Brazilian Electronic Journal of Mathematics, v.1 - n.2, jul/dez, 2020. ISSN: 2675-1313

# Estatística de jogos: blackjack

## Game's statistics: blackjack

Fábio Pascoal dos Reis[1]
Cristiano de Siqueira Esteves[2]
Elisangela Aparecida y Castro[3]

**Resumo**. O principal objetivo deste artigo é compreender a fundo o mecanismo de funcionamento do jogo de cartas *blackjack*, bem como estabelecer uma estratégia simples para maximizar as chances de vitória. Inicialmente, foi feito um breve histórico sobre esse jogo, discutindo suas origens e evolução. Explicou-se em detalhes as regras do blackjack (sistema de pontuação e procedimentos do jogo). Para contextualizar a popularidade do blackjack, foram citados livros, filmes e jogos de vídeo game onde o mesmo apresenta algum protagonismo. A seguir, foi apresentado na íntegra o programa computacional utilizado para fazer o levantamento numérico das probabilidades de obter uma dada pontuação usando o sistema de pontos do blackjack, quando são sorteadas aleatoriamente 2, 3, 4, 5, 6, 7, 8, 9, 10 e 11 cartas de um baralho francês comum e completo, contendo 52 cartas. Explicou-se minuciosamente o algorítimo desse programa, apresentou-se seus resultados e estes foram analisados em detalhes. Por fim, simulou-se várias rodadas de blackjack com apenas um jogador e um crupiê. Nessas simulações, proibiu-se que o jogador pedisse novas cartas quando sua pontuação ultrapassasse uma certa pontuação máxima (Pmax) ou quando ele chegasse a um certo número máximo de cartas ($N_{max}$). Mais uma vez, apresentou-se o programa computacional completo e o mesmo foi discutido linha a linha. Analisando os resultados de 100 simulações, onde varreu-se o valor de $P_{max}$ de 12 a 21 e o valor de $N_{max}$ de 2 a 11, procurou-se pela combinação de valores de $P_{max}$ e $N_{max}$ que trazem a melhor chance de vitória. Na melhor situação, as chances de vitória do crupiê são apenas 4,7% maiores do que as do jogador.

[1]Instituto de Ciências Exatas e Naturais do Pontal, Universidade Federal de Uberlândia, fabiopr@ufu.br
[2]Instituto de Ciências Exatas e Naturais do Pontal, Universidade Federal de Uberlândia, cristiano@ufu.br
[3]Instituto de Ciências Exatas e Naturais do Pontal, Universidade Federal de Uberlândia, elisangela.castro@ufu.br

**Palavras-chave**. Blackjack. Jogos de cartas. Estatística. Probabilidade. Algoritmo computacional.

**Abstract**. The main objectives of this article is to understand in depth the mechanism of the blackjack and establish a simple strategy to maximize the chances of winning the hand. First, we make a brief history about blackjack, discussing its origins and evolution. We explain in detail the rules of blackjack (scoring system and game procedures). In order to contextualize the popularity of blackjack, we provide some examples of books, films and video games where it has some role. Next, we present the computer program that we developed to compute the numeric probabilities to obtaining a given score using the blackjack points system, when we randomly draw 2, 3, 4, 5, 6, 7, 8, 9, 10 and 11 cards from a standard 52-card deck. We explain in detail the algorithm of this program, present its results and analyze them. Finally, we simulate several hands of blackjack with just one player and a dealer. In these simulations, we do not allow the player to hit on hands worth more than a certain maximum score ($P_{max}$) or to get more than a certain maximum number of cards ($N_{max}$). Once again, we present the complete computer program and discuss it line by line. Analyzing the results of 100 simulations, where we sweep the value of $P_{max}$ from 12 to 21 and the value of $N_{max}$ from 2 to 11, we look for the combination of values of $P_{max}$ and $N_{max}$ that bring the best chance of victory. The best situation is the one that the dealer's chances of winning are only 4.7 % higher than the player's.

**Keywords**. Blackjack. Card games. Statistics. Probability. Computational algorithm.

# 1 Introdução

O *blackjack* é um dos jogos de cassino mais populares do mundo. Sua origem é desconhecida, mas alguns historiadores defendem que na Roma Antiga já se jogava uma forma primitiva de blackjack com blocos de madeira numerados [1]. A primeira referência escrita a um jogo similar, o *veinteuna*, é encontrada no conto *Rinconete y Cortadillo* de Miguel de Cervantes do livro *Novelas Ejemplares* [2]. Os jogos de carta ganharam ainda mais popularidade nos cassinos de Las Vegas em 1931, quando o estado de Nevada tornou legal o jogo a dinheiro. Para atrair ainda mais jogadores, os cassinos introduziram uma regra especial no jogo 21, como era chamado na época. Se o jogador saísse com um valete preto (black jacks em inglês) e um ás de espadas, o cassino pagaria 10 vezes a aposta do jogador. Essa regra especial não durou muito tempo, mas o nome blackjack pegou.

