Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2233 Programación Avanzada (2022-1)

Tarea 0

Entrega

- **Tarea**
  - **Fecha y hora:** viernes 25 de marzo de 2022, 20:00
  - **Lugar:** Repositorio personal de GitHub — Carpeta: `Tareas/T0/`
- `README.md`
  - **Fecha y hora:** viernes 25 de marzo de 2022, 22:00
  - **Lugar:** Repositorio personal de GitHub — Carpeta: `Tareas/T0/`

Objetivos

- Aplicar competencias asimiladas en *Introducción a la Programación* para el desarrollo de una solución a un problema.
- Familiarizarse con el proceso de entrega de tareas y uso de buenas prácticas de programación.
- Procesar *input* del usuario de forma robusta, manejando potenciales errores de formato.
- Trabajar con archivos de texto para leer, escribir y procesar datos.
- Escribir código utilizando paquetes externos (*i.e.* código no escrito por el estudiante), como por ejemplo, módulos que pertenecen a la biblioteca estándar de Python.

Índice

1. *DCCorreos de Chile*

La Empresa de *DCCorreos de Chile* se dedica al servicio de correspondencia y al mercado de envíos de encomiendas tanto nacionales como internacionales. Producto de la pandemia, esta agencia ha cobrado mayor relevancia y los usuarios han reclamado acerca del mal servicio que ofrecen. Por ello, han decidido actualizar su plataforma, de modo tal que los usuarios puedan realizar encomiendas, ver el estado de sus pedidos, y realizar reclamos a la agencia. Por esta razón, el Departamento de Ciencia de la Computación (DCC) te ha elegido a ti para que crees la plataforma *DCCorreos de Chile*: un nuevo sitio que permita a todo tipo de usuario, interactuar con la encomienda de forma cómoda y rápida.

Figura 1: Logo de *DCCorreos de Chile*

2. Flujo del programa

DCCorreos de Chile es una plataforma de envío de encomiendas. Existen dos tipos de usuario: usuario registrado y administrador. Como Usuario registrado se pueden hacer encomiendas, realizar reclamos y ver el estado de los pedidos. Por otro lado, al ingresar como Administrador se pueden actualizar encomiendas y revisar los reclamos. La ejecución de este programa debe ser por consola, por lo que debes cuidar que todas las instrucciones y mensajes se visualicen de manera correcta.

Al iniciar el programa, se muestra el Menú de Inicio en la consola, donde podrás ingresar con tu usuario registrado, registrar tu nuevo usuario o ingresar como administrador. En caso de seleccionar la primera opción, se debe ingresar el usuario y verificar su existencia en `usuarios.csv`, para luego dar paso al Menú de Usuario. En el caso de que se escoja la opción de registrarse como un nuevo usuario, te pedirá el nombre y contraseña, debes asegurarte que cumpla con las condiciones detalladas en Menú de Inicio. Luego de eso, se procede a visualizar el Menú de Usuario en donde podrás hacer encomiendas, revisar estado de encomiendas realizadas, realizar reclamos, ver el estado de los pedidos personales y cerrar sesión. Tanto los usuarios ingresados como los recién registrados podrán observar este menú. Si se ingresa como administrador con su respectiva contraseña, el programa debe redirigir directamente al Menú de Administrador en donde se visualizan las opciones de actualizar encomiendas, revisar reclamos y de cerrar sesión.

La plataforma debe funcionar de tal manera que siempre se pueda volver al menú anterior.

3. Entidades

En DCCorreos de Chile existen distintas entidades necesarias para el funcionamiento de la plataforma. Estas corresponden a las Encomiendas, Reclamos, Usuarios registrados y Administrador.

3.1. Encomiendas

Las encomiendas corresponden a los pedidos que se trasladan de un lugar a otro a lo largo de Chile. Estas son realizadas por usuarios registrados en la plataforma, y están definidas por un **Nombre de articulo**, **Receptor**, **Peso**, **Destino** y **Estado**. Estas características deben cumplir con una serie de restricciones. El nombre del articulo y el destino no deben tener comas (`","`), el receptor debe ser un usuario registrado y el peso debe ser menor al parámetro `MAX_PESO`[1]. El estado de la encomienda depende del paso en el que se encuentre la encomienda y es actualizado por el administrador a medida que va avanzando desde `'Emitida'` a `'Revisada por agencia'` a `'En camino'` y por último a `'En destino'`.

