**3. WebServices**

A constante necessidade de integrar sistemas para que seus processos pudessem ser entendíveis e unificados gerou a criação dos serviços web. Muitas vezes sistemas distintos possuem diferentes estruturas e linguagens, e esses serviços têm como foco exatamente eliminar este tipo de empecilho.

Um WebService nada mais é do que um software de interoperabilidade entre redes, que atua com uma interface de comunicação. Sendo assim, diferentes plataformas podem se entender, agindo sob um consumo de dados, ou seja, envia-se informações, e se recebe uma resposta após algum tipo de processamento (UFRGS, 2011).

**3.1 WebServices REST**

O termo REST foi utilizado pela primeira vez pelo criador do protocolo HTTP, que ressaltou em seus trabalhos que este seria um protocolo utilizado por qualquer sistema distribuído que precisasse de requisições, sem camadas adicionais, sob algum domínio na grande rede de internet. É exatamente desta forma que funciona um serviço REST.

Podemos citar os princípios do REST, e seus pontos favoráveis, como (UFRGS, 2011):

* **Cliente-Servidor**: Assim que é solicitada alguma informação, o servidor pode tanto rejeitar, como aceitar, e seguir com o processamento, devolvendo uma resposta a ação executada;
* **Stateless e Cache**: Serviços REST utilizam como ajuda na performance o cache, salvando em memória as informações para poderem ser reutilizadas no futuro, quando a mesma requisição for feita novamente. Junto a isso, existe o modo *Stateless,* que requer todas as informações obrigatórias para uma requisição, visto que nada fica salvo na parte do servidor.
* **Uniforme**: O padrão REST denomina que sua execução seja uniforme e simples, que pode até mesmo ser baixada e utilizada direto do computador, aumentando sua extensibilidade.
* **Altas requisições:** REST suporta grandes quantias de dados sem problema, resolvendo cenários corporativos, que necessitam dessa integração, da forma mais veloz possível.

**Contras do REST Webservices**

Em contrapartida a sua possível extensibilidade, perde-se um pouco de visibilidade nas requisições, o que em alguns casos pode ser um problema. Comparando o padrão REST com outros existentes, obtemos maior velocidade e simplicidade, porém, ainda peca-se um pouco em robustez, visto que fica a cargo dos desenvolvedores definirem bem sua estrutura. Isso acaba também impactando, algumas vezes, em falhas de segurança de dados.