

# ***Programación Avanzada***

## **II C2233 2023-2**

Hernán Valdivieso - Daniela Concha - Francisca Ibarra - Joaquín Tagle - Francisca Cattán



# Experiencia 1

- Aplicar contenidos semana 1: OOP.
- Aplicar contenidos de Git.
- Entender errores de consola y cómo enfrentarlos sin que cunda el pánico.

---

# Experiencia 1: ¿Qué vamos a hacer?

1. Vamos a explorar el **repositorio** personal del curso.
2. Aprenderemos a manejar más de un archivo en paralelo.
3. Crearemos un archivo de Python y programaremos desde 0 en éste.
4. Aprenderemos a crear y ejecutar un archivo **main.py**
5. Programaremos principalmente con funciones y objetos.
6. Iremos respaldando nuestros avances en el repositorio mediante **checkpoints**, poniendo en práctica los comandos **add**, **commit** y **push**.

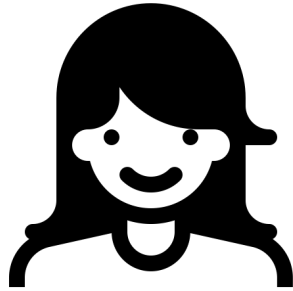
# DCCompras

Todo el mundo está usando apps de compras y tu no quieres quedarte atrás con la tecnología. No sólo te gustaría entender cómo funcionan, si no que hacer la tuya propia.

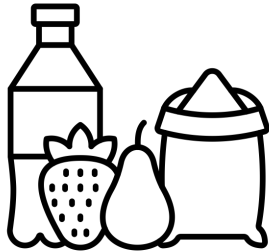
¡Sabes que puedes con este desafío!

---

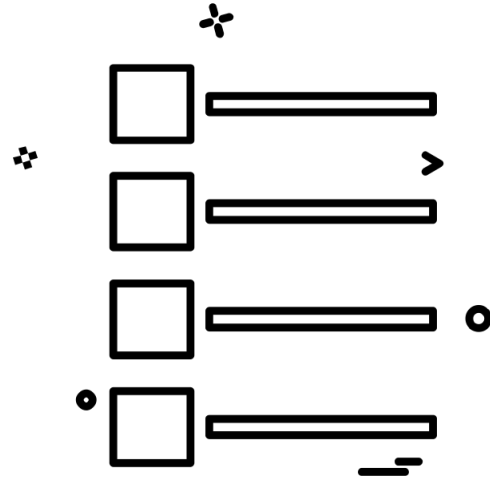
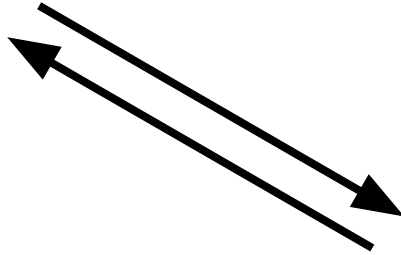
# ¿En qué consiste DCCompras?