3.2. Reclamos

Los reclamos corresponden a comentarios que pueden dejar los usuarios registrados en un buzón de reclamos, expresando algún problema que hayan tenido con la plataforma. Poseen un **Título** y una **Descripción**, y son revisados por el administrador quien accede a ellos a traves de una lista indexada donde aparecen los títulos de los reclamos y al elegir alguno aparece la descripción del respectivo reclamo.

```
** Buzón de Reclamos **

* Elija uno de los siguientes reclamos para visualizar su
descripción *

[1] Larga demora en llegar a destino
[2] Pequeña capacidad de peso
[3] Faltó revisión de la agencia
[0] Volver
```

(a) Ejemplo de buzon de reclamos.

```
* Reclamo *

-Título: Larga demora en llegar a destino
-Descripción: La encomienda enviada se demoró más de
un mes den pasar de estado 'En camino' a 'En destino'
```

(b) Ejemplo de reclamo.

3.3. Usuario registrado

Los usuarios registrados poseen dos características: un nombre de usuario y una contraseña, los cuales se encuentran en el archivo `usuarios.csv`. Al iniciar su sesión estos tendrán cuatro opciones. La primera corresponde a ingresar una encomienda, para lo cual deberán ingresar en orden los siguientes datos: **Nombre del artículo**, **Receptor**, **Peso** y **Destino**, los cuales deberán cumplir con una serie de restricciones detalladas en Menú de Usuario para que la encomienda sea creada. La segunda corresponde a revisar el estado de las encomiendas que ha realizado este usuario, donde se visualizará el detalle de todas sus encomiendas, con el respectivo nombre del articulo, receptor, peso, destino y el estado en el que se encuentra. La tercera acción corresponde a que el usuario puede realizar reclamos acerca de algún problema que haya tenido con el sistema de encomiendas ingresando un **Título** acerca de lo que trata el reclamo y una **Descripción** para explicar su problema. Y la cuarta corresponde a ver el estado de los pedidos personales, en donde se visualizará el detalle de las encomiendas en las que el receptor es el usuario actual.

Por último, se pueden crear nuevos usuarios, registrándose e ingresando un nombre de usuario y contraseña nueva con ciertos requisitos detallados en el Menú de Usuario. Los usuarios válidos deben agregarse al archivo `usuarios.csv`.

3.4. Administrador

El administrador, a diferencia de los usuarios registrados, es único y posee una única contraseña definida por el parámetro `ADMIN_CONTRASENA`. Para ingresar a la sesión del administrador hay que utilizar la opción **Iniciar sesión como administrador** e ingresar utilizando esta contraseña. Existen dos funciones particulares para el administrador. La primera corresponde a actualizar el estado de las encomiendas,

[1] Las palabras escritas en `ESTE_FORMATO` son parámetros que tendrás que definir e importar desde el archivo `parametros.py`

pasando de `'Emitida'` a `'Revisada por agencia'` a `'En camino'` y por último a `'En destino'`, es decir, a medida que la encomienda va avanzando a través del proceso de envío, el administrador oprimirá un botón de actualizar y dependiendo del paso en el que se encuentre el pedido, este avanzará al siguiente paso. Por ejemplo, si el estado de una encomienda es de `'Revisada por agencia'` y el administrador desea actualizarla, deberá cambiarse al estado `'En camino'`. En caso de que la encomienda se encuentre en el último paso (`'En destino'`) y el administrador nuevamente desea actualizarla, se deberá avisar que la opción es inválida. La segunda función corresponde a revisar los reclamos de los usuarios registrados acerca del funcionamiento del sistema de encomiendas, donde se podrá visualizar una lista indexada de reclamos junto con sus respectivos títulos, de tal forma que al elegir alguno aparezca la descripción del respectivo reclamo.

