



PROYECTOS DEL SEMESTRE

Proyecto # 1:

Se desea construir una aplicación que permita simular el uso, por parte de un cliente, de una máquina expendedora de alimentos, la cual contiene 4 productos. La aplicación permite al usuario (o cliente de la máquina) añadir monedas a la máquina para tener un crédito de compra, comprar productos, recibir el cambio cuando haya terminado una compra y conocer la información sobre las compras que ha realizado en la máquina.

Adicionalmente, la máquina permite que se done un porcentaje del valor recaudado, por la compra de algunos de sus productos, al FOPRE (Fondo de Programas Especiales). El aporte al FOPRE corresponde al 6% del valor del producto. No todos los productos de la máquina donan al FOPRE.

El sistema debe permitir:

1. Visualizar la información de los productos.
2. Agregar crédito (monto de dinero) a la máquina.
3. Comprar un producto.
4. Calcular la cantidad total de unidades compradas (de todos los productos).
5. Calcular el valor total por las compras realizadas.
6. Calcular el porcentaje de disponibilidad de la máquina.
7. Consultar el producto más comprado (del cual ha comprado un mayor número de unidades).
8. Calcular el valor de la donación total al FOPRE por las compras realizadas.
9. Conocer el valor de la donación al FOPRE por tipo de producto.
10. Conocer las unidades compradas de productos FOPRE y por tipo de producto.
11. Terminar la compra.





Proyecto # 2:

Se quiere crear un programa que permita administrar una tienda de libros. La tienda tiene un catálogo de libros, que son los libros que desea poner a la venta. La aplicación permite abastecer la tienda con ejemplares de los libros del catálogo y venderlos. Adicionalmente permite saber cuánto dinero se tiene en caja, empezando con una inversión inicial de \$1.000.000.

De cada libro se conoce:

- ISBN. Identificador del libro. No pueden existir dos libros en la tienda con el mismo ISBN.
- Título. El nombre del libro.
- Precio de compra: Valor pagado por la compra de cada ejemplar en la tienda.
- Precio de venta: Valor por el cual se vende cada ejemplar del libro.
- Cantidad actual. Cantidad actual de ejemplares que tiene la tienda. Solo puede ser modificada mediante la venta o abastecimiento.

Adicionalmente, de cada libro se conoce todas las transacciones que se han realizado sobre él. De cada transacción se conoce:

- El tipo de transacción. Puede ser venta o abastecimiento.
- La fecha de realización.
- La cantidad de ejemplares incluidos en la transacción.

El abastecimiento de libros permite aumentar la cantidad actual de ejemplares del libro y registrar una transacción de tipo abastecimiento.

La venta de libros permite disminuir la cantidad actual de ejemplares del libro y registrar una transacción de venta. Esta transacción solo se podrá realizar si la cantidad actual de ejemplares es mayor a la cantidad que se quiere vender.

El programa debe permitir al usuario:

1. Registrar un libro en el catálogo.
2. Eliminar un libro del catálogo.
3. Buscar un libro por título.
4. Buscar un libro por ISBN.
5. Abastecer ejemplares de un libro.
6. Vender ejemplares de un libro.
7. Calcular la cantidad de transacciones de abastecimiento de un libro particular.
8. Buscar el libro más costoso.
9. Buscar el libro menos costoso.
10. Buscar el libro más vendido.





Proyecto # 3:

Se quiere crear un programa que tenga como objetivo el manejo de reservas de un avión. Este avión cuenta con un número fijo de 50 sillars. De ellas, 8 son de clase ejecutiva, mientras que el resto son de clase económica. Cada silla puede ser asignada a un pasajero que cuenta con un nombre y una cédula. Este último dato es la entrada principal para poder consultar una reserva o eliminarla del sistema.

Cuando se asigna una silla es necesario conocer las preferencias del usuario. Este puede elegir la posición de la silla, ventana, pasillo o centro, y la clase, ejecutiva o económica. En el caso especial de las sillars ejecutivas, solo es posible elegir las posiciones: ventana o pasillo. Las sillars son asignadas de forma secuencial según su ubicación y su clase. De igual forma, el programa permite buscar la reserva de un pasajero y visualizar los datos de la reserva.

