# UniTunes Release 06/2016

Este documento tem como finalidade apresentar as features presentes na entrega do projeto UniTunes. Além disso, o documento apresenta também uma visão geral das principais diferenças de design e implementação entre o produto entregue e o projeto inicial deste.

## Entrega

O UniTunes está disponível em produção no seguinte endereço:

* <http://uni-tunes.herokuapp.com>

O usuário pode cadastrar-se na aplicação e ter acesso às features de Acadêmico e Autor. Caso deseje acessar as features de Administrador, as seguintes credenciais podem ser utilizadas:

* Usuário: admin@unitunes.com
* Senha: admin123

O código fonte da aplicação se encontra em:

* <https://github.com/fuadsaud/uniTunes/>

### Casos de uso entregues

Os seguintes casos foram propostos e especificados no projeto inicial:

* UC1: Autenticar
* UC2: Catálogo de produtos
* UC3: Visualizar dashboard de vendas
* UC4: Comprar produto
* UC5: Registrar novo usuário
* UC6: Recarregar créditos
* UC10: Gerenciar usuários
* UC11: Visualizar detalhes de Produto
* UC12: Remover produto
* UC13: Criar produto

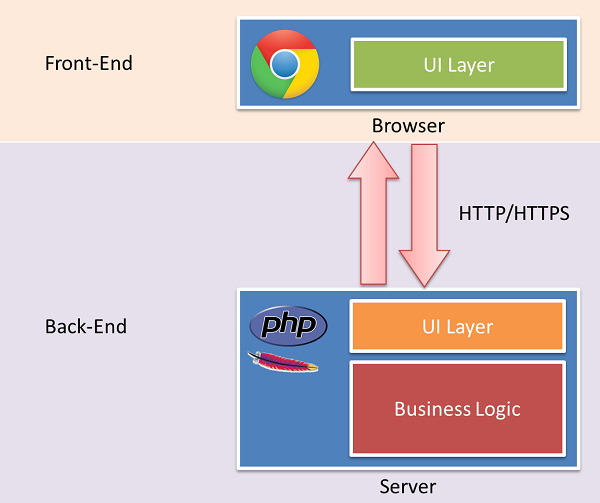
Levando em conta o curto prazo de entrega e a baixa disponibilidade de recursos disponível, o time optou por implementar os casos de uso abaixo pois considera que estes representem maior valor ao negócio neste primeiro momento. Alguns fluxos alternativos e regras de negócio destes casos de uso também não estarão presentes nesta entrega devido ao mesmo motivo.

* UC1: Autenticar
* UC2: Catálogo de produtos
* UC4: Comprar produto
* UC5: Registrar novo usuário
* UC11: Visualizar detalhes de Produto
* UC12: Remover produto
* UC13: Criar produto

## Alterações de design

### Design inicial

O projeto inicial consistia na construção de uma API REST capaz de fornecer serviços para uma aplicação front-end. A tecnologia escolhida para a implementação do sistema foi PHP e Angular, para o backend e frontend respectivamente.



### Design atual

Logo após o começo dos esforços de codificação, o time percebeu que as tecnologias escolhidas não seriam produtivas e comprometeriam a entrega no prazo determinado, pois:

* Muitos desenvolvedores precisariam realizar uma curva de aprendizado antes de começarem a ser produtivos nestas tecnologias;
* Haveria problemas ao integrar a aplicação frontend com os serviços da API

Feita esta constatação, foi sugerida a utilização de uma tecnologia que permitisse maior produtividade da equipe em um menor espaço de tempo. Realizou uma prova de conceito utilizando Ruby on Rails e verificou-se que se obteria esta vantagem e então o produto passou a ser construído nesta tecnologia.