4. Menús

La ejecución de DCCorreos de Chile se realizará únicamente mediante la interacción en consola a través de diferentes menús, los cuales se mostrarán dependiendo de las opciones que se elijan. Todos los menús deberán ser a prueba de errores, es decir, tu programa no debe caerse al ingresar una opción inválida y debe informar sobre el error. El formato ejemplificado de los menús puede modificarse a gusto propio, mientras se mantengan las opciones mínimas y el flujo del programa no se vea alterado.

Se deberá implementar los siguientes menús:

4.1. Menú de Inicio

El primer menú se deberá mostrar al ejecutar el programa. Este tiene que tener las opciones de **Iniciar sesión como usuario**, **Registrarse como usuario**, **Iniciar sesión como administrador** y **Salir del programa**.

Al escoger la primera opción, se procede a iniciar la sesión de un usuario específico, donde se solicita el nombre de usuario y su respectiva contraseña. Se debe verificar que el nombre pertenezca a la lista de nombres entregados en el archivo `usuarios.csv` y que la contraseña coincida con la correspondiente al nombre del usuario. Si el nombre no está dentro de la lista o la contraseña no corresponde al usuario, se debe notificar el error respectivo y volver al Menú de Inicio. En caso contrario, se debe dar paso al Menú de Usuario.

La opción **Registrarse como usuario** solicita el ingreso de un nombre nuevo y una contraseña. El nombre debe tener `MIN_CARACTERES` (incluyéndolo) alfabéticos y no debe repetirse con los nombres del archivo `usuarios.csv`. La contraseña debe ser de un largo mínimo (inclusive) de `LARGO_CONTRASENA` con caracteres alfanuméricos[2] y puede repetirse con las contraseñas ya existentes [3]. En el caso de que exista algún error en los datos ingresados, se debe informar en específico el campo erróneo. Luego de tener un registro exitoso, se procede a visualizar el Menú de Usuario.

Por último, al **Iniciar sesión como administrador**, se pide la contraseña `ADMIN_CONTRASENA`. Esta contraseña no se encuentra en el archivo `usuarios.csv` y tampoco debe ser agregada a este, ya que está reservada para el administrador. Si el texto ingresado no coincide exactamente con la contraseña predeterminada, se debe notificar el error y mostrar las opciones de **Repetir contraseña** o **Volver** al Menú de Inicio.

[2] Alfanumérico hace referencia a los caracteres formados por letras (ya sean minúsculas o mayúsculas) sin tildes y/o dígitos numéricos.

[3] El almacenamiento de contraseñas de los usuarios en un archivo de texto se implementa en esta tarea por simplicidad, pero en la vida real es una muy mala y peligrosa idea este método. Más adelante en el curso se conocerá maneras de poder ocultar esta información.

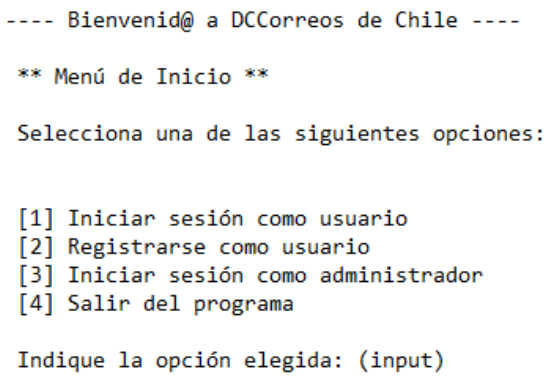

Figura 3: Ejemplo de Menú de Inicio.

4.2. Menú de Usuario

Este menú se muestra una vez ingresado o registrado el usuario. Contendrá las opciones de **Ingresar encomienda**, **Revisar estado de encomiendas realizadas**, **Realizar reclamos**, **Ver el estado de los pedidos personales** y **Cerrar sesión**. Esta última opción permitirá volver al Menú de Inicio. Todas las opciones de este menú deben permitir Volver al menú anterior con el propósito de tener una navegación fluida en el programa.

