



Plone Python CMS / Evowave

Renato Novais / Enoque Joseneas

INF022 – Tóicos Avançados

renato@ifba.edu.br

Objeto de Estudo

- Plone Python CMS
- Commits por arquivo
 - N° total de mudanças
 - Commiter (quem comitou)
 - Data do commit
 - Url do blob
 - Nome do arquivo

Sobre o Plone



- É um CMS (Content Management System)
- Escrito em Python
- Roda sobre o servidor web Zope
- Surgiu em 1999 / 2008 no github
- Software Livre - usa a licença GPL
- Multiplataforma (Linux, Windows, Mac, BSD)

Sobre o Plone



- 1119509 Downloads
- 11320 Commits (Plone CSM + Plone Rest Api + Plone templates + Plone plugins)
- 3559 Commits (Plone CMS - até 20/04/2017)
- 176 Pull Requests
- Disponível em mais de 40 idiomas
- Suporte a intranets
- Suporte a multi-sites (Single Admin)

Porque o Plone?



- Software livre
 - Customizável, código fonte disponível, total controle da aplicação
- Adotado e recomendado para sites governamentais em vários países
 - Tem foco em segurança, escalabilidade, admin responsivo e multi-plataforma
- É um caso de sucesso do Serpro
 - Primeira aparição foi durante o FISL de 2003, através de uma apresentação do Luciano Ramalho, líder da comunidade Python
 - Aprovado no trabalho de investigação realizado pelo serpro de 2003 a 2004
- Utilizado em vários portais federais

Quem usa o plone?



- Principais portais do governo federal
 - <http://brasil.gov.br/>, <http://presidencia.gov.br/>
 - Secretaria de Comunicações da Presidência da República
 - <http://www.secom.gov.br/>
 - Ministério dos Transportes - DNIT
 - <http://www.dnit.gov.br/>
 - Ministério do Desenvolvimento, Indústria e Comércio Exterior - INPI
 - <http://www.inpi.gov.br/>
- FBI (USA)

Sobre a implementação



- Código QML (Qt) / C++ -> Json
 - 1º Requisição ao github para pegar a lista de *Trees*
 - 2º Algoritmo em c++ - Implementação do pattern *Composite* para contruir a árvore de diretórios e arquivos
 - 3º Requisição ao github para pegar a lista de commits
 - Do primeiro commit (no github - 2008) até 20/04/2017
 - 4º Construção de um Map com os commits por arquivo

Sobre a implementação

- Código QML (Qt) / C++ -> Json
 - 5º Construção das moléculas por arquivo em um outro map
 - Arquivo → lista de moléculas
 - 6º Construção final do json da visualização:
 - Inicia a iteração da árvore de diretórios e arquivos
 - Para cada arquivo na árvore, anexa as moléculas (já indexadas em outro arquivo json)

Sobre a implementação

1. Método para gerar as moléculas:

```
2. // recebe um map contendo o json de um commit
3. void setCommitHistory(const QVariantMap &fileData)
4. {
5.     QVariantMap map;
6.     QVariantMap object;
7.     QVariantMap commitAuthor = fileData.value("commit").toMap().value("author").toMap();
8.     int format = Qt::DateFormat::ISODate;
9.     QDate commitData = QDate::fromString(commitAuthor.value("date").toString(), format);
10.    map.insert("committer", commitAuthor.value("name"));
11.
12.    // itera a lista de arquivos commitados
13.    foreach (const QVariant &item, fileData.value("files").toList()) {
14.        object = item.toMap();
15.        map.insert("committed", commitData.toString("dd/MM/yyyy"));
16.        map.insert("url", object.value("blob_url"));
17.        map.insert("changes", object.value("changes"));
18.        map.insert("filename", object.value("filename"));
19.        QString color(moleculesColor(object.value("changes").toInt()));
20.        object.clear();
21.        object.insert("color", color);
22.        object.insert("data", map);
23.        object.insert("window", commitData.year());
24.        m_history.append(object);
25.        map.clear();
26.    }
27. }
```

