Padrões para desenvolvimento de conteúdos

Sumário

1- Descrição

Este documento descreve os padrões que devem ser utilizados, no desenvolvimento de conteúdos que podem armazenar dados no sistema AVA MEC. Ao desenvolver um conteúdo, seguindo as instruções deste documento, será possível, armazenar informações do conteúdo, tais como: unidades, tópicos, atividades e seus respectivos gabaritos.

Descreve também, uma API com diversos métodos que podem ser utilizados para realizar a comunicação do conteúdo com o sistema. Utilizando esses métodos, será possível armazenar e obter dados da participação dos cursistas, tais como: última página lida, respostas enviadas para as atividades, progresso no curso etc. E também obter dados de configuração da conclusão das unidades, verificar se o usuário está utilizando o sistema em alto contraste, se existe próxima unidade para o curso, entre outros.

Com os dados de conteúdo e participação dos cursistas, armazenados no sistema para um determinado curso, será possível realizar avaliação e certificação automática dos cursistas que participarem do curso.

Caso o conteúdo não utilize os padrões estabelecidos neste documento, será possível adicioná-lo normalmente no sistema, mas não será possível realizar a comunicação desse conteúdo com o sistema e realizar a avaliação e certificação automática, por meio de atividades disponibilizadas no conteúdo.

2- Objetivos

- Desenvolver conteúdos que comunicam com o sistema;
- Permitir o armazenamento da participação dos cursistas;
- Permitir utilizar atividades do conteúdo, na configuração da forma de avaliação de cursos;
- Permitir utilizar atividades do conteúdo, na avaliação e certificação automática.

3 - Configurações do conteúdo

3.1 - Regras para envio do arquivo do conteúdo

- Não será permitido o envio de arquivos executáveis (.exe, .bat, .scr, .bin e .sh).
- O tamanho máximo de cada arquivo do conteúdo, deve ser 100MB.
- O nome de cada arquivo ou diretório do conteúdo deve possuir somente letras, números, . (ponto) e _ (underline)

- Se o conteúdo for composto por vários arquivos e diretórios, deve ser gerado um arquivo compactado nos formatos zip ou rar, contendo todos os seus arquivos e diretórios, para que seja possível enviá-lo de uma só vez. Não há limite de tamanho para envio do arquivo completo, mas cada arquivo de dentro do conteúdo deve ter no máximo 100MB.

3.2 - Desenvolvimento de conteúdos que comunicam com o sistema

3.2.1 - Tecnologia utilizada

Para que seja possível a comunicação do conteúdo com o sistema, ele deve ser desenvolvido com alguma tecnologia que permite a utilização da API disponibilizada pelo sistema (verificar item 5).

3.2 - Estrutura do conteúdo

Para armazenar no sistema, os dados do conteúdo, ele deve ser desenvolvido de acordo com a seguinte estrutura:

- **Unidades** - deve possuir pelo menos uma unidade;
- **Tópicos** - tópicos de cada unidade (não obrigatório);
- **Atividades** - cada atividade deve estar ligada a uma unidade (não é obrigatório que existam atividades para todas as unidades).

Esses dados devem ser armazenados no sistema, utilizando um arquivo de configuração (json), preenchido conforme as instruções do item 4. Assim que o conteúdo completo for enviado, o sistema identificará esse arquivo de configuração e realizará o cadastro da estrutura do conteúdo: unidades, tópicos e atividades.

Observação:

- *Se o arquivo de configuração não for identificado, será criada, automaticamente, uma unidade padrão, com o mesmo nome do conteúdo.*

4 - Arquivo de configuração do conteúdo

4.1 - Configuração do arquivo

O arquivo de configuração do conteúdo deve:

- ser disponibilizado na pasta raíz do conteúdo;
- ser um arquivo no formato Json
- ter o nome: **configuracao_conteudo.json**

4.2 - Conteúdo do arquivo

- Para as unidades do conteúdo, devem constar as seguintes informações:
  - Identificador
  - Nome da unidade
  - Descrição da unidade
  - Se permite porcentagem
  - Lista de tópicos da unidade
  - Lista de atividades da unidade
- Para cada tópico de uma unidade, devem constar as seguintes informações:
  - Identificador
  - Nome do tópico
  - Identificador do tópico pai (se for um subtópico)
- Para cada atividade de uma unidade, devem constar as seguintes informações:
  - Identificador
  - Nome da atividade
  - Lista de questões
- Para cada questão de uma atividade, devem constar as seguintes informações:
  - Tipo da questão: INTEGRAL ou PROPORCIONAL
  - Identificador
  - Enunciado
  - Gabarito de cada alternativa, contendo:
    - Texto da alternativa
    - Chave da alternativa
    - Valor da alternativa

4.3 - Definição dos atributos

Os atributos utilizados para cadastrar as informações do conteúdo, devem obedecer às seguintes regras:

- Identificador de unidades, tópicos, atividades e questões: Pode conter letras e números (String);
- Nome de unidades, tópicos e atividades, descrições, enunciado de questões e texto de alternativas: Pode conter letras e números (String)
- Permite porcentagem:
  - Deve ser true ou false;
  - Colocar como true somente para as unidades do conteúdo que estão utilizando o método que armazena a porcentagem de conclusão do conteúdo, para os cursistas.
- Identificador pai de um tópico, deve ser preenchido somente para subtópicos;
- Tipo da questão:

- Deve ser INTEGRAL ou PROPORCIONAL;
- Para avaliações do tipo INTEGRAL, o aluno terá que acertar todas as alternativas para receber nota para a questão (exemplo: questão que possui somente uma alternativa correta);
- Para avaliações do tipo PROPORCIONAL, o aluno receberá nota para cada alternativa que acertar (exemplo: questões com alternativas do tipo certo e errado).

- Gabarito da questão:
  - Chave: Pode conter letras e números;
  - Valor: Pode conter letras e números.