As regras simples do blackjack são um dos motivos de sua popularidade [3]. Nele, um ou mais jogadores jogam contra o crupiê sem competir entre si, utilizando 1 a 8 baralhos franceses de 52 cartas. Cada ás vale 11 ou 1 ponto, o que for mais conveniente para aquele que os retira. As cartas numeradas têm pontuações iguais aos números nelas estampados. Valetes, damas e reis valem 10 pontos. Os ases de uma mesma mão podem assumir valores diferentes, caso seja conveniente para aquele que os tem. No início de cada rodada, os jogadores depositam suas apostas nos lugares delimitados na mesa. Depois disso, o crupiê distribui duas cartas para cada jogador com as faces viradas para cima e retira duas cartas para si, mas revela apenas uma delas. Quando solicitado, o jogador pode pedir cartas (uma a uma) para aumentar sua pontuação total. Depois que todos os jogadores completarem suas mãos, é a vez do crupiê. Inicialmente ele revela sua carta oculta e, se necessário, começa a retirar cartas uma a uma. O crupiê está limitado a 4 cartas extras e, em qualquer momento, se somar 17 pontos ou mais, não pode retirar mais cartas. O vencedor será aquele que somar a maior pontuação. Jogador e crupiê não podem exceder 21 pontos. Aquele que exceder (estourar) a pontuação perde automaticamente. Para um melhor entendimento do sistema de pontuação, apresentamos alguns exemplos na Tabela 1.

Tabela 1: Exemplos de combinações de cartas e suas pontuações.

| Combinação de Cartas | Pontuação |
|---|---|
| Q 10 A | 21 |
| J A | 21 |
| J 2 A | 13 |
| A A | 12 |

Existem vários livros sobre blackjack. A maioria deles trata de sistemas de contagem de cartas que prometem aumentar as chances de *quebrar a banca* (veja por exemplo [4], [5] e [6]). Como em todos os jogos de cartas, a sorte é muito importante no blackjack, mas o jogador tem a chance de agir quando decide entre pedir uma carta ou parar. O método de contagem ajuda o jogador nesse ponto, indicando de uma forma simplificada e prática a probabilidade de receber cartas com pontuação alta ou baixa.

O blackjack também marca presença no mundo do entretenimento. Alguns filmes como *Rain Man* (1988), *The Last Cassino* (2004) e *21* (2008) dão um papel de destaque a esse jogo de cartas e às técnicas de contagem de cartas. Também é possível jogar blackjack como minijogo dentro de alguns vídeo games como *Fallout: New Vegas* (2010), *Resident Evil 7* (2017) e *Red Dead Redemption 2* (2018). Como curiosidade, vale ressaltar que a ideia deste trabalho nasceu quando um dos autores jogava *Red Dead Redemption 2*. Neste jogo, propõem-se um desafio de ganhar 3 rodadas de blackjack com 5 ou mais cartas na mão.

Neste artigo, levantamos numericamente as probabilidades de obter todas as pontuações resultantes quando retiramos de 2 a 11 cartas de um baralho francês completo de 52 cartas. Além disso, simulamos várias rodadas de jogos de blackjack com apenas um jogador e o crupiê. Limitamos o jogador a pedir cartas até ultrapassar uma pontuação $P_{max}$ ou a chegar em um número máximo de cartas $N_{max}$. De posse dessas simulações, procuramos qual a melhor estratégia.

## 2 Levantamento estatístico das pontuações

Nesta seção, estudamos o levantamento estatístico da pontuação resultante segundo as regras do blackjack, quando tiramos de 2 a 11 cartas de um baralho francês de 52 cartas. Como nesse caso, a ordem das cartas não influencia no resultado, se o número de cartas retiradas do baralho for $N$, o número total de combinações será