El programa debe permitir al usuario:

1. Asignar una silla a un pasajero
2. Consultar una reserva
3. Eliminar reserva
4. Buscar pasajero
5. Calcular el porcentaje de ocupación del avión.
6. Consultar el valor total de ventas por concepto de sillars ocupadas en el avión.
7. Consultar el valor promedio de venta por concepto de sillars ocupadas / pasajero en el avión.





Proyecto # 4:

Se quiere construir una aplicación para la Central de Préstamos de la Universidad, la cual se encarga de manejar el préstamo de todos los recursos que la universidad ofrece a sus estudiantes.

Los recursos pueden ser de cualquier naturaleza, se identifican con un código y tienen además un nombre. Los códigos son únicos, pero los nombres pueden repetirse. Cada recurso que se quiera prestar a los estudiantes debe ser registrado en la aplicación. Un recurso se puede prestar sólo si está disponible, es decir que no se ha prestado a otro estudiante.

Un estudiante se identifica por su código, que también es único, y tiene un nombre que eventualmente otro estudiante también podría tener. Para que un estudiante pueda prestar algún recurso debe registrarse. Si el estudiante no está registrado no se le prestará ningún recurso.

La aplicación debe permitir:

1. Agregar un recurso
2. Agregar un estudiante
3. Prestar un recurso disponible
4. Consultar los préstamos de un estudiante
5. Consultar la información de un préstamo
6. Devolver un recurso prestado





Proyecto # 5:

Se quiere construir una aplicación para administrar un parqueadero. Dicho parqueadero tiene 40 puestos numerados. En cada puesto se puede parquear un carro. El parqueadero tiene una tarifa por hora o fracción de hora.

De cada puesto se conoce:

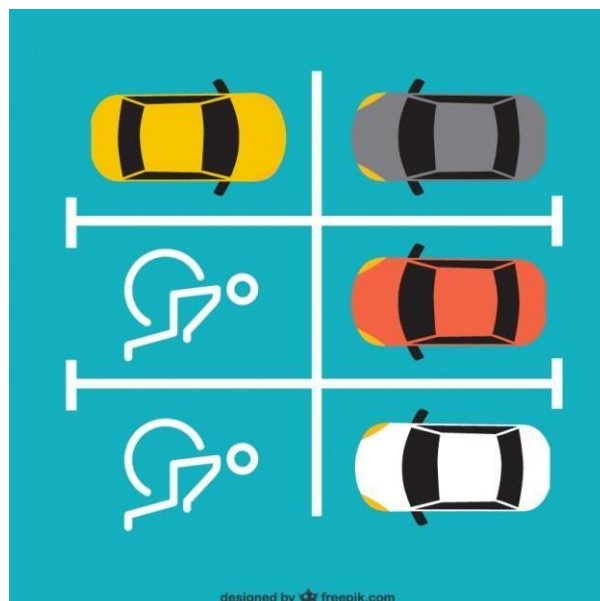
- El número del puesto.
- El carro que se encuentra parqueado en el puesto actualmente. Un puesto puede no tener ningún carro.

De cada carro se conoce:

- Placa: Identificado único del carro. No puede haber dos carros con la misma placa.
- Hora de entrada. Corresponde a un valor entre 6:00 y 21:00, dado que el parqueadero está abierto entre 6 de la mañana y cierra a las 9 de la noche.

La aplicación debe permitir:

1. Ingresar un carro al parqueadero.
2. Dar salida a un carro del parqueadero.
3. Informar los ingresos totales del parqueadero.
4. Consultar la cantidad de puestos disponibles.
4. Consultar el porcentaje de disponibilidad.
5. Avanzar el reloj del parqueadero.
6. Cambiar la tarifa del parqueadero.





Proyecto # 6:

Se quiere crear una aplicación que permita administrar la información de los censos realizados en los municipios de todo el país. Cuando se registra el censo de un municipio en un departamento, se incluye el nombre, la población total, la cantidad de hombres, la edad promedio de la población, el ingreso promedio de la población y la temperatura media del departamento. Cada vez que registra un censo, se recalculan los valores totales del censo del departamento donde se registró el municipio los valores totales del censo a nivel nacional.