4.4 - Exemplo de conteúdo de um arquivo json

Apresentamos a seguir, o modelo de um json, para cadastrar as informações de um conteúdo que possui: 3 unidades e seus respectivos tópicos e atividades:

```
[{
"identificador": "unidade_01",
"nome": "Primeira unidade do conteúdo",
	"descricao": "Descrição da primeira unidade do conteúdo.",
	"permitePorcentagem": "true",
	"topicos": [
		{
			"identificador": "topico_1",
			"nome": "Tópico 1 da unidade 01"
		},
		{
			"identificador": "subtopico_1.1",
			"nome": "Subtópico 1 do tópico 1",
			"identificadorPai": "topico_1"
		},
		{
			"identificador": "topico_2",
			"nome": "Tópico 2 da unidade 01"
		}
	],
	"atividades": [
		{
			"identificador": "atividade_1_uni_1",
			"nomeAtividade": "Atividade 1 da unidade 1",
			"questoes": [
				{
					"tipoQuestaoString": "PROPORCIONAL",
					"identificador": "Q1_A1_U1_Enumere",
					"enunciado": "Enumere a segunda coluna de acordo
com a primeira:",
					"gabaritos": [
						{
						"alternativa": "Texto da alternativa 1",
						"chave": "1",
						"valor": "3"
						},
```

```
                                        {
                                        "alternativa": "Texto da alternativa 2",
                                        "chave": "2",
                                        "valor": "5"
                                        },
                                        {
                                        "alternativa": "Texto da alternativa 3",
                                        "chave": "3",
                                        "valor": "1"
                                        },
                                        {
                                        "alternativa": "Texto da alternativa 4",
                                        "chave": "4",
                                        "valor": "2"
                                        },
                                        {
                                        "alternativa": "Texto da alternativa 5",
                                        "chave": "5",
                                        "valor": "4"
                                        }
                                  ]}
                        ]}
      ]
},
{
"identificador": "unidade_02",
      "nome": "Segunda unidade do conteúdo",
      "descricao": "Descrição da segunda unidade do conteúdo.",
      "permitePorcentagem": "true",
      "topicos": [
            {
                  "identificador": "topico_1",
                  "nome": "Tópico 1 da unidade 02"
            },
            {
                  "identificador": "topico_2",
                  "nome": "Tópico 2 da unidade 02"
            }
      ],
      "atividades": [
            {
                  "identificador": "atividade_1_uni_2",
                  "nomeAtividade": "Atividade 1 da unidade 2",
                  "questoes": [
                        {
                              "tipoQuestaoString": "PROPORCIONAL",
                              "identificador": "Q1_A1_U2_Certo_Errado",
                              "enunciado": "Marque certo ou errado para cada
alternativa",
                              "gabaritos": [
                                    {
                                    "alternativa": "Texto da alternativa 1",
                                    "chave": "1",
                                    "valor": "certo"
                                    },
                                    {
                                    "alternativa": "Texto da alternativa 2",
```

```
                        "chave": "2",
                        "valor": "errado"
                        },
                        {
                        "alternativa": "Texto da alternativa 3",
                        "chave": "3",
                        "valor": "errado"
                        },
                        {
                        "alternativa": "Texto da alternativa 4",
                        "chave": "4",
                        "valor": "certo"
                        },
                        {
                        "alternativa": "Texto da alternativa 5",
                        "chave": "5",
                        "valor": "certo"
                        }
                ]
        },
        {
                "tipoQuestaoString": "PROPORCIONAL",
                "identificador": "Q2_A1_U2_Varias_Corretas",
                "enunciado": "Selecione as alternativas corretas ",
                "gabaritos": [
                        {
                        "alternativa": "Texto da alternativa 1",
                        "chave": "1",
                        "valor": "1"
                        },
                        {
                        "alternativa": "Texto da alternativa 2",
                        "chave": "2",
                        "valor": "0"
                        },
                        {
                        "alternativa": "Texto da alternativa 3",
                        "chave": "3",
                        "valor": "0"
                        },
                        {
                        "alternativa": "Texto da alternativa 4",
                        "chave": "4",
                        "valor": "1"
                        },
                        {
                        "alternativa": "Texto da alternativa 5",
                        "chave": "5",
                        "valor": "1"
                        }
                ]
        },
        {
                "tipoQuestaoString": "INTEGRAL",
                "identificador": "Q3_A1_U2_Correta",
                "enunciado": "Marque alternativa correta",
                "gabaritos": [
```

```
                                        {
                                        "alternativa": "Texto da alternativa 1",
                                        "chave": "1",
                                        "valor": "0"
                                        },
                                        {
                                        "alternativa": "Texto da alternativa 2",
                                        "chave": "2",
                                        "valor": "0"
                                        },
                                        {
                                        "alternativa": "Texto da alternativa 3",
                                        "chave": "3",
                                        "valor": "1"
                                        },
                                        {
                                        "alternativa": "Texto da alternativa 4",
                                        "chave": "4",
                                        "valor": "0"
                                        },
                                        {
                                        "alternativa": "Texto da alternativa 5",
                                        "chave": "5",
                                        "valor": "0"
                                        }
                                ]}
                        ]}
        ]
},
{
"identificador": "unidade_03",
        "nome": "Terceira unidade do conteúdo",
        "descricao": "Descrição da terceira unidade do conteúdo.",
        "permitePorcentagem": "true",
        "topicos": [
                {
                        "identificador": "topico_1",
                        "nome": "Tópico 1 da unidade 03"
                },
                {
                        "identificador": "topico_2",
                        "nome": "Tópico 2 da unidade 03"
                }
        ],
        "atividades": [
                {
                        "identificador": "atividade_1_uni_3",
                        "nomeAtividade": "Atividade 1 da unidade 3",
                        "questoes": [
                                {
                                        "tipoQuestaoString": "PROPORCIONAL",
                                        "identificador": "Q1_A1_U3_Certo_Errado",
                                        "enunciado": "Marque certo ou errado para cada
alternativa",
                                        "gabaritos": [
                                                {
                                                "alternativa": "Texto da alternativa 1",
```

```
                                    "chave": "1",
                                    "valor": "errado"
                                    },
                                    {
                                    "alternativa": "Texto da alternativa 2",
                                    "chave": "2",
                                    "valor": "certo"
                                    },
                                    {
                                    "alternativa": "Texto da alternativa 3",
                                    "chave": "3",
                                    "valor": "errado"
                                    },
                                    {
                                    "alternativa": "Texto da alternativa 4",
                                    "chave": "4",
                                    "valor": "certo"
                                    },
                                    {
                                    "alternativa": "Texto da alternativa 5",
                                    "chave": "5",
                                    "valor": "certo"
                                    }
                              ]}
                  ]
            },
            {
                  "identificador": "atividade_2_uni_3",
                  "nomeAtividade": "Atividade 2 da unidade 3",
                  "questoes": [
                        {
                              "tipoQuestaoString": "INTEGRAL",
                              "identificador": "Q1_A2_U3_Correta",
                              "enunciado": "Marque a alternativa correta",
                              "gabaritos": [
                                    {
                                    "alternativa": "Texto da alternativa 1",
                                    "chave": "1",
                                    "valor": "1"
                                    },
                                    {
                                    "alternativa": "Texto da alternativa 2",
                                    "chave": "2",
                                    "valor": "0"
                                    },
                                    {
                                    "alternativa": "Texto da alternativa 3",
                                    "chave": "3",
                                    "valor": "0"
                                    },
                                    {
                                    "alternativa": "Texto da alternativa 4",
                                    "chave": "4",
                                    "valor": "0"
                                    }
                              ]
                        },
```

