

SISTEMA DE AVALIAÇÃO AUTOMÁTICA PARA E-LEARNING



FEUP

João Xavier

Orientador

António Coelho

CONTEXTO

- Ensino massificado
- Número elevado de alunos por docente
- Necessidade de avaliação frequente
- Falta de *feedback* rápido e construtivo

SISTEMA DE AVALIAÇÃO AUTOMÁTICA

- Desenvolvido no ano lectivo 2009/2010
- *DOMjudge* como componente
- Integração com o *Moodle*
- Validação de *input - output*



PROBLEMA

- Funcionalidades limitadas
- Pouco customizável
- Não foi testado com alunos

PROBLEMA

COMO TORNAR MAIS EFICAZ O SISTEMA DE AVALIAÇÃO
AUTOMÁTICA?

OBJECTIVOS

- Desenvolvimento de módulos avançados
 - *Feedback* personalizado
 - Ferramentas de autoria de provas mais intuitivas
 - Detecção de plágio
 - Análise da qualidade estática do código
- Teste do sistema numa unidade curricular

SISTEMA BASE

SISTEMA BASE

- ✓ Parâmetros de avaliação suportados
- ✓ Definição de casos de teste

SISTEMA BASE

- ✓ Parâmetros de avaliação suportados
- ✓ Definição de casos de teste
 - *Feedback* personalizado
 - Suporte à elaboração de soluções

SISTEMA BASE

- ✓ Parâmetros de avaliação suportados
- ✓ Definição de casos de teste
 - *Feedback* personalizado
 - Suporte à elaboração de soluções
- ✗ Análise da qualidade estática do código
- ✗ Detecção de plágio

ESTADO DE ARTE

- Sistemas de avaliação automática
 - CourseMarker
 - BOSS
 - Submit!
 - RoboProf
 - GAME