La información que se debe visualizar, tanto en los consolidados por departamento, como en los consolidados totales del país es:

- Población.
- Cantidad total de hombres.
- Cantidad total de mujeres.
- Edad promedio de la población.
- Ingreso promedio.
- Temperatura media.

Los 32 departamentos a censar son:

Amazonas, Antioquia, Arauca, Atlántico, Bolívar, Boyacá, Caldas, Caquetá, Casanare, Cauca, Cesar, Chocó, Córdoba, Cundinamarca, Guainía, Guajira, Guaviare, Huila, Magdalena, Meta, Nariño, Norte de Santander, Putumayo, Quindío, Risaralda, San Andrés y Providencia, Santander, Sucre, Tolima, Valle Del Cauca, Vaupés, Vichada.

La aplicación debe permitir:

1. Agregar un municipio.
2. Editar la información de un municipio.
3. Eliminar un municipio.
4. Buscar un departamento de un municipio.
5. Consultar si existen departamentos con un ingreso superior a un valor dado.
6. Consultar si existen departamentos con una población igual a un valor dado.
7. Visualizar el total del censo por departamento.
8. Visualizar el total del censo a nivel nacional.





Proyecto # 7:

Se quiere construir una aplicación que permita estudiar el sistema solar a través de un observatorio de planetas, donde se puedan registrar los satélites naturales que se han descubierto. Cada planeta tiene un nombre, una distancia medial al sol (expresada en UA o unidades astronómicas), un periodo orbital (sinódico), una velocidad orbital media y una inclinación orbital. Asimismo, un satélite cuenta con un nombre, una excentricidad, un periodo orbital y una inclinación orbital.

La excentricidad de un planeta es la medida de la deformación de la elipse que lo conforma. Puesto que un planeta no es completamente redondo, la excentricidad es un cociente entre los focos de la elipse y eje mayor que va de cero a uno. .

El período orbital o sinódico es el tiempo que tarda el planeta en volver a aparecer en el mismo punto del cielo respecto del Sol cuando se observa desde la Tierra. La unidad utilizada para medir este tiempo es días.

La velocidad orbital media es la distancia recorrida durante un determinado periodo de tiempo por un planeta, satélite o similar al orbitar alrededor de una estrella o planeta. La unidad utilizada para medir esta velocidad es km/s.

La inclinación orbital es el ángulo que el plano de la órbita de un astro forma con un plano de referencia (para los planetas, eso es la eclíptica). La unidad utilizada para medir esta inclinación son los grados.

La aplicación debe permitir:

1. Registrar un satélite natural
2. Editar la información de un satélite natural
3. Eliminar un satélite natural
4. Consultar los planetas con una distancia media al sol menor que la distancia dada
5. Consultar los planetas con una inclinación orbital menor a la del planeta seleccionado





Proyecto # 8:

Se quiere crear una aplicación para controlar los gastos telefónicos de una empresa. La empresa cuenta con tres líneas telefónicas a través de las cuales se pueden realizar llamadas locales, de larga distancia y a celulares.

La empresa cuenta con planes telefónicos que establecen las siguientes tarifas:

Minuto Llamada Local: \$35 pesos

Minuto Llamada Larga Distancia: \$380 pesos

Minuto Llamada Celular: \$999 pesos

El programa debe permitir:

1. Agregar una llamada a alguna de las líneas telefónicas de la empresa.
2. Visualizar la información de cada línea. Esta información incluye el número total de llamadas realizadas, la duración total de las llamadas en minutos y el costo total de las llamadas en pesos.
3. Visualizar la información consolidada de todas las líneas telefónicas de la empresa. Esta información incluye el costo total en pesos de las tres líneas, el número total de llamadas realizadas, la duración total de llamadas en minutos y el cálculo del costo promedio por minuto según el costo total y el total de minutos.
4. Reiniciar la información de las líneas telefónicas de la empresa.





Proyecto # 9:

Se desea construir una aplicación que permita simular un álbum con equipos históricos de mundiales de fútbol. La aplicación permite agregar equipos, modificar la información de los jugadores, pegar láminas de distintos tipos, ver la ficha técnica de los jugadores y conocer algunas estadísticas.

