

**FACULTAD DE INGENIERÍA**  
**10110 - FUNDAMENTOS DE COMPUTACION Y PROGRAMACION**  
**10145 - FUNDAMENTOS DE PROGRAMACIÓN PARA INGENIERÍA**

**PRUEBA PARCIAL PROGRAMADA N° 2 – SERIE 1**

**INSTRUCCIONES**

**Antes de comenzar, lea atentamente las instrucciones, los requerimientos de entrega y los problemas. Tenga en cuenta que:**

- La PEP consta de 1 pregunta de desarrollo, con un total de 60 puntos y una exigencia del 60% (36 puntos para nota 4,0).
- La prueba está pensada para 90 minutos de duración. Dispone, no obstante, de un tiempo límite de 180 minutos para responder.
- Durante el período de evaluación el estudiante solo puede resolver dudas con el cuerpo docente del curso, es decir, profesores y coordinadores. Consultas a otras personas, como, por ejemplo: tutores, compañeros de su sección u otra, personas que ya rindieron el curso, etc. se considera como una falta a la ética y significará sanción de nota mínima.
- Durante el período de evaluación el estudiante solo puede usar Python, el resumen que la coordinación provee y Campus Virtual, este último exclusivamente para descargar este enunciado, los archivos de prueba asociados y hacer su entrega, la revisión de otros materiales, apuntes o código propio será sancionada con la nota mínima.
- La evaluación es de carácter individual. Cualquier indicio de intervención de otra persona será calificado con nota mínima a la evaluación y quedará impedido de rendir cualquier instancia de evaluación optativa. Adicionalmente deberá rendir las siguientes evaluaciones en forma presencial.
- En caso de detectar programas iguales o con suficientes similitudes entre sí, tanto en una misma sección como en secciones distintas, se calificará con la nota mínima a todos los involucrados, quienes quedarán impedidos de rendir cualquier instancia de evaluación optativa y con la obligación de rendir las evaluaciones restantes en forma presencial. Para ello se compararán las entregas con un algoritmo de reconocimiento de texto y se revisarán los casos sospechosos mediante un proceso de validación visual que realizará el cuerpo docente del curso.
- Estudiantes que entreguen fuera de plazo serán calificados con nota mínima en esta evaluación.
- Entregas que se realicen por vías distintas a Campus Virtual serán calificados con nota mínima. En caso de que la plataforma presente un problema, puede enviar su archivo como respaldo al correo de contacto del profesor de Teoría, y posteriormente subir el archivo a Campus Virtual.
- Estudiantes que entreguen sin seguir los criterios de identificación estipulados en el apartado de “entrega” serán calificados con nota mínima en esta evaluación.
- La difusión de este enunciado, solicitando ayuda, ofreciendo dinero a cambio de la resolución e incluso la publicación de éste dentro del plazo de desarrollo de la prueba, se considera un acto deshonesto, que daña la imagen de la Facultad de Ingeniería y de la Universidad de Santiago de Chile y significará la reprobación inmediata de la asignatura y la aplicación de los procedimientos disciplinarios que el reglamento de ética de la Universidad dispone para estos casos.
- Considere que cualquier supuesto que el estudiante haga debe ser explicitado en los comentarios de código.

- La subida del archivo es responsabilidad de su autor, por lo que, archivos que no estén en el formato estipulado, que vengan corruptos o con problemas para ser leídos, no serán revisados.
- La cuenta @usach.cl y los accesos a los sistemas que provee son personales e intransferibles. No se pueden prestar, ni dejarse suplantar. Se considera al estudiante como el único responsable por resguardar la privacidad de sus entregas
- Cualquier situación no contemplada en este documento será dirimida por la Facultad de Ingeniería.

## ENTREGA

Se recibirán soluciones hasta las 12:45 del día 2 de diciembre de 2022 en la plataforma Campus Virtual, en el espacio habilitado para ello en la pestaña “Evaluaciones -> Pruebas”. Se requiere entregar un único archivo **.py** con la solución del problema. El archivo debe llevar por nombre el RUT de su autor (sin puntos ni guion) por ejemplo:

19984321K.py

Además, al inicio del archivo .py, se debe añadir el siguiente encabezado del programa, con los datos solicitados para identificar su trabajo (rellene el encabezado del programa con sus datos personales en formato IDÉNTICO al indicado en el ejemplo del código a continuación):

```
# FUNDAMENTOS DE PROGRAMACIÓN PARA INGENIERÍA
# SECCIÓN DEL CURSO: 2-L-1
# PROFESOR DE TEORÍA: FELIPE MORENO
# PROFESOR DE LABORATORIO: JUAN A. PADILLA
#
# AUTOR
# NOMBRE: Juan Carlos Pérez González
# RUT: 20.345.432-2
# CARRERA: Ingeniería Civil Mecánica
# <Incluya aquí una breve descripción del programa>1
```

