Entrega tardía dentro de las 24hs luego del vencimiento o. <b>-1 punto</b>	Entrega en término. <b>0 puntos</b>
c.) Entrega del archivo comprimido.	No cumple convenciones solicitadas para el nombre del archivo comprimido. <b>-1 puntos</b>	Cumple convenciones solicitadas para el nombre del archivo comprimido. <b>0 puntos</b>		

d.) Uso de al menos dos módulos (uno para la clase y otro para el programa principal).	Todo el programa se realiza en un único módulo. <b>0 puntos</b>	Divide correctamente en módulos e importa correctamente. <b>1 puntos</b>		
e.) Definición de la clase requerida.	Mal o muy mal. No define correctamente la clase o no directamente no la define. <b>0 puntos</b>	Regular. <b>1 punto</b>	Correcto. <b>2 puntos</b>	No se define una clase con el constructor y el método __str__.



<b>f.) Estilo de escritura de código fuente (Es simple: prestar atención a los subrayados en gris que el IDE remarque en el código fuente).</b>	Muy mal. Muy impropio. No respeta convenciones. <b>0 puntos</b>	Regular. <b>1 punto</b>	Muy bien. <b>2 puntos</b>	
<b>g.) Menú de opciones.</b>	Mal o muy mal. No realizado, o no funciona. <b>0 puntos</b>	Regular. <b>1 punto</b>	Muy bien. <b>2 puntos</b>	
<b>h.) Interfaz de usuario (se controla también aquí la ortografía en los mensajes de salida).</b>	Muy pobre. <b>0 puntos</b>	Regular. <b>1 punto</b>	Muy bien. <b>2 puntos</b>	Faltan acentos. Es ticket, no tiket.

<b>i.) Opción 1: Carga correcta del archivo de texto y generación correcta del arreglo.</b>	Mal o muy mal. No genera el arreglo o no lo hace con objetos.          <b>0 puntos</b>	Regular. <b>1 punto</b>	Muy bien. <b>2 puntos</b>	En cada iteración se define una nueva clase tiket. Cada objeto en el arreglo termina siendo una instancia de una clase diferente. Nunca se cierra el archivo.
<b>j.) Opción 2: Carga manual del arreglo (incluye procesos de validación).</b>	Mal o muy mal. No realizado o incompleto o no funciona.          <b>0 puntos</b>	Regular. <b>1 punto</b>	Muy bien. <b>2 puntos</b>	No se valida que el código de identificación sea un número entero. Para ingresar la cantidad de kilómetros, hay que ingresar los 3 dígitos. No permite ingresar por ejemplo: 22. De nuevo, se crea una clase vacía para resolver este punto.

<p><b>k.) Opción 3: Ordenar y mostrar el arreglo (incluye una línea por cada objeto, y que aparezca correctamente el nombre del país original de la patente).</b></p>	<p>Muy mal. No realiza do o incompleto o no funciona, o no muestra el nombre del país. <b>0 puntos</b></p>	<p>Mal. Muestra pero no ordena, ordena mal. <b>1 puntos</b></p>	<p>Regula r. Ordena y muestra pero ocupa más de dos líneas por objeto. <b>2 puntos</b></p>	<p>Muy bien. Ordena y muestra ocupa ndo hasta dos líneas por objeto. Muestra el nombre del país. <b>3 puntos</b></p>	<p>Se debería separar el ordenamiento de la visualización en dos funciones. Se hace el print en primer for del ordenamiento por lo que el último elemento nunca se va a mostrar. No se utiliza el método <code>__str__</code> de la clase, ni tampoco una función <code>to_string</code> que se encargue de recibir el objeto y mostrarlo. En el ordenamiento, se intercambian los códigos no los objetos.</p>
---	--	---	--	--	--

<p><b>I.) Opción 4: Búsqueda de primera ocurrencia, por patente.</b></p>	<p>Muy mal. No desarrollado o no funciona.</p> <p><b>0 puntos</b></p>	<p>Mal. Busca pero aplica el algoritmo equivocado.</p> <p><b>1 punto</b></p>	<p>Regular. No corta cuando encuentra lo que busca o resuelve mal cuando no existe.</p> <p><b>2 puntos</b></p>	<p>Muy bien.</p> <p><b>3 puntos</b></p>	<p>No encuentra los tickets que fueron cargados de manera manual ya que en la carga manual, el país de la cabina es un entero y cuando se le pide al usuario el país a buscar se toma un string. No busca en todo el arreglo. Busca hasta n-1.</p>
--	---	--	--	---	--

