

Projeto Final em Engenharia Informática

Bloco Moodle

BLOCO NAVEGAÇÃO MOODLE

<<Nuno Guerra 1900245>

< Prof. Vítor Rocio + Dr. Bruno Tavares>

<Lisboa, 29-6-22>

Índice

Introdução	3
Organização do Relatório.....	3
1. Problema a resolver e objetivos para a sua resolução	4
1.1. Problema a resolver	4
1.2. Proposta para o trabalho.....	4
1.3. Objetivos e resultados esperados	4
1.4. Calendarização do projeto	5
2. Moodle	6
2.1 O que é o MOODLE?	6
2.1.1 Desenvolvimento moodle	6
2.1.2 Plugin/Bloco Moodle.....	7
2.2. Web Server	7
Desenvolvimento de blocos.....	9
3.1. Course overview on campus	10
3.2 COURSE OVERVIEW UAb.....	10
3.2.1 Estrutura do bloco	11
3.2.2 Descrição dos ficheiros desenvolvidos.....	12
3.3 Data manipulation API.....	14
3.3.1 Conceitos gerais.....	15
3.4 Enrolment Api.....	16
3.5 Templates	17
3.6 Funcionalidade mensagens publicadas após último login.....	17
3.6.1 Programação da funcionalidade mensagens não lidas.....	18
3.7 Funcionalidade lista de ucs inscritas.....	22
3.7.1 Programação da funcionalidade lista de ucs inscritas.....	22
3.8 Otimizar a consulta à base de dados	23
3.8 Manual de utilização	24
Conclusões e Trabalho Futuro	27
4.1. Resultados principais	27
4.2. Limitações	28
4.3. Desenvolvimento futuro e melhorias.....	28

Bibliografia.....	29
Anexos	30
6.1 Ficheiro course_overviewua.php	30
6.2 Ficheiro acess.php	32
6.3 Ficheiro block_course_overviewua.php	32
6.4 Ficheiro block_course_overviewuab.php	34
6.5 Ficheiro edit_form.php	38
6.6 Ficheiro version.php	39

Índice de figuras

Fig.1 Ambiente de Desenvolvimento Moodle	7
Fig.2 Configuração do Servidor	8
Fig.3 Ferramenta front-end de gestão da base de dados.....	8
Fig.4 Página Moodle em funcionamento localmente.....	9
Fig.5 Tabela descritiva dos ficheiros utilizados	11
Fig.6 Relação tabelas – Consulta (SQL).....	19
Fig.7 Relação tabelas – Consulta (SQL)	19
Fig.8 Relação tabelas – Consulta (SQL).....	20
Fig.9 Bloco em funcionamento (Sem mensagens novas).....	21
Fig.10 Bloco em funcionamento (Com mensagens novas).....	22
Fig.11 Bloco em funcionamento com informação sobre UC em que está inscrito	23
Fig.12 Consulta com o uso do loop foreach	24
Fig.13 Consulta com o uso da função array_column ().....	24
Fig.14 Instalação de plugin.....	25
Fig.15 Upload ficheiro Zip	25
Fig.16 Instalar plugin	25
Fig.17 Validação bem-sucedida.....	26
Fig.18 Concluir Instalação Manualmente	26

Introdução

O presente relatório descreve o trabalho desenvolvido no âmbito da unidade curricular Projeto de Engenharia Informática do curso Licenciatura em Engenharia Informática da Universidade Aberta.

Esta unidade curricular implica que os estudantes tentem aplicar os conhecimentos que foram adquirindo durante a frequência do curso, elaborando um trabalho de desenvolvimento de média dimensão com aplicações práticas, concretas e de reconhecida utilidade.

Nesse sentido o projeto escolhido foi o “Bloco de navegação expedita na plataforma Moodle” proposto pela docência e que consiste basicamente no desenvolvimento de um bloco de navegação Moodle que tem como objetivo repor algumas das funcionalidades do bloco descontinuado *course overview on campus*.

Tendo em conta que a plataforma Moodle é usada pela Universidade Aberta, entendi que era a motivação necessária para a criação deste bloco pelo facto de ser uma ferramenta útil para uso de colegas futuros e presentes.

Organização do relatório

Este relatório começa por descrever detalhadamente o problema a resolver, os objetivos propostos para a sua resolução, de seguida a descrição da implementação/dificuldades encontradas, quais os objetivos alcançados, uma pequena reflexão sobre o trabalho desenvolvido e finalmente o que poderá ser desenvolvido no futuro.

Capítulo 1

1. Problema a resolver e objetivos para a sua resolução

Este projeto consiste no desenvolvimento de um bloco de navegação Moodle “course overview on campus” que foi descontinuado e que tem o objetivo de repor algumas dessas funcionalidades.

1.1. Problema a resolver

A plataforma de elearning Moodle pode alojar uma grande quantidade de cursos e unidades curriculares, e quando os utilizadores participam em vários destes espaços, pode tornar difícil a gestão do acompanhamento e intervenção em todos esses espaços. Os utilizadores sentem necessidade de saber de forma expedita em que espaços há novas intervenções, atividades calendarizadas ou que tenham necessidade de alguma intervenção, sem terem de percorrê-los todos um a um. O bloco “course overview on campus” resolvia parte destes problemas, mas tendo sido descontinuado, deixou de existir alternativa com funcionalidade semelhante.

1.2. Proposta para o trabalho

Este trabalho consiste no desenvolvimento/adaptação de um bloco para a plataforma Moodle que reponha alguma da funcionalidade do bloco “Course overview on campus”, a funcionalidades a serem implementadas irão consistir em mostrar informação sobre:

- ➡ Mensagens publicadas após último login.
- ➡ Lista de UCs inscritas.
- ➡ Otimizar as consultas à base de dados por forma a que o bloco seja rápido a carregar.

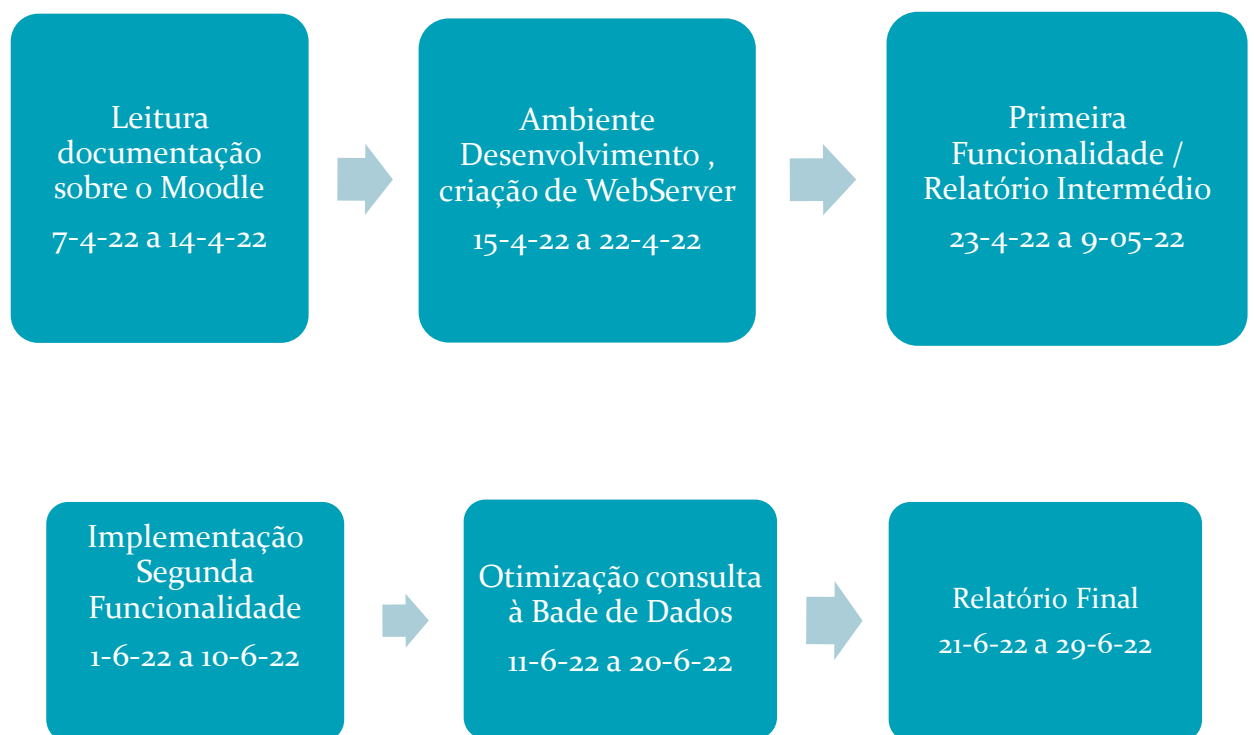
1.3. Objetivos e resultados esperados

O objetivo final será ter um bloco de navegação moodle a funcionar corretamente e de acordo com as funcionalidades acima descritas, de forma a ir de encontro às necessidades mencionadas para que seja uma ajuda adicional na consulta mais célere da informação pretendida fazendo com que o bloco criado seja uma mais valia reconhecida por todos os utilizadores.

Numa primeira fase a implementação do bloco seria feita apenas com um webserver instalado localmente, mas com a perspectiva de poder ser também experimentado no ambiente de qualidade da UAb.

Com este trabalho pretendi melhorar a experiência utilizador/tecnologia para que seja uma experiência mais proveitosa no cumprimento dos objetivos de todos os intervenientes.

1.4. Calendarização do projeto



Capítulo 2

2. Moodle

2.1 O que é o moodle?

MOODLE é um software livre, de apoio à aprendizagem, executado num ambiente virtual. É um sistema de gestão da aprendizagem em trabalho colaborativo, acessível através da Internet ou de rede local. O programa é disponibilizado livremente sob a licença de software livre GNU Public License) e pode ser instalado em diversos ambientes (Unix, Linux, Windows, Mac OS) desde que os mesmos consigam executar a linguagem PHP. Como base de dados podem ser utilizados MySQL, PostgreSQL, Oracle, Access, Interbase ou qualquer outra acessível via ODBC.