El sistema debe permitir:

1. Agregar equipo.
2. Modificar la información de un jugador.
3. Visualizar la ficha técnica de un jugador.
4. Buscar un jugador por su número de camiseta, país y año en el que participó en el mundial.
5. Conocer los equipos del álbum que participaron en un año dado.
6. Conocer la edad más común de un equipo.
7. Conocer el porcentaje de completitud del álbum, discriminado por tipo de lámina.





Proyecto # 10:

Se desea construir una aplicación para el manejo de información de los cursos que está tomando un estudiante. El estudiante toma solo 4 cursos en el semestre.

De cada estudiante se conoce:

- Código
- Nombre
- Apellido
- Promedio

De cada curso se conoce:

- Código. Es el identificador del curso y no puede haber dos cursos con el mismo código.
- Nombre.
- Departamento. Puede ser Matemáticas, Física, Sistemas o Biología.
- Cantidad de créditos.
- Nota obtenida en el curso. Este valor debe estar entre 1.5 y 5.

Para poder calcular el promedio del estudiante, se deben ponderar las notas, teniendo en cuenta la cantidad de créditos de las materias. Para esto, para cada curso se debe multiplicar la nota del curso con su cantidad de créditos, sumar estos valores y dividir esta suma por la cantidad total de créditos vistos por el estudiante.

Adicionalmente, se quiere poder saber si un estudiante está en prueba académica o si es candidato para beca. Para esto se debe tener en cuenta las siguientes reglas.

- Se considera que un estudiante está en prueba académica si su promedio es inferior a 3.25.
- Se considera que un estudiante es candidato a beca si su promedio es igual o superior a 4.75.

La aplicación debe permitir:

1. Visualizar la información del estudiante.
2. Visualizar la información de los cursos.
3. Modificar la información de un curso.
4. Asignar una nota a un curso.
5. Calcular el promedio del estudiante.
6. Indicar si el estudiante está en prueba académica.
7. Indicar si el estudiante es candidato a beca.





Proyecto # 11:

Se quiere construir una aplicación que permita administrar una sala de cine. Esta aplicación permite hacer reservas y registrar sus pagos.

La sala de cine tiene 220 sillas. De cada silla se conoce:

- Fila a la que pertenece, representada por un valor entre A y K.
- Número de la silla, valor entre 1 y 20.
- Tipo. Puede ser general o preferencial.
- Estado de la silla. Puede ser disponible, reservada o vendida.

El costo de boleta se determina según el tipo de la silla, y esta a su vez se determina según su número, de la siguiente manera:

- General: sillas en las filas A – H. Costo por boleta de \$8,000.
- Preferencial: sillas en las filas I – K. Costo por boleta de \$11,000.

Para poder adquirir una boleta, el cliente debe primero hacer una reserva. Cada cliente puede reservar hasta 8 sillas. De cada reserva se conoce:

- Cédula de la persona que hizo la reserva.
- Sillas que hacen parte de la reserva.
- Estado de pago de la reserva.

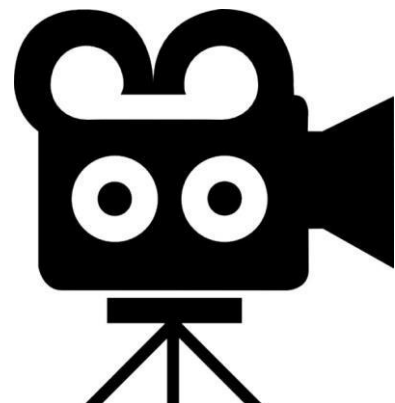
El cliente puede pagar sus reservas en efectivo o utilizando la tarjeta CINEMAS. Esta tarjeta le otorga a su dueño un descuento del 10% en sus boletas. De cada tarjeta se conoce:

- Cédula del dueño de la tarjeta. No pueden existir dos tarjetas con la misma cédula.
- Saldo de la tarjeta: Cantidad de dinero disponible para pagar reservas.

Cuando se adquiere una tarjeta, el cliente debe cargar la tarjeta con un valor inicial de \$70,000. Cada tarjeta puede ser recargada una cantidad ilimitada de veces, sin embargo, cada recarga se debe hacer por un monto de \$50,000.

