3. Módulo de Monitoramento Energético:

```
def calcular_producao_energia_solar(self, incidencia_solar, area_placa_solar,
eficiencia_placa_solar):

producao_energia_solar = {}  ->  1
      for mes, incidencia in incidencia_solar.items(): -> 1* n
         producao = incidencia * area_placa_solar * eficiencia_placa_solar *30  -> 1
         producao_energia_solar[mes] = producao  -> 1
      return producao_energia_solar  -> 1
```

Pior caso: O(n)
Médio caso: Θ(n)
Melhor caso: Ω(n)

2. Módulo de Gerenciamento de Serviços Automotivos:

```
def agendar(self, nome, horas):

      mec_existente = None   -> 1
      for mec in self.mecanicos: -> 1*n
         if mec[0] == nome: -> 1
            mec_existente = mec -> 1
            break -> 1

      if mec_existente:  -> 1

         if horas in mec_existente[1:]: -> 1
            print("Horário indisponível") -> 1
         else:  -> 1

            mec_existente.append(horas)  -> 1
            print(f"Horário agendado com sucesso para {nome} às {horas}.")  -> 1
      else: -> 1

         novo_mecanico = [nome, horas] -> 1
         self.mecanicos.append(novo_mecanico) -> 1
         print(f"Novo mecânico {nome} criado com horário às {horas}.")  -> 1

      self.salvar_dados() -> 1
```

Pior caso: O(n)
Melhor caso: Ω(n)

Médio caso: Θ(n)

4. Módulo de Relatórios e Análises:

```
def gerar_relatorio_vendas(self):
     vendas_diarias = [1500, 1700, 1800, 1600, 1750]  -> 1
     vendas_semanais = sum(vendas_diarias)  -> 1
     vendas_mensais = vendas_semanais * 4  -> 1
     return {
        "vendas_diarias": vendas_diarias,
        "vendas_semanais": vendas_semanais,
        "vendas_mensais": vendas_mensais,
     }  -> 1

   # Função para gerar relatórios de desempenho de serviços automotivos
   def gerar_relatorio_desempenho_servicos(self):
      tempos_execucao = [30, 45, 60, 35, 40]  -> 1
      satisfacao_clientes = [4, 5, 4, 5, 3]   -> 1
      tempo_medio_execucao = sum(tempos_execucao) / len(tempos_execucao)  -> 1
      media_satisfacao = sum(satisfacao_clientes) / len(satisfacao_clientes)  -> 1
      return {
         "tempos_execucao": tempos_execucao,
         "satisfacao_clientes": satisfacao_clientes,
         "tempo_medio_execucao": tempo_medio_execucao,
         "media_satisfacao": media_satisfacao,
      }  -> 1


   def gerar_relatorio_eficiencia_energetica(self):
      economia_energia_solar = 2000     -> 1
      return {
         "economia_energia_solar": economia_energia_solar,
      } -> 1
```

Pior caso: O(n)
Médio caso: Θ(n)
Melhor caso: Ω(n)

1. Módulo de Controle de Estoque de Produtos:

```
estoque = pd.read_csv(self.arquivo_csv)  -> 1


      if compra: -> 1
         estoque[nome_produto] += quantidade  -> 1


      else: ->1
```

```
            if estoque[nome_produto][0] >= quantidade: ->1
                estoque[nome_produto] -= quantidade ->1
            else: -> 1
                print(f"{cor_mensagem_erro} Não é possível fazer a venda! Estoque indisponível
{Style.RESET_ALL}\n") ->1


        for produto in estoque.columns:  -> 1*n
            if estoque[produto][0] < 5: -> 1
                self.emitir_alerta(produto) -> 1

        estoque.to_csv(self.arquivo_csv, index=False) -> 1
```

Pior caso: O(n)
Médio caso: Θ(n)
Melhor caso: Ω(n)