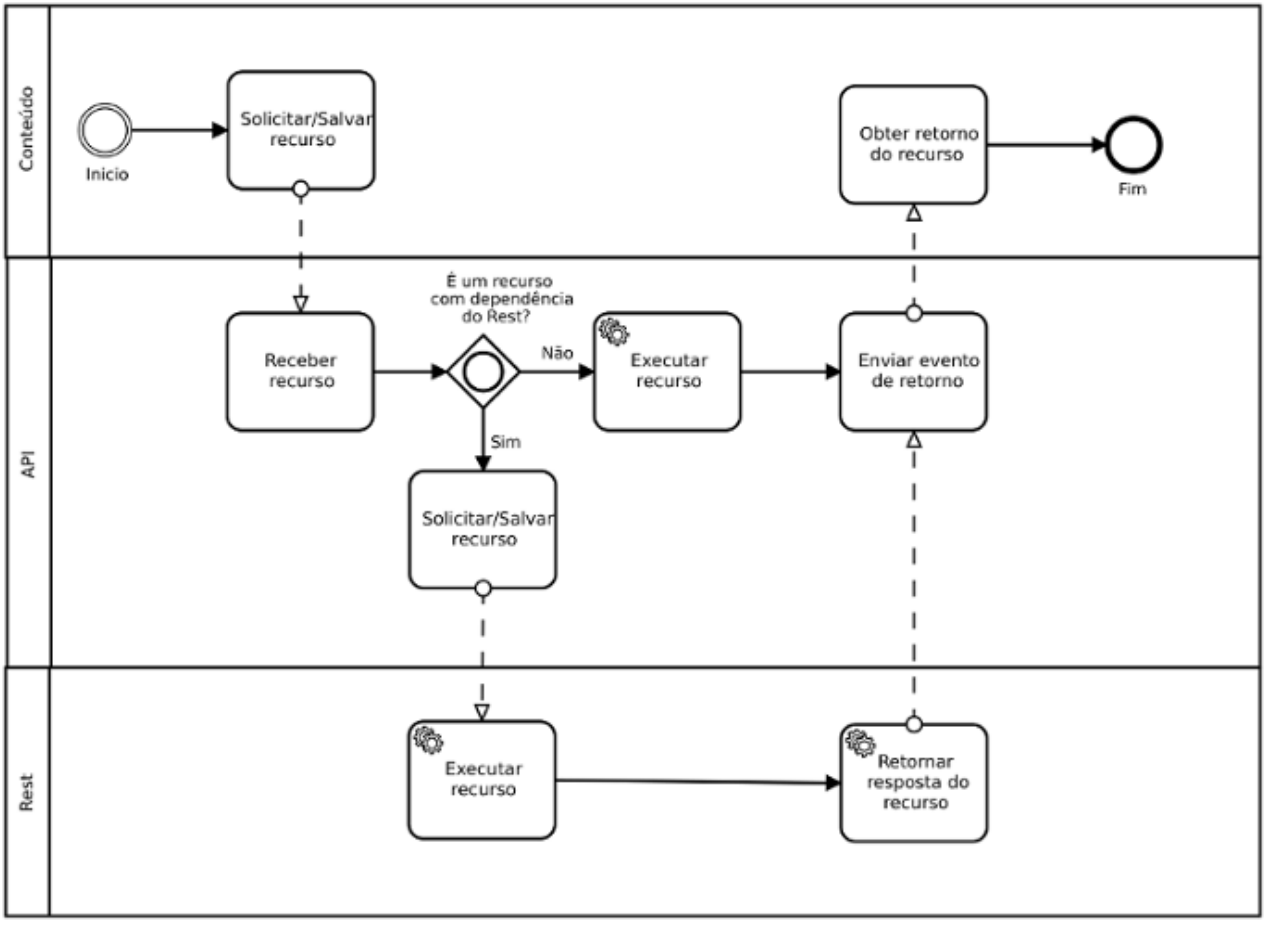
```
                        {
                              "tipoQuestaoString": "INTEGRAL",
                              "identificador": "Q2_A2_U3_Correta",
                              "enunciado": "Marque alternativa correta",
                              "gabaritos": [
                                    {
                                    "alternativa": "Texto da alternativa 1",
                                    "chave": "1",
                                    "valor": "0"
                                    },
                                    {
                                    "alternativa": "Texto da alternativa 2",
                                    "chave": "2",
                                    "valor": "0"
                                    },
                                    {
                                    "alternativa": "Texto da alternativa 3",
                                    "chave": "3",
                                    "valor": "1"
                                    },
                                    {
                                    "alternativa": "Texto da alternativa 4",
                                    "chave": "4",
                                    "valor": "0"
                                    }
                              ]}
                  ]}
      ]}
]
```

5 - API de conteúdo

Para permitir a comunicação dos conteúdos com o sistema AVA MEC, disponibilizamos a API de conteúdos (Bridge REST). Essa API, é para fornecer uma interface de comunicação entre o conteúdo e a plataforma AVA MEC.

