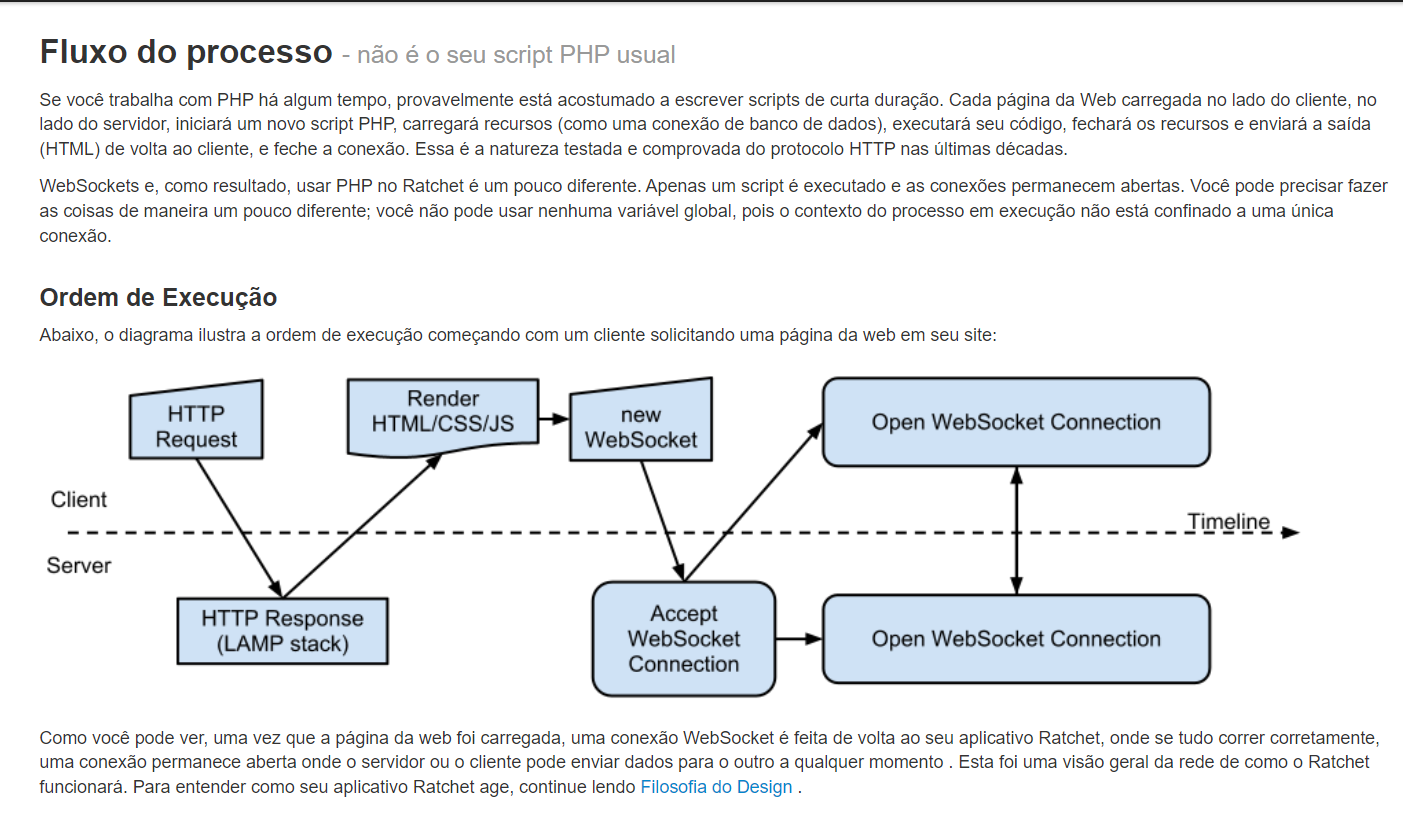
**. Anotações sobre o Websockerts para php**

