

### Indicaciones específicas:

- Esta evaluación contiene 12 páginas (incluyendo esta página) con 4 preguntas. El total de puntos son 20.
- El tiempo límite para la evaluación es 100 minutos.
- Crea la carpeta de nombre PC1.
- Crea el proyecto utilizando Pycharm
- Adiciona uno a uno los programas que dan respuesta a cada pregunta planteada
- Cada pregunta deberá ser respondida en un solo archivo con el número de la pregunta. Por ejemplo:
  1. p1.py
  2. p2.py
  3. p3.py
  4. p4.py
- Recuerda que el Gradescope solo conserva el último envío que se realiza, por lo tanto una vez que tengas las 4 preguntas resueltas, **deberás arrastrar los 4 archivos de manera simultánea y subirlos al Gradescope.**  
[www.gradescope.com](http://www.gradescope.com)
- **Para asignar el puntaje total a cada pregunta, es indispensable que en la solución se utilice las estructuras y/o conceptos que se indican en la rúbrica.**

### Criterios de desempeño:

- Para los alumnos de las carreras de **Ciencia de la Computación y Ciencia de Datos**:
  - 1.3 (nivel 1): Aplicar conocimientos de computación apropiados para la solución de problemas definidos y sus requerimientos en la disciplina del programa.
  - 3.2 (nivel 1): Diseñar, implementar y evaluar soluciones a problemas complejos de computación.

- 4.1 (nivel 1): Crear, seleccionar, adaptar y aplicar técnicas, recursos y herramientas modernas para la práctica de la computación y comprende sus limitaciones.
  - Para los alumnos de las carreras de **Ingeniería:**
    - 1.3 (nivel 1): Aplica conocimientos de ingeniería en la solución de problemas complejos de ingeniería.
    - 3.2 (nivel 1): Diseña soluciones relacionadas a problemas complejos de ingeniería .
    - 5.1 (nivel 1): Crea, selecciona y utiliza técnicas, habilidades, recursos y herramientas modernas de la ingeniería y las tecnologías de la información, incluyendo la predicción y el modelamiento, con la comprensión de sus limitaciones.
  - Para los alumnos de la carrera de **Administración y Negocios Digitales**
    - 1.1 (nivel 1): Analizar información verbal y/o lógica proveniente de distintas fuentes, encontrando relaciones y presentándola de manera clara y concisa.
    - 2.3 (nivel 1): Resolver problemas pensando computacionalmente y empleando herramientas de programación
    - 4.5 (nivel 1): Integrar habilidades analíticas, digitales e interpersonales para el diseño de soluciones a problemas relevantes de personas y organizaciones.
- 

## Calificación:

Tabla de puntos (sólo para uso del professor)

Question	Points	Score
1	5	
2	5	
3	5	
4	5	
Total:	20	

## 1. (5 points) Evalúa uso de expresiones

La embotelladora Socosani, ofrece a sus clientes agua mineral en las siguientes presentaciones:

- Caja de 20 litros
- Bidón de 7 litros
- Botella de 2 litros
- Botella de 600 ml

El gerente de planta, requiere contar con un programa en el que dado como dato la cantidad de mililitros (ml) de agua, el programa indique cuantas unidades de cada presentación se puede embotellar y cuántos mililitros sobran. Recuerde que 1 litro tiene 1000 mililitros.

Para poder asignar la totalidad del puntaje a esta pregunta, es necesario que **realice el programa sin utilizar estructuras de control**.

Algunos ejemplos de diálogo de este programa serían:

Listing 1: Ejemplo 1

```
Militros : 29800

Cajas de 20 litros      : 1
Bidones de 7 litros    : 1
Botellas de 2 litros   : 1
Botellas de 600 ml     : 1
Mililitros que sobran  : 200
```

Listing 2: Ejemplo 2

```
Militros : 31200

Cajas de 20 litros      : 1
Bidones de 7 litros    : 1
Botellas de 2 litros   : 2
Botellas de 600 ml     : 0
Mililitros que sobran  : 200
```

Listing 3: Ejemplo 3

```
Militros : 55000

Cajas de 20 litros      : 2
Bidones de 7 litros    : 2
Botellas de 2 litros   : 0
```

Botellas de 600 ml	:	1
Mililitros que sobran	:	400

Listing 4: Ejemplo 3

Mililitros : 2345		
Cajas de 20 litros	:	0
Bidones de 7 litros	:	0
Botellas de 2 litros	:	1
Botellas de 600 ml	:	0
Mililitros que sobran	:	345

