Versão 0.3.1

Ponto de situação e roadmap para 0.4

Alterações desde a versão 0.2

- Adição de funcionalidades que permitem devolver os vários tipos de conteúdos à app:
  - Conteúdos “estáticos”: professores, escolas, …;
  - Testes (tanto de leitura como de escolha múltipla).
- Adição de funcionalidades que permitem receber testes resolvidos da app.

Alterações desde a versão 0.3

- Alteração do procedimento que recebe correções da app para permitir receber correções “parciais”;
- Início (finalmente!) da criação dos formulários de CRUD para o Back-Office;
- Limpeza de código.

Roadmap para 0.4

O que ficou por fazer e o que vamos fazer

O que ficou por fazer para 0.3 e 0.3.1

- Autenticação + Autorização;
- Estudo de tecnologias para o Back-Office:
  - Gravação de áudio no browser;
  - Edição (recortar) imagens no browser;
- Implementar SSL para encriptação.

Foco para 0.4

- Implementar autenticação + autorização;
- Continuar o Back-Office:
  - Implementar o resto dos formulários de CRUD;
  - Implementar o protótipo exploratório do Back-Office de AS.
- Melhorar a validação de dados, especialmente quando estes vêm do Back-Office.

“Coisas Técnicas”

Tecnologias escolhidas para melhorar performance

Ficheiros

Envio, recepção e armazenamento

Envio de ficheiros (para a app)

- 0.2: Utilização de Base64 Encoding no JSON:
  - Ineficiente: Conversão lenta + overhead de ~30% em relação ao tamanho original;
- 0.3: JSON devolvido com links + ficheiros estáticos:
  - Consideravelmente mais eficiente: Caching, não há necessidade de conversão, etc.;
  - App apenas precisa de usar o link para obter o ficheiro.

Receção de ficheiros (da app)

- 0.2: Semelhante ao envio na versão 0.2;
- 0.3: Uso de multipart/form-data para enviar dados:
  - Usado por browsers para o upload de ficheiros em forms;
  - (Mais) eficiente: Não há necessidade de fazer conversões;
  - Problema: API menos consistente.

Armazenamento de ficheiros (no servidor)

- 0.2: “Misto”, primariamente uso de BLOB storage:
  - Problemático: SGBD não foram feitos para isto;
  - Performance: Mais lento que o FS para ficheiros > 1 MB;
  - Caching por parte do servidor mais complexa;
  - Backups mais simples.
- 0.3: Uso do FS para guardar ficheiros, BD guarda link:
  - Backups mais complexos (não é só um “dump” da BD).

Queries mais complexas

SELECTs e INSERTs em múltiplas tabelas

Problemas

- Problema: Fazer queries diretamente na lógica do servidor:
  - Aumento da complexidade do código (várias queries, podia (ou não) implicar *callback hell*);
  - Se o SGBD estivesse presente noutra máquina, seriam mais dados a “viajar” na rede.

Solução

- Uso de *stored procedures* no SGBD:
  - Toda a lógica mais complexa está só num lado:
    - Saber qual a tabela à qual se tem que fazer um JOIN;
    - INSERTs em múltiplas tabelas.
  - Só há uma query a ser executada (implica menos dados a viajar pela rede).