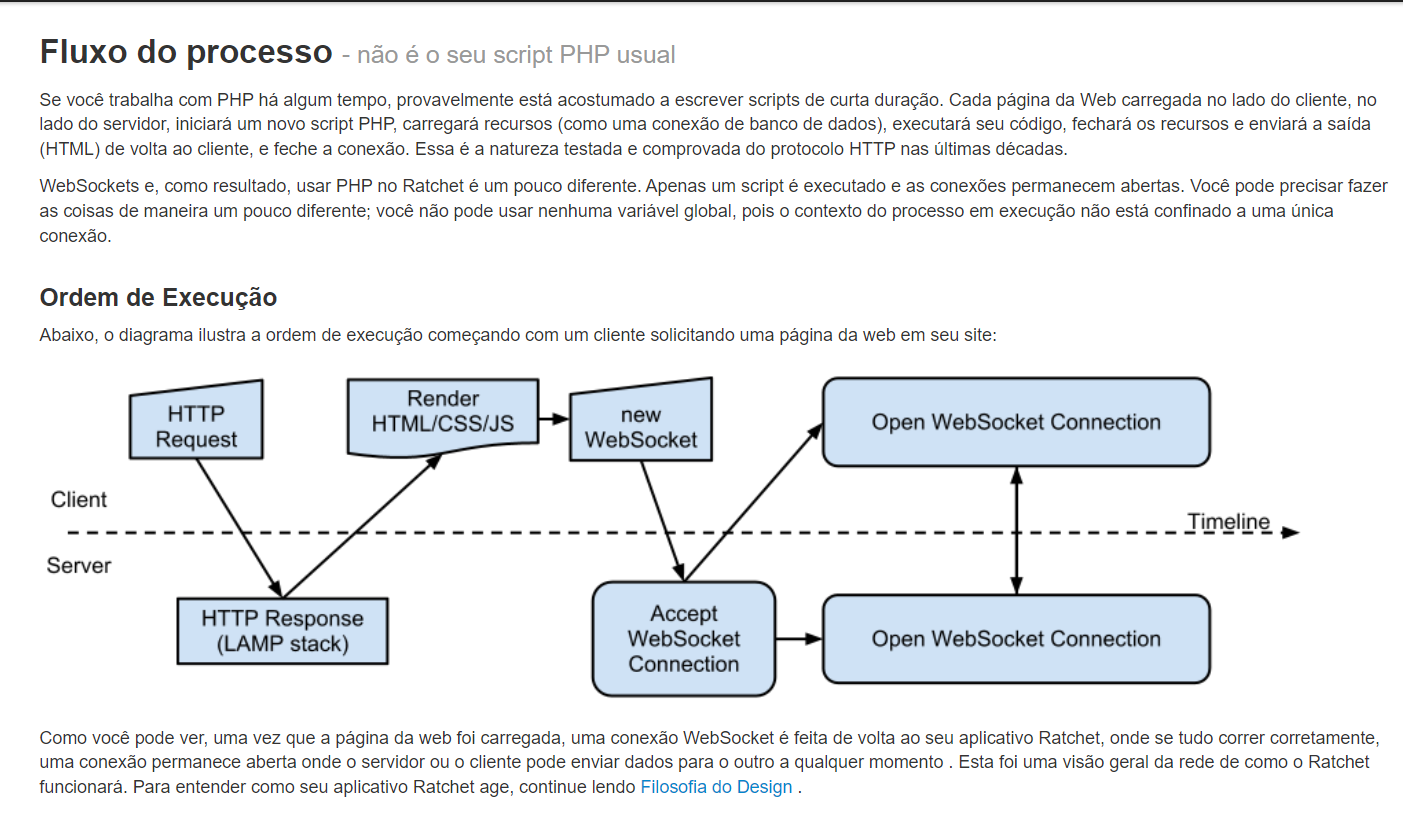
1. Explicação sobre seu funcionamento

. **Ratchet** é uma biblioteca PHP fracamente acoplada que fornece aos desenvolvedores ferramentas para criar aplicativos bidirecionais em tempo real entre clientes e servidores por meio de **WebSockets**. Esta não é a Internet do seu avô.