ESTADO DE ARTE

ESTADO DE ARTE

ANÁLISE ESTÁTICA

- Tipografia
- Complexidade
- Estrutura
- Estilo

ESTADO DE ARTE

ANÁLISE ESTÁTICA

- Tipografia
- Complexidade
- Estrutura
- Estilo

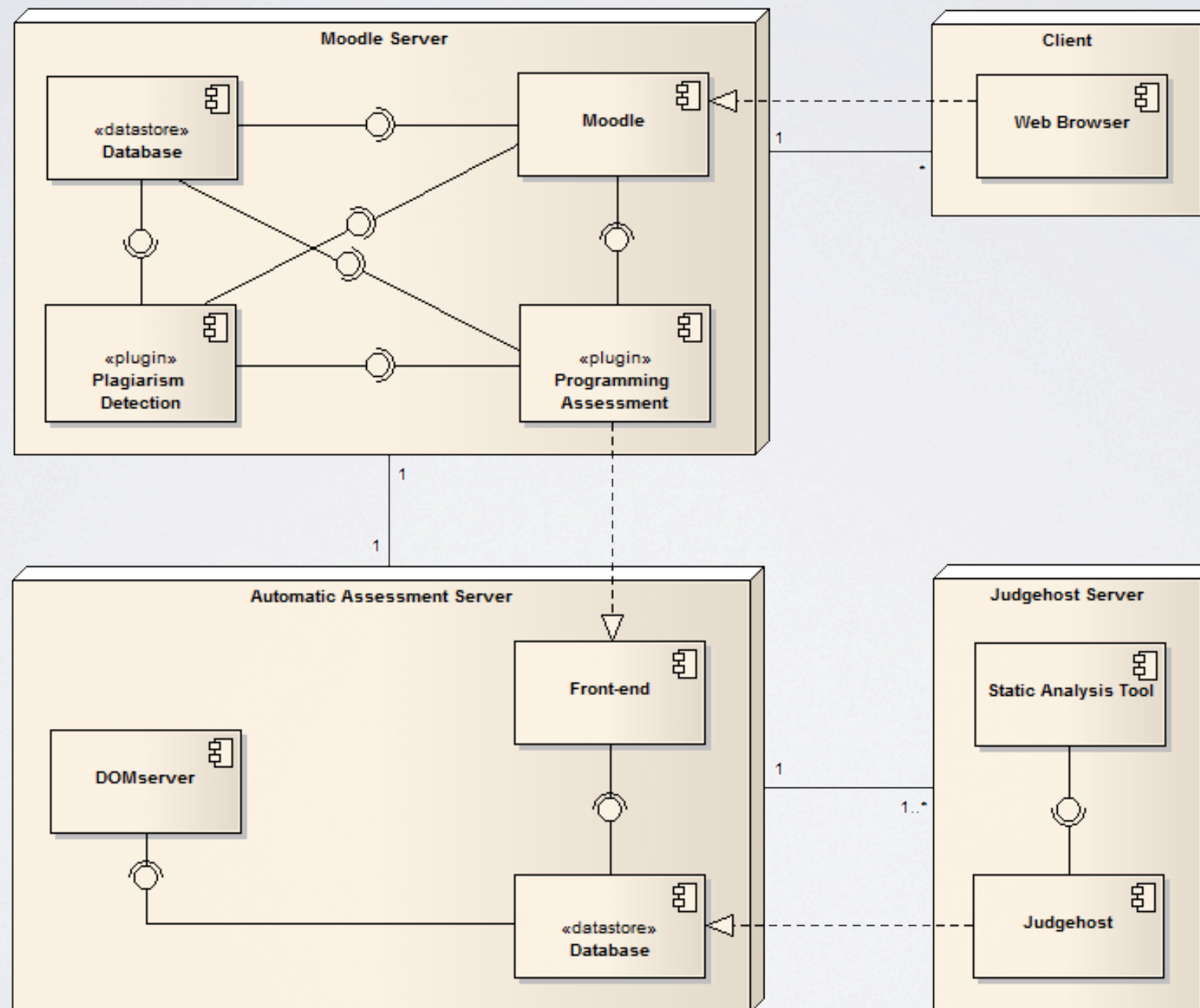
DETECÇÃO DE PLÁGIO

- Comparação de estrutura
- Técnica da “marca de água”
- Ferramentas externas
- *Plugins para o Moodle*

REQUISITOS

- Integração com o *Moodle* 2.0
- Adição de novas linguagens de programação
- Implementação dos módulos propostos

ARQUITECTURA PROPOSTA



IMPLEMENTAÇÃO

- Caso de estudo
- Módulo de análise estática da qualidade do código
- Módulo de detecção de plágio

CASO DE ESTUDO

- Linguagem de programação SQL
 - Bases de dados *Oracle*
- *Input*: detalhes da ligação
- *Output*: solução do docente

