

Universidade Federal de Ouro Preto



Engenharia de Software II

Sistema de *Pedidos de Restaurante*

Grupo: *As Chinchilas do Deserto*

Alunos:	Ana Luiza Almeida Soares Bernardo Brandão Freguglia Bruno José Baêta Barbosa Everton de Souza Kenedy Hugo Eduardo Ziviani Suleimane Ducure Ulisses Viana Taveira Wendel Francisco Furtado e Lima
Professor:	Msc prof. Johnatan Oliveira
Horário:	Seg & Qua - 08:20 -10:00

Ouro Preto, 06 de Janeiro de 2021

Conteúdo

1	Histórico de Revisões	1
2	Processo e Software	1
3	Cronograma	2
4	Levantamento de Requisitos	2
5	Especificação de Requisitos	4
5.1	Requisitos Funcionais	4
5.2	Requisitos Não Funcionais	6
6	Plano de VVT	6
6.1	Requisitos a serem testados	7
6.2	Estratégias e ferramentas de teste	7
6.3	Equipe e infra-estrutura	7
6.4	Execução do Plano de Teste	7
7	Medição e Qualidade de Software	7
8	Observações	9
9	Referências	10

1 Histórico de Revisões

Data	Versão	Descrição	Autor
08/03/2021	0.0	Início da escrita do documento	Ana Luiza
08/03/2021	0.0	Justificativa do Processo Utilizado	Ulisses
08/03/2021	0.0	Escrita do Cronograma	Bernardo e Bruno
14/03/2021	0.0	Diagrama de Casos de Uso	Ana Luiza e Suleimane
20/03/2021	0.0	Levantamento de requisitos	Ana Luiza
27/03/2021	0.0	Plano de verificação, validação e teste de software	Ulisses e Ana Luiza
01/04/2021	0.0	Descrição de técnicas e ferramentas para a medição	Everton
04/04/2021	0.0	Entrega dos videos	Todos
05/04/2021	0.0	Edição dos videos e post Youtube	Bernardo

Tabela 1: Revisões do Documento

2 Processo e Software

O grupo as Chinchilas do Deserto escolheu como processo de software para o projeto de desenvolvimento do software de pedidos de restaurante o Modelo Incremental.

Sendo uma melhoria do Modelo Cascata, o Modelo Incremental trabalha com incrementos, ou seja, pequenos pedaços de software entregues de cada vez. Este modelo combina elementos do Modelo em Cascata aplicados de maneira iterativa, ou seja, de forma que o progresso aconteça através de sucessivos refinamentos, melhorados a cada iteração.

Cada incremento é desenvolvido de forma linear e exposto aos comentários dos clientes, sendo necessária alguma alteração na implementação é feito um novo incremento e mais uma vez ele é apresentado aos clientes. A cada aprimoramento é lançada uma nova versão, e todas as atividades de especificação, projeto, implementação e validação são intercalados, acontecendo em cada nova versão.

Esse feedback rápido e a implementação rápida de um software útil foram as principais características que fizeram o grupo optar por esse modelo. Além disso, a natureza da ferramenta desenvolvida, um Software de Pedidos, exige uma troca rápida de versões e um feedback rápido das versões em desenvolvimento pode ser feito com facilidade.

3 Cronograma

Segue na tabela a baixo um cronograma inicial das atividades que serão realizadas no projeto e seus respectivos responsáveis.

Nome	Tarefa	Prazo
Suleimane	Diagrama de Casos de Uso	08/03 a 13/03
Suleimane e Ana Luiza	Descrição dos Cenários	08/03 a 13/03
Suleimane	Justificativa do Processo	08/03 a 10/03
Ana Luiza	Levantamento dos Requisitos	15/03 a 20/03
Everton	Descrição da Técnicas de Medição	22/03 a 01/04
Hugo	Implementação parcial do Software	15/03 a 05/04
Bruno	Edição e atualização do Trello	15/03 a 12/04
Bernardo	Edição do vídeo	04/04 a 05/04
Bernardo	Criação dos Slides	05/04 a 12/04
Todos os participantes	Primeira apresentação	01/04 a 05/04
Todos os participantes	Segunda apresentação	05/04 a 08/04
Todos os participantes	Última apresentação	08/04 a 12/04

Tabela 2: Cronograma

4 Levantamento de Requisitos

Independente da metodologia de desenvolvimento utilizada, o levantamento de requisitos é o ponto de partida de qualquer projeto de software, pois é a partir dos resultados obtidos durante esta etapa que será possível definir como as próximas etapas do desenvolvimento serão executadas.