É desenvolvido em colaboração por uma comunidade virtual, que reúne programadores e desenvolvedores de software livre, administradores de sistemas, professores, designers e utilizadores de todo o mundo. Evolui constantemente adequando-se às necessidades dos seus utilizadores. Constitui-se num software intuitivo e fácil de utilizar, que tanto pode dar origem a uma página de um único professor/formador, como à página de uma Universidade, com dezenas de milhares de alunos/utilizadores.

(<https://pt.wikipedia.org/wiki/Moodle>)

2.1.1 Desenvolvimento moodle

Para que fosse possível efetuar algum tipo de desenvolvimento na plataforma Moodle, neste caso o desenvolvimento de um plugin na forma de um bloco, foram seguidos os seguintes passos necessários:

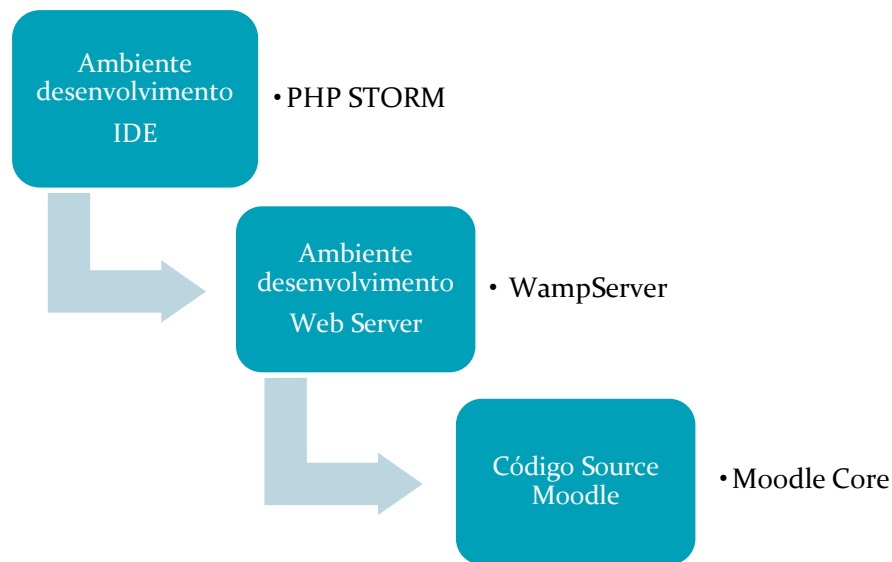


Fig. 1 - Ambiente desenvolvimento Moodle

Este desenvolvimento foi efetuado usando essencialmente as linguagens de programação PHP e SQL, acedendo à base de dados padrão do Moodle.

2.1.2 Plugin/Bloco moodle

O Moodle é um sistema de gestão de aprendizagem poderoso e que faz uso de um conjunto de ferramentas flexíveis (plugins) para alcançar um dos seus princípios centrais que é a sua extensibilidade, e isso é alcançado principalmente através do desenvolvimento de plugins, permitindo assim que os utilizadores do Moodle estendam os recursos do site. O Moodle tem centenas de plugins disponíveis que estendem os recursos da funcionalidade principal do Moodle, tais como blocos.

Um bloco é um tipo de plugin que consiste basicamente numa pequena tela de ferramentas que mostram informações e recursos complementares em diferentes partes do Moodle.

2.2. Web Server

O ambiente de desenvolvimento escolhido para a criação do web server foi o software “WAMPSEVER” (Fig. 2). Este ambiente de desenvolvimento permite que se crie aplicativos da web com Apache2, PHP e um banco de dados MySQL automaticamente, o que tendo em conta a complexidade do projeto e o tempo disponível para o fazer, foi a melhor forma que encontrei de poupar tempo, ao invés de ter de efetuar a configuração do mesmo manualmente sem perder o objetivo final proposta. Para além disso, este

software tem incorporado uma ferramenta ” front end “ (Fig. 3) que permite que a gestão dos bancos de dados seja feita mais facilmente.