La opción de **Ingresar encomiendas** debe mostrar de forma secuencial los campos de información a rellenar. Primero se debe solicitar el **Nombre del artículo**, el cual no debe contener comas (`","`). Luego de llenar el campo y comprobar su formato, se debe pedir el **Nombre del destinatario**, el cual debe ser un usuario registrado de *DCCorreos de Chile*, presente en el archivo `usuarios.csv`. Luego de corroborar el dato, se rellena el peso, el cual debe tener un máximo de `MAX_PESO`. Finalmente, se pide ingresar el **Destino**, el cual no debe contener comas (`","`). Una vez ingresada toda la información de manera correcta, el programa debe informar el registro exitoso de la encomienda. En caso de no cumplir con el formato en cualquier campo solicitado, la consola debe indicar el error y ofrecer las opciones de: **Continuar** o **Cancelar encomienda**. Si se escoge continuar, se debe solicitar rellenar nuevamente el valor erróneo, de tal manera que se pueda llenar el resto de la información. Por otro lado, si se cancela la encomienda, se regresa al menú anterior.

La opción **Revisar el estado de encomiendas realizadas** muestra los atributos nombre y estado del artículo de las encomiendas del archivo `encomiendas.csv` que fueron realizadas por el usuario de la sesión. La información contenida en este archivo se encuentra en la entidad Encomiendas. Dicha información debe actualizarse cada vez que un usuario crea una encomienda nueva, de tal manera que al revizar las encomiendas, la consola muestre todo lo agregado en el archivo.

Por otra parte, los usuarios pueden realizar reclamos respecto al funcionamiento de la plataforma. Al seleccionar la opción de **Realizar reclamo**, se debe ingresar un título y posteriormente una descripción del reclamo. Una vez completados, se guardará en el archivo `reclamos.csv` y redirigirá automáticamente al Menú de Usuario.

Al elegir la opcion de **Ver el estado** de los pedidos personales, se deben mostrar todos los atributos que poseen las encomiendas (del archivo `encomiendas.csv`) donde el receptor coincide con el usuario de la sesión.

```
** Menú de usuario **

[1] Hacer encomienda
[2] Revisar estado de encomiendas realizadas
[3] Realizar reclamos
[4] Ver el estado de los pedidos personales
[5] Cerrar sesión

Indique la opción elegida: (input)
```

(a) Ejemplo de Menú de usuario.

```
* Ingresa los datos de tu encomienda a continuación *

- Nombre del artículo: Computadora gamer
- Nombre del destinatario: Alex
- Peso (kg): 100000000000000000

¡Dato ingresado en Peso incumple las políticas! Desea:

[1] Continuar
[2] Cancelar encomienda

Ingrese su opción: (input)
```

(b) Ejemplo de opción Hacer encomienda.

4.3. Menú de Administrador

Este menú maneja el funcionamiento de las encomiendas y visualiza las críticas de sus usuarios, por lo que el programa debe mostrar las opciones de: **Actualizar encomiendas**, **Revisar reclamos** y **Cerrar sesión**.

Al seleccionar **Actualizar encomiendas**, se mostrará una tabla con todas las encomiendas registradas en el archivo `encomiendas.csv`, con todos los atributos que posee. Cada fila tendrá un número asociado que permitirá elegir alguna de ellas. Además debe contener la opción de **volver** al menú anterior. Si se decide cambiar el estado, se debe ingresar el número de la encomienda elegida. Al ingresarlo, el programa se encarga de actualizar de forma automática el estado del producto, el cual también se actualizará en el archivo `encomiendas.csv` y, por lo tanto, los usuarios que tengan una relación con la encomienda podrán ver el cambio desde sus menús. Una vez terminado el proceso, el programa redirige al Menú de Administrador y al volver a revisar las encomiendas, la tabla impresa debe mostrar los cambios efectuados.