A API foi desenvolvida utilizando o *TypeScript* e o *Post Message*, que permitem uma comunicação segura, entre domínios diferentes.

Segue abaixo, um diagrama que representa a comunicação do conteúdo com a API e os serviços rest.



Verifique a seguir, as configurações que devem ser realizadas no conteúdo, para permitir sua comunicação com a API.

Lembrando que, será possível testar a comunicação do conteúdo com a API, somente se o conteúdo estiver hospedado em alguma instituição do sistema AVA MEC e estiver sendo acessado, por meio de um curso (iframe do curso).

5.1 - Incluindo a API no conteúdo

Para utilizar a API, em seu conteúdo, realize o *download*, do arquivo minificado da API, clicando aqui, e inclua-o em seu conteúdo. Verifique no exemplo abaixo, a inclusão do script, no cabeçalho do HTML.

```
<script src="bridge-rest-api.bundle.js"></script>
```

Se preferir, aponte diretamente para o CDN, conforme o exemplo abaixo:

```
<script
src="https://cdn.jsdelivr.net/npm/@labtime/avamec-api-bridge-cliente@latest/dist/bridge-rest-api.bundle.js"></script>
```

5.2 - Instanciando o objeto BridgeRestApi

Para que seja possível utilizar os métodos disponibilizados na API, será necessário instanciar o objeto, conforme o exemplo a seguir.

```
var API = new BridgeRestApi();
```

5.3 - Efetuando a chamada de um método

Para efetuar a chamada de um dos métodos fornecidos pela API, será necessário utilizar a constante "API". Verifique abaixo, a chamada de um método que obtém a última página do conteúdo, que foi visualizada por um cursista.

```
onClick="API.registrarUltimaPaginaAcessada('unidade_02','unidades/uni02_sl01');"
```

Verifique na tabela do item 5.5, a lista de todos os métodos disponibilizados pela API.

5.4 - Obtendo retorno dos métodos

Para obter o retorno dos métodos disponibilizados pela API, é necessário utilizar a interface *EventListener*. Essa interface permite monitorar os eventos de retorno de cada método invocado. Verifique abaixo um exemplo de retorno de um método:

```
// Adicionando listener
window.addEventListener("evObtemRegistraUltimaPaginaAcessada",
resultadoRegistrarUltimaPaginaAcessada, false);
function resultadoRegistrarUltimaPaginaAcessada(evento){
    console.log(evento.detail);
```

```
}
```

As informações das respostas, estão disponibilizadas em um arquivo json, no atributo ***detail*** do evento e possui a seguinte estrutura:

- **status**: os possíveis retornos de status de um método são:
  - 200: indica que o método foi executado com sucesso;
  - 412: indica que algum atributo obrigatório não foi enviado ou que alguma pré-condição não foi atendida. Em muitos casos, são retornadas mensagens informando o motivo do erro.
  - 500: indica que houve um erro interno no servidor.
- **data:** é o valor do retorno. Os dados são retornado de acordo com cada método, se por exemplo, for solicitado, obter a última página que um cursista acessou, será retornada uma string referenciando a URL da página.

Verifique no item 5.5, a descrição de retorno de todos os métodos disponibilizados pela API.

5.5 - Descrição dos métodos

Verifique na tabela a seguir, a visão geral de todos os métodos disponibilizados na API de conteúdos. E em seguida, a descrição e exemplo de retorno de cada método

Objeto	Nome do método	Método
Dados genéricos	Armazenar dados genéricos	registrarDadosGenericos(chave, valor)
	Obter dados genéricos	obterDadosGenericos(chave)
Última página acessada	Armazenar a última página acessada	registrarUltimaPaginaAcessada(identificadorUnidade, urlUltimaPaginaAcessada)
	Obter última página acessada	obterUltimaPaginaAcessada()
Unidade	Obter dados de configuração para conclusão de uma unidade	obterConfiguracaoConclusaoUnidade(identificadorUnidade)
	Verificar se existe próxima unidade para o curso	obterSeExisteProximaUnidade(identificadorUnidade)
	Verificar se existe unidade anterior para o curso	obterSeExisteUnidadeAnterior(identificadorUnidade)
	Acessar próxima unidade do curso	obterProximaUnidade(identificadorUnidade)
	Acessar unidade anterior do curso	obterUnidadeAnterior(identificadorUnida

		de)
	Acessar unidade do curso	obterUnidade(identificadorUnidade)
Porcentagem de conclusão	Armazenar a porcentagem de conclusão de uma unidade	registrarPorcentagemConclusaoUnidade(identificadorUnidade, porcentagem)
Atividade	Obter dados de configuração de uma atividade	obterConfiguracaoAtividade(identificadorAtividade)
	Armazenar as respostas enviadas para uma atividade	registrarRespostaAtividade(atividade)
	Obter respostas armazenadas para uma atividade	obterRespostaAtividade(identificadorAtividade)
Conclusão do cursista	Obter dados de conclusão da unidade	obterDadosConclusaoUnidade(identificadorUnidade)
	Obter dados de conclusão do curso	obterDadosConclusaoCurso()
Contraste	Obter configuração de contraste do usuário	obterStatusContraste()
Tamanho de fonte	Obter configuração de tamanho da fonte	obterTamanhoFonteSistema()
Conteúdo	Obter informação para ir ao primeiro item do conteúdo	obterAncoraIrParaConteudo()
Curso	Obter informações do curso	obterDadosCurso()

5.5.1 - Dados genéricos

Permite armazenar, para um cursista em um determinado curso, qualquer informação referente ao conteúdo para que, posteriormente, seja possível recuperar essa informação. Exemplo: Pontuação obtida por um usuário em um jogo, cada página acessada pelo usuário etc.