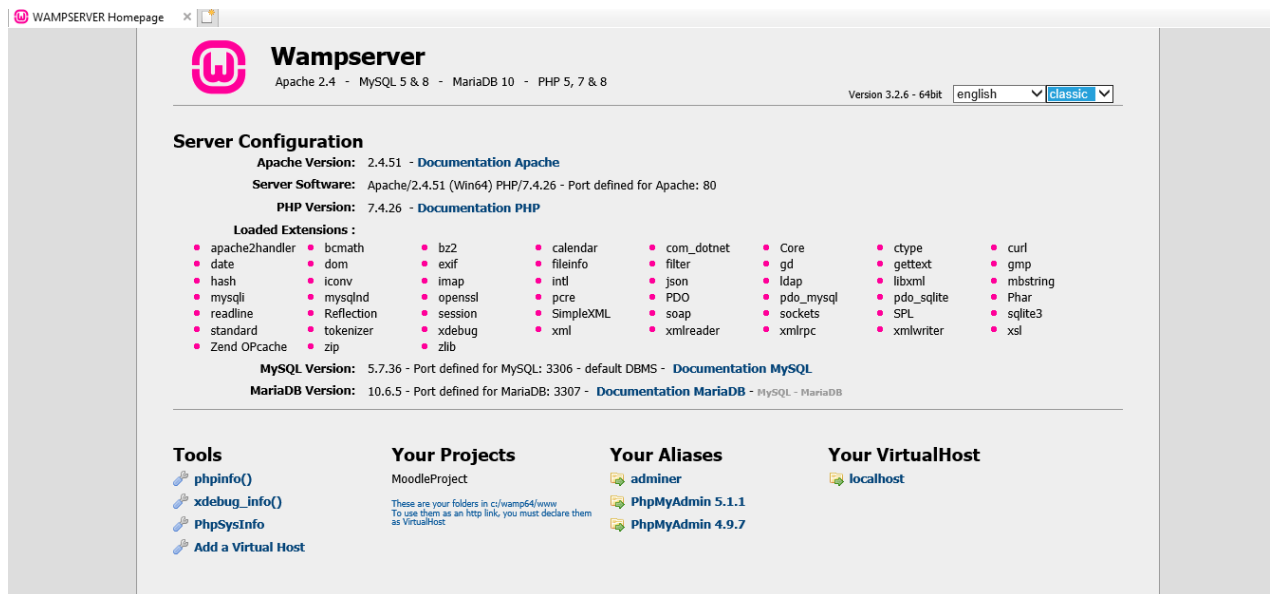


Fig. 2 - Configuração do Servidor

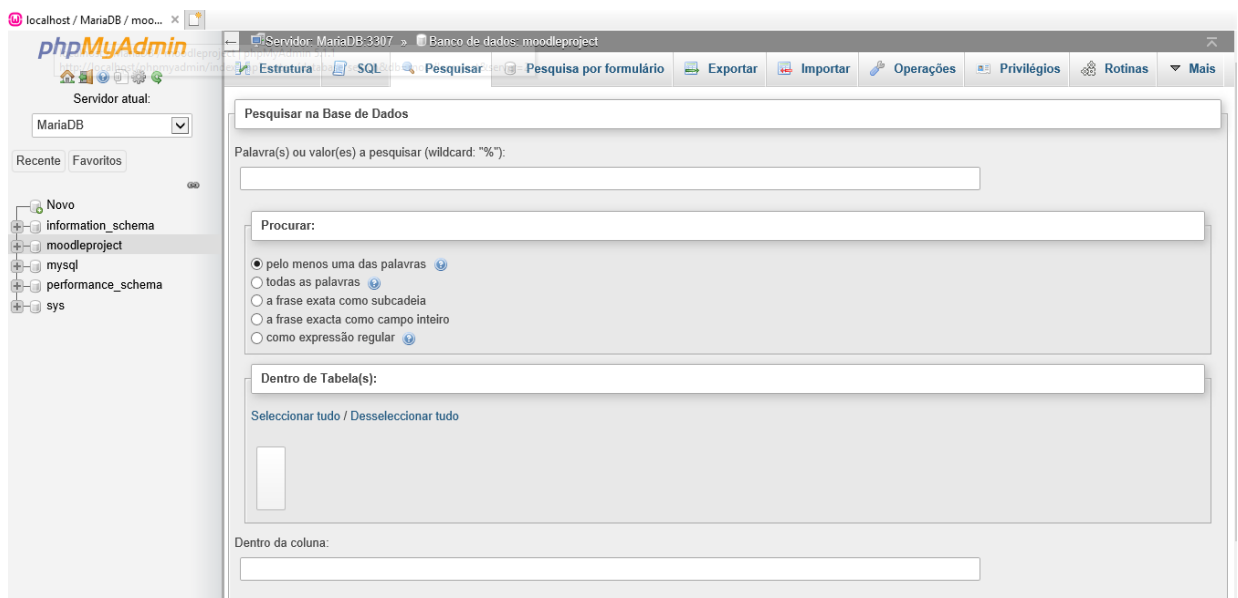


Fig. 3 - Ferramenta front-end de gestão da BD

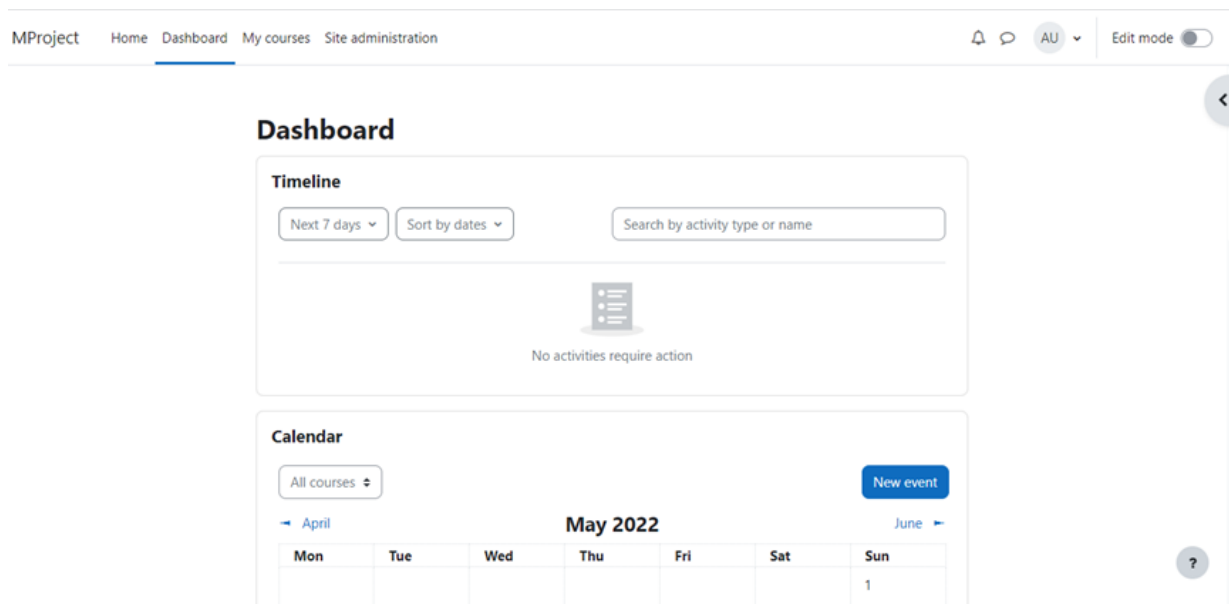


Fig. 4 – Página Moodle em funcionamento localmente

Capítulo 3

Desenvolvimento de blocos

O desenvolvimento de blocos pode ser feito de raiz ou reutilizando algum tipo de bloco já feito. Como normalmente esse desenvolvimento segue uma estrutura padrão, essa reutilização é mais simples visto que apenas será necessário adaptar às necessidades específicas das funcionalidades que se pretendem que o bloco obtenha. No entanto e embora este projeto consista na reposição de algumas funcionalidades de um bloco já existente, mas descontinuado, optei por criar de raiz para perceber exatamente como se constrói um bloco e quais as suas dificuldades/particularidades e assim entender de forma clara todo o potencial inerente.

3.1. COURSE OVERVIEW ON CAMPUS

Este bloco descontinuado consistia no fornecimento de todas as funcionalidades do bloco Course Overview, e ainda fornecia filtros adicionais (categoria do curso, período do curso e professores do curso) para serem usados em ambientes universitários, bem como a possibilidade de ocultar cursos e notícias de cursos da lista de cursos.

Basicamente consistia em 8 opções:

1. Geral

Alterar o título do bloco que é mostrado na visualização do bloco

2. Lista de visão geral do curso

Alterar a aparência da lista de visão geral do curso, especialmente se o nome completo ou o nome curto do curso deve ser exibido nas entradas da lista de visão geral do curso. Além disso, pode-se estilizar a exibição de alguns metadados numa segunda linha da entrada da lista de visão geral do curso e pode-se definir os cursos nos quais o utilizador tem uma função de professor, se sim então são listados primeiro na lista de visão geral do curso.

3. Ocultar cursos

Habilitar ou desabilitar o sistema para ocultar cursos da lista de visão geral do curso.

4. Funções do professor

Definir quais funções num curso serão exibidas para além do nome do curso como professor e listadas no filtro de professor.

5. Filtro de categoria parente

Ativar e configurar um filtro que permite que os utilizadores filtrem os cursos por categoria principal.

6. Filtro de categoria de nível superior

Ativar e configurar um filtro que permite que os utilizadores filtrem os cursos por categoria de nível superior.

7. Filtro do professor

Ativar e configurar um filtro que permite que os utilizadores filtrem os cursos por professor.

8. Filtro de período

Ativar e configurar um filtro que permite que os utilizadores filtrem os cursos por período.

Durante o desenvolvimento do Moodle, foram adicionados vários recursos ao core do Course Overview e ao core do Moodle que entraram em conflito com a funcionalidade do Course overview on campus. Foi então decidido desconsiderar várias funcionalidades do Moodle para este bloco.

O bloco Course Overview permitia que os alunos verificassem o progresso do curso e visualizassem os próximos prazos. Permitia também que os professores vissem facilmente as atividades a precisar de classificação.

3.2 COURSE OVERVIEW UAb

Este bloco com o nome Course Overview UAb foi então desenvolvido e repõe as funcionalidades já acima descritas em 1.2. Esse desenvolvimento é feito através da adição deste mesmo bloco ao código fonte Moodle no diretório “blocks”.

3.2.1 Estrutura do bloco

Os plugins de blocos estão localizados no diretório /blocks. Cada plug-in está num subdiretório separado e consiste em vários ficheiros obrigatórios e quaisquer outros ficheiros que sejam necessários ao desenvolvimento do bloco.

Neste caso o diretório usado é o /blocks/course_overview_uab/.

Descrição	Ficheiro	Obrigatório
Classe de definição de bloco	block_course_overviewuab.php	Sim
Recursos de plug-in	access.php	Sim
Arquivos de idioma	block_course_overviewua.php	Sim
Metadados da versão	version.php	Sim
Classe de formulário de edição de bloco	edit_form.php	Não
Classe base do bloco	course_overviewua.php	Não
Arquivo folha de estilo	styles.css	Não
Instruções	README.md	Não

Fig. 5 – Tabela descritiva dos ficheiros utilizados

3.2.2 Descrição dos ficheiros desenvolvidos

- ***block_course_overviewuab.php***

Este ficheiro contém a definição de classe para o bloco e é usado tanto para o gerir como um plugin quanto para o renderizar na tela. De referir que método *init* é essencial para todos os blocos e a sua finalidade é fornecer valores a qualquer variável membro de classe que precise ser instanciada, mais sobre este método no seguimento deste tópico.

- ***access.php***

Contém a configuração inicial para as regras de controle de acesso de um plugin. O controle de acesso é feito no Moodle pelo uso de Funções e Recursos.

- ***block_course_overviewua.php***

Cada plugin deve definir um conjunto de strings de idioma com, no mínimo, uma tradução em inglês. São especificados no diretório lang/en do plug-in num arquivo com o nome do plug-in. No contexto deste projeto apenas foi utilizado a linguagem inglesa.

- ***version.php***

Contém metadados sobre o plugin. É usado durante a instalação e atualização do plugin. Este ficheiros contém metadados usados para descrever o plug-in e inclui informações como: o número da versão, uma lista de dependências, a versão mínima do Moodle necessária, maturidade do plugin

- ***edit_form.php***

Este ficheiro só é necessário se o plugin tiver um formulário de configuração específico. Não é necessário para a maioria dos plugins. Podemos estender este formulário de configuração e adicionar campos de preferências personalizados, para que os utilizadores possam adequar melhor o nosso bloco a uma determinada tarefa ou página.

- ***course_overviewua.php***

Classe principal que estende a classe de bloco principal e que contém as consultas à base de dados.

“Block base class API methods”

Da classe base *block_base* derivam todas as outras classes do bloco, e também todos os métodos que podem ser editados por desenvolvedores de blocos. No entanto, existem métodos que não devem ser editados. Os métodos são divididos em três categorias:

- Métodos que podem ser usados e editados;
- Métodos que não podem ser editados, mas podem ser usados;
- Métodos internos que não devem ser usados nem editados.

Tendo em conta o contexto deste projeto alguns métodos foram utilizados no desenvolvimento do bloco, e pertencem à categoria dos métodos que podem ser usados e editados. Esses métodos são :

• `init ()`

O método `init` é chamado antes que o bloco seja exibido. É essencial para todos os blocos e sua finalidade é fornecer valores a qualquer variável membro de classe que precise ser instanciada. No entanto, ele é chamado antes de `$this->config` ser definido, se o plugin precisar de algum valor de configuração para definir atributos globais como o título do bloco, isso deve ser feito no método de especialização.

• `specialization ()`

Esta função é chamada na subclasse logo após o carregamento de uma instância. É usado para personalizar o título e outros atributos do bloco dependendo do tipo de página, contexto, configuração, etc.

• `get_content ()`

Método para que este bloco realmente exiba algo na tela, precisamos adicionar mais um método à nossa classe (antes da chave de fecho final no ficheiro) dentro do script `block_course_overviewuab.php`.

O `get_content` pode ser chamado várias vezes durante a renderização da página. Para evitar que esta classe o calcule todas as vezes, este plugin verifica se `$this->content` já está definido no início do método.

• `applicable_formats ()`

Blocos podem ser adicionados a qualquer tipo de página. No entanto, alguns blocos podem ser exibidos apenas em determinados tipos de página. Este método é usado para definir os tipos de página em que o bloco pode ser exibido.

- `instance_allow_multiple()`

Por padrão, apenas uma instância de cada plug-in de bloco pode ser adicionada a uma página. No entanto, se o plug-in permitir várias instâncias, pode-se substituir o método `instance_allow_multiple` para “true” que foi o caso.

A classe *block_base* também possui algumas variáveis membro padrão que são manipuladas pelos métodos a que pertence e tendo em conta que todos os blocos devem fornecer uma classe principal que estende a classe de bloco principal, assim que a instância do bloco é criada, existem vários atributos `$this` que podem e que foram utilizados:

\$this->config. A configuração da instância do bloco. Por padrão é um objeto vazio, mas se o bloco tiver um arquivo `edit_form.php`, será um objeto com os dados do formulário, que é o caso.

\$this->content. Esta variável contém todo o conteúdo real que é exibido dentro de cada bloco. Os valores válidos são NULL ou um objeto da classe `stdClass`, que deve ter variáveis de membro específicas dependendo da classe base do bloco estendido.

\$this->page O objeto de página em que o bloco está sendo exibido.

\$this->context O objeto de contexto no qual o bloco está sendo exibido.

\$this->title O título do bloco.

O ficheiro principal de desenvolvimento *block_course_overviewuab.php* faz uso destes métodos e das variáveis membro padrão descritos.