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Algoritmo y codificación (4 pts)	Elabora un algoritmo preciso, definido y finito que da solución exacta a lo que el enunciado requiere. <b>Construye expresiones</b> para codificar el algoritmo y lo hace con el 100% de precisión. <b>(4pts)</b>	Elabora un algoritmo preciso, definido y finito que da solución al menos al 80 % de lo que el enunciado requiere. Construye expresiones para codificar el algoritmo y lo hace con al menos el 80% de precisión. <b>(3pts)</b>	Elabora un algoritmo preciso, definido y finito que da solución al menos al 65 % de lo que el enunciado requiere. Construye expresiones para codificar el algoritmo y lo hace con al menos el 65% de precisión. <b>(2pts)</b>	Elabora un algoritmo que hace menos del 65% de lo que el enunciado requiere. Construye expresiones para codificar el algoritmo y lo hace con menos del 65% de precisión. <b>(0pts)</b>
Sintaxis y legibilidad (1 pt)	El algoritmo es correcto, y es codificado sin errores de sintaxis. El nombre de las variables y funciones son descriptivas. <b>(1pts)</b>	El algoritmo es correcto, y es codificado con algunos errores de sintaxis, pero que no afectan el resultado de manera significativa. El nombre de las variables y funciones son descriptivas. <b>(0.75pts)</b>	El algoritmo es correcto, y es codificado con algunos errores de sintaxis, que afectan el resultado de manera mínima, o el nombre de las variables y funciones no son descriptivas. <b>(0.5pts)</b>	El algoritmo es incorrecto o es codificado con errores de sintaxis, que afectan el resultado de manera significativa. El nombre de las variables y funciones no son descriptivas. <b>(0pts)</b>

2. (5 points) Evalúa estructuras de control selectivas.

Cuenta una leyenda, que según el mes y día en que una persona nace se le asigna una piedra preciosa, que le dará buena suerte a lo largo de toda su vida.

Desarrolle un programa que pida al usuario el nombre del mes y el año en que nació, para que luego el programa indique, cuál es la piedra preciosa que le corresponde considerando la siguiente tabla:

Mes	Año Par	Año Impar
Enero, Febrero, Marzo, Abril	Esmeralda	Zafiro
Mayo, Junio, Julio, Agosto	Rubí	Topacio
Setiembre, Octubre, Noviembre, Diciembre	Amatista	Jade

- Si los datos no corresponden, el programa dará un mensaje de error.

Algunos ejemplos de diálogo de este programa serían:

Listing 5: Ejemplo 1

```
Nombre del mes : Setiembre
Anio : 2005
Su piedra preciosa es : Jade
```

Listing 6: Ejemplo 2

```
Nombre del mes : Agosto
Anio : 1999
Su piedra preciosa es : Topacio
```

Listing 7: Ejemplo 3

```
Nombre del mes : Junio
Anio: 2008
Su piedra preciosa es : Rubi
```

Listing 8: Ejemplo 4

```
Nombre del mes : Jupiter
Anio : 2020
Los datos no corresponden a una fecha
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Algoritmo y codificación (4 pts)	Elabora un algoritmo preciso, definido y finito que da solución exacta a lo que el enunciado requiere. <b>Utiliza estructuras de control selectivas</b> para codificar el algoritmo y lo hace con el 100% de precisión. <b>(4pts)</b>	Elabora un algoritmo preciso, definido y finito que da solución al menos al 80 % de lo que el enunciado requiere. Utiliza estructuras de control selectivas para codificar el algoritmo y lo hace con al menos el 80% de precisión. <b>(3pts)</b>	Elabora un algoritmo preciso, definido y finito que da solución al menos al 65 % de lo que el enunciado requiere. Utiliza estructuras de control selectivas para codificar el algoritmo y lo hace con al menos el 65% de precisión. <b>(2pts)</b>	Elabora un algoritmo que hace menos del 65% de lo que el enunciado requiere. Utiliza estructuras de control selectivas para codificar el algoritmo y lo hace con menos del 65% de precisión. <b>(0pts)</b>
Sintaxis y legibilidad (1 pt)	El algoritmo es correcto, y es codificado sin errores de sintaxis. El nombre de las variables y funciones son descriptivas. <b>(1pts)</b>	El algoritmo es correcto, y es codificado con algunos errores de sintaxis, pero que no afectan el resultado de manera significativa. El nombre de las variables y funciones son descriptivas. <b>(0.75pts)</b>	El algoritmo es correcto, y es codificado con algunos errores de sintaxis, que afectan el resultado de manera mínima, o El nombre de las variables y funciones no son descriptivas. <b>(0.5pts)</b>	El algoritmo es incorrecto o es codificado con errores de sintaxis, que afectan el resultado de manera significativa. El nombre de las variables y funciones no son descriptivas. <b>(0pts)</b>

3. (5 points) Evalúa estructuras de control repetitivas.

Un número es Perfecto, cuando el número es igual a la suma de sus divisores positivos menores que él.

El 6 es un número perfecto porque la suma de sus divisores  $1 + 2 + 3$  es igual a 6.

El 28 es un número Perfecto, porque la suma de sus divisores  $1 + 2 + 4 + 7 + 14$  es igual a 28

Realice un programa que permita:

- Leer como dato un número desde 1 a 9000. Dato que el programa tiene que validar.
- El programa debe indicar si el número es un "Numero Perfecto" o "No es un número perfecto".

Algunos ejemplos de diálogo de este programa serían:

Listing 9: Ejemplo 1

