



PROYECTO – SIMULACIÓN DE FÁBRICA

CS1111 - PROGRAMACIÓN I

LABORATORIO 1.01, 1.02

November 1, 2025

1 ENUNCIADO DEL PROYECTO

En este proyecto los alumnos construirán una simulación de una fábrica vista desde el punto de vista de un Gerente General. La fábrica opera por turnos, es decir, el Gerente General debe tomar decisiones a lo largo de varios turnos. Cada turno representa un mes de operaciones en el que se gestiona la producción, la demanda, los recursos humanos, las compras y las finanzas. La lógica principal se basa en un diccionario de Python llamado `estado`, que almacena todos los indicadores clave (caja, inventario, pedidos, insumos, empleados, máquinas, reputación, etc.). Durante cada mes, el alumno deberá hacer elecciones mediante funciones que permiten:

- Primero, ejecutar una acción de gestión (producción, contratación, marketing, compras, pagos, etc.).
- Segundo, resolver el efecto de una carta del caos: son cartas de eventos aleatorios (desastres, accidentes, cambios de mercado, etc.). En cada turno solo se ejecutará 1 carta del caos. No debe seleccionar una carta al azar, sino que debe ingresar el número de la carta como input a su programa.
- Tercero, recalcular el estado final del mes, aplicando las fórmulas de ingreso, gasto y actualización de contadores. Este estado final debe reflejar las acciones tomadas en el paso 1 y la carta del caos elegida en el paso 2.

Es decir, el código Python ejecuta primero la acción tomada por el Gerente General, después se ejecuta el código de las cartas del caos, y, después de eso, recién se ejecutará el código de la actualización del estado final.

La interfaz gráfica se mostrará con Pygame, pero no se requiere conocimientos de Pygame: el corazón del proyecto es implementar correctamente las funciones en los módulos correspondientes, tal y como se ha estudiado a lo largo del curso.

1.1 REQUERIMIENTOS DEL PROYECTO

Los alumnos deberán:

- Definir y mantener un diccionario `estado` en `estado.py` con todos los indicadores de la empresa.
- Implementar en `acciones.py` todas las funciones contenidas en las cinco áreas funcionales (Producción, Recursos Humanos, Marketing, Compras y Finanzas), siguiendo las especificaciones comentadas en cada función. Cada una de estas funciones debe modificar el diccionario `estado` para que las acciones tomadas impacten en el paso 2 (cartas del caos) y paso 3 (actualización del estado final del mes).
- Implementar en `cartas.py` las 40 cartas de caos, cada una modificando el diccionario `estado` según su descripción en los comentarios.
- En `estado.py`, completar la función `calcular_estado_final()` para aplicar los efectos acumulados de las acciones y las cartas al finalizar cada turno.
- No modificar `main.py` bajo ninguna circunstancia.

- Cada turno es un mes de operación: primero se llama a la acción elegida, luego a la carta del caos, y finalmente a `calcular_estado_final()` en ese orden estricto para actualizar el diccionario.
- Utilizar Pygame únicamente para visualizar el estado actual (tablero, valores numéricos, botones), sin necesidad de agregar más gráficos, animaciones, sonidos, etc.

1.2 DEMOSTRACIÓN

Para la entrega y exposición los alumnos deberán:

- Proyectar la ventana de Pygame en la que se muestre la fábrica y los indicadores actuales.
- Ejecutar al menos cinco turnos completos, seleccionando acciones y mostrando la resolución de cartas.
- Explicar en su exposición cómo cada función de `acciones.py` modifica el diccionario `estado` y cómo luego las cartas de `cartas.py` aplican efectos adicionales.
- Mostrar el código de `estado.py` donde se realiza `calcular_estado_final()`, detallando las fórmulas utilizadas para producir, vender, pagar sueldos y ajustar inventarios.
- Responder preguntas sobre la estructura de módulos, el flujo de datos en el diccionario y la relación entre acciones, cartas y recálculo del estado.

1.3 ENTREGA DEL PROYECTO

- Durante el ciclo, se realizará un solo tema de proyecto, que será evaluado en dos partes.
- La Parte I corresponde al rubro de evaluación P1, y la Parte II al rubro de evaluación P2.
- El proyecto se realiza de manera grupal, grupos de máximo 5 alumnos; su profesor determinará la forma de conformarlos.
- Cada grupo elaborará el proyecto y un informe. La entrega se realiza vía assignment en Gradescope e incluye:
 1. El programa con las funcionalidades implementadas.
 2. Un informe en PDF que explique las principales funciones del programa.
- El informe deberá incluir:
 - **Carátula:** Título del proyecto, curso, universidad, nombres y códigos de los integrantes, nombre del profesor.
 - **Resumen:** Síntesis de todas las secciones elaboradas.
 - **Introducción:** Breve presentación del tema y antecedentes.
 - **Definiciones:** Descripción de los actores principales del programa.

- **Funcionamiento del programa:** Explicación del flujo y fragmentos de código, con capturas si enriquecen.
 - **Ejemplos de Ejecución:** Debe incluir ejemplos de ejecución de su programa y salidas reales.
 - **Conclusiones y recomendaciones:** Deberá presentar como mínimo 4 conclusiones y 4 recomendaciones relacionadas.
- El programa e informe se suben a Gradescope hasta el **lunes 10 de noviembre a las 20:00**. Una entrega por grupo.
 - **La Exposición de la primera parte es grupal** en la sesión 13B y 14A según su horario. Solo deberá mostrar la implementación de las acciones, sin cartas del caos y sin actualizar el estado.
 - **Segunda Entrega:** Hasta el lunes 1 de diciembre a las 20:00; exposición en sesión 16A y 16B, bajo las mismas pautas, pero con las 40 cartas del caos implementadas y el actualizador del estado.

Adicionalmente, tome en cuenta las siguientes consideraciones:

- La asistencia a la presentación es obligatoria; sin justificación válida, quien falte recibe nota cero.
- Si faltan integrantes pero el código está disponible, el resto puede presentar indicando quién trabajó qué.
- Si no pueden presentar porque el código lo tiene un ausente, se considerará no presentado.
- No hay recuperación de esta entrega.
- Si durante las preguntas alguien no comprende o no conoce el código, se penalizará a todo el grupo.
- El profesor puede asignar notas diferenciadas si detecta dominio heterogéneo.
- Si un alumno no responde a las preguntas, se le restará puntaje en programación, funcionalidad, legibilidad/documentación o presentación.

2 CRITERIOS DE EVALUACIÓN

Criterio	Excelente (5 pts)	Adecuado (4 pts)	Mínimo (2 pts)	Insuficiente (0 pts)
Programación	Cumple el 100% de los requerimientos técnicos.	Cumple más del 80% de los requerimientos técnicos.	Cumple al menos el 65%.	Cumple menos del 65%.
Funcionamiento	100% de funcionalidades operativas, sin errores.	Más del 80% operativas, sin errores graves.	Al menos 65%, con errores menores.	Menos del 65%, con errores críticos.
Legibilidad y documentación	Más del 90% del código documentado y modularizado.	Más del 80% documentado.	Más del 65% documentado.	Menos del 50% documentado.
Presentación	Explica claramente, responde 100% preguntas.	Responde 80% preguntas y justifica decisiones.	Responde 65%, con dudas.	No responde o desconoce el código.