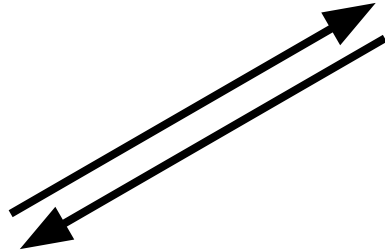
Usuario



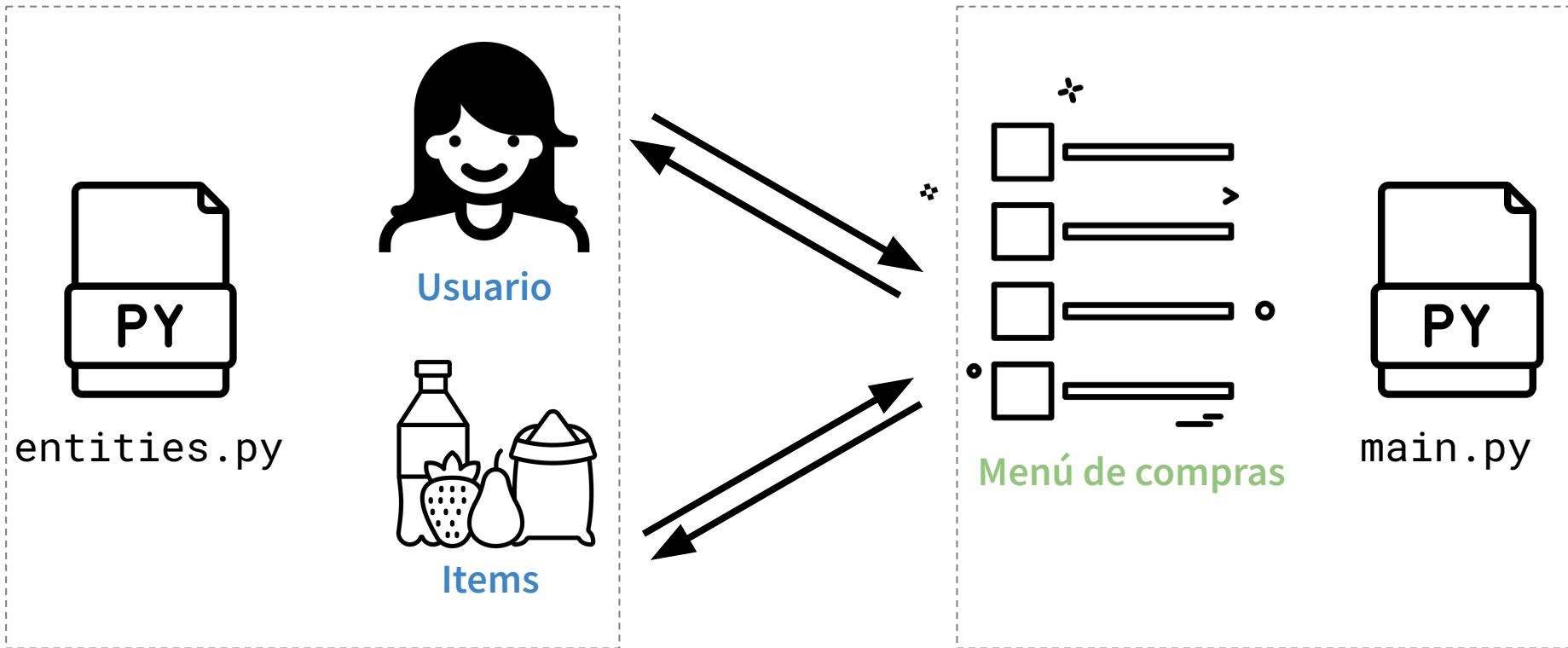
Items



Menú de compras



# ¿En qué consiste DCCompras?

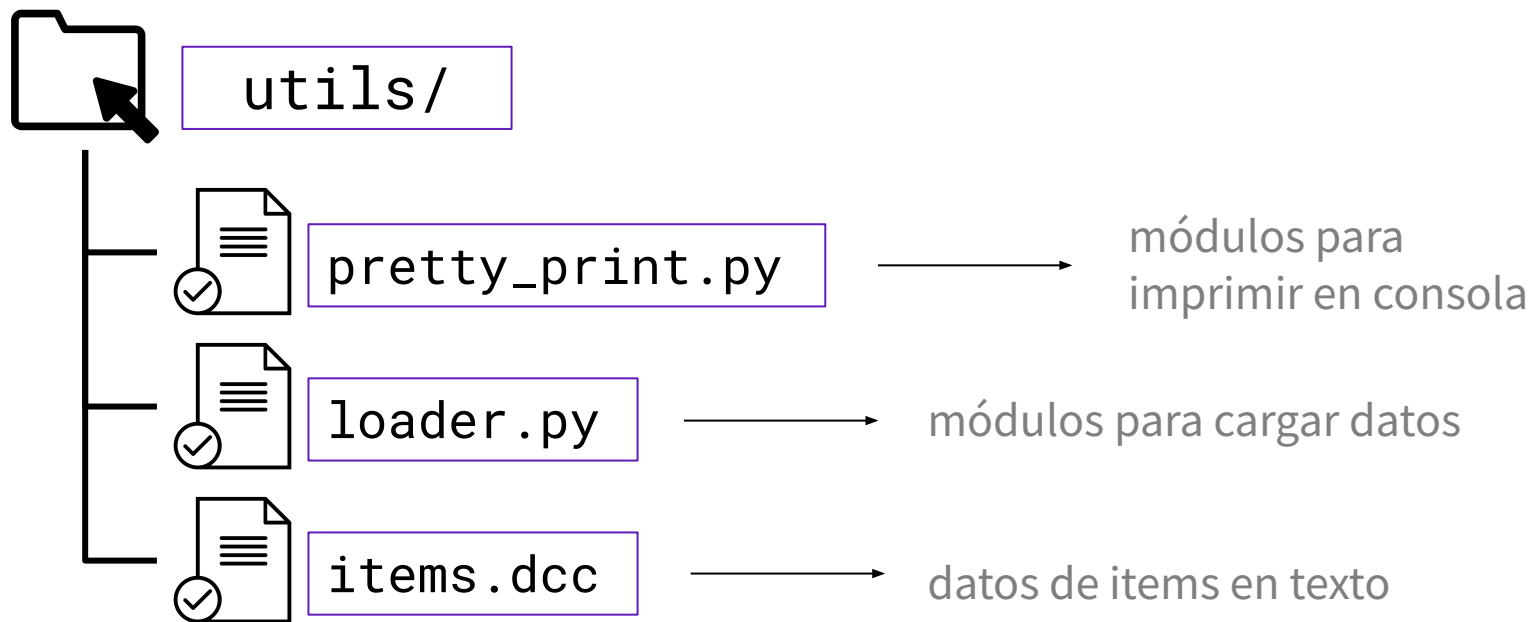


# Paso 1: Repositorio del curso

<https://github.com/IIC2233/syllabus>

1. En Syllabus, vayan a la carpeta “Experiencias”, luego “E1”.
2. Luego descarguen los archivos de la experiencia (E1). Hay 2 funciones y 1 archivo ya creados.
  - Pueden descargarlos desde la web.
  - Si tienen el repo clonado, pueden hacer `git pull`.
3. **No se pongan a programar**, vamos a ir paso a paso entre todos.

# Paso 1: ¿Que tenemos ya descargado?





# Paso 1: ¿Que tenemos ya descargado?



pretty\_print.py

```
def print_opciones_menu() -> None:
```

- Imprime las opciones línea por línea.

```
def print_usuario() -> None:
```

- Imprime si el objeto usuario tiene o no suscripción, y sus puntos.

```
def print_canasta() -> None:
```

- Imprime todos los objetos ítems presentes en la canasta del usuario.

```
def print_items() -> None:
```

- Imprime la info de los objetos ítems disponibles que no se han agregado a la canasta.

```
def print_opcion_invalida() -> None:
```

- Imprime todos los objetos ítems presentes en la canasta del usuario.

