Onde cada padrão é utilizado:

Obs: os padrões da primeira entrega não serão listados pois já foram validados.

* Builder:
  + Classes:
    - ClienteBuilder
  + Implementação:
    - Cliente
  + Test:
* Prototype
  + Classe:
    - Nenhuma
  + Implementação:
    - Pedido, o método pedirNovamente().
  + Test:
* Flyweight:
  + Classes:
    - Endereco, enderecoFactory
  + Implementado em:
    - Cliente
  + Test:
    - deveRetornarTotalEnderecos em clientTest
* Command
  + Classes:
    - ChamadoSuporte, AbrirChamadoSuporte, FecharChamadoSuporte, Tarefa
  + Implementado em:
    - Cliente, executarTarefa e cancelarTarefa
  + Test:
    - ClienteTest, deveAbrirChamadoSuporte, deveFecharChamadoSuporte, deveCancelarFecharChamadoSuporte
* Memento
  + Classes:
    - Nenhuma
  + Implementado em:
    - Cliente, List<Tarefa> tarefas
  + Test:
    - deveCancelarFecharChamadoSuporte
* Iterator
  + Classes:
    - HistoricoDePedidos
  + Implementado em:
    - Cliente, List<Pedido> pedidos, iterator
  + Test:
    - deveContarTotalPedidosHistoricoDePedidos
* TemplateMethod:
  + Classes:
    - Avaliacao, AvaliacaoComImagem, AvaliacaoSemImagem
  + Implementado em:
    - Cliente, avaliarPedido
    - Pedido, receberAvaliacao
  + Test:
    - deveAvaliarPedidoSemImagemESemPassarPelaGerencia
* Mediator
  + Classes:
    - Gerencia
  + Implementado em:
    - Avaliacao , AvaliacaoComImagem, AvaliacaoSemImagem, notificarGerencia.
  + Test:
    - deveAvaliarPedidoComImagem, deveAvaliarPedidoSemImagemPassandoPorGerencia
* Adapter:
  + Classes:
    - PedidoAdapter
  + Implementado em:
    - .
  + Test:
    - Pedido, deveTestarPedidoIntegracao
* Facade:
  + Classes:
    - Financeiro, ClienteFacade
  + Implementado em:
    - Cliente, fazerPedido()
  + Test:
    - ClientTest, deveLancarExcecaoQuandoClienteTemPendencia

Estrutural

Falta colocar:

1. Proxy (deixar esse pro final, ele vai ter a parte de verificar se o aluno está logado)
2. Composite (se não implementar esse, implementa o bridge...ou nenhum dos dois fodasse)
3. Visitor (talvez esse não seja implementado)
4. Decorator (talvez fazer um esquema tipo colocar um adicional no pedido)
5. Bridge (talvez esse não, ou esse ou o composite)

Já colocados:

* Observer;
* Chain of Responsability;
* State;
* Singleton;
* Strategy;
* Factory
* Prototype
* Memento
* Command
* iterator
* Mediator
* Template method