CASO DE ESTUDO

nome	idade
Adelino	27
Inês	24
Paulo	23
Rui	22

id	nome	idade
1	Inês	24
4	Adelino	27
2	Rui	22
3	João	23
5	Paulo	23

CASO DE ESTUDO

nome	idade
Adelino	27
Inês	24
Paulo	23
Rui	22

id	nome	idade
1	Inês	24
4	Adelino	27
2	Rui	22
3	João	23
5	Paulo	23

✗ Número total de colunas

CASO DE ESTUDO

nome	idade
Adelino	27
Inês	24
Paulo	23
Rui	22

id	nome	idade
1	Inês	24
4	Adelino	27
2	Rui	22
3	João	23
5	Paulo	23

✗ Número total de colunas

✗ Número total de linhas

CASO DE ESTUDO

nome	idade
Adelino	27
Inês	24
Paulo	23
Rui	22

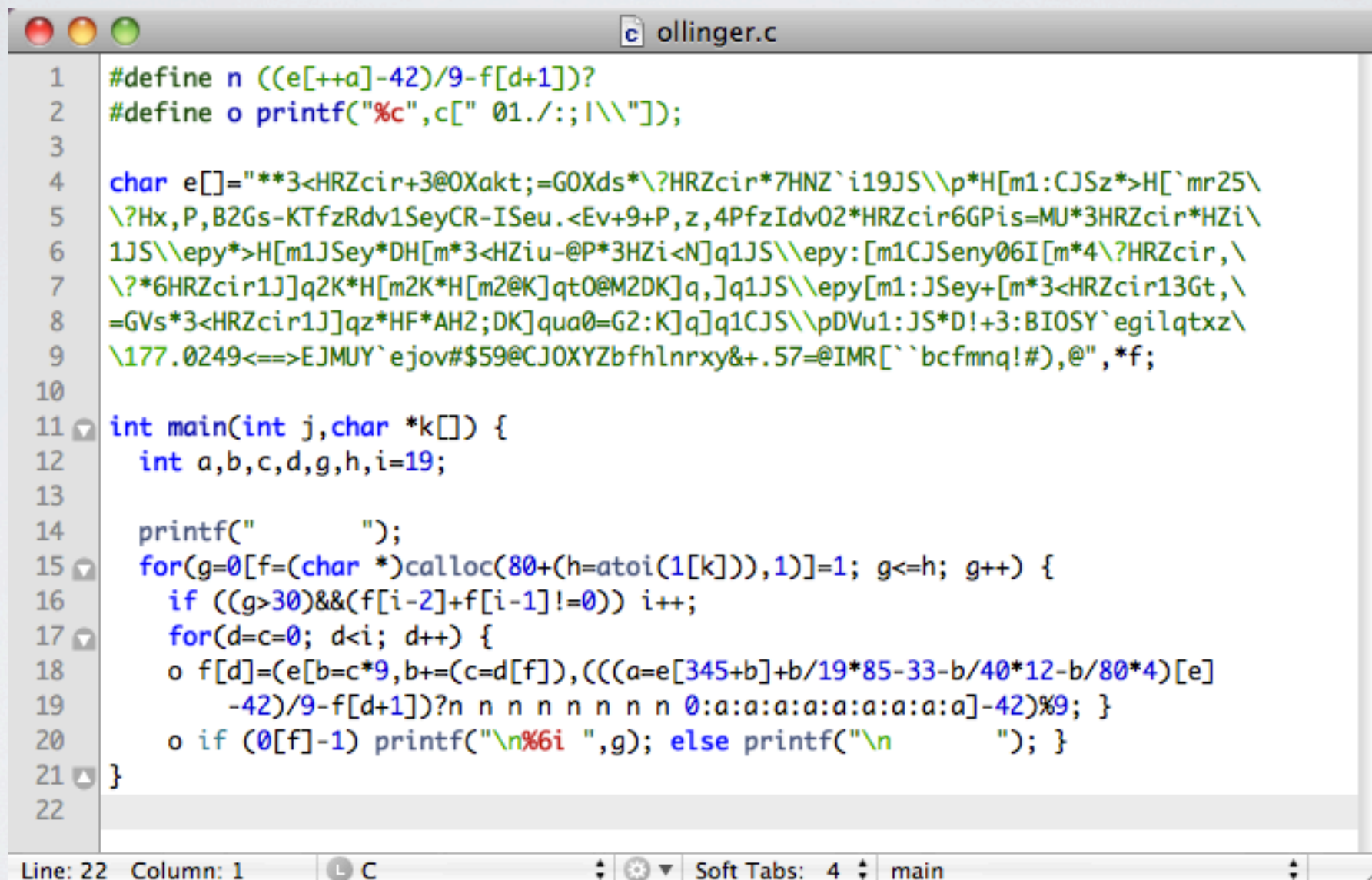
id	nome	idade
1	Inês	24
4	Adelino	27
2	Rui	22
3	João	23
5	Paulo	23

- ✗ Número total de colunas
- ✗ Número total de linhas
- ✗ Número de entradas

IMPLEMENTAÇÃO

- Caso de estudo
- Módulo de análise estática da qualidade do código
- Módulo de detecção de plágio

ANÁLISE ESTÁTICA



```
1  #define n ((e[++a]-42)/9-f[d+1])?
2  #define o printf("%c",c[" 01./:;|\\"]);
3
4  char e[]="**3<HRZcir+3@0Xakt;=G0Xds*\\?HRZcir*7HNZ`i19JS\\p*H[m1:CJSz*>H[`mr25\\
5  \\?Hx,P,B2Gs-KTfzRdv1SeyCR-ISEu.<Ev+9+P,z,4PfzIdv02*HRZcir6GPis=MU*3HRZcir*HZi\\
6  1JS\\ep*y*>H[m1JSey*DH[m*3<HZiu-@P*3HZi<N]q1JS\\ep*y:[m1CJSeny06I[m*4\\?HRZcir,\\
7  \\?*6HRZcir1J]q2K*H[m2K*H[m2@K]qt0@M2DK]q,]q1JS\\ep*y[m1:JSey+[m*3<HRZcir13Gt,\\
8  =GVs*3<HRZcir1J]qz*HF*AH2;DK]qua0=G2:K]q]q1CJS\\pDVu1:JS*D!+3:BIOSY`egilqtxz\\
9  \\177.0249<==>EJMUY`ejov#$59@CJOXYZbfhlnrxy&+.57=@IMR[``bcfmnq!#),@",*f;
10
11 int main(int j,char *k[]) {
12     int a,b,c,d,g,h,i=19;
13
14     printf("      ");
15     for(g=0[f=(char *)calloc(80+(h=atoi(1[k])),1)]=1; g<=h; g++) {
16         if ((g>30)&&(f[i-2]+f[i-1]!=0)) i++;
17         for(d=c=0; d<i; d++) {
18             o f[d]=(e[b=c*9,b+=(c=d[f]),(((a=e[345+b]+b/19*85-33-b/40*12-b/80*4)[e]
19                 -42)/9-f[d+1])?n n n n n n n n 0:a:a:a:a:a:a:a:a[a]-42)%9; }
20             o if (0[f]-1) printf("\\n%6i ",g); else printf("\\n      "); }
21     }
22 }
```

Line: 22 Column: 1 C Soft Tabs: 4 main

ANÁLISE ESTÁTICA

ANÁLISE ESTÁTICA

- Geração de uma “*Abstract Syntax Tree*”