```
def print_salida() -> None:
```

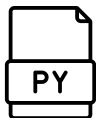
- Mensaje para terminar la sesión.

# Paso 1: ¿Que tenemos ya descargado?



`items.dcc`

Es un archivo donde cada línea tiene el nombre de un ítem, su precio y sus puntos. Esto está en el formato “nombre, precio, puntos”.



`loader.py`

```
def cargar_items() -> list:
```

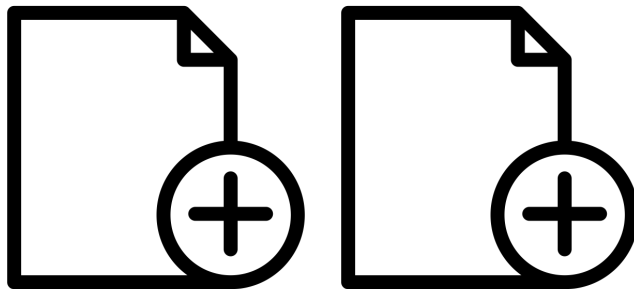
- Esta función abre el archivo `items.dcc` que está en la carpeta `utils`. Luego, por cada línea, separa los tres elementos, los entrega como parámetros al constructor de la clase `Item` y los guarda en la lista de objetos “items”. Finalmente, retorna dicha lista de objetos.

## Paso 2: Repositorio personal

Vamos a nuestro repositorio



## Paso 3: Crear nuevos archivos de python



`main.py`

`entities.py`



# Paso 4: Crear el menú y probar



```
def menu() :
```

- Importa los módulos necesarios.
- Carga la lista de objetos del tipo `Item`.
- Muestra un menú de opciones para que el usuario ingrese una opción (“a”, “b”, “c” o “d”). Este menú debe correr hasta que el usuario quiera salir.
- ¿Qué debería ocurrir en cada opción?
- ¿Cómo ejecuto la función de forma automática?

