

Facultad de Ingeniería

Departamento de Gestión de Proyectos y Sistemas

Algoritmos y Programación BPTSP05

Profesores: Luis Eduardo Bello, Antonio Guerra y José Quevedo

Trimestre 2122-2

**Proyecto:**

**🏟 saman\_show**

Te han contratado para el desarrollo de un nuevo proyecto, un sistema de venta de Tickets en donde los usuarios (dueños del sistema) podrán administrar su negocio y vender entradas de eventos, recibir pagos, promociones y permitir a sus clientes realizar compras en su feria de comida.

El sistema consta de 4 módulos fundamentales:

1. [Módulo de gestión de eventos](#_5vakqdk99m61)
2. [Módulo de Ventas de tickets](#_rbe4f9c8psr5)
3. [Módulo de Gestión de Feria de comida](#_ebvwi0doc89j)
4. [Modulo de Venta de Feria de comida](#_fasiptyjr820)
5. [Módulo de Estadísticas](#_gsk7eex4ibio)

Información importante en: [observaciones](#_6ao532ltlzo5)

## 

## **Módulo 1: Gestión de Eventos**

Este módulo permitirá al establecimiento administrar sus eventos; para eso tendrás que tener en cuenta la información del “Samán Arena” en donde se realizarán estos eventos, los datos serán dados con una API interna de la empresa, ([ver observaciones](#_v8cft0vz9vi5)). Con esta información deberán desarrollar lo siguiente:

1. Ver los eventos, se proporcionará con la siguiente información:
   1. Nombre del Evento.
   2. Tipo: Musical o Teatro
      1. Si es Musical contará información de cuántas bandas se presentarán
      2. Si es Teatro contará con una breve sinopsis de lo que se trata
   3. Cartel
   4. Información de los asientos en el Saman Arena
   5. Fecha

2. Deberá existir una opción para cerrar la venta de tickets y volverlas abrir

3. Buscar eventos, esto deberá retornar toda la información del evento, empleando uno de los siguientes filtros:

* 1. Tipo.
  2. Fecha
  3. Actor o cantante en el cartel.
  4. Nombre del evento

## **Módulo 2: Venta de tickets**

saman\_show necesitará un sistema para administrar las ventas de los tickets; para esto necesitará solicitar los siguientes datos:

1. Datos del cliente:
   1. Nombre del cliente.
   2. Cedula.
   3. Edad.
   4. Evento que desea comprar ticket (para esto se deberá mostrar toda la información de los eventos.)
   5. Cantidad de tickets que desea comprar
2. Luego el sistema deberá arrojarle un ‘mapa’ del Samán Arena, el cual el cliente podrá seleccionar el o los asientos, si selecciona un asiento ocupado deberá notificarle al cliente que seleccione otro.
3. Por último le aparecerá el costo de la entrada según los siguientes casos:
   1. si su cédula es un [número vampiro](https://es.wikipedia.org/wiki/N%C3%BAmero_vampiro) su entrada tiene un 50% de descuento y se le notificara al cliente
   2. Las entradas hay que sumarle el 16% del impuesto del valor agregado (IVA)
4. Luego deberá mostrar un mensaje indicando al cliente su asiento, costo (con información del subtotal, descuento, IVA y Total) y si desea proceder a pagar la entrada, de ser así, se ocupa el o los asientos y se muestra un mensaje de pago exitoso.

## **Módulo 3: Gestión de Feria de Comida**

saman\_show necesitará un sistema para administrar su feria de comida en eventos con las siguientes funcionalidades:

1. Al tener que guardar el producto en su estructura de datos local, luego de haberla descargado del API, deberá guardar
   1. Nombre del alimento/bebida.
   2. Clasificación (alimento o bebida).
      1. Si es bebida se debe guardar el tamaño: pequeño, mediano, grande; Para esto el inventario se dividirá en 3 partes iguales para que existan las mismas cantidades de pequeño, mediano y grande.
      2. Si es alimento se debe guardar si es de empaque o de preparación.
   3. Precio (se le deberá sumar el 16% del IVA).
2. Eliminar productos del menú.
3. Buscar productos por nombre, tipo, o rango de precio.

## **Módulo 4: Venta de Feria de Comida**

Para la venta en la feria de comida se necesitará que el cliente ya haya comprado una entrada, esto se validará con su cédula, si es así se procederá de la siguiente manera:

1. Se guardan los datos del cliente:
   1. Cedula.
   2. Comida(s) que desee comprar
2. Luego deberá mostrarle los productos que desea comprar con el monto total, siguiendo los siguientes casos:
   1. Si la cédula es un [número narcisista](https://es.wikipedia.org/wiki/N%C3%BAmero_narcisista#:~:text=En%20matem%C3%A1tica%20recreativa%20un%20n%C3%BAmero,%22quererse%20a%20s%C3%AD%20mismos%22.) obtendrá un 15% de descuento.
3. Por último, si el cliente desea proceder con la compra, se le mostrará un mensaje de pago exitoso con un resumen de su compra donde se muestre el monto con su subtotal, descuento y total.
4. Se debe restar del inventario la cantidad de productos que el cliente compro

## **Módulo 5: Estadísticas**

Toda empresa necesita evaluar su gestión y ver que le está funcionando y que no, para eso es importante un módulo de estadísticas que le indique a saman\_show lo siguiente:

1. ¿Cuál es el promedio de gasto de un cliente en un evento (ticket + feria de comida)?
2. ¿Cuál es el porcentaje de clientes que no compran en la feria de comida?
3. ¿Quiénes son los clientes (los 3 primeros) de mayor fidelidad en el saman\_show?
4. Top de 3 Eventos con más ingresos.
5. Top 5 productos más vendidos en la feria de comida.
6. Realizar gráficos con dichas estadísticas con las librerías de mathplolib o Bokeh (**Bono**).

## 

## **Observaciones**

* **saman\_show** posee una Arena donde realiza sus eventos, en cada evento se puede tener un layout distinto, toda la información del evento lo podrán encontrar en la siguiente API:
  + **Documentación:** <https://github.com/Algoritmos-y-Programacion/api_saman_show>
  + **Endpoint:** <https://raw.githubusercontent.com/Algoritmos-y-Programacion/api_saman_show/main/api.json>
* La API tiene que funcionar como una opción de pre-cargado de datos antes de empezar a usar el programa, es decir esta opción crea el estado inicial del programa, posteriormente no se debe usar la API a menos que se quiera borrar los datos y cargar su estado inicial
* Se deben usar los conceptos de programación orientado a objetos
* Antes de realizar el código, es imperante que realicen un **diagrama de clases** y que la implementación de su proyecto sea uno a uno con el diagrama
* Se evaluará que el código este comentado (docstring)
* Se evaluará que el sistema contenga validaciones
* Se deberán guardar datos en un archivo TXT para preservar los datos
* El proyecto deberá ser entregado en Github Classroom a más tardar el Domingo 27 de Marzo del 2022 a las 8:00PM