**Análise Geral dos Atores e da Estrutura**

* **Perspectiva do Usuário:**
  + O sistema está claramente dividido para três tipos de pessoas: o Administrador (Adm), que gerencia o sistema; o "dono" do churrasco (Usuário - Criador), que organiza tudo; e o Usuário - Participante, que é um convidado. Essa separação de papéis é lógica e fácil de entender. Fica claro quem é responsável por cada grande tarefa.
* **Perspectiva do Projetista:**
  + A estrutura de atores (Adm, Criador, Participante) define a necessidade de um sistema de Controle de Acesso Baseado em Papéis (RBAC). A generalização entre Usuário - Criador e Usuário - Participante indica que ambos são tipos de "Usuário", compartilhando funcionalidades como Visualizar Meus Churrascos, mas com permissões diferentes. **Observação:** A seta de generalização do Participante para o Criador está tecnicamente invertida; o correto seria ambos apontarem para um ator "Usuário" mais genérico, mas a intenção é compreensível.
  + O caso de uso Autenticar Usuário é central. A relação <<include>> partindo de quase todas as outras funcionalidades em direção a ele significa que a autenticação é um pré-requisito obrigatório para a maior parte do sistema. Isso aponta para a necessidade de um *middleware* de autenticação ou um *gateway* que proteja as rotas da aplicação.
* **Perspectiva do Testador:**
  + O foco principal dos testes será em cima das permissões de cada ator. É necessário criar uma matriz de testes para garantir que:
    - Um Participante não consiga acessar as funcionalidades de um Criador (ex: Criar Churrasco).
    - Um Criador não consiga acessar as funcionalidades de um Adm (ex: Gerenciar Usuários).
    - Um usuário não autenticado seja sempre redirecionado para a tela de login ao tentar acessar qualquer funcionalidade protegida.

**Análise dos Casos de Uso do Administrador (Adm)**

* **Casos de Uso:** Gerenciar Usuários, Definir Regras Globais de Consumo.
* **Perspectiva do Usuário (Administrador):**
  + Meu papel é puramente administrativo. Eu configuro as regras que todos os churrascos vão usar (como a quantidade de comida por pessoa) e controlo quem pode acessar o sistema. Minhas tarefas são separadas da organização de um evento específico.
* **Perspectiva do Projetista:**
  + Essas funcionalidades devem ser agrupadas em um "Painel de Administração" separado. Definir Regras Globais irá popular uma tabela de configurações que será consultada por outras funcionalidades (como Montar Carrinho). Gerenciar Usuários implica em operações CRUD (Criar, Ler, Atualizar, Deletar) na tabela de usuários. Ambas as funcionalidades devem ter uma verificação rigorosa para garantir que apenas um usuário com o papel Adm possa acessá-las.
* **Perspectiva do Testador:**
  + **Cenários de Sucesso:** Validar que um administrador consegue criar, editar e deletar usuários, e que consegue salvar as regras globais de consumo.
  + **Cenários de Permissão:** Tentar acessar as URLs de gerenciamento de usuários e regras com uma conta de Usuário - Criador e verificar se o acesso é negado.
  + **Validação de Impacto:** Após alterar uma regra global, criar um novo churrasco e validar se a nova regra está sendo aplicada corretamente no cálculo do carrinho.