3.3 Data manipulation api

Esta funcionalidade core descreve as funções disponíveis de forma a aceder aos dados no banco de dados Moodle. Estas funções devem ser usadas exclusivamente para recuperar ou modificar o conteúdo do banco de dados pelo facto de fornecerem um alto nível de abstração e garantir que a manipulação do banco de dados funcione em diferentes RDBMS (relational database management system). Existem inúmeras funções que pertencem à manipulação de dados, neste tópico apenas serão descritas as funções usadas neste projeto.

3.3.1 Conceitos gerais

Objeto de banco de dados:

- A API de manipulação de dados é exposta por meio de métodos públicos do objeto \$DB.
- O núcleo do Moodle encarrega-se de configurar a conexão com o banco de dados de acordo com os valores especificados no arquivo config.php principal.
- O objeto global \$DB é uma instância da classe moodle_database. É instanciado automaticamente durante a configuração do bootstrap, ou seja, como parte da inclusão do arquivo config.php principal.
- O objeto DB está disponível no escopo global logo após incluir o arquivo config.php:

Prefixo da tabela

- A maioria das instalações do Moodle usa um prefixo para todas as tabelas do banco de dados, como mdl_. Este prefixo NÃO deve ser usado no próprio código porque todos os parâmetros \$table nas funções devem ser o nome da tabela sem prefixos.
- Em consultas SQL personalizadas, os nomes das tabelas devem ser colocados entre chavetas. Serão automaticamente convertidos para o nome real da tabela prefixada.

Placeholders

- Todos os parâmetros \$params nas funções são arrays de valores usados para preencher espaços reservados em instruções SQL.
- Os espaços reservados ajudam a evitar problemas com injeção de SQL e/ou citações inválidas em consultas SQL. Facilitam o código seguro e compatível com cross-db.
- Há suporte para dois tipos de espaços reservados - pontos de interrogação (SQL_PARAMS_QM) e espaços reservados nomeados (SQL_PARAMS_NAMED).
- Os parâmetros nomeados devem ser exclusivos, mesmo que o valor passado seja o mesmo.

É exemplo dessa utilização o método “get_discussion_count () “, explicado com detalhe no tópico 3.6.1. .

get_records_sql ()

Retorna uma lista de registos como uma matriz de objetos usando uma consulta SELECT personalizada.

Sendo este um método que retorna um array de objetos, a matriz é indexada pela primeira coluna dos campos retornados pela consulta. Assim, para garantir dados consistentes, pareceu-me ser a melhor forma e por sua vez uma prática recomendada de forma a garantir que a consulta incluía uma "coluna id" como o primeiro campo.

3.4 Enrolment api

O que é o enrolment (inscrição)?

Os utilizadores inscritos podem participar integralmente num curso. A inscrição de utilizador ativo permite que o utilizador entre no curso. Somente utilizadores inscritos podem ser membros do grupo. As notas são armazenadas apenas para utilizadores inscritos.

A API de inscrição dá acesso aos métodos de inscrição e também às instâncias de plug-ins de inscrição. Existe, no entanto, uma diferença entre inscrição de utilizador e atribuição de função. Os utilizadores matriculados num curso possuem pelo menos um registo na tabela `user_enrolments`. Esta tabela tem a relação entre cursos e utilizadores por meio de uma instância de plug-in de inscrição. No entanto, `user_enrolments` não contém informações sobre a função do utilizador no curso, apenas informações sobre:

- Instância de plug-in de inscrição
- Status da inscrição (ativa ou suspensa).
- Datas de início e término da inscrição.

As atribuições de funções específicas estão relacionadas ao contexto, não apenas ao curso (já que as atividades e outras páginas podem usar suas próprias). As funções específicas de um utilizador são armazenadas na tabela `role_assignments`. Esta tabela armazena:

- O *user role id* no *context*
- O *component item* que atribuiu a função. No caso de um curso regular, o *component* é o nome do plugin de inscrição e o *item id* é a instância específica do plugin.

Funções API:

No desenvolvimento deste projeto apenas foi usado a função *enrol_get_my_courses ()*. Esta função retorna a lista de cursos em que o utilizador está matriculado e pode aceder, mais sobre o uso desta função no tópico seguinte.

3.5 Templates

O Moodle usa o sistema de templates “Mustache” para renderizar a maior parte da sua saída HTML. Os modelos são definidos como texto simples, que normalmente inclui HTML e uma variedade de marcas “Mustache” e espaços reservados. Os placeholders do “Mustache” são substituídos por valores reais durante a renderização da página. Os modelos de “Mustache” podem ser renderizados tanto no lado do servidor em PHP quanto no lado do cliente usando JavaScript. Os temas podem substituir os modelos definidos noutros componentes, se necessário.

No contexto deste projeto foi aproveitado o facto de existir uma pré configuração no que toca aos blocos. O sistema de modelo “Mustache” que pode ser estendido através dos chamados pragmas de “Mustache”. Pragmas são extensões não padronizadas para a especificação do “Mustache”. O Moodle 3.0 e superior tem o pragma BLOCKS instalado e ativado. A extensão permite que se defina um modelo pai com blocos substituíveis. A pasta *templates* contem dois ficheiros *courses* e *discussion*, ambos formatados para corresponder ao desenvolvimento efetuado. Existe também a pasta *pix* que contém vários ficheiros que funcionam como um auxiliar de imagem de ícones.

Foi também usado o método de renderização em PHP *render_from_template ()* para renderizar os dados de contexto fornecidos com o modelo.

3.6 Funcionalidade mensagens publicadas após último login

Funcionamento do bloco que apresenta um “alerta” de mensagens não lidas após o último login. O desenvolvimento desta funcionalidade assentou na criação de algumas funções que acedem às tabelas através de consultas em SQL, usando as funções existentes do core do moodle e que retornam o resultado em forma de objeto.

3.6.1 Programação da funcionalidade mensagens não lidas

A programação desta funcionalidade foi executada em basicamente dois ficheiros. O ficheiro `course_overviewua.php` e o ficheiro `block_course_overviewua.php`.

course_overviewua.php:

Este ficheiro consiste na chamada através do namespace do bloco principal e a invocação do “`defined('MOODLE_INTERNAL') || die ();`”. Este `die` switch é colocado em todos os ficheiros `php` que não devem ser acedidos diretamente. Ou seja, impede que alguém digite o URL para um desses ficheiros e acidentalmente execute algum código ou veja um erro estranho. O restante ficheiro contém apenas duas funções, a função `get_discussion_forum ()` e a função `get_discussion_count ()`, e recebe a informação pretendida como objeto.

get_discussion_forum ():

Esta função faz uso do acesso à base de dados através do objeto global `$DB` da classe `core` do moodle `moodle_database` e aos utilizadores através da variável global `$USER`, de seguida é separada em duas partes. Na primeira parte faz uma consulta (SQL) para obter e guardar os utilizadores que estão inscritos em determinados cursos usando as tabelas da fig. 6. Na segunda parte faz uma consulta (SQL) para obter os fóruns pertencentes aos cursos onde existem utilizadores inscritos fig. 7. E por fim retorna o resultado pretendido.

get_discussion_count ():

Esta função inicializa um contador que soma as mensagens criadas pelos utilizadores inscritos desde o último login e retorna esse valor de mensagens não lidas por tópicos fig.10. Faz uso do acesso à base de dados e aos utilizadores de forma já descrita anteriormente. Aqui é usado o placeholder `$params` já descrito anteriormente onde verifica a condição de “`lastlogin`”, ou seja, se o utilizador não tiver o último login calcula-se a hora atual. Depois se o utilizador nunca fez login, são consideradas todas as mensagens que surgiram desde que foi inscrito. (fig.8)

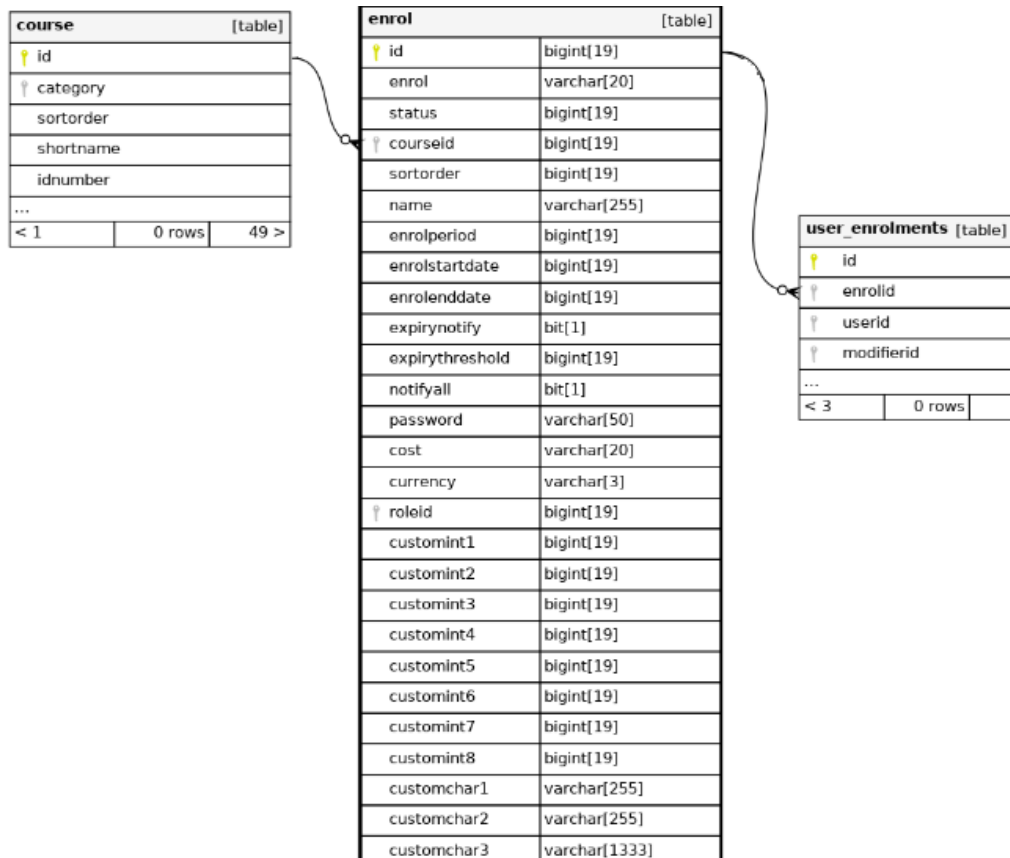


Fig. 6 – Relação tabelas – Consulta (SQL)