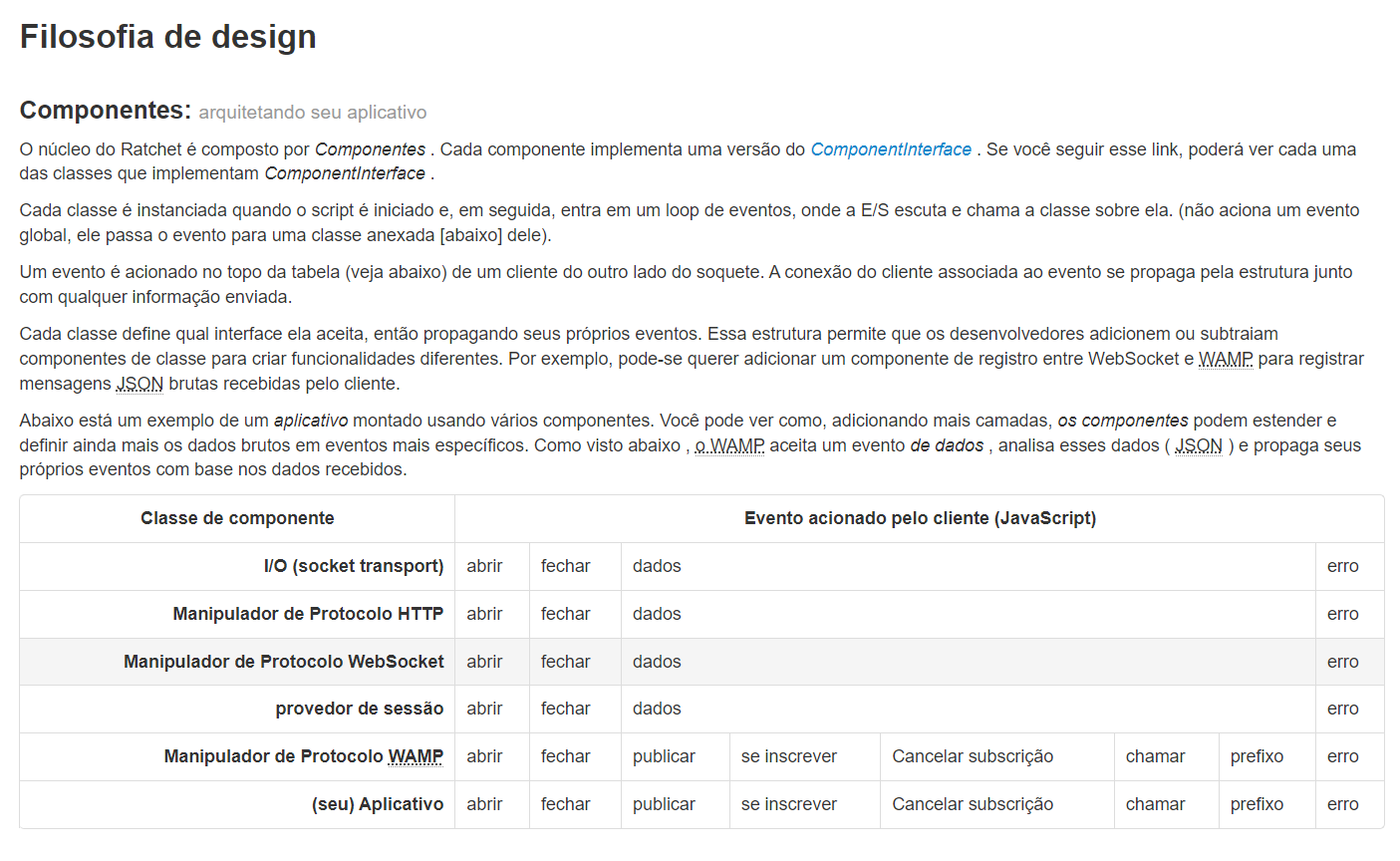
**Diagrama de funcionamento**



**Diagrama como funciona na internet**



**Estrutura de design e funcionamento**

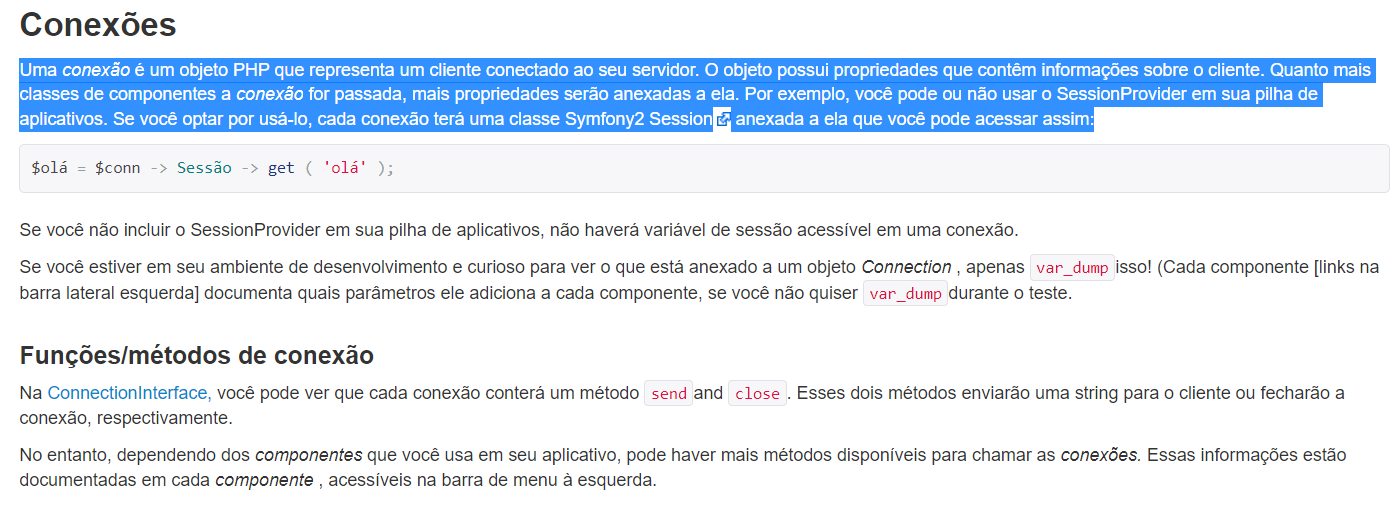


**Conexões**: objetos passados ​​por seus componentes de aplicativo

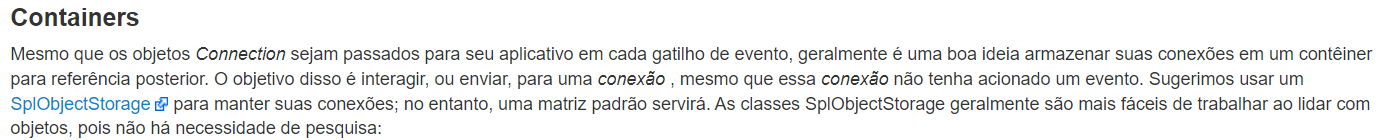
Com sua arquitetura de aplicativo configurada e instanciada, os recursos são passados ​​por seu aplicativo. Quando os eventos são acionados pela base ( IoServer ) e propagados até seu aplicativo, um objeto Connection é enviado representando o cliente na outra ponta do fio. Esses objetos Connection são usados ​​para interagir com o cliente e são passados ​​por meio de seu aplicativo.