$$C_N^{52} = \frac{52!}{(52-N)!\,N!}.$$

Analisar todas as combinações, seria uma tarefa muito cansativa e demorada, por esse motivo, elaboramos programas computacionais para esse fim. A seguir, apresentamos o código fonte na linguagem c++ da versão mais otimizada:

```
#include<iostream>
using namespace std;
int N = 7; int j; int Soma; int PA;
int Valor[53]; int i[20]; int H[22];
int main() {
 for (j = 1; j <= 40; j++) { Valor[j] = (j + 3) / 4; }
```

```
 7| for (j = 41; j <= 52; j++) { Valor[j] = 10; }
 8| for (j = 1; j <= N; j++) { i[j] = j; }
 9| while (i[1] < 54 - N) {
10|  PA = 0; if (i[1] <= 4) { PA = 1; }
11|  Soma = Valor[i[1]];
12|  for (j = 2; j <= N; j++) {
13|   if (Soma <= 21) {
14|    Soma = Soma + Valor[i[j]];
15|    if (Soma > 21) { i[j]++; }}}
16|  if (PA == 1 && Soma <= 11) { Soma = Soma + 10; }
17|  if (Soma <= 21) { H[Soma]++; }
18|  i[N]++;
19|  for (j = 0; j <= N - 1; j++) { if (i[N - j] > 52 - j) { i[N - 1 - j]++; } }
20|  for (j = 0; j <= N - 2; j++) { if (i[N - j] > 52 - j) { i[N - j] = N - j; } }
21|  for (j = 1; j <= N - 1; j++) { if (i[j] >= i[j + 1]) { i[j + 1] = i[j] + 1; }}}
22| for (j = 1; j <= 21; j++) { cout << j << "\t" << H[j] << endl; }
23| return 0;}
```

Nas linhas 3 e 4, declaramos as variáveis utilizadas no programa. As variáveis `N`, `j`, `Soma` e `PA`, correspondem ao número de cartas sorteadas $N$, ao índice de laço, à soma dos valores da carta sorteada e à presença de ases nas cartas sorteadas, respectivamente. Os vetores `Valor`, `i` e `H`, correspondem aos valores de cada carta do baralho, aos índices das $N$ cartas sorteadas em um determinado passo do laço principal e ao número de combinações que dão origem a uma determinada pontuação.

Nas linhas 6 e 7, associamos um índice a cada carta e declaramos sua pontuação. Associamos as primeiras 4 componentes aos ases, e inicialmente, atribuímos a elas o valor 1. As componentes de 5 a 40, representam as cartas numeradas: as cartas 2 são representadas pelas componentes 5 a 8, as cartas 3 são representadas pelas componentes 9 a 12, as cartas 4 são representadas pelas componentes 13 a 16 e assim por diante. Note que na linha 6, temos a expressão `Valor[j] = (j + 3) / 4`, que é uma manipulação algébrica envolve apenas a variável `j` e constantes, ambas do tipo int. Neste caso, o programa força o resultado da divisão a ser também do tipo int, descartando a parte decimal do resultado da divisão. Para `j` entre 17 e 20, por exmeplo, teremos que `Valor` será igual a 5. Associamos valetes, damas e reis às últimas 12 componentes (posto 41 a 52 do vetor), e estes assumiram o valor 10. Na linha 8, iniciamos os valores do vetor `i`.

Na linha 9, começamos o laço principal. Esse laço funciona enquanto o índice da primeira carta for menor que $54 - N$, o que varre todas as combinações possíveis de $N$ cartas.

A linha 10 é dedicada a atribuir o valor 0 à variável `PA` se não houver um ás entre as cartas sorteadas ou o valor 1, caso haja ao menos um às entre elas.

As linhas 11 a 15 são dedicadas ao cálculo da variável `Soma`, que corresponde à pontuação total das cartas sorteadas sem a correção devido à presença de ases. Nessas linhas, ainda incluímos um mecanismo para eliminar do laço parte das combinações que geram uma pontuação superior a 22. Isso foi feito para diminuir o tempo de execução do programa. Para $N = 10$, por exemplo, esse mecanismo encurtou o tempo de execução previsto de mais de 1.000 horas, para algo menor que um minuto. Como sabemos calcular analiticamente o número total de combinações, não perdemos nenhuma informação relevante.

A linha 16 é dedicada a corrigir o valor de `Soma`, caso haja um ás entre as cartas sorteadas.

Na linha 17, construiu-se o vetor `H`. A cada passo do laço principal, caso `Soma`$\leq 21$, somávamos 1 à componente `Soma` do vetor `H`.

Antes de fecharmos o laço principal, nas linhas 18 à 21, atualizamos os valores do vetor `i`.

A linha 22 imprime na tela o resultado do programa.

A seguir, na Tabela 2, apresentamos os resultados obtidos pelo programa.

