



Sessão 1.

Cada vez que o cliente clicar em um produto (Detalhe ou ver mais) é gerado um uid temporário, no FrontEnd, que se ele sair da página, efetuar compra ou gerar um carrinho é disparado um evento, para uma API em node que grave as informações da **sessão 2**, mesmo vale se ele sair da pagina sem gerar um carrinho ou uma compra.

Podendo ser efetuado por monitoramento das ações do usuário no FrontEnd através de auxiliares de frameworks javascript da atualidade (emberjs, reactjs, angularjs etc...).

Sessão 2.

Esse evento tem como objetivo, enviar para a api as seguintes informações em formato de JSON:

Obs.: o Uid é um valor temporário seguindo os mesmos critérios dos bancos de dados noSQL, garantindo que os dados sejam referente a uma sessão (acesso a aquele momento onde gerou), Eliminando duplicidade e gerando dados consistentes.

```
{
  uid: xxx,
  pageUrl: xxxx,
  Cookie-user: xxxx,
  DataVisita: xxxx,
  Geolp: xxx,
  CarrinhoGerado: true|false,
  CompraEfetuada: true|false,
  produto: {
    id: xxxx,
    nome: xxxx,
    foto: xxx,
    disponibilidade: xxxx
  }
}
```

Sessão 3.

Os dados que foram enviados para a API tem o intuito de auxiliar os Gestores com as seguintes informações:

1. Produtos mais acessados.
2. Período em que o produto foi mais acessado.
3. Geolp do usuário para possível identificação do que mais é buscado por ele.
4. Cookie-user, para redirecionar a conteúdos similares mais buscados.
5. Compra efetuada (qual produto), gerou carinho (qual produto).

Essa api será desenvolvida em um Framework nodejs com padrão de desenvolvimento (arquitetura limpa, OO, SOLID e etc...), irá conter alguns methods tais como.:

1. Cadastrar um acesso.
 1. Entidade apresentada na **Sessão 2**.
2. Filtro de acesso e Download.
 1. Por período.
 2. Por período e compra.
 3. Produtos mais comprados em um período,
 4. Produtos que mais geraram carrinho.
 5. Produto mais acessado
 1. Efetuando um contador com base no id do produto e um distinct.
 6. Faixa de IP de uma região que mais acessou um produto.
 7. Dookie-user, identificar o que mais o usuário acessou.

Dessa forma será possível gerar relatórios em formatos (CSV, JSON e PDF) com totalizadores, em base64 com o formato explicito para integrações de FrontEnd e download do arquivo.

Banco de dados utilizado para o trabalho em questão é o Mongodb, pois trabalha com o formato JSON sendo bem robusto e rápido para consultas. Tem opções pre configuradas na AWS ou pode ser instalado em um servidor da própria empresa, podendo utilizar até mesmo um docker.

Sessão 3.

Collections a serem criadas (**products**, **productVisited**).

products irá conter uma estrutura de chave e valor no seguinte modelo

```
{
  id: xxx
  name: xxx,
  photo: base64,
  available: booleano
}
```

A photo pode ser um base64 da representação da imagem, fica melhor para representar em FrontEnd, por ser mais leve e exigir menos da parte lógica de programação, alternativa para desenvolver mais rapido.

productsVisited

```
{
  uid: xxx
  pageUrl: xxxx,
  Cookie-user: xxxx,
  DateVisit: xxxx,
  Geolp: xxx,
  cargenerate: true|false,
  buyefective: true|false,
  productId: xxxx
}
```

Através da pageUrl e possível fazer um relatório de quando acessos teve aquele determinado produto, juntamente com o id do produto que tem relação na collection (productId).