Para o levantamento de requisitos do seguinte projeto, foi utilizado o levantamento orientado a pontos de vista. O método VORD (viewpoint-oriented requirements definition – definição de requisitos orientada a ponto de vista) consiste em definir os diferentes usuários do sistema como pontos de vista e descrever as ações que esses usuários irão realizar. A técnica é mais adequada devida às condições em que o trabalho está inserido e ao formato em que vai ser desenvolvido, pois é um processo que pode ser feito sem o contato com o cliente, já que não existe um cliente real. Além disso, a técnica permite a visualização completa das funções do sistema e das relações entre usuários, e pode prever possíveis conflitos no sistema. Etapas:

1. Identificação do Pontos de vista (Todos os pontos de vista são de funcionários do restaurante):

- Garçom
 - Copa
 - Cozinheiro
2. Estruturação e documentação dos Pontos de Vista: O ponto de vista do Garçom é hierarquicamente superior aos outros pontos de vista, pois o Garçom executa as ações principais. Os pontos de vista de Copa e Cozinheiro vem em seguida pois dependem da interação do Garçom para qualquer atividade realizada. A seguir as tarefas a serem realizadas pelos pontos de vista:
- Garçom: Selecionar pedidos, enviar pedidos, confirmar a disponibilidade dos pedidos;
 - Copa: Confirmar a disponibilidade das bebidas e notificar a prontidão dos pedidos;
 - Cozinheiro: Confirmar a disponibilidade das bebidas e notificar a prontidão dos pedidos.

Para demonstrar as ações dos nossos Pontos de Vista, utilizamos o Diagrama dos Casos de Uso colocando-os como autores.

O Diagrama dos casos de uso é uma importante ferramenta no processo de desenvolvimento de software pois esquematiza os papéis de cada usuário e as funções do software. Como o software a ser desenvolvido é bem simples, apresentamos abaixo o diagrama referente a ele. Tendo como atores os funcionarios, mais específico, o Garçom responsavel por selecionar os pedidos feitos pelos clientes e enviar para a cozinha e copa, o Cozinheiro/a que a partir dos pedidos eviados pelo Garçom, notifica a disponibilidade dos pedidos recebidos e após ter os pedidos prontos notifica o Garçom, e por último a Copa, que faz o mesmo processo do cozinheiro. Ações do Diagrama de Casos de Uso e seus respectivos atores:

- Selecionar os Pedidos: O Garçom seleciona os produtos pedidos pelos clientes;
- Enviar Pedido: O Garçom envia tanto para Cozinha como para Copa os pedidos;
- Confirmar Disponibilidade: A cozinha e a Copa confirmam a disponibilidade dos pedidos que foram feitos pelo Garçom;
- Confirmar a Prontidão: A cozinha e a copa notificam o Garçon a prontidão do pedidos.