Tabela 2: Levantamento estatístico das combinações que geram uma dada pontuação sorteadas $N$ cartas.

| $P$ | $N=2$ | $N=3$ | $N=4$ | $N=5$ | $N=6$ | $N=7$ | $N=8$ | $N=9$ | $N=10$ |
|---|---|---|---|---|---|---|---|---|---|
| 4 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 22 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 32 | 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 38 | 48 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 48 | 92 | 16 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 54 | 136 | 52 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11 | 64 | 200 | 128 | 4 | 0 | 0 | 0 | 0 | 0 |
| 12 | 124 | 620 | 1385 | 1552 | 734 | 112 | 1 | 0 | 0 |
| 13 | 128 | 828 | 1936 | 2492 | 1472 | 324 | 16 | 0 | 0 |
| 14 | 118 | 984 | 2781 | 3784 | 2698 | 816 | 68 | 0 | 0 |
| 15 | 112 | 1172 | 3712 | 5724 | 4624 | 1756 | 240 | 4 | 0 |
| 16 | 102 | 1304 | 4961 | 8348 | 7570 | 3500 | 649 | 28 | 0 |
| 17 | 96 | 1448 | 6192 | 12016 | 12016 | 6412 | 1568 | 116 | 0 |
| 18 | 86 | 1540 | 7700 | 16612 | 18512 | 11244 | 3360 | 368 | 6 |
| 19 | 80 | 1640 | 9184 | 22384 | 27904 | 18808 | 6688 | 1004 | 32 |
| 20 | 136 | 1688 | 10837 | 29432 | 40548 | 30712 | 12494 | 2368 | 140 |
| 21 | 64 | 2012 | 12384 | 37624 | 57904 | 48276 | 22320 | 5164 | 432 |

Na Figura 1, apresentamos o gráfico da probabilidade de obter faixas de pontuação sorteando um número $N$ de cartas. Ao analisar os resultados do programa, é possível chegar às seguintes conclusões:

- Para 11 cartas, há apenas 4 combinações que geram 21 pontos. Todo o restante gera mais de 22 pontos

- Há uma chance de 4,8% de o jogador obter 21 pontos apenas com as duas cartas iniciais.

- Existem 2 combinações (sem importar a ordem) que dão origem a um blackjack (ás de espadas e um valete preto), portanto uma pequena chance de aproximadamente 0,2% de obtê-lo.
- As chances de obter 21 pontos são maiores com 3 cartas (9,1 %).
- Com duas cartas, não há possibilidade de obter uma pontuação maior que 21. No entanto, as chances de estourar a pontuação aumentam conforme aumentamos o número de cartas.
- Acima de 12 cartas é impossível não estourar a pontuação.

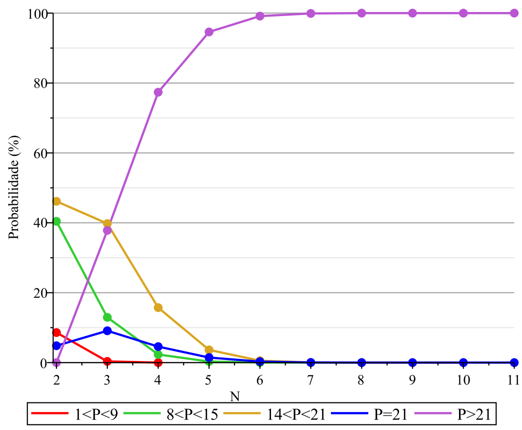


Figura 1: Gráfico da probabilidade de obter faixas de pontuação sorteando um número $N$ de cartas

# 3 Simulações

Nesta seção estudamos as simulações de várias rodadas de jogos de blackjack com apenas um jogador e o crupiê. Limitamos o jogador a pedir cartas até ultrapassar uma pontuação $P_{max}$ ou a chegar em um número máximo de cartas $N_{max}$. Ao avaliar se convém parar de pedir cartas, tanto jogador quanto crupiê devem tomar essa decisão contabilizando o valor do ás como 11, sempre que for possível.