1 - Armazenar dados genéricos

- **Descrição:** Salva uma lista de tuplas para o usuário logado (com chave e valor), e se existir tupla com a chave informada atualiza o valor;
- **Método:** registrarDadosGenericos(chave, valor)
- **Parâmetros**: chave e valor (string)
- **Evento de retorno:** evObtemRegistraDadosGenericos

Exemplos:

Chamada do método:

```
//Armazena o slide que o usuário acessou
onClick="API.registrarDadosGenericos('acessou_uni02_sl02','1');"
```

```
//Armazena a pontuação que o usuário obteve em um jogo
onClick="API.registrarDadosGenericos('jogo_uni02','80');"
```

Retorno do método:

```
// Adicionando listener
window.addEventListener("evObtemRegistraDadosGenericos",
resultadoRegistrarDadosGenericos, false);
function resultadoRegistrarDadosGenericos(evento){
    console.log(evento.detail);
}
```

```
//Dados retornados quando a ação é executada com sucesso
{
status: 200,
data: "Operação efetuada com sucesso."
}
```

//Dados retornados quando o usuário não for um cursista

{

status: ,

data: {

codigo: "ME34_01",

mensagem: "Não foi possível armazenar as informações solicitadas, pois o usuário não é um cursista ativo do curso."

}

```
//Dados retornados quando não foram informados todos os parâmetros obrigatórios
{
status: 412,
data: {
       codigo: "ME34",
       mensagem: "Não foi possível armazenar as informações solicitadas, pois os
dados enviados estão incorretos."
       }
}
```

2 - Obter dados genéricos

- **Descrição:** obtém as tuplas cadastrada para o usuário logado, que deve ser um cursista, conforme prefixo informado no parâmetro chave;
- **Método:** obterDadosGenericos(chave)
- **Parâmetros**: chave e valor (string)
- **Evento de retorno:** evObtemRegistraDadosGenericos

Exemplos:

Chamada do método:

```
//Obtém o slide que o usuário acessou
onClick="API.obterDadosGenericos('acessou_uni02_sl02');"
```

```
//Obtém a pontuação que o usuário alcançou em um jogo
onClick="API.obterDadosGenericos('jogo_uni02');"
```

Retorno do método:

```
// Adicionando listener
window.addEventListener("evObtemDadosGenericos", resultadoObtemDadosGenericos,
false);
function resultadoObtemDadosGenericos(evento){
    console.log(evento.detail);
}
```

```
//Dados retornados quando a ação é executada com sucesso
{
status: 200,
data: [
      {
            id: 10,
            chave: "acessou_uni02_sl_02",
            valor: "1",
            mapaAtributos: {}
      }
]
}
```

//Dados retornados quando o usuário não for um cursista

{

status: ,

data: {

codigo: "ME34_01",

mensagem: "Não foi possível armazenar as informações solicitadas, pois o usuário não é um cursista ativo do curso."

}

```
//Dados retornados quando não foram informados todos os parâmetros obrigatórios
{
status: 412,
data: {
       codigo: "ME37",
       mensagem: "Não foram encontrados dados com os parâmetros informados."
       }
}
```

5.5.2 - Última página acessada

Permite armazenar a última página do conteúdo de um curso, que o cursista acessou, para que posteriormente seja possível recuperar essa informação. Desta forma, será possível redirecionar o cursista para o ponto que ele parou na última vez em que acessou o conteúdo do curso.

1 - Armazenar última página acessada

- **Descrição:** armazena a última página que o usuário acessou;
- **Método:** registrarUltimaPaginaAcessada(identificadorUnidade, urlUltimaPaginaAcessada)
- **Parâmetros**: identificador da unidade e caminho da página (não obrigatório)
- **Evento de retorno:** evObtemRegistraUltimaPaginaAcessada

Observações:

- *Informar o caminho da página conforme o pacote do conteúdo. Se, por exemplo, o arquivo estiver na pasta principal do pacote do conteúdo, informar o seguinte caminho: nomedoarquivo.html. O prefixo da URL, será adicionado pela API, conforme o domínio do sistema.*
- *Se o caminho da página não for enviado, a API utilizará a última página acessada, que está armazenada nos dados do navegador do usuário (window.location.pathname), podendo em alguns casos, não corresponder à página que o usuário está acessando, dentro do conteúdo.*

Exemplos:

Chamada do método:

```
//Armazena a última página acessada por um usuário
onClick="API.registrarUltimaPaginaAcessada('unidade_01','unidades/uni01_sl01');"
```

Retorno do método:

```
// Adicionando listener
window.addEventListener("evObtemRegistraUltimaPaginaAcessada",
resultadoRegistrarUltimaPaginaAcessada, false);
function resultadoRegistrarUltimaPaginaAcessada(evento){
    console.log(evento.detail);
}
```

```
//Dados retornados quando a ação é executada com sucesso
{
status: 200,
data: "Operação efetuada com sucesso."
}
```

//Dados retornados quando o usuário não for um cursista

{

status: ,

data: {

codigo: "ME34_01",

mensagem: "Não foi possível armazenar as informações solicitadas, pois o usuário não é um cursista ativo do curso."

}

```
//Dados retornados quando não foram informados todos os parâmetros obrigatórios
{
status: 412,
data: {
       codigo: "ME34",
       mensagem: "Não foi possível armazenar as informações solicitadas, pois os
dados enviados estão incorretos."
       }
}
```

2 - Obter última página acessada

- **Descrição:** obtém a última página do conteúdo de um curso, que o usuário acessou;
- **Método:** obterUltimaPaginaAcessada()
- **Parâmetros**: sem parâmetros
- **Evento de retorno:** evObtemUltimaPaginaAcessada

Obs: Se o conteúdo não utilizar o método que armazena a última página que um usuário acessou, o sistema irá armazenar, automaticamente, a página inicial da última unidade que o usuário acessou. Portanto, neste caso, a última página armazenada, pode não corresponder à última página que o usuário realmente acessou dentro do conteúdo.

Exemplos:

Chamada do método:

```
//Obtém a última página que um usuário acessou
onClick="API.obterUltimaPaginaAcessada();"
```

Retorno do método:

```
// Adicionando listener
window.addEventListener("evObtemUltimaPaginaAcessada",
resultadoObtemUltimaPaginaAcessada, false);
function resultadoObtemUltimaPaginaAcessada(evento){
    console.log(evento.detail);
}
```

//Dados retornados quando o usuário não for um cursista

{

status: ,

data: {

codigo: "ME34_01",

mensagem: "Não foi possível armazenar as informações solicitadas, pois o usuário não é um cursista ativo do curso."