ANÁLISE ESTÁTICA

- Geração de uma “*Abstract Syntax Tree*”
- Filtragem dos resultados

ANÁLISE ESTÁTICA

- Geração de uma “*Abstract Syntax Tree*”
- Filtragem dos resultados
- Remoção dos comentários

ANÁLISE ESTÁTICA

- Geração de uma “*Abstract Syntax Tree*”
- Filtragem dos resultados
- Remoção dos comentários
- Desenvolvimento das métricas (*Bash scripts*)

ANÁLISE ESTÁTICA

ANÁLISE ESTÁTICA

- *McCabe's Cyclomatic Complexity*

ANÁLISE ESTÁTICA

- *McCabe's Cyclomatic Complexity*
- Métricas de *Halstead* (*Software Science*)

ANÁLISE ESTÁTICA

- *McCabe's Cyclomatic Complexity*
- Métricas de *Halstead* (Software Science)
- Métricas de estilo

ANÁLISE ESTÁTICA

ANÁLISE ESTÁTICA

- Personalização
 - Percentagem da nota final
 - Peso, valores mínimos e máximos para cada métrica

ANÁLISE ESTÁTICA

- Personalização
 - Percentagem da nota final
 - Peso, valores mínimos e máximos para cada métrica
- Extensibilidade
 - Linguagens de programação
 - Variedade de métricas

IMPLEMENTAÇÃO

- Caso de estudo
- Módulo de análise estática da qualidade do código
- Módulo de detecção de plágio

DETECÇÃO DE PLÁGIO

- Implementação de um algoritmo
- Integração de uma ferramenta externa num *plugin*
- Adaptação de um *plugin* existente
 - Turnitin
 - Crot

DETECÇÃO DE PLÁGIO


- Implementação de um algoritmo
- Integração de uma ferramenta externa num *plugin*
- Adaptação de um *plugin* existente
 - Turnitin
 - Crot

DETECÇÃO DE PLÁGIO

- Uso da *API* de plágio do *Moodle* 2.0
- Possibilidade de integrar mais *plugins*
- Interface para comparação de submissões similares

DETECÇÃO DE PLÁGIO

Student name	Similar assignments		
student 4	Name	Course	Similarity score
	student 3	FPRO	100 %
	student 4	FPRO	92.86 %

Global plagiarism detection is supported by Bing search engine 

DETECÇÃO DE PLÁGIO

**student 4:
FPRO, Test**

```
bool isSieveDone = false;
int primes[MAX], pCount = 0;

bool is_prime(int a)
{
    doSieve();
    return isprime[a];
}

void doSieve()
{
    if (isSieveDone) return;

    int i, j, root = (int) sqrt(MAX) + 1;
    memset(isprime, true, sizeof(isprime));
    isprime[0] = isprime[1] = false;

    for (i = 2; i <= root; i++)
```

**student 3:
FPRO, Test**

```
bool isSieveDone = false;
int primes[MAX], pCount = 0;

bool is_prime(int a)
{
    doSieve();
    return isprime[a];
}

void doSieve()
{
    if (isSieveDone) return;

    int i, j, root = (int) sqrt(MAX) + 1;
    memset(isprime, true, sizeof(isprime));
    isprime[0] = isprime[1] = false;
```


CONCLUSÕES

CONCLUSÕES

- Especificação de módulos avançados

CONCLUSÕES

- Especificação de módulos avançados
- Nova arquitectura

CONCLUSÕES

- Especificação de módulos avançados
- Nova arquitectura
- Caso de estudo na Unidade Curricular “Bases de Dados”

CONCLUSÕES

- Especificação de módulos avançados
- Nova arquitectura
- Caso de estudo na Unidade Curricular “Bases de Dados”
- Suporte à linguagem de programação SQL

CONCLUSÕES

- Especificação de módulos avançados
- Nova arquitectura
- Caso de estudo na Unidade Curricular “Bases de Dados”
- Suporte à linguagem de programação SQL
- Análise da qualidade estática do código

CONCLUSÕES

- Especificação de módulos avançados
- Nova arquitectura
- Caso de estudo na Unidade Curricular “Bases de Dados”
- Suporte à linguagem de programação SQL
- Análise da qualidade estática do código
- Detecção de plágio

TRABALHO FUTURO

- Correção automática de projectos com diversos ficheiros fonte
- Mais métricas e suporte para outras linguagens de programação
- Optimização do algoritmo de detecção de plágio
- Melhorias a nível de usabilidade

*“I do not fear computers.
I fear the lack of them.”*

Isaac Asimov

João Xavier

<http://jcxavier.com>
joao.xavier@fe.up.pt

11 de Julho de 2011