

**Taller No. 2**

Se necesita dar solución a los siguientes ejercicios teniendo en cuenta las siguientes recomendaciones:

- Realice la lectura del planteamiento del problema tantas veces como sea conveniente, hasta comprender claramente de qué se trata. Si es necesario, realice las consultas adecuadas para recordar conceptos o ideas para entender el planteamiento del problema.
- Resuelva el problema planteando un ejemplo donde realice las operaciones y procesos necesarios para encontrar la solución, si es necesario aplique formulas.
- Teniendo en cuenta lo anterior, piense y organice en los pasos lógicos necesarios para resolver el problema, es decir, elabore el algoritmo. Se deben identificar asignaciones, decisiones o ciclos que intervienen en el proceso para solucionar el problema.
- Identifique las variables en la memoria del computador necesarias para resolver el problema, identifique si es necesario utilizar técnicas para almacenar datos como banderas, contadores o acumuladores.

**ESPECIFICACIONES SOBRE LA ENTREGA**

1. Cada ejercicio debe tener su carpeta aparte, que incluye los archivos:
  - Ejemplo con datos (Microsoft Excel)
  - Análisis orientado a objetos (Microsoft Word)
  - Diseño orientado a objetos (WhiteStarUML)
  - Programación orientada a objetos (Visual Studio Code)
2. Cada algoritmo debe mostrar los derechos de autor, el nombre del estudiante, el semestre, el programa y la institución.
3. Se debe utilizar CamelCase para nombrar los archivos, las carpetas, las variables y los arreglos. En idioma **INGLÉS**
4. La carpeta debe comprimirse en un archivo con WINRAR (.rar). El archivo comprimido debe tener el nombre del estudiante escrito así: Apellido1Nombre1.rar
5. El archivo debe subirse a la plataforma Aulas Virtuales, en el enlace respectivo. en la fecha establecida
6. Cualquier duda por favor se resuelve utilizando Whatsapp o mediante llamada telefónica.

**ESPECIFICACIONES SOBRE LA CODIFICACIÓN**

- Manejar un menú de opciones, por lo menos una opción por cada funcionalidad que se solicita, incluyendo la opción salir del programa
- Manejar diseño gráfico con la librería PyQt5.
- Manejar archivos
- Se debe utilizar estructuras dinámicas listas.

**ASIGNACIÓN DE EJERCICIOS POR GRUPOS**

El taller debe entregarse individualmente. La fecha de entrega es el **domingo, 22 de enero del 2023, hasta las 12pm**, mediante la plataforma Aulas Virtuales de la Universidad de Nariño.

La siguiente tabla muestra la distribución de ejercicios para cada estudiante y la fecha y hora de la asesoría.

Código1	Código2	Ejercicio	Fecha	Franja	
220151035	220151029	ProjectComputerEquipmentMaintenance	Jueves, 9 de febrero	8:00 a. m.	8:30 a. m.
218151006	218151012	ProjectClaimsSuggestionsEPS	Jueves, 9 de febrero	8:30 a. m.	9:00 a. m.
220151021	220151031	ProjectInternetSubscriptions	Jueves, 9 de febrero	9:00 a. m.	9:30 a. m.
219151044	219151098	ProjectParkingLog	Jueves, 9 de febrero	9:30 a. m.	10:00 a. m.
219151065	219151087	ProjectPharmacyDrug	Jueves, 9 de febrero	10:30 a. m.	11:00 a. m.
221151029	221151047	ProjectPharmacyDrug	Jueves, 9 de febrero	11:00 a. m.	11:30 a. m.
221151007	221151024	ProjectClaimsSuggestionsEPS	Jueves, 9 de febrero	11:30 a. m.	12:00 p. m.
221151017	221151048	ProjectInternetSubscriptions	Viernes, 10 de febrero	8:00 a. m.	8:30 a. m.
219151006	219151012	ProjectClaimsSuggestionsEPS	Viernes, 10 de febrero	8:30 a. m.	9:00 a. m.
221151049	221151052	ProjectComputerEquipmentMaintenance	Viernes, 10 de febrero	9:00 a. m.	9:30 a. m.
219151020	219151091	ProjectParkingLog	Viernes, 10 de febrero	9:30 a. m.	10:00 a. m.
218151087	218151097	ProjectPharmacyDrug	Viernes, 10 de febrero	10:30 a. m.	11:00 a. m.
220151051		ProjectParkingLog	Viernes, 10 de febrero	11:00 a. m.	11:30 a. m.

La asesoría se desarrollará en la Universidad de Nariño, en un salón de clases disponible.