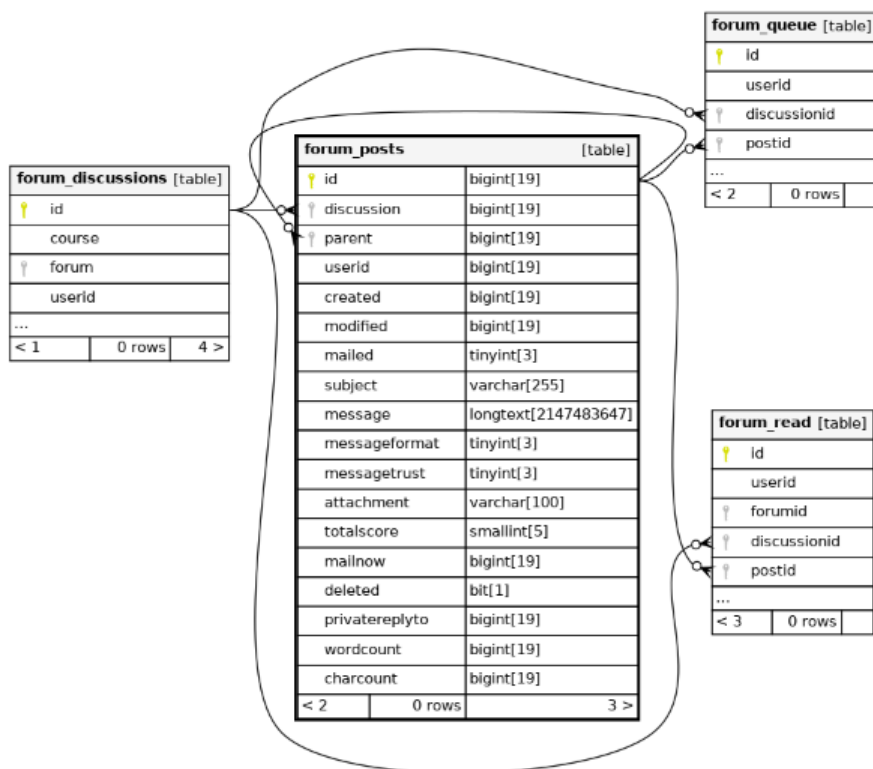


Fig. 7 – Relação tabelas – Consulta (SQL)

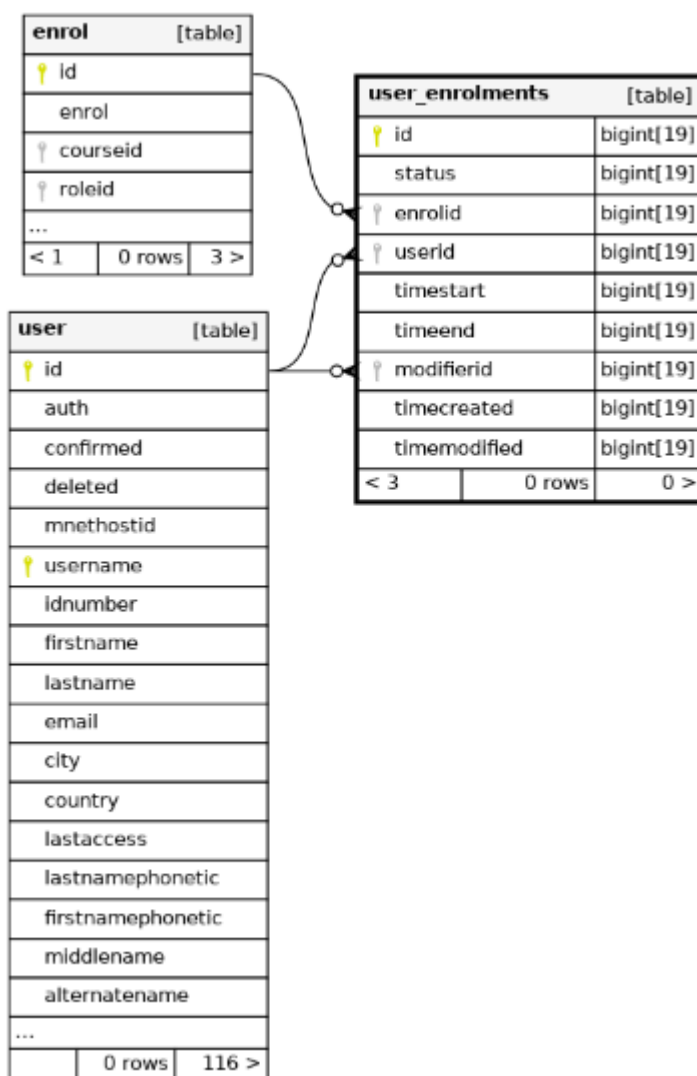


Fig. 8 – Relação tabelas – Consulta (SQL)

block_course_overviewua.php:

Este é o ficheiro principal das funcionalidades do bloco, onde usa a extensão da base do bloco já previamente descrita para “ligar” com as funções desenvolvidas no ficheiro course_overviewua.php. Começa por chamar a função init () para que assim seja possível ao Moodle diferenciar entre os blocos. De seguida é usado a função principal do bloco que é uma função pertencente ao core do Moodle e que foi modelada para corresponder aos resultados pretendidos com este bloco.

Função get_content ()

Esta função contém os parâmetros padrão já descritos anteriormente e faz a chamada à classe personalizada do fórum criada e chama as duas funções desenvolvidas que contém os objetos com os resultados pretendidos das consultas.

De seguida entra num loop foreach que verifica se existem mensagens novas (fig.9), e se sim, percorre os vários fóruns, contabiliza o número de mensagens existentes por fórum e imprime uma mensagem no bloco com esse resultado (fig. 10). Essa mensagem tem a forma de uma hiperligação que ao clicar conduz ao fórum respetivo.

Faz ainda a chamada da função specialization (), que carrega o título definido pelo utilizador e certifica-se de que nunca esteja vazio, da função instance_allow_multiple () que permite várias instâncias num único curso e por fim a função applicable_formats () que define os formatos aplicáveis ao bloco.

Esta função contém também o código da segunda funcionalidade que será explicada no próximo tópico.

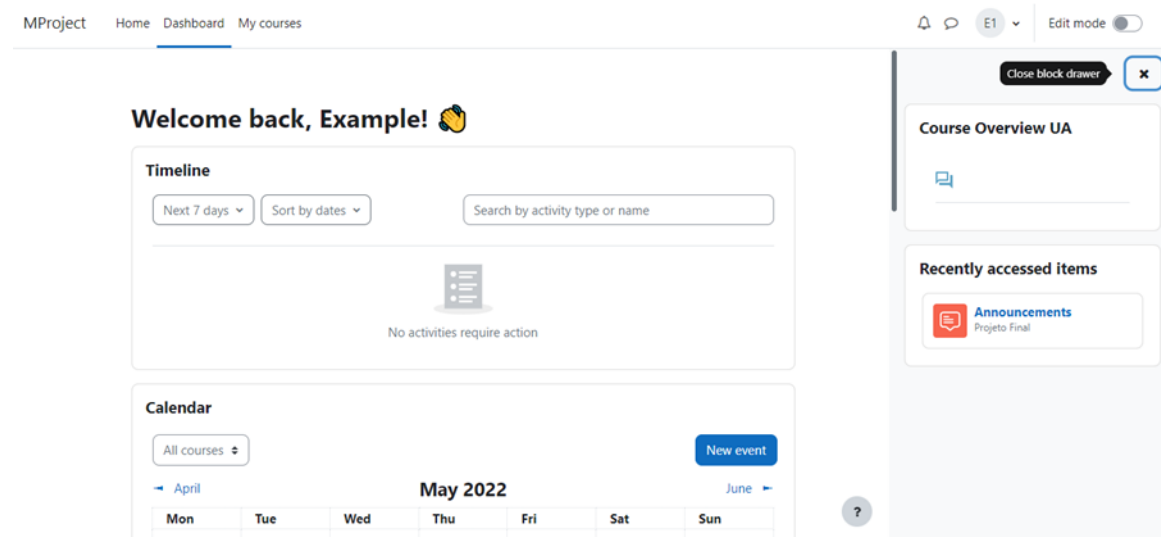


Fig. 9. – Bloco em funcionamento (Sem mensagens novas)

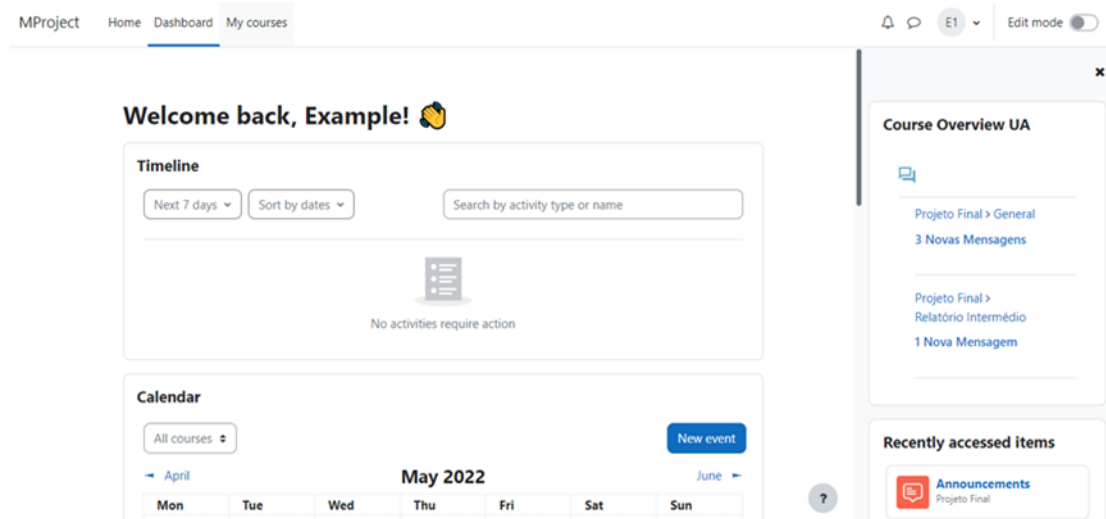


Fig. 10. – Bloco em funcionamento (Com mensagens novas)

3.7 Funcionalidade lista de ucs inscritas

Esta segunda funcionalidade tem como objetivo a indicação de quais as unidades curriculares em que dado utilizador esteja inscrito a surgirem no bloco. Este desenvolvimento assentou basicamente numa adição ao ficheiro principal de um pequeno trecho de código que contém uma consulta pelo resultado pretendido.