**Conexões**

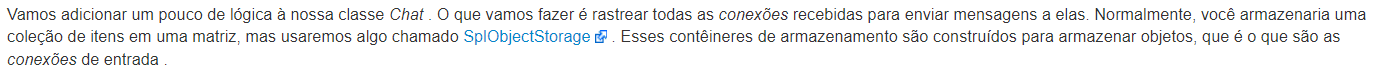
. Uma conexão é um objeto PHP que representa um cliente conectado ao seu servidor. O objeto possui propriedades que contêm informações sobre o cliente. Quanto mais classes de componentes a conexão for passada, mais propriedades serão anexadas a ela. Por exemplo, você pode ou não usar o [SessionProvider](http://socketo.me/docs/sessions) em sua pilha de aplicativos. Se você optar por usá-lo, cada conexão terá uma classe [Symfony2 Session](http://symfony.com/doc/master/components/http_foundation/sessions.html) anexada a ela que você pode acessar assim:



**Armazenado a conexão da porta soquete do cliente**



**Construção parte logica (Container)**



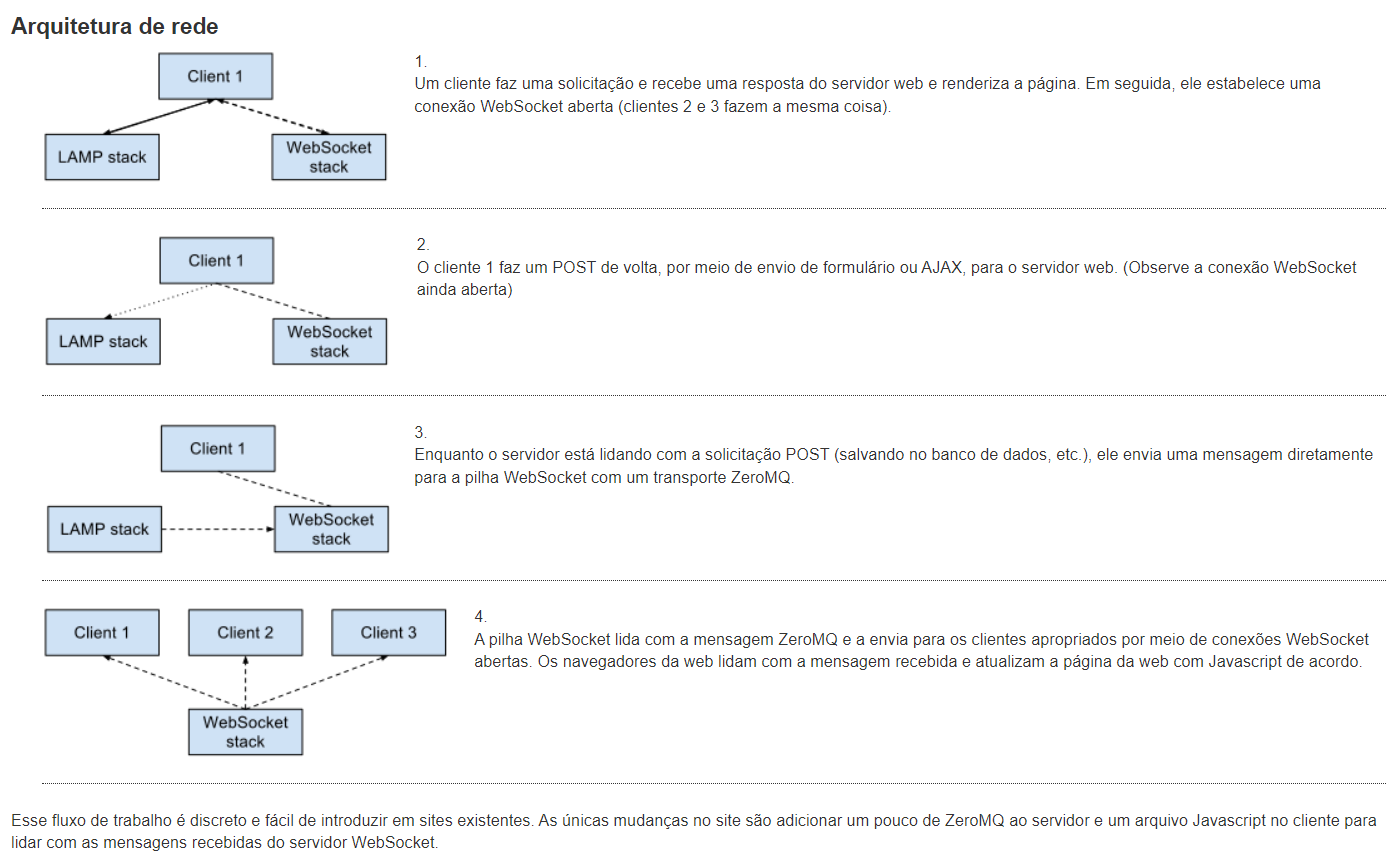
**Código**

****

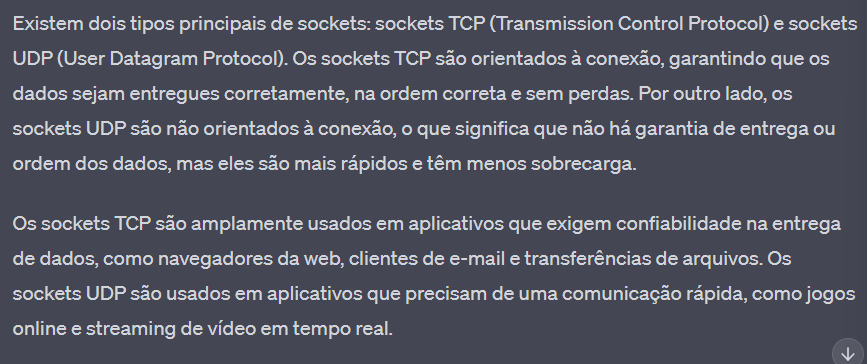
**Meta do projeto**

Quando um usuário, seja você mesmo em seu admin ou um usuário postando um comentário em seu blog, faz um POST através de um envio de formulário ou AJAX, queremos que essa alteração seja imediatamente enviada a todos os outros visitantes dessa página. Adicionaremos atualizações em tempo real ao nosso site sem interromper sua base de código ou afetar sua estabilidade atual.

**Estrutura de Redes com push (ainda não estudar essa parte)**

****

**. Socket**: permite a comunicação entre processos de software em uma rede, Força de estabelecer uma comunicação de 2 pontas **(bidirecional)**, criando um canal de comunicação entre um cliente e um servidor, permitindo a que eles troquem de dados



**. WebSockets:** Os WebSockets são uma tecnologia que estende a funcionalidade dos **sockets** para a **Web,** fornecendo maneiras de estabelecer uma ***conexão persistente e bidirecional*** entre um navegador da web e um servidor, permitindo uma comunicação em tempo real.

. Sendo sua principal diferencia entre os **sockets tradicionais**, é que os WebSockets são baseados em **TCP** e operam sobre o **protocolo HTTP** podendo assim passar por firewalls e proxies normalmente presentes nas redes corporativas e na Internet. Após uma vez estabelecido a conexão Websockets, tanto cliente como servidor podem enviar mensagens um para o outro no mesmo momento, sem a necessidade de fazer de fazer solicitações HTTP separadas, pois invés do servidor voltar com a requisição do cliente e fechar a conexão, o Websockets iria criar uma conexão aberta para receber requisição de dados de ambas as partes, criando um servidor de E/S onde ele armazena todas as conexões estabelecidas.