<CONTINÚE CON EL PROGRAMA A PARTIR DE AQUÍ><sup>1</sup>

Código 1: Ejemplo de encabezado

---

<sup>1</sup> El texto incluido dentro de los símbolos “<” y “>” debe ser reemplazado la información solicitada respectiva, sin incluir los símbolos “<” y “>”.



## RESTRICCIONES DE IMPLEMENTACIÓN

Para esta prueba, se prohíbe el uso de las siguientes herramientas de programación:

- Módulos no vistos en clases. **Se permiten, sin embargo, los módulos numpy, pandas y matplotlib.**
- Construcciones que usen las palabras reservadas `try` y `except`.
- Expresiones regulares.
- Los tipos de dato conjunto (`set`), tupla (`tuple`) y diccionario (`dict`).
- Programación orientada a objetos, es decir, definición de clases y métodos.
- El uso de las funciones `eval` y `exec`.

El **uso** de alguna de estas herramientas significará la **nota mínima**.

## CRITERIOS DE EVALUACIÓN

Los aspectos que evaluar son:

- Abstracción de datos y de procesos.
- Buenas prácticas de programación.
- Manejo de estructuras de control.
- Manejo de tipos de datos.
- Cálculo de la solución.

## 1. (60 puntos) En la pista de carreras 2

La famosa empresa Mintiendo™ ha decidido realizar un importante evento en línea a fin de motivar a sus clientes antes del lanzamiento de una nueva versión del popular videojuego María Carting: María Carting DePelos. El equipo de marketing ha decidido que el evento consista en una carrera de 5 vueltas a un extenso circuito.

El sistema en línea genera dos grandes archivos de texto: `participantes.csv` y `tiempos.csv`. El primero contiene un listado de los participantes, registrando: apodo, país, identificador único de participación (entero, asignado por orden de inscripción) y correo de contacto. El segundo registra el identificador único de participación, el número de vuelta y el tiempo (en horas, minutos, segundos y milisegundos) que tarda el competidor en completar dicha vuelta al circuito (tenga en cuenta que los jugadores pueden abandonar el evento antes de completarlo, por lo que únicamente aquellos participantes que registren tiempo para cada una de las cinco vueltas habrán completado la carrera).

Ejemplo:

**Participantes.csv (apodo;identificador;correo):**

```
Diego;CL;1;dondi@algo.xd
Asgaard;AR;2;vikingo_logo@odin.nw
Chika_Ruda;EN;3;frutillita@jmay1.kon
```

...

**Tiempos.csv (identificador;nº vuelta;tiempo vuelta):**

```
3;1;2:03:05,872
1;1;2:12:16,831
4;1;2:37:51,371
3;2;1:47:57,037
```

...

A fin de determinar a los ganadores y, al mismo tiempo, obtener información valiosa para nuevos videojuegos de la saga, el gerente de marketing de Mintiendo™ le ha solicitado que construya un programa en Python que realice las siguientes tareas:

- Generar, para cada país participante, el archivo `resultados-XX.csv`, el cual debe registrar: apodo del competidor, correo de contacto y tiempo total empleado para completar las 5 vueltas. Dicho archivo NO debe incluir a los participantes que abandonan antes de completar la carrera. Adicionalmente, este archivo debe estar ordenado de manera creciente según el tiempo total.
- Generar el archivo `ganadores.csv`, el cual debe registrar a los 10 mejores competidores (que completan la carrera) a nivel global, indicando apodo del competidor, país, correo de contacto y tiempo total empleado para completar las 5 vueltas.
- Mostrar por pantalla el mejor tiempo para completar **una única vuelta** al circuito y el apodo del(los) jugador(es) que logra(n) dicha marca, sin importar si el jugador completó la carrera.

Tenga en cuenta lo siguiente:

- Su programa debe funcionar para **cualquier cantidad de jugadores**.
- Asuma que los países se representan con códigos de dos letras mayúsculas y que los códigos siempre serán correctos.
- El archivo de tiempo siempre tendrá registros válidos, es decir, no contiene vueltas repetidas para un competidor ni registros faltantes (es decir, para cada participante se registran vueltas consecutivas).