3.7.1 Programação da funcionalidade lista de ucs inscritas

A programação desta funcionalidade foi executada apenas num ficheiro. O ficheiro `block_course_overviewua.php`.

Neste ficheiro faz-se o uso da função `enrol_get_my_courses()` do core do moodle que faz a parametrização dos dados pretendidos sobre o utilizador associando-os às unidades curriculares em que está escrito e caso não existam, imprime uma mensagem a informar que não tem informação a mostrar. A informação dessa UC(s) é uma hiperligação que ao clicar encaminha para a página principal da UC.

Por fim faz uso da função `render_from_template()` para renderizar e assim mostrar toda a informação. (Fig. 11)

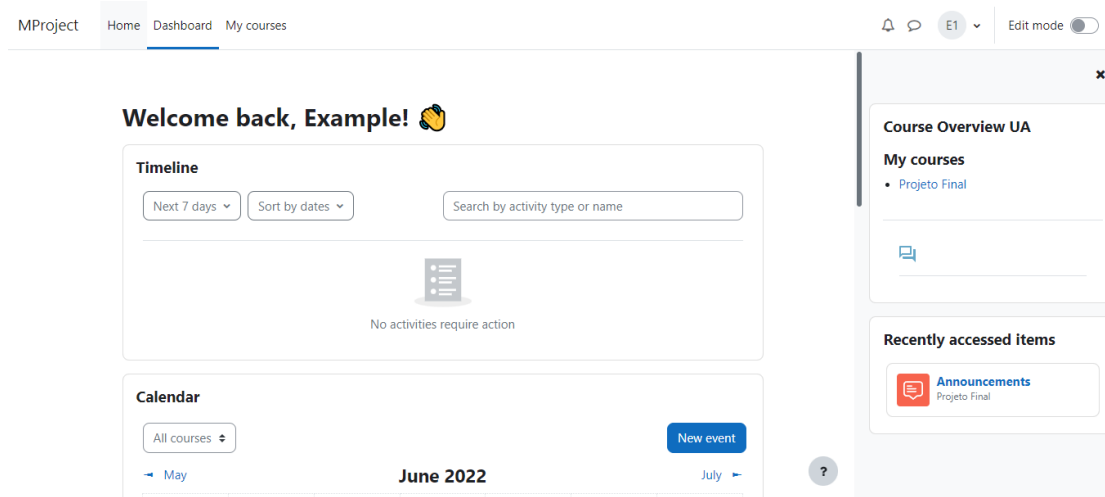


Fig. 11 – Bloco em funcionamento com informação sobre UC em que está inscrito

3.8 Otimizar a consulta à base de dados

O Moodle pode ser instalado numa infra-estrutura de forma a ter um desempenho muito bom, em pequenos níveis de uso ou escalando para muitos milhares de utilizadores. Os fatores envolvidos no desempenho são basicamente os mesmos de qualquer sistema baseado em banco de dados baseado em PHP. Na tentativa de se otimizar um servidor, é importante focar no fator que fará mais diferença para o utilizador. Por exemplo, se existirem relativamente mais utilizadores a navegar do que a acessar ao banco de dados, dever-se-ia melhorar o desempenho do servidor web.

No entanto, e tendo em conta que o servidor é local, esta parte da otimização do servidor web não foi efetuada. Portanto a única tentativa de otimização foi feita nas consultas pelo bloco.

Nesse sentido a consulta que me pareceu necessitar de algum tipo de otimização foi a segunda função que recebe a mensagem de contagem de formulários inscritos, e desde o lastlogin retorna com base em todas as respostas existentes, onde removi o foreach(fig.12) da consulta e substitui pelo array_column() (fig.13) de forma a extrair a informação mais eficientemente.


```

public function get_discussion_count() {
    global $DB, $USER;
    $params = array();
    $where = '';
    $params['timecreated'] = isset($USER->lastlogin) ? $USER->lastlogin : time(); //lastlogin condicao utilizador não tiver o último login ca
    $sql = "SELECT fp.discussion, count(fp.discussion) as newdiscussions from {forum_posts} as fp where fp.created > :timecreated group by fp
    $discussions = $DB->get_records_sql($sql, $params);
    $reply_count = array();
    foreach ($discussions as &$discussion) {
        $reply_count[$discussion->discussion] = $discussion->newdiscussions;
    }
    return $reply_count;
}

```

Fig.12 – Consulta com o uso do loop foreach

```

public function get_discussion_count() {
    global $DB, $USER;
    $params = array();
    $params['timecreated'] = isset($USER->lastlogin) ? $USER->lastlogin : time(); //lastlogin condicao utilizador não tiver o último login c
    $sql = "SELECT fp.discussion, count(fp.discussion) as newdiscussions from {forum_posts} as fp where fp.created > :timecreated group by fp
    $discussions = $DB->get_records_sql($sql, $params);
    if($discussions){
        return array_column($discussions, 'newdiscussions', 'discussion');
    }
    return array();
}

```

Fig.13 – Consulta com o uso da função array_column ()

3.8 Manual de utilização

Existem várias formas de instalar o Plugin/Bloco:

Instalar via arquivo ZIP carregado

1. Faça login no seu site Moodle como administrador e vá para _Administração do site> Plugins > Instalar plugins_. (Fig.14)

Site administration

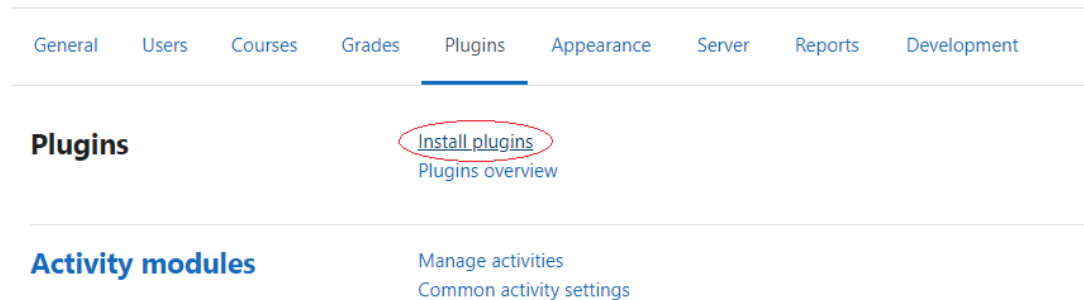


Fig.14 – Instalação de plugin

2. Carregue o arquivo ZIP com o código do plugin. (Fig.15) Apenas deve ser solicitado a adicionar detalhes extras se o seu tipo de plug-in não for detetado automaticamente.

▼ Install plugin from ZIP file ?

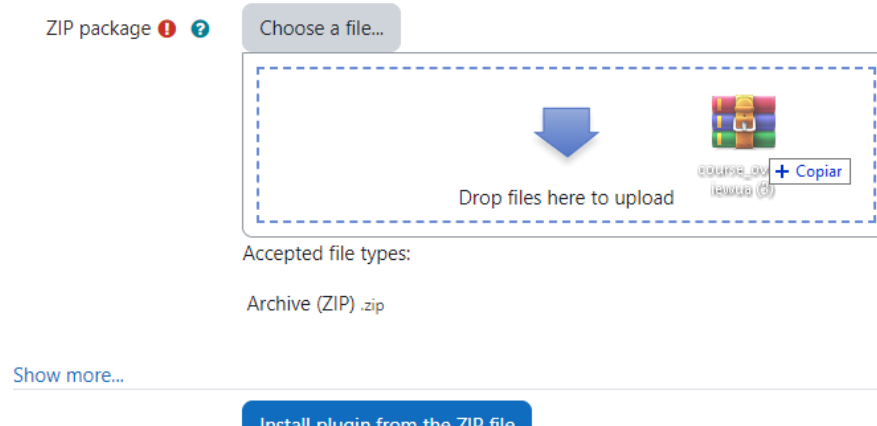


Fig. 15 – Upload ficheiro Zip

▼ Install plugin from ZIP file ?

