

# SOPAS - Submissão Online Para Análise de Software (fase 4)

José Pedro Silva   Pedro Faria   Ulisses Costa

Engenharia de Linguagens  
Projecto integrado

June 27, 2011

# Index

- 1 Objectivos
- 2 Aplicação Web
  - Correção de bugs
  - Adição de funcionalidades
- 3 Metricas
- 4 Conclusão e trabalho futuro

# Até agora:

Concretizado até ao início da quarta fase:

- Termina da aplicação web e adição de funcionalidades extra ✓
- Implementação de um script para instalação do sistema ✓
- Melhoramento do interface pelo terminal ✓
- Implementação de algumas métricas ✓

# Motivação e Objectivos

Objectivos para terceira fase:

- Desenvolvimento de uma API para gerir as métricas
- Terminar a implementação das métricas que pretendíamos
- Melhoramento da *script* de instalação do sistema (difícil!)
- Permitir inserção de informação pelo terminal
- Corrigir e melhorar a aplicação Web

# Index

- 1 Objectivos
- 2 Aplicação Web
  - Correcção de bugs
  - Adição de funcionalidades
- 3 Métricas
- 4 Conclusão e trabalho futuro

# Index

- 1 Objectivos
- 2 Aplicação Web
  - Correcção de bugs
  - Adição de funcionalidades
- 3 Metricas
- 4 Conclusão e trabalho futuro

# Correcção de bugs

- Adicionada informação que estava em falta:
  - warnings ou erros emitidos na altura da compilação já são guardados e apresentados ao utilizador

## Tentativa

### Path:

/Users/Pedro/Dropbox/MEI/EL/PI/Static-Code-Analyzer/sample\_app/data/concursos/contest-4/en-6/user-1/tent-20110626214707/somaParams.c

### Compilou: Não

### Erros:

somaParams.c: In function 'main': somaParams.c:14: error: expected ';' before '}' token

# Index

- 1 Objectivos
- 2 Aplicação Web
  - Correcção de bugs
  - Adição de funcionalidades
- 3 Metricas
- 4 Conclusão e trabalho futuro



# Adição de funcionalidades

Preparação do sistema para:

- gerar relatório dos resultados e de métricas para determinado concurso
- permitir o download ou a visualização dentro do browser dos relatórios

## Concursos

- Exame pi | [editar](#) | [eliminar](#) | (Estado: Inactivo)  
Métricas: [Download](#) | [Visualizar](#)  
Resultados: [Gerar](#)

# Gerar relatório de resultados

- Ir à tabela de melhores resultados e ir buscar todas as entradas para o concurso em questão
- Para cada utilizador encontrar o resultado de cada enunciado
- Apenas contar os que estiverem 100% correctos
- Calcular o resultado de cada enunciado, tendo em conta o seu peso no concurso
- Apresentar o resultado calculado e o tempo de execução

# Relatório de resultados

## Resultados do concurso: Exame pi

Exercício 6 (50%): Soma dos argumentos recebidos

Exercício 7 (50%): Conta os parametros recebidos

T = Tempo de execução de todas as baterias de teste para o enunciado.

Grupo	Ex. 6	T	Ex. 7	T	Total
ze	50.0	0.20436	50.0	0.09497	100.0
joao	50.0	0.0919	0	0	50.0

# Index

- 1 Objectivos
- 2 Aplicação Web
  - Correção de bugs
  - Adição de funcionalidades
- 3 Metricas
- 4 Conclusão e trabalho futuro

# Metricas Implementadas

- Grafo de includes do sistema e de cada ficheiro
- Nr linhas de comentários (que não são pedaços de código comentados)
- Densidade de comentários
- Index de McCabe
- NLOC (nr de linhas do pretty print)
- Nr de linhas físicas
- Clones por bloco
- Assinaturas de funções e nomes de funções