Al ingresar a la opción **Revisar reclamos** se debe ver una lista de todos los títulos de los reclamos disponibles en el archivo `reclamos.csv`. Esta lista estará indexada, de tal manera que se pueda ingresar un índice y ver la descripción de un reclamo en específico. Al ver un reclamo, se podrá escoger visualizar otro de la lista o simplemente volver al menú anterior.

```
** Menú de administrador **

[1] Actualizar encomiendas
[2] Revisar reclamos
[3] Cerrar sesión

Indique la opción elegida: (input)
```

Figura 5: Ejemplo de Menú de administrador.

```
* Encomiendas registradas *

   |      Nombre artículo      |     Receptor     | Peso |   Destino   |        Estado        |

[1]  Receta para la paz mundial       Maxy15        1.5       Santiago          Emitida
[2]    Pergamino de Python          Gatochico       0.3        Arica       Revisada por agencia
[3]      DCConstitucion           diegocostares     2.3        Tongoy           En camino
[4]     Manual de Excel           magdalenanario    2.5       Rancagua          Emitida

[5] Volver

Ingrese la opción elegida: (input)
```

Figura 6: Ejemplo de opción Actualizar encomienda.

5. Archivos

Los siguientes archivos contienen la base de datos de usuarios, encomiendas y reclamos que usarás en tu programa. Cada vez que edites uno de estos archivos, deberás mantener el formato especificado para dicho archivo. Además, podrás notar que cada uno de los archivos viene con un **encabezado** o *header* en la primera fila. Esto indica a cual columna corresponde cada uno de los elementos de las siguientes filas separados por comas (`","`). Por último, debes asegurarte de que los archivos se actualicen constantemente debido a cualquier interacción con el programa.

Importante: Al momento de leer y escribir en los archivos existe la posibilidad de que ciertos caracteres especiales, como la `ñ` o las tildes no se escriban correctamente. Para solucionar esto, existe un argumento en la función `open()` llamado *encoding* donde pueden especificar el formato de codificación de los caracteres. Para esta tarea, será **obligatorio** el uso de `encoding='utf-8'`[4] ya que permite la correcta interpretación de estos caracteres.

5.1. usuarios.csv

Este archivo contiene todos los usuarios registrados en *DCCorreos de Chile*. Cada fila contiene el nombre de usuario y su respectiva contraseña. Al iniciar sesión debes verificar que el nombre y la contraseña ingresados se encuentren en este archivo. Cuando un usuario es registrado correctamente, debe quedar guardado en este archivo.

Un ejemplo del archivo **usuarios.csv** es el siguiente:

```
usuario,contrasena
matiasmasjuan,qC81jFiTbB
Emiliax16,idcSzaqMbP
catalina-arcila,Y8yuD1dRo6
Gatochico,cuGALudSqA
```

5.2. encomiendas.csv

Este archivo contiene la información de cada encomienda realizada en el programa. Cada fila contiene seis datos importantes: el **nombre del artículo**, el **nombre del usuario** que recibe la encomienda, el **peso**, el **destino**, la **fecha de creación** (en formato `yyyy/mm/dd hh:mm:ss`) y el **estado** que puede ser `'Emitida'`, `'Revisada por agencia'`, `'En camino'` y `'Llegada al destino'`. Es necesario destacar que para obtener acceso a la fecha, será necesario hacer uso de la librería datetime.

[4] Para ver ejemplos de uso, pueden visitar la documentación.

Un ejemplo del archivo **encomiendas.csv** es el siguiente:

```
nombre_articulo,receptor,peso,destino,fecha,estado
III Guerra Mundial ilustrada,Maxy15,1.5,Santiago,2022/03/10 18:04:26,Emitida
Carta de renuncia,Gatochico,0.3,Rancagua,2022/03/01 20:35:55,Revisada por agencia
Receta para la paz mundial,drcid98,0.2,Puerto Montt,2022/02/25 10:00:32,En camino
Pergamino de Python,Emiliax16,0.2,Punta Arenas,2022/02/25 10:00:32,Llegada al destino
```

5.3. reclamos.csv

Este archivo contiene todos los reclamos del sistema. Cada línea posee tres datos relevantes para un reclamo: el **usuario** que lo escribe, el **título** y la **descripción**. Cabe destacar que **la descripción de un reclamo puede poseer comas**, por lo que cualquier coma encontrada después de la que separa el título de la descripción, es parte de esta última.[5].