}

```
//Dados retornados quando a ação é executada com sucesso
{
status: 200,
data:
"http://www.labtime.ufg.brava-mec-ws/instituicao/labtime/conteudo/modulo/11/unidades
```

```
/uni01_sl01"
}
```

5.5.3 - Unidade

Para unidades, são disponibilizados métodos que permitem obter as configurações de conclusão de uma unidade, verificar se existe próxima unidade para o curso e se existir, permite acessar essa unidade.

1 - Obter dados de configuração para conclusão de uma unidade

- **Descrição:** obtém os requisitos configurados para a conclusão de uma determinada unidade em um curso;
- **Método:** obterConfiguracaoConclusaoUnidade(identificadorUnidade)
- **Parâmetros**: identificador da unidade
- **Evento de retorno:** evObtemDadosConfiguracaoConclusaoUnidade

Exemplos:

Chamada do método:

```
//Obtém os dados de configuração para conclusão de uma unidade
onClick="API.obterConfiguracaoConclusaoUnidade('unidade_01');"
```

Retorno do método:

```
// Adicionando listener
window.addEventListener("evObtemDadosConfiguracaoConclusaoUnidade",
resultadoObterDadosConfiguracaoConclusaoUnidade, false);
function resultadoObterDadosConfiguracaoConclusaoUnidade(evento){
    console.log(evento.detail);
}
```

```
//Dados retornados quando a ação é executada com sucesso
{
status: 200,
data: {
id: 20,
realizarAtividade: true,
acessarUnidade: false,
navegarConteudo: true,
aprovarComMedia: true,
mediaFinalUnidade: 7,
avaliativas: [
```

```
{
	id:30,
	peso: 1,
	considerarNaMediaAtividade: true,
	ativo: false,
	atividade: {
		id: 40,
		identificador: "Atividade1_uni1"
	}
}]
}
```

//Dados retornados quando o usuário não for um cursista

{

status: ,

data: {

codigo: "ME34_01",

mensagem: "Não foi possível armazenar as informações solicitadas, pois o usuário não é um cursista ativo do curso."

}

```
//Dados retornados quando não foram informados todos os parâmetros obrigatórios
{
status: 412,
data: {
	codigo: "ME37",
	mensagem: "Não foram encontrados dados com os parâmetros informados."
	}
}
```

2 - Verificar se existe próxima unidade para o curso

- **Descrição:** permite verificar se existe próxima unidade para um determinado curso;
- **Método:** obterSeExisteProximaUnidade(identificadorUnidade)
- **Parâmetros**: identificador da unidade
- **Evento de retorno:** evObtemSeExisteProximaUnidade

Exemplos:

Chamada do método:

```
//Verifica se existe próxima unidade
onClick="API.obterSeExisteProximaUnidade('unidade_01');"
```

Retorno do método:

```
// Adicionando listener
window.addEventListener("evObtemSeExisteProximaUnidade",
resultadoObterSeExisteProximaUnidade, false);
function resultadoObterSeExisteProximaUnidade(evento){
    console.log(evento.detail);
}
```

```
//Dados retornados quando existir próxima unidade para o curso
{
status: 200,
data: true
}
```

```
//Dados retornados quando não existir próxima unidade para o curso
{
status: 200,
data: false
}
```

```
//Dados retornados quando não foram informados todos os parâmetros obrigatórios
{
status: 412,
data: {
       codigo: "ME37",
       mensagem: "Não foram encontrados dados com os parâmetros informados."
       }
}
```

3 - Acessar próxima unidade do curso

- **Descrição:** redireciona o usuário para a próxima unidade do curso (se existir);
- **Método:** obterProximaUnidade(identificadorUnidade)
- **Parâmetros**: identificador da unidade
- **Evento de retorno:** evObtemProximaUnidade

Exemplos:

Chamada do método:

```
//Obtém os dados de configuração para conclusão de uma unidade
```

```
onClick="API.obterProximaUnidade('unidade_01');"
```

Retorno do método:

Se existir próxima unidade para o curso, o usuário é redirecionado para a página inicial da próxima unidade do curso.

```
//Dados retornados quando não forem informados todos os parâmetros obrigatórios
{
status: 412,
data: {
       codigo: "ME37",
       mensagem: "Não foram encontrados dados com os parâmetros informados."
       }
}
```

```
//Dados retornados quando não existir próxima unidade para o curso
{
status: 412,
data: {
       codigo: "ME49",
       mensagem: "Não existem próximas unidades para o curso."
       }
}
```

```
//Dados retornados quando a unidade informada não estiver sendo utilizada no curso
{
status: 412,
data: {
       codigo: "ME48",
       mensagem: "Não foi possível identificar a próxima unidade, pois a unidade
informada, não está sendo utilizada no curso."
       }
}
```

```
//Dados retornados quando a unidade informada estiver bloqueada utilizada no curso
{
status: 412,
data: {
       codigo: "ME87",
       mensagem: "Não foi possível acessar a unidade, pois ela está bloqueada."
       }
}
```

5.5.4 - Porcentagem de conclusão

Permite armazenar, para um determinado usuário, a porcentagem do conteúdo de uma unidade que ele visualizou. Neste caso, o conteúdo deve realizar, de alguma forma, o cálculo da porcentagem realizada e utilizar o sistema somente para armazenar essa quantidade.

Uma sugestão é que, ao desenvolver o conteúdo, seja utilizado o método que permite armazenar dados genéricos, para armazenar cada página que o usuário acessou e, posteriormente, obter esses dados e realizar o cálculo da porcentagem de conclusão da unidade.

