



# Clase 11

Trabajos finales y exámenes

# Mecanismos a elección



**Si ustedes quieren ser empleados estas son las opciones que les recomiendo:**

- **Cursar sin final: examen integrador**
- **Rendir el final: examen integrador con días de tiempo para resolver**

**Si quieren trabajar de manera independiente:**

- **Trabajo final integrador realizado con uno o dos compañeras/os**
- **Trabajo final integrador individual**

# Contenidos de la materia que se van a evaluar



- Condicionales
- Ciclos for y/o while
- Listas
- Diccionarios
- Funciones
- Manejo de archivos
- Manejo de errores

Todo esto se va a evaluar tanto para cursar, rendir el final o para el trabajo final. Los exámenes de cursada y final van a tener estas cosas. Los trabajos finales deberán contener al menos 5 de estos 7 puntos, pero bien implementados. Implementar mal alguno de estos puntos o de manera forzada no va a sumar.



## Exámenes

Los exámenes de cursada y de final van a ser extremadamente similares así que mi recomendación va a ser que se tiren a rendir el final.

Les voy a dar los exámenes cuando los termine y les voy a dejar elegir que quieren hacer. Para el final van a tener una semana para resolverlo ( supongo va a ser la última semana de cursada o la que sigue, a confirmar).

Si van a rendir el examen final no hagan el examen para cursar. Si no les alcanza para aprobar el final vamos a hablar para tomarles algo extra para que por lo menos tengan cursada la materia.

*Mi recomendación: Si eligen rendir, elijan el final así se sacan la materia de encima rápido.*

# Trabajos finales



Antes de largarse a hacer el código del trabajo final, hagan primero el algoritmo/pseudocodigo y muestrenmelo. Preparen una especie de presentación de no más de diez minutos donde expliquen lo siguiente:

1. Problema a resolver (no más de un minuto) contando una historia personalizada
2. Cómo sería el uso desde el punto de vista del usuario (no más de dos minutos)
3. Paso a paso como es el algoritmo del proyecto (los 7 minutos restantes o menos)



## Ejemplo



**Problema:** Un kiosco tiene problemas para llevar el control de stock y el kiosquero no sabe cuánta mercadería tiene. Aparte con la inflación que hay remarcar cada alfajor representa un problema. la/el kiosquera/o podría beneficiarse de un sistema integrado de control de stock y de precios.

**Experiencia del usuario:** El sistema que solucionaría el problema de la/del kiosquera/o deberá tener una manera de introducir el nombre de un producto, o su código de barra, y una acción. Esta acción puede ser cobrar, agregar ítem, consultar stock. Si elige cobrar el sistema va a hacer esto, si elige esta otra opción va a pasar esto otro, y etc.

# Algoritmo



Pueden hacer un diagrama de flujo si lo desean. Esto hace que sea más fácil escribir el código luego.

Sino pueden escribirlo en formato más coloquial en forma de pseudocódigo.

tarea = ingrese una acción

si tarea==agregar item:

hacer esto

si tarea==cobrar:

Hacer esto otro

## Realización



Los bloques que dicen “hacer esto” en la slide anterior puede hacerlo un miembro del equipo y otro bloque otro, pero todos van a tener que revisar el código entero y ayudarse. Cada bloque va a tener que ser una función por pedido mio. Si copian y pegan código quiere decir que ahí les conviene usar una función.

Prueben cómo un usuario puede cometer errores y cómo harían para que el código no se rompa. Si yo puedo romper su código haciendo el papel de usuario, el código está mal. Recuerden el try- except.



# Exposición final



Una vez probado el código les voy a pedir una exposición como si tuvieran que venderme el sistema a mí.

Esa exposición tiene que tener lo siguiente:

1. Presentación del equipo ( nombre de los integrantes) ( 30 segs)
2. Explicación del problema a resolver (hasta un minuto)
3. mostrar el funcionamiento del código como si fuera un usuario ( no mas de 5 minutos)
4. Voy a pedirles que introduzcan ciertos valores a las variables de entrada para testear el código.
5. me lo van a mandar al código en un .rar junto con un pdf que tenga la explicación del problema y el pseudocódigo