Sobre a visualização



- Arquivo final com 21,1 MB / 316.533 linhas
- Janelas representam os anos de 2008~2017
 - Total de 9 anos de commits
- Setores representam os diretórios e arquivos
 - Diretórios podem ser compostos de ambos
- Moléculas representam detalhes de commit por arquivo:
 - Data do *commit*, total de mudanças, *filename*, *committer*, *blob url*

Screenshots da visualização

Molecule Info:

Sector: Plone Python CMS

Window: 9

Nº changes:

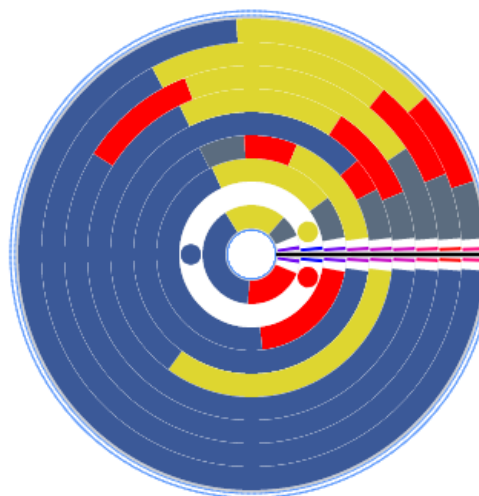
Committed at:

Committer:

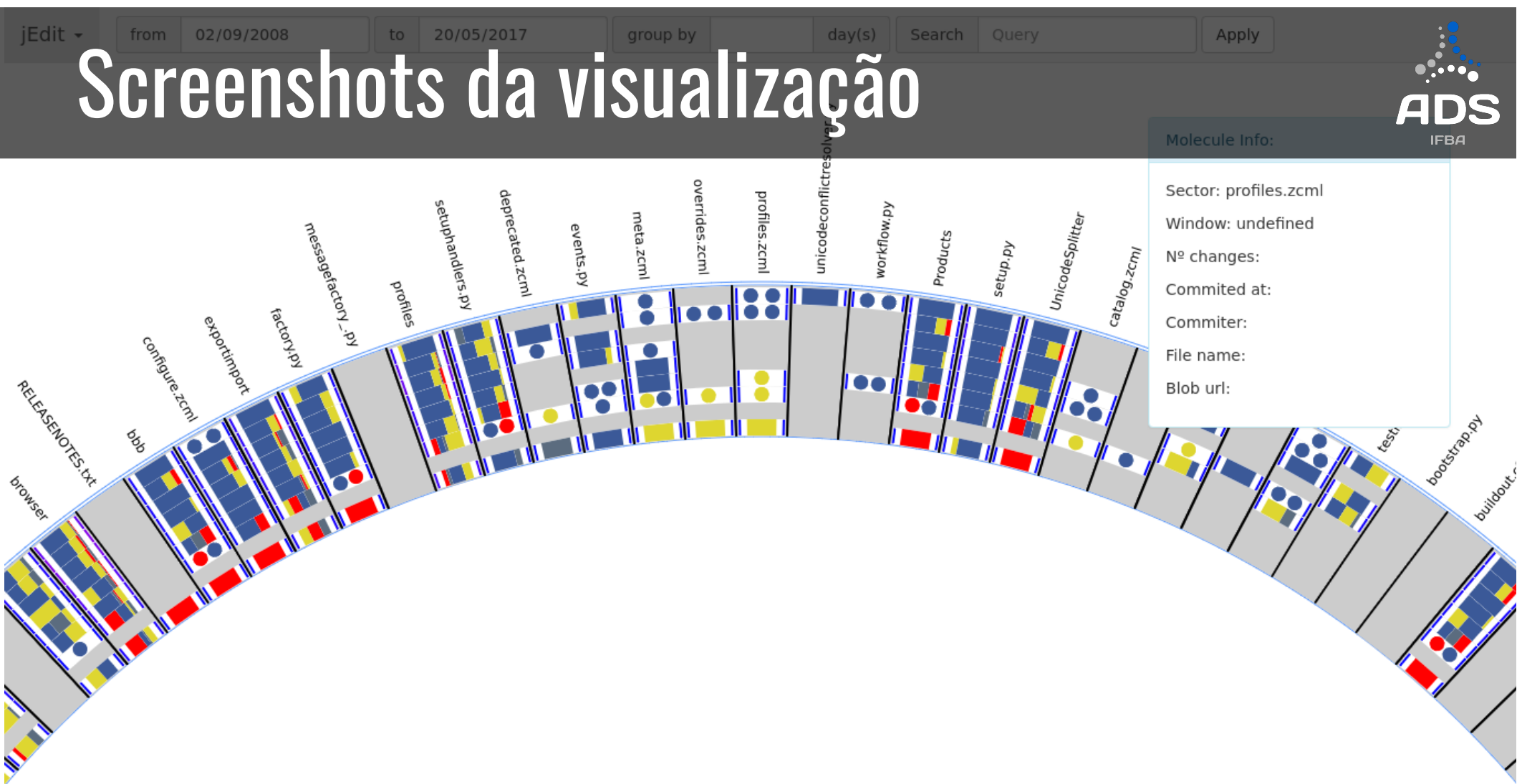
File name:

Blob url:

Plone Python CMS



Screenshots da visualização



Screenshots da visualização

Molecule Info:

Sector: profiles

Window: 9

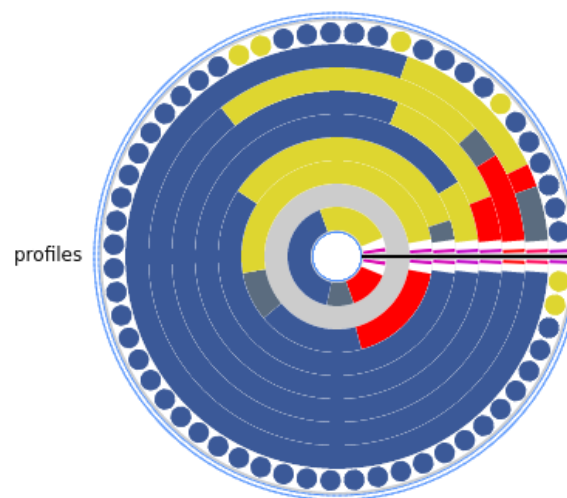
Nº changes: 12

Committed at: 11/03/2016

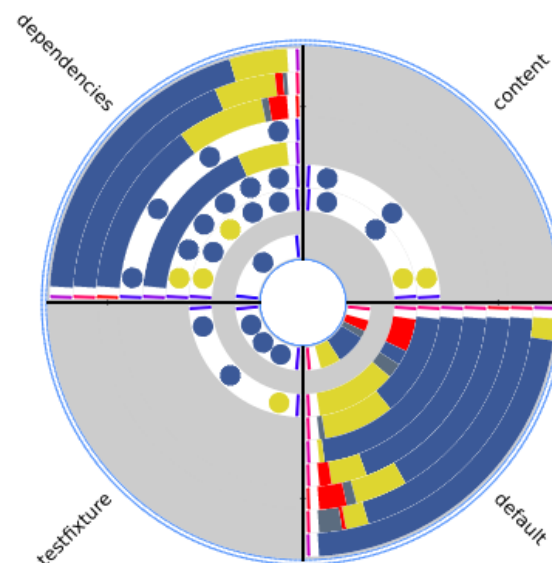
Committer: Jens W. Klein

File name: actions.xml

Blob url: <https://github.com/plone/Products.CMFPlone/blob/c531f10b2bc4a95707f4ccadc029c565de2e41d0/Products/CMFPlone/profiles/default/actions.xml>



Screenshots da visualização



Molecule Info:

Sector: default

Window: 9

Nº changes: 12

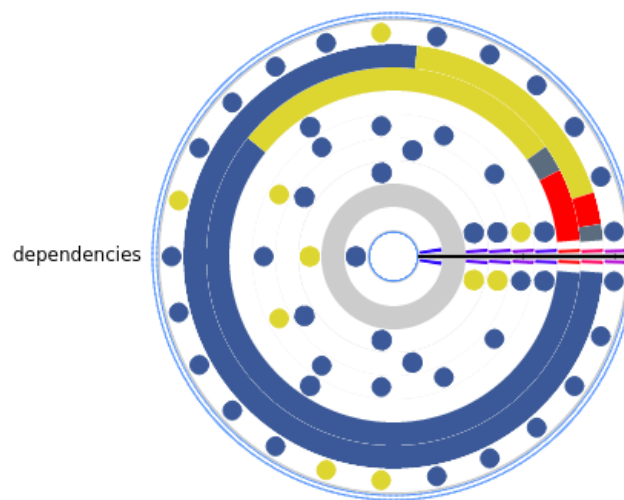
Committed at: 13/02/2016

Committer: Eric BREHAULT

File name: actions.xml

Blob url: <https://github.com/plone/Products.CMFPlone/blob/b28f02cead34148debd39560d47cae2bf414f6e8/Products/CMFPlone/profiles/default/actions.xml>

Screenshots da visualização



Molecule Info:

Sector: dependencies

Window: 9

Nº changes: 2

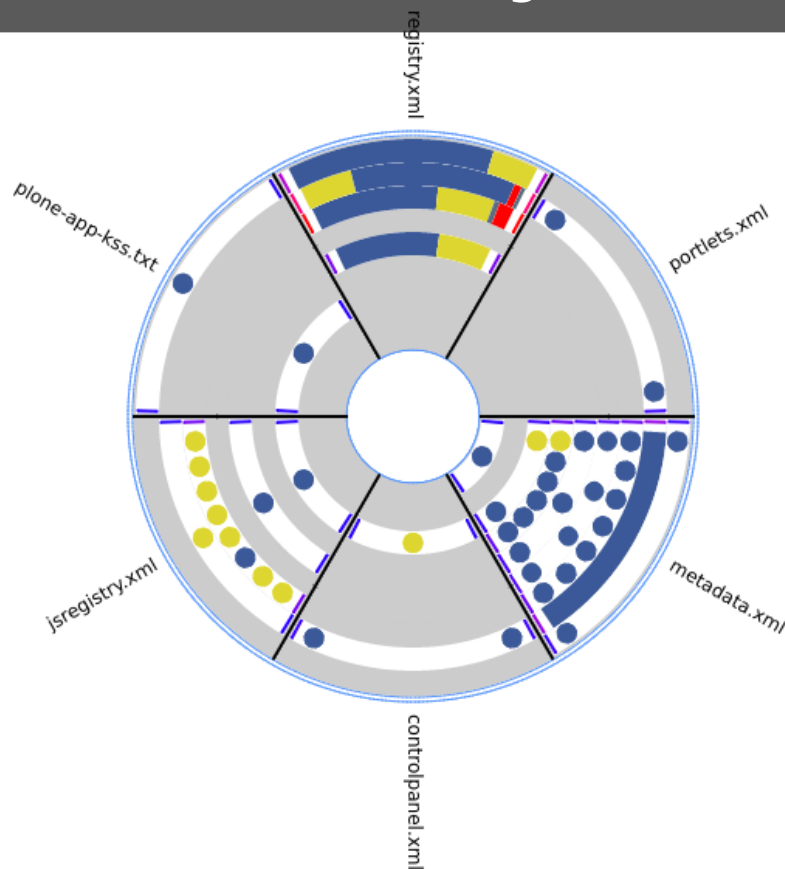
Committed at: 21/06/2016

Committer: Johannes Raggam

File name: metadata.xml

Blob url: <https://github.com/plone/Products.CMFPlone/blob/56147f7c15bddb4ae9c4573f6593f4b3e1546a86/Products/CMFPlone/profiles/dependencies/metadata.xml>

Screenshots da visualização



Molecule Info:

Sector: metadata.xml

Window: 9

Nº changes: 2

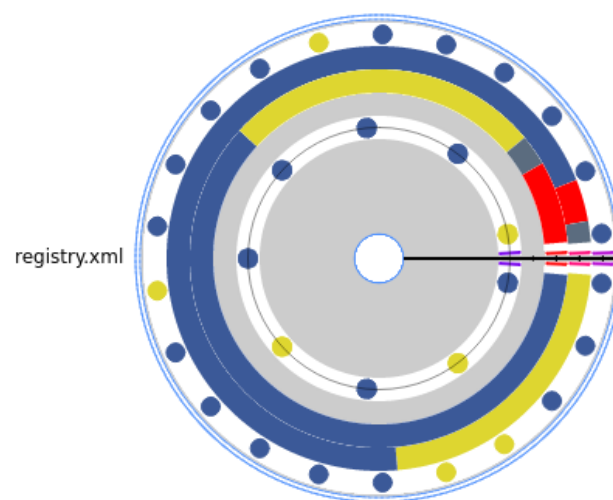
Committed at: 21/06/2016

Committer: Johannes Raggam

File name: metadata.xml

Blob url: <https://github.com/plone/Products.CMFPlone/blob/56147f7c15bddb4ae9c4573f6593f4b3e1546a86/Products/CMFPlone/profiles/dependencies/metadata.xml>

Screenshots da visualização



Molecule Info:

Sector: registry.xml

Window: 5

Nº changes: 28

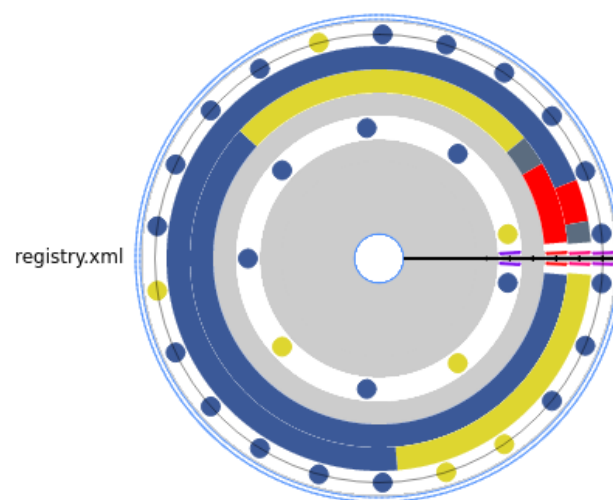
Committed at: 04/07/2012

Committer: Karl Johan Kleist

File name: registry.xml

Blob url: <https://github.com/plone/Products.CMFPlone/blob/fcf0e285c46ede20505a508f1443552e4a2dcadc/Products/CMFPlone/profiles/dependencies/registry.xml>

Screenshots da visualização



Molecule Info:

Sector: registry.xml

Window: 9

Nº changes: 5

Committed at: 08/08/2016

Committer: Johannes Raggam

File name: registry.xml

Blob url: <https://github.com/plone/Products.CMFPlone/blob/d21822f21e755a8eb9d34d70f9319e40707b1813/Products/CMFPlone/profiles/dependencies/registry.xml>

Screenshots da visualização

Molecule Info:

Sector: skins

Window: 9

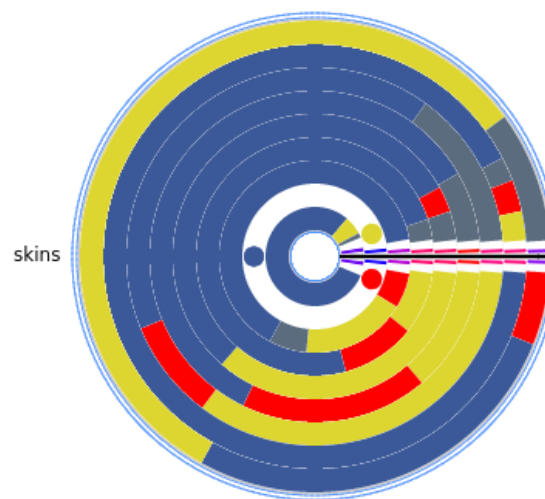
Nº changes:

Committed at:

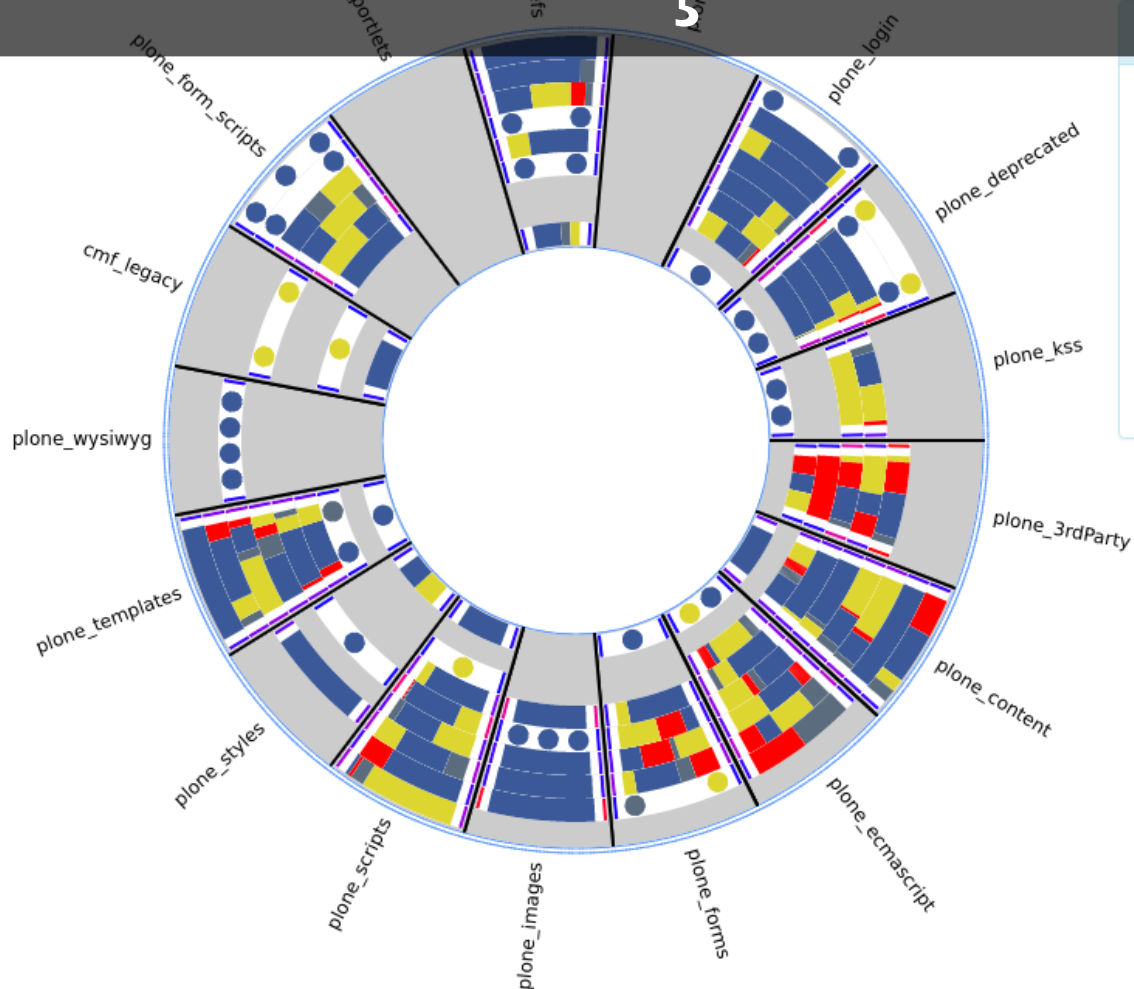
Committer:

File name:

Blob url:



Screenshots da visualização



Molecule Info:

Sector: plone_ecmascript

Window: undefined

Nº changes:

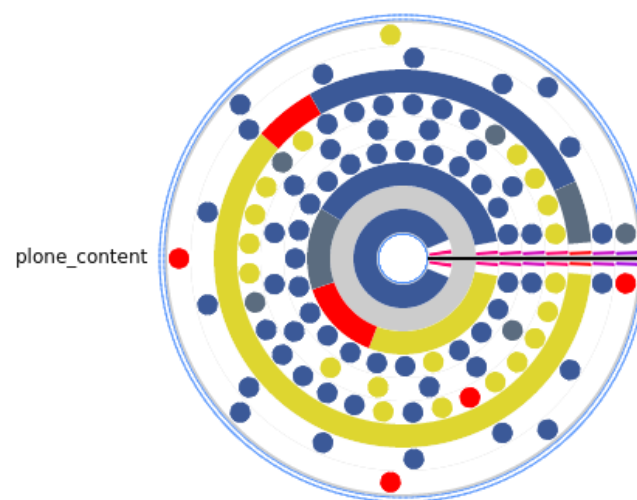
Committed at:

Committer:

File name:

Blob url:

Screenshots da visualização



Molecule Info:

Sector: link_redirect_view.py

Window: undefined

Nº changes: 79

Committed at: 20/11/2016

Committer: Gil Forcada

File name: folder_full_view_item.p
t

Blob url: https://github.com/plone/Products.CMFPlone/blob/78456286b47c29dd6409ea269450851d87fee55e/Products/CMFPlone/skins/plone_content/folder_full_view_item.pt

Screenshots da visualização

Molecule Info:

Sector: author.cpt

Window: undefined

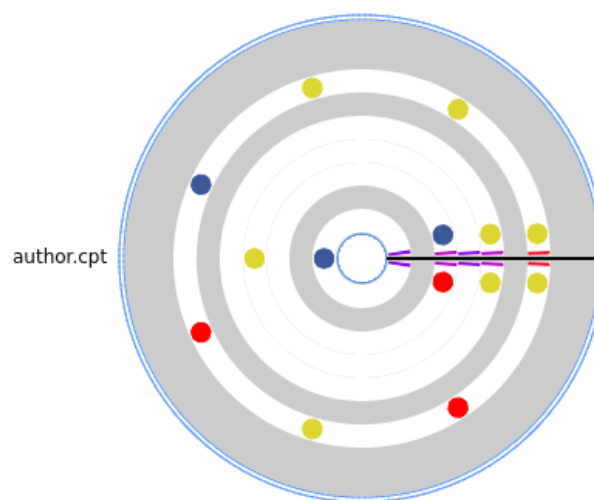
Nº changes: 2

Committed at: 17/12/2010

Committer: Laurence Rowe

File name: author.cpt

Blob url: https://github.com/plone/Products.CMFPlone/blob/a059ad3d9d75a86f2244a5b023a0d695ddae1cca/Products/CMFPlone/skins/plone_content/author.cpt



Screenshots da visualização

Molecule Info:

Sector: author.cpt

Window: 3

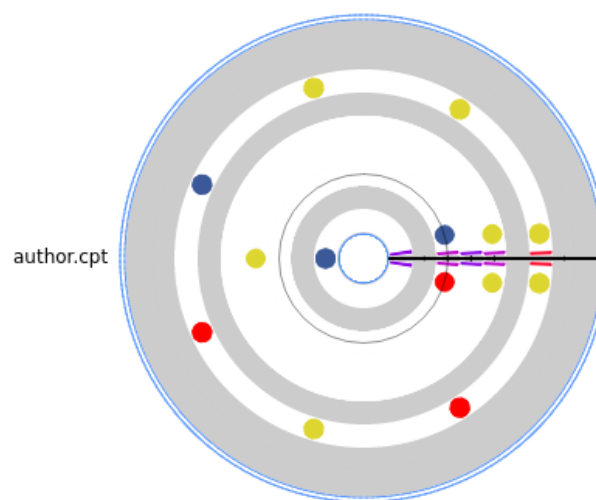
Nº changes: 294

Committed at: 27/11/2010

Committer: Laurence Rowe

File name: author.cpt

Blob url: https://github.com/plone/Products.CMFPlone/blob/854be6e30d1905a7bb0f20c66fbc1ba1f628eb1b/Products/CMFPlone/skins/plone_content/author.cpt



Análise dos dados



- Testes automatizados entraram a partir de 2010
 - Cada módulo contém arquivos de teste
- Painel de controle é um módulo que sofre alteração desde 2008
- Skeens (temas) sofreram alterações frequentemente
- Plone tem uma organização ruim de diretórios

Análise dos dados

- Arquivos com nomes fora do padrão "pythônico"
 - *TemArquivoCamelCase.py* e
separado_por_undercore.py
- Ainda usa muito arquivo xml, poderia ter migrado para json
- Mudanças no geral foram pequenas (~abaixo de 100) por arquivo

Análise dos dados

- Arquivos *static* (imagens, css, *.js) receberam poucas alterações (maioria no início)
- Plone apresenta-se como um software focado em estabilidade, com poucas alterações e atualizações.
- Menor número de bugs e melhor usabilidade



perguntas?