



UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS

ALGORITMOS

2025-2

Docente: Henry Antonio Mendoza Puerta

Caso: Préstamo de Tablets en la Universidad

En la universidad se cuenta con un stock de tablets que los estudiantes pueden solicitar para trabajar en proyectos, prácticas o exposiciones. Cada tablet está identificada por un código y un modelo, además de un estado que indica si está disponible o prestada.

Los **estudiantes**, registrados con su código universitario y nombre, pueden pedir prestada una tablet en la oficina correspondiente. Cada vez que se entrega una, se genera un préstamo donde se registra el estudiante que la solicitó, la tablet asignada, la fecha de inicio y la fecha límite de devolución, junto con un estado que refleja si la tablet ya fue devuelta, sigue en uso o está retrasada.

La norma es que una tablet no puede ser prestada a más de un estudiante al mismo tiempo, y que debe poder revisarse qué tablets tiene actualmente un estudiante y cuál ha sido el historial de préstamos de cada tablet.

Lo que debes realizar:

1. Construir el diagrama de clases a partir del caso:
 - Colocar atributos representativos y los métodos en cada clase.
 - Definir asociaciones y las cardinalidades.
2. Implementar un programa en C++ que cumpla:
 - Crear clases con atributos privados, constructores y métodos get/set.
 - Separar la lógica de negocio en una clase de servicio.
 - Usar arreglos dinámicos (`new` y `delete`) para manejar los registros.
 - Incluir un menú con las siguientes opciones:
 1. Registrar estudiantes
 2. Registrar tablets
 3. Registrar préstamos
 4. Registrar devolución de Tablet

Caso: Juego de Personaje con Lanzamiento de Vitaminas

Se desea desarrollar un juego en consola donde un héroe pueda desplazarse y lanzar proyectiles representados como vitaminas, con la posibilidad de destruir obstáculos generados aleatoriamente.

Requerimientos del sistema:

1. Movimiento del héroe
 - El héroe se controla con las teclas W, A, S, D para moverse arriba, izquierda, abajo y derecha.
 - Debe mantenerse dentro de los límites del escenario definido.
2. Lanzamiento de vitaminas
 - Al presionar una tecla especial (por ejemplo, la barra espaciadora), el héroe lanza una vitamina.
 - Cada vitamina se mueve automáticamente en la dirección en la que fue lanzada.
 - Pueden existir N vitaminas simultáneamente activas en pantalla.
3. Estados de las vitaminas
 - Cada vitamina tiene un estado: activa o inactiva.
 - Una vitamina se vuelve inactiva si llega al límite del escenario o colisiona con un obstáculo.
4. Obstáculos
 - Se generan aleatoriamente en el escenario.
 - Pueden colisionar con vitaminas y ser destruidos.
 - Si un obstáculo es destruido, la vitamina también se desactiva.
5. Condición de victoria
 - El juego finaliza cuando el héroe logra eliminar 4 obstáculos mediante el lanzamiento de vitaminas.
 - El usuario también puede terminar la partida con una tecla de salida.
6. Estructura del diseño
 - Clase Héroe: atributos de posición (x, y) y métodos de movimiento.
 - Clase Vitamina: atributos de posición, estado y método para desplazarse.
 - Clase Obstáculo: atributos de posición y estado (activo/inactivo).
 - Clase Juego (controladora/servicio): gestiona al héroe, vitaminas y obstáculos; maneja el bucle principal y actualiza el estado de la partida.
 - Se deben usar arreglos dinámicos (`new` y `delete`) para administrar vitaminas y obstáculos activos.
7. Dinámica de ejecución
 - El juego debe incluir un bucle de ejecución que permita mover al héroe y lanzar vitaminas en tiempo real.
 - Se debe visualizar en la consola la posición del héroe, las vitaminas y los obstáculos activos.
 - El juego termina al eliminar 4 obstáculos o al presionar una tecla de salida.

Lo que debes realizar

- Construir el diagrama de clases a partir del caso
- Implementar un programa en C++ que cumpla