A seguir, apresentamos o código fonte do programa que utilizamos para fazer esse estudo (linguagem c++).

```
#include <iostream>
#include <string>
using namespace std;
```

```
 4| int i; int j; int k; int Temp; int NJ;
 5| int PAJ; int PAC; int SomaJ; int SomaC;
 6| int NmaxJ = 2; int NmaxC = 6; int PmaxJ = 21; int PmaxC = 17;
 7| int DE; int VE; int D; int V; int E; int Valor[53];
 8| int Aleatorio(int maior) { return 1 + rand() % maior; }
 9| int main(){
10|   srand(22);
11|   for (i = 1; i <= 1000000; i++) {
12|     for (j = 1; j <= 4; j++) { Valor[j] = 11; }
13|     for (j = 5; j <= 40; j++) { Valor[j] = (j + 3) / 4; }
14|     for (j = 41; j <= 52; j++) { Valor[j] = 10; }
15|     SomaJ = 0; SomaC = 0; PAJ = 0; PAC = 0; NJ = 0;
16|     for (j = 1; j <= NmaxJ; j++) {
17|       if (SomaJ < PmaxJ) {
18|         Temp = Aleatorio(53 - j); NJ++;
19|         if (Valor[Temp] == 11) { PAJ = 1; }
20|         SomaJ = SomaJ + Valor[Temp];
21|         for (k = Temp; k <= 52 - j; k++) { Valor[k]=Valor[k+1]; } Valor[53-j]=0;}}
22|     for (j = 1; j <= NmaxC; j++) {
23|       if (SomaC < PmaxC) {
24|         Temp = Aleatorio(52 - NJ - j);
25|         if (Valor[Temp] == 11) { PAC = 1; }
26|         SomaC = SomaC + Valor[Temp];
27|         for (k=Temp; k<=51-NJ-j; k++) {Valor[k]=Valor[k+1];} Valor[52-NJ-j]=0;}}
28|     if (SomaJ > 21 && PAJ == 1) { SomaJ = SomaJ - 10; }
29|     if (SomaC > 21 && PAC == 1) { SomaC = SomaC - 10; }
30|     if (SomaJ > 21) { DE++; }
31|     if (SomaJ < 22 && SomaC > 21) { VE++; }
32|     if (SomaJ < 22 && SomaC < 22 && SomaJ > SomaC) { V++; }
33|     if (SomaJ < 22 && SomaC < 22 && SomaJ < SomaC) { D++; }
34|     if (SomaJ < 22 && SomaC < 22 && SomaJ == SomaC) { E++; }}
35|   cout << "Resultado ( NmaxJ=" << NmaxJ << ", PmaxJ=" << PmaxJ << ")" << endl;
36|   cout << "Vitorias Simples: \t" << V << endl;
37|   cout << "Vitorias por Estouro: \t" << VE << endl;
38|   cout << "Derrotas Simples: \t" << D << endl;
39|   cout << "Derrotas por Estouro: \t" << DE << endl;
40|   cout << "Empates: \t \t" << E << endl;
41|   return 0;}
```

Nas linhas 4 a 7, definimos as variáveis utilizadas no programa. `i`, `j`, `k` são índices de laço usados no programa. A variável `Temp` assume temporariamente um valor gerado aleatoriamente. A variável `NJ` corresponde ao número total de cartas retiradas pelo jogador a cada rodada. `PAJ` e `PAC` são variáveis que indicam a presença de ases nas cartas retiradas pelo jogador e pelo crupiê. As variáveis `SomaJ` e `SomaC`, correspondem à pontuação total do jogador e do crupiê. `NmaxJ`, `NmaxC`, `PmaxJ` e `PmaxC`, correspondem ao número máximo de cartas que podem ser retiradas pelo jogador e pelo crupiê e à pontuação acima da qual param de pedir cartas. `DE`, `VE`, `V`, `D` e `E`, correspondem aos contadores de derrotas ou vitórias (simples ou por estouro de pontuação) e empates em cada rodada. Por fim, o vetor `Valor` indica o valor de cada uma das 52 cartas do baralho. A associação dos índices de cada carta e seus respectivos valores, foi feita de forma análoga à seção anterior, no entanto, neste programa, foi mais conveniente associar aos ases o valor 11.

Na linha 8, definimos a função `Aleatorio`, que gera um número aleatório entre 1 e `maior`. A variável `maior` é definida dentro da função, ainda na linha 8. Na linha 11, definimos a semente dos números aleatórios. Na linha 11, começamos o laço principal do programa. Ele corresponde à simulação das 1.000.000 de rodadas a serem simuladas.

Nas linhas 12 a 14, determinamos os valores das componentes do vetor `Valor`. Na linha 15, zeramos os valores das variáveis `SomaJ`, `SomaC`, `PAJ`, `PAC` e `NJ`.

Nas linhas 16 a 21, temos o laço em que as cartas do jogador são sorteadas. No laço garantimos que não serão retiradas mais que $N_{max}$ cartas. Uma condição tipo *if* (na linha 17) garante que a pontuação total do jogador seja alterada apenas enquanto ela for menor que $P_{max}$. Na linha 18 sorteamos as cartas e atualizamos o valor de `NJ`. Na linha 19, verificamos se a carta sorteada era um ás, nesse caso a variável $PAJ$ assume o valor 1. Na linha 20 atualizamos o valor da pontuação do jogador. Por fim, na linha 21 "retiramos a carta sorteada do baralho".