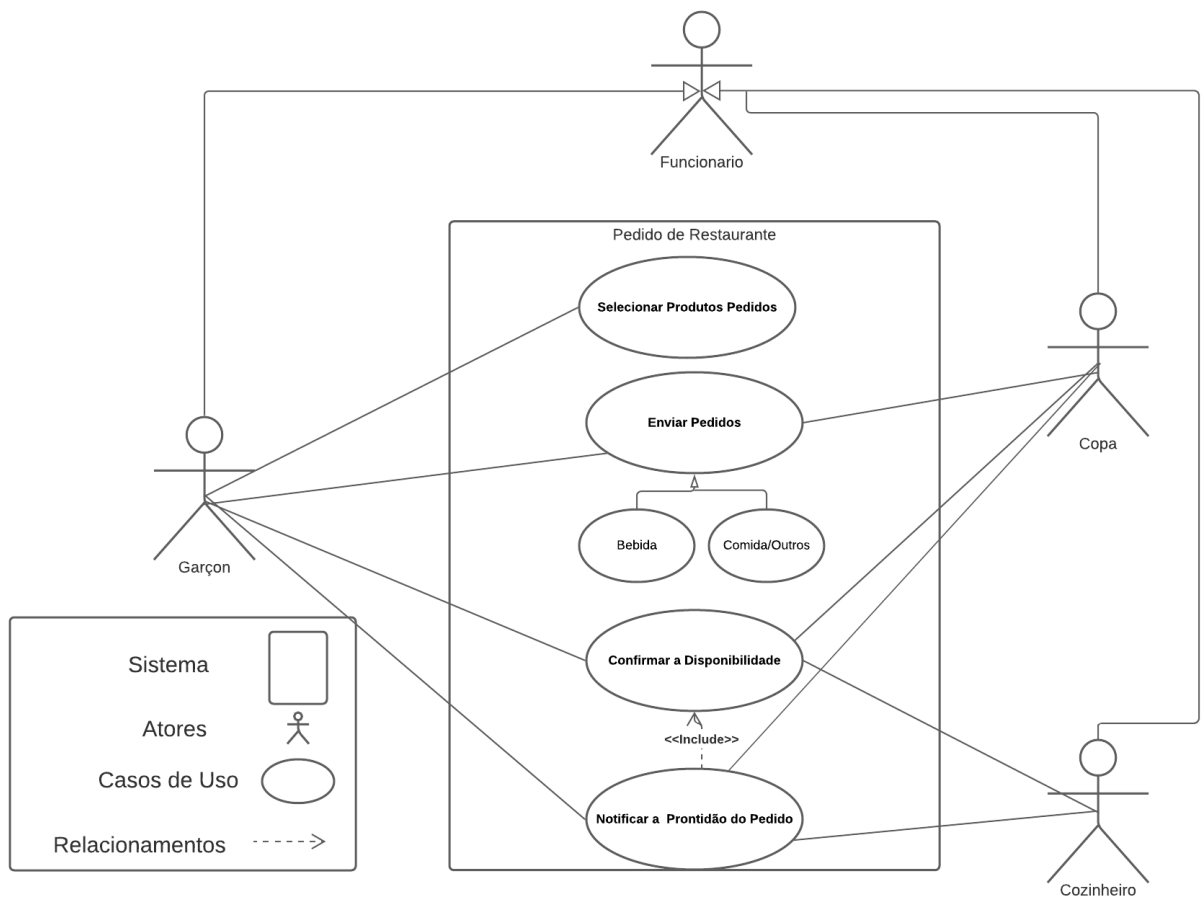


Figura 1: Diagrama de Casos

5 Especificação de Requisitos

5.1 Requisitos Funcionais

Requisitos Funcionais são assuntos de importância fundamental ou essencial ao produto. Eles descrevem o que o produto tem de fazer ou que ações processuais deve tomar.

RF01–Tela HOME O software deve mostrar todos os pedidos já realizados. **Informações:** ID (identificação do pedido), Nome do Cliente, Prato e quantidade, Bebida e quantidade, Mesa, checkbox para o prato e checkbox pra a bebida. **Regras:** O sistema deve permitir que o usuário visualize todas as informações contidas a tela home. **Informações:** ID do pedido, nome do cliente referente a aquele pedido, prato escolhido e quantidade do prato, be-

bebida escolhida e quantidade da bebida, mesa em que o cliente está e checkbox do prato e da bebida.

- Os campos Nome do cliente, Prato e Bebida devem ser palavras;
- Os campos de ID, Quantidade e Mesa devem ser números inteiros;
- Os campos checkbox devem mostrar se o prato ou a bebida do cliente está pronto

RF02–Tela GARÇOM O software deve permitir que o Garçom registre os itens do pedido de um determinado cliente. **Informações:** Nome do cliente, Prato e quantidade, bebida e quantidade e a mesa. **Regras:** O sistema deve permitir que o Garçom, antes de clicar no botão enviar, adicione a informações do pedido nos campos correspondentes. Os campos são: Nome do cliente, Prato Escolhido, Quantidade do prato escolhido, Bebida Escolhida, Quantidade da Bebida Escolhida e Mesa em que o cliente está.

- Os campos Nome do cliente, Prato e Bebida devem ser palavras;
- Os campos de Quantidade e Mesa devem ser números inteiros;
- O usuário precisa clicar no botão enviar para que o pedido seja encaminhado para as outras telas;
- O usuário pode modificar o pedido antes de clicar no botão enviar.

RF03–Tela COZINHA O software deve permitir que o Cozinheiro visualize os pedidos pendentes, bem como as informações do pedido. **Informações:** Prato, Quantidade, Mesa, Cliente e checkbox do prato. **Regras:** O sistema deve permitir que o Cozinheiro visualize todos os pedidos e seus campos. Também deve permitir que selecione o check box do pedido escolhido. Os campos são: Nome do cliente, Prato Escolhido, Quantidade do prato escolhido, Mesa em que o cliente está e Check box do pedido.