Un ejemplo del archivo **reclamos.csv** es el siguiente:

```
usuario,titulo,descripcion
Gatochico,Odio a DCCorreos,Hace más de un año que espero mi paquete y Nada
Maxy15,Aprendan a usar GPS,Sus carteros no saben como buscar una dirección?!
DCCollao,Aprovechadores,Me querían cobrar \$800 por cada paquete, los muy #@&%?!
matiasmasjuan,Ladrones,Se robaron mi paquete mientras caminaba al trabajo
```

5.4. `parametros.py`

Al momento de escribir programas de mayor complejidad, como lo son las tareas del curso, una buena práctica es **parametrizar** ciertos datos o variables que podemos querer modificar antes de cada ejecución, sin tener que modificar el código. El tener todos los parámetros almacenados en un único archivo permite identificarlos y modificarlos de forma simple y rápida.

Para esta tarea te entregamos un archivo `parametros.py` ya rellenado. En este archivo se encontrarán los parámetros mencionados anteriormente en el enunciado (en `ESTE_FORMATO`), en donde cada línea almacena una constante con su respectivo valor. En tu tarea deberás **importar**[6] correctamente este archivo y utilizar los parámetros almacenados.

Particularmente, este archivo contiene el mínimo de caracteres para el nombre de usuario al momento de registrarse, el largo mínimo de la contraseña, el peso máximo permitido para una encomienda, y la contraseña del administrador. Por lo tanto, deberás usar las variables de este archivo para verificar si las operaciones de registro de usuario, inicio de sesión y el peso de la encomienda son permitidas. El contenido del archivo `parametros.py` es el siguiente:

```python
# Mínimo de caracteres que debe tener el nombre de un nuevo usuario
MIN_CARACTERES = 5
# Contraseña del administrador
CONTRASENA_ADMIN = "DCCorreos"
# Peso máximo de una encomienda
PESO = 50
# Largo mínimo de la contraseña
LARGO_CONTRASENA = 6
```

[5] Podría ser útil el parámetro **maxsplit** del método split de strings.

[6] Para mas información revisar el material de modularización de la semana 0.

6. *Buenas prácticas*

Se espera que durante todas las tareas del curso, se empleen buenas prácticas de programación. Esta sección detalla dos aspectos que deben considerar a la hora de escribir sus programas, que buscan mejorar la forma en que lo hacen.

- **PEP8**:

  PEP8 es una guía de estilo que se utiliza para programar en Python. Es una serie de reglas de redacción al momento de escribir código en el lenguaje y su utilidad es que permite estandarizar la forma en que se escribe el programa para sea más legible[7]. En este curso te pediremos seguir un pequeño apartado de estas reglas, el cual puede ser encontrado en la **guía de estilo**.

- **Modularización**:

  Al escribir un programa complejo y largo, se recomienda organizar en múltiples módulos de poca extensión. Se obtendrá puntaje si ningún archivo de tu proyecto contiene más de 400 líneas de código.

Normalmente, estos dos aspectos son considerados como descuentos. A manera de excepción, para esta tarea serán parte del puntaje de la tarea, buscando que los apliques y premiando su correcto uso. Ten en cuenta que en las siguientes tareas, funcionarán como cualquier otro descuento de la sección Descuentos.

7. README

Para todas las tareas de este semestre deberás redactar un archivo `README.md`, un archivo de texto escrito en Markdown, que tiene por objetivo explicar su tarea y facilitar su corrección para el ayudante. Markdown es un lenguaje de marcado (como LaTeX o HTML) que permite crear un texto organizado y simple de leer. Pueden encontrar un pequeño tutorial del lenguaje en este ***link***.