# Metrics Datatypes

```
type Metrics = M.Map MetricName MetricValue
type MetricName = (String, Maybe FileSrc, Maybe FunctionName)
data MetricValue =
  | Num Double
  | Clone (M.Map FileDst [(Ocurrency, LineSrc, LineDst)])
  | Includes ([SystemIncludes],[Includes])
  | FunSig [FunSignature]
  | Graphviz DotFile
  | GraphvizProject DotFile
```

# Metrics API

```
(>.) :: Metrics -> (MetricName, MetricValue) -> Metrics  
m >.> (mn, mv) =  
  case M.lookup mn m of  
    Nothing    -> m'  
    (Just mv') -> if mv' == mv then m else m'  
  where m' = M.insert mn mv m
```

## Caso de uso

```
emptyMetrics >>> (("mccbaIndex", Nothing, Nothing), Num 10)
```

# Metrics API

```
(>+>) :: Metrics -> Metrics -> Metrics  
m1 >+> m2 = M.union m1 m2
```

## Caso de uso

```
concatMetrics :: [Metrics] -> Metrics  
concatMetrics = foldl (>+>) emptyMetrics
```



# Metrics API

```
foldrM :: (MetricName -> MetricValue -> c -> c) -> c -> Metrics -> c  
foldrM f s = M.foldrWithKey f s
```

## Caso de uso

```
...  
foldrM step noop m  
  where step k v r = "\\begin{dot2tex}[]"  
                >> (fromString $ fromGraphvizP v)  
                >> "\\end{dot2tex}"  
                // r  
          fromGraphvizP (GraphvizProject 1) = 1
```

# Metrics API

```
getMetricsFrom :: (a -> IO Metrics) -> [a] -> IO Metrics
getMetricsFrom f l =
  forkMapM f l >>=
    return . concatMetrics . map (either (const emptyMetrics) id)
```

## Caso de uso

```
getListOfCFiles :: FilePath -> IO [FilePath]
getTreeFromFile :: FilePath -> [FilePath] -> IO [(FilePath, CTranslUnit)]
mccabePerFun :: (FilePath, CTranslUnit) -> IO Metrics

getListOfCFiles fp >>= getTreeFromFile fp >>= getMetricsFrom mccabePerFun
```

# Implementação

```
mccabeIndex :: Data a => a -> IO Int
mccabeIndex = applyTU (full_tdtTU typesOfInstr)

typesOfInstr = constTU 0
  'ad hocTU' loop
  'ad hocTU' binaryOp
loop :: CStat -> IO Int
loop = return . loop_
  where loop_ (CIf _ _ _ _) = 1
        loop_ (CSwitch _ _ _) = 1
        loop_ (CWhile _ _ _ _) = 1
        loop_ (CFor _ _ _ _ _) = 1
        loop_ _ = 0
binaryOp :: CBinaryOp -> IO Int
binaryOp = return . binaryOp_
  where binaryOp_ CLndOp = 1
        binaryOp_ CLorOp = 1
        binaryOp_ _ = 0
```

# Implementação

```
ncloc :: (FilePath, CTranslUnit) -> IO Metrics
ncloc (file, tree) =
  let len = (length . filter (not . null) . lines . show . pretty) tree
  in return $ emptyMetrics
    >.> (("ncloc", Just file, Nothing), Num $ fromIntegral len)
```

# Index

- 1 Objectivos
- 2 Aplicação Web
  - Correção de bugs
  - Adição de funcionalidades
- 3 Metricas
- 4 Conclusão e trabalho futuro

# Conclusão e trabalho futuro

- Todas as métricas pretendidas foram implementadas
- Foi desenvolvida uma api para o sistema de extracção de métricas
- Sistema preparado para extracção de qualidade de um programa através das métricas calculadas
- Ambas as interfaces de comunicação com a aplicação (Web e linha de comandos) ficaram terminadas
- São gerados relatórios sobre os resultados de cada utilizador nos concursos, e sobre as métricas

# Perguntas

?