La aplicación debe permitir:

1. Crear una nueva tarjeta.
2. Recargar una tarjeta.
3. Crear una reserva.
4. Eliminar la reserva actual.
5. Pagar una reserva en efectivo.
6. Pagar la reserva con tarjeta CINEMAS.
7. Visualizar las sillas del cine.
8. Visualizar el dinero en caja.





Proyecto # 12:

Se desea construir una aplicación que permita administrar la información de los integrantes de una banda musical y las ganancias obtenidas en los conciertos que ofrecen. La banda está conformada por 4 integrantes.

De cada integrante se conoce:

1. Nombre del integrante
2. Instrumento que toca el integrante
3. Ruta de la imagen de la foto del integrante.
4. Cantidad de seguidores del integrante.
5. Cantidad de conciertos en los que se ha presentado.
6. Total de ganancias adquiridas en los conciertos que se ha presentado.

El sistema debe permitir:

1. Visualizar la información de los integrantes.
2. Agregar seguidores a un integrante.
3. Quitar seguidores a un integrante.
4. Reiniciar la cantidad de conciertos y ganancias de un integrante.
5. Registrar un concierto repartiendo en partes iguales las ganancias para los integrantes.
6. Consultar el total de seguidores de la banda.
7. Consultar el promedio de ganancias por integrante.





Proyecto # 13:

Se desea construir una aplicación que permita administrar la información de los integrantes de una banda musical y las ganancias obtenidas en los conciertos que ofrecen. Cada concierto debe tener una ganancia mínima de 10.000\$ y máxima de 100.000\$.

La banda está conformada por 4 integrantes. Cada integrante puede donar o no un 10% de sus ganancias a una caridad.

De cada integrante se conoce:

1. Nombre del integrante
2. Instrumento que toca el integrante
3. Ruta de la imagen de la foto del integrante.
4. Cantidad de seguidores del integrante.
5. Cantidad de conciertos en los que se ha presentado.
6. Total de ganancias adquiridas en los conciertos que se ha presentado.
7. Estado de si dona el 10% de sus ganancias a caridades.
8. Género favorito del integrante, puede ser Jazz, Rock, Pop o Tropical.

El sistema debe permitir:

1. Visualizar la información de los integrantes.
2. Modificar un integrante.
3. Consultar la cantidad de integrantes que donan a caridad.
4. Consultar el integrante con más seguidores.
5. Consultar la cantidad de integrantes que tienen un género favorito dado.
6. Consultar el porcentaje de ganancias de un integrante con respecto a la banda.





Proyecto #14

Se desea crear una aplicación que permita manejar el inventario de una tienda, conocer cuánto dinero hay en caja y tener un control de estadísticas de ventas.

La tienda maneja cuatro productos. De cada uno de ellos se maneja la siguiente información:

- Nombre. No puede haber dos productos con el mismo nombre.
- Tipo (puede ser un producto de papelería, supermercado o droguería).
- Cantidad actual del producto en la tienda (número de unidades disponibles para la venta que hay en bodega).
- Cantidad mínima para abastecimiento (número de productos por debajo del cual se puede hacer un nuevo pedido al proveedor).
- Precio base de venta por unidad.

Para calcular el precio final de cada producto, se deben sumar los impuestos que define la ley (IVA). Dichos impuestos dependen del tipo del producto, de la siguiente manera:

- Papelería: 16%
- Supermercado: 4%
- Droguería: 12%.

Eso quiere decir que si un lápiz tiene un precio base de \$10, el precio final será de \$11,6 considerando que un lápiz es un producto de papelería, y sobre estos se debe pagar el 16% de impuestos.

El programa debe permitir al usuario:

1. Visualizar la información de los productos.
2. Vender un producto.
3. Abastecer la tienda con un producto.
4. Cambiar un producto.
5. Calcular estadísticas de ventas:
 - a) El producto más vendido.
 - b) El producto menos vendido.
 - c) La cantidad total de dinero obtenido por las ventas de la tienda.
 - d) La cantidad de dinero promedio obtenido por unidad de producto vendida.