Observação:

- *Se, ao armazenar a porcentagem de conclusão de uma unidade, for identificado que o usuário concluiu todos os requisitos para conclusão da unidade, o sistema irá, automaticamente, concluir o cursista na unidade. Podendo também, aprovar o cursista no curso, caso ele tenha concluído todos os requisitos para conclusão do curso e se o curso estiver configurado para avaliar automaticamente os cursistas.*

1 - Armazenar a porcentagem de conclusão de uma unidade

- **Descrição:** armazena a porcentagem de conclusão de um usuário para uma unidade do curso;
- **Método:** registrarPorcentagemConclusaoUnidade(identificadorUnidade, porcentagem)
- **Parâmetros**: identificador da unidade e porcentagem obtida
- **Evento de retorno:** evRegistraPorcentagemConclusaoAtividade

Observação:

- *Informar no parâmetro porcentagem, um valor numérico de 0 a 100.*

Exemplos:

Chamada do método:

```
//Armazena a última página acessada por um usuário
onClick="API.registrarPorcentagemConclusaoUnidade('unidade_01','80');"
```

Retorno do método:

```
// Adicionando listener
window.addEventListener("evRegistraPorcentagemConclusaoAtividade",
resultadoRegistrarPorcentagemConclusaoUnidade, false);
function resultadoRegistrarPorcentagemConclusaoUnidade(evento){
    console.log(evento.detail);
}
```

```
//Dados retornados quando a ação é executada com sucesso
{
status: 200,
data: "Operação efetuada com sucesso."
}
```

//Dados retornados quando o usuário não for um cursista

{

status: ,

data: {

codigo: "ME34_01",

mensagem: "Não foi possível armazenar as informações solicitadas, pois o usuário não é um cursista ativo do curso."

}

```
//Dados retornados quando não forem informados todos os parâmetros obrigatórios, não for identificada a unidade ou o valor da porcentagem estiver fora do padrão.
{
status: 412,
data: {
       codigo: "ME34",
       mensagem: "Não foi possível armazenar as informações solicitadas, pois os dados enviados estão incorretos."
       }
}
```

5.5.5 - Atividade

Para atividades, são disponibilizados métodos que permitem obter e armazenar as respostas de um usuário, para uma atividade de uma unidade do curso e também obter os dados da atividade nas configurações de avaliação de uma determinada unidade em um curso.

Observação:

- *Se, ao armazenar as respostas de uma atividade, for identificado que o usuário concluiu todos os requisitos para conclusão da unidade, o sistema irá, automaticamente, concluir a participação do cursista na unidade. Podendo também, aprovar o cursista no curso, caso*

ele tenha concluído todos os requisitos para conclusão do curso e se o curso estiver configurado para avaliar automaticamente os cursistas.

1 - Obter dados de configuração de uma atividade

- **Descrição:** obtém os dados da atividade, nas configurações de conclusão de uma determinada unidade em um curso;
- **Método:** obterConfiguracaoAtividade(identificadorAtividade)
- **Parâmetros**: identificador da atividade
- **Evento de retorno:** evObtemConfiguracaoAtividade

Exemplos:

Chamada do método:

```
//Obtém os dados de configuração de uma atividade
onClick="API.obterConfiguracaoAtividade('atividade_1_uni_1');"
```

Retorno do método:

```
// Adicionando listener
window.addEventListener("evObtemConfiguracaoAtividade",
resultadoObterConfiguracaoAtividade, false);
function resultadoObterConfiguracaoAtividade(evento){
    console.log(evento.detail);
}
```

```
//Dados retornados quando a ação é executada com sucesso
{
status: 200,
data: {
id: 15,
peso: 1,
considerarNaMediaAtividade:: true,
ativo: true,
atividade: {
       id: 40,
       identificador: "atividade_1_uni_1",
       }
}
}
```

//Dados retornados quando o usuário não for um cursista

{

status: ,

data: {

codigo: "ME34_01",

mensagem: "Não foi possível armazenar as informações solicitadas, pois o usuário não é um cursista ativo do curso."

}

```
//Dados retornados quando não foram informados todos os parâmetros obrigatórios ou a atividade não for identificada.
{
status: 412,
data: {
       codigo: "ME37",
       mensagem: "Não foram encontrados dados com os parâmetros informados."
       }
}
```

2 - Armazenar as respostas enviadas para uma atividade

- **Descrição:** armazena as respostas que um cursista enviou para uma determinada atividade do conteúdo de um curso;
- **Método:** registrarRespostaAtividade(atividade)
- **Parâmetros**: json com o objeto atividade (verifica exemplo abaixo)
- **Evento de retorno:** evRegistraRespostaAtividade

Exemplos:

Chamada do método:

```
//Armazena a última página acessada por um usuário
onClick="API.registrarRespostaAtividade({
   "identificador": "atividade_1_uni_1",
   "questoes": [{
      "identificador": "Q1_A1_U1_Enumere",
      "gabaritos": [
         { "chave": "1", "valor": "3"},
         { "chave": "2", "valor": "5"},
         { "chave": "3", "valor": "1"},
         {"chave": "4", "valor": "2" },
         {"chave": "5", "valor": "4" }
      ]
   }]
});"
```

Retorno do método:

```
// Adicionando listener
window.addEventListener("evRegistraRespostaAtividade",
resultadoRegistrarRespostaAtividade, false);
function resultadoRegistrarRespostaAtividade(evento){
    console.log(evento.detail);
}
```

```
//Dados retornados quando a ação é executada com sucesso
{
status: 200,
data: "Operação efetuada com sucesso."
}
```

//Dados retornados quando o usuário não for um cursista

{

status: ,

data: {

codigo: "ME34_01",

mensagem: "Não foi possível armazenar as informações solicitadas, pois o usuário não é um cursista ativo do curso."