<p><b>m.) Opción 5: Búsqueda de primera ocurrencia, por código (incluye actualización del contenido del objeto encontrado).</b></p>	<p>Muy mal. No desarrollado o no funciona. <b>0 puntos</b></p>	<p>Mal. Busca pero aplica el algoritmo equivocado, o no actualiza el contenido al encontrarlo. <b>1 punto</b></p>	<p>Regular. No corta cuando encuentra lo que busca o resuelve mal cuando no existe. <b>2 puntos</b></p>	<p>Muy bien. <b>3 puntos</b></p>	<p>Independientemente si el arreglo tiene un objeto o más, la lógica debería ser la misma. Está de más el condicional si el arreglo tiene un solo objeto. No se busca en todo el arreglo, se está buscando hasta n-1.</p>
---	--	---	---	--------------------------------------	---

<p><b>n.) Opción 6: Conteo por país de origen (vector de conteo, o nada: fueron avisados).</b></p>	<p>Muy mal. No realizado o incompleto o no funciona, o no usó un vector de conteo.</p> <p><b>0 puntos</b></p>	<p>Mal. Cuenta pero en forma incorrecta o usa malos índices.</p> <p><b>1 punto</b></p>	<p>Regular (a criterio del profesor).</p> <p><b>2 puntos</b></p>	<p>Muy bien.</p> <p><b>3 puntos</b></p>	<p>Independientemente si el arreglo tiene un objeto o más, la lógica debería ser la misma. No se está recorriendo el arreglo completo, se recorre hasta n-1.</p>
--	---	--	--	---	--

<p>ñ.) Opción 7: Acumulación de importes por tipo de vehículo (vector de acumulación, o nada: fueron avisados).</p>	<p>Muy mal. No realiza do o incompleto o no funciona, o no usó un vector de acumulación.</p> <p><b>0 puntos</b></p>	<p>Mal. Cuenta pero en forma incorrecta o usa los índices.</p> <p><b>1 punto</b></p>	<p>Regular (a criterio del profesor).</p> <p><b>2 puntos</b></p>	<p>Muy bien.</p> <p><b>3 puntos</b></p>	<p>Se rompe el programa si se ingresa a esta opción sin haber cargado los tickets previamente. Lo mismo acá, la lógica debería ser la misma independientemente cuantos objetos tenga el arreglo.</p>
---	---	--	--	---	--

<p><b>o.) Opción 8: Tipo de vehículo con mayor monto acumulado, y porcentaje sobre el total. (a realizar sobre el vector obtenido en la Opción 7).</b></p>	<p>Muy mal. No realiza do o incompleto o no funciona. <b>0 puntos</b></p>	<p>Mal. No trabaja sobre el vector de la Opción 7. <b>1 punto</b></p>	<p>Regular (a criterio del profesor). <b>2 puntos</b></p>	<p>Muy bien. <b>3 puntos</b></p>	<p>No se recorren todos los tickets.</p>
<p><b>p.) Opción 9: Distancia promedio y cantidad de distancias mayores a ese promedio.</b></p>	<p>Muy mal. No realiza do o incompleto o no funciona. <b>0 puntos</b></p>	<p>Mal. Calcula mal el promedio o cuenta mal. <b>1 punto</b></p>	<p>Regular (a criterio del profesor). <b>2 puntos</b></p>	<p>Muy bien. <b>3 puntos</b></p>	<p>Si el vector tiene un solo objeto, se rompe el programa. Se intenta acceder a <math>v[i]</math> mientras que <math>i</math> no está definida. Además de no recorrer todo el arreglo para sumar todas las distancias, cuando se calcula el promedio se divide por <math>n</math> (y se sumaron <math>n-1</math> distancias) por lo que el promedio no es correcto.</p>

2 (47,06 %)



**Calificado  
sobre**

domingo, 24 de septiembre de 2023, 22:12

**Calificado por**



Federico Javier Bertola

◀ [Desafío 03 \[Problema: Conteo de Frecuencias y Valor Modal\]](#)

Ir a...

[Guía 19 de Ejercicios Prácticos](#) ▶

✉ [Contactar con el soporte del sitio](#)

Usted se ha identificado como Juan Manuel Coro Reader (Cerrar sesión)

[AED \(2023\)](#)

[Resumen de retención de datos](#)

[Descargar la app para dispositivos móviles](#)