Un buen `README.md` debe **facilitar la corrección de la tarea.** Una forma de lograr esto es explicar de forma breve y directa el **idioma** en qué programaste (puedes usar inglés o español) y **qué cosas fueron implementadas, y qué cosas no**, usualmente **siguiendo la pauta de evaluación**. Esto permite que el ayudante dedique más tiempo a revisar las partes de tu tarea que efectivamente lograste implementar, lo cual permite entregar un *feedback* más certero. Para facilitar la escritura del `README`, se entregará una **plantilla** (*template*) a rellenar con la información adecuada.

Finalmente, como forma de motivarte a redactar buenos `README`s, todas las tareas tendrán **décimas de des-descuento** si el ayudante considera que tu `README` fue especialmente útil para la corrección. Estás décimas anulan décimas de descuento que les hayan sido asignadas hasta un máximo de cinco.

8. Descuentos

En todas las tareas de este ramo habrá una serie de descuentos que se realizarán para tareas que no cumplan ciertas restricciones. Estas restricciones están relacionadas con malas prácticas en programación, es decir, formas de programar que hacen que tu código sea poco legible y poco escalable (difícil de extender con más funcionalidades). Los descuentos tienen por objetivo que te vuelvas consciente de estas prácticas e intentes activamente evitarlas, lo cual a la larga te facilitará la realización de las tareas en éste y próximos ramos donde tengas que programar. Los descuentos que se aplicarán en esta tarea serán los siguientes:

- `README`: (1 décima)

[7] Símil a como la ortografía nos ayuda a estandarizar la forma es que las palabras se escriben.

Se descontará una décima si no se indica(n) los archivos principales que son necesarios para ejecutar la tarea o su ubicación dentro de su carpeta. También se descontará una décima si es que no se hace entrega de un `README` o si se entrega incompleto en el mismo estado inicial. Esto se debe a que este archivo facilita considerablemente la recorrección de las tareas.

- **Formato de entrega**: (hasta 5 décimas)

  Se descontarán hasta cinco décimas si es que no se siguen reglas básicas de la entrega de una tarea, como son el uso de groserías en su redacción, archivos sin nombres aclarativos, no seguir restricciones del enunciado, entre otros[8]. Esto se debe a que en próximos ramos (o en un futuro trabajo) no se tiene tolerancia respecto a este tipo de errores.

- **Cambio de líneas**: (hasta 5 décimas)

  Se permite cambiar **hasta 20 líneas de código** por tarea, ya sea para corregir un error o mejorar una funcionalidad. Este descuento puede aplicarse si se requieren cambios en el código después de la entrega (incluyendo las entregas atrasadas). Dependiendo de la cantidad de líneas cambiadas se descontará entre una y cinco décimas.

- **Adicionales**: (hasta 5 décimas)

  Se descontarán hasta cinco décimas a criterio del ayudante corrector en caso de que la tarea resulte especialmente difícil de corregir, ya sea por una multitud de errores o porque el programa sea especialmente ilegible. Este descuento estará correctamente justificado.

- ***Built-in* prohibidos**: (entre 1 a 5 décimas)

  En cada tarea se prohibirán algunas funcionalidades que Python ofrece y se descontarán entre una y cinco décimas si se utilizan, dependiendo del caso. Para cada tarea se creará una *issue* donde se especificará qué funcionalidades estarán prohibidas. Es tu responsabilidad leerla.

- **Malas prácticas**: (hasta 5 décimas)

  Al igual que los *built-ins* prohibidos, también se prohibirán ciertas malas practicas y se descontarán entre una y cinco décimas si se realizan. Recuerda que las malas prácticas por lo general inducen a errores en tu código, por lo que recomendamos fuertemente evitar estas acciones. Para cada tarea se creará una *issue* donde se especificará qué malas prácticas estarán prohibidas. Es tu responsabilidad leerla y preguntar en caso de tener dudas sobre las malas prácticas establecidas.

- **Entrega atrasada**: (entre 5 a 20 décimas)

  Las tareas serán recolectadas automáticamente y no se considerará ningún avance realizado después de la hora de entrega. Sin embargo, se puede optar por entregar la tarea de forma atrasada y se descontarán 5 a 20 décimas dependiendo de cuánto tiempo de diferencia haya entre la hora de entrega y la entrega atrasada.