- Os campos Nome do cliente e Prato devem ser palavras;
- Os campos Quantidade e Mesa devem ser números inteiros;
- O usuário deve clicar no check box pra certificar que o pedido foi realizado.

RF04–Tela COPA O software deve permitir que o responsável pela Copa visualize os pedidos pendentes, bem como as informações do pedido. **Informações:** Bebidas, Quantidade, Mesa, Cliente e checkbox da bebida.

Regras: O sistema deve permitir que o responsável pela Copa visualize todos os pedidos e seus campos. Também deve permitir que selecione o check box do pedido escolhido. Os campos são: Nome do cliente, Bebida Escolhida, Quantidade da bebida escolhida, Mesa em que o cliente está e Check box do pedido.

- Os campos Nome do cliente e Bebida devem ser palavras;
- Os campos Quantidade e Mesa devem ser números inteiros;
- O usuário deve clicar no check box pra certificar que o pedido foi realizado.

5.2 Requisitos Não Funcionais

Requisitos Não Funcionais são as propriedades que as funções devem ter, tais como desempenho e usabilidade. Não se detenha ao seu nome pouco apropriado (nós o usamos porque é a maneira mais comum de se referir a estes tipos de requisitos) estes requisitos são tão importantes quanto as exigências funcionais, para o sucesso do produto.

RNF01 O sistema deve registrar corretamente os pedidos enviados pela Tela Garçom na API. **Informações:** Nome do cliente, Prato e quantidade, bebida e quantidade e a mesa.

RNF02 O sistema deve direcionar corretamente os pedidos enviados pela Tela Garçom para a Tela Cozinha. **Informações:** Nome do cliente, Prato e quantidade, bebida e a mesa.

RNF03 O sistema deve direcionar corretamente os pedidos enviados pela Tela Garçom para a Tela Copa. **Informações:** Nome do cliente, bebida e quantidade e a mesa.

RNF04 O sistema deve exibir corretamente todos os pedidos enviados pela Tela Garçom na Tela Home. Bem como a situação do pedido, ou seja, se já foi preparado ou não. **Informações:** Nome do cliente, Prato e quantidade, bebida e quantidade, a mesa e situação do pedido.

6 Plano de VVT

Asseguram que o software cumpra com suas especificações e atenda às necessidades dos usuários. Você deve apresentar um plano de testes, ferramentas que serão utilizadas e coisas do tipo.

Veja um exemplo no link: https://www.cin.ufpe.br/~gta/rup-vc/extend.formal_resources/guidances/examples/resources/test_plan_v1.htm

6.1 Requisitos a serem testados

Deverão ser testadas as telas nomeadas como Cozinha, Copa, Garçom e Home. Na tela denominada Garçom deve-se testar se os itens adicionados pelo usuário estão sendo contabilizados e se os botões para alteração das quantidades estão funcionando. Na tela Cozinha deverá ser testado a informação recebida da tela Garçom, se essa informação está completa e correta. Na tela Copa deve-se testar a adição e modificação de informações e se será possível a visualização da mesma. Além disso, deve-se ser testado também a integração entre as partes. Na tela Home será testado a integração das partes e se as informações condizem com as outras telas.

6.2 Estratégias e ferramentas de teste

Ferramentas utilizadas:

- Biblioteca react-testing-library
- Visual Studio Code

Como o aplicativo foi desenvolvido utilizando o framework React, os testes serão feitos utilizando a biblioteca react-testing-library, que simula usuários reais na aplicação. Os testes podem ser feitos manualmente ou automaticamente a partir da renderização da interface dos componentes na página. Optamos por utilizar os testes automáticos. É necessário a instalação dessa biblioteca no projeto.

6.3 Equipe e infra-estrutura

Contém descrição da equipe e da infra-estrutura utilizada para o desenvolvimento das atividades de testes, incluindo: pessoal, equipamentos, software de apoio, materiais, dentre outros. Isto visa garantir uma estrutura adequada para a execução das atividades de testes previstas no plano.

6.4 Execução do Plano de Teste

7 Medição e Qualidade de Software

Utilizaremos a ferramenta React Profiler, que é um plugin no React 16.5 que utiliza a Profiler API (ainda em experimentos). A ideia dele é basicamente coletar informações de cronometragem dos componentes. Não é nada mágico o seu funcionamento, consiste basicamente em gravações e leituras.

