

Código Fonte:

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Simulador de Amigo Oculto</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>Simulador de Amigo Oculto</h1>
    <form id="form">
      <input type="text" id="name" placeholder="Digite um nome" required>
      <button type="button" id="add">Adicionar Participante</button>
    </form>
    <ul id="participants"></ul>
    <button id="draw" class="draw-btn">Sortear Amigo Oculto</button>
    <div id="results" class="hidden">
      <h2>Resultados:</h2>
      <ul id="pairs"></ul>
      <button id="reset">Novo Sorteio</button>
    </div>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

CSS

```
body {
  font-family: Arial, sans-serif;
  background-color: #f0f8ff;
  margin: 0;
  padding: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.container {
  text-align: center;
  background: #ffffff;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
  width: 90%;
  max-width: 400px;
}

h1 {
  color: #333;
}

input {
  padding: 10px;
  margin-bottom: 10px;
  width: 70%;
  border: 1px solid #ddd;
  border-radius: 5px;
}

button {
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  background-color: #007bff;
  color: #fff;
  cursor: pointer;
}

button:hover {
  background-color: #0056b3;
}

.draw-btn {
  margin-top: 10px;
}

ul {
  list-style: none;
  padding: 0;
  margin: 10px 0;
}

.hidden {
  display: none;
}
```

Javascript

```
document.addEventListener('DOMContentLoaded', () => {
  const participants = [];
  const form = document.getElementById('form');
  const nameInput = document.getElementById('name');
  const addButton = document.getElementById('add');
  const participantsList = document.getElementById('participants');
  const drawButton = document.getElementById('draw');
  const resultsDiv = document.getElementById('results');
  const pairsList = document.getElementById('pairs');
  const resetButton = document.getElementById('reset');

  // Adicionar participante
  addButton.addEventListener('click', () => {
    const name = nameInput.value.trim();
    if (name && !participants.includes(name)) {
      participants.push(name);
      const li = document.createElement('li');
      li.textContent = name;
      participantsList.appendChild(li);
      nameInput.value = "";
    } else {
      alert("Nome inválido ou já adicionado!");
    }
  });

  // Sortear pares
  drawButton.addEventListener('click', async () => {
    if (participants.length < 2) {
      alert("É necessário pelo menos dois participantes para o sorteio.");
      return;
    }
    const response = await fetch('/draw', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify({ participants })
    });
    const data = await response.json();
    if (data.error) {
      alert(data.error);
      return;
    }
    pairsList.innerHTML = "";
    data.pairs.forEach(pair => {
      const li = document.createElement('li');
      li.textContent = `${pair.giver} → ${pair.receiver}`;
      pairsList.appendChild(li);
    });
    resultsDiv.classList.remove('hidden');
  });

  // Resetar sorteio
  resetButton.addEventListener('click', () => {
    resultsDiv.classList.add('hidden');
    pairsList.innerHTML = "";
  });
});
```

Node.js

```
const express = require('express');
const app = express();
const PORT = 3000;

app.use(express.json());
app.use(express.static('public'));

// Rota para sortear os pares
app.post('/draw', (req, res) => {
  const { participants } = req.body;

  if (!participants || participants.length < 2) {
    return res.json({ error: 'Número insuficiente de participantes.' });
  }

  const shuffled = [...participants].sort(() => Math.random() - 0.5);
  const pairs = shuffled.map((giver, index) => ({
    giver,
    receiver: shuffled[(index + 1) % shuffled.length],
  }));

  res.json({ pairs });
});

// Iniciar servidor
app.listen(PORT, () => {
  console.log(`Servidor rodando em http://localhost:${PORT}`);
});
```

Padrão de pastas

/simulador-amigo-oculto

| -- /public

| | -- index.html

| | -- styles.css

| | -- script.js

| -- server.js

| -- package.json