- **Des-descuento**: (entre 1 a 5 décimas)

  Finalmente, se des-descontarán hasta 5 décimas por un `README` especialmente útil para la corrección de la tarea.

En la **guía de descuentos** se puede encontrar un desglose más específico y detallado de los descuentos.

[8] Uno de los puntos a revisar es el uso de *paths* relativos, para más información revisar el siguiente **material**.

9. `.gitignore`

Cuando estés trabajando con repositorios, muchas veces habrán archivos y/o carpetas que no querrás subir a la nube. Por ejemplo, puedes estar trabajando con planillas de Excel muy pesadas, o tal vez estás utilizando un Mac y no quieres subir la carpeta `__MACOSX`, o el archivo `.DS_Store`, entre otros.

Una posible solución es simplemente tener cuidado con lo que subes a tu repositorio. Sin embargo esta "solución" es extremadamente vulnerable al error humano y podría terminar causando que subas muchos *gigabytes* de archivos y carpetas no deseados a tu repositorio.[9]

Para solucionar esto, `git` nos da la opción de crear un archivo `.gitignore`. Éste es un archivo **sin nombre, y con extensión `.gitignore`**, en el cual puedes detallar **archivos y carpetas a ser ignoradas por `git`**. Esto quiere decir que todo lo especificado en este archivo **no será subido a tu repositorio accidentalmente**, evitando los problemas anteriores.

En esta ocasión el uso de este archivo **no será evaluado**, pero se recomienda su realización para que aprendan a crearlo y utilizarlo ya que será evaluado en las siguientes tareas.

Se recomienda utilizar el archivo `.gitignore` para ignorar archivos en tu entrega, específicamente el enunciado, los archivos indicados en Archivos y todos los que no sean pertinentes para el funcionamiento de tu tarea. El archivo `.gitignore` debe encontrarse dentro de tu carpeta `T0`. Puedes encontrar un ejemplo de `.gitignore` en el siguiente ***link***.

10. Importante: Corrección de la tarea

Para esta tarea, el carácter funcional del programa será el pilar de la corrección, es decir, **sólo se corrigen tareas que se puedan ejecutar**. Por lo tanto, se recomienda hacer periódicamente pruebas de ejecución de su tarea y *push* en sus repositorios.

Cuando se publique la distribución de puntajes, se señalará con color **amarillo** cada ítem que será evaluado a nivel de código, todo aquel que no esté pintado de amarillo significa que será evaluado si y sólo si se puede probar con la ejecución de su tarea.

En tu archivo `README.md` deberás señalar el archivo y la línea donde se encuentran definidas las funciones o clases relacionados a esos ítems.

Finalmente, si durante la realización de tu tarea se te presenta algún problema o situación que pueda afectar tu rendimiento, no dudes en contactar al ayudante jefe de Bienestar al siguiente correo: **bienestar.iic2233@ing.puc.cl**.

11. Restricciones y alcances

- Esta tarea es **estrictamente individual**, y está regida por el **Código de honor de Ingeniería**.
- Tu programa debe ser desarrollado en Python 3.8.
- Tu programa debe estar compuesto por uno o más archivos de extensión `.py`.
- Si no se encuentra especificado en el enunciado, supón que el uso de cualquier librería Python está prohibido. Pregunta en la *issue* especial del **foro** si es que es posible utilizar alguna librería en particular.

[9] Lamentablemente basado en una historia real.

- Debes adjuntar un archivo `README.md` **conciso y claro**, donde describas los alcances de tu programa, cómo correrlo, las librerías usadas, los supuestos hechos, y las referencias a código externo. **Tendrás hasta 2 horas después del plazo de entrega** de la tarea para subir el `README` a tu repositorio.
- Tu tarea podría sufrir los descuentos descritos en la **guía de descuentos**.
- Entregas con atraso de más de 24 horas tendrán calificación mínima (1,0).
- Cualquier aspecto no especificado queda a tu criterio, siempre que no pase por sobre otro.

Las tareas que no cumplan con las restricciones del enunciado obtendrán la calificación mínima (1,0).