Funcionalidade: Tela Garçon		Tempo Despendido: 1h
Contador: 01		Críticidade: Média
Objeto de Teste: Validar os itens adicionados pelo usuário Verificar se os itens estão sendo contabilizados da maneira correta Verificar o botão alterar quantidade		
Descrição do Caso de Teste: A tela deverá executar corretamente um determinado pedido, recebendo os parâmetros de entrada e adicionar e excluir itens de acordo com a função dos botões.		
Pré-Condição: O usuário deverá inserir um conjunto de itens, excluir alguns itens e alterar a quantidade.		
ID	Passo	Procedimento
1	P1	Executar o aplicativo e selecionar adicionar itens
1	P2	Alterar a quantidade de itens utilizando o botão
1	P3	Excluir itens
2	V1	Verificar se as operações acima foram executadas corretamente
Resultado Esperado: As operações deverão funcionar corretamente cumprindo todas as regras acima		

Figura 2: Teste 1

Inicialmente executamos o Profiler e em seguida abrimos nossa aplicação, em sequencia interagimos normalmente com nossa aplicação para que o Profile consiga coletar informações de performance, depois de algumas interações pode-se terminar a gravação com o Profiler. O react trabalha em duas fases: renderização e commit. Na fase de renderização o React determina o que foi mudado. Durante essa fase, o React chama a função render dos componentes e compara o novo resultado com a renderização anterior. Já na fase de commit, é quando de fato o React aplica essas mudanças. Essa mudança pode ser uma inserção, atualização ou remoção de nós no DOM (no caso do React DOM). Porém o Profiler trabalha na fase de commit, logo é necessário pelo menos uma renderização para que a coleta de informações seja realizada.

Após executado temos algumas informações coletadas como:

- **Ranked:** mostra os componentes que mais demoraram para ser renderizados no topo da lista; Figura 7.1.
- **Component:** mostra quantas vezes um determinado componente renderizou durante a gravação; Figura 7.2.
- **Flame chart:** mostra o estado da aplicação para um determinado commit(quando foi commitado, tempo de renderização e as interações feitas); Figura 7.3.

Funcionalidade: Tela Cozinha		Tempo Despendido: 2h
Contador: 02		Criticidade: Baixa

Objeto de Teste: Conferir se as informações da tela Garçom estão chegando na tela Cozinha de forma correta e completa

Descrição do Caso de Teste: Essa tela depende que outra tela envie informações para ela, e o teste visa conferir se essas informações estão corretas.

Pré-Condição: O usuário deverá executar a tela Garçom e executar suas funcionalidades e fazer o envio para a tela Cozinha.
--

ID	Passo	Procedimento
1	P1	Executar o aplicativo e selecionar adicionar itens na tela Garçom
1	V1	Enviar as informações para a Cozinha
1	P2	Abrir a tela cozinha
2	V1	Verificar se as informações enviadas pela tela Garçom chegaram corretamente

Resultado Esperado: As operações deverão funcionar corretamente cumprindo todas as regras acima
--

Figura 3: Teste 2

- Interactions: ainda está em testes essas informações coletadas, não sabemos dizer muito sobre ela; Figura 7.4.

8 Observações

Apresente aqui as dificuldades na disciplina, trabalho prático e coisas do tipo.

Funcionalidade: Tela Copa		Tempo Despendido: 1h
Contador: 03		Criticidade: Baixa

Objeto de Teste: Conferir a adição e modificação de informações e visualização.

Descrição do Caso de Teste: Essa tela depende que outra tela envie informações para ela, e o teste visa conferir se essas informações estão corretas.

Pré-Condição: É preciso que as telas de Garçom e Cozinha sejam executadas para conferir a alteração das informações na tela Copa.

