Questão 1:

A saída gerada pelo código, exceção lançada, foi:

```
Exception in thread "main" java.lang.NullPointerException
   at base.service.impl.CartaServiceImpl.geraCartaAleatoria(CartaServiceImpl.java:27)
   at base.service.impl.BaralhoServiceImpl.preencheAleatorio(BaralhoServiceImpl.java:32)
   at base.controle.Controle.executa(Controle.java:43)
   at Main.main(Main.java:7)
```

A estratégia que adoto para analisar todas as exceções lançadas pelo código e localizar o erro é analisar as linhas indicadas como originárias do erro e ir analisando os métodos, do mais específico para o mais amplo, até encontrar o erro.

Questão 2:

O código pedido na questão 2 foi implementado da seguinte maneira:

```
try {
         for (int i = 0; i < tamanho; ++i) {
            cartas.add(cartaService.geraCartaAleatoria(gerador, maxMana, maxAtaque, maxVida,
null));
         }
      } catch (BaralhoVazioException e) {
         e.printStackTrace();
         System.out.print("Partida encerrada");
         System.exit(0);
      }
```

Questão 3:

No caso do baralho a exceção é tratada na própria classe BaralhoServiceImpl, não tendo avanço no programa. Nesse caso, apenas informamos que há um problema, este deverá ser tratado por terceiros, como faremos do seguinte modo na classe Controle:

```
try {
         mesa = mesaService.adicionaLacaios(mesa, gerador, TipoCarta.LACAIO);
      } catch (MesaNulaException e) {
         e.printStackTrace();
         System.out.print("Partida encerrada");
         System.exit(0);
      }
```

Enquanto na classe MesaServiceImpl, fazemos:

```
if (mesa == null) {
  throw new MesaNulaException("A mesa está vazia");
}
```

Questão 4:

É recomendável que se use as exceções do próprio java para casos como no exemplo anterior pois se tem mais segurança no procedimento realizado e diminui-se a possibilidade de erros.

Questão 5:

O código como pedido na questão 5 fica da seguinte maneira:

```
if ((mesa.getMaoP().size() < 3) || (mesa.getMaoS().size() < 3)) {
        throw new ValorInvalidoException("Valor inicial da maoP: " + mesa.getMaoP().size() +
        "Valor inicial maoS: " + mesa.getMaoS().size());
    }
```

Questão 6:

Para conseguir tal feito podemos imprimir "Partida encerrada" antes de finalizar a execução do programa, como feito nos seguintes trechos da classe "Controle":

```
catch (ValorInvalidoException e) {
        e.printStackTrace();
        System.out.print("Partida encerrada");
        System.exit(0);
    }
```

ou em:

```
catch (MesaNulaException e) {
        e.printStackTrace();
        System.out.print("Partida encerrada");
        System.exit(0);
    }
```

Questão 7:

O trecho de código mostrado no enunciado é uma má prática de programação pois ele aborda o erro de uma forma muito genérica. Isto é, a exceção fornecida não traria muitos insumos para descobrirmos qual o erro a ser tratado.