**Análise dos Casos de Uso do Criador do Churrasco**

* **Casos de Uso:** Criar Churrasco, Editar Churrasco, Convidar Participantes, Montar Carrinho, Acompanhar Pagamentos, Registrar Compras, Consolidar contas, Realizar Check-in.
* **Perspectiva do Usuário (Criador):**
  + Eu tenho o controle total do meu evento, desde sua criação até a prestação de contas. O fluxo parece lógico: eu crio o evento, convido as pessoas, monto o carrinho de compras, registro os gastos e, no final, consolido as contas. A possibilidade de Anexar Comprovantes quando registro uma compra é uma opção útil para transparência.
* **Perspectiva do Projetista:**
  + Este é o fluxo principal do sistema. Há uma forte coesão entre esses casos de uso, que devem ser apresentados em um painel de gerenciamento do evento.
  + A relação <<extend>> de Registrar Compras para Anexar Comprovantes indica que a interface de registro de compra deve ter um ponto de extensão (um botão "Anexar Comprovante", por exemplo) que é opcional.
  + A relação <<extend>> (no diagrama está com a seta de <<include>>, mas o label é <<extend>>) de Criar Churrasco para Convidar Participantes sugere que o convite de participantes é uma ação opcional que pode ser feita a partir da criação do churrasco, mas também posteriormente.
  + A integridade dos dados é crucial, pois a Consolidação de Contas depende diretamente dos dados de Acompanhar Pagamentos e Registrar Compras.
* **Perspectiva do Testador:**
  + É preciso testar o ciclo de vida completo de um evento. Um "teste de ponta a ponta" (end-to-end) deve ser criado, simulando um criador que:
    1. Autentica-se.
    2. Cria um churrasco.
    3. Convida participantes.
    4. Monta o carrinho.
    5. Registra uma compra e, no mesmo fluxo, anexa um comprovante (testando o <<extend>>).
    6. Registra outra compra sem comprovante.
    7. Acompanha e confirma os pagamentos.
    8. Realiza o check-in de um participante pago.
    9. Consolida as contas e verifica o balanço.
    10. Gera um relatório final.

**Análise dos Casos de Uso do Participante**

* **Casos de Uso:** Visualizar Convite, Visualizar Prestação.
* **Perspectiva do Usuário (Participante):**
  + Minha interação com o sistema é mais passiva. Eu preciso ver os detalhes do convite para saber do que se trata e como pagar. Depois do evento, tenho interesse em ver a prestação de contas para saber como o dinheiro foi usado.
* **Perspectiva do Projetista:**
  + As funcionalidades do participante são primariamente de leitura (Visualizar). O sistema deve apresentar visões simplificadas dos dados inseridos pelo criador. A tela de Visualizar Prestação só deve ser acessível após o criador ter executado o caso de uso Consolidar contas. Isso requer uma verificação de estado no evento.
* **Perspectiva do Testador:**
  + **Cenários de Sucesso:** Validar que um participante convidado consegue ver os detalhes do convite e, após o evento ser consolidado, consegue ver a prestação de contas.
  + **Cenários de Permissão:** Testar se um participante consegue acessar links ou botões de edição ou gerenciamento, o que não deve ser permitido.
  + **Teste de Estado:** Tentar acessar a prestação de contas de um evento que ainda não foi encerrado pelo criador e validar que o acesso é bloqueado ou uma mensagem apropriada é exibida.

**Análise dos Casos de Uso Compartilhados**

* **Caso de Uso:** Visualizar Meus Churrascos, Gerar Relatórios.
* **Perspectiva do Usuário:**
  + Tanto como criador quanto como participante, eu quero poder ver a lista de churrascos dos quais faço parte. Também é útil poder gerar um relatório com os dados do evento, seja para organizar (como criador) ou para conferir (como participante).
* **Perspectiva do Projetista:**
  + Visualizar Meus Churrascos exige uma consulta que una eventos criados pelo usuário e eventos para os quais ele foi convidado.
  + Gerar Relatórios pode ser a mesma funcionalidade, mas que exibe informações diferentes dependendo do papel do usuário. Por exemplo, um criador pode ver dados detalhados dos participantes, enquanto um participante só vê o resumo financeiro. Isso simplifica o código, controlando o conteúdo do relatório com base no papel do usuário que o solicitou.
* **Perspectiva do Testador:**
  + **Cenário 1 (Criador):** Logar como criador, acessar Visualizar Meus Churrascos e verificar se o evento criado está listado. Em seguida, gerar um relatório e validar se ele contém todos os dados de gerenciamento.
  + **Cenário 2 (Participante):** Logar como participante, acessar Visualizar Meus Churrascos e verificar se o evento para o qual foi convidado está listado. Em seguida, gerar um relatório e validar que ele contém apenas os dados públicos da prestação de contas.