ID	Passo	Procedimento
1	P1	Executar o aplicativo e selecionar adicionar itens na tela Garçom
1	P2	Enviar as informações para a Cozinha
1	P3	Fazer alterações e adicionar informações na tela Copa
2	V1	Verificar se as informações na tela Copa estão corretas e visualizar

Resultado Esperado: As operações deverão funcionar corretamente cumprindo todas as regras acima

Figura 4: Teste 3

9 Referências

- [1] Chapman, S.J. – Electric Machinery Fundamentals, 4th Edition;
- [2] Fitzgerald, A. E. – Máquinas Elétricas, 2da Edição;
- [3] <https://www.devmedia.com.br/tecnicas-para-levantamento-de-requisitos/9151>

Funcionalidade: Tela Home		Tempo Despendido: 2h
Contador: 04		Críticidade: Média
Objeto de Teste: Conferir a integração de toda a aplicação.		
Descrição do Caso de Teste: A tela Home é a tela central da aplicação e integra todas as partes.		
Pré-Condição: É preciso que todas as outras telas estejam integradas.		
ID	Passo	Procedimento
1	P1	Executar o aplicativo
1	P2	Navegar pela aplicação utilizando o menu da home
1	P3	Fazer os acessos via Home
2	V1	Verificar o acesso das outras telas a partir da Home
Resultado Esperado: As operações deverão funcionar corretamente cumprindo todas as regras acima		

