

Preguntas del Examen

1. Análisis de Requerimientos

- Identifica las clases necesarias para desarrollar el sistema propuesto. Justifica tu elección.
- Define los atributos necesarios para cada clase identificada. Explica por qué cada atributo es necesario.
- Describe los métodos que debe tener cada clase para cumplir con los requisitos del sistema.

2. Diseño de la Pantalla

- Elige una de las necesidades especificadas (gestión de clientes, gestión de vehículos, gestión de servicios, gestión de citas, gestión de mecánicos).
- Diseña una pantalla que resuelva dicha necesidad describe brevemente la funcionalidad de cada uno.

3. Desarrollo del Programa

- Implementa en código una de las funcionalidades del sistema basado en la pantalla que has diseñado.
- El código debe incluir la creación de clases con sus respectivos atributos y métodos.

Por cada necesidad a mejorar se debe desarrollar una clase, cada una con sus propios atributos y métodos. A continuación, se describe una por una:

Clase Gestión de Clientes.

Clase que servirá para registrar clientes, modificarlos, consultarlos y borrarlos

Se comienza creando la clase llamada GestionClientes, donde se necesitará crear los siguientes atributos:

Nombre del Atributo	Tipo	Descripción
nombre	str	Nombre completo del cliente
cedula	str	Cédula del cliente
direccion	str	Dirección física del cliente
correo	str	Correo electrónico del cliente
telefono	str	Número de teléfono del cliente
n_visitas	int	Cantidad de visitas pasadas para tener métricas de clientes frecuentes
n_faltas	Int	Cantidad de veces que el cliente no se presentó a la cita

Y los siguientes métodos:

Nombre del método	Tipo del retorno	Descripción
agregar_cliente	void	Método para agregar nuevos clientes, el método consulta cada atributo y debe tomar en cuenta si ya existían duplicados por cédula. si ya existía el cliente, el método debe

		preguntar si sobrescribe, esto debe funcionar para modificar
borrar_cliente	void	Método para borrar cliente especificado por cedula
consultar_cliente	void	Método que despliega información del cliente especificado por cédula o nombre
exportar_clientes	void	Método que exporta la base de datos a Excel
sumar_visita	void	Método que suma 1 al atributo n_visita
sumar_falta	void	Método que suma al atributo n_faltas

Clase Gestión de Vehículos

Se hacen los siguientes atributos:

Nombre del Atributo	Tipo	Descripción
cedula	str	Cedula del propietario. Se hace con la cédula ya que es un número único, de esta forma puede ligarse con los datos de la clase GestionClientes
marca	str	Marca del vehículo
modelo	str	Modelo del vehículo
anio	int	Año del vehículo
placa	str	Registro vehicular
color	str	Color del vehículo
fallas	str	Descripción de fallas, golpes o rayaduras que presenta el vehículo al presentarse en el taller
database	dict	Diccionario con arreglos donde se almacenan los datos/atributos, de la forma {‘Cedula’:[123,456], ‘Marca’:[‘Toyota’,’Nissan’]} y así conteniendo el resto de atributos

Y los siguientes métodos:

Nombre del método	Tipo del retorno	Descripción
agregar_vehiculo	void	Método para agregar nuevo vehiculo, el método consulta cada atributo y debe tomar en cuenta si ya existían duplicados por placa. Esto debe funcionar para modificar
borrar_vehiculo	void	Método para borrar vehiculo especificado por placa
consultar_vehiculo	void	Método que despliega información del vehículo especificado por placa

Clase Gestión de Servicios

Nombre: GestionServicios

Atributos:

Nombre del Atributo	Tipo	Descripción
Id_servicio	str	Código del servicio
servicio	str	Nombre del servicio
descripcion	str	descripción del servicio
costo	float	Costo del servicio en colones

Métodos:

Nombre del método	Tipo del retorno	Descripción
agregar_servicio	void	Método para agregar nuevo servicio, debe generar un código único para el servicio agregado
borrar_servicio	void	Método para borrar servicio
consultar_servicio	void	Método que despliega información y costo del servicio

Clase Gestión de Mecánicos

Atributos:

Nombre del Atributo	Tipo	Descripción
Id_mecanico	str	Código del mecánico
nombre	str	Nombre completo del mecánico
cedula	str	Cédula del mecánico
telefono	str	Número de teléfono del mecanico
especialidad	int	Especialidad del mecánico
disponibilidad	Int	Disponibilidad del mecánico

Y los siguientes métodos:

Nombre del método	Tipo del retorno	Descripción
agregar_mecanico	void	Método para agregar nuevos mecanicos, el método consulta cada atributo y debe tomar en cuenta si ya existían duplicados por cédula. si ya existía el mecanico, el método debe preguntar si sobrescribe, esto debe funcionar para modificar
borrar_mecanico	void	Método para borrar mecanico especificado por cedula

consultar_mecanico	void	Método que despliega información del mecanico especificado por cédula o nombre
--------------------	------	--

Clase Gestión de Citas

Nombre: GestionCitas

Atributos:

Para este caso solo se solicitan los principales atributos como cedula, placa y id del servicio, ya que la demás información se obtiene llamando las respectivas clases

Nombre del Atributo	Tipo	Descripción
Id_reservacion	str	Código unico de reservación
cedula	str	Cédula del cliente
placa	str	Placa del vehiculo
Id_servicio	str	Código del servicio
Id_mecanico	str	Código del mecánico asignado
fecha_hora	datetime	Fecha y hora de la cita
estado	str	Estado de la cita, si está próxima, en progreso, cancelada, o completada

Nombre del método	Tipo del retorno	Descripción
crear_cita	void	Método para agregar una nueva cita, debe preguntar cada atributo y agregar el estado “próxima” y crear un código único de reservación
cancelar_cita	void	Método para cancelar cita, debe modificar el atributo estado a “cancelada”
reagendar_cita	void	Método para reagendar cita
enviar_recordatorio	void	Método para enviar recordatorio de citas que se encuentren a un día de la cita vía correo y sms
completar_cita	void	Método que cambia el estado a completado, en progreso, o ausencia, según se indique
crear_factura	void	Método que crea una factura con el nombre del cliente, servicio, costo, vehiculo
metricas	void	Método que imprime métricas como servicio más solicitado, clientes frecuentes, marcas y modelos más frecuentes en el taller, dinero recaudado por mes, etc

Para el caso de código se escoge construir la clase Gestión de Vehiculos para dar prioridad al control de los vehículos que actualmente están en el Taller