}

```
//Dados retornados quando não forem informados todos os parâmetros obrigatórios ou a atividade não for identificada
{
status: 412,
data: {
      codigo: "ME34",
      mensagem: "Não foi possível armazenar as informações solicitadas, pois os dados enviados estão incorretos."
      }
}
```

3 - Obter respostas armazenadas para uma atividade

- **Descrição:** obtém as respostas da última tentativa que um cursista realizou para uma determinada atividade do conteúdo de um curso;
- **Método:** obterRespostaAtividade(identificadorAtividade)
- **Parâmetros**: identificador da atividade

- **Evento de retorno:** evObtemDadosRespostaAtividade

Exemplos:

Chamada do método:

```
//Obtém a resposta enviada por um usuário para uma atividade
onClick="API.obterRespostaAtividade("atividade_1_uni_1");"
```

Retorno do método:

```
// Adicionando listener
window.addEventListener("evObtemDadosRespostaAtividade",
resultadoObtemRespostaAtividade, false);
function resultadoObtemRespostaAtividade(evento){
    console.log(evento.detail);
}
```

```
//Dados retornados quando a ação é executada com sucesso
{
status: 200,
data:{
   id: 40,
   nota: 10,
   data: "2018-05-22T16:30:49",
   questoesUsuario:[{
      id: 80,
      questao:{
      identificador: "Q1_A1_U1_Enumere",
      gabaritos: [],
      mapaAtributos: {}
      },
      respostas: [
         {
            id: 100,
            chave: "1",
            valor: "3"
         },{
            id: 101,
            chave: "2"
            valor: "5"
         },{
            id: 102,
            chave: "3",
            valor: "1"
         },{
            id: 103,
            chave: "4",
            valor: "2"
```

```
         },{
            id: 104,
            chave: "5",
            valor: "4"
         }
      ],
      mapaAtributos: {}
   }],
   mapaAtributos: {
      quantidadeRespostasRegistradas: 1
   }
}}
```

//Dados retornados quando o usuário não for um cursista

{

status: ,

data: {

codigo: "ME34_01",

mensagem: "Não foi possível armazenar as informações solicitadas, pois o usuário não é um cursista ativo do curso."

}

```
//Dados retornados quando não forem informados todos os parâmetros obrigatórios ou a atividade não for identificada
{
status: 412,
data: {
       codigo: "ME34",
       mensagem: "Não foi possível armazenar as informações solicitadas, pois os dados enviados estão incorretos."
       }
}
```

5.5.6 - Conclusão

Permite obter a situação de conclusão de um aluno em uma unidade do curso e também no curso. Possibilitando, por exemplo, realizar alguma marcação no conteúdo informando que ele já concluiu determinada unidade, disponibilizar link para download do certificado no conteúdo ou até mesmo, bloquear a realização de alguma atividade para alunos que já concluíram a unidade ou o curso.

1 - Obter dados de conclusão da unidade

- **Descrição:** obtém os dados de conclusão do usuário em uma unidade.
- **Método:** obterDadosConclusaoUnidade(identificadorUnidade)
- **Parâmetros**: identificador da unidade
- **Evento de retorno:** evObtemDadosConclusaoUnidade

Exemplos:

Chamada do método:

```
//Obtém os dados de conclusão de um cursista para uma unidade
onClick="API.obterDadosConclusaoUnidade('unidade_01');"
```

Retorno do método:

```
// Adicionando listener
window.addEventListener("evObtemDadosConclusaoUnidade",
resultadoObterConclusaoUnidade, false);
function resultadoObterConclusaoUnidade(evento){
    console.log(evento.detail);
}
```

```
{
status: 200,
data: {
   id: 15,
   concluido: true,
   porcentagemConclusaoo: 100,
   quantidadeAcessos: 15,
   unidade:{
      id: 20,
      identificador: "unidade_01",
      situacao: "ATIVO",
      topicos: [],
      atividades: [],
      permitePorcentagem: true,
      mapaAtributos: {}
   }
   media: 9,
   mapaAtributos: {}
}}
```

//Dados retornados quando o usuário não for um cursista

{

status: ,

data: {

codigo: "ME34_01",

mensagem: "Não foi possível armazenar as informações solicitadas, pois o usuário não é um cursista ativo do curso."

}

```
//Dados retornados quando não foram informados todos os parâmetros obrigatórios ou a unidade não for identificada.
{
status: 412,
data: {
       codigo: "ME37",
       mensagem: "Não foram encontrados dados com os parâmetros informados."
       }
}
```

2 - Obter dados de conclusão do curso

- **Descrição:** obtém os dados de conclusão do usuário em um curso.
- **Método:** obterDadosConclusaoCurso()
- **Parâmetros**: sem parâmetros
- **Evento de retorno:** evObtemDadosDadosConclusaoCurso

Exemplos:

Chamada do método:

```
//Obtém os dados de conclusão de um cursista para o curso
onClick="API.obterDadosConclusaoCurso();"
```

Retorno do método:

```
// Adicionando listener
window.addEventListener("evObtemDadosDadosConclusaoCurso",
resultadoObtemDadosConclusaoCurso, false);
function resultadoObtemDadosConclusaoCurso(evento){
    console.log(evento.detail);
}
```

```
//Dados retornados quando a ação é executada com sucesso
{
status: 200,
```

```
data:{
   mediaFinal: 9.5,
   caminhoCertificado: "http://www.labtime.ufg.br/ava-mec-ws/documento/401/obtem",
   situacaoConclusao: "APROVADO"
}}
```

//Dados retornados quando o usuário não for um cursista

{

status: ,

data: {

codigo: "ME34_01",

mensagem: "Não foi possível armazenar as informações solicitadas, pois o usuário não é um cursista ativo do curso."

}

5.5.7 - Contraste

Permite verificar se o usuário está utilizando alto contraste no AVA MEC, possibilitando apresentar automaticamente o conteúdo em alto contraste para o usuário.

1 - Obter dados de configuração de contraste

- **Descrição:** verifica se o usuário está utilizando alto contraste no AVA MEC.
- **Método:** obterStatusContraste()
- **Parâmetros**: sem parâmetros
- **Evento de retorno:** evObtemStatusContraste

Exemplos:

Chamada do método:

```
//Obtém os dados de conclusão de um cursista para uma unidade
onClick="API.obterDadosConclusaoUnidade('unidade_01');"
```

Retorno do método:

```
// Adicionando listener
window.addEventListener("evObtemStatusContraste", resultadoObterStatusContraste,
false);
function resultadoObterStatusContraste(evento){
    console.log(evento.detail);
```

```
}
```

```
//Dados retornados quando a ação é executada com sucesso
{
   status: 200,
   data: true
}
```