Proyecto #15

Se desea crear una aplicación que permita manejar información y estadísticas sobre unas elecciones. Esta permite manejar la información de 3 candidatos. De cada candidato se conoce:

- El nombre completo.
- La edad.
- El partido político.
- El costo de la campaña.
- La cantidad de votos obtenidos.
- El porcentaje de votos que tiene el candidato (con respecto al total de votos).

Para calcular el costo de la campaña, se tiene en cuenta el medio de comunicación (Televisión, radio o Internet) que influyó a cada votante. De esta manera, para retornar el retorno de la inversión realizada en publicidad, se incrementa el costo de la campaña usando los siguientes valores:

- Voto influenciado por publicidad en Televisión: \$1.000
- Voto influenciado por publicidad en Radio: \$500
- Voto influenciado por publicidad en Internet: \$100

Lo anterior quiere decir que, por cada voto que reciba un candidato influenciado por publicidad en televisión, su costo de campaña se incrementara en \$1.000, del mismo modo por cada voto influenciado por publicidad en radio, su costo de campaña se incrementara en \$500, y finalmente, por cada voto influenciado por publicidad en Internet, su costo de campaña se incrementará en \$100.

Se desea que la aplicación permita:

1. Visualizar la información de un candidato.
2. Votar por un candidato.
3. Calcular el número total de votos.
4. Calcular el costo promedio por campaña.
5. Visualizar estadísticas por rango de edad y género.
6. Vaciar la urna.





Proyecto #16

Se quiere construir una aplicación para manejar un AppStore de juegos. Esta aplicación debe permitir manejar 4 juegos y contar con la posibilidad de compra y venta de licencias de los mismos.

Cada juego se caracteriza por tener:

- Nombre, que es único.
- Categoría a la que pertenece. Puede ser: rompecabezas, acción o deporte.
- Tamaño: memoria que ocupa en kilobytes (KB).
- Precio de la unidad en pesos.
- Cantidad de licencias disponibles.
- Cantidad de licencias vendidas.
- Imagen del juego.

Adicionalmente, el programa debe permitir calcular posibles descuentos para clientes que desean comprar licencias de juego en un volumen alto. En este caso, debido a las políticas de ventas y mercadeo, las promociones vigentes son las siguientes:

1. Si el comprador solicita al menos 25 licencias de juegos de rompecabezas sin importar la cantidad de juegos que solicite de otras categorías, se dará el 20% de descuento sobre el valor total del pedido.
2. Si el comprador solicita al menos 20 licencias de juegos de deportes y 15 licencias de juegos de acción, se dará el 15% de descuento sobre el valor total del pedido.

Si en una compra se cumplen ambas condiciones, sólo se aplica la primera promoción que aplique.

El programa debe permitir:

1. Visualizar la información detallada de cada juego.
2. Comprar licencias de un juego.
3. Vender licencias de un juego.
4. Consultar el juego más vendido.
5. Consultar los descuentos aplicados por un volumen de compra.





Proyecto #17

Se quiere construir una aplicación para administrar un club social. Esta aplicación permite manejar los socios, sus consumos y las personas autorizadas por cada socio para que pueda entrar y usar los servicios del club.

De cada socio se conoce:

- La cédula. No puede haber dos socios con la misma cédula.
- El nombre.
- Los fondos disponibles.
- El tipo de suscripción. Puede ser VIP o Regular. En el club puede haber máximo 3 socios VIP.
- Las facturas sin pagar.
- La lista de personas autorizadas.

Cuando un nuevo socio se afilia al club, debe contar con un fondo inicial para cubrir sus consumos y los de las personas afiliadas, El monto del fondo inicial se determina según el tipo de suscripción del socio: los socios regulares con \$50.000 y los socios VIP con \$100.000. Los socios pueden aumentar sus fondos, pero tienen una restricción máxima, que también depende del tipo de suscripción, de la siguiente manera: regulares \$1'000.000 y VIP \$5'000.000.