Nas linhas 22 a 27, temos o laço em que as cartas do crupiê são sorteadas. O procedimento é muito similar ao usado no sorteio das cartas do jogador.

Nas linhas 28 e 29, corrigimos os valores das pontuações do jogador e do crupiê, caso tenham sido sorteados ases e seja necessário.

Nas linhas 30 a 34, verificamos se o resultado do sorteio resulta em vitória do jogador (simples ou por estouro da pontuação pelo crupiê), derrota do jogador (simples ou por estouro da pontuação pelo jogador) ou empate. Na linha 34, fechamos o laço principal.

As linhas 35 a 40 imprimem os resultados do programa.

Ao executarmos o programa diversas vezes, alterando o valor da semente dos números aleatórios, percebemos uma variação muito sutil no resultado final. Fizemos várias simulações (30), mudando a semente e mantendo fixos $N_{max}$ e $P_{max}$. De posse desses dados, calculamos a média e desvio padrão do número de vitórias, derrotas e empates. Repetimos o procedimento para todos os valores de $N_{max}$ e $P_{max}$. Em todos os casos, o desvio padrão tem ordem de grandeza de $10^2$.

Nas Tabelas 3, 4 e 5, apresentamos os resultados do programa. Analisando esses resultados, podemos chegar às seguintes conclusões

- Ao utilizar a estratégia em estudo, os melhores resultados ocorrem quando o jogador decide adotar $P_{max} = 15$ e $N_{max} \geq 5$ (43,5 % de vitórias, 48,1% de derrotas e 8,4% de empates ).

- Os piores resultados ocorrem para $P_{max} = 21$ e $N_{max} \geq 5$ (21,0 % de vitórias, 76,0% de derrotas e 3,0% de empates ).

- Para $N_{max} = 2$, as chances de vitória, derrota e empate independem de $P_{max}$. Isso já era um resultado esperado, uma vez que, ao escolher $N_{max} = 2$, o jogador escolhe ficar apenas com as cartas distribuídas inicialmente pelo crupiê sem antes avaliar a pontuação das mesmas.

- Para um dado $P_{max}$ fixo, as probabilidades variam com $N_{max}$ até que estabilizam para $N_{max} \geq 5$ ($13 \leq P_{max} \leq 19$) ou $N_{max} \geq 6$ ($20 \leq P_{max} \leq 21$).
- Para $P_{max} \leq 16$, os resultados melhoram (aumento das chances de vitória e diminuição das chances de derrota) com o aumento de $N_{max}$.
- Para $P_{max} = 21$, os resultados pioram com o aumento de $N_{max}$.

Tabela 3: Chances de vitória em função de $N_{max}$ e $P_{max}$.

| $P_{max}$ | $N_{max} = 2$ | $N_{max} = 3$ | $N_{max} = 4$ | $N_{max} = 5$ | $6 \leq N_{max} \leq 11$ |
|---|---|---|---|---|---|
| 13 | 37,7 % | 42,1 % | 42,7 % | 42,8 % | 42,8 % |
| 14 | 37,7 % | 42,6 % | 43,3 % | 43,3 % | 43,3 % |
| 15 | 37,7 % | 42,8 % | 43,4 % | 43,5 % | 43,5 % |
| 16 | 37,7 % | 42,6 % | 43,2 % | 43,2 % | 43,2 % |
| 17 | 37,7 % | 42,3 % | 42,6 % | 42,5 % | 42,5 % |
| 18 | 37,7 % | 41,8 % | 41,5 % | 41,4 % | 41,4 % |
| 19 | 37,7 % | 40,2 % | 38,6 % | 38,1 % | 38,1 % |
| 20 | 37,7 % | 37,8 % | 33,6 % | 32,4 % | 32,3 % |
| 21 | 37,7 % | 31,6 % | 23,7 % | 21,3 % | 21,0 % |

Tabela 4: Chances de derrota em função de $N_{max}$ e $P_{max}$.