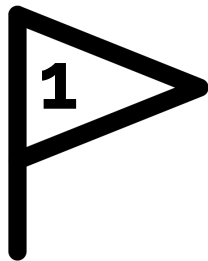
```
if __name__ == "__main__":  
    menu()
```

## Paso 4: Crear el menú y probar

Creemos la base del menú  
Vamos al código



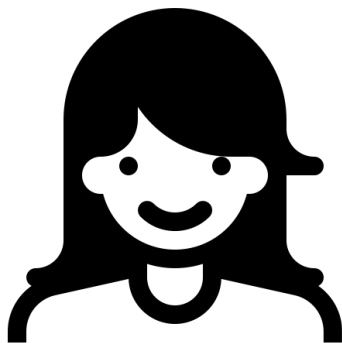
# Checkpoint 1



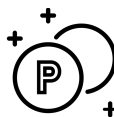
**Guardemos todo en  
nuestro repo privado**



# Paso 5: Crear los objetos



Usuario



## Atributos

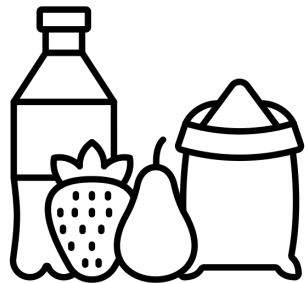
- suscripcion: bool
- canasta: list
- **\_puntos**: int

## Métodos

- agregar\_item(item)
- comprar()
- **puntos**



# Paso 5: Crear los objetos



Item



## Atributos

- nombre: str
- puntos: int
- **\_precio**: int

## Métodos

- **precio**

# Paso 5: Crear los objetos



entities.py

`class` Usuario:

- Inicializa los atributos `suscripcion` (`bool`), `canasta` (`list`) y `_puntos` (`int`).
- La función `agregar_item` recibe un objeto del tipo `Item` y lo agrega a la canasta. Si el usuario tiene suscripción, le aplica al item un 10% de descuento.
- La función `comprar` aplica al usuario los puntos de todos los ítems en la canasta, y la vacía.
- ¿Cómo aplico `properties` al atributo `_puntos`?

`class` Item:

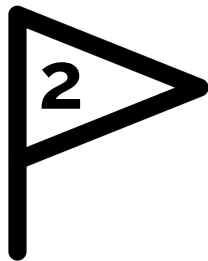
- Inicializa los atributos `nombre` (`str`), `puntos` (`int`) y `_precio` (`int`).
- ¿Cómo aplico `properties` al atributo `_precio`?

## Paso 5: Crear los objetos

Creemos los dos objetos  
Vamos al código



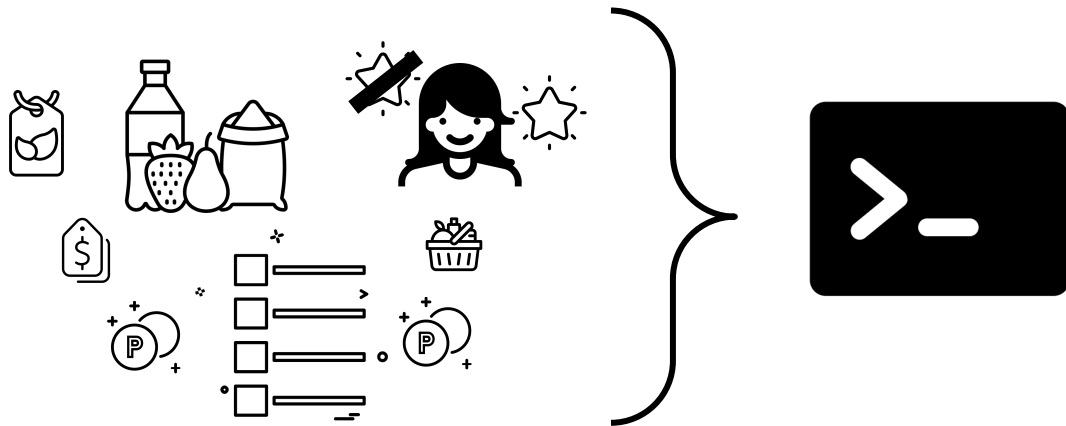
# Checkpoint 2



**Guardemos todo en  
nuestro repo privado**



## Paso 6: Ejecutar, probar, pensar

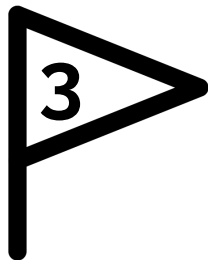


¡Mejoremos el menú para que este sistema funcione super!

**¡Vamos a la consola!**



# Checkpoint final



**Guardemos todo en  
nuestro repo privado**



# ***Programación Avanzada***

## **IIC2233 2023-2**

Hernán Valdivieso - Daniela Concha - Francisca Ibarra - Joaquín Tagle - Francisca Cattán