Una factura se genera cada vez que un socio o uno de sus asociados hacen un consumo. Para hacer un consumo, el socio debe contar con fondos suficientes para pagarlo. Una factura sólo puede ser pagada si el socio tiene fondos suficientes para hacerlo. Al pagar la factura, esta es eliminada de la lista de facturas sin pagar y se descuenta el valor de los fondos del socio. De cada factura se conoce:

- El concepto
- El valor
- El nombre del socio o de la persona autorizada que generó el consumo.

Cada socio puede agregar personas autorizadas a su lista, siempre y cuando cuente con fondos. Adicionalmente puede eliminar un autorizado de la lista, si este no tiene una factura sin pagar a su nombre.

La aplicación debe permitir:

1. Afiliar un socio al club.
2. Registrar una persona autorizada por un socio.
3. Pagar una factura.
4. Registrar un consumo en la cuenta de un socio
5. Aumentar fondos de la cuenta de un socio



Proyecto #18

Se quiere crear una aplicación para el manejo de una feria empresarial. La feria cuenta con varios puestos de exposición en donde las empresas expositoras muestran sus productos a las empresas visitantes.

Cada puesto tiene restricciones de número mínimo y máximo de personas expositoras según su ubicación:

- Zona Norte: mínimo=4, máximo=8
- Zona Oriente: mínimo=7, máximo=13
- Zona Sur: mínimo=3, máximo=7
- Zona Occidente: mínimo=8, máximo=14
- Zona Centro: mínimo=6, máximo=12

Una empresa visitante debe ingresar su nombre y el número de personas que asistirán a la feria, el número máximo de personas asistentes para una empresa es de 20. Por su parte una empresa expositora debe ingresar su nombre, el número de expositores y la ubicación del puesto de exposición. Por último, es importante saber el porcentaje de ocupación de los puestos de la feria en cualquier instante.

La aplicación debe permitir:

1. Mostrar los integrantes de la empresa.
2. Ingresar una empresa visitante
3. Ingresar una empresa expositora, ocupando el respectivo puesto
4. Desocupar un puesto
5. Sugerir un puesto dado el número de expositores
6. Mostrar el porcentaje de ocupación de puestos





Proyecto #19

En una exposición canina se realizan desfiles de ejemplares de diferentes razas. De cada ejemplar se manejan la siguiente información: nombre (el cual debe ser único en toda la exposición), raza, edad y puntos otorgados en la exposición.

El cliente necesita una aplicación que permita manipular la información de los participantes de los desfiles caninos. Además de la información básica, se tiene también la foto de cada perro.

La aplicación debe:

1. Mostrar la lista de los perros registrados en la exposición, ordenada por raza, puntos o edad.
2. Mostrar la información de un perro específico.
3. Registrar un nuevo perro.
4. Localizar un perro por su nombre.
5. Buscar el perro ganador de la exposición (el que tiene un mayor puntaje asignado).
6. Buscar el perro con el menor puntaje.
7. Buscar el perro más viejo de todos (con mayor edad).





Proyecto #20

Se desea realizar una aplicación para el manejo de una mini empresa de transporte. La empresa tiene capacidad para administrar 4 camiones y sus cargas respectivas.

De cada camión se conoce:

- La matrícula. Este es el identificador del camión.
- La capacidad de carga, en kilogramos.
- El consumo de gasolina, en galones/kilómetro.
- La carga actual, en kilogramos.

La aplicación también permite cargar y descargar los camiones. Al cargar un camión, se debe verificar si la capacidad del camión permite que el camión transporte la carga deseada.

Adicionalmente, se desea que la aplicación sea capaz de determinar el mejor camión disponible para transportar una carga determinada. Se considera el mejor camión aquel que tiene, entre los camiones que una capacidad suficiente para transportar el peso dado, el menor consumo. En caso de no tener un camión con la capacidad suficiente, debe informar que no hay ningún camión disponible.



La aplicación deberá:

1. Registrar un camión con su matrícula, capacidad de carga, consumo de gasolina y carga actual.
2. Consultar los datos de un camión a partir de su matrícula.
3. Cargar un camión verificando que no se exceda su capacidad máxima.
4. Descargar un camión reduciendo su carga actual.
5. Determinar el camión con menor consumo de gasolina que tenga capacidad suficiente para una carga dada.
6. Informar cuando no haya un camión disponible para transportar una carga dada.