| $P_{max}$ | $N_{max} = 2$ | $N_{max} = 3$ | $N_{max} = 4$ | $N_{max} = 5$ | $6 \leq N_{max} \leq 11$ |
|---|---|---|---|---|---|
| 13 | 57,0 % | 50,7 % | 49,8 % | 49,7 % | 49,7 % |
| 14 | 57,0 % | 49,8 % | 48,7 % | 48,7 % | 48,7 % |
| 15 | 57,0 % | 49,2 % | 48,2 % | 48,1 % | 48,1 % |
| 16 | 57,0 % | 49,0 % | 48,0 % | 47,9 % | 47,9 % |
| 17 | 57,0 % | 49,0 % | 48,1 % | 48,1 % | 48,1 % |
| 18 | 57,0 % | 50,2 % | 50,2 % | 50,4 % | 50,4 % |
| 19 | 57,0 % | 52,3 % | 54,2 % | 54,9 % | 54,9 % |
| 20 | 57,0 % | 55,2 % | 60,4 % | 61,8 % | 62,0 % |
| 21 | 57,0 % | 62,9 % | 72,6 % | 75,7 % | 76,0 % |

Na Figura 2, apresentamos de forma mais detalhada as chances de vitória, derrota e empate em função de $N_{max}$ para $P_{max} = 15$. Analisando esse gráfico, chegamos às seguintes conclusões:

Tabela 5: Chances de empate em função de $N_{max}$ e $P_{max}$.

| $P_{max}$ | $N_{max} = 2$ | $N_{max} = 3$ | $N_{max} = 4$ | $N_{max} = 5$ | $6 \leq N_{max} \leq 11$ |
|---|---|---|---|---|---|
| 13 | 5,4 % | 7,3 % | 7,5 % | 7,5 % | 7,5 % |
| 14 | 5,4 % | 7,7 % | 8,0 % | 8,0 % | 8,0 % |
| 15 | 5,4 % | 8,0 % | 8,4 % | 8,4 % | 8,4 % |
| 16 | 5,4 % | 8,3 % | 8,9 % | 8,9 % | 8,9 % |
| 17 | 5,4 % | 8,6 % | 9,3 % | 9,4 % | 9,4 % |
| 18 | 5,4 % | 8,0 % | 8,2 % | 8,2 % | 8,2 % |
| 19 | 5,4 % | 7,5 % | 7,2 % | 7,0 % | 7,0 % |
| 20 | 5,4 % | 6,9 % | 6,0 % | 5,7 % | 5,7 % |
| 21 | 5,4 % | 5,5 % | 3,7 % | 3,1 % | 3,0 % |

- Para $P_{max} = 15$, as chances de vitória por pontuação aumentam com $N_{max}$ até estabilizarem em 22,1 % para $N_{max} \geq 5$.

- Já as chances de vitória por estouro, ou seja, as chances que o jogador tem de vencer pelo crupiê somar uma pontuação superior a 21, apresentam o valor máximo de 21,9 % quando o jogador escolhe $N_{max} = 3$.

- As chances de derrota por pontuação caem com $N_{max}$ até estabilizarem em 36,2 % para $N_{max} \geq 5$.

- As chances de derrota por estouro de pontuação começa em zero para $N_{max} = 2$ e crescem com $N_{max}$ até estabilizarem em 11,5 % para $N_{max} \geq 5$.

As simulações indicam que os resultados sempre estabilizam para $N_{max} \geq 5$, ou seja, as chances de o jogador vencer são as mesmas se ele se propõe a retirar cartas até ter uma mão com 5, 6, 7 ou mais cartas. Para entender melhor porque isso acontece, alteramos o programa para contabilizar em quantas vezes o jogador parava de pedir cartas apenas porque seu número de cartas havia ultrapassado $N_{max}$, apenas porque sua pontuação havia ultrapassado $P_{max}$ e as duas condições ao mesmo tempo. Esse resultado é apresentado na Figura 3. Algumas observações:

- Se o jogador escolher $P_{max} = 15$ e $N_{max} \geq 5$, o que o faz parar de pedir cartas é ter ultrapassado a pontuação $P_{max}$.

- As simulações foram repetidas para diferentes sementes de número aleatório e os resultados foram muito parecidos, com discrepância inferior a 0,1 %.

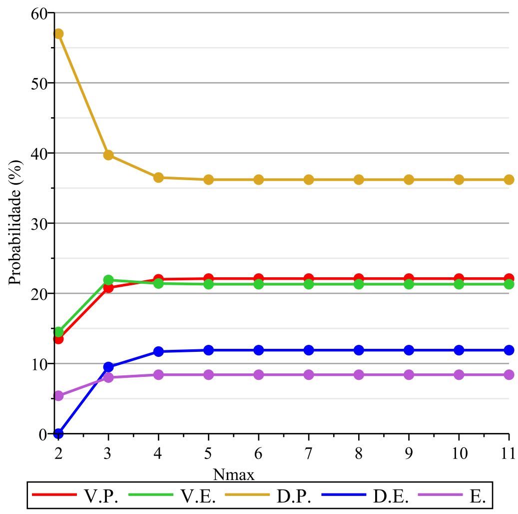


Figura 2: Gráfico das chances de vitória por pontuação (V.P.), vitória por estouro (V.E.), derrota por pontuação (D.P.), derrota por estouro (D.E.) e empate (E.) em função de $N_{max}$ para $P_{max} = 15$.