```
Numero [1-9000]: 12456
Numero [1-9000]: 10000
Numero [1-9000]: 28
Es un numero perfecto
```

Listing 10: Ejemplo 2

```
Numero [1-9000]: 496
Es un numero perfecto
```

Listing 11: Ejemplo 3

```
Numero [1-9000]: 150
No es un numero perfecto
```

Listing 12: Ejemplo 4

```
Numero [1-9000]: 8128
Es un numero perfecto
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Algoritmo y codificación (4 pts)	Elabora un algoritmo preciso, definido y finito que da solución exacta a lo que el enunciado requiere. <b>Utiliza estructuras de control repetitivas</b> para codificar el algoritmo y lo hace con el 100% de precisión. <b>(4pts)</b>	Elabora un algoritmo preciso, definido y finito que da solución al menos al 80 % de lo que el enunciado requiere. Utiliza estructuras de control repetitivas para codificar el algoritmo y lo hace con al menos el 80% de precisión. <b>(3pts)</b>	Elabora un algoritmo preciso, definido y finito que da solución al menos al 65 % de lo que el enunciado requiere. Utiliza estructuras de control repetitivas para codificar el algoritmo y lo hace con al menos el 65% de precisión. <b>(2pts)</b>	Elabora un algoritmo que hace menos del 65% de lo que el enunciado requiere. Utiliza estructuras de control repetitivas para codificar el algoritmo y lo hace con menos del 65% de precisión. <b>(0pts)</b>
Sintaxis y legibilidad (1 pt)	El algoritmo es correcto, y es codificado sin errores de sintaxis. El nombre de las variables y funciones son descriptivas. <b>(1pts)</b>	El algoritmo es correcto, y es codificado con algunos errores de sintaxis, pero que no afectan el resultado de manera significativa. El nombre de las variables y funciones son descriptivas. <b>(0.75pts)</b>	El algoritmo es correcto, y es codificado con algunos errores de sintaxis, que afectan el resultado de manera mínima, o el nombre de las variables y funciones no son descriptivas. <b>(0.5pts)</b>	El algoritmo es incorrecto o es codificado con errores de sintaxis, que afectan el resultado de manera significativa. El nombre de las variables y funciones no son descriptivas. <b>(0pts)</b>



4. (5 points) Evalúa estructuras de control selectivas y repetitivas.

En el gripo "La esquina", se vende gasolina de: 84 octanos, 90 octanos, 95 octanos y 97 octanos. El dueño el grifo desea tener un programa que permita registrar la cantidad de galones de cada tipo de gasolina que sus clientes prefieren. Se pide que realice un programa que permita leer la cantidad de clientes y para cada uno saber que tipo de gasolina comprará y cuantos galones. Para que la finalizar el día el programa muestre la cantidad de galones vendidos de cada tipo de gasolina.

Si un cliente solicita un tipo de gasolina que no existe, se dará un mensaje apropiado.