**Confirmar la asistencia a la asesoría al WhatsApp Personal. Estar puntual en la hora de la asesoría.**

**EJERCICIOS**

<b>Ejercicio No.:</b>	01
<b>Título:</b>	ProjectEmployeeEntryControl
<b>Planteamiento:</b>	
Se necesita manejar la información del control de ingreso de los empleados de una empresa	
Debe permitir:	
<ul style="list-style-type: none"> <li>- Registrar la información de cada empleado. Cada empleado tiene: código interno, número de identificación, nombre, cargo. Se debe verificar que el código interno no se repita.</li> <li>- Registrar las diferentes entradas y salidas de cada empleado con fecha y hora cada una, calculando el tiempo que estuvo dentro de la empresa.</li> <li>- Mostrar un reporte general con la información de los empleados</li> <li>- Mostrar el historial de entradas y salidas de un empleado usando el código interno.</li> </ul>	

<b>Ejercicio No.:</b>	02
<b>Título:</b>	ProjectInternetSubscriptions
<b>Planteamiento:</b>	
Se necesita manejar la información del control de suscripciones a internet de una empresa que opera ese servicio	
Debe permitir:	
<ul style="list-style-type: none"> <li>- Registrar la información del plan de internet. Cada plan tiene: código interno, número de identificación, nombre del cliente, dirección del cliente y número de teléfono del cliente y ancho de banda del plan. Se debe verificar que el código interno no se repita.</li> <li>- Registrar los cobros mensuales del plan de internet. Cada cobro tiene: fecha de emisión, fecha de pago y valor.</li> <li>- Mostrar el reporte general con la información de los planes de internet</li> <li>- Mostrar el histórico de cobros asociados a un plan de internet usando el código interno.</li> </ul>	

<b>Ejercicio No.:</b>	03
<b>Título:</b>	ProjectApartmentRental
<b>Planteamiento:</b>	
Se necesita manejar la información sobre apartamentos en alquiler de una empresa inmobiliaria	
Debe permitir:	
<ul style="list-style-type: none"> <li>- Registrar la información de cada apartamento para alquiler. Cada apartamento debe tener: código interno, nombre del propietario, dirección, superficie en m2 y precio del alquiler mensual, estado (Ocupado o Disponible). Se debe verificar que el código interno no se repita</li> <li>- Registrar el alquiler de un apartamento disponible. Cada alquiler debe tener: el nombre del arrendatario, la fecha de inicio, la fecha final. El estado del apartamento debe cambiar de Disponible a Ocupado</li> <li>- Mostrar el reporte general con la información de los apartamentos disponibles para alquiler.</li> <li>- Mostrar el historial de alquileres de un apartamento usando el código interno del mismo.</li> </ul>	

<b>Ejercicio No.:</b>	04
<b>Título:</b>	ProjectParkingLog
<b>Planteamiento:</b>	
Se necesita manejar la información de entrada y salida de vehículos en un parqueadero.	
Entre las opciones se tiene:	
<ul style="list-style-type: none"> <li>- Ingresar la información de entrada de un vehículo. Al ingresar un vehículo al parqueadero se registra: la matrícula del vehículo, la fecha y hora de entrada. La matrícula del vehículo no se puede repetir</li> <li>- Ingresar la información de salida del vehículo. Usando la matrícula del vehículo se debe buscar el registro de entrada de este y registrar la fecha y hora de salida, también la cantidad de minutos que estuvo en el parqueadero y el costo del servicio.</li> <li>- Calcular el costo del servicio del parqueadero, teniendo en cuenta que por un minuto de servicio se cobra un valor fijo.</li> <li>- Mostrar un reporte con la información de los servicios prestados en un día y el costo total recaudado.</li> <li>- Mostrar un reporte con las entradas y salidas de un vehículo usando la matrícula.</li> </ul>	

<b>Ejercicio No.:</b>	05
<b>Título:</b>	ProjectClaimsSuggestionsEPS
<b>Planteamiento:</b>	
Se necesita manejar la información de las peticiones, quejas, reclamos y sugerencias de una EPS, realizadas por los usuarios	
Entre las opciones se tiene:	
<ul style="list-style-type: none"> <li>- Ingresar la información de los usuarios de la EPS. Un usuario tiene: identificación, nombre, dirección, teléfono y email. La identificación no se puede repetir</li> <li>- Ingresar los datos de un reclamo o sugerencia realizadas por los usuarios. Una queja o solicitud tiene: código, fecha, hora, información de usuario que la realizó, descripción y estado (PENDIENTE o REVISADA) y un tipo (Petición, Queja, Reclamo o Sugerencia). El código no se puede repetir</li> <li>- Buscar una queja o sugerencia usando un código y registrar la respectiva respuesta por parte de la EPS</li> <li>- Mostrar un reporte con el listado de quejas o sugerencias, filtradas por estado o por tipo</li> <li>- Mostrar el detalle de una queja o sugerencia usando la identificación del usuario o el código de la EPS</li> </ul>	