- As simulações foram repetidas para valores diferentes de $P_{max} = 15$. Para $6 \leq N_{max} \leq 11$, foi observado que faz o jogador parar de pedir cartas é ter ultrapassado a pontuação $P_{max}$. Também foi observada a pouca influência das sementes de número aleatório (mais uma vez, as discrepâncias apresentadas eram inferiores a 0,1 %).

Por fim, voltemos ao desafio proposto no jogo Red Dead Redemption 2, ganhar partidas de blackjack com uma mão com 5 ou mais cartas. Neste caso, para cumprir o desafio, o foco do jogador não é apenas ganhar a partida. A melhor estratégia, nesse caso, seria escolher $P_{max} = 21$ e $N_{max} = 5$. Modificamos o programa das simulações inserindo algumas estruturas condicionais, e chegamos aos seguintes resultados:

- As chances de fazer 21 pontos antes de pedir a quinta carta são de 8,2%.
- As chances de estourar a pontuação neste processo são de 59,0%.
- As chances de conseguir pedir a quinta carta são de 32,9%.
- As chances de ganhar com uma mão de 5 cartas são de apenas 1,1%.

# 4 Conclusão

Se o jogador adotar a estratégia de pedir cartas até ultrapassar uma pontuação $P_{max}$ ou até chegar em um número máximo de cartas $N_{max}$, ele obterá o melhor resultado ao escolher

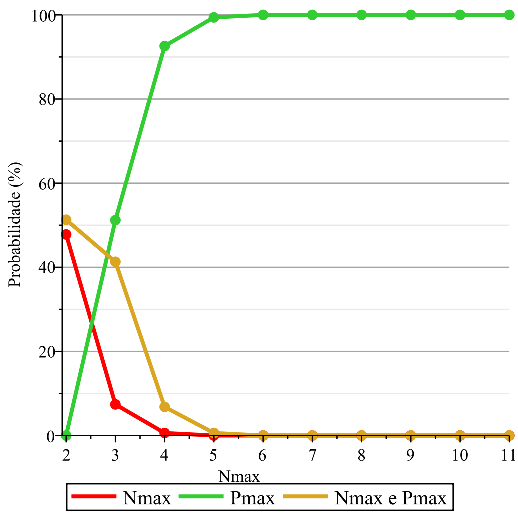


Figura 3: Percentual de vezes que o jogador para de pedir cartas apenas porque seu número de cartas ultrapassa $N_{max}$, apenas porque sua pontuação ultrapassa $P_max$ ou quando atinge as duas condições ao mesmo tempo em função de $N_{max}$. Esse gráfico foi feito com $P_{max} = 15$ e semente de 20.000.

$P_{max} = 15$ e $5 \leq N_{max} \leq 11$. Nesse caso, as chances de o crupiê vencer são 4,7% maiores que as do jogador. Existem estratégias mais eficientes, como o método High - Low [5] que promete reduzir essa diferença a menos de 1%, no entanto a estratégia apresentada aqui é mais simples.

## Agradecimentos

Gostaríamos de agradecer a Luiz Fernando de Oliveira, Alex Farah Pereira e Henrique Borges Machado pelas valiosas dicas e discussões. Agradecemos também aos árbitros que avaliaram esse trabalho. Suas críticas e sugestões ajudaram muito a enriquecer o texto deste artigo.

# Referências

[1] HISTORY of Black Jack. **The world of playng cards**. Diponível em: `https://www.wopc.co.uk/history/blackjack/blackjack`. Acesso em: 02 abr. 2019.

[2] CERVANTES, Miguel de. **Novelas Exemplares**, São Paulo, Cosac & Naify, 2015.

[3] CARDS Games Rules: Blackjack. **Pagat.com.** Diponível em: https://www.pagat.com/banking/blackjack.html. Acesso em 02 abr. 2019.

[4] THORP E. O. **Beat the Dealer**, New York, Random House, 1966.

[5] WONG, S. **Professional Blackjack**, Las Vegas, Piyee Press, 1994.

[6] GRIFFIN, P. A. **The Theory of Blackjack: The Compleat Card Counter's Guide to the Casino Game of 21**, Las Vegas, Huntington Press, 1999.

Submetido em 03 mar. 2020
Aceito em 23 maio. 2020