Algunos ejemplos de diálogo de este programa serían:

Listing 13: Ejemplo 1

```
Numero de clientes : 5

Atendiendo al cliente numero 1
Gasolina de 97, 95, 90 o 84 octanos ? 97
Numero de galones : 12

Atendiendo al cliente numero 2
Gasolina de 97, 95, 90 o 84 octanos ? 95
Numero de galones : 10

Atendiendo al cliente numero 3
Gasolina de 97, 95, 90 o 84 octanos ? 90
Numero de galones : 10

Atendiendo al cliente numero 4
Gasolina de 97, 95, 90 o 84 octanos ? 84
Numero de galones : 5

Atendiendo al cliente numero 5
Gasolina de 97, 95, 90 o 84 octanos ? 99
Numero de galones : 9
No hay ese tipo de gasolina!

Galones vendidos de cada tipo
Galones de 97 octanos : 12
Galones de 95 octanos : 10
Galones de 90 octanos : 10
Galones de 84 octanos : 5
```

Listing 14: Ejemplo 2

```
Numero de clientes : 3
```

```
Atendiendo al cliente numero 1
Gasolina de 97, 95, 90 o 84 octanos ? 97
Numero de galones : 20
```

```
Atendiendo al cliente numero 2
Gasolina de 97, 95, 90 o 84 octanos ? 97
Numero de galones : 5
```

```
Atendiendo al cliente numero 3
Gasolina de 97, 95, 90 o 84 octanos ? 90
Numero de galones : 14
```

```
Galones vendidos de cada tipo
Galones de 97 octanos : 25
Galones de 95 octanos : 0
Galones de 90 octanos : 14
Galones de 84 octanos : 0
```

#### Listing 15: Ejemplo 3

```
Numero de clientes : 7

Atendiendo al cliente numero 1
Gasolina de 97, 95, 90 o 84 octanos ? 90
Numero de galones : 10

Atendiendo al cliente numero 2
Gasolina de 97, 95, 90 o 84 octanos ? 84
Numero de galones : 5

Atendiendo al cliente numero 3
Gasolina de 97, 95, 90 o 84 octanos ? 84
Numero de galones : 7

Atendiendo al cliente numero 4
Gasolina de 97, 95, 90 o 84 octanos ? 45
Numero de galones : 3
No hay ese tipo de gasolina!

Atendiendo al cliente numero 5
Gasolina de 97, 95, 90 o 84 octanos ? 90
Numero de galones : 4

Atendiendo al cliente numero 6
Gasolina de 97, 95, 90 o 84 octanos ? 90
```

```
Numero de galones : 4

Atendiendo al cliente numero 7
Gasolina de 97, 95, 90 o 84 octanos ? 97
Numero de galones : 10

Galones vendidos de cada tipo
Galones de 97 octanos : 10
Galones de 95 octanos : 0
Galones de 90 octanos : 18
Galones de 84 octanos : 12
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Algoritmo y codificación (4 pts)	Elabora un algoritmo preciso, definido y finito que da solución exacta a lo que el enunciado requiere. <b>Utiliza estructuras de control selectivas y repetitivas</b> para codificar el algoritmo y lo hace con el 100% de precisión. <b>(4pts)</b>	Elabora un algoritmo preciso, definido y finito que da solución al menos al 80 % de lo que el enunciado requiere. Utiliza estructuras de control selectivas y repetitivas para codificar el algoritmo y lo hace con al menos el 80% de precisión. <b>(3pts)</b>	Elabora un algoritmo preciso, definido y finito que da solución al menos al 65 % de lo que el enunciado requiere. Utiliza estructuras de control selectivas y repetitivas para codificar el algoritmo y lo hace con al menos el 65% de precisión. <b>(2pts)</b>	Elabora un algoritmo que hace menos del 65% de lo que el enunciado requiere. Utiliza estructuras de control selectivas y repetitivas para codificar el algoritmo y lo hace con menos del 65% de precisión. <b>(0pts)</b>
Sintaxis y legibilidad (1 pt)	El algoritmo es correcto, y es codificado sin errores de sintaxis. El nombre de las variables y funciones son descriptivas. <b>(1pts)</b>	El algoritmo es correcto, y es codificado con algunos errores de sintaxis, pero que no afectan el resultado de manera significativa. El nombre de las variables y funciones son descriptivas. <b>(0.75pts)</b>	El algoritmo es correcto, y es codificado con algunos errores de sintaxis, que afectan el resultado de manera mínima, o el nombre de las variables y funciones no son descriptivas. <b>(0.5pts)</b>	El algoritmo es incorrecto o es codificado con errores de sintaxis, que afectan el resultado de manera significativa. El nombre de las variables y funciones no son descriptivas. <b>(0pts)</b>