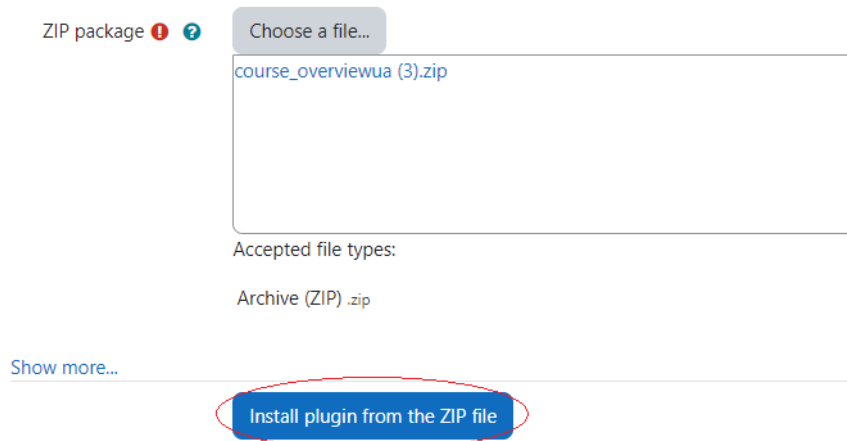


Fig. 16 – Instalar Plugin

3. Verifique o relatório de validação do plug-in e finalize a instalação. (Fig.16)

MoodleProject

Install plugin from ZIP file

Validating block_course_overviewua ... OK

[Warning] Declared maturity level [MATURITY_RC]



[Warning] Target location already exists and will be removed [C:\wamp64\www\MoodleProject\blocks/course_overviewua]



Validation successful, installation can continue

Continue Cancel

Fig. 17 – Validação bem-sucedida

Instalar manualmente

O plugin também pode ser instalado colocando o conteúdo deste diretório em

{your/moodle/dirroot} /blocks/course_overviewua

Depois, faça login no seu site Moodle como administrador e vá para _Administração do site> Notificações_ para concluir a instalação. (Fig.17)

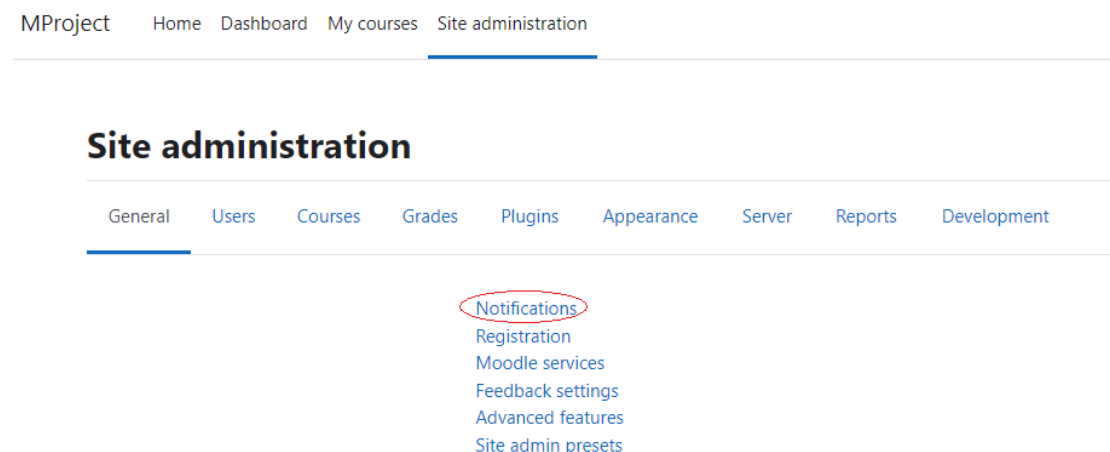


Fig. 18 – Concluir Instalação Manualmente

Alternativamente, também é possível executar:

```
$ php admin/cli/upgrade.php
```

para concluir a instalação a partir da linha de comandos.

Capítulo 4

Conclusões e Trabalho Futuro

Fazendo uma retrospectiva, desde a escolha do projeto até à conclusão do mesmo creio que os objetivos que foram identificados no início do projeto, os quais estão descritos no início deste relatório, boa parte dos mesmos foram atingidos. No entanto creio que existindo mais tempo/disponibilidade este mesmo projeto poderia ter sido desenvolvido com mais funcionalidades, e que certamente ficará para trabalho futuro.

4.1. Resultados principais

Desde o trabalho de pesquisa para saber exatamente o que é Moodle, como funciona, qual é a estrutura de arquivos, encontrar uma maneira de criar um bloco, o que é um bloco, como funciona e entender o que foi encontrado, e por fim o trabalho de pesquisa para desenvolver o bloco foi um processo de aprendizagem e consolidação de conhecimento extremamente relevante que me fez atingir estes três objetivos que me foram propostos e aos quais me propus.

- “Mensagens publicadas após último login”

Este primeiro objetivo foi sem qualquer dúvida a funcionalidade mais trabalhosa. As consultas necessárias para procurar por mensagens não lidas desde o último login são bastante complexas e foram sem sombra de dúvidas onde despendi mais tempo. Tendo em conta que foi a primeira funcionalidade a ser desenvolvida e tendo eu pouca experiência em desenvolvimento PHP atingir este objetivo foi um grande passo para que o restante projeto viesse a ser concluído.

- “Listar as UCs inscritas”

Este segundo objetivo conseguido que era composto pela segunda funcionalidade do bloco acabou por saber bastante simples relativamente à primeira funcionalidade, visto que o desenvolvimento principal já estava feito, foi relativamente simples de encontrar as Unidades Curriculares em que um dado utilizador estava inscrito e listá-las no bloco.

- “otimizar as consultas à base de dados por forma a que o bloco seja rápido a carregar.”

Este terceiro objetivo creio ter sido parcialmente conseguido, tendo em conta algumas limitações de teste que possam ser feitas relativamente à otimização das consultas. Encontrei uma forma de otimizar uma consulta ao número de mensagens existentes desde o último login ao substituir um ciclo por uma consulta direta. Embora o bloco me pareça otimizado, a ideia era que, em consultas repetidas, e em situações em que o utilizador está inscrito em muitos fóruns, a query é pesada, e poderia criar uma tabela

temporária com a informação já consultada e que só seria atualizada para o momento mais recente.

4.2. Limitações

Existiram algumas limitações no que toca ao terceiro objetivo, tendo em conta que apenas foi possível colocar o bloco a aceder ao servidor localmente, quando a ideia inicial seria testá-lo no ambiente de qualidade UAb, por motivos que me são alheios isso não foi possível. Teria sido uma experiência relevante para que a otimização do bloco pudesse ser feita com outro tipo de informações.

4.3. Desenvolvimento futuro e melhorias

Este bloco tem inúmeras hipóteses de desenvolver novas funcionalidades e melhorias em alguns casos relacionados as enunciadas na proposta inicial. Com certeza que continuarei a desenvolver algumas dessas funcionalidades restantes por motivos de aprendizagem contínua e por ser uma área que tenho algum interesse. Portanto para desenvolvimento de trabalho futuro e melhorias ficaria:

- atividades com prazo a aproximar-se
- permitir ao utilizador definir prioridades para a informação a apresentar
- aviso caso exista updates no cartão de aprendizagem
- aviso especial caso receba mensagem de resposta direta ao post seu
- acrescentar a língua portuguesa como opção
- templates da página mais apelativos
- otimização da consulta à base de dados
- utilização do bloco em Mobile

Bibliografia

Coulouris, G., Dollimore, J., Kindberg, T., Blair, G. (2011). Distributed Systems: Concepts and Design. 5th Edition, Addison Wesley Longman.

Silberschatz, A., Korth, H., Sudarshan, S. (2010). Database System Concepts. 6th Edition. McGraw-Hill

Pereira, A., Poupa, C. (2017). Linguagens WEB. 6ª Edição. Sílabo

Nixon, R. (2018). Learning PHP, MySQL & JavaScript - With jQuery, CSS & HTML5. 5th Edition, O'Reilly Media

Bento, E. (2014). Desenvolvimento web com PHP e MySQL. 1ª Edição. Casa do Código

Wild, I. (2017). Moodle 3.x Developer's Guide. 1st Edition. Packt Publishing

Dougiamas, M. (2002). Moodle. <https://moodle.org/>

Green, M. (2015). Moodle Database. <https://www.examulator.com/er/>

Bias, A. (2020). Course overview on campus.

https://moodle.org/plugins/block_course_overview_campus

Papaioannou, J., Preece, G. (2022). Moodle Development.

<https://docs.moodle.org/dev/Tutorial>

Anexos

6.1. Ficheiro course_overviewua.php

```
<?php

// This file is part of Moodle - http://moodle.org/
//
// Moodle is free software: you can redistribute it and/or modify
// it under the terms of the GNU General Public License as published by
// the Free Software Foundation, either version 3 of the License, or
// (at your option) any later version.
//
// Moodle is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License
// along with Moodle. If not, see <http://www.gnu.org/licenses/>.

/**
 * Utility class
 *
 * @package    block_course_overviewua
 * @copyright  2022 Nuno Guerra (npguerra10@gmail.com)
 * @license    http://www.gnu.org/copyleft/gpl.html GNU GPL v3 or later
 */

namespace block_course_overviewua;

defined('MOODLE_INTERNAL') || die();

class course_overviewua {

    protected $context;

    public function __construct($context) {
        $this->context = $context;
    }

    /**
     * Obter discussões do fórum
     * return objeto fórum
     */
}
```

```

    public function get_discussion_forum($page = 0, $perpage = 0,
$where = null, $sort = ' fp.id DESC') {
        global $DB, $USER;
        $params = array();
        $where = '';
        //Obter a lista de utilizadores inscritos no curso, exceto
siteadmin
        if (!is_siteadmin()) { //condicao para non-admin
            $sql = "SELECT c.id FROM {course} c JOIN {enrol} en ON
en.courseid = c.id "
                . "JOIN {user_enrolments} ue ON ue.enrolid = en.id
WHERE ue.userid = $USER->id";
            $courses_user_enrolled = $DB->get_records_sql($sql);
            $courseslist = implode(',',
array_keys($courses_user_enrolled));
            if ($courseslist) {
                $where = " and c.id IN ($courseslist)";
            }
        }

        //Obter a lista de fóruns de cursos inscritos
        $sql = "SELECT distinct(fp.id),fp.discussion,fd.course,
COUNT(distinct(fp2.id)) as replies, c.fullname coursename, fp.parent,
fp.subject, fp.message, "
            . "fp.created,fp.userid FROM {course} c"
            . " left join {forum_discussions} fd ON fd.course =
c.id left join"
            . " {forum_posts} fp on fd.id = fp.discussion LEFT JOIN
mdl_forum_posts fp2 ON fp.id = fp2.parent"
            . " WHERE fp.parent =0  $where GROUP BY fp.id,
fp.subject";
        $coursediscussion = $DB->get_records_sql($sql, $params);
        return $coursediscussion;
    }
    /*
    * Obter o numero de mensagens dos forums inscritos desde o ultimo
login
    * return todas as respostas do forum
    */
    public function get_discussion_count() {
        global $DB, $USER;
        $params = array();
        $params['timecreated'] = isset($USER->lastlogin) ? $USER-
>lastlogin : time(); //condicao do ultimo login, caso utilizador nao
tenha ultimo login calcula-se o tempo atual