Figura 5: Teste 4

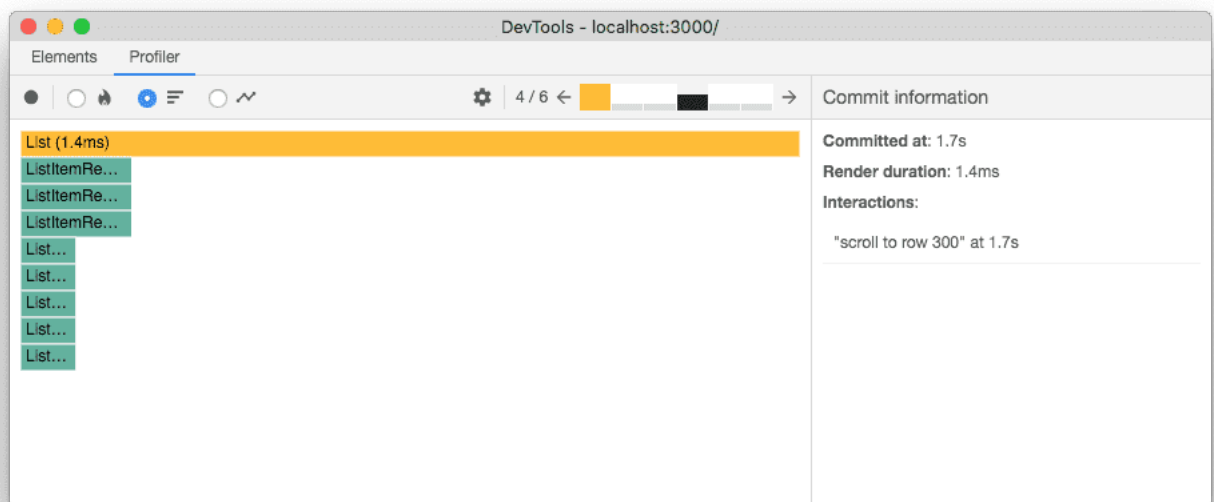


Figura 6: Ranked chart

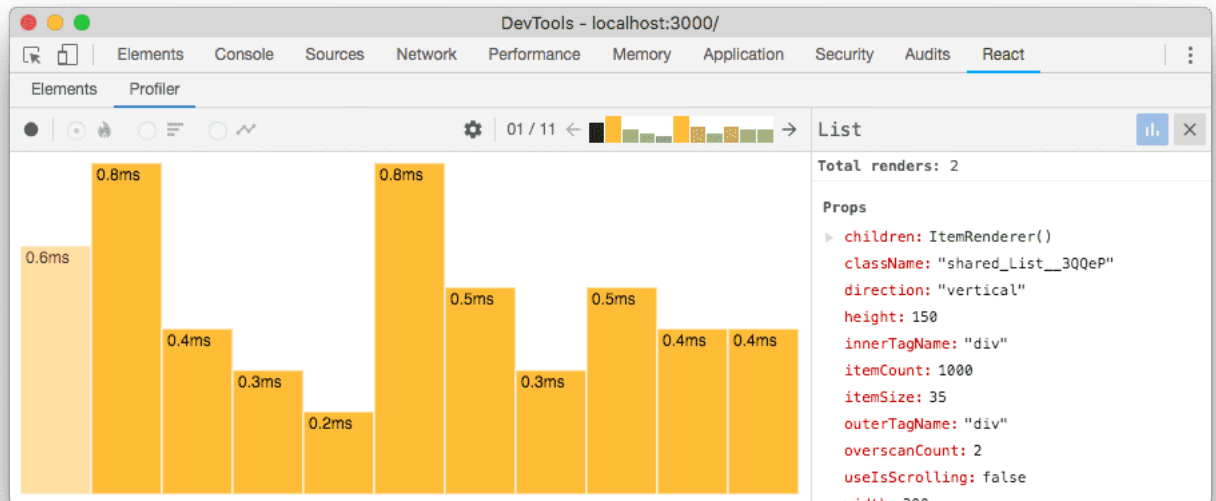


Figura 7: Component chart

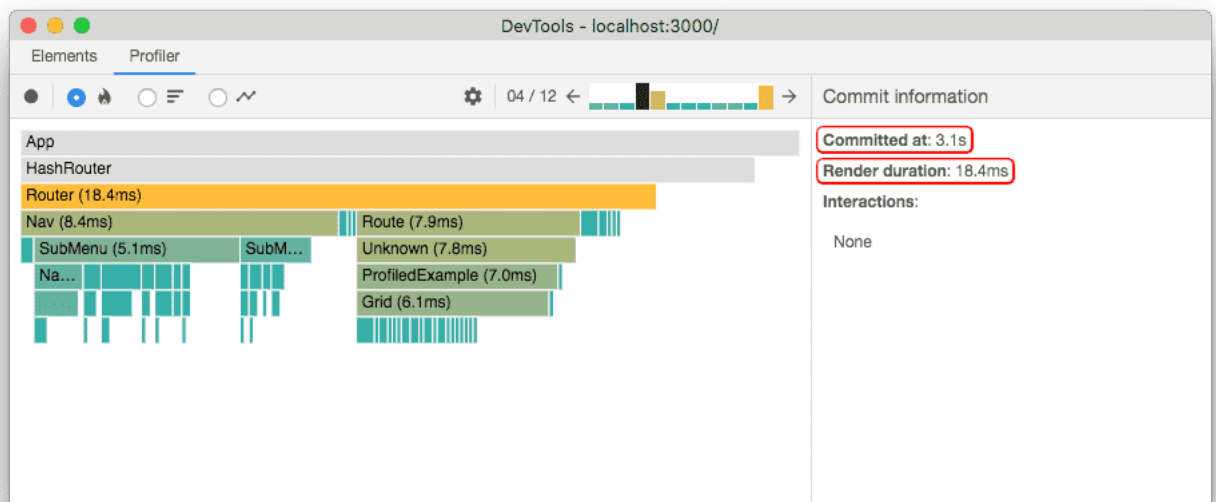


Figura 8: Flame chart

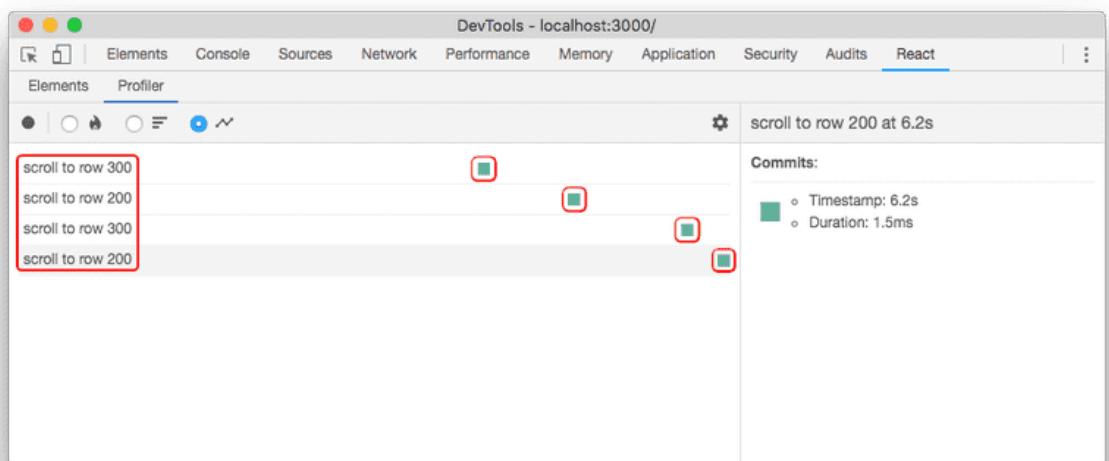


Figura 9: Interactions