```



```

        $sql = "SELECT fp.discussion, count(fp.discussion) as
newdiscussions from {forum_posts} as fp where fp.created > :timecreated
group by fp.discussion";
        $discussions = $DB->get_records_sql($sql, $params);
        if($discussions){
            return array_column($discussions, 'newdiscussions',
'discussion');
        }
        return array();
    }
}

```

6.2. Ficheiro access.php

```

<?php
// This file is part of Moodle - http://moodle.org/
//
// Moodle is free software: you can redistribute it and/or modify
// it under the terms of the GNU General Public License as published by
// the Free Software Foundation, either version 3 of the License, or
// (at your option) any later version.
//
// Moodle is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License
// along with Moodle. If not, see <http://www.gnu.org/licenses/>.

/**
 * Recursos plugin para block_course_overviewuab .
 *
 * @package    block_course_overviewuab
 */

defined('MOODLE_INTERNAL') || die();
/*
 * Condições de acesso padrão ao Moodle
 */
$capabilities = array(

    'block/course_overviewuab:myaddinstance' => array(

```

```

        'riskbitmask' => RISK_SPAM | RISK_XSS,

        'captype' => 'write',
        'contextlevel' => CONTEXT_SYSTEM,
        'archetypes' => array(
            'editingteacher' => CAP_ALLOW,
            'manager' => CAP_ALLOW,
            'student' => CAP_ALLOW,
            'teacher' => CAP_ALLOW,
        )
    ),
    'block/course_overviewuab:addinstance' => array(
        'riskbitmask' => RISK_SPAM | RISK_XSS,

        'captype' => 'write',
        'contextlevel' => CONTEXT_BLOCK,
        'archetypes' => array(
            'editingteacher' => CAP_ALLOW,
            'manager' => CAP_ALLOW
        )
    ),

    'clonepermissionsfrom' => 'moodle/site:manageblocks'
),
);

```

6.3. Ficheiro block_course_overviewua.php

```

<?php
// This file is part of Moodle - https://moodle.org/
//
// Moodle is free software: you can redistribute it and/or modify
// it under the terms of the GNU General Public License as published by
// the Free Software Foundation, either version 3 of the License, or
// (at your option) any later version.
//
// Moodle is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License
// along with Moodle. If not, see <https://www.gnu.org/licenses/>.
/**

```

```

* Plugin strings sao definidas aqui.
*
* @package      block_course_overviewua
* @category     string
* @copyright    2022 Nuno Guerra <npguerra10@gmail.com>
* @license      https://www.gnu.org/copyleft/gpl.html GNU GPL v3 or
later
*/

defined('MOODLE_INTERNAL') || die();

$string['pluginname'] = 'Course Overview UA';
$string['blocktitle'] = 'Title';
$string['alldiscussions'] = 'All Discussions';
$string['nothingtodisplay'] = 'Nothing to display.';
$string['course_overviewua:myaddinstance'] = 'Course Overview UA  block
addinstance';

```

6.4. Ficheiro block_course_overviewuab.php

```

<?php

// This file is part of Moodle - https://moodle.org/
//
// Moodle is free software: you can redistribute it and/or modify
// it under the terms of the GNU General Public License as published by
// the Free Software Foundation, either version 3 of the License, or
// (at your option) any later version.
//
// Moodle is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License
// along with Moodle. If not, see <https://www.gnu.org/licenses/>.

/**
 * Block course_overviewuab definido aqui.
 *
 * @package      block_course_overviewuab
 * @copyright    2022 Nuno Guerra <npguerra10@gmail.com>

```

```

* @license      https://www.gnu.org/copyleft/gpl.html GNU GPL v3 or
later
*/
class block_course_overviewuab extends block_base {

    /**
     * Inicializacao das variaveis membro
     */
    public function init() {
        // Necessario para o Moodle diferenciar entre blocos
        $this->title = get_string('pluginname',
'block_course_overviewua');
    }

    /**
     * Retorna o conteudo do bloco.
     *
     * @return stdClass Conteudo do bloco.
     */
    public function get_content() {
        global $USER, $DB, $CFG, $OUTPUT, $PAGE;
        if ($this->content !== null) {
            return $this->content;
        }

        if (empty($this->instance)) {
            $this->content = '';
            return $this->content;
        }

        $this->content = new stdClass();
        $this->content->items = array();
        $this->content->icons = array();
        $this->content->text = '';
        $this->content->footer = '';

        if (!empty($this->config->text)) {
            $this->content->text = $this->config->text;
        } else {
            global $CFG;
            require_once $CFG->dirroot.'/lib/enrolllib.php';
            // Chamada da classe personalizada do forum
            $sitecontext = context_system::instance();
            $sitediscussions = new
\block_course_overviewuab\course_overviewuab($sitecontext);
            $discussions = $sitediscussions->get_discussion_forum();
            $reply_count = $sitediscussions->get_discussion_count();

```

```

        $data = array();
        $i = 0;
        if(!empty($discussions)){
            foreach ($discussions as &$amp;discussion) {
                //Salta se nao houver mensagens novas no forum
                if(empty($reply_count)
|| !isset($reply_count[$discussion-
>discussion])){
                    continue;
                }

                $data['discussion'][$i]['coursename'] = $discussion-
>coursename;
                $data['discussion'][$i]['subject'] = $discussion-
>subject;
                $data['discussion'][$i]['courselink'] = $CFG-
>wwwroot.'/course/view.php?id='.$discussion->course;
                $data['discussion'][$i]['forumlink'] = $CFG-
>wwwroot.'/mod/forum/discuss.php?d='.$discussion->discussion;
                $data['discussion'][$i]['reply_count'] =
$reply_count[$discussion->discussion];
                $data['discussion'][$i]['message'] =
($reply_count[$discussion->discussion] > 1) ? 'Novas Mensagens' :
'Nova Mensagem';
                $i++;
            }
            $url = $CFG->wwwroot.'/course/view.php?id=';
            $allcourses = [];
            if ($courses = enrol_get_my_courses()) {
                $courses = array_values($courses);
                $allcourses['courses'] = $courses;
                $allcourses['title'] = get_string('mycourses');
                $allcourses['curl'] = $url;
                $this->content->text .= $OUTPUT-
>render_from_template('block_course_overviewuab/courses', $allcourses);
            }

            $this->content->text .= $OUTPUT-
>render_from_template('block_course_overviewuab/discussion', $data);
        }else{
            //Se nao existir dados entao mostra esta mensagem
            $this->content->text .=
html_writer::div(get_string('nothingtodisplay',
'block_course_overviewuab'), 'alert alert-info mt-3');
        }
    }
}

```

```

        return $this->content;
    }

    /**
     * Define a configuracao dos dados.
     *
     * Funcao chamada imediatamente depois de init().
     */
    public function specialization() {

        // Faz load ao titulo definido pelo utilizador e certifica-se
que nunca esta vazio.
        if (empty($this->config->title)) {
            $this->title = get_string('pluginname',
'block_course_overviewuab');
        } else {
            $this->title = $this->config->title;
        }
    }

    /**
     * Permitir multiplas instancias num unico curso?
     *
     * @return bool True se multiplas instancias permitidas, false caso
contrario.
     */
    public function instance_allow_multiple() {
        return true;
    }

    /**
     * Define os applicable formats para o bloco.
     *
     * @return string[] Array de paginas e permissoes.
     */
    public function applicable_formats() {
        return array(
            'all' => true, 'site' => true, 'course' => true, 'my' =>
true
        );
    }
}

```

6.5 Ficheiro edit_form.php

```
<?php
// This file is part of Moodle - https://moodle.org/
//
// Moodle is free software: you can redistribute it and/or modify
// it under the terms of the GNU General Public License as published by
// the Free Software Foundation, either version 3 of the License, or
// (at your option) any later version.
//
// Moodle is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License
// along with Moodle. If not, see <https://www.gnu.org/licenses/>.

/**
 * Formulário para editar instancias do bloco course_overviewua .
 *
 * @package      block_course_overviewuab
 * @copyright    2022 Nuno Guerra <npguerra10@gmail.com>
 * @license      https://www.gnu.org/copyleft/gpl.html GNU GPL v3 or
later
 */
class block_course_overviewua_edit_form extends block_edit_form {

    /**
     * Extende a configuracao formulario para o bloco
block_course_overviewua.
     *
     * @param MoodleQuickForm $mform O form a ser construido.
     */
    protected function specific_definition($mform) {

        // Seccao header do titulo.
        $mform->addElement('header', 'configheader',
get_string('blocksettings', 'block'));

        // Elementos definidos aqui tem que comecar com 'config_'.
        $mform->addElement('text', 'config_title',
get_string('blocktitle', 'block_course_overviewuab'));
        $mform->setType('config_title', PARAM_RAW);

    }
}
```

```
}
```

6.6 Ficheiro version.php

```
<?php
// This file is part of Moodle - https://moodle.org/
//
// Moodle is free software: you can redistribute it and/or modify
// it under the terms of the GNU General Public License as published by
// the Free Software Foundation, either version 3 of the License, or
// (at your option) any later version.
//
// Moodle is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License
// along with Moodle. If not, see <https://www.gnu.org/licenses/>.

/**
 * Versao do Plugin e outros metadados sao definidos aqui.
 *
 * @package      block_course_overviewuab
 * @copyright    2022 Nuno Guerra <npguerra10@gmail.com>
 * @license      https://www.gnu.org/copyleft/gpl.html GNU GPL v3 or
later
 */

defined('MOODLE_INTERNAL') || die();

$plugin->component = 'block_course_overviewuab';
$plugin->release = '0.1.0';
$plugin->version = 2022040101.03;
$plugin->requires = 2020061500;
$plugin->maturity = MATURITY_RC;
```